Introduction

Event Log Files are subsets of salesforce.com logs where each file represents one type of log events that took place over one day.

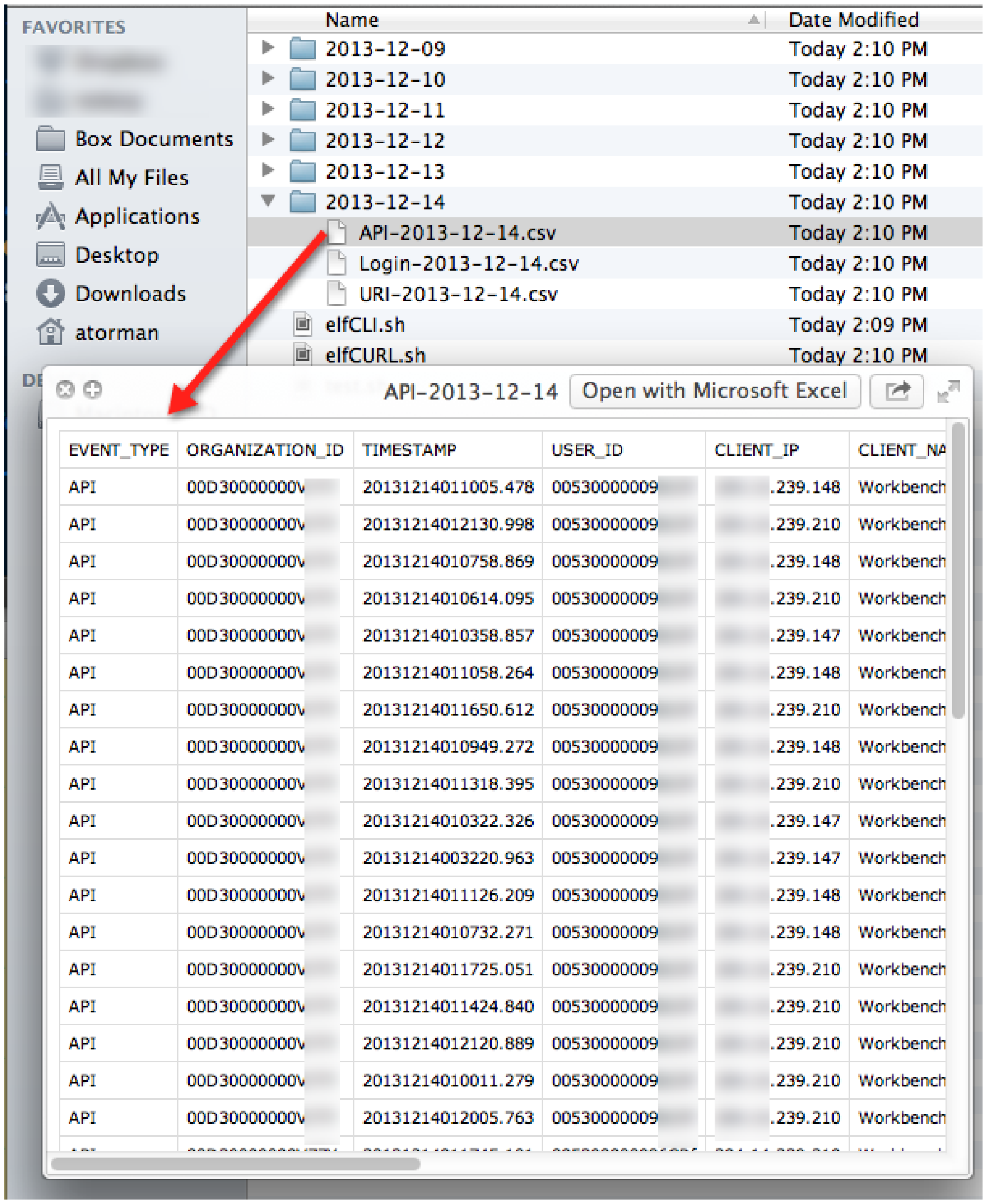
These files can grow quite large, consisting of hundreds of thousands of lines contained in a 150 MB file. As a result, normal tools like workbench and data loader become ineffective downloading these files to disk.

This document contains two bash scripts that can be run from your computer's terminal. While I recommend using the first script since it leverages the authentication power of the force.com CLI (Command Line Interface), both scripts will fundamentally solve the same problem.

Both scripts are designed to download a directory for each day selected (e.g. Yesterday, Last_Week, Last_n_Days:5, etc…) containing one file for each log line type.

These scripts have been verified on both MacOSX and Ubuntu Linux.

Please send any questions or comments to atorman@salesforce.com. Thanks for using Event Log Files!



Name	Date Modified
2013-12-09	Today 2:10 PM
2013-12-10	Today 2:10 PM
2013-12-11	Today 2:10 PM
2013-12-12	Today 2:10 PM
2013-12-13	Today 2:10 PM
2013-12-14	Today 2:10 PM
API-2013-12-14.csv	Today 2:10 PM
Login-2013-12-14.csv	Today 2:10 PM
URI-2013-12-14.csv	Today 2:10 PM
elfCLI.sh	Today 2:09 PM
elfCURL.sh	Today 2:10 PM

EVENT_TYPE	ORGANIZATION_ID	TIMESTAMP	USER_ID	CLIENT_IP	CLIENT_NA
API	00D30000000V	20131214011005.478	00530000009	.239.148	Workbench
API	00D30000000V	20131214012130.998	00530000009	.239.210	Workbench
API	00D30000000V	20131214010758.869	00530000009	.239.148	Workbench
API	00D30000000V	20131214010614.095	00530000009	.239.210	Workbench
API	00D30000000V	20131214010358.857	00530000009	.239.147	Workbench
API	00D30000000V	20131214011058.264	00530000009	.239.148	Workbench
API	00D30000000V	20131214011650.612	00530000009	.239.210	Workbench
API	00D30000000V	20131214010949.272	00530000009	.239.148	Workbench
API	00D30000000V	20131214011318.395	00530000009	.239.210	Workbench
API	00D30000000V	20131214010322.326	00530000009	.239.147	Workbench
API	00D30000000V	20131214003220.963	00530000009	.239.147	Workbench
API	00D30000000V	20131214011126.209	00530000009	.239.148	Workbench
API	00D30000000V	20131214010732.271	00530000009	.239.148	Workbench
API	00D30000000V	20131214011725.051	00530000009	.239.210	Workbench
API	00D30000000V	20131214011424.840	00530000009	.239.210	Workbench
API	00D30000000V	20131214012120.889	00530000009	.239.210	Workbench
API	00D30000000V	20131214010011.279	00530000009	.239.210	Workbench
API	00D30000000V	20131214012005.763	00530000009	.239.210	Workbench

Example of output from bash script

Bash Script #1: CURL and the Force.com CLI

*This is the recommended way of downloading Event Log Files to your hard drive

Pre-requisites:

1. download jq to parse JSON responses - http://stedolan.github.io/jq/ - you should copy jq to /usr/bin with execution (`chmod +x /usr/bin/jq`) rights or add to $PATH
2. download force.com CLI to manage authorization process - https://force-cli.heroku.com/ - you should copy force to /usr/bin with execution (`chmod +x /usr/bin/force`) rights

or add to $PATH

Advantages:

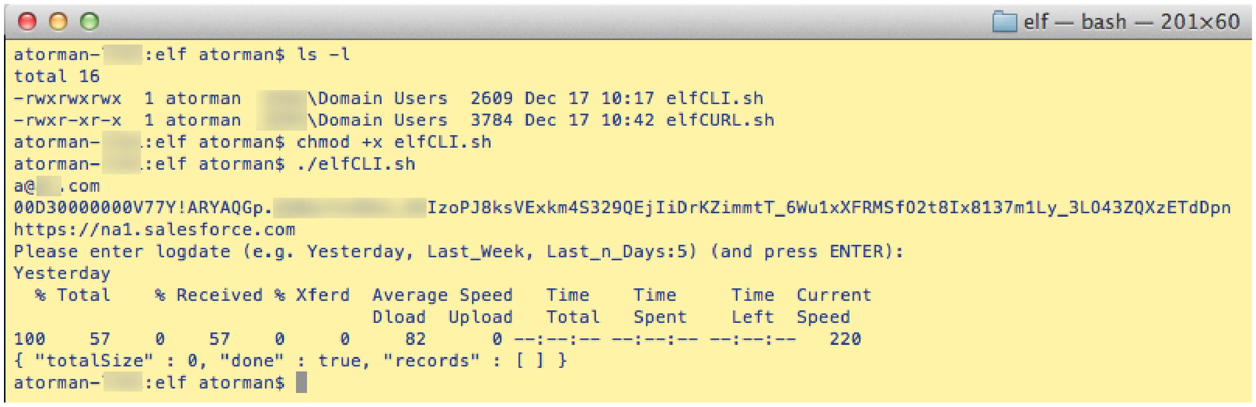
1. Useful automating the download for a single org
2. Authorization is more secure since the OAuth browser flow handles the password
3. Doesn't require creating a connected app to manage the authorization process

Disadvantages:

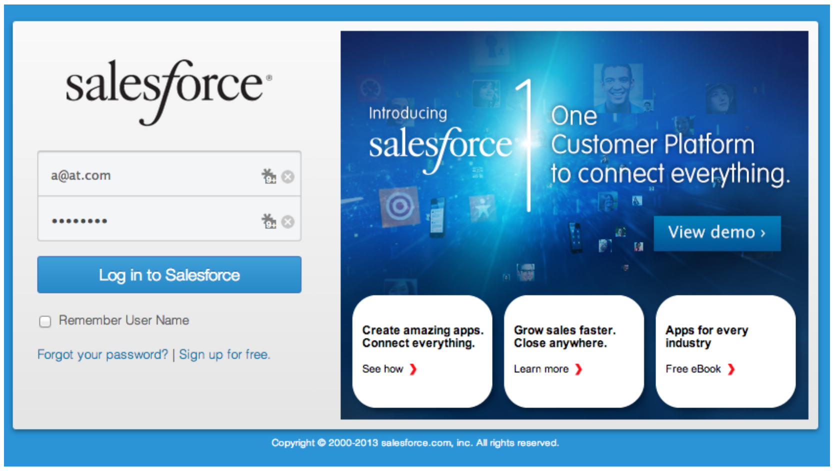
1. Not useful for automating across multiple orgs as each org will require the OAuth browser authentication flow

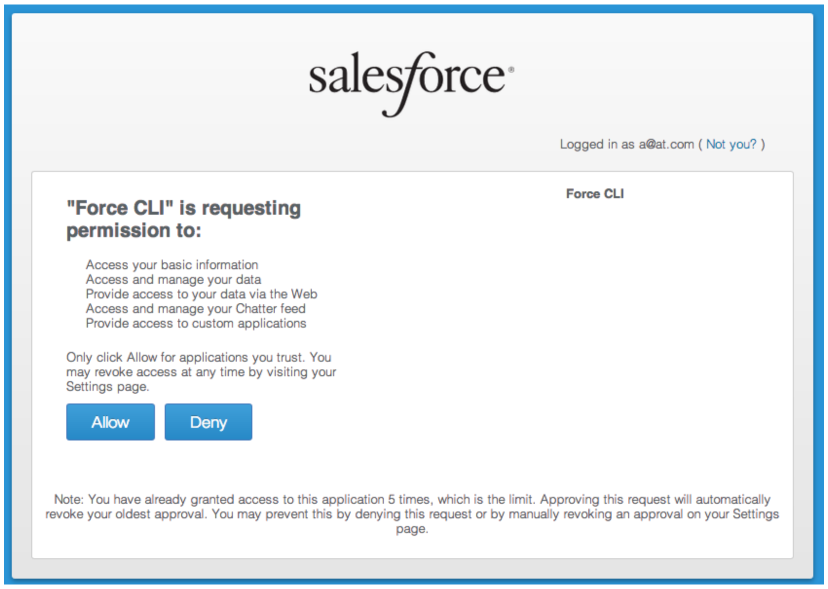
Directions:

1. save the following script as `elfCLI.sh` in the directory where you want to download the Event Log Files
2. make sure you have execution rights on the script (from the command line: `ls -l`). If you don't have execution rights, change them (`chmod +x elfCLI.sh`)
3. run the script (from the command line: `./elfCLI.sh`)



```
atorman-[redacted]:elf atorman$ ls -l
total 16
-rwxrwxrwx  1 atorman  [redacted]\Domain Users  2609 Dec 17 10:17 elfCLI.sh
-rwxr-xr-x  1 atorman  [redacted]\Domain Users  3784 Dec 17 10:42 elfCURL.sh
atorman-[redacted]:elf atorman$ chmod +x elfCLI.sh
atorman-[redacted]:elf atorman$ ./elfCLI.sh
a@[redacted].com
00D30000000V77Y!ARYAQGp.[redacted]IzoPJ8ksVExkm4S329QEjIiDrKZimmtT_6Wu1xXFRMSf02t8Ix8137m1Ly_3L043ZQXzETdDpn
https://na1.salesforce.com
Please enter logdate (e.g. Yesterday, Last_Week, Last_n_Days:5) (and press ENTER):
Yesterday
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    57    0    57    0     0     82      0 --:--:-- --:--:-- --:--:--   220
{ "totalSize" : 0, "done" : true, "records" : [ ] }
atorman-[redacted]:elf atorman$ 
```


salesforce
a@at.com
••••••••
Log in to Salesforce
Remember User Name
Forgot your password? | Sign up for free.
Introducing salesforce1
One Customer Platform to connect everything.
View demo ›
Create amazing apps. Connect everything.
See how ›
Grow sales faster. Close anywhere.
Learn more ›
Apps for every industry
Free eBook ›
Copyright © 2000-2013 salesforce.com, inc. All rights reserved.


Logged in as a@at.com (Not you?)

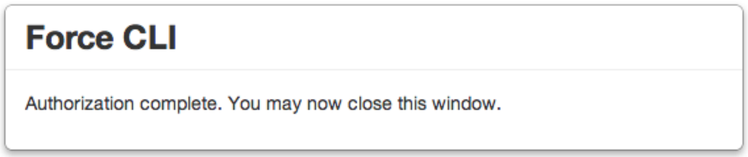
Force CLI

"Force CLI" is requesting permission to:

Access your basic information
Access and manage your data
Provide access to your data via the Web
Access and manage your Chatter feed
Provide access to custom applications

Only click Allow for applications you trust. You may revoke access at any time by visiting your Settings page.

Note: You have already granted access to this application 5 times, which is the limit. Approving this request will automatically revoke your oldest approval. You may prevent this by denying this request or by manually revoking an approval on your Settings page.


Force CLI

Authorization complete. You may now close this window.

Script:

```
#!/bin/bash
# Bash script to download EventLogFiles

# Pre-requisite: download jq - http://stedolan.github.io/jq/ to parse
JSON
# Pre-requisite: download force CLI - https://force-cli.heroku.com/

#login through OAuth flow to CLI
force login

#set username using force whoami
username=`force whoami | grep Username | sed 's/^Username: //'`
#echo ${username}

#set AccessToken from force accounts
#echo more ~/.force/accounts/${username}
access_token=`more ~/.force/accounts/${username} | jq -r
'.AccessToken'`
```

```
#echo ${access_token}

#set InstanceURL from force accounts
instance_url=`more ~/.force/accounts/${username} | jq -r
'.InstanceUrl'`
#echo ${instance_url}

#prompt the user to enter what date they want to get the logs for or
just press enter to take the default of 'Yesterday'
read -p "Please enter logdate (e.g. Yesterday, Last_Week,
Last_n_Days:5) (and press ENTER): " day
day=${day:-Yesterday}
#echo ${day}

#set elfs to the result of ELF query
elfs=`curl
${instance_url}/services/data/v29.0/query?q=Select+Id+,+EventType+,+L
ogDate+From+EventLogFile+Where+LogDate+=+${day} -H "Authorization:
Bearer ${access_token}" -H "X-PrettyPrint:1"`

#uncomment next line if you want to see the result of elfs
#echo ${elfs}

#uncomment next line if you want to see the array of Ids from the ELF
query
#echo ${elfs} | jq -r ".records[].Id"

#set the three variables to the array of Ids, EventTypes, and
LogDates which will be used when downloading the files into your
directory
ids=( $(echo ${elfs} | jq -r ".records[].Id") )
eventTypes=( $(echo ${elfs} | jq -r ".records[].EventType") )
logDates=( $(echo ${elfs} | jq -r ".records[].LogDate" | sed
's/'T.*'//' ) )

#loop through the array of results and download each file with the
following naming convention: EventType-LogDate.csv
for i in "${!ids[@]}"; do

    #uncomment the next three lines if you want to see the array of
Ids, EventTypes, and LogDates
    #echo "${i}: ${ids[$i]}"
    #echo "${i}: ${eventTypes[$i]}"
    #echo "${i}: ${logDates[$i]}"

    #make directory to store the files by date
    mkdir "${logDates[$i]}"

    #uncomment the next line to see the curl command to download log
files
    #echo "curl
\"${instance_url}/services/data/v29.0/sobjects/EventLogFile/${ids[$i]
}/LogFile\" -H \"Authorization: Bearer ${access_token}\" -H
\"X-PrettyPrint:1\" -o \"${eventTypes[$i]}-${logDates[$i]}.csv\""

    #download files into the logDate directory
    curl
"${instance_url}/services/data/v29.0/sobjects/EventLogFile/${ids[$i]}
/LogFile" -H "Authorization: Bearer ${access_token}" -H
"X-PrettyPrint:1" -o
"${logDates[$i]}/${eventTypes[$i]}-${logDates[$i]}.csv"
done
```

Bash Script #2: Native CURL

Pre-requisites:

1. download jq to parse JSON responses - http://stedolan.github.io/jq/ - you should copy jq to /usr/bin with execution (`chmod +x /usr/bin/jq`) rights or add to $PATH
2. create a connected app and substitute the client_id / client_secret in the following script

Advantages:

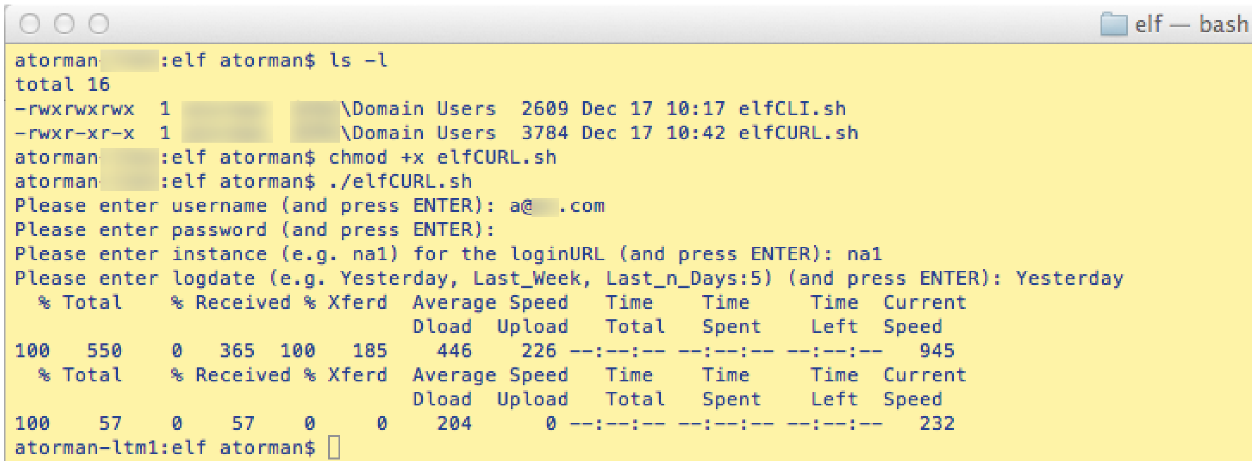
1. Only one pre-requisite makes for a faster setup
2. Useful automating the download across multiple orgs

Disadvantages:

1. Password less secure in that it can easily be stored in clear text, defaulted, or displayed to a user
2. Requires knowledge of what instance (na1) your org is on
3. Requires creating a connected app client_id/client_secret that is stored in clear text. Creating a connected app is not difficult (https://help.salesforce.com/htviewhelpdoc?id=connected_app_create.htm&siteLang=en_US); however, it should be done prior to using this script

Directions:

1. save the following script as elfCURL.sh in the directory where you want to download the Event Log Files
2. make sure you have execution rights on the script (from the command line: ls -l). If you don't have execution rights, change them (chmod +x elfCURL.sh)
3. run the script (from the command line: ./elfCURL.sh)



```
atorman[redacted]:elf atorman$ ls -l
total 16
-rwxrwxrwx  1 [redacted]  [redacted]\Domain Users  2609 Dec 17 10:17 elfCLI.sh
-rwxr-xr-x  1 [redacted]  [redacted]\Domain Users  3784 Dec 17 10:42 elfCURL.sh
atorman[redacted]:elf atorman$ chmod +x elfCURL.sh
atorman[redacted]:elf atorman$ ./elfCURL.sh
Please enter username (and press ENTER): a@[redacted].com
Please enter password (and press ENTER):
Please enter instance (e.g. na1) for the loginURL (and press ENTER): na1
Please enter logdate (e.g. Yesterday, Last_Week, Last_n_Days:5) (and press ENTER): Yesterday
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   550    0   365  100   185    446    226 --:--:-- --:--:-- --:--:--   945
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    57    0    57    0     0    204      0 --:--:-- --:--:-- --:--:--   232
atorman-ltm1:elf atorman$ 
```

Script:

```
#!/bin/bash
# Bash script to download EventLogFiles
# Pre-requisite: download - http://stedolan.github.io/jq/ to parse
JSON

#prompt the user to enter their username or uncomment #username line
for testing purposes
read -p "Please enter username (and press ENTER): " username

#prompt the user to enter their password
read -s -p "Please enter password (and press ENTER): " password

#prompt the user to enter their instance end-point
echo
read -p "Please enter instance (e.g. na1) for the loginURL (and press
ENTER): " instance

#uncomment next line to set default for testing purposes
#instance=${instance:-na1}
```

```
#prompt the user to enter the date for the logs they want to download
read -p "Please enter logdate (e.g. Yesterday, Last_Week,
Last_n_Days:5) (and press ENTER): " day

#uncomment next line to set default for testing purposes
#day=${day:-Yesterday}

#uncomment next line if you want to check your username, instance, or
password input
#echo "Username ${username} and instance ${instance} and day ${day}"
#and password ${password} "

#set access_token for OAuth flow
#change client_id and client_secret to your own connected app
access_token=`curl
https://${instance}.salesforce.com/services/oauth2/token -d
"grant_type=password" -d
"client_id=3MVG99OxTyEMCQ3hSjz15qIUWtJCt6fADLrtDeTQA9Lb.liLd5pGQXzLy9
qjrph.UIv2UkJWtwt3TnxQ4KhuD" -d "client_secret=2447913710583473942"
-d "username=${username}" -d "password=${password}" -H
"X-PrettyPrint:1" | jq -r '.access_token'`

#uncomment next line if you want to check your access token
#echo "Access token: ${access_token}"

#uncomment next line if you want to see the curl command to query ELF
#echo "curl
https://${instance}.salesforce.com/services/data/v29.0/query?q=Select
+Id+From+EventLogFile+Where+LogDate+=+${day} -H 'Authorization:
Bearer ${access_token}' -H \"X-PrettyPrint:1\""

#set elfs to the result of ELF query
elfs=`curl
https://${instance}.salesforce.com/services/data/v29.0/query?q=Select
+Id+,+EventType+,+LogDate+From+EventLogFile+Where+LogDate+=+${day} -H
"Authorization: Bearer ${access_token}" -H "X-PrettyPrint:1"`

#uncomment next line if you want to see the result of elfs
#echo ${elfs}

#uncomment next line if you want to see the array of Ids from the ELF
query
#echo ${elfs} | jq -r ".records[].Id"
```

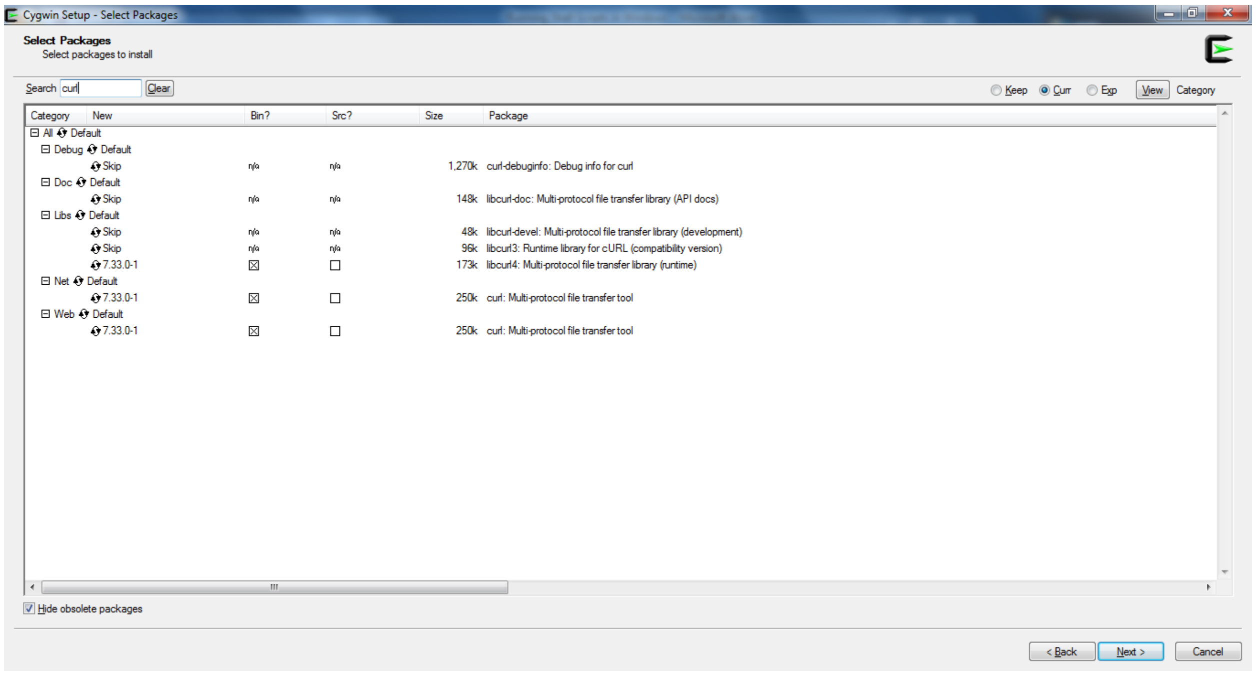
```
#set the three variables to the array of Ids, EventTypes, and
LogDates which will be used when downloading the files into your
directory
ids=( $(echo ${elfs} | jq -r ".records[].Id") )
eventTypes=( $(echo ${elfs} | jq -r ".records[].EventType") )
logDates=( $(echo ${elfs} | jq -r ".records[].LogDate" | sed
's/'T.*'//' ) )

#loop through the array of results and download each file with the
following naming convention: EventType-LogDate.csv
for i in "${!ids[@]}"; do

    #uncomment the next three lines if you want to see the array of
Ids, EventTypes, and LogDates
    #echo "${i}: ${ids[$i]}"
    #echo "${i}: ${eventTypes[$i]}"
    #echo "${i}: ${logDates[$i]}"

    #make directory to store the files by date
    mkdir "${logDates[$i]}"

    #uncomment the next line to see the curl command to download log
files
    #echo "curl
\"https://${instance}.salesforce.com/services/data/v29.0/sobjects/Eve
ntLogFile/${ids[$i]}/LogFile\" -H \"Authorization: Bearer
${access_token}\" -H \"X-PrettyPrint:1\" -o
\"${eventTypes[$i]}-${logDates[$i]}.csv\""

    #download files into the logDate directory
    curl
"https://${instance}.salesforce.com/services/data/v29.0/sobjects/Even
tLogFile/${ids[$i]}/LogFile" -H "Authorization: Bearer
${access_token}" -H "X-PrettyPrint:1" -o
"${logDates[$i]}/${eventTypes[$i]}-${logDates[$i]}.csv"
done
```

Downloading Event Log Files using Windows 7

Pre-requisites for all methods specified in this doc:

Download cygwin for windows: http://cygwin.com/install.html - you will need to know whether you have 32 or 64 bit Windows to download the correct binary.

I suggest doing a full install; however, if you do a partial install, make sure to include curl:


Cygwin Setup - Select Packages

Select Packages
Select packages to install

Search: curl

Category	New	Bin?	Src?	Size	Package
⊟ All Default					
⊟ Debug Default					
	Skip	n/a	n/a	1,270k	curl-debuginfo: Debug info for curl
⊟ Doc Default					
	Skip	n/a	n/a	148k	libcurl-doc: Multi-protocol file transfer library (API docs)
⊟ Libs Default					
	Skip	n/a	n/a	48k	libcurl-devel: Multi-protocol file transfer library (development)
	Skip	n/a	n/a	96k	libcurl3: Runtime library for cURL (compatibility version)
	7.33.0-1	☒	☐	173k	libcurl4: Multi-protocol file transfer library (runtime)
⊟ Net Default					
	7.33.0-1	☒	☐	250k	curl: Multi-protocol file transfer tool
⊟ Web Default					
	7.33.0-1	☒	☐	250k	curl: Multi-protocol file transfer tool

☑ Hide obsolete packages

To download an Event Log File using curl in cygwin:

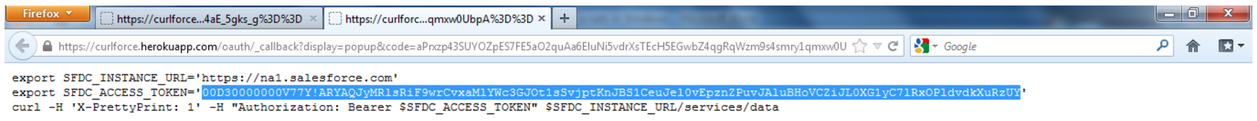
This is not an ideal solution since it involves a lot of copy and paste. However, it will get you started with Event Log Files if you are not familiar with coding integrations using the Salesforce REST API. The ideal solution is to leverage the power of an integration service that is more robust when downloading large CSV files than using a browser based client such as workbench or an apex built application.

Steps:

1. Determine what instance you are on – in any REST API call, you will need to substitute this instance in the beginning of the HTTP statement. In the following examples, I use na1; however, you should use your own (e.g. emea)
2. Get a REST API access token by going to https://curlforce.herokuapp.com

    a. Enter your username and password – this site uses Oauth and does not store your username or password – it simply returns the access token and other authorization information.

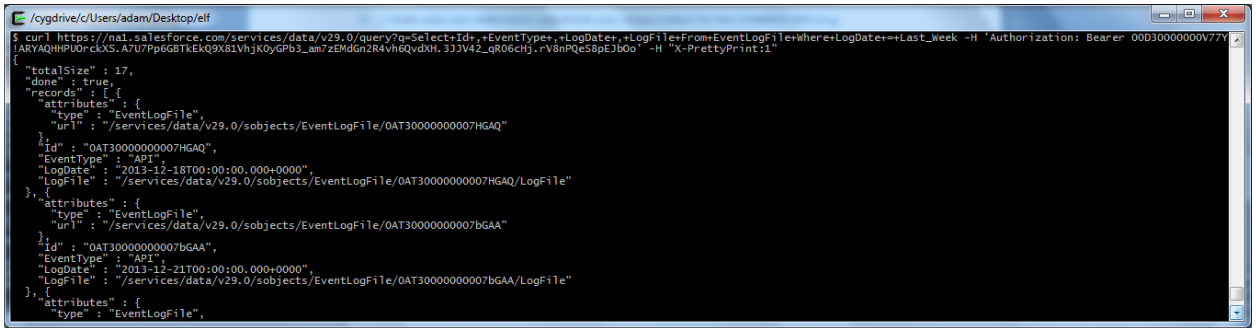
3. Copy your access token



```
export SFDC_INSTANCE_URL='https://na1.salesforce.com'
export SFDC_ACCESS_TOKEN='00D30000000V77Y!ARYAQJyMRlsRiF9wrCvxaM1YWc3GJOt1sSvjptKnJBS1CeuJel0vEpznZPuvJAluBHoVCZiJL0XG1yC71RxOPldvdkXuRzUY'
curl -H 'X-PrettyPrint: 1' -H "Authorization: Bearer $SFDC_ACCESS_TOKEN" $SFDC_INSTANCE_URL/services/data
```

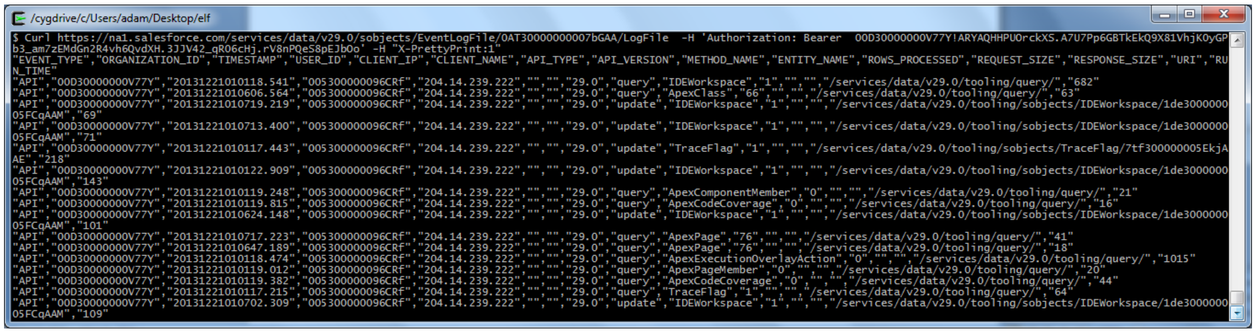
4. Create your curl statement in an ASCII editor like notepad. Substitute your access token for the {ACCESSTOKEN} and {INSTANCE} for your server instance in the following example
    a. curl https://{ INSTANCE}.salesforce.com/services/data/v29.0/query?q=Select+Id+,+EventType+,+LogDate+,+LogFile+From+EventLogFile+Where+LogDate+=+Last_Week -H 'Authorization: Bearer { ACCESSTOKEN } -H "X-PrettyPrint:1"
    b. for example:
    c. **curl https://na1.salesforce.com/services/data/v29.0/query?q=Select+Id+,+EventType+,+LogDate+,+LogFile+From+EventLogFile+Where+LogDate+=+Last_Week -H 'Authorization: Bearer 00D30000000V77Y!ARYAQHHPUOrckXS.A7U7Pp6GBTkEkQ9X81VhjK0yGPb3_am7zEMdGn2R4vh6QvdXH.3JJV42_qR06cHj.rV8nPQeS8pEJbOo' -H "X-PrettyPrint:1"**
5. Launch cygwin terminal (e.g. Start > Cygwin Terminal)

6. Paste your curl statement into the terminal



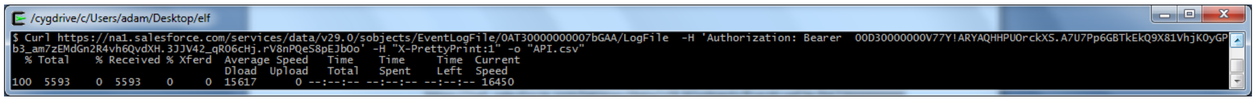
```
$ curl https://na1.salesforce.com/services/data/v29.0/query?q=Select+Id+,+EventType+,+LogDate+,+LogFile+From+EventLogFile+Where+LogDate+=+Last_Week -H 'Authorization: Bearer 00D30000000V77Y
!ARYAQHHPUOrckXS.A7U7Pp6GBTkEkQ9X81VhjKOyGPb3_am7zEMdGn2R4vh6QvdXH.3JJV42_qR06cHj.rV8nPQeS8pEJbOo' -H "X-PrettyPrint:1"
{
  "totalSize" : 17,
  "done" : true,
  "records" : [ {
    "attributes" : {
      "type" : "EventLogFile",
      "url" : "/services/data/v29.0/sobjects/EventLogFile/0AT30000000007HGAQ"
    },
    "Id" : "0AT30000000007HGAQ",
    "EventType" : "API",
    "LogDate" : "2013-12-18T00:00:00.000+0000",
    "LogFile" : "/services/data/v29.0/sobjects/EventLogFile/0AT30000000007HGAQ/LogFile"
  }, {
    "attributes" : {
      "type" : "EventLogFile",
      "url" : "/services/data/v29.0/sobjects/EventLogFile/0AT30000000007bGAA"
    },
    "Id" : "0AT30000000007bGAA",
    "EventType" : "API",
    "LogDate" : "2013-12-21T00:00:00.000+0000",
    "LogFile" : "/services/data/v29.0/sobjects/EventLogFile/0AT30000000007bGAA/LogFile"
  }, {
    "attributes" : {
      "type" : "EventLogFile",
```

7. You should get a series of JSON records returned including the actual log file urls.
8. Copy any LogFile url (e.g. **`/services/data/v29.0/sobjects/EventLogFile/0AT30000000007bGAA/LogFile`**) and substitute it into your curl statement. Substitute your access token for the {ACCESSTOKEN} and {INSTANCE} for your server instance in the following example
    a. Curl https://{INSTANCE}.salesforce.com/services/data/v29.0/sobjects/EventLogFile/0AT30000000007bGAA/LogFile -H 'Authorization: Bearer {ACCESSTOKEN} -H "X-PrettyPrint:1"
9. You should get a series of CSV rows returned



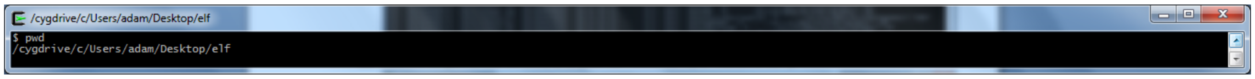
```
$ Curl https://na1.salesforce.com/services/data/v29.0/sobjects/EventLogFile/0AT30000000007bGAA/LogFile  -H 'Authorization: Bearer  00D30000000V77Y!ARYAQHHPUOrckXS.A7U7Pp6GBTkEkQ9X81VhjKOyGP
b3_am7zEMdGn2R4vh6QvdXH.3JJV42_qR06cHj.rV8nPQeS8pEJbOo' -H "X-PrettyPrint:1"
"EVENT_TYPE","ORGANIZATION_ID","TIMESTAMP","USER_ID","CLIENT_IP","CLIENT_NAME","API_TYPE","API_VERSION","METHOD_NAME","ENTITY_NAME","ROWS_PROCESSED","REQUEST_SIZE","RESPONSE_SIZE","URI","RU
N_TIME"
"API","00D30000000V77Y","20131221010118.541","005300000096CRf","204.14.239.222","","","29.0","query","IDEWorkspace","1","","","/services/data/v29.0/tooling/query/","682"
"API","00D30000000V77Y","20131221010606.564","005300000096CRf","204.14.239.222","","","29.0","query","ApexClass","66","","","/services/data/v29.0/tooling/query/","63"
"API","00D30000000V77Y","20131221010719.219","005300000096CRf","204.14.239.222","","","29.0","update","IDEWorkspace","1","","","/services/data/v29.0/tooling/sobjects/IDEWorkspace/1de3000000
05FCqAAM","69"
"API","00D30000000V77Y","20131221010713.400","005300000096CRf","204.14.239.222","","","29.0","update","IDEWorkspace","1","","","/services/data/v29.0/tooling/sobjects/IDEWorkspace/1de3000000
05FCqAAM","71"
"API","00D30000000V77Y","20131221010117.443","005300000096CRf","204.14.239.222","","","29.0","update","TraceFlag","1","","","/services/data/v29.0/tooling/sobjects/TraceFlag/7tf300000005EkjA
AE","218"
"API","00D30000000V77Y","20131221010122.909","005300000096CRf","204.14.239.222","","","29.0","update","IDEWorkspace","1","","","/services/data/v29.0/tooling/sobjects/IDEWorkspace/1de3000000
05FCqAAM","143"
"API","00D30000000V77Y","20131221010119.248","005300000096CRf","204.14.239.222","","","29.0","query","ApexComponentMember","0","","","/services/data/v29.0/tooling/query/","21"
"API","00D30000000V77Y","20131221010119.815","005300000096CRf","204.14.239.222","","","29.0","query","ApexCodeCoverage","0","","","/services/data/v29.0/tooling/query/","16"
"API","00D30000000V77Y","20131221010624.148","005300000096CRf","204.14.239.222","","","29.0","update","IDEWorkspace","1","","","/services/data/v29.0/tooling/sobjects/IDEWorkspace/1de3000000
05FCqAAM","101"
"API","00D30000000V77Y","20131221010717.223","005300000096CRf","204.14.239.222","","","29.0","query","ApexPage","76","","","/services/data/v29.0/tooling/query/","41"
"API","00D30000000V77Y","20131221010647.189","005300000096CRf","204.14.239.222","","","29.0","query","ApexPage","76","","","/services/data/v29.0/tooling/query/","18"
"API","00D30000000V77Y","20131221010118.474","005300000096CRf","204.14.239.222","","","29.0","query","ApexExecutionOverlayAction","0","","","/services/data/v29.0/tooling/query/","1015"
"API","00D30000000V77Y","20131221010119.012","005300000096CRf","204.14.239.222","","","29.0","query","ApexPageMember","0","","","/services/data/v29.0/tooling/query/","20"
"API","00D30000000V77Y","20131221010119.382","005300000096CRf","204.14.239.222","","","29.0","query","ApexCodeCoverage","0","","","/services/data/v29.0/tooling/query/","44"
"API","00D30000000V77Y","20131221010117.215","005300000096CRf","204.14.239.222","","","29.0","query","TraceFlag","1","","","/services/data/v29.0/tooling/query/","64"
"API","00D30000000V77Y","20131221010702.309","005300000096CRf","204.14.239.222","","","29.0","update","IDEWorkspace","1","","","/services/data/v29.0/tooling/sobjects/IDEWorkspace/1de3000000
05FCqAAM","109"
```

10. Add the output to CSV command to your curl statement (-o "filename.csv")



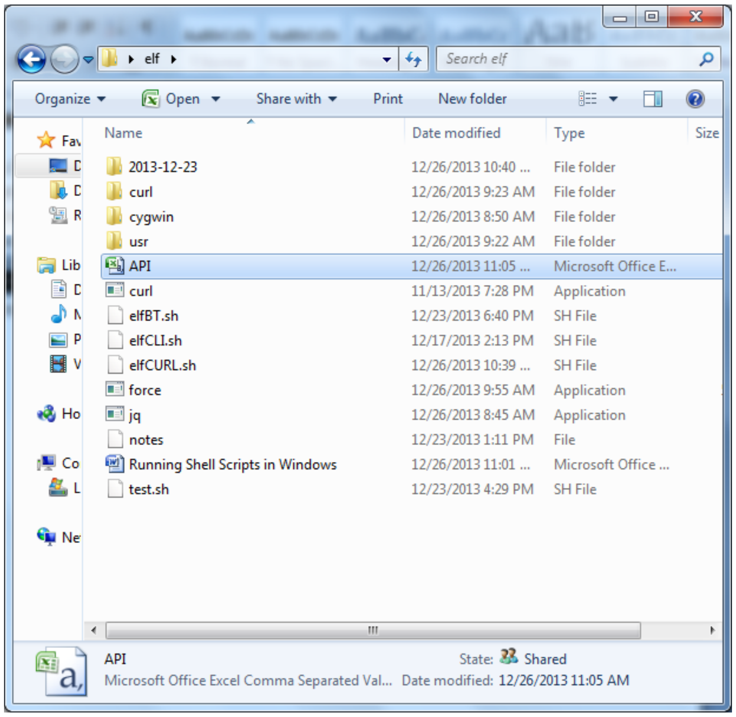
```
$ Curl https://na1.salesforce.com/services/data/v29.0/sobjects/EventLogFile/0AT30000000007bGAA/LogFile  -H 'Authorization: Bearer  00D30000000V77Y!ARYAQHHPUOrckXS.A7U7Pp6GBTkEkQ9X81VhjKOyGP
b3_am7zEMdGn2R4vh6QvdXH.3JJV42_qR06cHj.rV8nPQeS8pEJbOo' -H "X-PrettyPrint:1" -o "API.csv"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  5593    0  5593    0     0  15617      0 --:--:-- --:--:-- --:--:-- 16450
```

    a. Curl https://{INSTANCE}.salesforce.com/services/data/v29.0/sobjects/EventLogFile/0AT30000000007bGAA/LogFile -H 'Authorization: Bearer {ACCESSTOKEN} -H "X-PrettyPrint:1" -o "API.csv"
11. Check the directory where you downloaded the file using the 'pwd' command in cygwin



```
$ pwd
/cygdrive/c/Users/adam/Desktop/elf
```

12. View the directory in your windows explorer

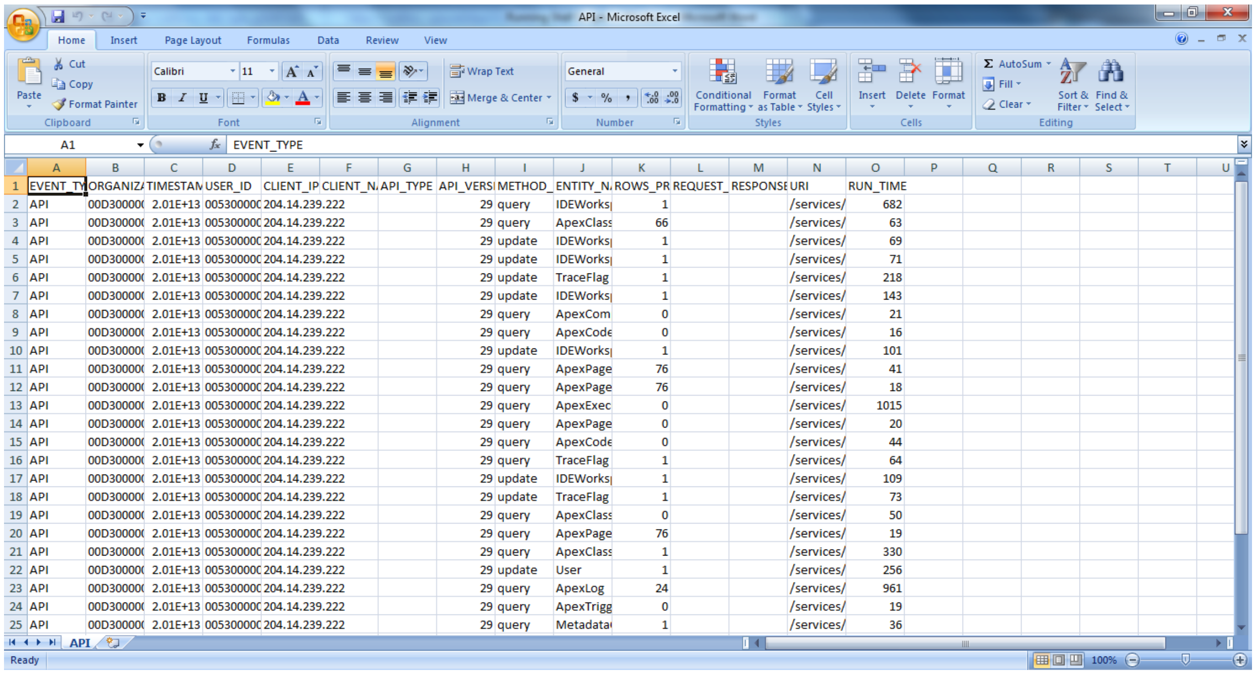


elf

Name	Date modified	Type
2013-12-23	12/26/2013 10:40 ...	File folder
curl	12/26/2013 9:23 AM	File folder
cygwin	12/26/2013 8:50 AM	File folder
usr	12/26/2013 9:22 AM	File folder
API	12/26/2013 11:05 ...	Microsoft Office E...
curl	11/13/2013 7:28 PM	Application
elfBT.sh	12/23/2013 6:40 PM	SH File
elfCLI.sh	12/17/2013 2:13 PM	SH File
elfCURL.sh	12/26/2013 10:39 ...	SH File
force	12/26/2013 9:55 AM	Application
jq	12/26/2013 8:45 AM	Application
notes	12/23/2013 1:11 PM	File
Running Shell Scripts in Windows	12/26/2013 11:01 ...	Microsoft Office ...
test.sh	12/23/2013 4:29 PM	SH File

API
Microsoft Office Excel Comma Separated Val... State: Shared
Date modified: 12/26/2013 11:05 AM

13. Open your downloaded file to view the contents



API - Microsoft Excel

A1 EVENT_TYPE

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	EVENT_TY	ORGANIZA	TIMESTAN	USER_ID	CLIENT_IP	CLIENT_N.	API_TYPE	API_VERSI	METHOD_	ENTITY_N.	ROWS_PR	REQUEST_	RESPONSE	URI	RUN_TIME
2	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	query	IDEWorks	1			/services/	682
3	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	query	ApexClass	66			/services/	63
4	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	update	IDEWorks	1			/services/	69
5	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	update	IDEWorks	1			/services/	71
6	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	update	TraceFlag	1			/services/	218
7	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	update	IDEWorks	1			/services/	143
8	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	query	ApexCom	0			/services/	21
9	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	query	ApexCode	0			/services/	16
10	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	update	IDEWorks	1			/services/	101
11	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	query	ApexPage	76			/services/	41
12	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	query	ApexPage	76			/services/	18
13	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	query	ApexExec	0			/services/	1015
14	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	query	ApexPage	0			/services/	20
15	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	query	ApexCode	0			/services/	44
16	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	query	TraceFlag	1			/services/	64
17	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	update	IDEWorks	1			/services/	109
18	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	update	TraceFlag	1			/services/	73
19	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	query	ApexClass	0			/services/	50
20	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	query	ApexPage	76			/services/	19
21	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	query	ApexClass	1			/services/	330
22	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	update	User	1			/services/	256
23	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	query	ApexLog	24			/services/	961
24	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	query	ApexTrigg	0			/services/	19
25	API	00D30000(	2.01E+13	00530000C	204.14.239.222			29	query	Metadata(	1			/services/	36

To Download Event Log Files using a Shell Script

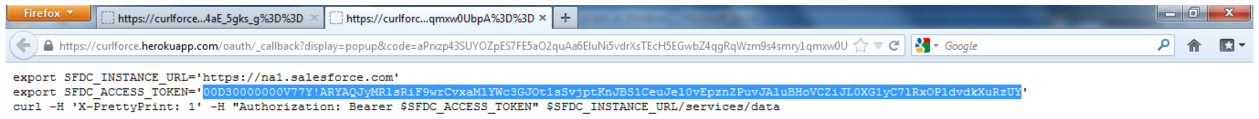
This technique will be more optimal than processing curl directly in cygwin. You will still need to copy the access token into the shell script and save it before running it in cygwin; however, it should generate a series of downloaded files automatically based on a date range (e.g. Yesterday, Last_Week, Last_n_Days:5, etc…).

Pre-requisites:

1. Create a folder in windows explorer (I recommend on your desktop for simplicity) called 'elf' to store your Event Log Files.
2. Download JQ to parse JSON - http://stedolan.github.io/jq/download/ - you will need to know whether you have 32 or 64 bit Windows to download the correct binary. Copy jq.exe to your newly created directory (/Desktop/elf).

Steps:

1. Get a REST API access token by going to https://curlforce.herokuapp.com
    a. Enter your username and password – this site uses Oauth and does not store your username or password – it simply returns the access token and other authorization information.
2. Copy your access token


Firefox
https://curlforce...4aE_5gks_g%3D%3D
https://curlforc...qmxw0UbpA%3D%3D
https://curlforce.herokuapp.com/oauth/_callback?display=popup&code=aPrxzp43SUYOZpES7FE5aO2quAa6EluNi5vdrXsTEcH5EGwbZ4qgRqWzm9s4smry1qmxw0U
Google

```
export SFDC_INSTANCE_URL='https://na1.salesforce.com'
export SFDC_ACCESS_TOKEN='00D30000000V77Y!ARYAQJyMRlsRiF9wrCvxaMlYWc3GJOt1sSvjptKnJBS1CeuJel0vEpznZPuvJAluBHoVCZiJL0XG1yC7lRxOPldvdkXuRzUY'
curl -H 'X-PrettyPrint: 1' -H "Authorization: Bearer $SFDC_ACCESS_TOKEN" $SFDC_INSTANCE_URL/services/data
```

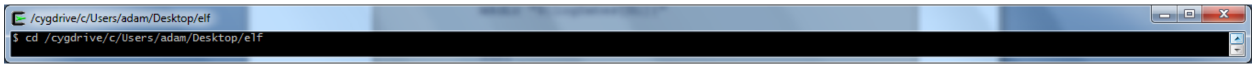
3. Paste your access token into the following script where you find {ACCESSTOKEN} and save it in your newly created directory (e.g. /Desktop/elf) as **`elfCURL.sh`**:

```
#!/bin/bash
# Bash script to download EventLogFiles
# Pre-requisite: download - http://stedolan.github.io/jq/
to parse JSON

#prompt the user to enter their instance end-point
echo
```

```
read -p "Please enter instance (e.g. emea) for the
loginURL (and press ENTER): " instance

#uncomment next line to set default for testing purposes
- default currently set to na1
instance=${instance:-na1}

#prompt the user to enter the date for the logs they want
to download
read -p "Please enter logdate (e.g. Yesterday, Last_Week,
Last_n_Days:5) (and press ENTER): " day

#uncomment next line to set default for testing purposes
- default currently set to last 4 days
day=${day:-Last_n_Days:4}

#uncomment next line if you want to check your username,
instance, or password input
echo "instance ${instance} and day ${day}"

#set elfs to the result of ELF query
elfs=`curl
https://${instance}.salesforce.com/services/data/v29.0/qu
ery?q=Select+Id+,+EventType+,+LogDate+From+EventLogFile+W
here+LogDate+=+${day} -H 'Authorization: Bearer
{ACCESSTOKEN}' -H "X-PrettyPrint:1"`

#uncomment next line if you want to see the result of
elfs
#echo ${elfs}

#uncomment next line if you want to see the array of Ids
from the ELF query
#echo ${elfs} | ./jq -r ".records[].Id"

#set the three variables to the array of Ids, EventTypes,
and LogDates which will be used when downloading the
files into your directory
ids=( $(echo ${elfs} | ./jq -r ".records[].Id" | sed 's/[
\t]*$//') )
eventTypes=( $(echo ${elfs} | ./jq -r
".records[].EventType" | sed 's/[ \t]*$//') )
logDates=( $(echo ${elfs} | ./jq -r ".records[].LogDate"
| sed 's/'T.*'//' | sed 's/[ \t]*$//') )

#loop through the array of results and download each file
with the following naming convention: EventType-
LogDate.csv
for i in "${!ids[@]}"; do

    #uncomment the next three lines if you want to see
```

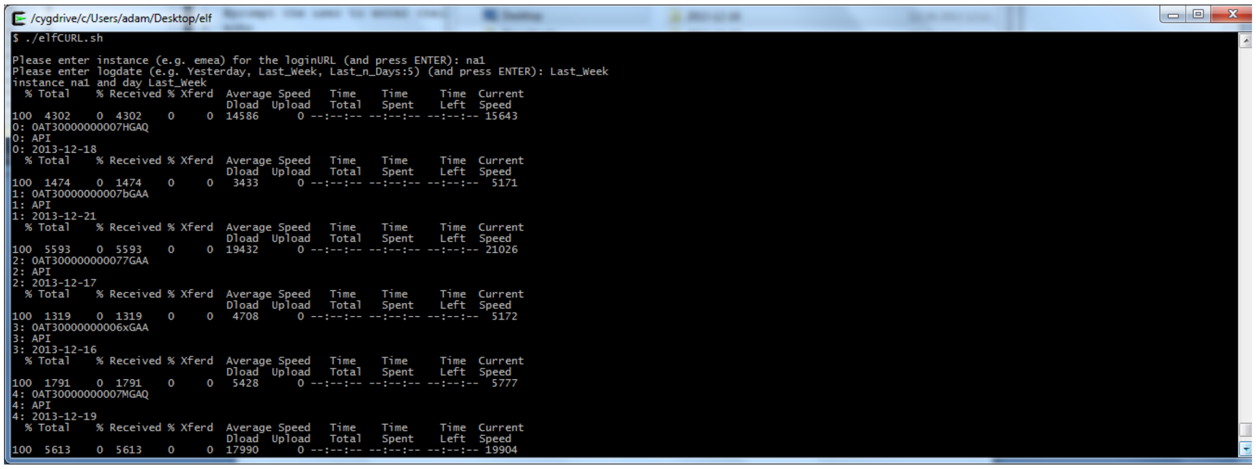
```
the array of Ids, EventTypes, and LogDates
    echo "${i}: ${ids[$i]}"
    echo "${i}: ${eventTypes[$i]}"
    echo "${i}: ${logDates[$i]}"

    #make directory to store the files by date
    mkdir "${logDates[$i]}"

    #download files into the logDate directory
    curl
"https://na1.salesforce.com/services/data/v29.0/sobjects/
EventLogFile/${ids[$i]}/LogFile" -H 'Authorization:
Bearer {ACCESSTOKEN}' -H "X-PrettyPrint:1" -o
"${logDates[$i]}/${eventTypes[$i]}.csv"
done
```

4. Open Cygwin (e.g. Start > Cygwin Terminal) and navigate to your newly created directory where your script is stored (e.g. **`cd /cygdrive/c/Users/adam/Desktop/elf`**).



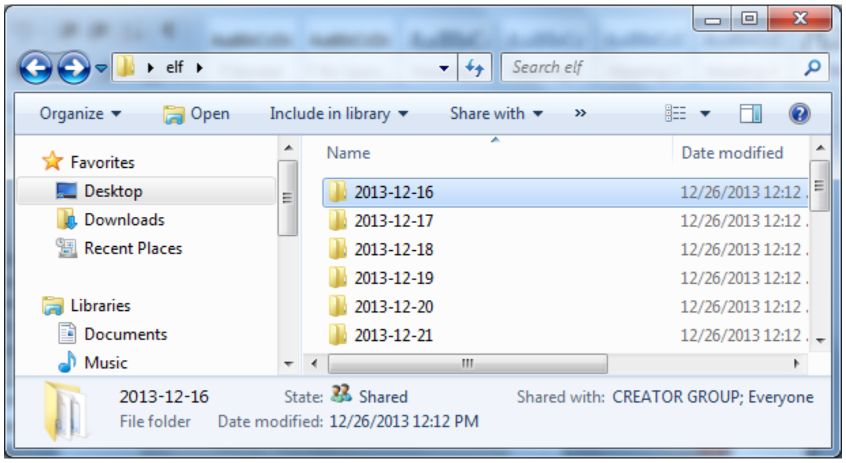
```
/cygdrive/c/Users/adam/Desktop/elf
$ cd /cygdrive/c/Users/adam/Desktop/elf
```

5. Run your bash script by typing **`./elfCURL.sh`** - you will be prompted to enter your instance (e.g. emea) and the date range you want (e.g. Yesterday, Last_Week, Last_n_Days:5, etc…)



```
/cygdrive/c/Users/adam/Desktop/elf
$ ./elfCURL.sh

Please enter instance (e.g. emea) for the loginURL (and press ENTER): na1
Please enter logdate (e.g. Yesterday, Last_Week, Last_n_Days:5) (and press ENTER): Last_Week
instance na1 and day Last_Week
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  4302    0  4302    0     0  14586      0 --:--:-- --:--:-- --:--:-- 15643
0: 0AT30000000007HGAQ
0: API
0: 2013-12-18
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1474    0  1474    0     0   3433      0 --:--:-- --:--:-- --:--:--  5171
1: 0AT30000000007bGAA
1: API
1: 2013-12-21
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  5593    0  5593    0     0  19432      0 --:--:-- --:--:-- --:--:-- 21026
2: 0AT30000000077GAA
2: API
2: 2013-12-17
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1319    0  1319    0     0   4708      0 --:--:-- --:--:-- --:--:--  5172
3: 0AT30000000006xGAA
3: API
3: 2013-12-16
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1791    0  1791    0     0   5428      0 --:--:-- --:--:-- --:--:--  5777
4: 0AT30000000007MGAQ
4: API
4: 2013-12-19
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  5613    0  5613    0     0  17990      0 --:--:-- --:--:-- --:--:-- 19904
```

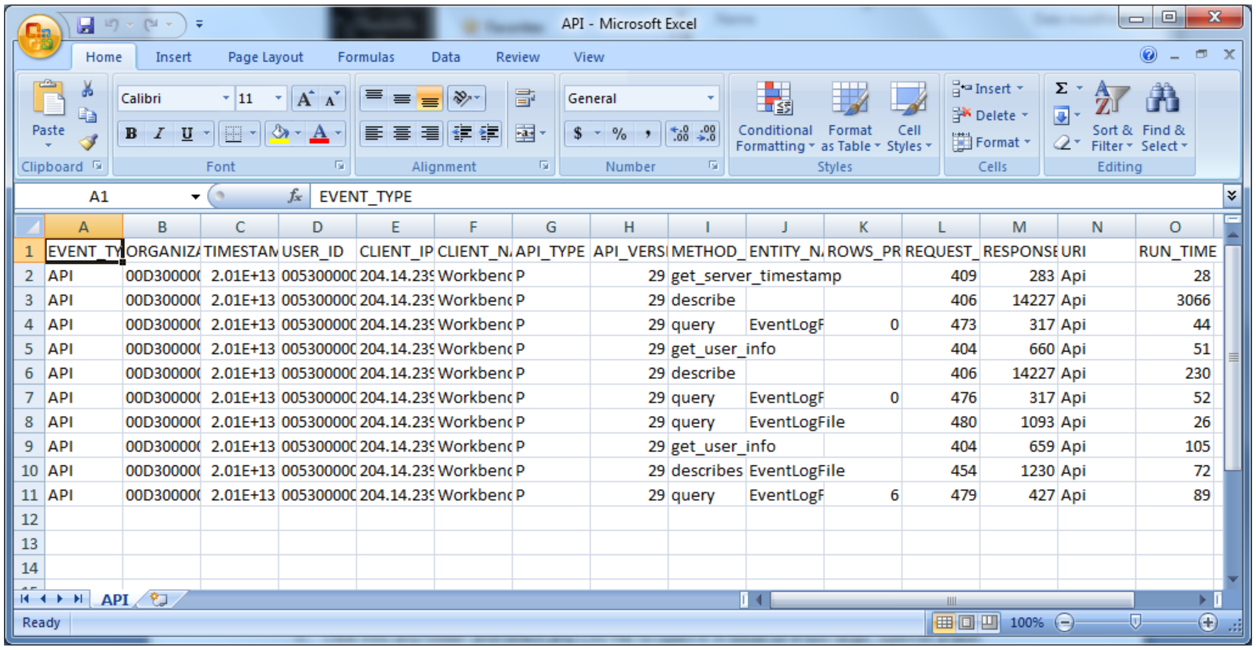
6. Check your newly created directory for a series of folders with dates



Name	Date modified
2013-12-16	12/26/2013 12:12
2013-12-17	12/26/2013 12:12
2013-12-18	12/26/2013 12:12
2013-12-19	12/26/2013 12:12
2013-12-20	12/26/2013 12:12
2013-12-21	12/26/2013 12:12

2013-12-16 State: Shared Shared with: CREATOR GROUP; Everyone
File folder Date modified: 12/26/2013 12:12 PM

7. Click into any folder and select any CSV file to open it in Excel or if too large, open in a text editor like notepad



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	EVENT_TY	ORGANIZA	TIMESTAM	USER_ID	CLIENT_IP	CLIENT_N	API_TYPE	API_VERS	METHOD_	ENTITY_N	ROWS_PR	REQUEST_	RESPONSE	URI	RUN_TIME
2	API	00D30000	2.01E+13	00530000	204.14.23	Workbenc	P	29	get_server_timestamp			409	283	Api	28
3	API	00D30000	2.01E+13	00530000	204.14.23	Workbenc	P	29	describe			406	14227	Api	3066
4	API	00D30000	2.01E+13	00530000	204.14.23	Workbenc	P	29	query	EventLogF	0	473	317	Api	44
5	API	00D30000	2.01E+13	00530000	204.14.23	Workbenc	P	29	get_user_info			404	660	Api	51
6	API	00D30000	2.01E+13	00530000	204.14.23	Workbenc	P	29	describe			406	14227	Api	230
7	API	00D30000	2.01E+13	00530000	204.14.23	Workbenc	P	29	query	EventLogF	0	476	317	Api	52
8	API	00D30000	2.01E+13	00530000	204.14.23	Workbenc	P	29	query	EventLogFile		480	1093	Api	26
9	API	00D30000	2.01E+13	00530000	204.14.23	Workbenc	P	29	get_user_info			404	659	Api	105
10	API	00D30000	2.01E+13	00530000	204.14.23	Workbenc	P	29	describes	EventLogFile		454	1230	Api	72
11	API	00D30000	2.01E+13	00530000	204.14.23	Workbenc	P	29	query	EventLogF	6	479	427	Api	89
12															
13															
14															