Daniel Pilla, Sr. Enterprise Architect
October 2019

Qlik Sense Data Connection Analyzer

The Qlik Sense Data Connection Analyzer is a Qlik application that parses script log files and queries the QRS API, allowing analysis of data connection usage, patterns, age, and cleanliness across a Qlik Sense site. In short, this app will help to run a leaner, more performant, and more easily and holistically governed Qlik Sense site.

The application is supported by Qlik's Americas Enterprise Architecture team, and uses only native connections and capabilities of the Qlik platform—making it plug-and-play.

Common Questions & Associated Benefits of the App

- **Which connections are no longer used?**

  A data connection is found in a script log and exists in the Qlik Sense site, however no associated applications currently exist that at one point had used it.

  - Removing unused connections will increase performance across a site, as less connections will have to be evaluated in security rule evaluations. If a site has hundreds or thousands of connections, this calculation time can build up.
  - Removing unused connections makes general administration easier, as there is less to manage.

- **What connections have never been used?**

  A data connection exists in the Qlik Sense site, but no reference exists to it in any script log.

  - It is common that users will create data connections to test connectivity, but then never actually use them. By leveraging this app, one can identify connections that have never been used and have existed for x amount of time (say 90+ days), so that action can be taken to remove them. The benefits from both simpler management and performance are listed in the bullet above.

- **What connections have been deleted that used to be used?**

  A data connection that is found in script logs, but no longer exists in the Qlik Sense site and no app is using it.

  - By parsing the script logs, one can visualize old data connection names/paths that can help to serve as an audit trail.

- **What are the most widely used data connections?**

  - Depending on how this application is deployed, whether it be strictly administrative or potentially visible to developers, this metric is important both administratively and socially throughout the organization.

- **Do we have duplicate data connections?**

  - By analyzing the connection strings, one can tackle duplicate connections to the same source data. This eases administration overhead and will ensure that there is reusability/consistency across the platform.

- **Who owns what data connections?**

  - This is valuable from an access/security perspective to ensure governance of data sources.

- **Who is using what data connections?**

  - While “User A” might own “Data Connection A”, “User B” might also have read access to that data connection. This can of course be visualized through the audit capability of the QMC, however this application will physically reveal who is executing any reloads of those data connections, giving greater visibility and allowing a deeper level of auditing and governance.

- **Where are data connections being used?**

  - Let’s say a data source is being transformed and will be moved from one database to location. The first question one might ask is, “What applications are using that connection, so we can re-route it to the new db and make adjustments to the load scripts?”. This historically has not been easy to answer. This application allows one to select that connection and visualize apps that are using it.

- **Via what mechanism are data connections being used?**
    - The application visualizes what connections are being run as tasks, manually, or in ODAG (or other API) requests. This is crucial in understanding user behavior.

Environmental & App Prerequisite Steps

- Import the qvf into the QMC
- Grant 'Read' access to the **'monitor_apps_REST_app'** data connection in the QMC (this is a default data connection shipped with the product) to the user that will be reloading the application if it is planned to reload the app from the Hub. Ensure that the user can see this connection in the Data Load Editor. Occasionally this takes a services restart to take effect.
- Ensure that the Qlik Sense service account has **'RootAdmin'** access. The **'monitor_apps_REST_app'** data connection is reloaded by the service account, and that account needs access to specific resources via the QRS API that are not available otherwise.
- Open up the application and navigate to the *Data Load Editor*. Navigate to *“Variables”* tab.
    - Ensure that **‘vCentralHostname’** contains the hostname of the Central Node and is directed to a virtual proxy using Windows authentication. If it has a virtual proxy prefix, be sure to include it (e.g. *mycentralnode/myprefix*). The user that is loading the app must also have access to the **ArchivedLogsFolder** data connection and optionally have access to the **ServerLogsFolder** data connection (as the server logs are required due to the fact that manual reload script logs are not archived).
    - **‘vNumLogDays’** refers to the age in days of the log files that should be parsed. The default is 99999999 or such that would be “all-time”. See “Reload Tips and Notes” below for some practical examples of what one might want to set this to if not the default.
    - **‘vNumDaysForUsedDataConnectionToBeConsideredUnused’** is not only the world’s longest variable name, but it is also the variable that determines the age in days of a used (at least once) QRS data connection that if beyond,

one would want to consider it unused. For example, if a data connection was once used, but has not been used in greater than 90 days, consider it unused.

- **‘vIsMultiNodeSite’** should be self explanatory 😉

- **‘vCaptureTaskLogsOnly’** is arguably the most important optional variable. This variable determines whether you only pull from the Archived Logs folder or from Server Log folders as well. Hub reloads are not ever archived, so if it is desired to track *all* reloads (e.g. for a Dev tier), ensure this is flipped to ‘0’. If it is flipped to ‘0’ and you have **‘vIsMultiNodeSite’** flipped to ‘0’, then the app will pull from the default ArchivedLogsFolder and ServerLogFolder data connections. If both **‘vCaptureTaskLogsOnly’** and **‘vIsMultiNodeSite’** are set to ‘1’, then one must ensure that connections exist to all node’s **‘%programdata%/qlik/sense/log/’** folders and the service account must have read access to them.

- Reload the application either via a Task in the QMC or as a user other than the service account from the Hub.

Reload Tips and Notes

- On the *“Variables”* tab of the script, take notice of the variable **‘vNumLogDays’**. This variable sets the age of the logs back from the current time of reload – meaning, if set to ‘30’, the app will reload all logs that are 30 days old or less. It is important to ask:

    - What is the largest reload cadence of an app? One week? One month?

    - How long is it until organizationally it is determined a connection is no longer used?

  These questions will help determine the size of that number for the most efficient reload. It is of course possible and common to keep the default and get all-time data, just note that this reload could potentially take many hours given the size of the site, size of the log files, age of the site (# of log files), etc. More often than not in very large sites, setting it to something like ‘90’ or ‘180’ is more than sufficient, unless it is desired to have a complete historical audit.

- If you have a large amount of logs, very large log files, and/or want a complete audit, you can load the app in “chunks” by reloading the app multiple times, incrementing the **‘vNumLogDays’** on each reload. The application performs incremental reloads, so if you set it to 30, and then 90, and then 180, and then 360, and so on, the result will stack.

Please see the “Reload Benchmarks” section below for time estimations. Chunking your reload could be beneficial if you have shorter reload windows.

- If **‘vCaptureTaskLogsOnly’** is set to ‘1’ on a multi-node site, _only the **ArchivedLogsFolder** connection is necessary. Connections to RIM nodes **ServerLogFolders** do not have to be made, and will be ignored even if they have been set in the inline table below the variable.

Reload Benchmarks

- The application has incremental loading of the log files built-in. QVD files are stored in ‘**%YourShareLocation%/ArchivedLogs/qs-data-connection-analyzer’**. These QVD files are *pre-transformational*, so that if transformational ETL changes are desired post-log scrape, you will not have to reload all of the logs.
- The reload time of the app will depend more heavily on the size of the log files than the quantity of log files. The first reload will always be heavy, so it is advised that this is run off hours, potentially over a weekend. All subsequent loads are automatically incremental and will be far faster, with qvds being stored into the *ArchivedLogsFolder* directory under the *‘qs-data-connection-analyzer/’* directory. Here are example execution results:
    - Three-node cluster with 1,400 logs (some of which are gigantic (500 MB+)
        - Initial load: 1 hr 10 minutes (all time)
        - Incremental load immediately following: 2 minutes
    - Single node environment with 1,000 logs
        - Initial load: 7 minutes (all time)
        - Incremental load immediately following: 10 seconds
    - Large environment with many rapid reloads (over 30k logs)
        - Initial load: 11 hours (all time)
        - Incremental load immediately following: 8 minutes

Usage & Interpretations

- As the section above states around common questions and benefits that the app provides, this application is generally designed to give oversight and allow governance and enable cleanup your data connections.

- Sheet 1: Intro

    - The intro sheet provides a link to this documentation.

- Sheet 2: Dashboard

    - Please keep in mind that if the `vNumLogDays` variable is set to a number that is not 9999999 (or another similarly high number to ensure all logs are captured) that the KPIs below ***could be misleading.*** For example, if the `vNumLogDays` variable is set to '7', the "Connections Used" KPI will only count the number of connections that are found in script logs from the last 7 days. There could be others that are used in bi-weekly reloads, monthly reloads, etc. The same goes for all other KPIs. Please set the variable accordingly.

    - This sheet is designed to provide an overall look into the data connection/app ecosystem without going to deep. Analysis can certainly be done on this sheet, but it is intended to support:

        - Quickly checking KPI's

            - "Connections Used" – This metric is the count of the number of data connections that are found in the script, the QRS, and an application associated to one of those script logs exists in the QRS. It does not care when the app that used it was last reloaded, so take this into consideration.

                - Second measure of "Unused last *x*" – This metric sums the number of connections where script logs exist, the data connection exists in the QRS, and associated applications exist, however the applications have not been reloaded in *x* days. This *x* number is configurable in the load script. These connections can be considered as tied to "stale" apps that have also not been used in any other apps since that variable amount of time.

            - “Connections No Longer Used” – This metric sums the number of connections that have been found in the load script and the QRS, but all of the associated apps have been deleted. This implies that at one point in time these connections were used, but are no longer. More than likely, these are connections that should be considered for removal.

            - “Connections Never Used” – This metric shows the total amount of data connections found in the QRS that do not exist in any script log (and therefore cannot be tied to any app). Most commonly, this bucket falls into data connections that were created to test connectivity, but then never part of an application’s reload. It could also be that ‘vNumLogDays’ is set to something less than all time, so these connections might in fact exist in logs that are older, but the app has been instructed not to parse them, and has therefore not found them. This should still qualify them as not used and they should be considered for removal provided the number of script log days is set to something substantial (30 days, 90 days, etc).

            - “Connections Only Found in Script” –This KPI will capture data connections that have been deleted from the QRS and where the associated applications have also been deleted. For example, these could be deprecated apps tied to old data systems that are no longer used. These could be useful for historical auditing purposes.

        - Selecting an app and seeing what data connections it uses

        - Selecting a data connection and seeing what applications are associated to it

    - It is then generally expected that one would navigate to another sheet, either holding a selection or otherwise.

- Sheet 3: “Ownership & Availability”

    - This sheet is intended to show connections by their owners, what apps they are being used in, what streams those apps are in, how old the data connections are (and if they are used), etc. It is intended more as an informational, supplementary sheet, not so much one that one would “take action” on other than holding selections and navigating to another sheet to continue analyzing down a path (such as navigating to the “Usage” sheet).

- Sheet 4: “Usage”
    - The usage analysis is critical in this application, however it is probably the most complex as well. The sheet includes two of the bar chart visualizations from the dashboard as helper visualizations, acting as app and data connection selectors without jumping sheets. The primary focus is to view *when* and *how* the data connections in question were used. Were they loaded once? One thousand times? Were they run as tasks, manually, as ODAG?
    - One can visualize the number of distinct times a connection has been used (per script log, not total times within that script log – so per reload), as well as visualize when those reloads occurred over time that included those data connections.
    - Usage should be a critical component of analysis when evaluating data connections.
- Sheet 5: “Used Connections That Have Not Been Used Within *x* Days”
    - This sheet specifically provides detail to the second measure of the first KPI on the dashboard, focusing on connections that exist in the script, QRS, and have applications associated, but said applications have not been reloaded in over *x* days (configurable in the *“Variables”* tab through ‘vNumDaysForUsedDataConnectionToBeConsideredUnused’). Yes, it is a long variable name.
    - These connections are important to focus on because it implies that more than likely stale applications exist on the server, as they haven’t been reloaded in *x* time (30, 60, 90 days, etc) – meaning that both the applications and data connections associated should be considered for cleanup.
- Sheet 6: “Unused Connection Analysis”
    - This sheet does exacly what the title states—It allows one to rapidly target which data connections are unused, and therefore should be considered for cleanup/removal. The coloring is based off of the age of the connections—either the age since creation if they have never been used, or the age of the last script log of when they were last used.

  - Consider these as potentially metrics that could be automated and programmatically acted upon—leveraging custom alerting (NPrinting, perhaps?) and custom scripts to do cleanup. An example being, weekly or monthly reports being sent out to connection owners or admins letting them know that their connections haven't been used in x days (or ever), and need to be acted upon or else they will be programmatically removed in another x days.

- Sheet 7: "Duplicate Analysis"

  - This sheet is designed to illustrate duplicate data connections, which are determined by their data connection strings. For example, if it is seen that multiple people are connecting to the same sources, it might be best to consider merging these into a single data connection and managing access using security rules ro reduce clutter and encourage more data governance and control.

- Sheet 8: "Detail"

  - The last sheet is simply a flat table so that one can drill to the detail if needed. Feel free to add additional columns here if necessary.

Pro-Tips

- For the first reload test, set the 'vNumLogDays' variable to 1 so that it can be tested that everything is working properly. Once the app has successfully loaded, then try increasing the number of days. This is especially helpful when testing a multi-node environment, where the data connections to the other nodes should be confirmed to be correctly established before trying a lengthy reload.

- After deploying the app and doing some manual cleanup of the Qlik Sense site, consider creating a sustainable, long-term approach that is proactive. Alerts (emails or even reports with NPrinting) could be tied into the app to email data connection owners (as described above in the notes on *Sheet 5: "*Used Connections That Have Not Been Used Within *x* Days*"*). Action could then be taken by the owners proactively, keeping data connections from growing unorderly.

- Consider applying section access to the application and making it available to developers. It could be controlled by either data connection owner or application owner, depending on how it would be leveraged.

- When setting a task for this application, consider its usage—it is likely that this is an application that only needs to run weekly or potentially even monthly. However, if the app is being deployed to developers and leveraged for alerting or otherwise, then one might want to consider reloading it daily. Incremental loads will be in seconds or minutes – see benchmarking above.

About Qlik

Qlik is on a mission to create a data-literate world, where everyone can use data to solve their most challenging problems. Only Qlik's end-to-end data management and analytics platform brings together all of an organization's data from any source, enabling people at any skill level to use their curiosity to uncover new insights. Companies use Qlik products to see more deeply into customer behavior, reinvent business processes, discover new revenue streams, and balance risk and reward. Qlik does business in more than 100 countries and serves over 48,000 customers around the world.

qlik.com