Talend User Component tGoogleAnalyticsUnsampledReports

Purpose and procedure

This component manages Google Analytics un-sampled reports.
Un-sampled reports are only available to premium analytics accounts.
There are limitations for the usage of un-sampled reports. Per account you can start 100 reports per day.
Un-sampled reports are reports, which avoids sampling regardless the size of the undelaying raw data.
These reports will be processed asynchronously.
Steps to run un-sampled reports:

1. Start the report
2. Wait for the status COMPLETED
3. Download the report result as CSV file from the Google Drive
4. Parse the result file
5. Delete a report

A service account can also have its own drive. Unfortunately there is no option to take a look into this drive for a service account from the Drive web interface. This component supports step 1, 2 and 4.
For step 3: (download the file) use the user component tGoogleDrive.

It is supposed to proceed this way consisting of these 3 tasks:

- Starting all necessary un-sampled reports.
- Read the metadata of all reports related to the current view (profile) and check their status frequently. To support a reliable proceeding it is strongly recommended to persist the report metadata into a database table. The component provides an input flow and schema, which can be used for this table.
- Download and parse (and store the data) all result files for the completed reports

All tasks should be done in separate jobs. Please be aware it could take minimum about 10 minutes for the Google servers to finish one report. A typical duration time is about one hour.

The component uses the Core Reporting API 3.0 and the Authentication API OAuth 2.0 final.
To provide the ability to run in multiple iterations the component has special capabilities to avoid multiple logins through iterations. Usually automated processes should not use personal accounts. This requirement is addressed by using a service account, which are the only preferred way to login into Google Analytics for automated processes.
Please in case of problems check the checklist at the end of this document.

This component can be set to different modes to support the different steps. A change of the mode reconfigures the component in its properties and flows.

Talend-Integration

This component can be found in the palette under Business->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 Mode	Switches between the different modes the component provides. **Start un-sampled report (START)**: Start an un-sampled report. Means the report will be sent to Google and is awaiting it processing. **List un-sampled reports (LIST)**: In this mode the component reads the meta information of all reports related to a given view. If the status is COMPLETED the result file can be downloaded. **Parse report result file (PARSE)**: In this mode the component reads the downloaded result file of one report and extracts the key figures. **Delete a Unsampled Report:** Because of the limitation of the number of current active reports it is necessary to delete done reports. This delete function was introduced by Google in December 2015. Before this function Google it self run automated processes but this leads sometimes to bottlenecks.

Parameters for the operational modes: START and LIST to establish the connection
Only in these both modes a connection to the Google servers and therefore authentication is needed.
It is supposed to use a service account because this is the preferred authentication mode for background processes.
For test proposes (especially if you want to see the result files in your personal Google account) it could be helpful to use the Application Client-ID authorization.

Property	Content	Data types
Application Name	Not necessary, but recommended by Google. Simple provide the name of your application gathering data. ***Required***	String
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 an Client-ID for native application

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.

Operational Mode: START

In this mode the component initiate an un-sampled report. Every new such requests is a new report regardless if the parameters are the same as the last one. Be aware of the quotas limiting the number of reports per day and web property. Currently the limit is 1000 reports per day and web property.
To start a report your need pretty much the same information as for the normal Reporting API but additional the account-id and web-property-id are needed.

Parameters to set the report context

Property	**Content**	**Data types**
Account-Id	Account-Id	Long, String
Web-property-Id	Web-Property-Id	String
View-Id (Profile-Id)	View-Id (formally known as profile-Id)	Long, String

Properties to define the report

Property	**Content**	**Data types**
Report title	A report must have a title. This title will be also the name of the result file in the Google Drive. It is supposed to give every report its own title. It is a good practice to add the report date to the title. Unlike in the web interface every request is a new report and existing reports cannot be reused via the API.	String
Start Date	All queries need always a time range (only date, not time). ***Required!***	Date, String (yyyy-MM-dd)
End Date	Time range end. If you want gather data for one date, use start date as end date. ***Required!***	Date, String (yyyy-MM-dd)
Dimensions	Dimensions are like group clauses. These dimensions will group the metric values. See advise for notations below. Separate multiple dimensions with a comma.	String
Metrics	Things you want to measure. Separate multiple metrics with a comma. See advise for notations below. ***Required!***	String
Filters	Contains all used filters as concatenated string. See advise for notation below.	String
Segment	Segments are stored filters within Google Analytics, which applies to sessions. If a service account is used it is necessary to declare dynamic segments here because normal segments are always bound to a personal account.	String
Report-ID	Only for the Delete function. This is the alpha-numeric unique ID of an report	String

Advice for filter and segment notation

For dimensions, metrics, filters and sorts you have to use the notation from the Google Core API:
https://developers.google.com/analytics/devguides/reporting/core/dimsmets

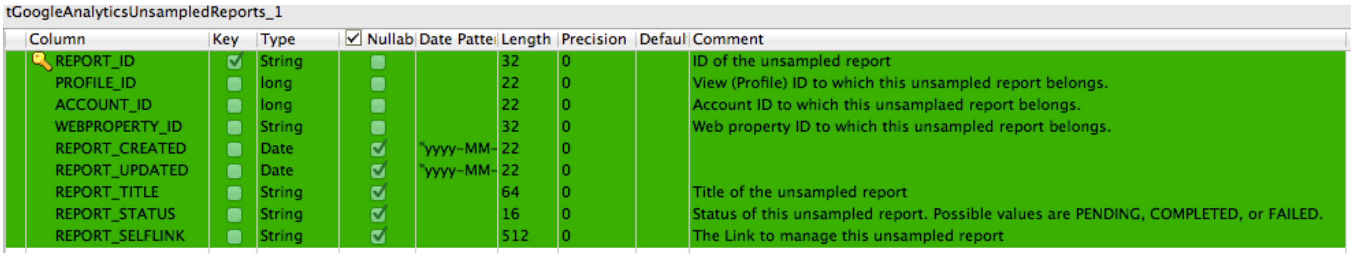
Filters can be concatenated with an OR or AND operator.
Separate filter expressions with a comma means OR
Separate filter expressions with a semicolon means AND

Comparison operators in filters and segment:

Operator	Meaning
“==”	Exact match to include
“!=”	Exact match to exclude
“=~”	Regex match to include (only for strings)
“!~”	Regex match to exclude (only for strings)
“>=”	Greater or equals than (only for numbers)
“=@”	Contains string
“!@”	Does not contains string
“>”	Greater than (only for numbers)
“<=”	Lower or equals than (only for numbers)
“<”	Lower than (only for numbers)

If you use a service account it is not possible to use predefined segments made by a user because they are always limited to the user context. At the moment the preferred way is using dynamic segments, which are made in exactly the same way as the normal segments but only for the current report and not as predefined and named segment.

In this mode the component provides and output flow with the very first meta-data of the transmitted report (as one record)



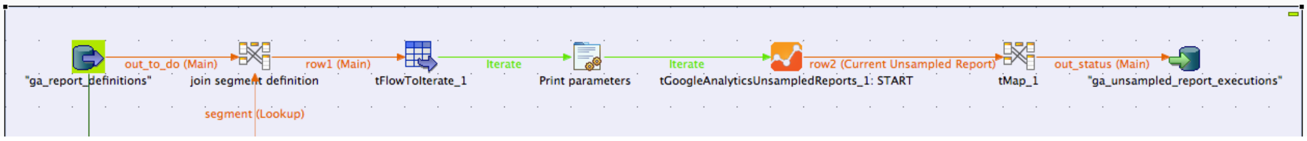
tGoogleAnalyticsUnsampledReports_1

Column	Key	Type	☑ Nullab	Date Patte	Length	Precision	Defaul	Comment
REPORT_ID	☑	String	□		32	0		ID of the unsampled report
PROFILE_ID	□	long	□		22	0		View (Profile) ID to which this unsampled report belongs.
ACCOUNT_ID	□	long	□		22	0		Account ID to which this unsamplaed report belongs.
WEBPROPERTY_ID	□	String	□		32	0		Web property ID to which this unsampled report belongs.
REPORT_CREATED	□	Date	☑	"yyyy-MM-	22	0		
REPORT_UPDATED	□	Date	☑	"yyyy-MM-	22	0		
REPORT_TITLE	□	String	☑		64	0		Title of the unsampled report
REPORT_STATUS	□	String	☑		16	0		Status of this unsampled report. Possible values are PENDING, COMPLETED, or FAILED.
REPORT_SELFLINK	□	String	☑		512	0		The Link to manage this unsampled report

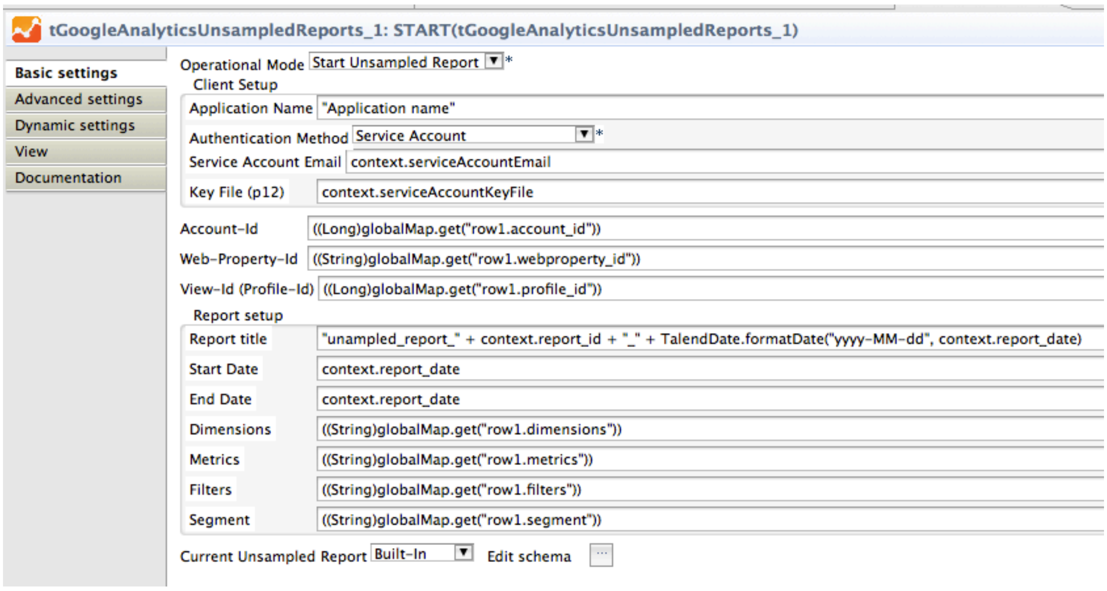
Return values

Return value	Content
ERROR_MESSAGE	Last error message
CURRENT_REPORT_ID	The ID of the current sent report as response of the successful request.

Scenario to start the report



In this scenario the necessary information about the report are stored in a database table.


tGoogleAnalyticsUnsampledReports_1: START(tGoogleAnalyticsUnsampledReports_1)

Operational Mode: Start Unsampled Report *

Client Setup

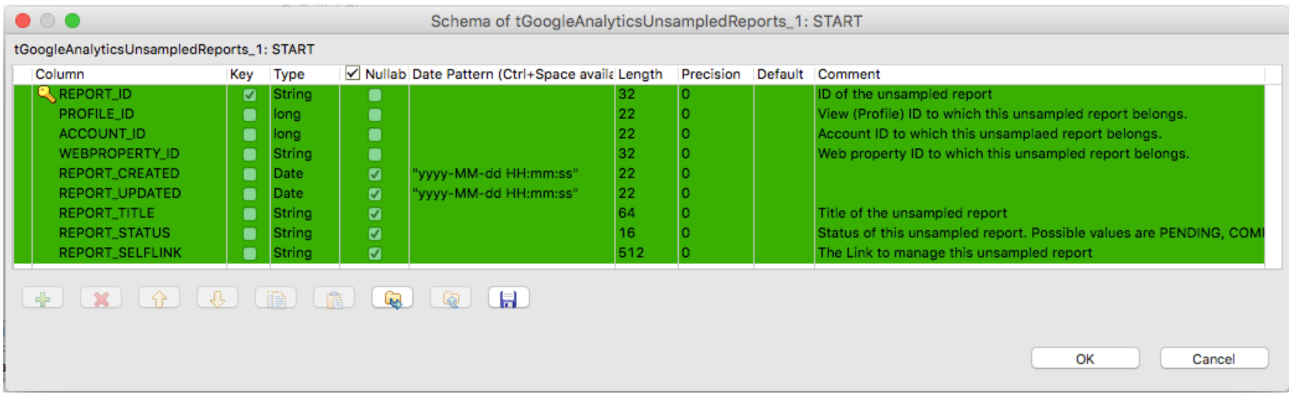
Application Name	"Application name"
Authentication Method	Service Account *
Service Account Email	context.serviceAccountEmail
Key File (p12)	context.serviceAccountKeyFile
Account-Id	((Long)globalMap.get("row1.account_id"))
Web-Property-Id	((String)globalMap.get("row1.webproperty_id"))
View-Id (Profile-Id)	((Long)globalMap.get("row1.profile_id"))

Report setup

Report title	"unampled_report_" + context.report_id + "_" + TalendDate.formatDate("yyyy-MM-dd", context.report_date)
Start Date	context.report_date
End Date	context.report_date
Dimensions	((String)globalMap.get("row1.dimensions"))
Metrics	((String)globalMap.get("row1.metrics"))
Filters	((String)globalMap.get("row1.filters"))
Segment	((String)globalMap.get("row1.segment"))

Current Unsampled Report: Built-In

Here the configuration of the component in this scenario. The output flow returns the first created metadata of the started report. The start and end date can also provided as java.util.Date typed objects.


Schema of tGoogleAnalyticsUnsampledReports_1: START

tGoogleAnalyticsUnsampledReports_1: START

Column	Key	Type	☑ Nullab	Date Pattern (Ctrl+Space availa	Length	Precision	Default	Comment
REPORT_ID	☑	String	□		32	0		ID of the unsampled report
PROFILE_ID	□	long	□		22	0		View (Profile) ID to which this unsampled report belongs.
ACCOUNT_ID	□	long	□		22	0		Account ID to which this unsamplaed report belongs.
WEBPROPERTY_ID	□	String	□		32	0		Web property ID to which this unsampled report belongs.
REPORT_CREATED	□	Date	☑	"yyyy-MM-dd HH:mm:ss"	22	0		
REPORT_UPDATED	□	Date	☑	"yyyy-MM-dd HH:mm:ss"	22	0		
REPORT_TITLE	□	String	☑		64	0		Title of the unsampled report
REPORT_STATUS	□	String	☑		16	0		Status of this unsampled report. Possible values are PENDING, COMI
REPORT_SELFLINK	□	String	☑		512	0		The Link to manage this unsampled report

Scenario to delete a report

This scenario is fairly simple. You have to select all reports which are already successful processed and iterate through the reports. Set the actual report-id as parameter in the component setting Report-ID and that’s it. In this mode the component has only the parameters to establish the connection (Setup Client) and choose the Account-Id, Web-Property-Id and View-Id and at least the Report-Id if the report to delete.

If the component does not throw an error, the delete has been finished successfully.

I suggest in the own database not to delete the record about the deleted report but set a flag (or date) to mark the report as deleted. This way the important information about the report processing still persists and can be used to restore data in an accident.
In this mode the only return value is the actual given report-Id as CURRENT_REPORT_ID. This helps to build convenient job designs.

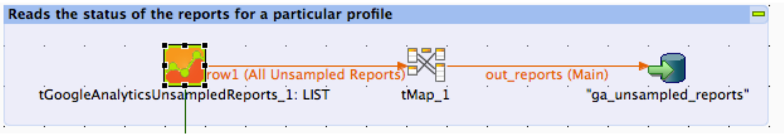
Operational Mode: LIST

The component connects to the Google servers and reads the metadata for the un-sampled reports for the given context.

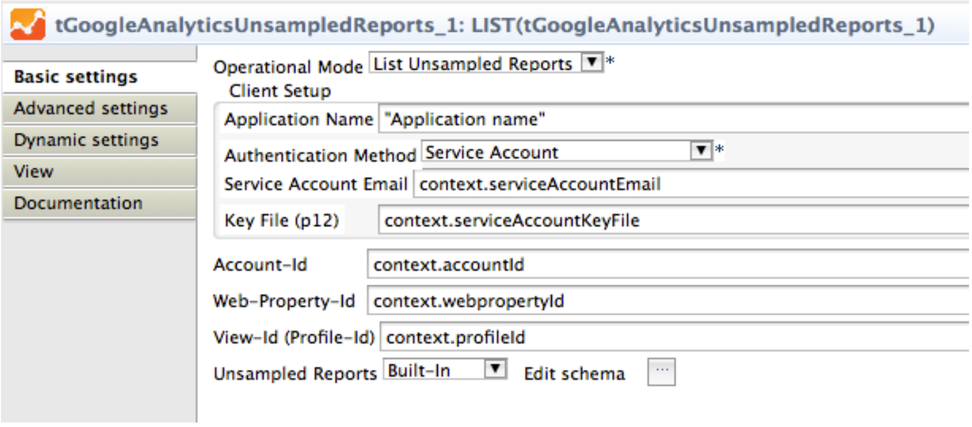
Parameters to set the context to list the reports for

Property	Content	Data-types
Account-Id	Account-Id.	String, Long
Web-Property-Id	Web-Property-Id is the ID of the web site.	String
View-Id (Profile-Id)	ID of the View (formally known as profile)	String, Long

Here a typical job gathering the report metadata for a context.


Reads the status of the reports for a particular profile
row1 (All Unsampled Reports)
out_reports (Main)
tGoogleAnalyticsUnsampledReports_1: LIST
tMap_1
"ga_unsampled_reports"

To recognize the different modes of the component in a job it is a good practice to set as View for the component this term: __UNIQUE_NAME__: __MODE__


tGoogleAnalyticsUnsampledReports_1: LIST(tGoogleAnalyticsUnsampledReports_1)

Operational Mode: List Unsampled Reports *

Client Setup

Application Name: "Application name"

Authentication Method: Service Account *

Service Account Email: context.serviceAccountEmail

Key File (p12): context.serviceAccountKeyFile

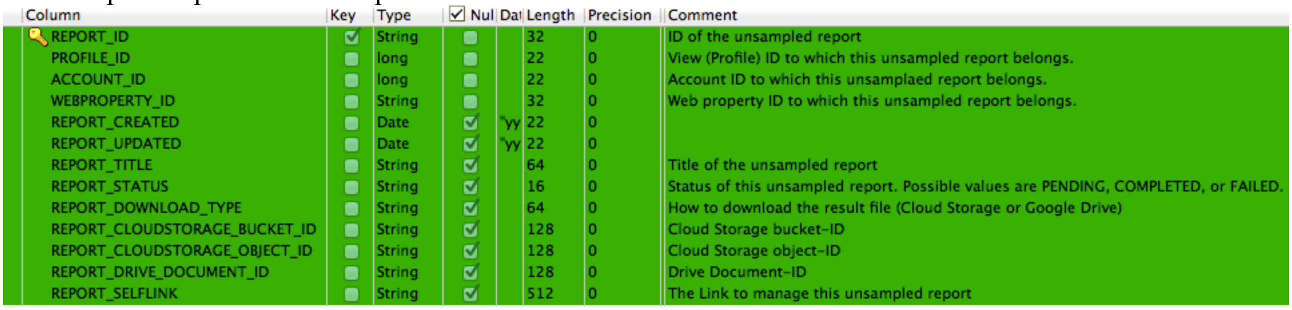
Account-Id: context.accountId

Web-Property-Id: context.webpropertyId

View-Id (Profile-Id): context.profileId

Unsampled Reports: Built-In

The component provides an output flow with this schema:



Column	Key	Type	☑ Nul	Dat	Length	Precision	Comment
REPORT_ID	☑	String	□		32	0	ID of the unsampled report
PROFILE_ID	□	long	□		22	0	View (Profile) ID to which this unsampled report belongs.
ACCOUNT_ID	□	long	□		22	0	Account ID to which this unsamplaed report belongs.
WEBPROPERTY_ID	□	String	□		32	0	Web property ID to which this unsampled report belongs.
REPORT_CREATED	□	Date	☑	"yy	22	0	
REPORT_UPDATED	□	Date	☑	"yy	22	0	
REPORT_TITLE	□	String	☑		64	0	Title of the unsampled report
REPORT_STATUS	□	String	☑		16	0	Status of this unsampled report. Possible values are PENDING, COMPLETED, or FAILED.
REPORT_DOWNLOAD_TYPE	□	String	☑		64	0	How to download the result file (Cloud Storage or Google Drive)
REPORT_CLOUDSTORAGE_BUCKET_ID	□	String	☑		128	0	Cloud Storage bucket-ID
REPORT_CLOUDSTORAGE_OBJECT_ID	□	String	☑		128	0	Cloud Storage object-ID
REPORT_DRIVE_DOCUMENT_ID	□	String	☑		128	0	Drive Document-ID
REPORT_SELFLINK	□	String	☑		512	0	The Link to manage this unsampled report

Return values

Return value	Content
ERROR_MESSAGE	Last error message
NB_LINE_UNSAMPLED_REPORTS	Number un-sampled reports for the given view (profile)

Operational Mode: PARSE

In this mode the component does not connect to the Google servers, instead it reads a downloaded result file and parse it. This file is actually a csv file but has an unstable number of header lines and the field order is sometimes disturbed if a segment was used in the report. This makes it hard to use here the normal file input components Talend provides. The header carries information about the profile, metrics, dimensions, filters and the segment. The component takes care about these specific issues and provides the same output features as the component tGoogleAnalyticsInput to make it easy to reuse the same methods to store the values.

Properties

Property	Content	Data types
Report Result File	Set here the file name of the result file, which is already downloaded (e.g. with the help of the tGoogleDrive component). ***Required!***	String
Normalized Output Flows	Choose if you want to use a plain schema (you have to know at design time what columns your file will provide).	Boolean
Exclude ga:date dimension and provide value as return value	Set this to exclude the ga:date dimension from the normalized output flow for dimension and instead set the ga:date value in the globalMap as return value (available while the flow runs).	Boolean
Use Header info for Dimensions and Metrics	If you set this option, the component ignores the dimension and metrics settings below and takes this information from the header of the result file.	Boolean
Dimensions	If not taken from the file header. The information is necessary to build the normalized schema. Example: “ga:date,ga:source,ga:keyword”	String
Metrics	If not taken from the file header. The information is necessary to build the normalized schema. Example: “ga:visits,ga:newVisits”	String

Using flat (plain) output

In the schema you need an amount of columns equals to the sum of the number of dimensions and metrics.
Columns in the schema must start at first with dimensions (if provided) and ends with metrics.
Schema column types must match to the data types of the dimensions and metrics. The schema column names can differ from the names of dimensions and metrics. Only the order and there types are important.
Metric columns should be of the type double. Google always provides a value and send never null or something different than a number.
In Talend schema columns must follow the Java naming rules therefore avoid writing ga:xxx instead use the name without the ga: namespace prefix.
Important: For date dimensions (e.g. ga:date) you must specify the date pattern as “yyyyMMdd” if you want it as Date typed value.

Using normalized output

If a normalized output is used the component reads internal the plain records and folds them into the normalized outputs.

Return values

Return value	Content	Type
ERROR_MESSAGE	Last error message	String
NB_LINE	Number plain records (only set if normalization is not used)	Integer
NB_LINE_METRIC_VALUES	Number of normalized metric records.	Integer
NB_LINE_DIMEMSION_VALUES	Number of normalized dimension records	Integer
REPORT_PROFILE_ID	View used to build the report	String
REPORT_METRICS	Metrics of the report	String
REPORT_DIMENSIONS	Dimensions of the report	String

REPORT_FILTERS	Filters used for the report	String
REPORT_SEGMENT	Segment used for the report	String
REPORT_START_DATE	Start date for the report (as String yyyy-MM-dd)	String
REPORT_END_DATE	End date for the report (as String yyyy-MM-dd)	String
CURRENT_DATE	The value of the ga:date dimension (if present in the file) for every row. This value is only available in the “Normalized Flow” mode.	Date

The REPORT_xxx values will be filled just before delivering the first output, though it can be used to enhance the output flows.

Explanation for the normalized output

The normalized output as made for scenarios in which the job will be configured with metrics and dimensions at runtime. In this use case it is not possible to declare the appropriated schema for the flat output.
The normalization creates 2 read only output schemas:

Dimensions

Column	Type	Meaning
ROW_NUM	int	The row number from the original flat result row. It identifies the records, which belongs to together.
DIMENSION_NAME	String	Name of the dimension
DIMENSION_VALUE	String	Value of the dimension

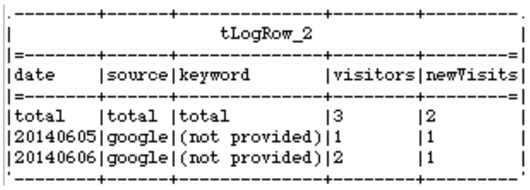
Metrics

Column	Type	Meaning
ROW_NUM	int	The row number from the original flat result row. It identifies the records, which belongs together.
METRIC_NAME	String	Name of the metric
METRIC_VALUE	Double	Value of the metric

Comparison of a plain output to a normalized output

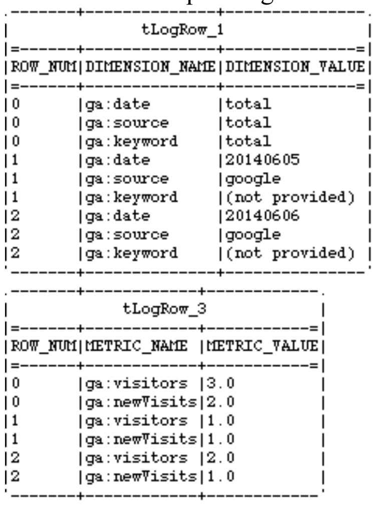
Given the dimensions was set to: "ga:date,ga:source,ga:keyword" and the metrics was set to: "ga:visitors,ga:newVisits"
The information about dimension and metrics can read retrieved from the input file if the option

Here some example outputs:
The plain output



```
.--------+------+--------------+--------+---------.
|                    tLogRow_2                    |
|=-------+------+--------------+--------+--------=|
|date    |source|keyword       |visitors|newVisits|
|=-------+------+--------------+--------+--------=|
|total   |total |total         |3       |2        |
|20140605|google|(not provided)|1       |1        |
|20140606|google|(not provided)|2       |1        |
'--------+------+--------------+--------+---------'
```

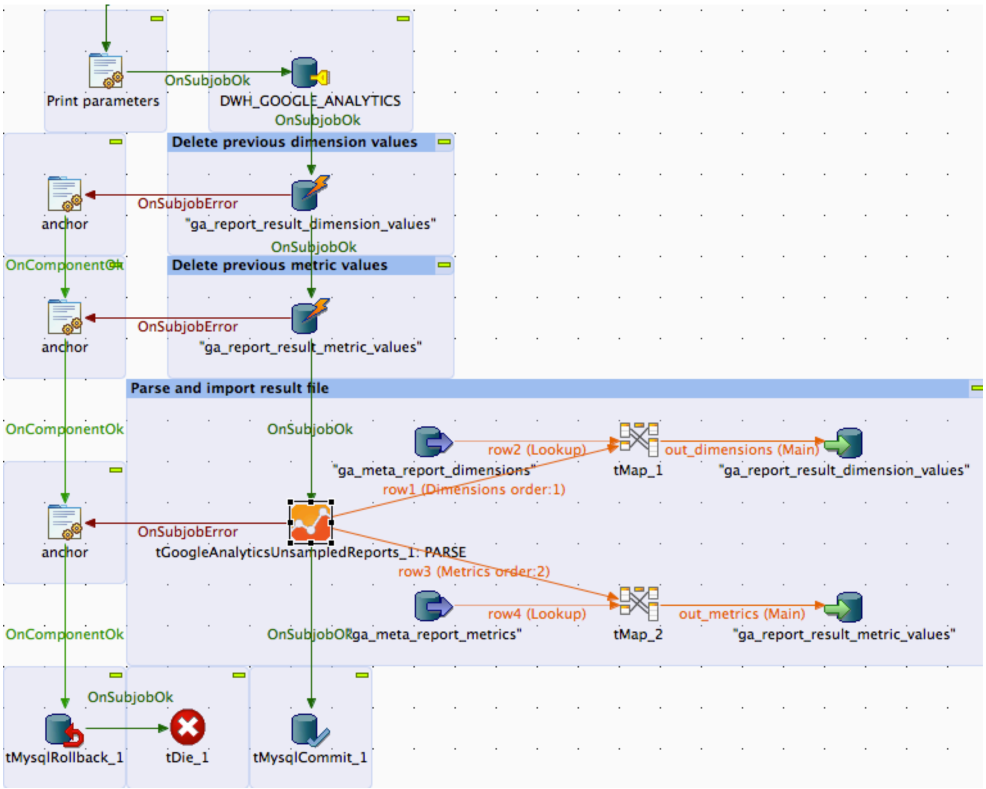
… and the corresponding normalized output



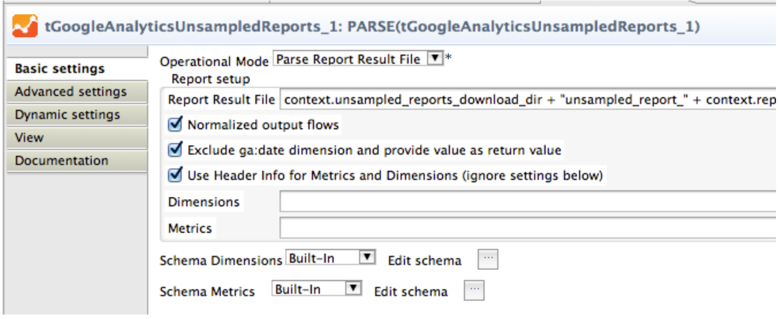
```
.-------+--------------+---------------.
|              tLogRow_1               |
|=------+--------------+--------------=|
|ROW_NUM|DIMENSION_NAME|DIMENSION_VALUE|
|=------+--------------+--------------=|
|0      |ga:date       |total          |
|0      |ga:source     |total          |
|0      |ga:keyword    |total          |
|1      |ga:date       |20140605       |
|1      |ga:source     |google         |
|1      |ga:keyword    |(not provided) |
|2      |ga:date       |20140606       |
|2      |ga:source     |google         |
|2      |ga:keyword    |(not provided) |
'-------+--------------+---------------'
```

```
.-------+-----------+------------.
|           tLogRow_3            |
|=------+-----------+-----------=|
|ROW_NUM|METRIC_NAME |METRIC_VALUE|
|=------+-----------+-----------=|
|0      |ga:visitors |3.0         |
|0      |ga:newVisits|2.0         |
|1      |ga:visitors |1.0         |
|1      |ga:newVisits|1.0         |
|2      |ga:visitors |2.0         |
|2      |ga:newVisits|1.0         |
'-------+-----------+------------'
```

Next a real live scenario for using the normalized output in conjunction with the usage of the meta-data (gathered with the component tGoogleAnalyticsManagement):



Here the configuration of the component for parsing a result file:



tGoogleAnalyticsUnsampledReports_1: PARSE(tGoogleAnalyticsUnsampledReports_1)

Operational Mode: Parse Report Result File *

Report setup

Report Result File: context.unsampled_reports_download_dir + "unsampled_report_" + context.rep

☑ Normalized output flows

☑ Exclude ga:date dimension and provide value as return value

☑ Use Header Info for Metrics and Dimensions (ignore settings below)

Dimensions

Metrics

Schema Dimensions: Built-In — Edit schema

Schema Metrics: Built-In — Edit schema

The file name in this scenario will be assembled with the download folder and the report_id and the report_date.

Note the checked option “Exclude ga:date dimension …”. The value of the ga:date dimension will be used in the tMap.

Expression for a e.g. report_date output column in the example:

```
((java.util.Date)globalMap.get("tGoogleAnalyticsUnsampledReports_1_CURRENT_DATE")) != null ?
((java.util.Date)globalMap.get("tGoogleAnalyticsUnsampledReports_1_CURRENT_DATE"))  : context.report_date
```

The given date for the report (as context parameter) will be replaced by the ga:date value.
This makes the import cleverer. For reports does not containing the ga:date dimensions the parameter will be used and for reports carrying the date its value will be used.
This job is designed to gather the data for one day and one report (a combination of a view, dimensions, metrics and filters very much like a custom report in the Google Analytics dashboard).
This job gets the view-ID, dimensions, metrics and filters as context variables and will be called, as much there are queries and dates to process.
The tMaps exchanges the dimension names and metric names with their numeric ids and adds a report-ID and the current date into the output flow for the database.
To get this job restart-able everything is done within a transaction and the previous data for the report and date will be deleted at first.
By the way, take note about the way to handle errors here, this is very easy and avoid implementing the error handling multiple times. The anchors are tJava components without any code.

It is supposed to use gather the Analytics metadata to be sure you have access to all necessary data and to be able to build a star schema for the dimensions and metrics. Take a look at the component tGoogleAnalyticsManagement.

Configuration checklist:

1. Is the email of the service account added to all relevant views (profiles)?
2. Is the system time of the host running the job synchronized with a NTP server?
3. Is the Google Analytics API enabled in the Google API Console?
4. Is the used account a premium account?

Tip:
Check your report at first in the Google Analytics API Explorer to get an idea if the data works for you.

Advanced Option Parameters

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)
Fetch Size	This is the amount of data the component fetches at once. The value is used to set the max_rows attribute. max_rows means not the absolute amount of data! The component manages setting the start index to get all data. To achieve this, the component iterates as long as the last result set are completely fetched.
Local Number Format	You can get numbers in various formats. Here you can define the locale in which format double or float values are should textual format by the API.
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.