



DLRunner is a tool that complements Salesforce Data Loader. It is intended to be used when Data Loader is run in batch mode from the command line. It performs many of the administrative tasks that Data Loader does not and that would otherwise have to be performed either manually or by custom scripts.

Some of the features of DLRunner include:

- Works with your existing command line Data Loader configuration
- Gives you more flexibility and control of how and when Data Loader jobs are run
- Simplifies your Microsoft Task Scheduler configuration if you are running multiple Data Loader jobs at different times or days
- Performs error checking before Data Loader runs so that issues that may cause Data Loader to fail are found upfront and identified
- Automates delivery of Data Loader export files, success files, and error files to end users and administrators
- Automates delivery of Data Loader job statuses to end users and administrators in an easy to read format
- Automates archiving of Data Loader job information and output files.

DLRunner DOES NOT perform any direct calls to Salesforce or any other APIs or Web Services and does not login to Salesforce. When you use DLRunner, all user authentications, log-ins, and communication with Salesforce is still performed by Data Loader.

DLRunner has no user interface. It consists of the DLRunner executable and one or more properties files which you configure in order to tell DLRunner how to process Data Loader jobs.

Installing and Configuring DLRunner

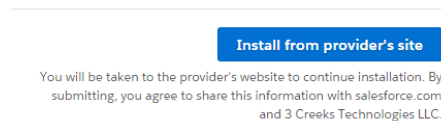
To set-up DLRunner, perform the following steps:

1. If you do not already have a working command line Data Loader installation, install Data Loader and configure it to run from a command line per these [Salesforce instructions](#). Once set up, validate that you can successfully run Data Loader from the command line.

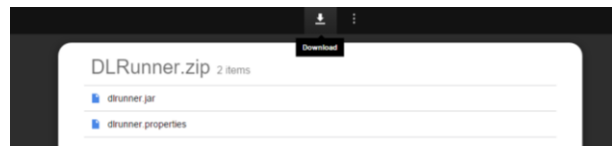
Important: DLRunner reads the process-conf.xml file in order to get attributes of the jobs that you are running. To successfully use DLRunner, depending on your version of Data Loader, you may need to upgrade to a newer version of Data Loader.

2. Ensure that you have a working installation of Java 8 (ver. 1.8) and that the java runtime is included in your system's PATH environment variable. See [the Java Windows installation instructions](#) for information on how to do this.
3. Click the Get It Now button on the DLRunner AppExchange web page to download the installation zip file. Since DLRunner is a standalone application and not a Salesforce package, you install it directly on your system. You will still be required to either login to your Salesforce instance or sign in as a guest once you fill out and submit the information form.

You will then see the following message indicating that to install, you will be redirected to the 3 Creeks Website:



Read the terms and conditions, click the checkbox indicated that you agree to them, and then click the Install from provider's site button. You will then be redirected to a google drive that contains the DLRunner installation zip file.



Click the download button in the upper middle of the page to download the zip file.

4. Unzip the DLRunner files into an appropriate folder on the same system that you are running command line Data Loader from. You will see two DLRunner files:
 1. **dlrunner.jar** – This is the DLRunner executable. Once DLRunner is configured, your Windows Task Scheduler will call this executable, along with appropriate parameters, to execute your Data Loader jobs. More on all this a little later.
 2. **dlrunner.properties** – You will edit and possibly replicate this file to configure DLRunner to run and administer your Data Loader jobs. Details on this are below so read on!
5. Use your favorite text editor to open the dlrunner.properties file. Most of the instructions necessary to configure DLRunner are within the comments contained in this file. Read and follow the instructions found in dlrunner.properties to configure DLRunner. When complete, save the changes that you have made to dlrunner.properties.
6. Once configured, DLRunner can be run by opening a command prompt window and typing:

```
java -jar dlrunner.jar
```

The above command line would run DLRRunner using the default properties file of `dlrunner.properties`. You can also have DLRRunner use a different properties file by typing the following in the command line:

```
java -jar dlrunner.jar -p [properties file name]
```

By the way, the above commands are also the commands that you will add to your Windows task scheduler to execute your Data Loader jobs using DLRRunner.

To see all of the DLRRunner command line options type:

```
java -jar dlrunner.jar -?
```

Using DLRRunner To Run Jobs On Different Schedules

As mentioned above, DLRRunner accesses a file that came with your a Data Loader install called `process-conf.xml`. Since you have already set up a working command line Data Loader environment, you are probably very familiar with this file. It is used by Data Loader to specify attributes of the jobs that you are having Data Loader run. DLRRunner reads this file to get path and other information about the jobs that you specified in the DLRRunner properties file.

You can think of the DLRRunner properties file as a specification for a session of one or more Data Loader jobs. Since you can pass different properties files to DLRRunner on the command line by using the `-p` option, DLRRunner can be easily configured to run different Data Loader sessions.

Here is an example. Suppose that you need to run a set of five Data Loader jobs nightly and then, in addition, you have a different set of three weekly extracts that you run on Saturday night for your Marketing department. Using DLRRunner, you can configure all of the nightly and weekly jobs in the same `process-conf.xml` file and then create two different properties files for DLRRunner to use.

Let's say that you call the DLRRunner properties file for nightly jobs `DlrNightly.properties` and that you have configured it with the five nightly Data Loader jobs. Then you created a properties file for the weekly jobs called `DlrSatExtracts.properties` and configured it with the three Marketing extracts.

Now to run these jobs, in your Windows task scheduler, you simply have two entries. One entry specifies your nightly jobs to run by passing `DlrNightly.properties` to DLRRunner and is scheduled to run every night:

```
java -jar dlrunner.jar -p DlrNightly.properties
```

The other entry is scheduled to run on Saturday nights and specifies the `DlrSatExtracts.properties` like so:

```
java -jar dlrunner.jar -p DlrSatExtracts.properties
```

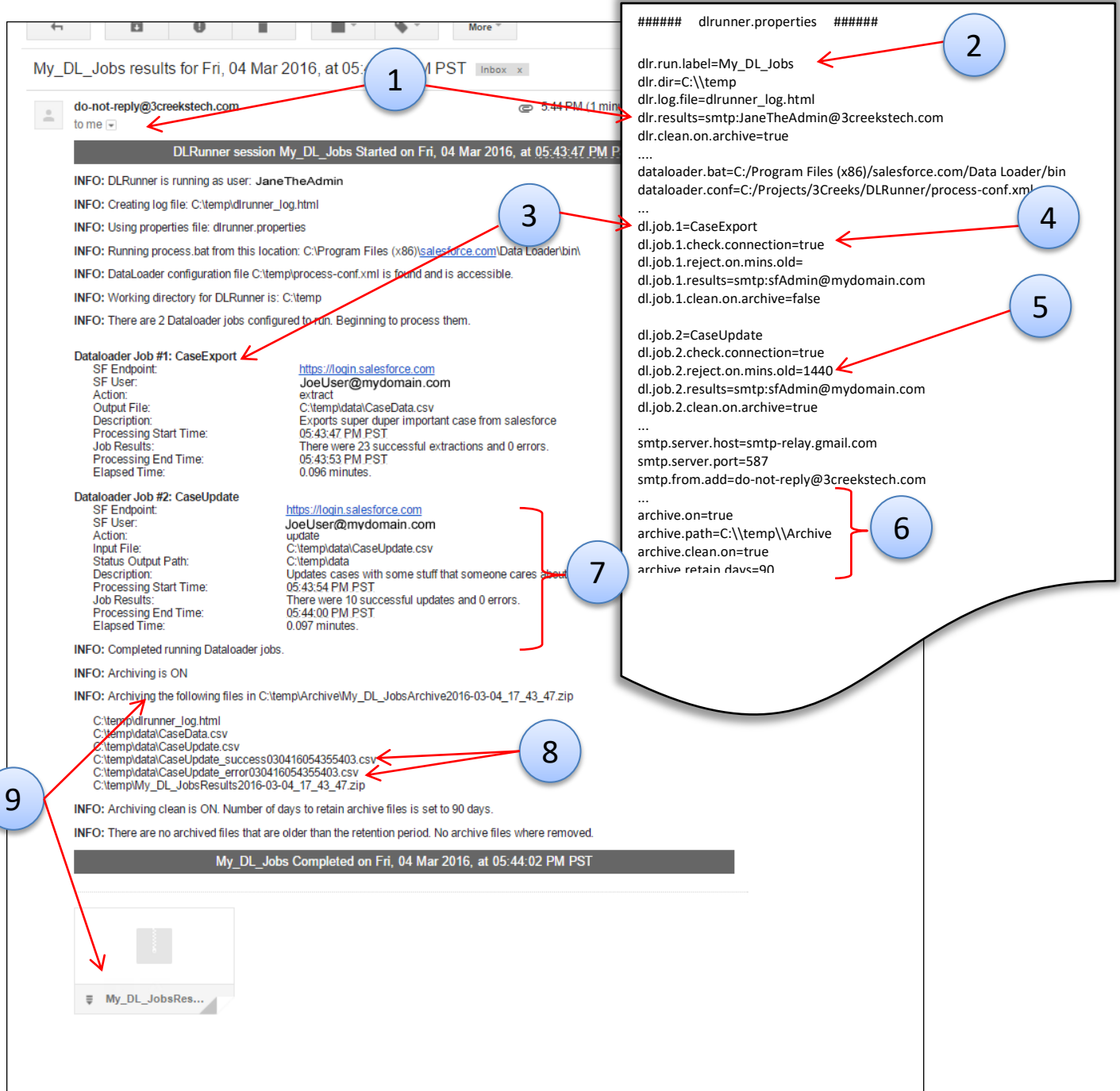
Without DLRRunner, this same scenario of five daily jobs and three weekly jobs would require you to configure eight separate entries in your Windows task scheduler.

DLRunner Notifications of Data Loader Job Results

In the DLRunner properties file, you can specify if you would like individual job results and DLRunner session results sent via email to users and administrators. Below are examples of results emails and how some of the DLRunner properties are used.

DLRunner Session Results

Session results emails are for administrators and specify the results of all Data Loader jobs for the session and contain information pertaining to the administrative tasks that DLRunner performs for you.



The image shows an email titled "My_DL_Jobs results for Fri, 04 Mar 2016, at 05:44 PM PST" and its properties. Numbered callouts (1-9) point to specific parts of the email and properties:

- 1**: Points to the email subject line.
- 2**: Points to the "dlrunner.properties" header in the properties pane.
- 3**: Points to the "DLRunner session My_DL_Jobs Started on Fri, 04 Mar 2016, at 05:43:47 PM PST" header.
- 4**: Points to the "dl.job.1=CaseExport" property.
- 5**: Points to the "dl.job.2=CaseUpdate" property.
- 6**: Points to the "archive.on=true" property.
- 7**: Points to the "Job Results:" section for "Dataloader Job #2: CaseUpdate".
- 8**: Points to the list of files being archived, including "C:\temp\My_DL_JobsResults2016-03-04_17_43_47.zip".
- 9**: Points to the "My_DL_Jobs Completed on Fri, 04 Mar 2016, at 05:44:02 PM PST" footer.

The email body contains the following information:

DLRunner session My_DL_Jobs Started on Fri, 04 Mar 2016, at 05:43:47 PM PST

INFO: DLRunner is running as user: JaneTheAdmin
 INFO: Creating log file: C:\temp\dlrunner_log.html
 INFO: Using properties file: dlrunner.properties
 INFO: Running process.bat from this location: C:\Program Files (x86)\salesforce.com\Data Loader\bin\
 INFO: DataLoader configuration file C:\temp\process-conf.xml is found and is accessible.
 INFO: Working directory for DLRunner is: C:\temp
 INFO: There are 2 DataLoader jobs configured to run. Beginning to process them.

Dataloader Job #1: CaseExport
 SF Endpoint: <https://login.salesforce.com>
 SF User: JoeUser@mydomain.com
 Action: extract
 Output File: C:\temp\data\CaseData.csv
 Description: Exports super duper important case from salesforce
 Processing Start Time: 05:43:47 PM PST
 Job Results: There were 23 successful extractions and 0 errors.
 Processing End Time: 05:43:53 PM PST
 Elapsed Time: 0.096 minutes.

Dataloader Job #2: CaseUpdate
 SF Endpoint: <https://login.salesforce.com>
 SF User: JoeUser@mydomain.com
 Action: update
 Input File: C:\temp\data\CaseUpdate.csv
 Status Output Path: C:\temp\data
 Description: Updates cases with some stuff that someone cares about
 Processing Start Time: 05:43:54 PM PST
 Job Results: There were 10 successful updates and 0 errors.
 Processing End Time: 05:44:00 PM PST
 Elapsed Time: 0.097 minutes.

INFO: Completed running DataLoader jobs.
 INFO: Archiving is ON
 INFO: Archiving the following files in C:\temp\Archive\My_DL_JobsArchive2016-03-04_17_43_47.zip
 C:\temp\dlrunner_log.html
 C:\temp\data\CaseData.csv
 C:\temp\data\CaseUpdate.csv
 C:\temp\data\CaseUpdate_success030416054355403.csv
 C:\temp\data\CaseUpdate_error030416054355403.csv
 C:\temp\My_DL_JobsResults2016-03-04_17_43_47.zip

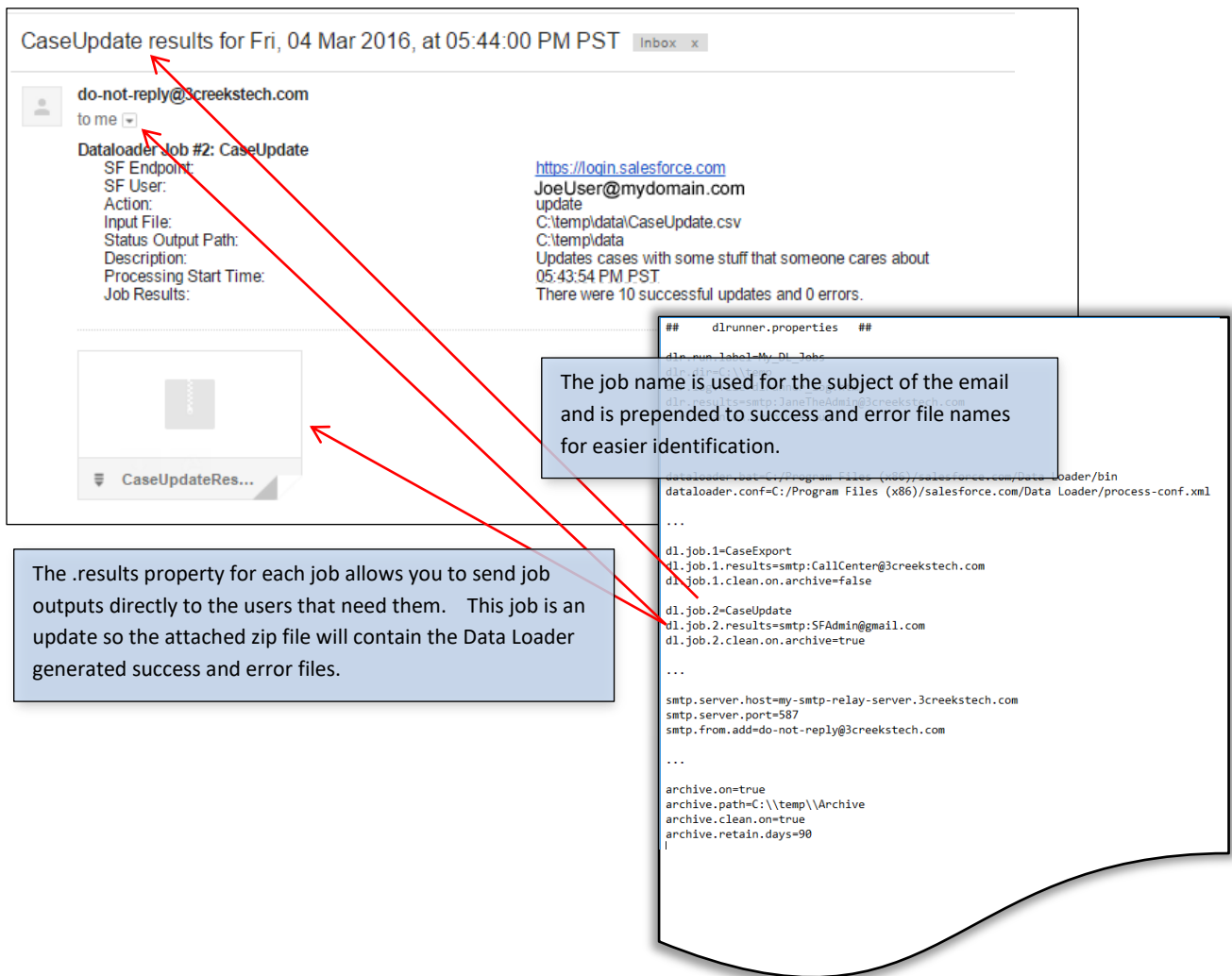
INFO: Archiving clean is ON. Number of days to retain archive files is set to 90 days.
 INFO: There are no archived files that are older than the retention period. No archive files were removed.

My_DL_Jobs Completed on Fri, 04 Mar 2016, at 05:44:02 PM PST

1	The session results email is intended for administrators. It shows a summary of all the jobs that ran for a session and the results of each job. Also, it shows the administrative tasks that DLRRunner performed such as archiving. The <code>dlr.results</code> property specifies who should receive the session results email.
2	Specify a label for the DLRRunner session in the DLRRunner properties file using the <code>dlr.run.label</code> property. This will allow you to differentiate groupings of Data Loader jobs. The label will also be used to name the archive and other session files.
3	Indicate the Data Loader jobs to run in the DLRRunner properties file, in the order in which they should be run, using the <code>dl.job.[number]</code> property. The value of <code>dl.job.[number]</code> value is the same the job name of the “id” tag of the bean attribute in your <code>process-conf.xml</code> file.
4	Configure DLRRunner to check that the Salesforce endpoint is accessible before attempting to run each Data Loader job. The endpoint that is checked is the one configured in the <code>process-conf.xml</code> file for each job.
5	If your insert, update, or upsert job depends on an input file, you can specify a value in minutes to ensure that the file being imported is current. If the file is older than the minutes indicated, the job will be skipped. Leave this value blank to not check the input file’s age and always process the input file.
6	When archiving is turned on, all files associated with a session are added to a single zip file and placed in the archive directory that you specify. You can also have DLRRunner delete old archives after a specified number of retention days.
7	For each job, DLRRunner pulls information about the job from the <code>process-conf.xml</code> and displays it for your reference along with the results of the job’s run.
8	Have you ever stared at a folder full of success and error files and wonder which one corresponds to the job you are interested in? Well, no more! DLRRunner automatically pre-pends the job name to success and error files so you can identify them easily.
9	Input source files, export files, success files, error files, logs, and everything else you need are grouped together for a session and put into one zip file that is attached to the session results email. If there is an issue with a Data Loader session, you have one place to look to get all the information you need to figure out what happened.

Job Results

Individual job results can be emailed directly to your users. The emails specify results of a single job and will include an attachment of the output of the job. For a Data Loader exports, the attachment will be the exported data. For inserts, updates, upserts, and deletes the attachment will contain the Data Loader generated success and error files.



The screenshot shows an email interface with the following details:

- Subject:** CaseUpdate results for Fri, 04 Mar 2016, at 05:44:00 PM PST
- From:** do-not-reply@3creekstech.com
- To:** me
- Job Name:** Dataloader Job #2: CaseUpdate
- Job Details:**
 - SF Endpoint:
 - SF User: JoeUser@mydomain.com
 - Action: update
 - Input File: C:\temp\data\CaseUpdate.csv
 - Status Output Path: C:\temp\data
 - Description: Updates cases with some stuff that someone cares about
 - Processing Start Time: 05:43:54 PM PST
 - Job Results: There were 10 successful updates and 0 errors.
- Attachment:** CaseUpdateRes...

Annotations:

- Job Name Annotation:** The job name is used for the subject of the email and is prepended to success and error file names for easier identification.
- Results Property Annotation:** The .results property for each job allows you to send job outputs directly to the users that need them. This job is an update so the attached zip file will contain the Data Loader generated success and error files.

DLRunner Configuration (dlrunner.properties):

```
## dlrunner.properties ##
dl.runner.label=My DL Job
dl.runner.description=Data Loader Job #2: CaseUpdate
dl.runner.config=C:\Program Files (x86)\salesforce.com\Data Loader\bin\dataloader.conf=C:\Program Files (x86)\salesforce.com\Data Loader\process-conf.xml
...
dl.job.1=CaseExport
dl.job.1.results=smtp:CallCenter@3creekstech.com
dl.job.1.clean.on.archive=false
...
dl.job.2=CaseUpdate
dl.job.2.results=smtp:SFAAdmin@gmail.com
dl.job.2.clean.on.archive=true
...
smtp.server.host=my-smtp-relay-server.3creekstech.com
smtp.server.port=587
smtp.from.add=do-not-reply@3creekstech.com
...
archive.on=true
archive.path=C:\\temp\\Archive
archive.clean.on=true
archive.retain.days=90
```

Validating Your DLRunner Configuration

Once you have configured DLRunner, be sure to run it in validation mode to check your configuration. To run DLRunner in validation mode, use the `-v` switch on the command line:

```
java -jar dlrrunner.jar -p DlrNightly.properties -v
```

When run in validation mode, DLRunner will go through all of the checks and validations that it would normally do during a live run, but it will not execute Data Loader for any of the jobs. Once the validation is complete, an email will be sent to the email address specified in the `dlr.results` attribute of the properties file. A sample of a validation email is below.

do-not-reply@3creektch.com Fri, Mar 8, 11:20 AM (4 days ago)

to me

DLRunner v2.0 by 3 Creeks Technologies LLC.
**** Validation Mode ** TestDL37 Started on Fri, 08 Mar 2019, at 11:20:32 AM PST ** Validation Mode ****

INFO: DLRunner is running as user: JaneTheAdmin

INFO: Creating log file: C:/Projects/3Creeks/DLRunner/Work/dlrunner_log.html

INFO: Using properties file: dlr.properties

INFO: Running process.bat from this location: C:/Program Files (x86)/salesforce.com/Data Loader/bin/

INFO: DataLoader configuration file C:/Projects/3Creeks/DLRunner/process-conf.xml is found and is accessible.

INFO: Working directory for DLRunner is: C:/Projects/3Creeks/DLRunner/Work

INFO: There are 2 Dataloader jobs configured to run. Beginning to process them.

Dataloader Job #1: CaseExport

SF Endpoint: <https://login.salesforce.com>

Connectivity Check: JoeUser@foo.bar

Action: extract

Output File: C:\Projects\3Creeks\DLRunner\data\CaseData.csv

Description: Exports cases from the system in order to test DLRunner export handling

Validation Start Time: 11:20:33 AM PST

Test Result: The output path C:\Projects\3Creeks\DLRunner\data is found and is accessible.

Processing End Time: 11:20:34 AM PST

Elapsed Time: 0.015 minutes.

Dataloader Job #2: CaseUpdate

SF Endpoint: <https://login.foo.salesforce.com>

Connectivity Check: **ERROR** - The endpoint <https://login.foo.salesforce.com> is NOT accessible. The connectivity check returned an unknown host exception. This could be because the host that is running Dataloader has no network access, has no Internet access, the endpoint is not reachable due to firewall rules, or that the endpoint does not exist. Due to this error the job cannot be run.

INFO: Archiving is ON

INFO: The smtp server smtp-relay@gmail.com is found and accessible on port 587

INFO: The Archive Path C:/Projects/3Creeks/DLRunner/Work/Archive is found and is accessible.

INFO: Archive Clean is ON.

INFO: Archive Retention Interval is 90 days.

TestDL37 Completed on Fri, 08 Mar 2019, at 11:20:55 AM PST

What's New in Version 2.0

New properties file entry: *dl.job.[job number].check.connection*

Sometime things change. Sometimes those things are network changes. Sometimes those network changes make other things inaccessible. So we've added a check that allows you to configure DLRRunner to check that the Salesforce endpoints that are specified in the *sfdc.endpoint* entry of your Dataloader's *process-conf.xml* file are accessible.

You configure the check by specifying 'true' for the *dl.job.[number].check.connection* property for each job in the DLRRunner properties file. DLRRunner will attempt to connect with the specified endpoint, and if successful, will evaluate the response code received back from the endpoint. A response code of 200 (Ok) constitutes a successful connection test. Any other response code will be considered and failed connection. The time out for the DLRRunner connectivity check is 45 seconds.

When the connectivity check is successful, you will see an additional entry in the results emails:

<u>Dataloader Job #2: CaseUpdate</u>	
SF Endpoint:	https://login.salesforce.com
Connectivity Check:	The endpoint https://login.salesforce.com is found and is accessible.
SF User:	ldoe@foobar.com
Action:	update
Input File:	C:\myfiles\CaseUpdate.csv
Status Output Path:	C:\myfile\DLRunner\work
Description:	Updates cases
Processing Start Time:	04:28:51 PM PST
Job Results:	There were 0 successful updates and 3 errors.
Processing End Time:	04:28:56 PM PST
Elapsed Time:	0.092 minutes.

With DLRRunner version 1, if there was a connectivity issue with the server running DLRRunner and Dataloader, your Dataloader job still attempts to run, but you would get the following generic error message:

DLRunner was not able to determine the status of this job. Most likely it did not complete successfully.

With DLRRunner version 2, the connectivity issue can be discovered upfront, preventing the Dataloader job from attempting to run, and the error message will be more specific about the issue:

ERROR - The endpoint <https://login.salesforce.com> is NOT accessible. The connectivity check returned an unknown host exception. This could be because the host that is running Dataloader has no network access, has no Internet access, the endpoint is not reachable due to firewall rules, or that the endpoint does not exist. Due to this error the job cannot be run.

Note that the DLRRunner connectivity check is not a guarantee that Data Loader will be able to communicate with Salesforce. There a number of firewall and other settings that can create scenarios where the DLRRunner connectivity check passes, but the Data Loader communications still fails.

Added connectivity check for the smtp.server.host and smtp.server.port entries

Similar to above, DLRunner will check to make sure the SMTP server configured in DLRunner properties is accessible on the port that is indicated. If the SMTP server cannot be reached, DLRunner will not attempt to send out any emails and log files can be found on the server running DLRunner.

Ability to specify a file reject age for Insert, Updates, and Upserts

If your DLRunner implementation gets input files from other sources and it is important to know that those input files are fresh, you can now specify an age in minutes that will make DLRunner skip a job if the input file is older than the specified number of minutes.

The new DLRunner properties entry is `dl.job.[job number].reject.on.mins.old`.

For instance, if you configure your first job like so:

```
dl.job.1.reject.on.mins.old=1440
```

Then if the source file is more than 24 hours old or more, the job will be skipped and the log entry for the job will indicate that the file was too old to process. If you do not care how old the source file is, just leave this entry blank and as long as the file is there and accessible, it will be processed regardless of age.

Note that this new properties file entry is only applicable to inserts, updates, and upserts. So if you configure it for an export job this properties attribute will be ignored.

Cleaned up formatting on emailed logs

The emailed log files had some formatting issues that should be cleaned up now.