TDE or Live? When to Use Tableau Data Extracts (or not)

I recently answered a question for a new Tableau user on when to use a Tableau Data Extract (TDE) vs. a live connection, here's a cleaned-up version of my notes:

Why Use a Tableau Data Extract?

My preference is to first consider using a live connection because extracting data adds another step to the data delivery chain. However there are many situations where that isn't a workable solution, so Tableau has created Tableau Data Extracts to support situations where a live connection to the source is:

1. **Not possible**. Sometimes a Tableau viz can't have live connection to a production system, for example when you want to share a viz with someone not inside your premises. The extract can be published (whether in a viz or as a published data source) to Tableau Server or Online, or saved in a Tableau packaged workbook (TWBX) or packaged datasource (TDSX), or sent "naked" as a TDE file.
2. **Too slow**. There are a number of variations on this:
    1. For example a production system is on a slow network connection so a TDE can be created locally and only have to go over the slow network.
    2. Because data extracts are highly optimized for queries they can be much, much faster than a live connection. I regularly see 100x improvement in load times using Tableau data extracts over MS Access, to the degree that my muscle memory is tuned to making an extract as the first thing I do after connecting to an Access-based source.
       Note that older posts (prior to the release of Tableau v8.2) on extracts will talk about them being faster than raw Excel/text connections because at the time the "legacy" aka MS JET engine was used. That is no longer the case with the new Excel/text connector. The new connector takes advantage of Tableau's data extract (data engine) technology to create a data extract in the background, that's why the first connection can be slow and then creating an extract can be instantaneous.
    3. The Tableau queries to the live connection might slow down operational queries too much, so having Tableau only query at scheduled extract refresh times is preferable.
    4. Data volumes could be such that millions of records per week of raw data that would be too slow to run live Tableau queries on might be aggregated in a TDE down to dozens or hundreds per week based on some set of categories/dimensions in the data.
3. **Unable to handle the record volumes**. MS Excel is limited to 1M records, MS Access tops out anywhere in the 100s of thousands to couple M records depending on the complexity of the table, etc. whereas a TDE can potentially handle billions of records. Another case is situations where the data is stored in multiple tables (potentially across multiple data bases) and a UNION query is used to generate a result that is too big for a live connection but fine for a TDE.
4. **Exposing too much data**. There are four cases where extracts can effectively improve security by reducing what data is made available:
    1. We can create extract filters on TDEs so only the necessary records are included.
    2. We can set up the extract to only include fields used in the workbook, in other words we can exclude columns from the extract.
    3. Extracts can be configured to aggregate the data and therefore hide record-level detail.
    4. For file-based sources when we include the files in a TWBX it's the whole file, so for an Excel file that means that every worksheet in that file is included in the TWBX. If we extract the data then only the necessary data for the workbook is in the TDE.
5. **Unable to handle the data volumes**. A related case is that since a TDE is highly compressed it can be a lot smaller than the original uncompressed source. I've seen people use TDEs instead of file-based sources to make distribution of packaged workbooks easier.
6. **Not supporting certain calculations**. Tableau data extracts have generally supported more functions than any particular data source (with the exception of RAWSQL functions). One example is that in the "old days" before Tableau 8.2 with the new Excel/text connector we were stuck with the MS Jet engine for connecting with Excel & text files and that couldn't handle COUNTD(), MEDIAN(), In/Out of Sets among other drawbacks, so we'd create an extract. Another example is that currently not all sources support the Level of Detail Expressions introduced in v9.0 and again we can work around that by creating an extract.
7. **Unable to handle the complexity**. There are various computations (such as using top and conditional filters, nested calculated fields, etc.) that TDEs can handle in combined ways that some data sources can't. For example MS Access databases are one of my main data sources and in some Tableau worksheets if I switch from the TDE to the live connection the MS JET engine gives me a "query too complex" error.
8. **Actually a situation where multiple file-based sources needed to be put together**...with TDEs it's possible to add data to an extract from multiple file-based sources, which can be handy when you are integrating data from various producers at different times. Tableau is working on improving this: At the 2015 Tableau Conference they demoed a feature for creating federated queries across multiple data sources (including server-based sources, other TDEs, etc.). From what I saw Tableau will be able to do this in a live connection, however I'm guessing that we'll often want to be using TDEs for performance reasons.

Other Features of Tableau Data Extracts

A few other advantages of TDEs are:

- **Materialized expressions**. Tableau will "materialize" record-level calculations that use only fields from a single data source and are not dependent on run-time values — i.e. not using TODAY(), NOW(), USERNAME(), ISMEMBEROF(), or a parameter — as fully indexed & compressed fields in an extract. This can improve performance in many cases, for example when splitting name or address fields and/or creating datetime fields out of strings.
- **Access to cloud-based data sources**. In order to make cloud-based sources such as Salesforce.com, Google Analytics, oData, and the Tableau Web Data Connectors useful for the kinds of at-the-speed-of-thought analytics that Tableau enables we have to use Tableau data extracts. Other cloud-based sources such as Amazon Redshift, Google BigQuery, and Microsoft Azure can be used as a live connection or extracted as needed.
- **Option to publish to Tableau Public**. For performance reasons we can only use TDEs when publishing to Tableau Public.

TDE Limitations

However, Tableau Data Extracts do have some limitations and there are cases when they are not suitable or more difficult to work with than a live connection:

1. TDEs are by definition not a live connection to the source. This means that Tableau Data Extracts are not usable if you're needing "real-time" data in your Tableau viz. Also if the refresh time of a TDE is more than the desired data refresh time then TDEs aren't really feasible.
2. Tableau Data Extracts can't be created from OLAP sources such as Oracle Essbase or SSAS. They can be created from SAP BW cubes, however.
3. Changing the data structure of the underlying data can require rebuilding the entire TDE, which may not be very easy, take too much time, become impossible if the file-based source you used for an incremental append is no longer available, etc.
4. Tableau's support for incremental loads, slowly changing dimensions, and updates to existing rows is minimal to non-existent.
5. Tableau Data Extracts do not support RAWSQL functions, nor can we use Custom SQL on an already-created extract. One use case for RAWSQL is when the underlying data source supports a given function and Tableau does not yet support that feature for that source.
6. TDEs can become too slow to refresh and/or queries on them become too slow based on the data structure, here are some known factors:
    1. many rows (anywhere from millions to billions)
    2. many columns (when they get into the hundreds)
    3. lots relatively non-compressible (high-cardinality) columns
    4. many complex materialized expressions

So a billion-row extract might be plenty fast and a million-row extract on a complex data structure might be too slow, your best bet is to do your own testing.

7. As of this writing (January 2016) I haven't heard of anyone else being licensed to read from TDEs so the only pieces of software that can read from TDEs are Tableau Desktop, Tableau Reader, Tableau Server, Tableau Online, and Tableau Public. There's no published API for reading TDEs and trying to save large CSVs from a Tableau worksheet is likely to run into out-of-memory problems so if you're looking for more permanent storage for data so you can get at it later you're likely to want to look elsewhere.
8. Refreshing TDEs puts more and more load onto Tableau Server and that can impact delivering visualizations, so doing the work to make the underlying source fast enough to use a live connection may be preferable to the extra hardware & configuration needed to make the TDE refresh fast enough.
9. TDEs don't include user-level security, those have to be set up higher up in the stack in the Tableau Server data source and/or Tableau workbooks that use the TDE, which means there's extra work to prevent unauthorized users from getting access to the data in the Tableau views and the TDE itself. It may be better to implement that security in the raw data source (which I know makes my DBAs happy because they get to retain control).

To eliminate and/or work around the performance limitations of TDEs I see people doing one or more of the following:

1. Read Designing Efficient Workbooks by Alan Eldridge and implementing the suggestions there, it's the [insert holy-book-of-your-choice metaphor here] for Tableau performance tuning.
2. Create multiple data sources on the same underlying data, the basic distinction is using a fast & lightweight TDE for the high-level views and then the detail reached via drill-down (i.e. Filter Actions) is stored in a big, relatively slower TDE or live connection.
3. Use ETL tools such as Alteryx or Trifacta to pre-compute, pre-aggregate, and transform the data to make it fast in Tableau (and potentially use a TDE).
4. Do the necessary performance tuning in the existing data source fast enough to use as a live connection.
5. Deal with high volume/high performance requirements by creating a new data source whether that be a tuned datamart/data warehouse/data lake or using something like Teradata, Vertica, Hadoop, Exasol, etc.

Conclusion

Thanks to Brian Bickell for To Extract or Not to Extract (published 2014-04-29) and Tom Brown for Tableau Extracts (published 2011-01-20), those posts helped validate and round-out bits that I'd missed. Also thanks to Alan Eldridge for Designing Efficient Workbooks, it's on my "must read" list of Tableau resources. If you have other pros & cons of extracts, please let me know!

Related

O Extract, Where Art Thou?
October 31, 2014
In "Tips and Techniques"

Creating Lists of Values for Tableau from Text & Excel Sources
January 8, 2018
In "Tips and Techniques"


I Have Wee Data - Microsoft Access and Tableau
August 11, 2014
In "Tips and Techniques"

This entry was posted in Tips and Techniques and tagged data extract, extract, extracts, incremental refresh, live connection, refresh, tableau data extract, Tableau data extracts, TDE, TDEs, TDS, TDSX on January 5, 2016 [http://drawingwithnumbers.artisart.org/tde-or-live-when-to-use-tableau-data-extracts/] by Jonathan Drummey.

15 thoughts on "TDE or Live? When to Use Tableau Data Extracts (or not)"

Lee
January 5, 2016 at 3:58 pm

A very timely and concise listing of the pros\cons of a TDE. Thanks for the efforts....

Joshua Milligan (@VizPainter)
January 5, 2016 at 5:13 pm

Another, relatively minor use for extracts is to create an aggregated extract. This can improve performance in some cases (as long as detail is not needed for analysis) and (especially before LOD calcs) allows for the creation of secondary sources to solve mis-matched granularity problems via blending.

Michael Mixon

January 5, 2016 at 7:38 pm

A really nice summary of the pros and cons of TDEs. At our organization, live connections to any database that isn't a desktop "database" like Access or Excel isn't permitted, so we rely very heavily on TDEs to optimize our Tableau experience. Prior to our purchase of an Alteryx license last month, we had to generate our TDEs via a very slow and error-prone Custom SQL union of multiple Excel files extracted from our data warehouse via Business Objects. Best case it would take ~4 hours to refresh the TDE every day, but more often than not the initial refresh would fail and then we'd have to kick off another one, resulting in about a 6-7 hour wait time for a refreshed TDE. With Alteryx we've been able to get that refresh time down to approx. 1.5 hours, with the largest portion of that now being the time it takes Business Objects to generate the Excel files. I'm hoping to convince our IT department to let us point Alteryx at the data warehouse directly and simply add in the Business Objects universe logic into our Alteryx workflow, but I'm not holding my breath on that one. 🙂

Chris
January 6, 2016 at 9:44 am

Nice post – thanks for taking the time to aggregate a lot of this.

Chris

Marc McD
January 11, 2016 at 10:12 am

We're just getting started here at my organization on Tableau – only 1 Desktop license and trying to figure out how to distribute dashboards to the business. This tip allows me to have Tableau Reader installed on users machines so they can interact with the content. Thanks!

Sean
January 12, 2016 at 4:40 am

A great article on the pros and cons. If you find that the cons outweigh the pros of TDE, then take a look at Exasol's Tableau Turbo which was specifically designed to help Tableau users overcome the limitations of TDEs. http://www.exasol.com/en/solutions/business/tableau/

Matt Lutton
April 29, 2016 at 2:10 pm

Currently, a TDE is also required for publishing workbooks with local file types to Tableau Online — although I'm not sure if that will change anytime soon.

The currently supported live connections for Tableau Online are listed here:
https://onlinehelp.tableau.com/current/online/en-us/to_publish.htm#live-connection

---

Tim Terrell
May 12, 2016 at 3:17 pm

Are extracts encrypted? If not, I would think they could be a problem for the health care industry that has requirements around protecting patient privacy. If they are not encrypted, I would think live connections would be preferable to the possibility of an extract with patient data living on a laptop computer.

Jonathan Drummey Post author
May 12, 2016 at 9:59 pm

Hi Tim,

Extracts are not encrypted. Where extracts live depends on how the environment has been configured and what permissions have been given. At one end of the spectrum is a Tableau Packaged Workbook (.twbx) file that is essentially a zip file with the workbook's metadata and a TDE that is not secured and non-encrypted, and any Tableau Desktop install could connect to it and read the data. At the other end of the spectrum would be an extract that has been published to Tableau Server or Tableau Online where only allowed users have permissions to access the underlying data. I've seen both ends of the spectrum in healthcare, the level of governance and control from organization to organization (or even department to department) has huge variation.

Jonathan

Pingback: Tableau TDE Vs Live DB Connection | Jose Uzcategui

Remy Rosenbaum
October 13, 2016 at 2:44 pm

Great post! One solution that companies use to live connect Tableau to large data sources is to add a query acceleration layer on the database. This lets you live-connect to your data while maintaining interactive speed. This is especially important when you want to run Tableau on data stored in Hadoop. Jethro is one example of a company that does this. After installing it, you just point Tableau to Jethro as the data source. No need to use/ maintain extracts.

Jonathan Drummey Post author
October 13, 2016 at 2:52 pm

Hi Remy, that's a nice point. For the more general reader, I'd also like to point out that Remy posted from a jethro.io address so I think Remy has some self-interest here. 🙂

Pingback: Sharing #1 – Discovering Analysis

Pingback: TDE or Live? When to Use Tableau Data Extracts (or not) | The Data Diaries

Pingback: The switch to self-service marketing analytics at zulily: best practices for using Tableau with BigQuery – Cloud Data Architect