Wudang Research Association

CoSort 9.5 and Tableau 8.2
Performance Analysis and Query Run Time Test

Prepared for:

IRI

The CoSort Company

August 30, 2014

Prepared by:

Terri Morgan

Table of Contents

Query Performance Test - CoSort 9.5 and Tableau 8.2 3

Environment 3

Data Sets 4

Installation 4

CoSort 9.5 4

Tableau 8.2 5

Preparation 5

CoSort 9.5 5

Tableau 8.2 5

Test Results Set 1 6

CoSort 9.5 6

Tableau 8.2 7

Test Results Set 2 9

Conclusions 10

CoSort 9.5 10

Tableau 8.2 11

Take-Aways 11

Query Performance Test - CoSort 9.5 and Tableau 8.2

Performance matters, especially for big data. The business emphasizes analysis – being able to detect patterns, create visualizations, and provide insights. But nothing can be analyzed or visualized until the data has been located, extracted, refined, aggregated, and prepared.

IRI's CoSort 9.5 was tested and compared with Tableau 8.2 for performance and usability in data extraction and preparation.

The key question: Given the same data sets, what does it take to prepare a list? And, importantly, how long does it take to run the queries that produce the list?

It should be noted that these two programs are not mutually exclusive. Both can be used to extract data from multiple sources and bring it into one location for reporting or further manipulation. Depending on the data sources, data volume, platforms, and resources, CoSort could be used with Tableau to build a selected data set (flat file) that Tableau could take as input for analysis and creating visualizations.

Put another way, what's the goal? Raw Data or Data with Visualizations for analysis. Obviously, the two are not mutually exclusive.

Tableau provides a refined set of tools to display and render data in more human-friendly formats – Charts, Graphs, Visualizations. It is designed to handle connections to multiple data sets, extract and filter selected data, and then provide that data to its visualization tools.

CoSort is designed to get the data, extract what's wanted, create a new data set and provide that data to whatever other tool(s) may want it.

The focus of this test is only on the data collection tasks – sources, queries, filters and generating the resulting data set – the "heavy lifting" needed to collect and prepare data for analysis.

Environment

The environment for this test was a modest 2012-model year desktop system.

Model:	HP p7-1414 x64
Processor:	AMD A8-5500
Processor Speed:	3200Mhz
Cores:	2 Cores
Processors:	4 Logical processors
Physical Memory:	6GB RAM
Virtual Memory:	12.2GB Virtual
Storage:	1TB HDD
OS:	Windows 8.1 Pro

Data Sets

Two data sets were used, each with two flat files. In the first instance, both sets were very small files. In the second, the files were quite a bit larger.

	Set 1		Set 2	
File	cust.dat	trans.dat	cust.dat	trans.dat
Size	2K	3K	67,969KB	91,094KB
Rows	20	50	1,000,000	2,000,000
Fields/Row	6	5	6	5

For the purposes of this test, the two files (cust.dat and trans.dat) were joined on a generated key intended to simulate an account number.

Installation

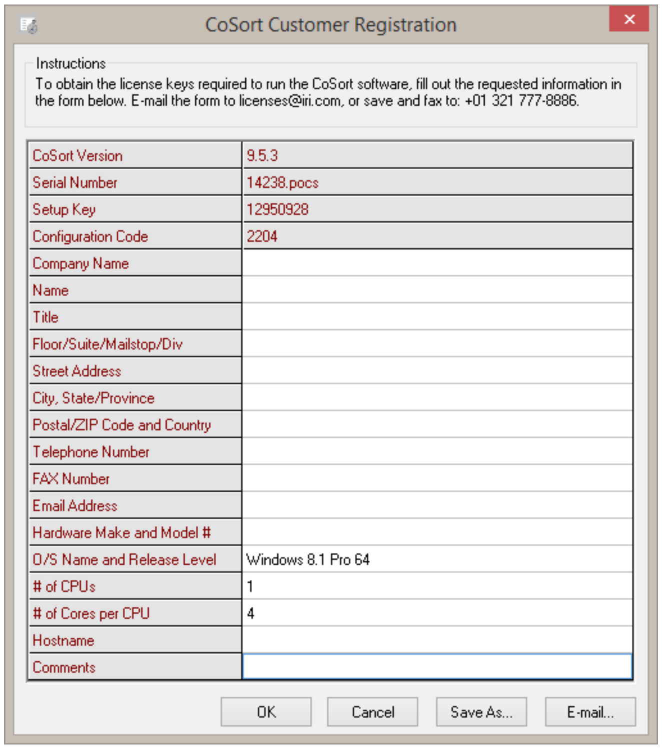
CoSort 9.5

A trial copy of CoSort can be requested. There is no option to directly download a trial from the web site. The full install file is 196,802K.

The IRI tool requires several steps to install. The InstallShield wizard runs as expected, unpacks itself and prompts for the standard options to select the install directory, create a desktop shortcut, and/or a start item.

When finished, it automatically presents a screen to enter a Serial Number (provided by the company with the download link.

Next, the Setup wizard will run. It gives options for "First Time Setup" and "Update License."



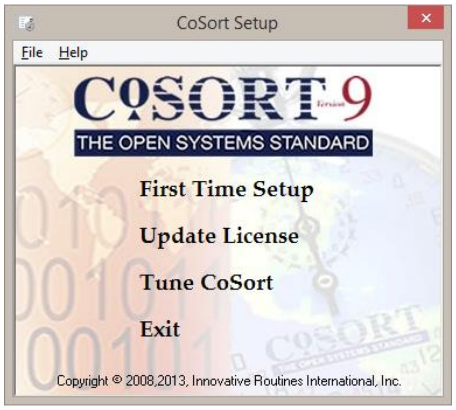
To obtain the license keys required to run the CoSort software, fill out the requested information in the form below. E-mail the form to licenses@iri.com, or save and fax to: +01 321 777-8886.

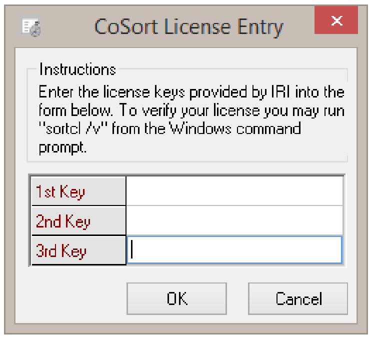
CoSort Version	9.5.3
Serial Number	14238.pocs
Setup Key	12950928
Configuration Code	2204
Company Name	
Name	
Title	
Floor/Suite/Mailstop/Div	
Street Address	
City, State/Province	
Postal/ZIP Code and Country	
Telephone Number	
FAX Number	
Email Address	
Hardware Make and Model #	
O/S Name and Release Level	Windows 8.1 Pro 64
# of CPUs	1
# of Cores per CPU	4
Hostname	
Comments	

The "First Time Setup" wizard extracts some basic information (Customer Registration) from the system and provides spaces to enter user and system details. These details can be saved and must be emailed to IRI to get the three keys needed to activate the software.

Once the keys are available, running /bin/cs_setup.exe opens the Setup wizard. Selecting Update License provides the prompt for entering the three keys.

Once the keys have been accepted, CoSort is ready for use.


CoSort Setup
File Help
COSORT 9
THE OPEN SYSTEMS STANDARD
First Time Setup
Update License
Tune CoSort
Exit
Copyright © 2008,2013, Innovative Routines International, Inc.


CoSort License Entry

Instructions
Enter the license keys provided by IRI into the form below. To verify your license you may run "sortcl /v" from the Windows command prompt.

1st Key	
2nd Key	
3rd Key	

IRI Workbench is an available GUI built with Eclipse running in a Java VM. It provides a visual interface and a code generator.

For this test, the command line utility **`sortcl`** was used together with a prepared script provided by IRI.

Tableau 8.2

A trial copy is available from the web site. It is a fully functional version good for 14 days. Once the registration form (for sales) is complete, the download starts automatically. The download is 125,608K.

The installation is quick and takes only a couple of minutes to run. The InstallShield wizard runs as expected, unpacks itself and prompts for the standard options to select the install directory, create a desktop shortcut, and/or a start item.

That's it. There is a useful option to "Start your trial now" or "Wait until later."

The normal trial-day countdown prompts are displayed at each startup.

Command line options to connect to a server and establish a login are available and can be included in a config file. All other operations are handled via the GUI.

Preparation

CoSort 9.5

CoSort provides two options for accessing data – command line + a scripted file and a GUI.

This test used the command line + a scripted file with two delimited flat files.

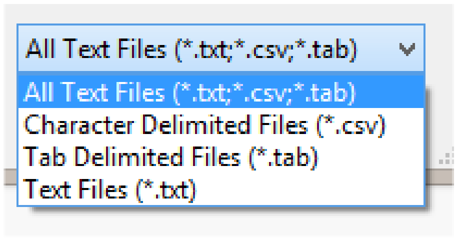
The script file for this test takes the two input files (cust.dat and trans.dat), extracts the specified data, sets up a join, formats (delimits) the result, calculates totals, and sends two filtered data sets to two separate output files.

The scripting language is proprietary but straightforward and easy to learn. Terms like /INFILE and /OUTFILE are self-evident. Language and usage references are provided by IRI.

Tableau 8.2

Tableau's GUI is intuitive. Selecting data (in a variety of formats) is point-and-click.

In this case, there was one additional step: cust.dat and trans.dat were copied and renamed to cust.txt and trans.txt. For the initial Connect To Data option, Tableau does not recognize the .dat extension. Once a workbook has been created, an "all files" option is available on the drop down.


All Text Files (*.txt;*.csv;*.tab)
All Text Files (*.txt;*.csv;*.tab)
Character Delimited Files (*.csv)
Tab Delimited Files (*.tab)
Text Files (*.txt)

Note: These usage differences are mentioned only to clarify the parameters for this performance test. It's hardly fair to compare a command line operation to the selection options in a fully developed GUI.

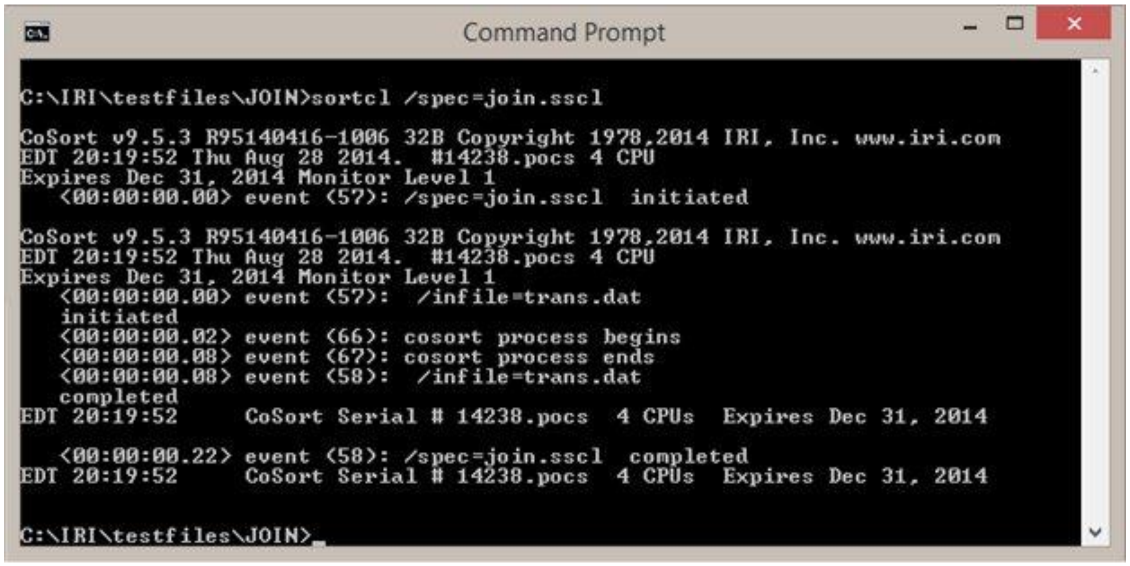
Test Results Set 1

As expected, the small files in Set 1 were no problem for either tool. In both cases, a finished record set of 18 rows was created.

	Start	End	Elapsed	Record Count
CoSort	20:19:52	20:19:52	00:00:00.22	18
Tableau	20:25:23.756	20:25:25.465	00:00:01.709	18

CoSort 9.5

Running CoSort from a command prompt was easy enough. The data files were purposefully put in the same directory as the command script. The command `sortcl /spec=join.sscl` (where join.sscl is the prepared script) ran and created the specified output files.

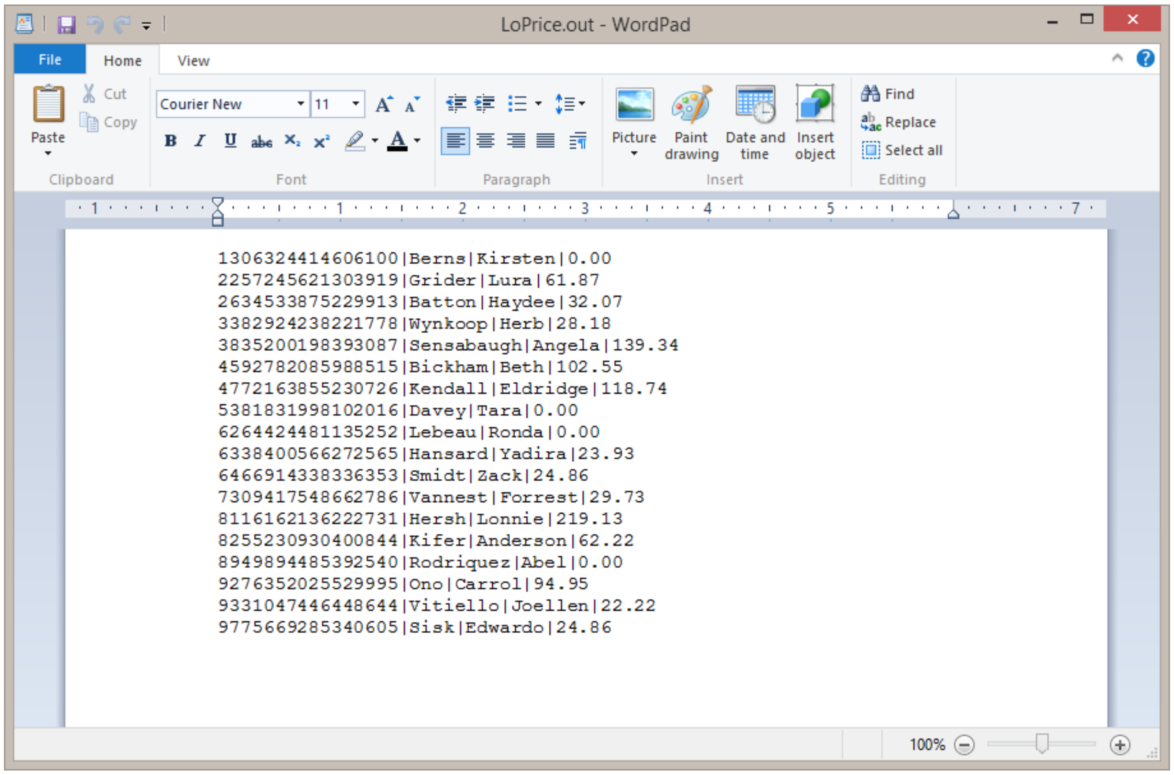


```
C:\IRI\testfiles\JOIN>sortcl /spec=join.sscl

CoSort v9.5.3 R95140416-1006 32B Copyright 1978,2014 IRI, Inc. www.iri.com
EDT 20:19:52 Thu Aug 28 2014.  #14238.pocs 4 CPU
Expires Dec 31, 2014 Monitor Level 1
     <00:00:00.00> event (57): /spec=join.sscl  initiated

CoSort v9.5.3 R95140416-1006 32B Copyright 1978,2014 IRI, Inc. www.iri.com
EDT 20:19:52 Thu Aug 28 2014.  #14238.pocs 4 CPU
Expires Dec 31, 2014 Monitor Level 1
     <00:00:00.00> event (57):  /infile=trans.dat
     initiated
     <00:00:00.02> event (66): cosort process begins
     <00:00:00.08> event (67): cosort process ends
     <00:00:00.08> event (58):  /infile=trans.dat
     completed
EDT 20:19:52        CoSort Serial # 14238.pocs   4 CPUs   Expires Dec 31, 2014

     <00:00:00.22> event (58): /spec=join.sscl  completed
EDT 20:19:52        CoSort Serial # 14238.pocs   4 CPUs   Expires Dec 31, 2014

C:\IRI\testfiles\JOIN>_
```

The start, end, and time-to-complete were conveniently displayed in the prompt window.

The resulting data set can be viewed by navigating to the target directory and opening the `.out` file using any text editor. WordPad will preserve line breaks; Notepad doesn't.

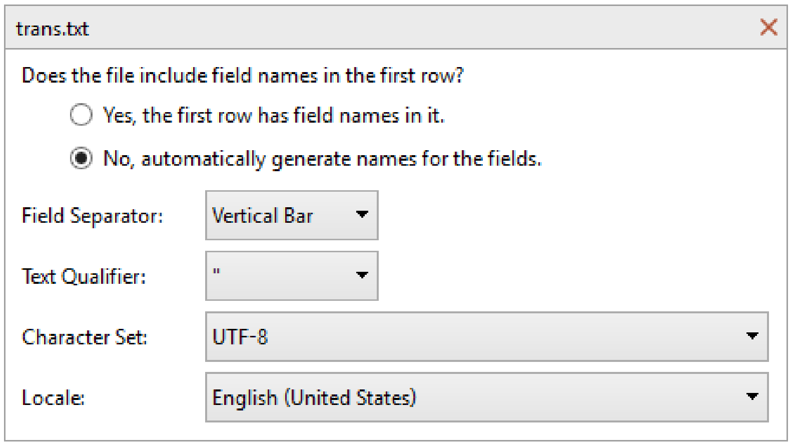
LoPrice.out = 18 records, 4 fields



```
1306324414606100|Berns|Kirsten|0.00
2257245621303919|Grider|Lura|61.87
2634533875229913|Batton|Haydee|32.07
3382924238221778|Wynkoop|Herb|28.18
3835200198393087|Sensabaugh|Angela|139.34
4592782085988515|Bickham|Beth|102.55
4772163855230726|Kendall|Eldridge|118.74
5381831998102016|Davey|Tara|0.00
6264424481135252|Lebeau|Ronda|0.00
6338400566272565|Hansard|Yadira|23.93
6466914338336353|Smidt|Zack|24.86
7309417548662786|Vannest|Forrest|29.73
8116162136222731|Hersh|Lonnie|219.13
8255230930400844|Kifer|Anderson|62.22
8949894485392540|Rodriquez|Abel|0.00
9276352025529995|Ono|Carrol|94.95
9331047446448644|Vitiello|Joellen|22.22
9775669285340605|Sisk|Edwardo|24.86
```

Tableau 8.2

In Tableau, the steps needed to prepare the worksheet were comparable to what was done in the IRI script – select the data, create the join, set a filter, display the data.

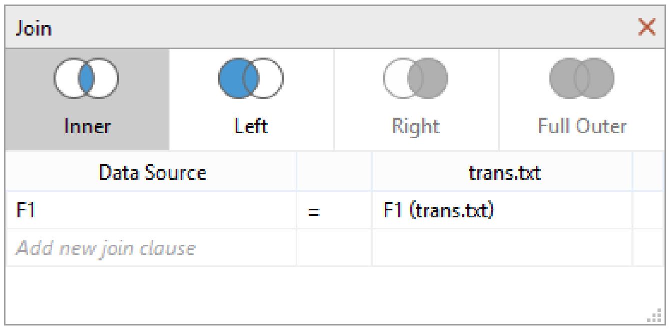


Does the file include field names in the first row?
Yes, the first row has field names in it. [✓ No, automatically generate names for the fields.]

Field Separator: Vertical Bar
Text Qualifier: "
Character Set: UTF-8
Locale: English (United States)

First, the two files (cust.txt and trans.txt) were selected. The delimiter and field heading assignments were automatically created for the cust.txt file, but not for trans.txt. They were quickly set manually.

Next, the join was set on the pseudo-account number. Again, quickly accomplished from the dialog.



Data Source		trans.txt
F1	=	F1 (trans.txt)

Finally, a filter was set to only show amounts less than or equal to 50. This matched the filter set in the IRI script to create the LoPrice.out file.

With the setup complete, the final step was to create a results table.

Another small change was needed to switch the account number from a measure (automatically defined by Tableau) to a dimension. Without this change, Tableau tried to calculate totals using the account number.

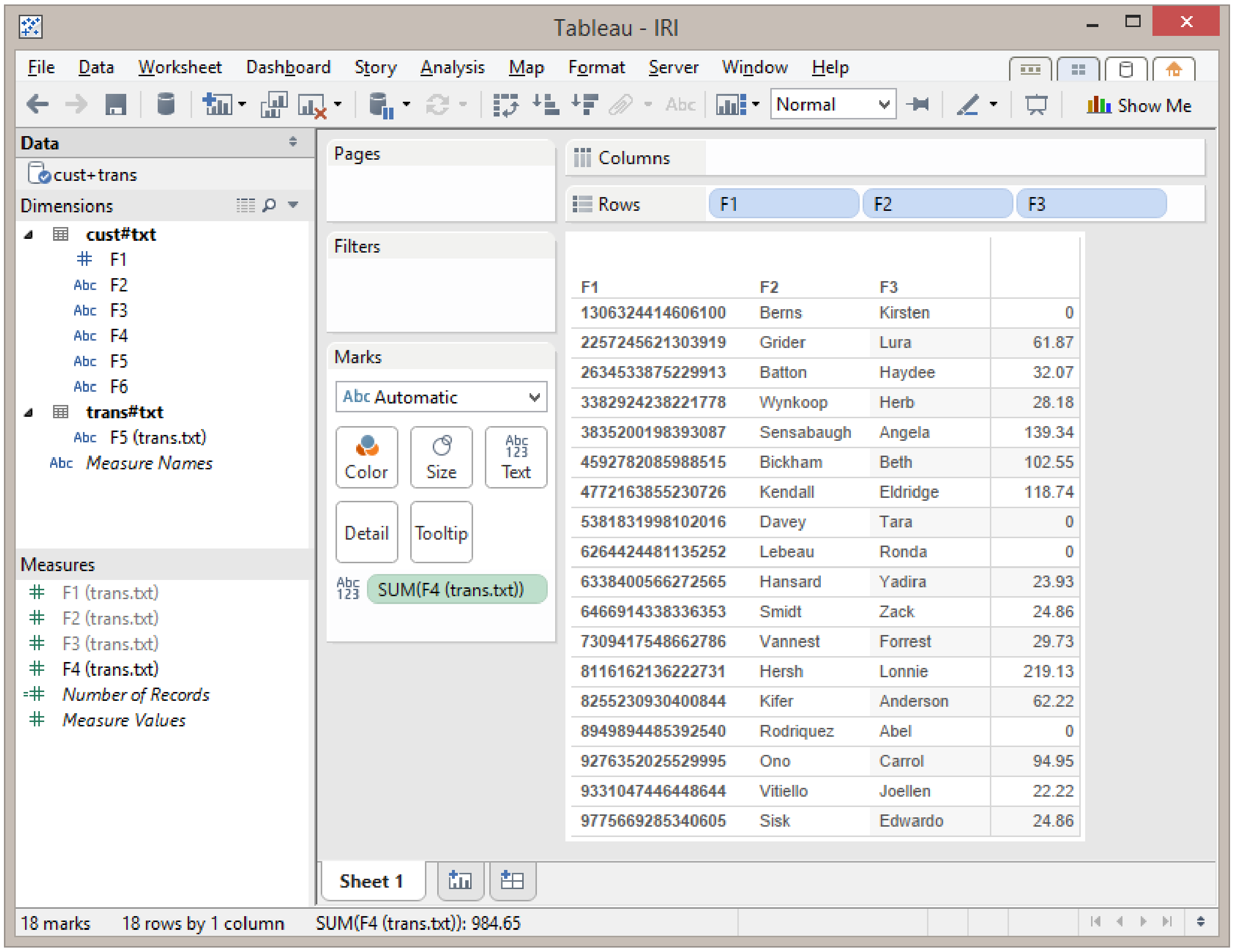
The resulting data set was instantly available for viewing.

The start, end, and time-to-complete were available in the Tableau log.txt file. \Documents\My Tableau Repository\Logs. It's a comprehensive (and seriously not easy to read) log of everything Tableau has done. The two relevant start-end segments for this metric are given below.

```
{"ts":"2014-08-28T20:25:23.756","pid":3664,"tid":"118c","sev":"info","req":"-
","sess":"-","site":"-","user":"-","k":"begin-data-interpreter","v":{}}
```

```
...... (3 queries – one for each of the two tables and one for the join)
```

```
{"ts":"2014-08-28T20:25:25.465","pid":3664,"tid":"118c","sev":"info","req":"-
","sess":"-","site":"-","user":"-","k":"end-data-interpreter","v":{"elapsed":1.703}}
```



F1	F2	F3	
1306324414606100	Berns	Kirsten	0
2257245621303919	Grider	Lura	61.87
2634533875229913	Batton	Haydee	32.07
3382924238221778	Wynkoop	Herb	28.18
3835200198393087	Sensabaugh	Angela	139.34
4592782085988515	Bickham	Beth	102.55
4772163855230726	Kendall	Eldridge	118.74
5381831998102016	Davey	Tara	0
6264424481135252	Lebeau	Ronda	0
6338400566272565	Hansard	Yadira	23.93
6466914338336353	Smidt	Zack	24.86
7309417548662786	Vannest	Forrest	29.73
8116162136222731	Hersh	Lonnie	219.13
8255230930400844	Kifer	Anderson	62.22
8949894485392540	Rodriquez	Abel	0
9276352025529995	Ono	Carrol	94.95
9331047446448644	Vitiello	Joellen	22.22
9775669285340605	Sisk	Edwardo	24.86

18 marks 18 rows by 1 column SUM(F4 (trans.txt)): 984.65

Test Results Set 2

When the data sets were expanded, performance differences were apparent.

Differences were expected due to the two methods: running from a command prompt with no display and running from a GUI application.

The time to run differences (how long it took the queries to run) were significant.

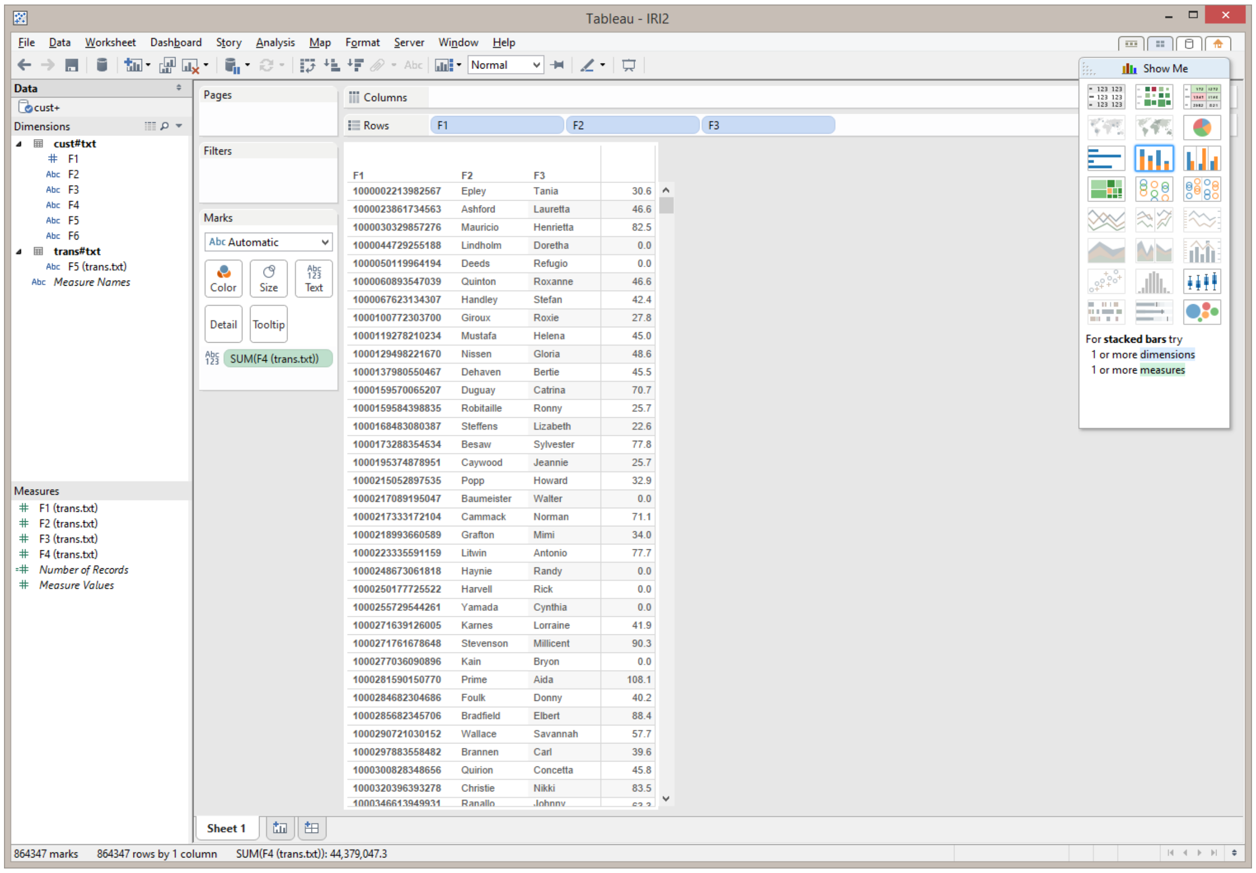
	Start	End	Elapsed	Record Count
CoSort	10:57:20	10:57:46	00:00:26.37	864,347
Tableau	19:22:16.031	19:25:35:933	00:03:19.902	864,347
Tableau*	19:22:16.031	19:22:43.520	00:00:27.481	864,347

* Queries only

LoPrice.out = 864,347 records, 4 fields

The time needed to run the queries in Tableau was comparable to the time CoSort needed to run the queries and generate the output file.

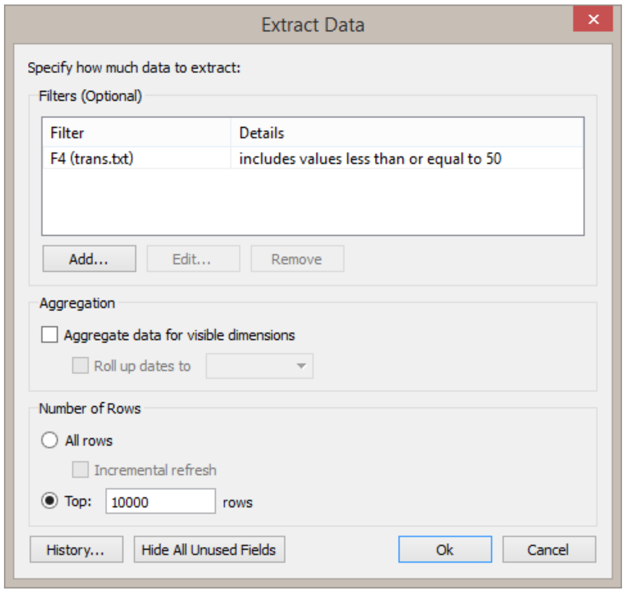
The time to return the full record set to the workbook display added significantly to the time required by Tableau. In addition, two warnings for low system resources were encountered.



F1	F2	F3	
1000002213982567	Epley	Tania	30.6
1000023861734563	Ashford	Lauretta	46.6
1000030329857276	Mauricio	Henrietta	82.5
1000044729255188	Lindholm	Doretha	0.0
1000050119964194	Deeds	Refugio	0.0
1000060893547039	Quinton	Roxanne	46.6
1000067623134307	Handley	Stefan	42.4
1000100772303700	Giroux	Roxie	27.8
1000119278210234	Mustafa	Helena	45.0
1000129498221670	Nissen	Gloria	48.6
1000137980550467	Dehaven	Bertie	45.5
1000159570065207	Duguay	Catrina	70.7
1000159584398835	Robitaille	Ronny	25.7
1000168483080387	Steffens	Lizabeth	22.6
1000173288354534	Besaw	Sylvester	77.8
1000195374878951	Caywood	Jeannie	25.7
1000215052897535	Popp	Howard	32.9
1000217089195047	Baumeister	Walter	0.0
1000217333172104	Cammack	Norman	71.1
1000218993660589	Grafton	Mimi	34.0
1000223335591159	Litwin	Antonio	77.7
1000248673061818	Haynie	Randy	0.0
1000250177725522	Harvell	Rick	0.0
1000255729544261	Yamada	Cynthia	0.0
1000271639126005	Karnes	Lorraine	41.9
1000271761678648	Stevenson	Millicent	90.3
1000277036090896	Kain	Bryon	0.0
1000281590150770	Prime	Aida	108.1
1000284682304686	Foulk	Donny	40.2
1000285682345706	Bradfield	Elbert	88.4
1000290721030152	Wallace	Savannah	57.7
1000297883558482	Brannen	Carl	39.6
1000300828348656	Quirion	Concetta	45.8
1000320396393278	Christie	Nikki	83.5
1000346613949931	Ranallo	Johnny	63.3

864347 marks 864347 rows by 1 column SUM(F4 (trans.txt)): 44,379,047.3

Prompts suggested using an extract rather than the full record set. A subsequent test with a limit of 10,000 records in an extract ran without issue, returning 4397 rows in the joined data set.



Filter	Details
F4 (trans.txt)	includes values less than or equal to 50

☐ Aggregate data for visible dimensions
☐ Roll up dates to

○ All rows
☐ Incremental refresh
● Top: 10000 rows

Conclusions

CoSort 9.5

CoSort takes a bit to get installed and made ready to use. Even getting the command line utility running requires a few steps – installation, activation, registration. The GUI depends on a Java VM, which may also need to be installed. Scripts can be generated by the GUI or hand-coded. Some basic scripts are provided by IRI along with extensive documentation.

There is no “quick start” for CoSort. Basic steps are not well-documented, although much more complex system functions are thoroughly explained. Depending on the reader, the documentation is either “very comprehensive” or “overdone and verbose.” If you are looking for extensive discussions, they have them. The materials are definitely targeted to those who enjoy delving into the details. This is a tool for Programmers and DBAs, not for the casual techie and definitely not for the business user.

After it’s been installed, like most other programs, it doesn’t need to be installed again. Well-versed programmers and DBAs will be able to pick up the scripting language easily. It’s linear and highly flexible with self-descriptive terms.

It’s only after a basic grasp of the command line syntax and coding language / structures has been attained that the program begins to show it’s value.

And value it has.

Scripts called from the command line run quickly, even with larger data sets. The flexibility of using a scripting language to extract data may not be immediately obvious. In many cases, it may seem like too much overhead, especially when a simple SQL query will do exactly the same thing.

Where CoSort shines is when the data extracts wanted are from multiple tables and a series of nested SQL queries is required to properly execute the joins. Being able to script the extracts, write them out to a flat file and then use those files as the input for analysis saves time and compute resources.

Tableau 8.2

Tableau installs easily and is ready to use in just a few minutes. It comes bundled with several sample data sets. A chart using the sample sets can be ready in under 10 minutes – download to complete.

The GUI is generally intuitive and the controls are well-designed. Navigating between data set selections and the charting workspace requires minimal effort. There are videos and user documentation that cover basic and advanced features.

Building data sets and creating BI charts are two separate activities.

Data source selection options in Tableau are extensive. Scripts are available to pre-set connections. There is no command line / scripting to get data or build joins. Those actions are handled via the GUI. Once all the connections are established, an experienced business user could create their own basic visualizations. Managing complex joins and nested queries is a different matter.

Tableau worked quickly and rendered basic data to the display when the live record sets were smaller (~10,000 rows). For the larger data sets, it prompts to "Create an extract."

Tableau is designed as a BI tool; not a data extract tool. It handles most extracts quite well and delivers them to the GUI with minimal effort. Visualizations can be generated easily. Time to load the results depends on the number of records and the particulars for rendering the display.

Take-Aways

Tableau is a first-class BI tool that provides great tools for collecting data and displaying visualizations. Performance slows when using larger, live data sets. The data access tools are comprehensive. Multiple data sources can be selected easily.

CoSort is a data extract utility that can be used as a stand-alone or combined with tools like Tableau. Its command line operations are speedy. Combined with prepared scripts, it can enhance performance and reduce run times for complex queries and data extracts.