

## Pravan kumar VASAM

**Project task :** Fichier de liste dans HDFS

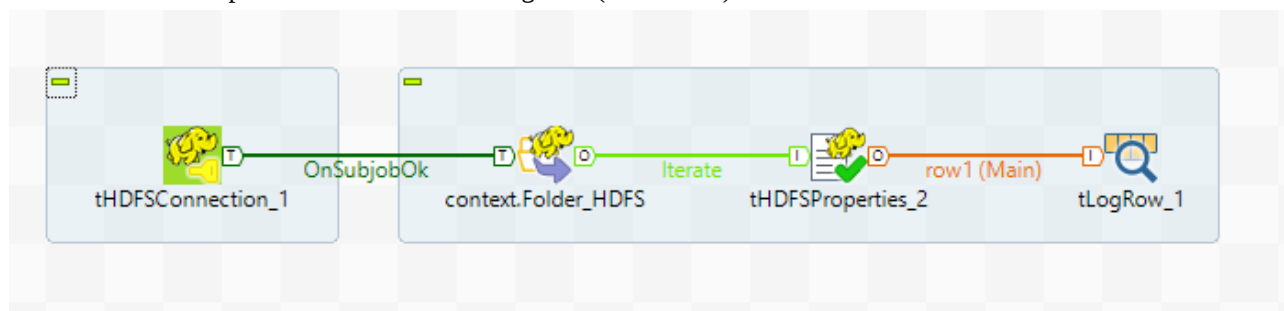
**Technology:** Talend

Créer un nouvel

- ➔ Ajouter le composant "tHDFSConnection": Permet la création d'une connexion HDFS.
- ➔ Ajoutez le composant "tHDFSList": Listez les différents contenus des fichiers dans le dossier hdfs.
- ➔ Ajouter le composant "tHDFSProperties": afficher les propriétés des différents fichiers (exemple: mode, heure, nom du répertoire ...)
- ➔ Ajoutez le composant "tLogRow ": affichez le résultat.

Créer des liens:

- ➔ "tHDFSConnection" est connecté à "tHDFSList" (via "OnSubjobOk")
- ➔ "tHDFSList" est connecté à "tHDFSProperties" (via "Iterate")
- ➔ "tHDFSProperties" est connecté à "tLogRun" (via "Main")



**Double-cliquez sur "tHDFSConnection" et définissez ses propriétés:**

- ➔ Ajoutez une distribution "Cloudera" et sélectionnez la dernière version de Cloudera
- ➔ Saisissez l'URL du nœud de nom.  
L'URL doit respecter ce format: "hdfs:// ip\_hdfs: port\_hdfs /"  
Utilisez des variables de contexte si possible: "hdfs: //" + context.IP\_HDFS + ":" + context.Port\_HDFS + "/"
- ➔ Ajouter l'utilisateur

**tHDFSConnection\_1**

Type de propriété: Built-In

Version: Cloudera CDH5.5(YARN mode)

URI du NameNode: "hdfs://" + context.IP\_HDFS + ":" + context.Port\_HDFS + "/"

Configurations: ☐ Inspect the classpath for configurations

Authentication: ☐ Utiliser l'authentification de Kerberos

Utilisateur: context.User\_HDFS

Propriétés Hadoop:

Propriété	Valeur
-----------	--------

☒ Use Datanode Hostname

**Double-cliquez sur "tHDFSList" et définissez ses propriétés:**


- ➔ Cochez "Utiliser une connexion existante" et sélectionnez la connexion établie par le composant "tHDFSConnection"
- ➔ Ajouter un dossier hdfs: context.Folder\_HDFS
- ➔ Ajoutez un masque de fichier.

Dans l'exemple, le masque de fichier est "\*" car ce travail recherche chaque fichier.

Si vous souhaitez rechercher uniquement les fichiers se terminant par l'extension ".csv", vous pouvez saisir "\*.csv".

L'étoile signifie "peu importe" avant ".csv".

➔ "Trier", sélectionnez "Nom du fichier"

 **context.Folder\_HDFS(tHDFSList\_1)**

**Paramètres simples**  
Advanced settings  
Paramètres dynamiques  
View  
Documentation

☒ Utiliser une connexion existante

Liste des composants

tHDFSConnection\_1 ▼ \*

Répertoire HDFS

context.Folder\_HDFS

Type de fichier dans la liste FileList

Files ▼

☐ Inclure les sous-répertoires

Sensible à la casse







Oui ▼

☒ Utiliser des Expressions Globales comme masque de fichier (Décocher la case signifie utilise

Files

Filemask

\*\*\*



Trier par

☐ Par défaut ☒ Par nom de fichier ☐ Par taille de fichier ☐ Par date de modification

Action de tri


☒ ASC ☐ DESC

Double-cliquez sur "tHDFSProperties":

➔ Cochez "Utiliser une connexion existante"

➔ Ajouter un fichier: ((String) globalMap.get ("tHDFSList\_1\_CURRENT\_FILEPATH"))

Cette commande utilise le fichier courant du composant tHDFS\_List.

 **tHDFSProperties\_2**

**Paramètres simples**  
Advanced settings  
Paramètres dynamiques  
View  
Documentation

☒ Utiliser une connexion existante

Liste des composants

tHDFSConnection\_1 ▼ \*

Schéma

Built-In ▼

Editer le schéma ...

File

((String)globalMap.get("tHDFSList\_1\_CURRENT\_FILEPATH"))

☐ Get file checksum