FREE eBook

LEARNING talend

Free unaffiliated eBook created from
Stack Overflow contributors.

#talend

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: talend

It is an unofficial and free talend ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official talend.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with talend

Remarks

This section provides an overview of what talend is, and why a developer might want to use it.

It should also mention any large subjects within talend, and link out to the related topics. Since the Documentation for talend is new, you may need to create initial versions of those related topics.

Examples

Installation or Setup

Detailed instructions on getting talend set up or installed.

Read Getting started with talend online: https://riptutorial.com/talend/topic/9503/getting-started-with-talend

Chapter 2: Connectiong Components

Examples

If / OnComponent / OnSubjob

There are 2/3 options to connect components together in Talend. You should always try to use OnSubjob connectors. This saves a lot of headaches. You'll see from the examples why.

What happens when you mix the connection types / What is the execution order?

1. If
2. OnComponent
3. OnSubjob

Keep in mind that the If connections gets evaluated runtime, which means if you use globalMap then be really careful about the order.

For example:

`(Boolean)globalMap.get("failure") == true` -> calls a subjob that resets this failure flag.
`(Boolean)globalMap.get("failure") == false` -> calls a subjob that lets the main job continue, because the failure path resetted the flag.

What is the a difference?

If and Oncomponent connections act as a function call. Which makes the Garbage collector to keep all the local data stored in memory. This could cause "memory leaks".

OnSubjob connections on the other hand let the subjob complete and return, thus the GC will free up some of the memory.

Other than the memory there's also a few things you need to keep in mind. If you have a data flow that reads from / writes to file, you should always go with OnSubjobOk as the data file will be closed once the job completes. If you use onComponent it can happen that the file is not saved hence you start working with a 0 byte file, and after the job completes you'll see a file with content. It's logical but really hard to figure out.

Read Connectiong Components online: https://riptutorial.com/talend/topic/9542/connectiong-components

Chapter 3: Types conversion in Talend

Introduction

A list of type conversion in talend with some examples.

Examples

Table of Conversions

From	To	Example
String	Integer	Integer.parseInt(str) OR Integer.valueOf(str).intValue()
String	Date	TalendDate.parseDate("dd-MM-yyyy", str)
String	BigDecimal	new BigDecimal(str)
String	Float	Float.parseFloat(str) OR Float.valueOf(str).floatValue();
String	Long	Long.parseLong(str) OR long l = Long.valueOf(str).longValue()
String	Double	double d = Double.valueOf(str).doubleValue()
Date	String	TalendDate.formatDate("yy-MM-dd", row1.myDate)
Float	String	row1.myFloat.toString()
Float	BigDecimal	new BigDecimal(Float.toString(row1.myFloat))
Float	Double	(float)d
Float	Integer	First round : Math.round(), Math.ceil(), Math.floor() then cast the result to Integer
Long	Int	(int)(row1.var + 0) The max possible value is 2147483647
Long	String	row1.myLong.toString
Integer	Long	row1.myInteger.longValue()
Integer	BigDecimal	new BigDecimal(row1.myInteger)
Integer	Float	new Float(row1.myInteger)
Integer	String	variable+"" OR variable.toString()
BigDecimal	Integer	As with Float, BigDecimal can have decimal places, so will need to be rounded prior to casting to Integer
BigDecimal	String	row1.myBigDecimal.toString()
Double	String	String str = Double.toString(d)
Double	Float	double d = f

Read Types conversion in Talend online: https://riptutorial.com/talend/topic/9516/types-conversion-in-talend

Chapter 4: Using Date in Talend

Examples

Parsing a date

Parsing date is used when having an input typed as `String` and when it is needed to get it as a `Date`. The class `TalendDate` contains method `TalendDate.parseDate("pattern","stringDate")`.

Pattern here is the **input** pattern, and not the expected output pattern.

Usage : For an input string like `"2017-05-03 17:09:00"` , the call will be :

```
TalendDate.parseDate("yyyy-MM-dd HH:mm:ss","2017-05-03 17:09:00")
```

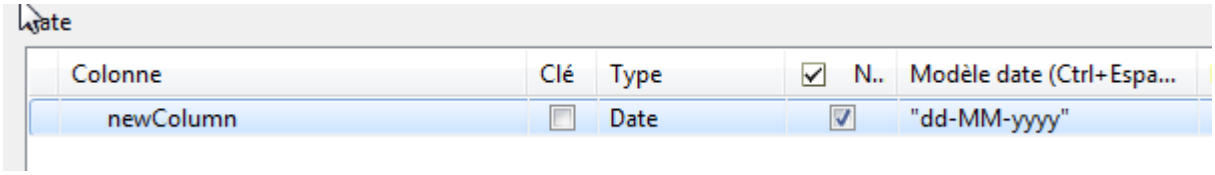
The result could be a date like :

```
2017-05-03 17:09:00
```

or

```
03/05/2017
```

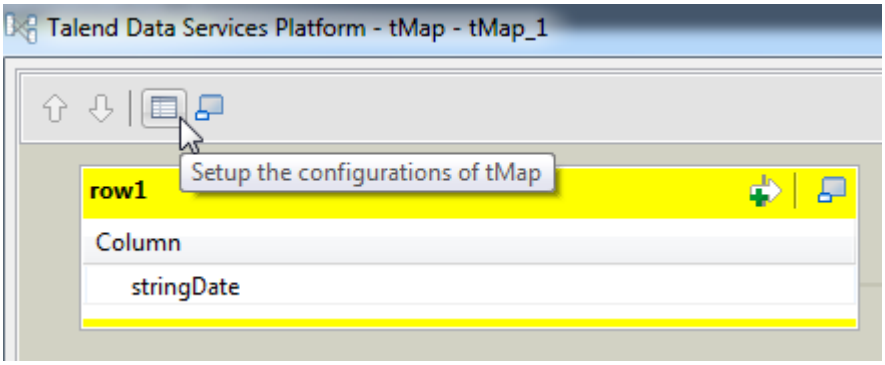
Depending on the **output pattern** which is defined outside the parseDate method.

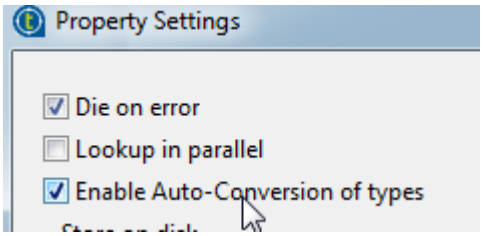


Automatic date parsing

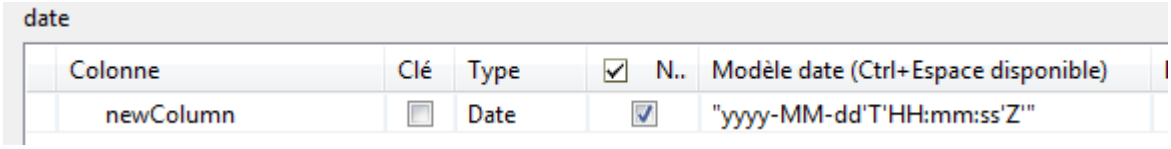
Since Talend 6.3 , an option in tMap allows to automatically convert types. When activated, output pattern is used as the expected input pattern to automatically convert data.

First, activate the option :





Then modify output pattern, used as the input pattern :



Read Using Date in Talend online: https://riptutorial.com/talend/topic/9873/using-date-in-talend

Credits

S. No	Chapters	Contributors
1	Getting started with talend	Community
2	Connectiong Components	Balazs Gunics
3	Types conversion in Talend	Théo Capdet
4	Using Date in Talend	Corentin