



Talend Component tGoogleDrive

Purpose and procedure

This component manages files on a Google Drive.

The component provides these capabilities:

1. Providing only the client for other tGoogleDrive components
2. List file on the Drive with various query parameters
3. Upload a file to the Drive (also in a particular folder on the Drive including update existing ones)
4. Download a file from the Drive (also from a particular folder)
5. Get the properties of a file on the Drive
6. Delete a file on the Drive
7. Move a file within the Drive from one folder to another

Talend-Integration

This component can be found in the palette under Cloud->Google

This component provides an input flow and several return values (depending on the operational mode).

Because of the very different functionality of this component in the different modes all modes are described with all aspects in separate chapters.

Parameters and Usage

There are 3 different functionalities, which have to choose with the operational mode switch.

Property	Content
Operational Modes	<p>Switches between the different modes the component provides.</p> <p>Establish only the client (CLIENT): In this mode the component acts like a typical connection component and connects to the Google Drive and provides this client for other components.</p> <p>List files (LIST): In this mode the component lists files by various query parameters.</p> <p>Retrieve file properties (PROPERTIES): In this mode the component retrieves the file properties. This is the only mode, which can be used to check the existence of a file.</p> <p>Upload a file (UPLOAD): In the mode the component uploads a file to the Drive (possible into a particular folder or simply in the root folder)</p> <p>Download a file (DOWNLOAD): In this mode the component downloads a file to the local file system.</p> <p>Delete a file (DELETE): In this mode the component deletes a file or folder permanently (it does not use the trash bin!)</p> <p>Move a file (MOVE): In this mode the component moves a file within the Drive without creating a copy.</p>

To distinguish between the different modes of the components in the job I recommend enhancing the view settings to show the mode next to the unique name in the label:

tGoogleDrive_2: PROPERTIES(tGoogleDrive_2)

Basic settings | Label format: __UNIQUE_NAME__: __MODE__

Advanced settings | Hint format: __UNIQUE_NAME__
__COMMENT__

Dynamic settings | Connection format: row

View

Documentation

Parameters to establish the client and connect to Google Drive

These settings will be used in all modes and therefore explained only once here.

Google suggest 2 different modes for authentication for backend processes and native applications:

1. Service Account: A service account is actually a new account and has the advantage it does not need any user interaction while the job runtime. If the job is supposed to manage real person drives, this mode actually does not help because the files have as owner the service account and you cannot access them directly.
2. Client-ID for native applications: A client-Id is needed if you want managing files on real person accounts. The disadvantage is, it depends on a user interaction (only for the first time any arbitrary job using this account runs the first time, all other jobs using the same account does not need any interaction anymore).

Property	Content	Data types
Application Name	Not necessary, but recommended by Google. Simple provide the name of your application gathering data. Required	String
Use existing client	Choose here the tGoogleDrive component which client do you want to reuse in this component instance.	Boolean
Authentication Method	Choose the method to authenticate: Service Account or Client-ID for native applications	String

Properties to use the service account

Property	Content	Data types
Service Account Email	The email address of the service account. Google creates this address within the process of creating a service account. Only for service accounts! Required	String
Key File (p12)	The Service Account Login works with private key file for authentication. In the process of creating a service account you download this file. Only for service accounts Required	String

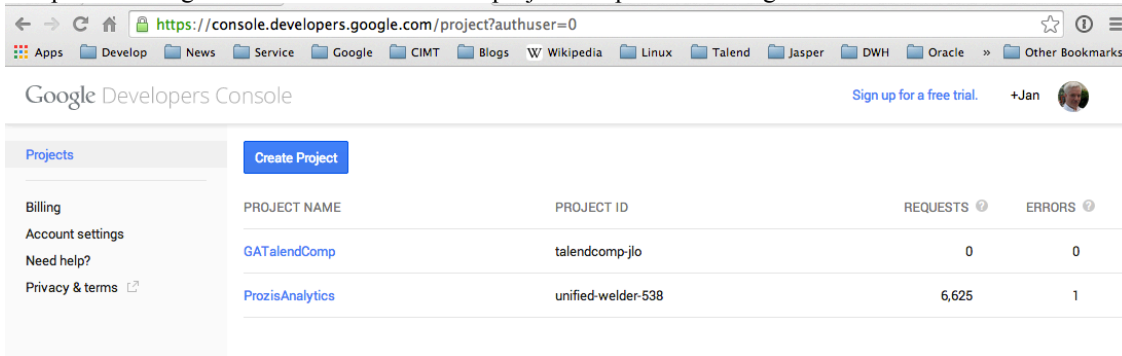
Properties to use the Application Client-ID authentication (if the option service account is switched off)

Property	Content	Data types
User Account Email	Email of the user account or the Client-ID	String
Client secret file (json)	This json file downloaded for the Client-ID	String

The usage of the “Client-ID for native applications” expects on the first run an user interaction with the Google web page and after finishing the form to approve the access right you need to close the browser to let the component continue, otherwise the authentication process will not complete.

How to enable and authenticate the drive client

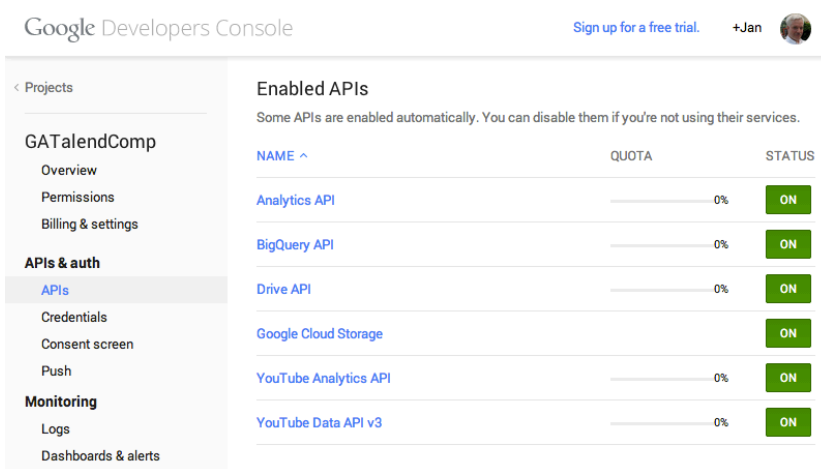
1. Open the Google API console and create a project or open an existing one



The screenshot shows the Google Developers Console interface. The left sidebar contains a menu with 'Projects' selected. The main area displays a table of projects. The table has columns for 'PROJECT NAME', 'PROJECT ID', 'REQUESTS', and 'ERRORS'. Two projects are listed: 'GATalendComp' with project ID 'talendcomp-jlo' and 'ProzisAnalytics' with project ID 'unified-welder-538'.

PROJECT NAME	PROJECT ID	REQUESTS	ERRORS
GATalendComp	talendcomp-jlo	0	0
ProzisAnalytics	unified-welder-538	6,625	1

2. Enable the Drive API for your project

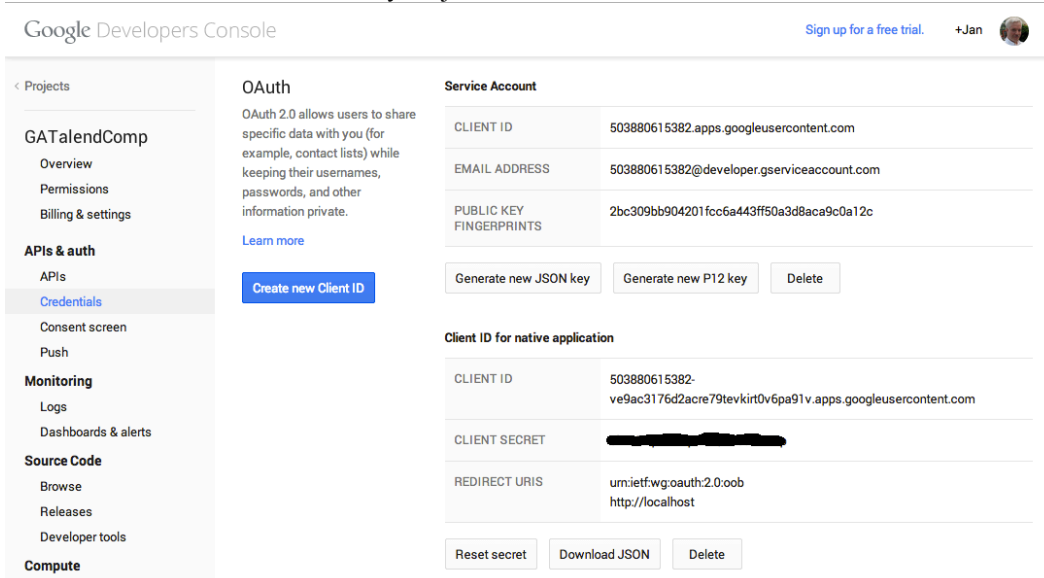


The screenshot shows the 'Enabled APIs' page in the Google Developers Console. The left sidebar shows the 'APIs & auth' section selected. The main area displays a table of enabled APIs. The table has columns for 'NAME', 'QUOTA', and 'STATUS'. Six APIs are listed: 'Analytics API', 'BigQuery API', 'Drive API', 'Google Cloud Storage', 'YouTube Analytics API', and 'YouTube Data API v3'. All APIs have a status of 'ON'.

NAME	QUOTA	STATUS
Analytics API	0%	ON
BigQuery API	0%	ON
Drive API	0%	ON
Google Cloud Storage		ON
YouTube Analytics API	0%	ON
YouTube Data API v3	0%	ON

Here an example with many enabled API. The important API here is the Drive API.

3. Create credentials to use them in your job



The screenshot shows the 'OAuth' and 'Service Account' pages in the Google Developers Console. The left sidebar shows the 'APIs & auth' section selected. The main area displays two sections: 'OAuth' and 'Service Account'. The 'OAuth' section shows a 'Create new Client ID' button. The 'Service Account' section shows a table of service account details, including 'CLIENT ID', 'EMAIL ADDRESS', and 'PUBLIC KEY FINGERPRINTS'. Below the table are buttons for 'Generate new JSON key', 'Generate new P12 key', and 'Delete'. The 'Client ID for native application' section shows a table of client ID details, including 'CLIENT ID', 'CLIENT SECRET', and 'REDIRECT URIS'. Below the table are buttons for 'Reset secret', 'Download JSON', and 'Delete'.

CLIENT ID	EMAIL ADDRESS	PUBLIC KEY FINGERPRINTS
503880615382.apps.googleusercontent.com	503880615382@developer.gserviceaccount.com	2bc309bb904201fcc6a443ff50a3d8aca9c0a12c

CLIENT ID	CLIENT SECRET	REDIRECT URIS
503880615382-ve9ac3176d2acre79tevkirt0v6pa91v.apps.googleusercontent.com	[REDACTED]	urn:ietf:wg:oauth:2.0:oob http://localhost

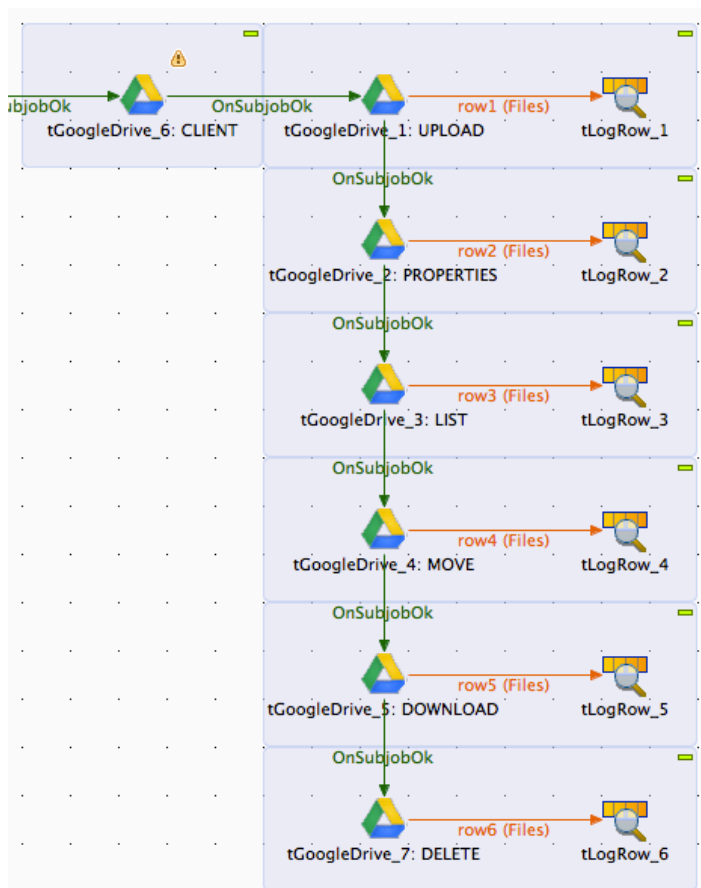
Here an example of both ways: Service Account and Client ID for native applications.

Explanation for all modes

All different modes are explained here by an example job using all modes.

Here is the job design of the example job and it does following:

1. Memorize the current time stamp (not visible in a tJava) in a context variable date
2. Connect to the Drive
3. Upload a file to a folder made by the current minute time stamp
4. Fetch the properties of the uploaded file
5. List all files in a folder for one owner
6. Moves the file into another folder
7. Downloads the file
8. Delete the file in the Drive



Job design of the example job. The settings for the components are explained in the next paragraphs.

Operational Mode: Establish only the client (CLIENT)

In this mode the component creates only the client for other components. This is the preferred way if you have more than one tGoogleDrive components in your job to avoid setting up each one separately and to avoid connecting too often.

Basic settings in the example job:

tGoogleDrive_6: CLIENT(tGoogleDrive_6)

Basic settings | Operational Mode: Establish only the Client

Client Setup

Application Name: "Google Drive Talend Job"

Authentication Method: Client ID for native application

Client-ID or user email: "jan.lolling@gmail.com"

Client Secret File (json): "/Volumes/Data/Talend/testdata/ga/config/client_secret_503880615382-ve9ac3176d2acre79tevkirt0v6pa91v.apps.googleusercontent.com.json"

Here an example with the client-ID.

If you run a job first time with this Client-ID the job opens a browser and shows a dialog with your identity and the necessary right to work on the drive. Accept it and – very important – close the browser.

The component keeps all information in a directory next to the json file. You can move both to any other location the job should run in a productive environment.

The usage for a service account does not need such user interaction. It simply runs fine and use only the key file.

The settings are already explained in the section about establishing a client.

The advanced settings only apply to clients. If a component uses an existing client they do not apply.

Property	Content
Timeout in s	How long should the component wait for getting the first result and fetching all result with one internal iteration
Static Time Offset (to past)	Within the process of login, the component requests an access token and use the local time stamp (because these tokens will expire after a couple of seconds) Google rejects all requests to access tokens when the request is in the future compared to the timestamp in Google servers. If you experience such kind of problems, this options let the requests appear to be more in the past (5-10s was recognized as good time shift)
Reuse Client for Iterations	If you use this component in iterations it is strongly recommended to set this option. It saves time to authenticate unnecessary often and avoids problems because of max amount of connects per time range.
Distinct Name Extension	The client will be kept with an automatically created name: Talend-Name-Component name + job name. In case this is not distinct enough, you can specify an additional extension to the name.

Operational Mode: Upload a file (UPLOAD):

In this mode the component uploads a file and set as owner for the file the current user. It can also set the permissions on this new file and allow others users to read/write the file (also being notified per email).

Basic settings in the example job:

tGoogleDrive_1: UPLOAD(tGoogleDrive_1)

Operational Mode: Upload a file

Client Setup

☒ Use existing client Drive Component: tGoogleDrive_6: MODE

File

Title of the file in the Drive: "Test file " + TalendDate.formatDate("yyyyMMdd_HH:mm", context.date) + ".pdf"

Folder within the Drive: "/Test/" + TalendDate.formatDate("yyyyMMdd_HH:mm", context.date) ☒ Create folders in Drive in necessary

Local file to upload: "/Volumes/Data/Talend/testdata/ga/drive/2008-02-14-REST---JUG-Berlin.pdf"

☒ Overwrite ☐ Delete source file in file system

Set permission as reader (email addresses): "jan.lolling@cimt-ag.de" ☒ Send email notification

Set permission as writer (email addresses): "doris.lolling@gmail.com,jan.lolling@posteo.de" ☒ Send email notification

Schema Files: Built-In Edit schema

The file is addressed here directly; the local file to upload can also be set by a context variable.

This component uses the existing client from another component.

The upload shows on the console some status information about the upload.

If none Drive-folder is set the file will be placed into the root folder.

Drive expects for uploads to set the mime-type of the file. This will be done automatically by the component. To do that the component ships with the latest mime-type definitions from the Apache web server. If the file extension is completely unknown, the mime-type "plain/text" will be used.

The component returns the meta-data of the uploaded file as flow with a fixed defined schema.

Column	Key	Type	<input checked="" type="checkbox"/> Nullab	Date Pattern (Ctrl+Space avail)	Length	Precision	Default	Comment
FILE_ID	<input checked="" type="checkbox"/>	String	<input checked="" type="checkbox"/>		64	0		ID of the file
FILE_TITLE	<input checked="" type="checkbox"/>	String	<input checked="" type="checkbox"/>		128	0		Title of the file
ORIGINAL_FILE_NAME	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		255	0		Original name of the uploaded file
FILE_EXTENSION	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		16	0		File extension without the dot
FILE_CREATED_AT	<input type="checkbox"/>	Date	<input checked="" type="checkbox"/>	"yyyy-MM-dd HH:mm:ss"	22	0		
FILE_LAST_MODIFIED_AT	<input type="checkbox"/>	Date	<input checked="" type="checkbox"/>	"yyyy-MM-dd HH:mm:ss"	22	0		
FILE_MIME_TYPE	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		64	0		Mime type of the file
FILE_DOWNLOAD_URL	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		255	0		URL to download the file with the API
FILE_WEB_CONTENT_LINK	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		255	0		URL for users to download the file
FILE_SIZE	<input type="checkbox"/>	Long	<input checked="" type="checkbox"/>		19	0		File size in byte
FILE_OWNERS	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		255	0		Pipe separated list of the own names
FILE_IS_FOLDER	<input type="checkbox"/>	Boolean	<input checked="" type="checkbox"/>		0	0		true if the file is a folder

This schema is available in all modes except in the mode CLIENT.

All of these variables also available as return values of the component (also not in the CLIENT mode).

Operational Mode: Delete a file (DELETE):

In this mode the component deletes a file. The file can be addressed as file-id or as file path in the Drive.

The component deletes a file or folder permanently and does not use the trash bin.

The component can ignore missing files.

Basic settings in the example job:

tGoogleDrive_7: DELETE(tGoogleDrive_7)

Operational Mode: Delete a file

Client Setup

☒ Use existing client Drive Component: tGoogleDrive_6: MODE

File

File-Id in the Drive: ((String)globalMap.get("tGoogleDrive_4_FILE_ID"))

File path in the Drive:

☒ Ignore missing file

Schema Files: Built-In Edit schema

The component returns a flow with the meta-data of the deleted file.

Operational Mode: Retrieve file properties (PROPERTIES):

In this mode the component returns the file properties. Additionally it sets a return value if the file exists or not. This is the only node with can handle missing files.

Basic settings in the example job:

tGoogleDrive_2: PROPERTIES(tGoogleDrive_2)

Basic settings

Operational Mode: Retrieve file properties ▼*

Client Setup

☒ Use existing client Drive Component: tGoogleDrive 6: MODE ▼*

File

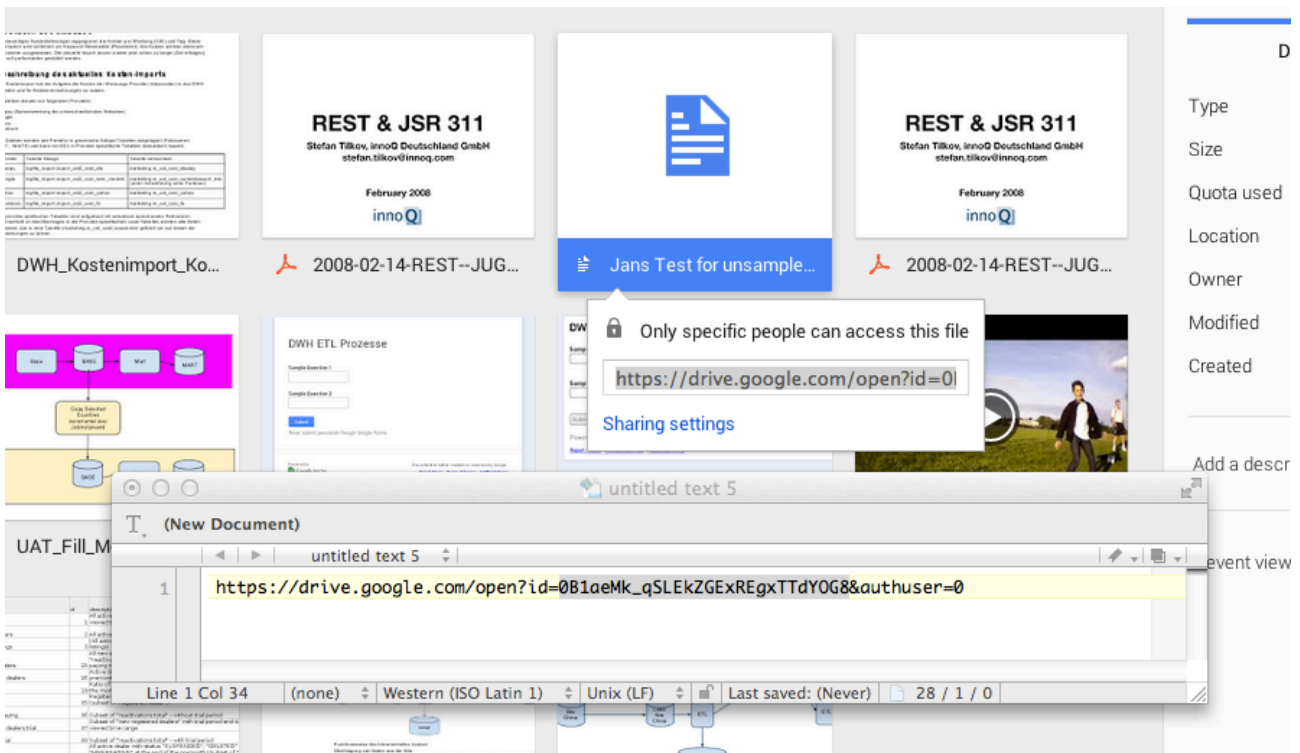
File-Id in the Drive: [Empty text box]

File path in the Drive: "/Test/" + TalendDate.formatDate("yyyyMMdd_HH:mm", context.date) + "/2008-02-14-REST--JUG-Berlin.pdf"

Schema Files: Built-In ▼ Edit schema [Button]

The component can use the file-Id or the file path within the drive.

Here is a way how to figure out the file-Id in the Google Drive web application:



Start an attempt to share the file and copy the link into a text editor.

The file-id (in this example: 0B1aeMk_qSLEkZGExREgxTTdYOG8) is the value of the URL parameter id and goes until the next & (start of the next parameter).

Operational Mode: List files (LIST):

In this mode the component lists the meta-data of the files of the drive. There are a couple of possible filters.

Property	Content	Data types
Local file filter	This works locally. The received files will be filtered with regularly expressions.	String
Case sensitive	Set this option to apply the local file filter case sensitive	Boolean
API Query String	If the available options to filter are not enough, here you can set additional filter conditions in the way the API expect them. It is highly recommended to test them in the Google API Explorer for Google Drive.	String
Remote Filter: Title starts with	Set the string with the title of the file start with. It is case sensitive.	String
Remote Filter: File modified at (Start time range)	Set here the date to filter files which are modified after/equal the given date	Date
Remote Filter: File modified at (End time range)	Set here the date to filter files which are modified equal/before the given date	Date
Search in folder	Set here the Drive folder name. The list method will fail if the folder does not exist.	String
Owner email	Set here the email address of the owners. Only files which have as owner this user will part of the list here.	String
Include folders them self	If true the folder objects are in the list, otherwise only files will be listed.	Boolean

Basic settings in the example job:

tGoogleDrive_3: LIST(tGoogleDrive_3)

Basic settings

Operational Mode: List files

Client Setup

☒ Use existing client Drive Component: tGoogleDrive_6: MODE

File

Local File Filter (regex)

☐ Case sensitive filter

API Query String

Remote Filter: Title start with: "2008"

Remote Filter: Full Text Search

Remote Filter: File modified at (start time range)

Remote Filter: File modified at (end time range)

Search in Folder: "/Test/" + TalendDate.formatDate("yyyyMMdd_HH:mm", context.date)

Owner email: "jan.lolling@gmail.com"

☒ Include folders them self in the list result

Schema Files: Built-In Edit schema

Here an example of list files where the title starts with “2008” and are in a folder dynamically build with the current minute and an owner.

Operational Mode: Move a file (MOVE):

In this mode the file will be moved from the current folder(s) to another folder. It is important to know a file can be located in different folders similar. This function removes all references to the folders and set a new reference to another parent folder.

The file can be addressed by the file-Id of the file path (folder + file-title).
The target folder has to set as name.

Basis settings in the example job:

The screenshot shows the configuration window for the 'tGoogleDrive_4: MOVE(tGoogleDrive_4)' component. The 'Operational Mode' is set to 'Move a file to another folder'. Under 'Client Setup', 'Use existing client Drive Component' is checked, and 'tGoogleDrive 6: MODE' is selected. The 'File' section contains the following fields: 'File-Id in the Drive' with the value '((String)globalMap.get("tGoogleDrive_1_FILE_ID"))', 'File path in the Drive' (empty), and 'Folder within the Drive' with the value '"/Test/" + TalendDate.formatDate("yyyyMMdd_HH:mm", context.date) + "/move/"'. A checkbox 'Create folders in Drive in necessary' is checked. At the bottom, 'Schema Files' is set to 'Built-In' and 'Edit schema' is available.

The component returns the current file with changed attributes as flow.

Operational Mode: Download a file (DOWNLOAD):

In this mode the component downloads a file to the local file system.
The file can be addressed by its id or by a file path (folder + file-title).
If the file-Id is set, the file path will be ignored.

The file can be saved with a different file name as it has in the Drive.
If the local target directory does not exists, the component can create the directory structure if it does not exists.
To keep the drive clean it is possible to delete the file in the Drive after downloaded successfully.

Basis settings in the example job:

The screenshot shows the configuration window for the 'tGoogleDrive_5: DOWNLOAD(tGoogleDrive_5)' component. The 'Operational Mode' is set to 'Download a file'. Under 'Client Setup', 'Use existing client Drive Component' is checked, and 'tGoogleDrive 6: MODE' is selected. The 'File' section contains the following fields: 'File-Id in the Drive' with the value '((String)globalMap.get("tGoogleDrive_4_FILE_ID"))', 'File path in the Drive' (empty), 'Local target folder' with the value '"/Users/jan/Desktop/"', and 'Rename file to' (empty). There are two checkboxes: 'Create local directory if necessary' (checked) and 'Delete source file in drive' (unchecked). At the bottom, 'Schema Files' is set to 'Built-In' and 'Edit schema' is available.

This mode has additional return values to provide the full file path of the downloaded file and its file size.