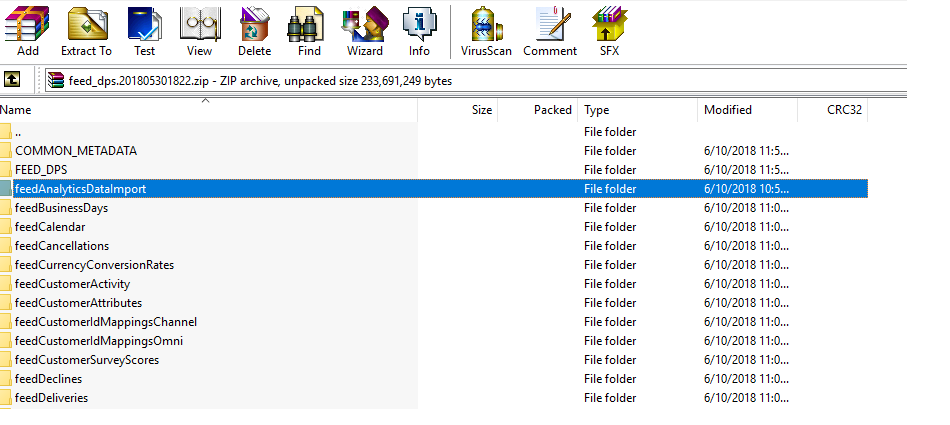
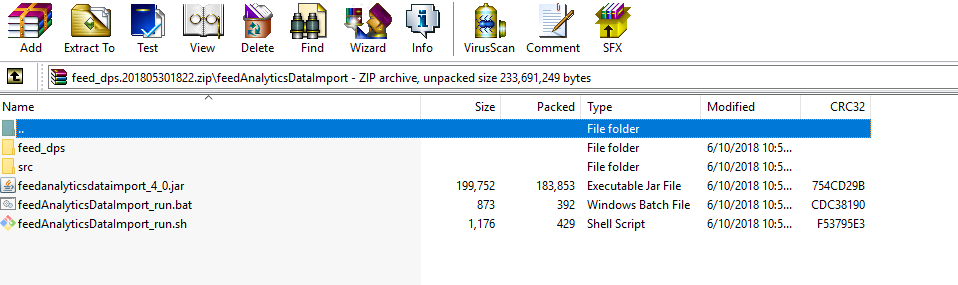
**Job Archive Files**

The first step to executing the Talend jobs is to extract them from the archive files through which they are delivered. The archive files will have each runnable job included in a sub-folder off of the root directory (as well as project folders for exported items), for example:



Here the individual runnable jobs are feedAnalyticsDataImport, feedBusinessDays, etc. Within each job subfolder contains the runnable jar and the scripts that are used to execute it:



Once the jobs have been extracted into a location where they can be executed, the process of running these jobs simply involves invoking the script appropriate for your environment (.bat or .sh) with the set of context parameters appropriate for the job. The following sections describe how to set the context parameters for each of our three job types (connectors, client ETL and feed\_dps).

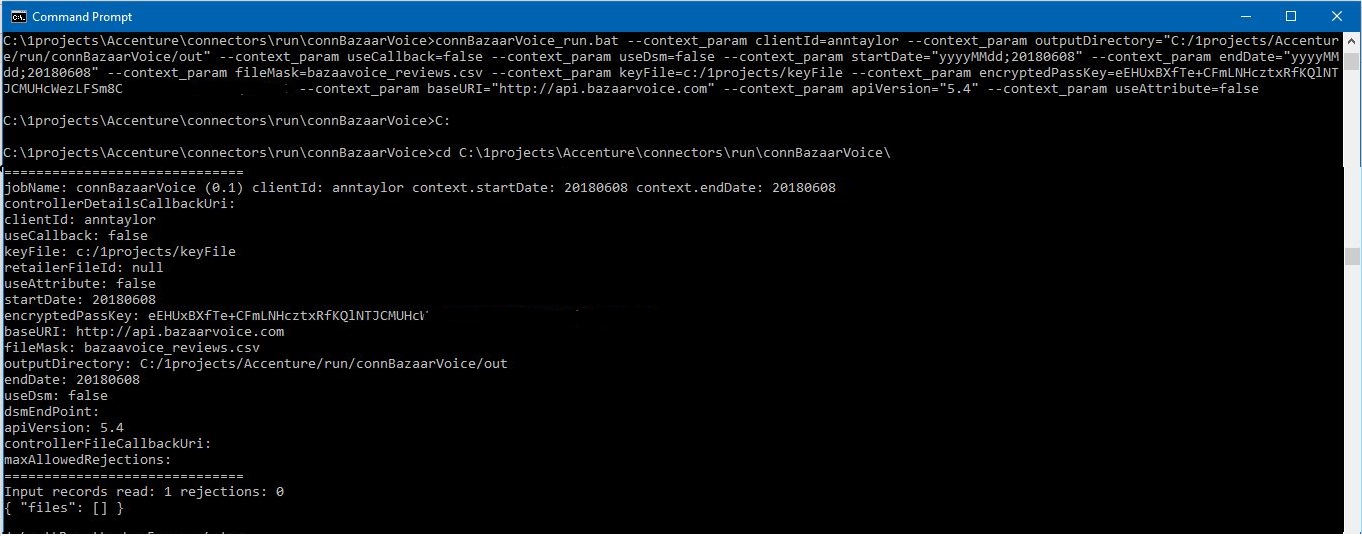
**Connectors**

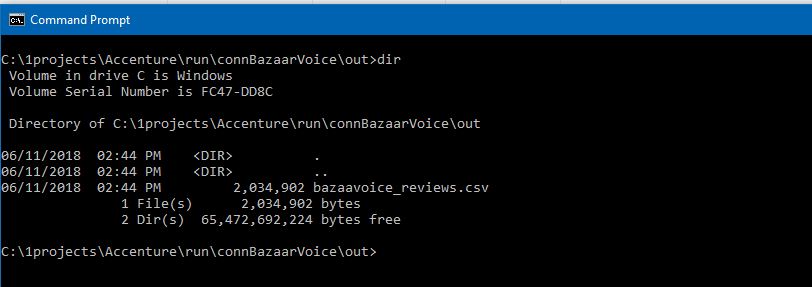
The following context parameters are used consistently across all connector jobs:

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Description** | **How to use when running jobs outside of DNA environment** |
|  |  |  |
| clientId | The internal client identifier for which the connector is being run (e.g. anntaylor). | **Required** for all ETL jobs, but not typically used by connectors. An internal id should be generated for each customer account and used consistently across all job types when run for that account. |
| startDate | The start date of the period for which to collect data for, formatted as “yyyyMMdd;20180101”. | **Required** – however some connectors may provide other context values to override and or provide more resolution for the time period to collect. |
| endDate | The end date of the period for which to collect data for, should be formatted as “yyyyMMdd;20180101”. | **Required** – however some connectors may provide other context values to override and or provide more resolution for the time period to collect. |
| outputDirectory | The fully qualified directory to which the resulting data file will be written | **Required** – set to a writable local directory |
| useCallback | Flag indicating whether to call back to the DNA platform with details about the connector execution. | **Required** - always set to false outside of DNA environment |
| useDsm | Flag indicating whether to upload files to s3 via DNA’s internal dsm middleware | **Required** – always set to false outside of DNA environment |
| fileMask | Name of file to write collected data to (in the outputDirectory). Note that a small number of connectors use fileName instead of fileMask, an inconsistency DNA will likely correct. | **Required** – typically set to a filename that reflects the type of data and date for which it is collected (e.g. GoogleAnalytics\_20180101) |
| retailerFileId | The internal identifier that reflects the DNA configuration entry that triggered this connector execution and used for reference in the call back details. | Not required when running outside of DNA environment – do not set |
| controllerDetailsCallbackUri | Uri for calling back connector details | Not required – do not set when useCallback is false |
| controllerFileCallbackUri | Uri for calling back with file details | Not required – do not set when useCallback is false |
| dsmEndPoint | DSM end point | Not required – do not set when useDsm is false |

In addition to the set of context parameters that are standard across all connectors, each connector will have specific context parameters that must be set in order to collect data. Credentials, end points and data values to retrieve are all examples of additional configuration that connectors may require. Full details for each connector can be found in the detailed documentation specific to each connector.

Example of running a connector:





**Client ETL Jobs**

Client ETL jobs execute against a single input file that was collected for a client to produce one or more fragments. Like connectors, client ETL jobs also have a set of context parameters that are consistent across all jobs:

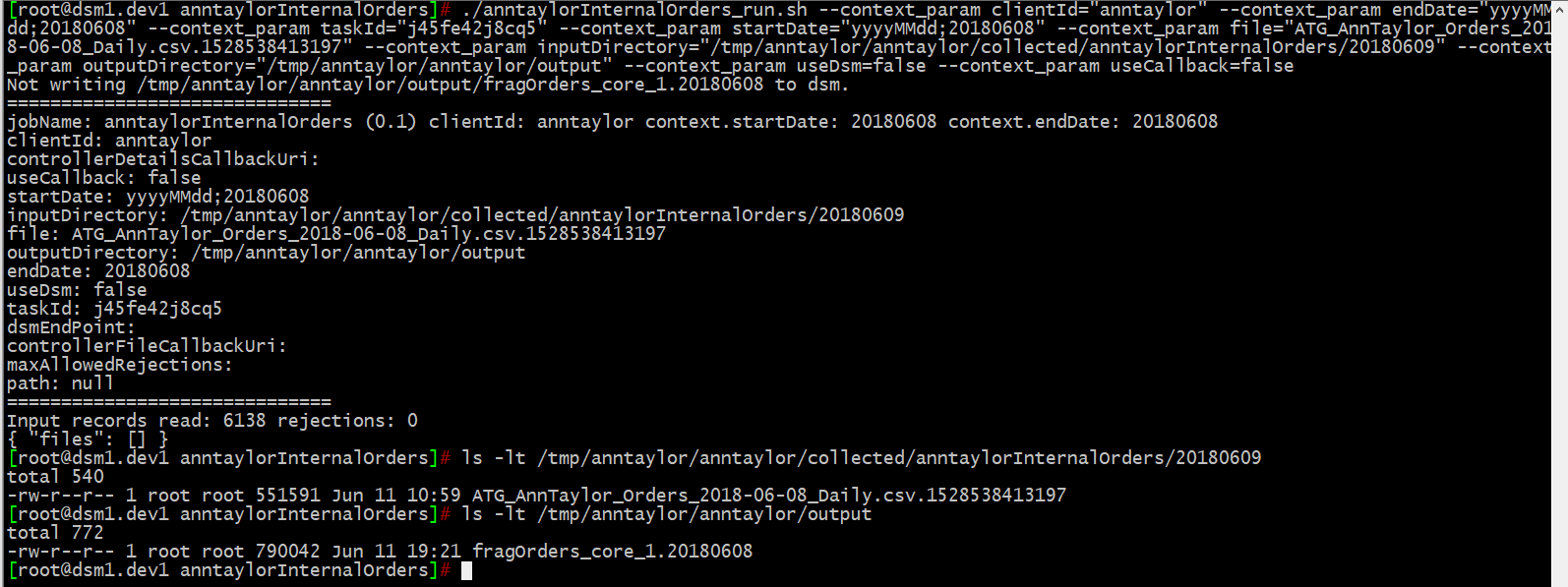
|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Description** | **How to use when running jobs outside of DNA environment** |
|  |  |  |
| clientId | The internal client identifier for which the client ETL job is being run (e.g. anntaylor). | **Required** for all ETL jobs, rarely used by client ETL jobs (typically only when a single etl job services multiple accounts for the same customer). An internal id should be generated for each customer account and used consistently across all job types when run for that account. |
| startDate | The start date of the period for which the data in the input file was collected for, formatted as “yyyyMMdd;20180101”. | **Required** |
| endDate | The end date of the period for which the data in the input file was collected for, should be formatted as “yyyyMMdd;20180101”. | **Required** |
| file | The file name of the input file to process. | **Required** – do not fully qualify as value is combined with inputDirectory to form the full path of the input file |
| inputDirectory | The fully qualified directory in which the input file will be located. | **Required** – set to the local directory where the input file exists |
| outputDirectory | The fully qualified directory to which the resulting fragment data files will be written to. | **Required** – set to a writable local directory |
| useCallback | Flag indicating whether to call back to the DNA platform with details about the client etl job execution. | **Required** - always set to false outside of DNA environment |
| useDsm | Flag indicating whether to upload fragments to s3 via DNA’s internal dsm middleware | **Required** – always set to false outside of DNA environment |
| taskId | Internal DNA identifier for the specific execution instance of this etl job. | **Required** – set to a random string, infrequently used to avoid collisions against shared resources from ETL jobs running concurrently |
| maxAllowedRejections | Integer value indicating the maximum number of input rows that can be rejected before the job is failed. Not all client ETL jobs support this (i.e. they fail if any input row cannot be processed) | Not required – inspection of etl job required to know whether option is supported |
| controllerDetailsCallbackUri | Uri for calling back connector details | Not required – do not set when useCallback is false |
| controllerFileCallbackUri | Uri for calling back with file details | Not required – do not set when useCallback is false |
| dsmEndPoint | DSM end point | Not required – do not set when useDsm is false |
| path | DSM location of input file | Not required – do not set when useDsm is false |

Note that all fragment output files will be written to the outputDirectory using the standard DNA naming convention:

[fragmentIdentifier]\_[job or date specific details].[yyyyMMdd|nodate]

* **fragmentIdentifier** indicates the type of fragment and which feed jobs will pick it up (and how they will utilize it)
* **job or date specific details** provides the client job with an optional way of labeling the fragment so that similar fragments from other jobs (and other dates) do not overwrite
* **yyyyMMdd|nodate** indicates whether the fragment is for a specific date (yyyyMMdd) or whether the fragment should be used for future dates until replaced (nodate)

Here is an example of the execution of a client ETL job:



**Feed\_Dps ETL Jobs**

Feed\_Dps jobs operate on fragments and generate a single file for a single day according to the DNA internal detailed feed specification. The set of context parameters consistent across feed jobs:

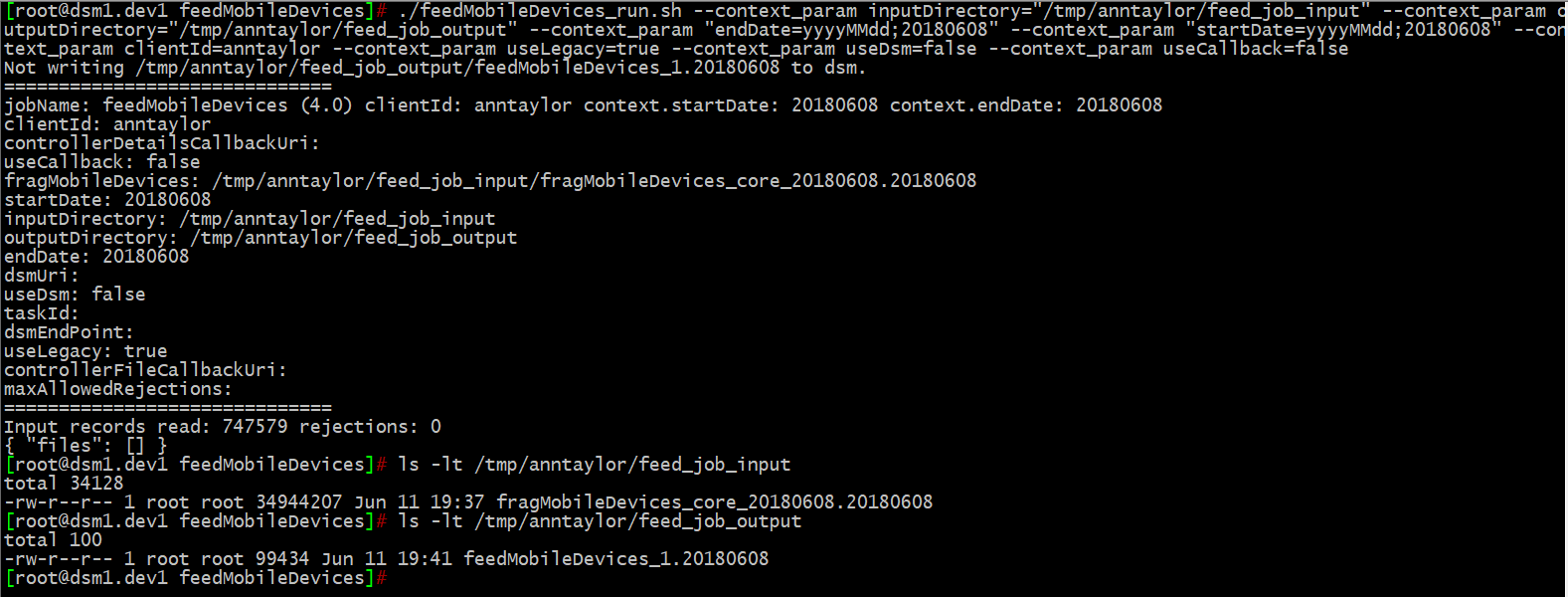
|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Description** | **How to use when running jobs outside of DNA environment** |
|  |  |  |
| clientId | The internal client identifier for which the feed job is being run (e.g. anntaylor). | **Required** for all ETL jobs, rarely used by feed jobs (other than when required to determine s3 location specific to an account) |
| startDate | The date for which the feed is being generated, formatted as “yyyyMMdd;20180101”. | **Required** |
| endDate | The date for which the feed is being generated, formatted as “yyyyMMdd;20180101”. | **Required** |
| inputDirectory | The fully qualified directory in which the input fragments will be located. | **Required** – set to the local directory where the fragments have been written |
| outputDirectory | The fully qualified directory to which the resulting feed file will be written to. | **Required** – set to a writable local directory |
| useCallback | Flag indicating whether to call back to the DNA platform with details about the client etl job execution. | **Required** - always set to false outside of DNA environment |
| useDsm | Flag indicating whether to upload fragments to s3 via DNA’s internal dsm middleware | **Required** – always set to false outside of DNA environment |
| useLegacy | There are two modes in which feed jobs can acquire their input fragments: Legacy where the input directory is scanned for any fragments that the feed job uses as input and Explicit where all of the fragments to be used are explicitly listed in context parameters that are specific to the feed job. For the purposes of running outside of the DNA environment, Legacy is typically the easier approach to use. | **Required** – in order to use all fragments available to the feed job set to true. If explicit fragment configuration is desired, specify fragments in context parameters that are specific to the feed job (consult detailed documentation). |
| controllerDetailsCallbackUri | Uri for calling back connector details | Not required – do not set when useCallback is false |
| controllerFileCallbackUri | Uri for calling back with file details | Not required – do not set when useCallback is false |

The feed file that is produced by the job is written to the outputDirectory using the standard feed naming convention of:

[feedIdentifier]\_1.yyyyMMDD

* **feedIdentifier** indicates the type of feed that was generated
* **yyyyMMdd** indicates the date for which it was generated for

Example of running a feed\_dps job:



**End-to-End Process**

In order use the ETL jobs to generate all of the feeds for a single day of client data outside of the DNA environment; the following paradigm can be used:

1. Collect all of the input files. This can be done by simply locating the files to be processed or by invoking connector jobs to acquire the data.
2. Determine which client ETL job to execute against each input file. Execute the client ETL job and collect all of the resulting fragments (across all of the jobs) into a single feed input directory.
3. Execute each of the desired feed\_dps ETL jobs specifying the input directory where the fragments have been collected. Each job will produce a single feed file (in the DNA internal feed specification format) for the date into the configured output directory.