kegsoft®

Kegsoft KBL Briefing:
Big Data: Handling Very Large Datasets (VLGs) in KBL

Part 1 Importing large datasets into KBL Studio

Peter Domanski: Kegsoft Co-Founder

Introduction

First we will look at some general factors in dealing with Very Large Datasets (VLDs) and then at how the KBL File Command Set can be used to import and process very large data files (>1Gb) into KBL Studio.

The generic term 'Big data' has become an increasingly important topic and has spawned an IT sub-industry all of its own. Most us are familiar with data visualisations based on impressively large datasets including trends in national economics as well as mind-boggling statistics and representations for personal internet usage and ecommerce. The big internet search engines and providers harness their immense processing power to mine patterns of individual and collective buying behaviour for the benefit of commercial advertisers and other interested parties – regularly processing 100s of millions of transactions and related data to establish buying trends and propensities. Knowledge is power and even small scraps of insight can provide competitive advantage.

Governments and corporations increasingly collect vast amounts of information about demographics, goods and services – including shopping transactions, patterns of communication, inventory movements, census records etc. And, electronic devices can generate phenomenal amounts of digital information continuously for both scientific and commercial use. All in all the volumes of data collected are increasing rapidly.

From an IT perspective, 'Big Data' - very large amounts of data – presents challenges (and opportunities) concerning collection, storage, access, processing and presentation – Computational and storage ability is key ….but not the whole story. Big Data is Big business.

To most of us, dealing with VLDs might seem 'out of our league', requiring specialist techniques and considerable investment in skills and systems. But, there are low cost solutions available for more modest requirements and it will be no surprise that skilful users are now well able to do quite sophisticated data analysis and visualisation with the recent versions of Excel and other office-style tools. However, when data volumes rise much above a million records deep and just a few columns wide, you'll soon hit the buffers as most of our familiar office tools and even Notepad can't open much above a million rows and PC performance really takes a dive. But suppose you could deal with several Gigabytes of data containing a few million records on your laptop PC, would this be useful to you or your organisation?

Suppose you could deal with several Gigabytes of data and a few million records on your laptop?

Well, you can, and quite easily, using KBL, but first a bit more about Big Data. There are more issues to deal with than just high data volumes and size in itself, is not the only problem. As anyone used to regularly dealing with data knows, your data is very rarely straightforward or perfect – especially if it has anything at all to do with people or organisations. 95%+ of it may well be perfect but a persistent 5% is likely to be exceptional or just erroneous – e.g. incorrectly formed or recorded. And it's the 5% that puts 'a spanner in the works'. Errors and exceptions are not just a pain to deal with but they can cause problems disproportionate to their frequency. Let's look at some factors that can plague data in general and large datasets in particular:

Volume

The quantity of generated and stored data. The size of the data determines the value and potential insight you will derive. VLDs do require different physical techniques from small datasets.

Variety

The type and structure of the data. To properly analyse it with confidence, you will need a complete understanding of it in order to effectively process it and be assured of any insight you are looking for.

Velocity

In this context, the speed at which the data is generated and processed to meet the demands and challenges that lie in the path of growth and development. How fast is it coming in, is it accumulating slowly or increasing rapidly?

Variability

Inconsistency of the data set can hamper processes to handle and manage it – often referred to as happy and unhappy data. Data created via human hands – especially by end-users can vary tremendously – but so can data collected automatically by remotely instruments – subject, say, to environmental conditions.

Veracity

The quality of captured data can vary greatly, affecting the accurate analysis – can we even tell whether data is good, bad or ugly?

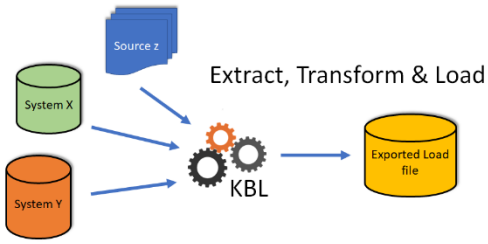
At Kegsoft, we understand that large datasets, whilst requiring some specific techniques to cope with large size and processing requirements, is not an issue in itself – you can scale storage and processing power. The real challenges are understanding the structure, quality, exceptions and compensations – and knowing how to deal with both the 'happy' AND the 'unhappy' data – identify it? Ignore it? Report it? Fix it? Reject it?

It's important to develop a strategy to deal with VLDs – to be clear about its meaning and content, what are the precise objectives are and how to get there in a logical fashion.

KBL can greatly extend your ability to deal with large amounts of data without need of expensive facilities or services. The extra KBL skills you need are easily learned – principally some straightforward techniques to turn big obstacles into smaller ones. If your data has complexity and / or imperfections and you need precision, you will need techniques to identify poor data and deal with exceptions, you need to know how to work around them and potentially report and or fix them.

KBL provides the tools to process VLDs – but it is up to you to understand your data and use the tools and your resources to the best effect.

ETL – Extract, Transform and Load



From time to time organisations need to upgrade their systems and / or merge data from dissimilar systems (perhaps as a result of a takeover or IT rationalisation) and there are a number of other scenarios when data needs to be extracted from one or more systems and loaded into another (e.g. refreshing a data warehouse) and as the ETL subtitle hints, the process can require several stages starting with data extraction followed by some form of clean-up and transformation before being reloaded into another system. KBL with its ability to import from large files perform complex business logic and data manipulation together with exporting to standard and bespoke formats – can provide a very effective and cost efficient way of processing bulk data.

Handling Large data volumes in KBL

The first stage has to be getting your data into the KBL Studio environment. KBL's standard IMPORT and EXPORT commands are commonly used for handling files (e.g. .csv, txt or xlsx) typically up to 0.5 million rows and processing this volume of data is usually conducted in much the same as you would for just a few hundreds of rows. But once volumes start increasing into the millions of records, smarter methods of processing are available – to cope physically with potentially long processing times and minimise machine memory requirements. For VLDs , different techniques and tactics are sometimes required. For example, for some types of data analysis, we may have many millions of source records in one or more files but perhaps only < 10% are really of interest to us in our investigation – let's say one UK region out of the UK's census data. If we were smart and able to confine our extract and analysis to the target region, we could significantly keep memory usage

down and minimise processing time – But we do need to pick out the useful data from the mass that we don't need.

Alternatively, supposing we had a database dump file of 10s of millions of records containing data from different tables. Could we distinguish between different tables and digest the data of interest in more manageable chunks? Must we use a sledgehammer or could we work smarter with a nutcracker?

The FILE Command Set in KBL

The KBL FILE command set is included in Kegsoft Studio Express and provides a convenient way of examining and processing bulk data, it functions separately from the normal IMPORT command which moves data directly into MAIN. Instead, FILE.IMPORT loads records in bulk, into a staging area where they can be viewed and packaged into manageable chunks before being transferred into MAIN working memory (for normal processing). In fact FILE.EXPORT can export records in chunks without even being loaded in MAIN. A further feature of this command set is that standard KBL script logic can be applied to data while in the File staging area – this gives you the ability to only import the data that meets your criteria into MAIN. These facilities greatly enhance your ability to work with VLDs and keep your memory and processing requirements to within acceptable limits.

Commands

File.set /Import Start 1

File.set /Import End 10000

File.Import <file specifications>

Assuming we have a file of many 100s of thousands to millions of records, the File.set command lets us decided the first and last row (number) to read in. Set commands are optional and If not specified, by default, File.Import would import all records into the staging area. **File.set /Import Start START** followed by **File.set /Import 20000** would read in rows 1-20,000

File.set /import Start 600001 followed by **File.set /Import End END** would read the 600,001st record to the end of the file.

As usual, the start and end numbers can be parameterised – so can be under script programme control e.g. **File.set /import Start [NEXTSTART]** followed by **File.set /Import [NEXTEND]**

Following the optional File.Set commands, File.Import will read the file as instructed (or not) E.g.

FILE.IMPORT /TEXTFILE ons:Postcode_Population_Data.csv

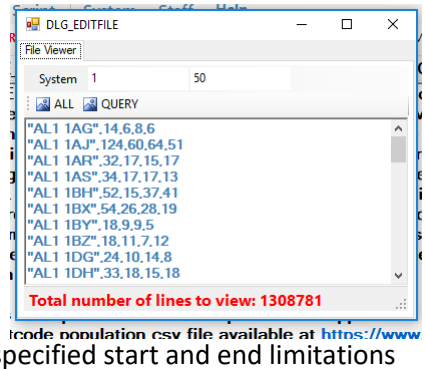
Unlike the ordinary Import command, the folder location of the data file can be specified so does not have to reside in the normal KBL /DATA folder, here a user prefix **ons** is used to indicate the location which has been preset by the Host.Location command, e.g. if we had the file in our normal Downloads directory it would be:

HOST.LOCATION ons C:/Users/user/Downloads

Note, typically it takes just a few seconds to load, say 5 million records with FILE.IMPORT.

File Staging Area Viewer

Once a file has been read into the staging area the data can be viewed, see Menu bar SYSTEM > Show File option to open the File viewer. Note, if you used the File.Set start and end (row numbers) options, the row numbers in the viewer are relative to Set rather than the full file.



Like TAB.SHOW <Dataset> command, you can invoke File Viewer as pop-up screen via a KBL command:

File.Show <start> <end> which can be START, END for full file, specified numbers or parameterised numbers. Note, that without specified start and end limitations a full list of millions of rows could be available for display – which may result in a slower response time.

Moving data from File staging area to MAIN working memory dataset

There are several means of populating MAIN, a key method is by moving between consecutive File rows and specifying what to do with the data. This is done via a LOOP with the implicit FILE parameter for the extent of rows. But first we specify the structure of the receiving dataset in MAIN e.g.

Define the MAIN dataset with the Column names

```
Delete /tab  main          // Clear MAIN and create a new table for file data
ADD /COL c.1  Postcode
ADD /COL c.2  Total
ADD /COL c.3  Males
ADD /COL c.4 Females
ADD /COL c.5  Occupied_Households

ADD /ROWS [NO_FILE_ROWS]        // add NO_FILE_ROWS (implicit File Row count) to MAIN
```

Then we can unpick the file rows into parameters with SPARSER (simple parser) commands

```
LOOP L1  FILE                              // loop to read a file row, unpick each file row
UTIL.SPARSER.NEW /PAR FILE_TEXTLINE              // the text in the current file row
UTIL.SPARSER.DELIMITER ,              // Here comma is the delimiter (standard .csv)
UTIL.SPARSER.FIRST Postcode           // first is Postcode
// could add logic here to accept / reject a row for processing
UTIL.SPARSER.NEXT Total
UTIL.SPARSER.NEXT Males
UTIL.SPARSER.NEXT Females
```

```
UTIL.SPARSER.NEXT Occupied_Households
UTIL.SPARSER.RUN
TAB.GET /FROM_PARS [L1]        // data in parameters is moved to matching col names in MAIN

LOOP END
```

The above script logic moves all data in the File staging area, row by row into MAIN. Note, as it is all under programmatic control, we could test for and handle changes in the data. Similarly, if say, in the above example, we were looking only for certain postcodes (e.g. for Postal Sector 'HR'), we could test the value of the postcode parameter (where marked) and do a LOOP CONTINUE to reject unwanted postcodes. This would have taken a little longer to process but substantially reduced our PC's memory requirements (the file used in this code example had 1.3M rows, HR post codes only numbered 6060 so this 'pre-processing' in the File staging area, minimised the size of MAIN).

Exporting Data from the File Staging Area

As an alternative to pushing data into MAIN and subsequently processing it with conventional KBL, we have the option of working directly on the source data in the File transit area. For example, we could have reformatted individual rows (using the FILE_TEXTLINE line text parameter). Furthermore, we could have divided up the VLD import file in blocks of say 200k records and exported in chunks to using the File.Export commands:

```
FILE.SET /EXPORT CLEARLINES                    // Clear any previous export file definitions
FILE.SET /EXPORT WRITELINE "ref,name,address"  // Create a bespoke line at start of export
FILE.SET /EXPORT WRITELINE  "---,-------,-----------  // And another

FILE.EXPORT ccc_customers_manip.xlsx START 200000  // Export full START to END or part dataset
```

Please note, that this is not a full description or explanation of how to handle big datasets in KBL. A video will be available shortly that will provide other tips and methods available to KBL Studio users.

There will be a follow-up article which will describe how to process Very Large Datasets once loaded into MAIN (and other in-memory datasets).

End of document