

Using contexts and context variables

Overview

In this module, you learned how to create and use variables to supply different values for your development and production environments.

Key steps

Follow these steps to configure Jobs and components using variables.

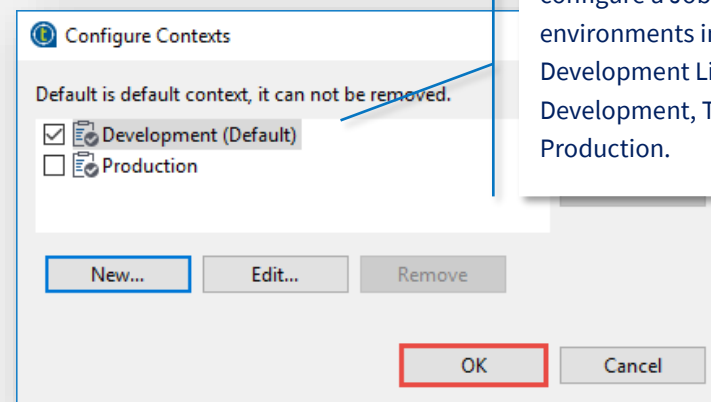
1

Manage contexts

A Job always runs in a specific context.

You can create several contexts to reflect the different configurations in which your Job should run.

- Create specific contexts in the **Context** view of the Job.
- You can create context groups in the Repository and import them to various Jobs.



Contexts are often used to configure a Job for the standard environments in the Software Development Life Cycle, such as Development, Test, and Production.

2

Create context variables

A variable takes on a value based on the context in which the Job is being run.

Context variables are assigned values either statically (in the Value text box) or interactively (you enter them using the Prompt option).

- Create built-in contexts and variables in the **Context** view of the Job.
- You can also create variables in the **Repository** within a context group.

Name	Development		Production
	Value		Value
FilePath	C:/StudentFiles/DI Basics/ContextVariables/DEV/		C:/StudentFiles/DI Basics/ContextVariables/PROD/

When creating a variable; you must include a name, a type, and values for the different contexts.

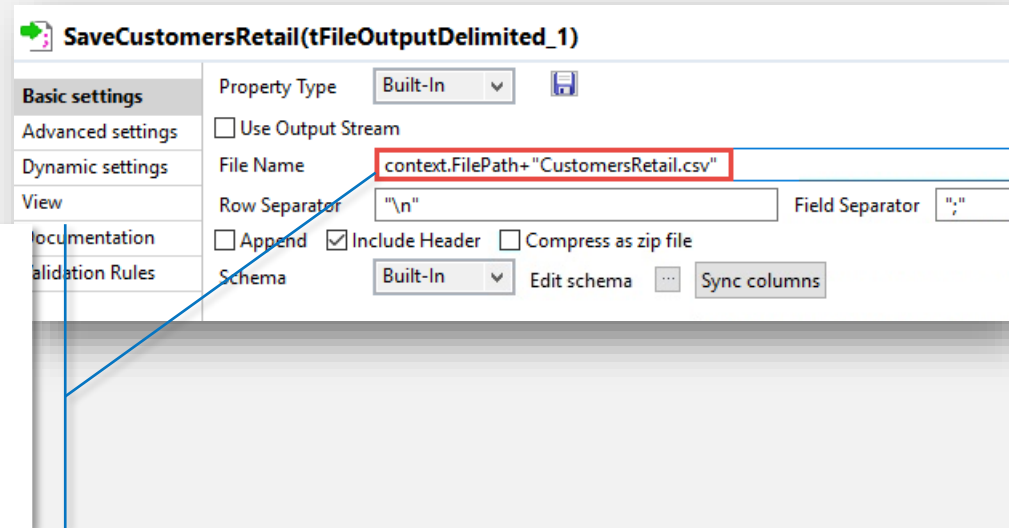
3

Use variables when configuring components

- To access a context variable, use the **context.VariableName** construction.

To take advantage of the auto completion feature, place the cursor in the parameter text box and press **Ctrl+Enter**.

To display available variables, start entering *context*, then select one.

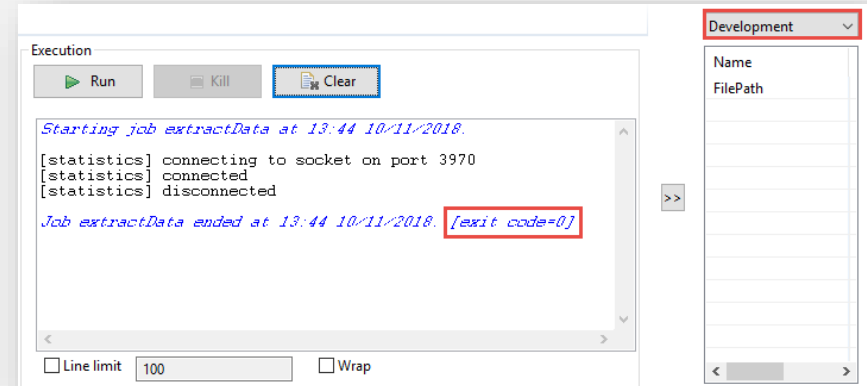


4

Run a Job with different configurations by changing the execution context.

Specify the execution context in the **Run** view.

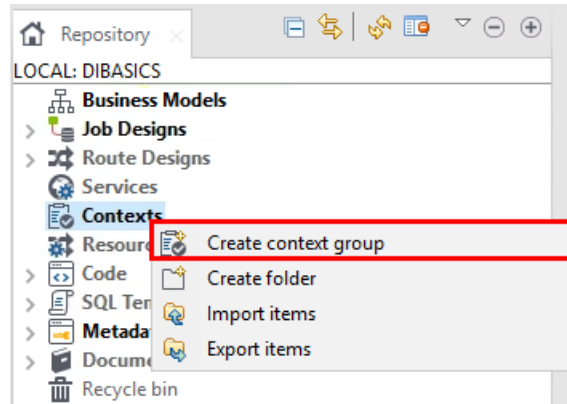
- The context, variables, and values are displayed
- When you change the context, values are updated but cannot be changed



Tips

Follow these guidelines to configure variables and context groups in the Repository.

Creating a context group



You can use context groups that you create in the Repository in all Jobs in the same project.

- To activate a context group for a specific Job, drag it from the **Repository** to the **Contexts** view of the Job.

Exporting variables as a context group

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

Default context environment: Default

You can export connection metadata as a context group. This way, you can set up a connection to two different databases for the Development and Production contexts.

- To export metadata as a context, edit the connection metadata and click the **Export as context** button.

Best practice

Always use the same context names (Development, Production, or whatever suits your needs). This helps when merging contexts from different sources (Built-In or Repository) for the same Job.

Document your settings by systematically using the **Comment** column in the contexts table.

Never activate a prompt for the Production context, as the Job must run in the production environment without human intervention.