



Analytics Canvas

Command Line operation Documentation

Analytics Canvas V3.0.0
March 2018

1. Installation and licensing

Installing Analytics Canvas and preparing for batch file operation

- 1) Signup for a trial version of Analytics Canvas, or purchase an Automation license, and a suitable user/company license at www.analyticscanvas.com
- 2) Ensure that you are running the most recent version of Analytics Canvas, and that you have both Automation feature licensed. Not all features are available for all subscriptions, if you do not have all features some capabilities will not be available.
- 3) Trial keys are by default set to allow command line automation for evaluation purposes., When purchasing a subscription, command line capability is licensed separately from the single user, company and agency plans, but can be used with all the accounts licensed under these plans. The automation subscription is on a per instance (machine) basis. If you are already an Analytics Canvas customer, we can grant a special temporary trial key, contact us at support@analyticscanvas.com for more information.
- 4) You can validate that the command line functionality is present- if when you run Analytics Canvas, the text bar at the top of the application includes “Automation” or “Premium Automation”
- 5) Authorize Analytics Canvas to connect to your Google Analytics data by running Analytics Canvas.

2. Creating and Running queries

1. Create a Canvas

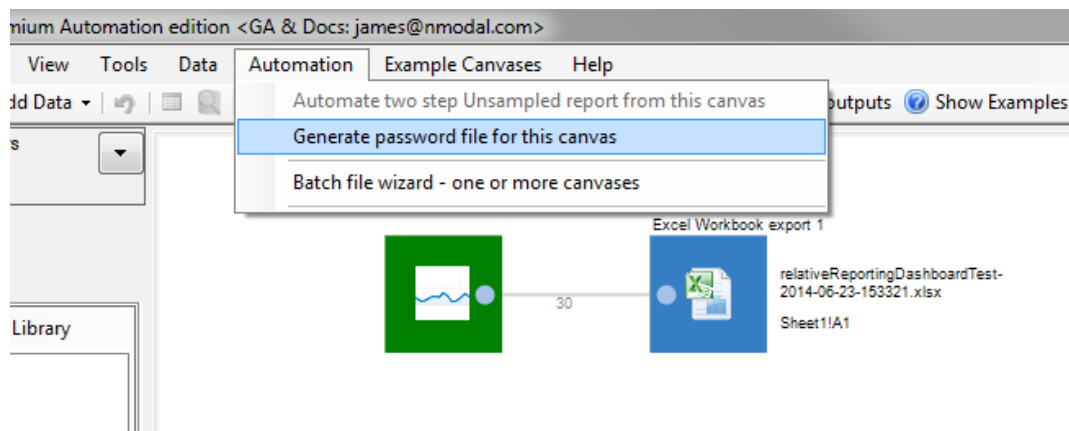
The first step in running a canvas in an automated way is to create a canvas, and test it in the desktop tool manually until the desired data transformation is done.

To learn more about how to create a canvas, query google analytics, and define data transforms etc, please refer to the help documentation within analytics canvas, the online documentation and the tutorial videos at www.analyticscanvas.com/support

2. Generate a password XML File for the Canvas

Because Analytics Canvas does not store passwords, or the Google Analytics tokens needed to connect to Google Analytics within the .ACC file for security reasons, it is necessary to generate an XML file that contains those passwords for use by the application when run in Batch mode.

To generate the password file, open the Analytics Canvas *.ACC file that you wish to automate, select the "Tools" menu, and select "Export Password File" (see screen shot).



You will be asked to confirm that you wish to export the passwords to the xml file. Save this password file, as it is required to run the canvas from the command line.

The other option available is the Batch File Wizard, which allows you to create not just the password file, but also the batch file that can be used to run one or more canvases. Since this batch file can then be edited and modified, this is often the best way to start.

3. Running from the command line

To do this you use the "RUN" command when running the Analytics Canvas executable from the command line along with the following command line parameters:

/f The full path and name of the Analytics Canvas ACC file. (required)

/p The full path and name of the Analytics Canvas password file. (required for GA queries)

/l The full path to the folder where you want the log file to be written. (required)

/logdaily The prefix name for the log file to be created daily. When this parameter is used, the log file is appended to by each run of a canvas, and the name of the log file is "prefixyyyyMMdd". By using the same prefix with multiple canvases sharing the same log folder then all logging can be consolidated into a single file per day.

/startdate The start date for the master date range in the format yyyy-MM-dd

/enddate The end date for the master date range in the format yyyy-MM-dd

/gamcache The full file and path for an xml file that provides the GA management API data- this avoids having to make a management api call at run- this will speed up automation tasks where canvases will be run hundreds of times.

/clearcache A flag to determine if data snapshot cache created by the canvas run should be cleared or not. If not set then the cache will be cleared of the files created by the canvas if the canvas runs successfully, and will not be cleared in the case of an error- This is the recommended setting (i.e. do not use this parameter). Otherwise, if this flag is set, then in all cases the cache is cleared or not based on the value (true or false)- if set to false then it is up to the user to monitor cache size, and remove files as needed. By running the client on the automation machine, the client will check to see if the cache has exceeded the limit set in options, and will remove older files- if you set /clearcache false then it is recommended that you open the desktop periodically to have it do this cleanup.

/usegaunsampledcache Set this to false to force a new request to the unsampled reporting API- otherwise if a previously downloaded query is identical the previously downloaded data set will be used.

/bigqueryoutname the name of the big query table to be exported to- this parameter only works with canvases that have a single BigQuery Export block. Useful to have a canvas generate multiple tables with a date suffix, for example, one table for each day for each data set.

Email parameters

Email can be sent by specifying the following email parameters, when they are set then an email will be sent including the log file contents if the Canvas run fails. Canvas can also be used to simply send an email- to do this instead of the RUN command directly after the analyticscanvas.exe use the EMAIL command. For example:

```
%exe% EMAIL /emailhost %host%:%port% /emailto "" /emailfrom %emailfrom%  
/emailfromname %emailfromname% /emailpwd %emailpwd% /emailsubject "Batch ran  
successfully" /emailbody "1 canvases ran successfully."
```

/emailhost host:port the host and port for the email- for example, smtp.gmail.com:587 would be the value for gmail

/emailto emailto@example.com the email to which the log should be sent

/emailfrom emailfrom@example.com the email account sending the email

/emailfromname "Email From Name" a label for the sending email- this can be something that identifies the job

/emailpwd emailpwd the password for the email account in emailfrom

/emailssl false (Optional) A flag that determines if SSL is enabled, the default is TRUE if this parameter is not included.

/emailincludemachine true A flag that is used when using the EMAIL switch only- if set to true will include the line "Sent by Machine: x" where x is the machine name at the beginning of the email body. Optional, if not included default is false.

Database Logging Parameters

/projectrunuid defines the unique identifier for the project- this should be a name plus a date time stamp ideally

/projectlabel defines a text label for the project- this does not need to be unique it can tie together multiple runs. For example, if you had a project "MonthlyRevenueReport003" then this would be the project label, and the projectrunuid might be "MonthlyRevenueReport003_20170304_230001" the exact time of a specific run of the project.

/dblog points to the xml file that defines the Database logging location and setup it must be used both with the INITPROJECTRUN command and with all canvas run commands to specify the SQL Server data base to log to.

Running Canvases connected to a project

After a project has been created using the INITPROJECTRUN command, when canvases are run the following parameters are used if the canvas should be included in the database log for the project:

/projectrunuid and **/dblog** to define the project run UID and the database

Time Zones and Server Time vs UTC vs Canvas Time Zone

When a Canvas is created, it has a specific time zone associated with it. This is the time zone of the machine on which it was created, but can be modified.

When a Canvas is run in desktop mode, it takes on the time zone of the machine it is running on. Functions like TODAY() and NOW() will return the current date and time of the machine.

However, when running canvases from multiple users in multiple time zones on an Analytics Canvas server, it may be desired to have canvases keep the local time zone of the Canvas creator rather than the

timezone of the server. In fact, we recommend if there are multiple time zones involved that the server should be set to UTC, and all logging should be in UTC.

There are parameters that can be used on the command line to control behavior in terms of dates and times.

/utc true or false This parameter is set to false by default if omitted. This command makes all date times in the database log tables set to UTC. This is useful if multiple servers are being used, in general UTC is best as it avoids issues with daylight savings time and timezone changes.

Advanced Parameters:

These parameter require the Premium subscription

/gaviewid A google analytics view ID- every query in the canvas will be pointed at this view- this cannot be used with canvases that query multiple profiles in a single import block.

/reportingdate The date to be used for relative dates in the master date range. For example, if the canvas master date range is set to "Previous 3 months" and the reporting date is set to "2014-01-01" then the canvas would use a date range of Oct 1 2013 to Dec 31 2013 (The previous three months before January 1 2014- this would be true for any date in January- if the reporting date was set to a date in February 2014, then the range would be Nov 1 2013 to Jan 31 2014 and so on.)

/txtoutname The name of the text file to be generated- this parameter only works with canvases that generate a single text file as an output.

/xlsxoutname The root name of the Excel report to be generated- this parameter only works with canvases that generate a single Excel workbook as an output.

/tdeoutname The full file and path that will be used for a single TDE Export block within the canvas. This parameter only works when there is one and only one TDE export.

/tdepubname1 TDEexportBlockName:publishName or TDEexportBlockName:projectName:publishName Specifies the name of a given TDE file (or the name and the project) in a given TDE publish point. This allows a single template to be published multiple times with different data sets and in different projects by also using a user parameter in filters or SQL queries to limit the data set included in the TDE. Names cannot contain the colon character (":") as it is used to separate the workbook name and the publish name in the parameter value, workbook names are not case sensitive. To specify multiple twbx publish names for multiple TWBX files use /tdepubname2, /tdepubname3 etc. To use this parameter, any specified tde files must be published to only one location within the canvas. The publish name is also the name used for file TDE file generation. Be sure to enclose the entire argument in quotes ("TDEexportBlockName:projectName:publishName") if there are any spaces, and do not use any characters that are not allowed in Tableau server/online in the publish Name (for example &).

/twbxoutname The root name of the TWBX workbook to be generated. This parameter only works when there is one and only one TWBX export.

/twbxpubname1 workbookName:publishName or workbookName:projectName:publishName Specifies the name of a given workbook (or the name and the project) in a given TWBX publish point- this allows a single template to be published multiple times with different data sets and in different projects by also using a user parameter in filters or SQL queries to limit the data set included in the TWBX. Names cannot contain the colon character (":") as it is used to separate the workbook name and the publish

name in the parameter value, workbook names are not case sensitive. To specify multiple twbx publish names for multiple TWBX files use /twbxpubname2, /twbxpubname3 etc. To use this parameter, any specified twbx files must be published to only one location within the canvas. Up to 10 unique twbx publish names can be specified. For example, if there is a workbook Workbook1 in the canvas that publishes to WorkbookDefault in the publish point by specifying the parameter /twbxpubname1 Workbook1:NewWorkbook it would publish a workbook with the name NewWorkbook into the same project. By using /twbxpubname1 Workbook1:newProject:newWorkbook it would publish a workbook with the name newWorkbook into a different project called newProject. Be sure to enclose the entire argument in quotes if there are any spaces, and do not use any characters that are not allowed in Tableau server/online in the publish Name (for example &).

/forcepartitiondays An integer value 1 or higher that forces the partition period for all partitioned query to a set value- this can be used if different Views require different levels of partitioning. If one view has very large traffic and requires partitioning at a daily level, then set this value to 1 when running the partitioned query canvas for that view, if another view needs weekly partitioning then set the value to 7, if other views do not require partitioning, then set this value to a large number (in excess of the number of days in the date range) as a result there will be only one partition. This results in a much faster execution time as each canvas run only partitions as needed to eliminate sampling.

User Defined parameters

Within Analytics Canvas the dollar sign denotes a parameter value. All analytics canvas reserved values start with \$AC_- For example, to access the current master date range start date, the reserved parameter is \$AC_STARTDATE and the end date is \$AC_ENDDATE.

Users can define any string parameter value they want and use it throughout Analytics Canvas, both in expressions and within SQL statements.

If a parameter is referenced, but is not set, then it will have a value of NULL.

For example, in a filter block you might want to be able to filter on rows based on a value passed in from the command line. In the filter block, use the following expression:

```
If(isnull($filterCondition),TRUE,[Original.ColumnToFilter]=$filterCondition)
```

This statement will return TRUE if the parameter is not set, and will compare the value of the parameter to the column "ColumnToFilter" if it does have a value, resulting in a filter.

To set the filter value in the .bat file, simply add the switch:

```
/ $filterCondition "value desired"
```

Now, as the canvas runs, the filter block will pass any rows where the column ColumnToFilter contains the value "value desired"

NOTE: parameter names are NOT case sensitive- i.e. \$MyParam is considered the same as \$myparam and \$MYparam etc.

Example of how to run this from a batch file, make a batch file containing the following (need to change the paths of the ACC file, XML password file and log file location to folders that exist on your machine):

```
"C:\Program Files\nModal Solutions Inc\Analytics Canvas\AnalyticsCanvas.exe" run /f  
C:\ACBatch\TestACBatch.ACC /p c:\ACBatch\GATestPWD.xml /l C:\ACBatch\Logs\
```

Create a text file containing the run command, and then rename the file .BAT. This file can now be run by any scheduler.

To schedule Analytics Canvas to run automatically, any scheduling software, or the windows scheduler can be used.

For basic scheduling using the windows scheduler:

<http://windows.microsoft.com/en-US/windows7/schedule-a-task>

Remember that it may take many minutes for a canvas to run, depending on the types of queries done.

Log Files:

After running analytics canvas from the command line, providing a valid log file path was provided in the command using the “/L” command, a log file will be generated in that folder specified.

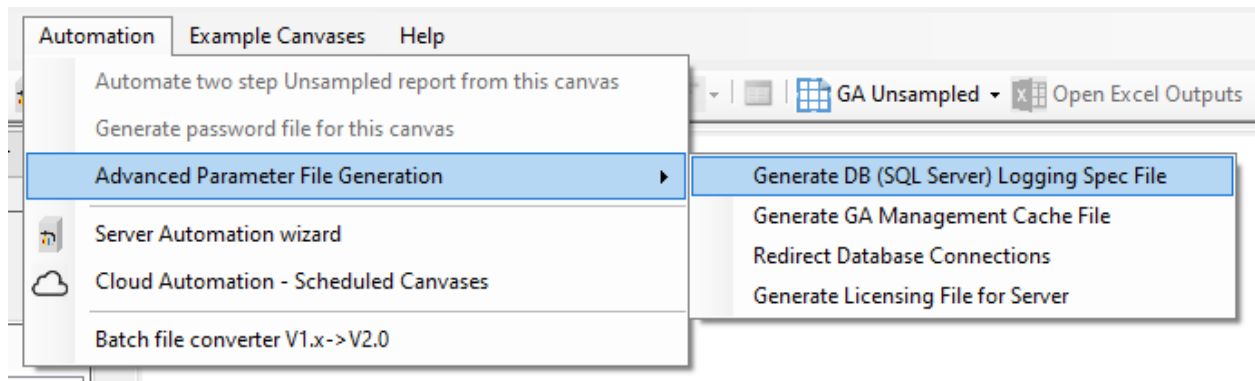
Logging to a SQL Server database

It is possible to configure automatically run canvases to write log information into an sql server database.

In this case, canvases are grouped together into “Projects” and every project has a declared ID and label.

Configuring the database file

To specify which SQL Server logs should be written to, open Analytics Canvas, and go to Automation > Advanced Parameter File Generation > Generate DB (SQL Server) Logging Spec File.



This defines where the table will be created.

Projects are first created using INITPROJECTRUN

For example the following command creates a new project and logs it to the database specified;

```
"C:\Program Files (x86)\nModal Solutions Inc\Analytics Canvas\AnalyticsCanvas.exe"  
INITPROJECTRUN /dblog "C:\example\logSpec.xml" /projectrunuid "%prjRunUID%"  
/projectlabel "This is a run of My Project"
```

Running Canvases connected to a project

After a project has been created using the INITPROJECTRUN command, when canvases are run the following parameters are used if the canvas should be included in the database log for the project:

/projectrunuid and **/dblog** to define the project run UID and the database

/utc true This command makes all date times in the logs set to coordinated universal time, false or omitted means that date times will be the local machine time. This is useful if multiple servers are being used, in general UTC is best as it avoids issues with daylight savings time and timezone changes.

For example- a complete run of 3 canvases, all grouped together to a single projectrunuid would look like this:

```
set exe="C:\Program Files (x86)\nModal Solutions Inc\Analytics Canvas\AnalyticsCanvas.exe"
```

```
For /f "tokens=2-4 delims=/ " %%a in ('date /t') do (set mydate=%%c-%%a-%%b)
```

```
For /f "tokens=1-2 delims=/: " %%a in ('time /t') do (set mytime=%%a%%b)
```

```
set prjRunUID="MyProjectName_%mydate%_%mytime%"
```

```
%exe% INITPROJECTRUN /dblog "C:\example\logSpec.xml" /projectrunuid "%prjRunUID%"  
/projectlabel "This is a run of My Project"
```

```
%exe% RUN /f "%~dp0CanvasFile001.ACC" /dblog "%~dp0logSpec.xml" /projectrunuid  
"%prjRunUID%" /p "%~dp0CanvasFile001.ACC-pwd.xml" /utc true
```

```
%exe% RUN /f "%~dp0CanvasFile002.ACC" /dblog "%~dp0logSpec.xml" /projectrunuid  
"%prjRunUID%" /p "%~dp0CanvasFile002.ACC-pwd.xml" /utc true
```

```
%exe% RUN /f "%~dp0CanvasFile003.ACC" /dblog "%~dp0logSpec.xml" /projectrunuid  
"%prjRunUID%" /p "%~dp0CanvasFile003.ACC-pwd.xml" /utc true
```

Example Log File Contents

The following is the contents of an example log file for a canvas that extracted data from the Google Analytics API, and wrote about 10,000 rows into a SQL Server database table;

Contacting licensing server:6/14/2012 10:22:54 PM

License validated:6/14/2012 10:22:55 PM

Analytics Canvas V1.3.0 6/14/2012 10:22:55 PM

Google analytics start date:2011-10-01

Google analytics end date:2011-12-31

Loading Password file:C:\ACBatch\NewAutoExample-pwd.xml 6/14/2012 10:22:55 PM

Loading:C:\ACBatch\NewAutoExample.ACC 6/14/2012 10:22:56 PM

Accessed Google Analytics Management API V3.0:6/14/2012 10:22:59 PM

Loaded:6/14/2012 10:22:59 PM

Init Blocks: 6/14/2012 10:22:59 PM

Run Canvas: 6/14/2012 10:22:59 PM

Cache folder set:C:\Users\nmodal\Documents\My Analytics Canvas\DMCache\

Run Duration: 00:00:01.5040860

No block errors or warnings.

Block [AnalyticsDB SalesAnalyticsTable] wrote 768 rows to DB:.\DMSQLServer\
Table:SalesAnalyticsTable

4. Running multiple canvases from a single batch file and handling errors

Often, when automating sophisticated analysis and extraction, it is necessary to run multiple Analytics Canvas canvases in order, and to validate if the canvas ran successfully, or if the process should be aborted.

The following is an example of a batch file that runs three canvases, the first two that could perform multiple extractions from the Google Analytics API, and put the data into an SQL Server database, and the third, providing the first two canvases were successful, that generates an Excel dashboard.

To determine if an Analytics Canvas run was successful, the parameter %ERRORLEVEL% is checked- if the run was successful, the value of zero is returned- if failed, an error value greater than zero is returned. If we see a value greater than zero, we jump to the end, aborting the run.

```
set dmexe="C:\Program Files\nModal Solutions Inc\Analytics Canvas\AnalyticsCanvas.exe"

set dmc="C:\FolderContainingACCFilesPWDFilesAndLogsFolder"

%dmexe% RUN /f %dmc%\001-Load-GA1.ACC /l %dmc%\logs\ /p %dmc%\001-pwd.xml

if %ERRORLEVEL% GTR 0 goto Error

%dmexe% RUN /f %dmc%\002-Load-GA2.ACC /l %dmc%\logs\ /p %dmc%\002-pwd.xml

if %ERRORLEVEL% GTR 0 goto Error

%dmexe% RUN /f %dmc%\003-Generate-XLSX.ACC /l %dmc%\logs\ /p %dmc%\003-pwd.xml

if %ERRORLEVEL% GTR 0 goto Error

goto End

:Error

echo Error in run

:End
```

5. Additional Switches

EMAIL - used as a switch instead of RUN, this will cause Analytics canvas to simply send and email. For example:

```
%exe% EMAIL /emailhost %host%:%port% /emailto "" /emailfrom %emailfrom%  
/emailfromname %emailfromname% /emailpwd %emailpwd% /emailsubject "Batch ran  
successfully" /emailbody "1 canvases ran succesfully."
```

To include the machine name in the email body add the /emailincludemachine true parameter.

6. More advanced Examples

Running a single template many times for different GA Views

With the parameterization option it is possible to run a single ACC file multiple times, each time querying from a different Google Analytics View and writing to a new Excel Dashboard. This means that one template can be reused again and again, generating dozens or hundreds of dashboards, one for each website or filter represented by the Google Analytics view.

To do this, you can use **/gaviewid** to specify the GA view, and **/xlsxoutname** to specify the root of the Excel output file name, and if you are using relative dates (Such as "last 12 months" etc.) then you can use **/reportingdate** to change the reporting time. If you do not include **/reportingdate** then the reporting date used is the current day. (So if you run the report in March, "last month" is February, but if you specify **/reportingdate** to be June of 2012, then "last month" is now May of 2012- it makes the reports relative dates be set as if the report was run on the date you provide with **/reportingdate**.

For example, to run a template for three different websites with GA view ids of 123456, 654321 and 123654 using the current day as the relative dates starting point the batch file would look like this:

```
set exe="C:\Program Files (x86)\nModal Solutions Inc\Analytics  
Canvas\AnalyticsCanvas.exe"
```

```
%exe% RUN /f "%~dp0TestParametersDashboard.ACC" /l "%~dp0logs" /p  
"%~dp0TestParametersDashboard.ACC-pwd.xml" /gaviewid 123456 /xlsxoutname  
"Report 1"  
%exe% RUN /f "%~dp0TestParametersDashboard.ACC" /l "%~dp0logs" /p  
"%~dp0TestParametersDashboard.ACC-pwd.xml" /gaviewid 654321 /xlsxoutname  
"Report 2"  
%exe% RUN /f "%~dp0TestParametersDashboard.ACC" /l "%~dp0logs" /p  
"%~dp0TestParametersDashboard.ACC-pwd.xml" /gaviewid 123654 /xlsxoutname  
"Report 3"
```

```
:End  
EXIT /B
```

For a full example of how to do the above, please continue to the detailed tutorial.

Tutorial- Running a single canvas many times for different GA Views and/or Date ranges

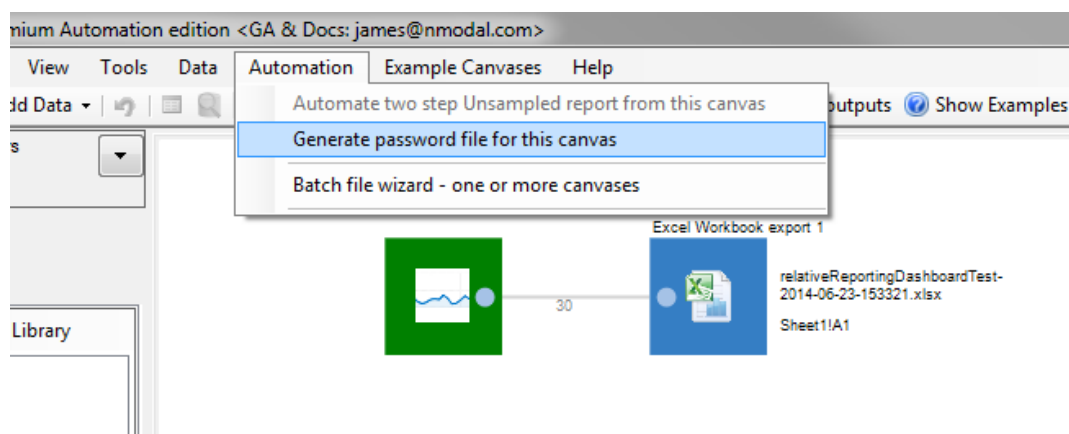
First, you need to create a canvas that is the “Master Canvas” - this is the canvas that contains all your queries, and transformations and generates the excel dashboard.

To be able to run a canvas as a master canvas multiple times it must :

- 1) Only query one Google Analytics View. When running with parameters, every GA import block will be pointed to the single View specified by the parameter.
- 2) Write to one, and only one Excel Dashboard file. When running with parameters, the excel output file name root can be specified, but there must be only one Excel output workbook.
- 3) The canvas must be generic enough so that it will work with all the Google Analytics Views you want to use with it. For example, if one of the Views has Goal 1 as the “purchase” goal, but another has Goal 2, and other Goal 6, then when the dashboard is run, it will not get the purchase goal in all cases, since the master canvas must specify one and only one goal number. This is also true for site specific things such as page paths, event categories etc. It is possible to build a template that adapts to this by using reference data keyed with the GA view ID, but this is beyond the scope of this first tutorial.

So the first step is to build a canvas (using a single GA Profile, and specifying an excel output file name). Test this dashboard and confirm that everything works as desired. Once this is done, you need to generate a batch file that will allow you to run this canvas multiple times, each time with a different GA View ID and a different output file name.

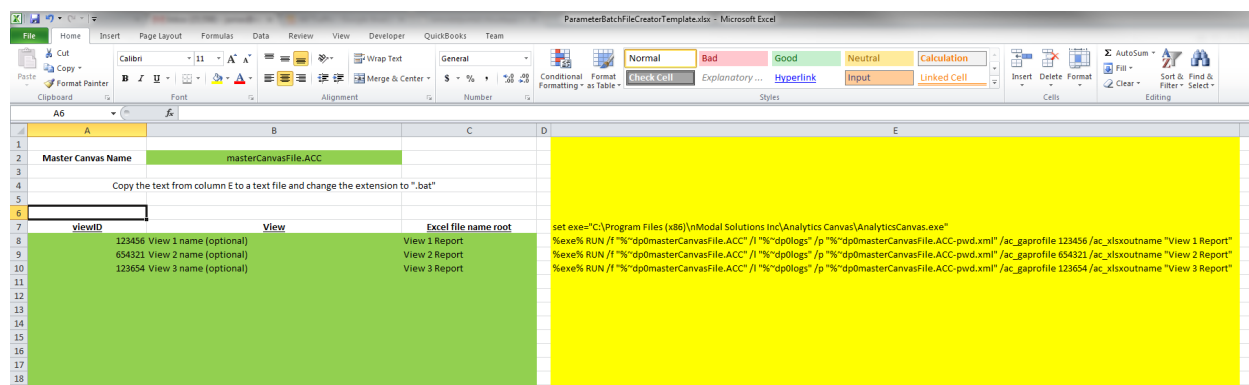
Once the master canvas is ready, you need to generate a password file for it. Do this in the Automation menu;



Leave the default name, and this must match the name of the canvas. If you rename your canvas, then you need to generate a new password file (or rename the old password file manually). It is best to set everything up so that the master ACC file, the password file and the Excel Template used by the master ACC file are all located in the same folder.

Now, make a copy of the excel workbook "ParameterBatchFileCreatorTemplate.xlsx"

Open this copy, you will see the following fields- we've filled them in to show how to run an ACC file for three Google views.



In B2, enter the file name (without the folder) of the master ACC file.

In the rows starting at A8, enter in the View ID, view name then the root of the file name that you want the output file for that profile to have.

To the right, in the yellow area, you will see as you complete these rows, another line is added to the batch file.

When you have all the profiles you want added, select the filled in rows in column E and copy them into a new text file using notepad.

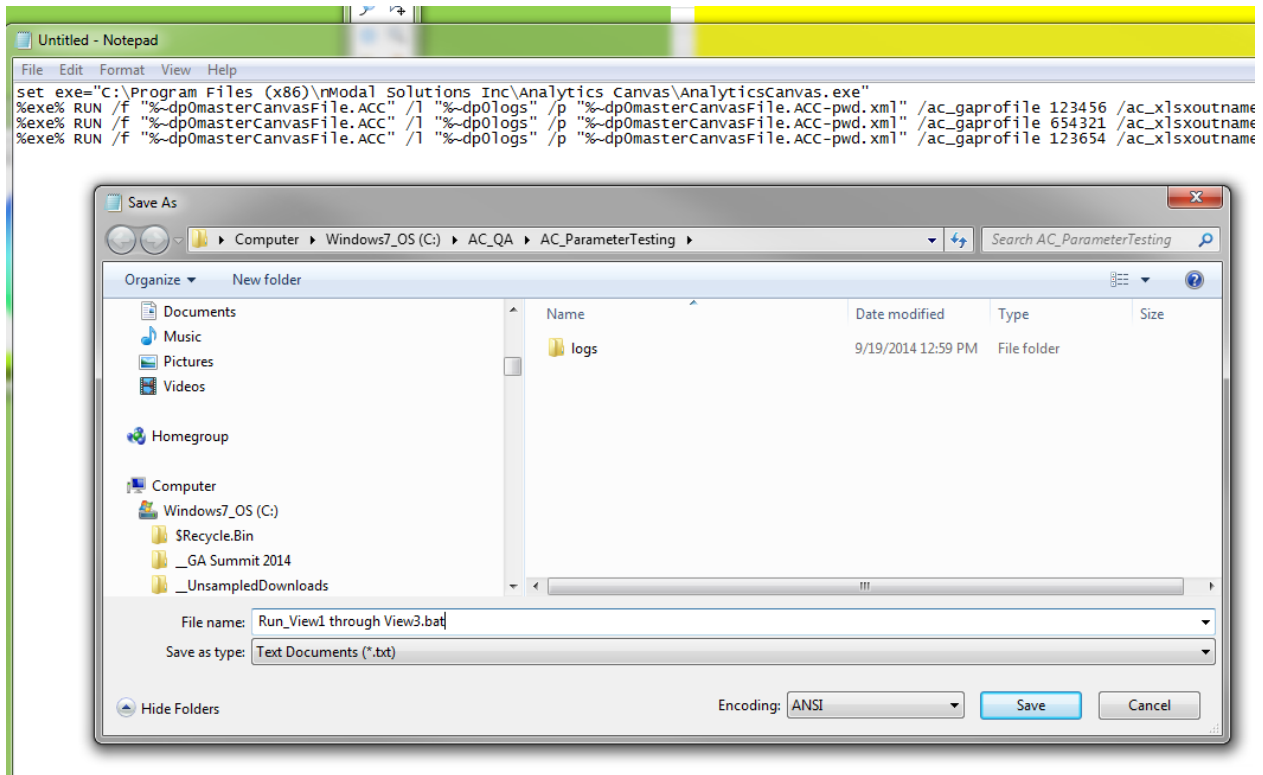
The screenshot shows an Excel spreadsheet with the following data:

viewID	View	Excel file name root
123456	View 1 name (optional)	View 1 Report
654321	View 2 name (optional)	View 2 Report
123654	View 3 name (optional)	View 3 Report

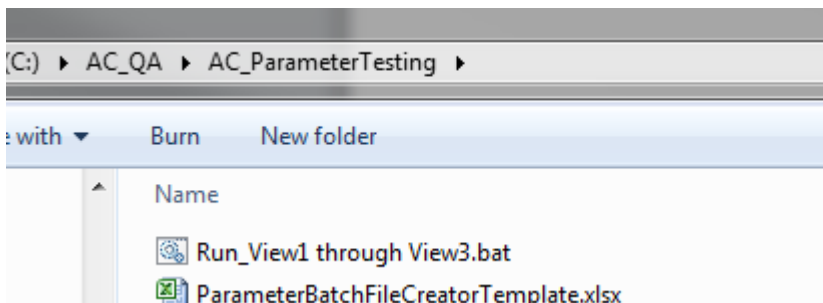
The spreadsheet also has a 'Master Canvas Name' field set to 'masterCanvasFile.ACC' and a formula bar showing a path to 'AnalyticsCanvas.exe'. A yellow area on the right indicates where the batch file content is being generated. Below the spreadsheet, a Notepad window shows the resulting batch file content:

```
set exe="C:\Program Files (x86)\nModal Solutions Inc\Analytics Canvas\AnalyticsCanvas.exe"
%exe% RUN /f "%~dp0masterCanvasFile.ACC" /i "%~dp0logs" /p "%~dp0masterCanvasFile.ACC-pwd.xml" /ac_gapofile
%exe% RUN /f "%~dp0masterCanvasFile.ACC" /i "%~dp0logs" /p "%~dp0masterCanvasFile.ACC-pwd.xml" /ac_gapofile
%exe% RUN /f "%~dp0masterCanvasFile.ACC" /i "%~dp0logs" /p "%~dp0masterCanvasFile.ACC-pwd.xml" /ac_gapofile
```

Save this file in the same folder as the master canvas and the password file, and name it so that it will have the .bat extension. It must be in the same folder, since it is designed to look there for the canvas

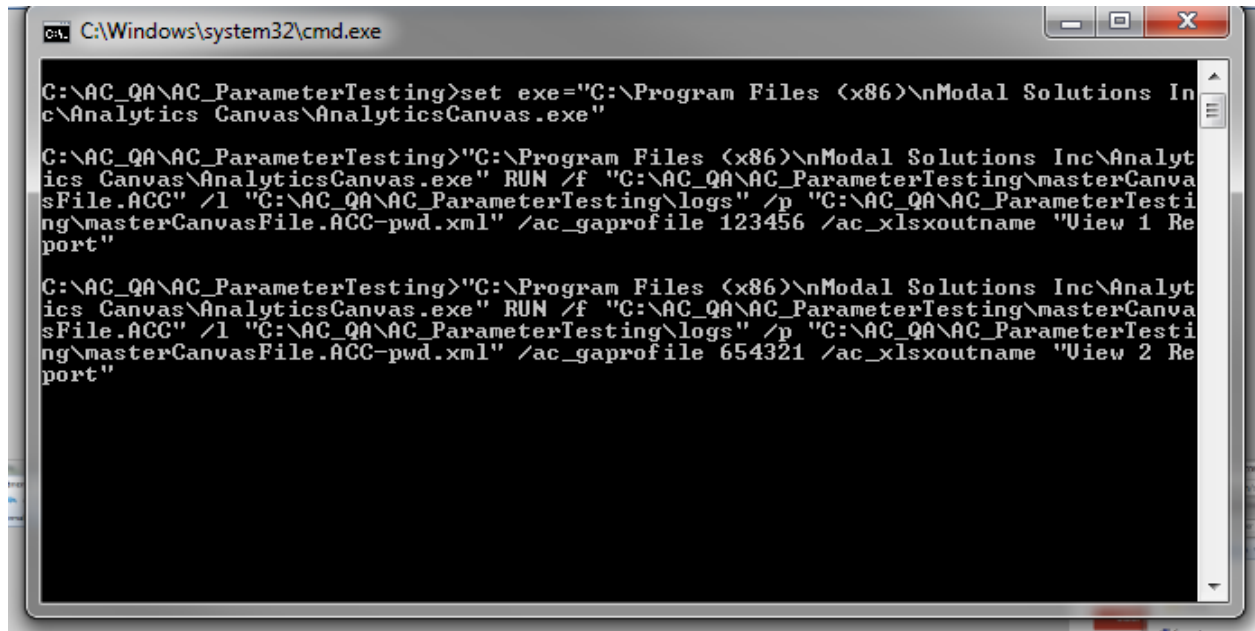


A batch file will have the “Gears” icons. If you double click on this file, it will run the canvas as per your configuration in the excel template.



While it is running, you will see each command in a command window.

It is important to NOT have Analytics Canvas open when you run the batch file- it is not possible to run two instances of Analytics Canvas, and so if you have it open already the batch run will fail. Each run will try to complete, but no excel files will be generated- in the log file you will see an error message.



```
C:\Windows\system32\cmd.exe

C:\AC_QA\AC_ParameterTesting>set exe="C:\Program Files (x86)\nModal Solutions Inc\Analytics Canvas\AnalyticsCanvas.exe"

C:\AC_QA\AC_ParameterTesting>"C:\Program Files (x86)\nModal Solutions Inc\Analytics Canvas\AnalyticsCanvas.exe" RUN /f "C:\AC_QA\AC_ParameterTesting\masterCanvasFile.ACC" /l "C:\AC_QA\AC_ParameterTesting\logs" /p "C:\AC_QA\AC_ParameterTesting\masterCanvasFile.ACC-pwd.xml" /ac_gaprofile 123456 /ac_xlsxoutname "View 1 Report"

C:\AC_QA\AC_ParameterTesting>"C:\Program Files (x86)\nModal Solutions Inc\Analytics Canvas\AnalyticsCanvas.exe" RUN /f "C:\AC_QA\AC_ParameterTesting\masterCanvasFile.ACC" /l "C:\AC_QA\AC_ParameterTesting\logs" /p "C:\AC_QA\AC_ParameterTesting\masterCanvasFile.ACC-pwd.xml" /ac_gaprofile 654321 /ac_xlsxoutname "View 2 Report"
```

You should see the Excel files being generated as each step runs if everything is working.

To see any error messages, look in the folder "logs" that will be created in the folder that holds all the files, in there you will find a log file for each run (or attempted run) these files will contain any error messages as outline in this document.