Log Parser Studio Quick Start

Need to skip all the details and run queries now? Start here!

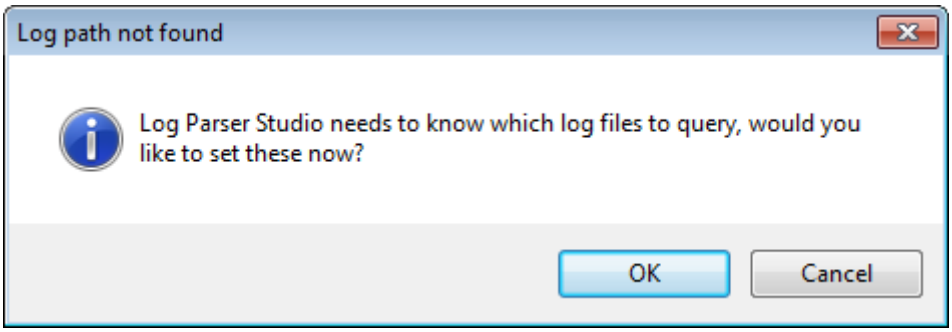
The very first thing Log Parser Studio needs to know is where the log files are, and the default location that you would like any queries that export their results as CSV files to be saved.

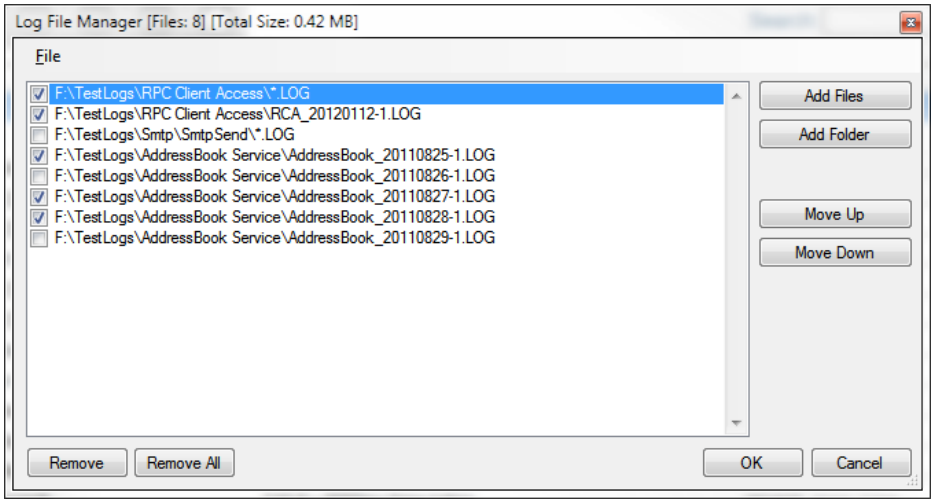
1. Setup your default CSV output path:

    a. Options > Preferences > Default Output Path.
    b. Browse to and select the folder you would like to use for exported results.
    c. Click Apply.
    d. Any queries that export CSV files will now be saved in this folder.

    NOTE: If you forget to set this path before you start the CSV files will be saved in **%AppData%\Microsoft\Log Parser Studio** by default but it is recommended that you move this to another location.

2. Tell LPS where the log files are by opening the Log File Manager. If you try to run a query before completing this step Log Parser studio will prompt and ask you to set the log path:

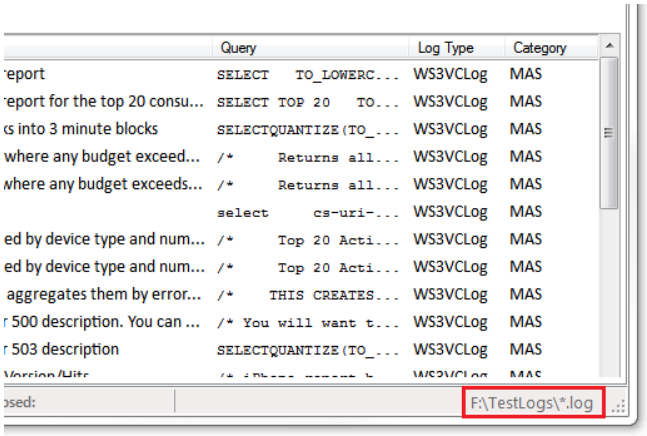




Upon clicking OK you are presented with the Log File Manager. Click Add Folder to add a folder or Add File to add a single or multiple files. *When when **adding a folder** you still must select at least one file so LPS will know*

which type of log we are working with. When doing so, LPS will automatically turn this into a wildcard (*.xxx) Indicating that all matching logs in the folder will be searched.

You can easily tell which folder or files are currently being searched by examining the status bar at the bottom-right of Log Parser Studio. To see the full path, roll your mouse over the status bar:



NOTE: LPS and LogParser handle multiple types of logs and queryable objects. It is important to remember that the type of log you are querying must match the query you are performing. In other words, when running a query that expects IIS logs, only IIS logs should be selected in the File Manager. Failure to do this (it's easy to forget) will result errors or unexpected behavior will be returned when running the query.

3. Choose a query from the library and run it:

    a. Click the library tab if it isn't already selected.
    b. Choose a query in the list and double-click it. This will open the query in its own tab.
    c. Click the Run Single Query button to execute the query:



Run Single Query Button

The query will begin doing its work in the background. Once the query has completed there are two possible outputs targets; the result grid in the top half of the query tab or a CSV file. Some queries return to the grid while other more memory intensive queries are saved to CSV.

As a general rule queries that may return very large result sets are probably best served going to a CSV file for further processing in Excel. Once you have the results there are many features for working with those results. See the rest of the manual for details.

NOTE: *Log Parser 2.2 is a 32Bit COM interface. This means that even on 64bit machines there is a 32 bit memory limit. Due to this limitation very large result sets can easily exceed the total memory LPS has at its disposal (slightly less than 2GB). It is much better to send very large results to CSV as this process bypasses having to load thousands and thousands of records into memory and goes directly to a file instead. For results that are less than 50,000 records the grid should be fine. Be sure to check out the "what is a good query" in the main help section.*

LOG PARSER STUDIO

Welcome To Log Parser Studio

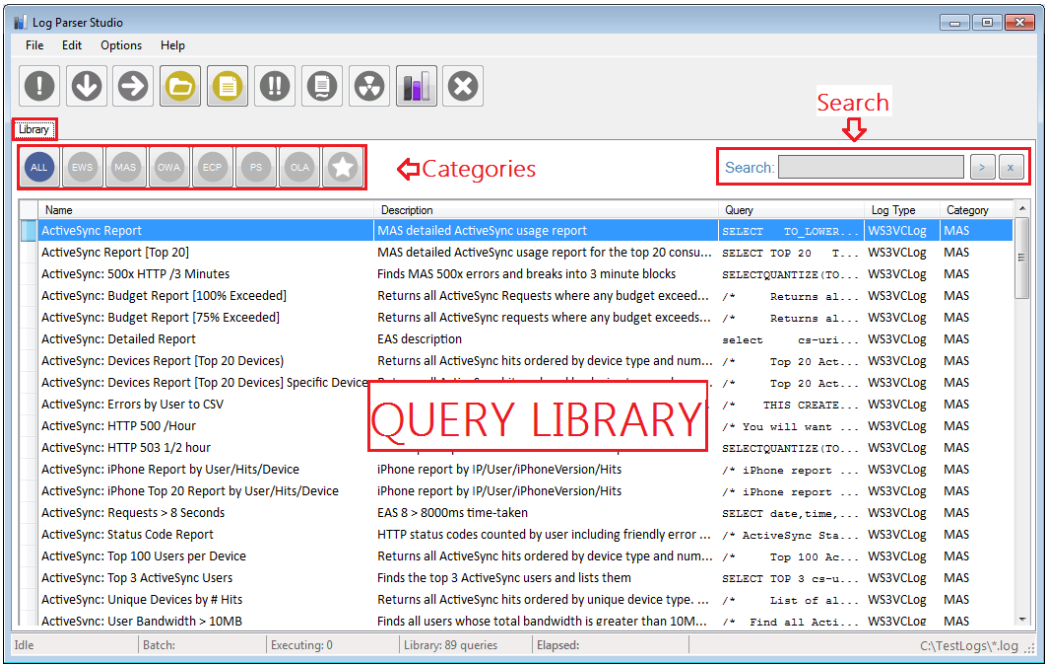
Anyone who regularly uses Log Parser 2.2 knows just how useful and powerful it can be for obtaining valuable information from IIS and other logs. In addition, adding the power of SQL allows explicit searching of gigabytes of logs returning only the data that is needed while filtering out the noise. The only thing missing is a great GUI to function as a front-end to Log Parser and a Query Library in order to manage all those great queries and scripts that one builds up over time.

Log Parser Studio was created to fulfill this need; by allowing those who use Log Parser 2.2 (and even those who don't due to lack of an interface) to work faster and more efficiently to get to the data they need with less "fiddling" with scripts and folders full of queries.

With Log Parser Studio (LPS for short) we can house all of our queries in a central location. We can edit and create new queries in the query editor and save them for later. We can search for queries using free text search as well as export and import both libraries and queries in different formats allowing for easy collaboration as well as storing multiple types of separate libraries.

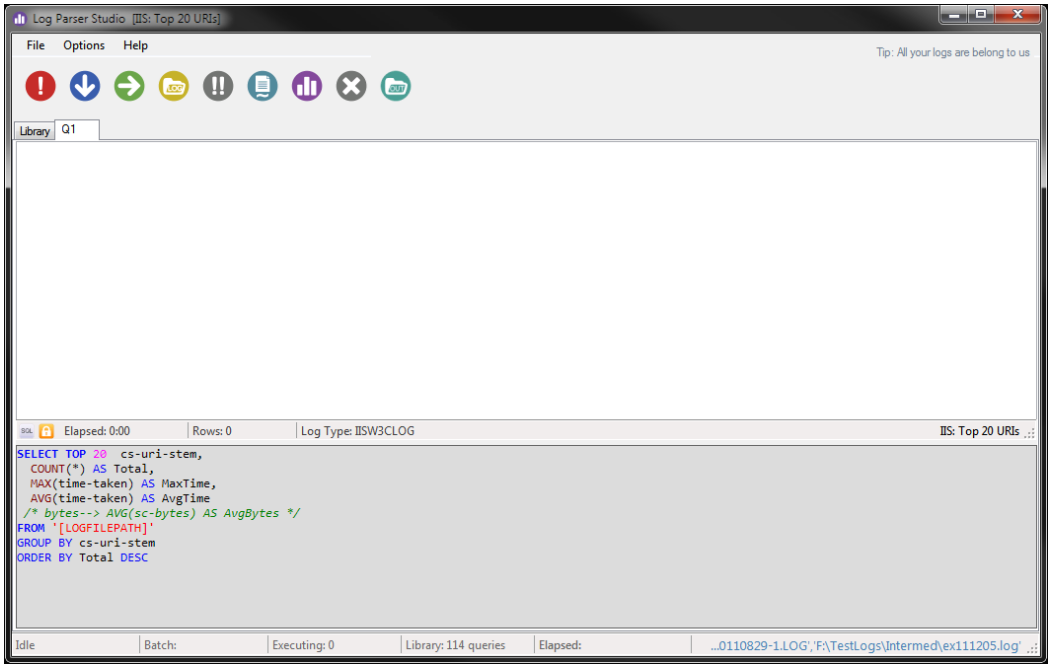
Query Library and Management

Upon launching Log Parser studio, the first thing you will see is the Query Library preloaded with queries. This is where we manage all of our queries:



The Library is always available by clicking on the Library tab as shown above (Ctrl+Q). You can load a query for review or execution using several methods. The easiest method is to simply select the query in the list and double-click it. Upon doing so the query will auto-open in its own "Query Tab" which can be seen below.

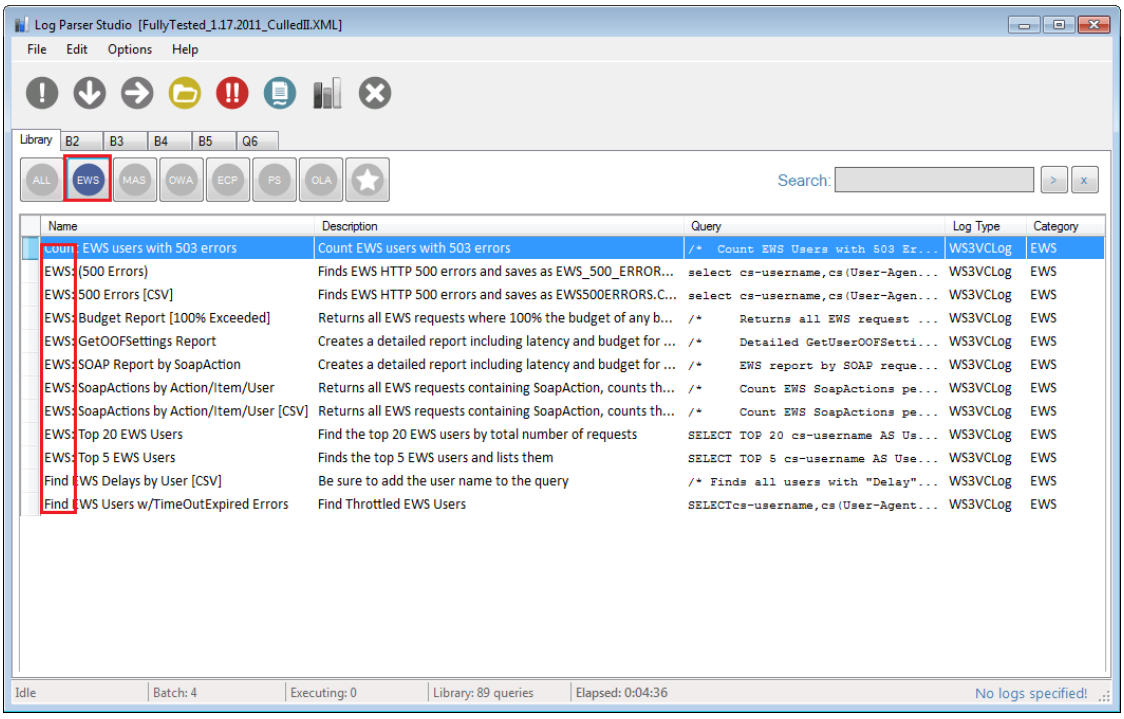
Query view after being double-clicked in the library



Query Library

The query library is home base for queries. All queries maintained by Log Parser Studio are stored in the Query Library.

Finding Queries
You can scroll to find the query that you need, but you can also use the category buttons to filter the results down into smaller categorical groupings. For example if you are only interested in queries concerning Exchange Web Services you can click the EWS filter button and only queries in the EWS category will be displayed:



Favorites Category

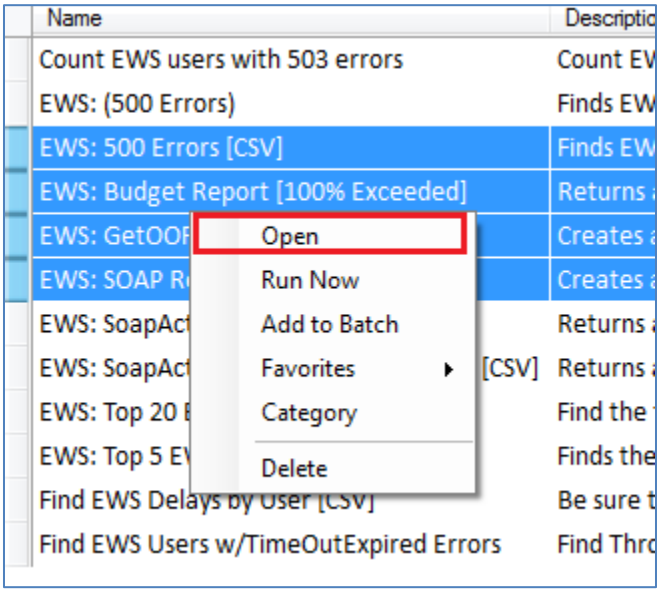
The favorite's button is located to the far right of the category filter buttons. Use this to add your most used queries or queries of interest for easy access later. To add/remove queries as favorites right-click the query and choose Favorites > Add/Remove and all selected queries will be added or removed from the favorites filter.

Query Search

Another method for finding queries quickly is the search feature. This feature searches the name and description of the entire library and returns any matches. To perform a search, enter part or all of the string you wish to search for and click the search button (>). To clear the search results click the clear search button (x).

Accessing Queries via Right-Click

Queries can be opened/accessed several ways. As previously mentioned in the quick start double-clicking any query will open it in its own query tab. Queries can also be opened by selecting, then right-clicking and choosing open from the context menu. Using this method multiple queries can be opened at once. For example to open four queries, select the queries, right-click > Open.



Additional Right-Click Options

Run Now

Immediately opens then runs the selected query. This is for convenience where the double-click method only opens the query but does not run it. Run now opens it and automatically begins execution for you without the extra step of clicking the Run Single Query button. However, most of the time it's a good idea to open and examine the query first if it isn't a query you have run previously.

Add To Batch

Adds the selected queries to the batch queue; batch jobs are discussed later in this document.

Favorites (add/remove)

Adds/removes the selected queries to/from the favorites filter.

Category

Changes the current category for selected queries.

Delete

Deletes all selected queries. You will be prompted to confirm the deletion.

Library Recovery

The initial library that ships with Log Parser Studio is embedded in the application and created upon install. The default location is %AppData%\Microsoft\Log Parser Studio\LibraryName.XML[k1]. If you ever delete, corrupt or lose the library you can easily reset back to the original by using the recover library feature (Options > Recover Library). When recovering the library **all existing queries will be deleted**. If you have non-default or modified queries that you do not want overwritten, you can export those first, then after the recovery merge them back in:

1. Use the Export feature to export the queries you wish to keep.
    a. Select the queries to export.
    b. Choose File > Export > Query(s) as XML.
    c. Give the exported file a name such as SavedQueries.XML

2. Recover the library.
    a. Choose Help > Recover Library.
    b. The existing library will be replaced with the default.

3. Merge the queries you exported in step 1 back into the library.
    a. Choose File > Import > XML To Library (**Merge**)
    b. Browse to and choose the previously exported file.
    c. The queries in the file will be merged back into the default Library

Importing and Exporting

The entire library or only subsets of the library can be imported and exported either as the default Log Parser Studio XML format or as SQL queries. For example if you have a folder full of Log Parser SQL queries, you can import some or all of them into Log Parser Studio's library. Usually, the only thing you will need to do after the import is make a few adjustments. All LPS needs is the base SQL query and to swap out the filename references with '[LOGFILEPATH]' and/or '[OUTFILEPATH]' as discussed previously.

Example
Below are two queries, the original Log Parser 2.2 SQL query as imported into LPS and the modified version to allow the query to execute properly now that it is in the Library:

Original
SELECT Count(*) AS Duration INTO SlowRequests.CSV FROM ex*.log WHERE time-taken > 10000

Modified
SELECT Count(*) AS Duration INTO '[OUTFILEPATH]\SlowRequests.CSV' FROM '[LOGFILEPATH]' WHERE time-taken > 10000

As you can see most existing queries will import and few changes should be needed. These changes need to be made manually. Log Parser Studio does not attempt to make these changes due to the wide variety of customized query parameters.

Importing an entire library as XML

Libraries can be imported and exported in their entirety when using the LPSXML format. This feature can be used to backup libraries as well as to collaborate with others and/or maintain separate libraries that have little or no relation to each other. For example a heavyweight user may wish to have separate libraries for separate applications, websites, servers etc. where the queries needed for each differ. In other words there could be an Exchange Server library, an IIS only library and so on. Or you can simply keep them all in a single library.

Import (Merge) XML
Imports an external library and merges the queries with the existing library. Duplicates are allowed. File > Import > XML to Library (Merge). Use this option to import a handful of queries that were previously exported from an existing library. This is the same option used earlier when restoring saved queries after a library recovery.

Import (Replace) XML
Imports an external library and replaces the existing library. This is as akin to switching between libraries. Duplicates are not allowed since the entire library is replaced with the imported queries.

Hint: When importing (Replace) the previous library is not actually overwritten. The new library is only copied into the application's %AppData% directory and LPS is pointed to the new library as its default. The replaced library still resides in this same directory and is left untouched. The only exception is when the imported library is the exact same name as the existing one.

If for some reason you needed to return to the previous library and don't have another copy of it, you can browse to %AppData%\Microsoft\Log Parser Studio and load the library you wish. If you are a power user it is possible this directory could eventually contain more than a few libraries. If this is a concern you can always delete or remove any libraries you no longer need.

Import SQL to Batch
Use this feature if you wish to import a number of SQL queries into a new batch job. There are some advantages to this that are not feasible with other methods. For example, let's say you have 25 queries that you regularly need to run as a batch job. However, you don't want to hunt for these and create the batch from scratch each and every time and you also don't want to modify the library using a normal import.

Import SQL to batch allows you to import LPSXML queries and assign them directly to a batch job, bypassing the library completely. You can then run the batch and work with the results without ever touching the library. It is not recommended to use this method for queries that have not previously executed successfully in Log Parser Studio. The best practice for this feature is to choose the queries you will need for batch jobs and export them from LPS. Later you can import the queries as a batch job and run it. Batch jobs are discussed in detail later in this document.

Save Library
Saves the existing library along with any changes made to any queries. Saving individual queries also save the main library but in some cases such as a delete operation the changes are not saved back to the main library as a precaution. Whenever this is the case you will be prompted when closing and/or you can choose File > Save Library.

Reload Library
Reloads the active library from disk. For example if you have made changes to several queries but have not yet saved them, you can use File > Reload Library to pull the unmodified changes from disk and reset all the changes. Keep in mind that File > Save Query also saves the active query AND the library meaning an edited query that has been saved will not be reverted to the original by using the Reload Library feature.

XML vs. SQL
The format you choose when importing/exporting is completely up to you. The XML format is the same that Log Parser Studio uses internally and SQL is what Log Parser 2.2 uses. These format options exist to allow users to export and import freely and to promote collaboration. Did you write a query that will restore world peace? Export it and share it with your friends and colleagues!

Queries

In this section we will discuss how we can work with queries in LPS as well as editing, creating, executing etc. Remember that a great query makes all the difference between a successful query that returns the concise information you need vs. a subpar query which taxes your system, returns much more information than you actually need and in some cases crashes the application.

The art of creating great SQL/Log Parser queries is outside the scope of this document, however all of the queries included with Log Parser Studio have been written to achieve the most concise results while returning the fewest records. Knowing what you want and how to get it with the least number of rows returned is key. For example we would never want to run a query such as the following:

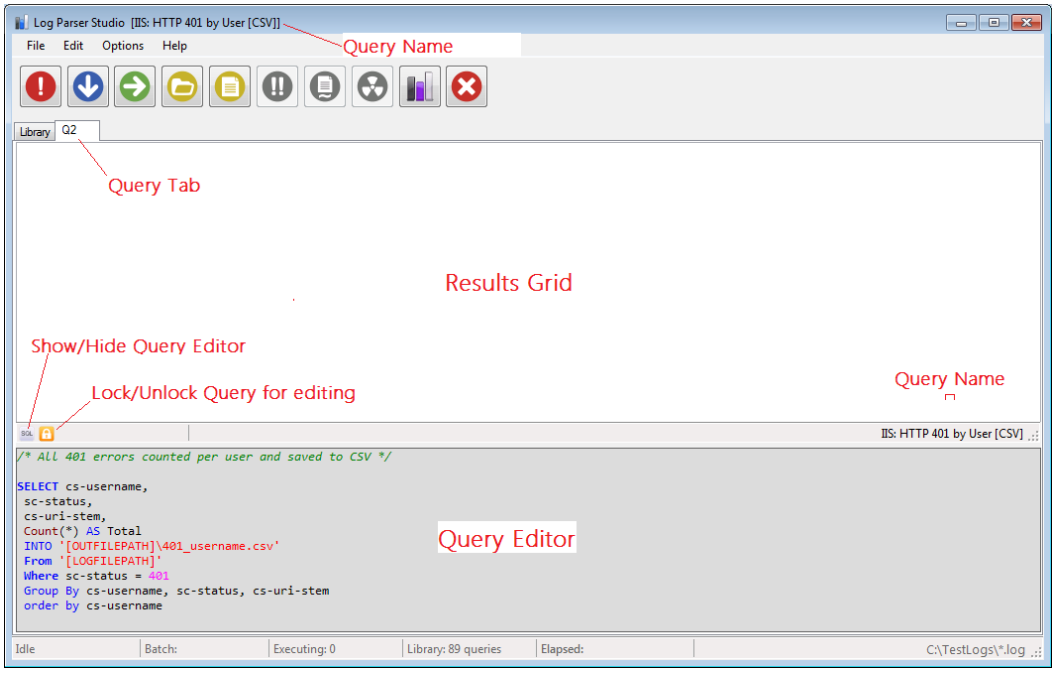
```
SELECT * FROM '[LOGFILEPATH]'
```

The above query will return every single row and every single column as-is from the logs being processed. This is really nothing more than a very inefficient file copy. You may at first think *"I need to search the IIS log but it won't open in my text editor due to how large it is"*.

The same problem above will exist in almost any application (if we try to load a 2GB file for example) and this is precisely why SQL, Log Parser 2.2 and Log Parser Studio are our friends. They allow us to greatly narrow down the results in very large files by searching for exactly what we want and returning only what we need.

Query Tabs

Double-click any query to open. Once a query is opened in Log Parser Studio a Query Tab is generated which is then populated by the query from the library. This Query Tab is your workspace for the active query. From this tab the query can be executed as-is or modified and tweaked. The Query Tab is divided into two main sections. The Results Grid and the query editor:

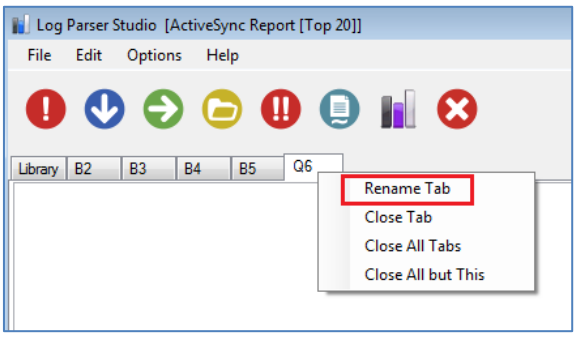


You can have as many tabs open as you wish with system resources being the only limit. Query tabs are always automatically named Q1, Q2, Q3 and so on and can be renamed at any time. Batch queries which will be discussed later are named B1, B2, B3 by default in order to distinguish between the two and can also be renamed.

Note: Once a query has been loaded into the query editor it is only virtual copy of the original query. Thusly, any changes made will not affect the original query unless you then save the query using the same name. The equivalent of Save-As is to save the query giving it a new name when saving.

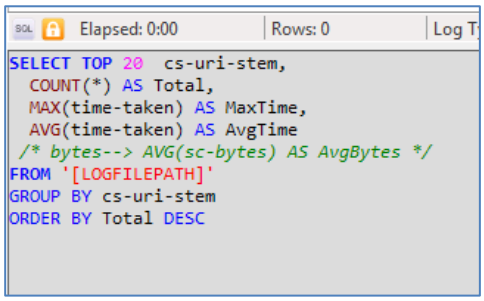
Renaming Tabs

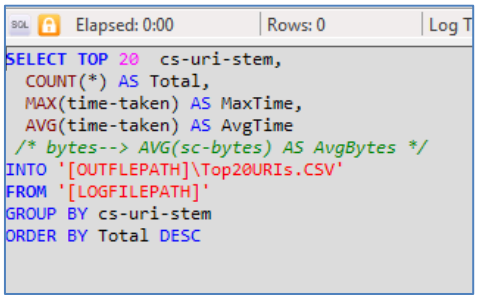
When running queries you are likely to end up with multiple queries and tabs containing results. By renaming tabs you can separate queries of interest for easy access. To rename a tab, right-click the tab and enter the new name in the rename dialog that pops up. There are additional options when right-clicking for closing tabs. You can also close any tab by double-clicking the tab.



Editing Queries

The query is always displayed in the bottom half of the Query Tab Window. You can edit any query anytime once it is unlocked. Not only can you edit existing queries but you can create your own from scratch by pressing CTRL+N or File > New Query. In the existing query below, notice the file tokens that are present in each query: '[LOGFILEPATH]' and/or '[OUTFILEPATH]'.





These tokens are automatically replaced with the real file names when the query is executed which is why we need to set these paths beforehand in the Log File Manager. By using tokens in this manner we can keep the queries clean and consistent. To change the logs being queried, change the path using the Log File Manager instead of the query itself and LPS will take care of the rest for you.

[OUTFILEPATH] is only used when the query is exporting results to a CSV file and is always part of an **INTO** statement If you wish to send the query to the grid instead, just remove this entire line.

[LOGFILEPATH] is always the log or logs being processed and always follows the **FROM** statement. This is always present and must always be properly set when querying physical log files. The only exception is, this token does not need to be present for queries that don't query logs directly such as the registry and computer event logs.

For example the first query below is how it might appear in Log Parser Studio, the second is how it might look right before it is executed and the two tokens have been replaced with the proper real filenames on the fly:

Before: SELECT Count(*) INTO **'[OUTFILEPATH]'** FROM **'[LOGFILEPATH]'** WHERE sc-status = 500

After: SELECT Count(*) INTO **'C:\CSV\Results.CSV'** FROM **'C:\TestLogs\123456.log'** WHERE sc-status = 500

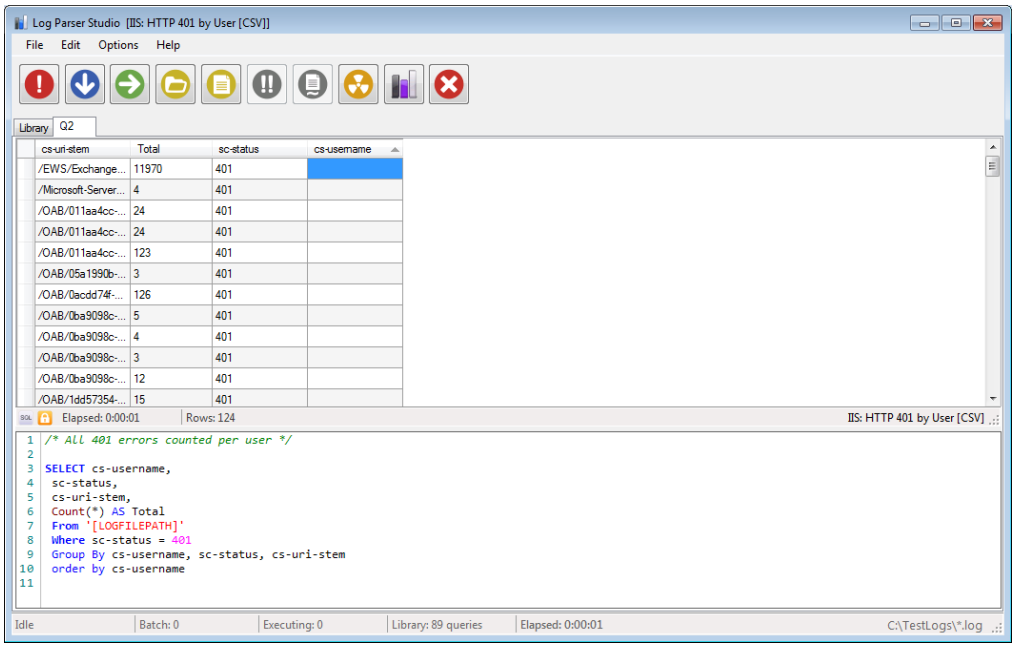
Note: Both '[LOGFILEPATH]' and '[OUTFILEPATH]' **MUST** be surrounded by brackets and single quotes!

*NOTE: Existing queries are locked (read-only) by default. This prevents accidental modification of an existing query that you have opened. However, many users are quite savvy at modifying and creating their own queries. Thusly, any query can be unlocked by clicking the orange lock at the top-left of the query editor: In addition, the [SQL] button to the left of the lock collapses the query window for convenience:

Once any changes have been made the query is now considered to be in a dirty state. Meaning that unless the query is saved these changes will not be written back to the library. Closing a query tab that has a query in a dirty state will cause a prompt to be shown asking if you would like to save the changes back to the library. If you want to save the query but you don't want to overwrite the original, then change the name of the query when saving.

Running Queries

Once you are happy with the query or are running a default query all that is needed is to click the Run Single Query button. At this time the query is prepared and sent to Log Parser 2.2 on its own background thread. This means that you can continue to work with Log Parser Studio and the results of the query will automatically return once the query has completed.



The results of the query allows two possible output types. The first is the result grid in the query tab and the second is a CSV file. If the query doesn't export its results to a CSV file then they will appear in the grid. If for some reason the query fails an error prompt will display with the known details of the failure.

About CSV Output

Any query can be sent to CSV instead of the grid. All that is required is to include the CSV output file token in the query accompanied by the INTO SQL statement. Below are two identical queries with the only difference being the first sends the results to the results grid and the second sends the results to a CSV file:

Results to grid:

SELECT Count(*) **cs-username FROM '[LOGFILEPATH]' WHERE cs-username IS NOT NULL**

Results to CSV:

SELECT Count(*) **cs-username INTO '[OUTFILEPATH]\UsersCount.CSV' FROM '[LOGFILEPATH]' WHERE cs-username IS NOT NULL**

CSV Output Options

There are user configurable options that affect how the CSV output behaves. There are three settings in preferences (Options > Preferences) that control this behavior:

Default CSV Output Path – As discussed this is the folder where all CSV files are saved by default and is user configurable. The default location is the user's %AppData% folder but it is recommended that this be changed to a common folder dedicated to CSV output.

Append Timestamp to CSV files – Since a single query always uses the same CSV filename, enabling this options allows running the query multiple times without overwriting the previous CSV file and will append the date time to the end of the filename in the following format: FILENAME_MM_DD_YYYY_HH.MMAM/PM_MS.CSV - Example: UserCount_1_24_2012_4.14PM_687.CSV which translates to 1/24/2012 @ 4:14 PM + 687 milliseconds.

Open CSV file on Query Completion – This option will cause the query to be automatically opened as soon as it completes using the default program that handles CSV files on the computer. This is where the append timestamp option really comes in handy. If you are running a query several times and auto-opening, an error well be returned on the second run if the first result is still opened. This is because the file may be locked in the other editor. Enabling Append Timestamp to CSV helps avoid this problem.

Other Query Window Elements

Query Timers
Each query begins its own time-elapsed counter as the query is being sent to LogParser 2.2. This provides quick visual feedback as to how long any query has been running.

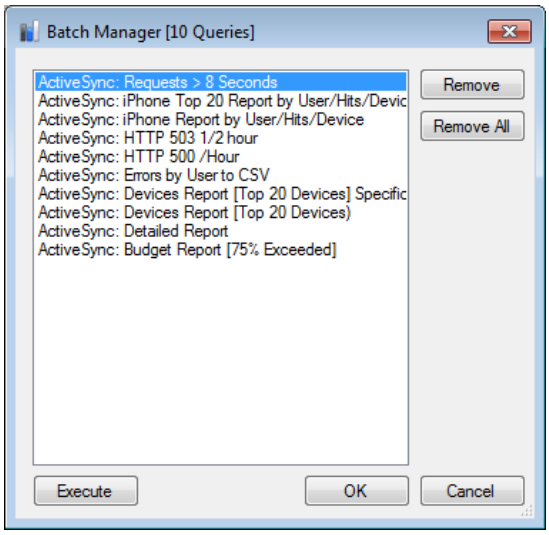
Query Rows
This counter lives just to the right of the Query timer and is blank until the query completes. Indicates how many rows were returned from the query when the results are being returned to the grid.

Query Name
Lives on the far right of the timer and row count. This is the name of the active/visible query. In addition the same information is appended to the title of the main Log Parser Studio window when the query is visible.

Batch Jobs & Multithreading

As you can see Log Parser Studio in combination with LogParser 2.2 is a very powerful tool. However, if all you could do was run a single query at a time and await the results you probably wouldn't be making near as much progress as you could be. In lieu of this LPS contains both batch jobs and multithreaded queries.

Batch Jobs
A batch job is simply a collection of predefined queries that can all be executed with the press of a single button.
To create a batch, select a query or queries and click CTRL+B. This will add the selected queries to the batch. You can also right-click the queries and choose "Add to Batch" in the query context menu. Once the queries are in the batch job you can view the list by clicking the Batch Manager Button which opens the Batch Manager Window:



From within the Batch Manager you can remove any single or all queries as well as execute them. You can also execute them by clicking the Run Multiple Queries button or the Execute button in the Batch Manager. Upon execution, Log Parser Studio will prepare and execute each query in the batch. By default LPS will send ALL queries to LogParser 2.2 at the soon as each is prepared. This is where multithreading works in our favor.

For example, if we have 50 queries setup as a batch job and execute the job, we'll have 50 threads in the background all working with LogParser simultaneously leaving the user free to work with other queries. As each job finishes the results are passed back to the grid or the CSV output based on the query type. Even in this scenario you can continue to work with other queries, search, modify and execute. As each query completes its thread is retired and its resources freed. These threads are managed very efficiently in the background so there should be no issue running multiple queries at once.

But what if we did want the queries in the batch to run concurrently for performance or other reasons? This functionality is already built into Log Parser Studio's options. Just make the change in Options > Preferences by checking the "Process Batch Queries in Sequence" checkbox. If this check box is checked, the first query in the batch is executed and the next query will not begin until the first one is complete. This process will continue until the last query in the batch has executed. It's doubtful you will need this option but it is there if you do.

You can also save batch jobs as XML. This is a similar format as the LPSXML library format and allows saving custom batch jobs for reuse without having to pick them all out of the library each time you want to run them. Otherwise a batch lasts from the time you created until you close LPS even if the batch has already run in the event you wish to run it twice. Also notice that batch query tabs by default are named B1, B2, B3, B4 etc. You can clear the batch manually by opening the batch manager and choosing "Remove all". Saving a batch job allows you to reload it at a later time even if you have closed LPS.

Automation

In conjunction with batches, Automation allows unattended scheduled automation of batches. For example we can create a scheduled task that will automatically run a chosen batch job which also operates on a separate set of custom folders. This process requires two components, a folder list file (.FLD) and a batch list file (.XML). We create these ahead of time from within Log Parser Studio:

1. Create the Batch job by selecting the queries from the library you wish to use add them to a batch job (CTRL+B) but don't execute the batch. Open the batch manager window via Alt+B or the Batch Manager Button. Export these as batchname.xml via File > Save Batch.. This will save the chosen queries as their own batch file. If you have a batch job already loaded, you can save it directly from the Batch manager window.

2. Create the log file/folder list by opening the file manager. Browse and choose any folders/files that you wish the batch to be run against. Choose File > Save from within the File manager window and save as a *.FLD file which is LPS file type for folder lists. (It's really just XML). You should now have an XML file with the queries and an .FLD file with the list of logs to be searched. For example Folders.FLD and Queries.XML

3. All that is needed is to pass both of these files to LPS. The order doesn't matter and switches aren't needed. Just both files separated by at least one space similar to the following

   LPS.EXE Folders.FLD Queries.XML

All that is left is to create a script or task that executes the above command. When the task runs, LPS.EXE will parse the folders and the queries, create a new batch then execute all of them automatically. The folders list and the queries are kept separate by design. This is so batches and target folders/logs can be used in different combinations if needed.

Charts and Charting

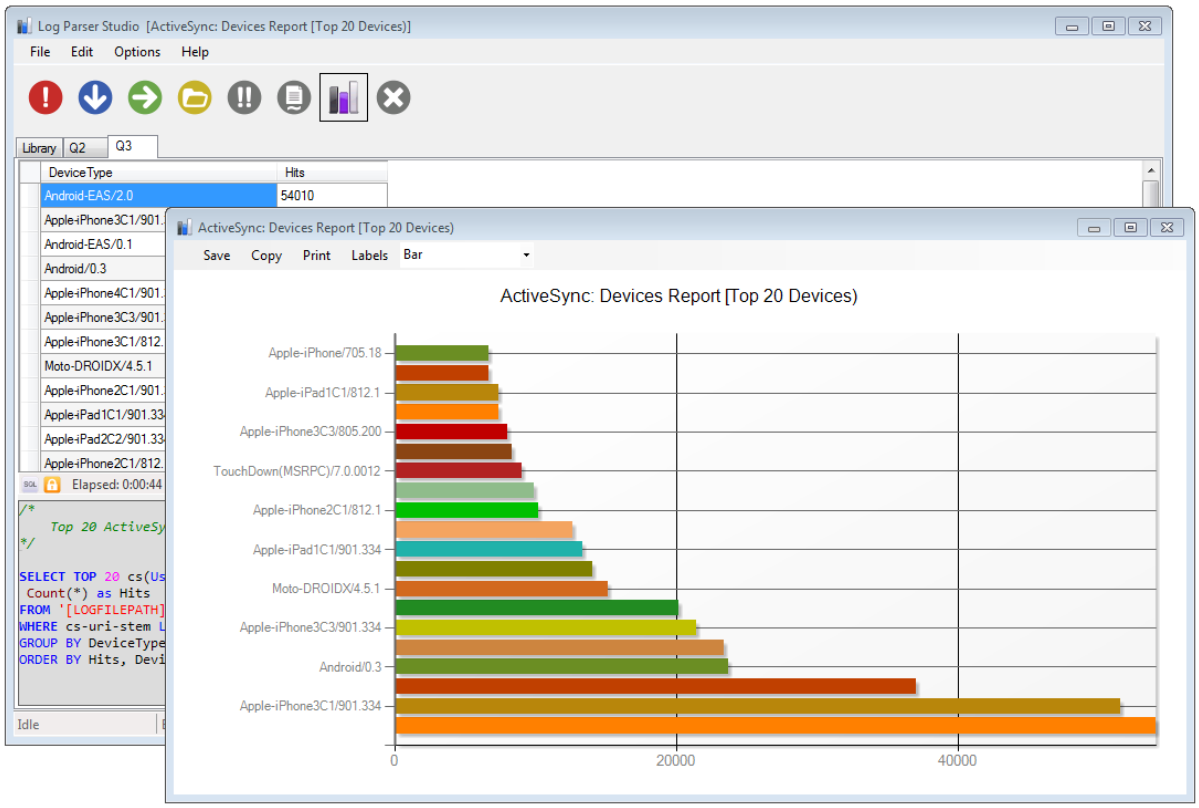
Many queries that return data to the Result Grid can be charted using the built-in charting feature. The basic requirements for charts are the same as LogParser 2.2:

1. The first column in the grid may be any data type (string, number etc.)
2. The second column must be some type of number (Integer, Double, Decimal). Strings are not allowed.

In many cases the chart is forgiving. For example if an attempt is made to build a chart with both columns one and two being text, the chart will open and simply show blank data. In other scenarios where incompatible data is sent to the chart a message box may appear stating that the data is incompatible with the chart. Keep the above requirements in mind when creating your own queries so that you will consciously write the query to include a number for column two. But wait there's more…

What if the query results comprise a text field followed but multiple numeric fields? For example Username, hits, error count, latency, time-taken and would like to chart any of these numeric fields? We are in luck because all result grids allow rearranging of their columns. This means we can drag columns around and create different charts based on the data we need. Using our scenario we can swap the time-taken column with the hits column (hits was column two) and our chart now shows time-taken instead of hits. Remembering our rules above any numeric column we drag into the column two positions is our new chart provided it is numeric.

To generate a chart click the chart button after a query has completed. Before clicking chart confirm column two is numeric. If the numeric column you care about is not in the second position, just drag it over then click the chart button. Below is a chart generated from a query that counts ActiveSync devices by hits:



When the result set contains a large number of rows the chart cannot display every row properly. However, we can click, drag, and zoom in any record. If the chart is at any zoom level other than 100% scroll bars will appear. The chart also has a tool tip that will display the X axis data point when moused over.

Chart Title
The title of the chart is always the same as the name of the query that was executed.

Save Chart
The chart can be saved in most popular image formats (TIFF, PNG, JPG, GIF, EMF and more). Click save and choose the format.

Print Chart
The chart can use any printer or print driver installed on your system including PDF/XPS provided you already have a driver that prints to these formats

Copy Chart
Copies a bitmap of the chart to the Windows clipboard for easy pasting into emails or other documents.

Labels
Toggles visibility of labels at the top of each data point.

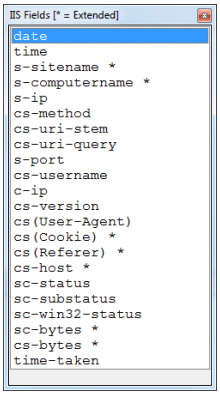
Chart Type
Chart types are selectable in the Chart Type dropdown menu. For most uses the bar chart works best since it displays the Y axis in the most readable form. If the chosen chart is not compatible an error message will appear to try a different chart type.

The chart is also resizable up to the maximized position and this size change is also reflected when exporting the chart

Keyboard Shortcuts/Commands

There are multiple keyboard shortcuts built-in to Log Parser Studio. You can view the list anytime while using LPS by clicking Help > Keyboard Shortcuts. The currently included shortcuts are as follows:

CTRL+N	Start a new query.
CTRL+S	Save active query in library or query tab depending on which has focus.
CTRL+Q	Open library window.
CTRL+B	Add selected query in library to batch.
ALT+B	Open Batch Manager.
CTRL+B	Add the selected queries to batch.
CTRL+ALT+E	Open the error log if one exists.
CTRL+E	Export current selected query results to CSV.
ALT+F	Add selected query in library to the favorites list.
CTRL+ALT+L	Open the raw Library in the first available text editor.
CTRL+F5	Reload the Library from disk.
F5	Execute active query.
F2	Edit name/description of currently selected query in the Library.
F3	Display the list of IIS fields* →



* When writing and modifying queries this option (F3) can be used as a quick reference for the actual field names of IIS fields without having to refer to IIS or an actual log file. Additionally, there are some hidden features in this list; double-click any single item to place the field on the clipboard. If multiple items are highlighted, press enter to copy them all to the clipboard so they can be easily pasted into the query you are currently working on. Note that this is only available when a query is active. In other words the field list will *not* appear when only the library is visible.

Supported Input and Output types

Log Parser 2.2 has the ability to query multiple types of logs. Since Log Parser Studio is a work in progress, only the most used types are currently available. Additional input and output types will be added when possible in upcoming versions or updates.

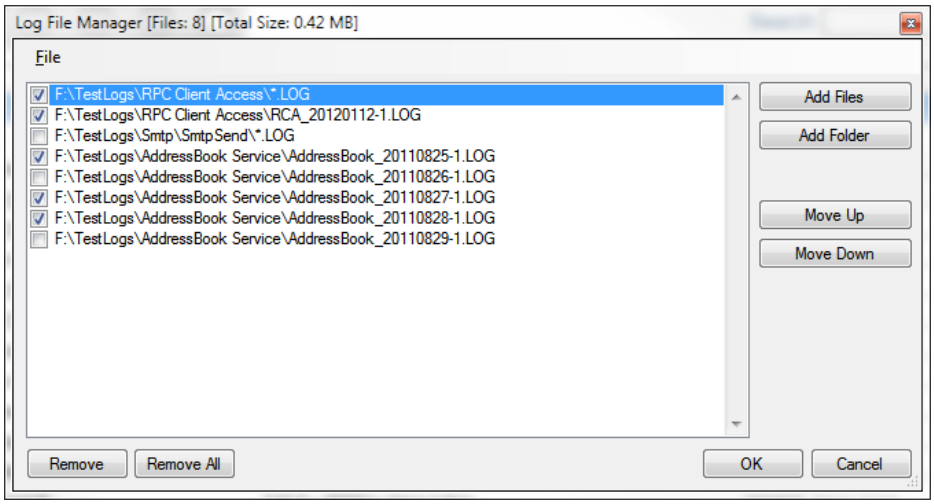
Currently supported input types

Full support for W3SVC/IIS, CSV, HTTP Error and basic support for all built-in LogParser 2.2 input formats. In addition some custom written Log Parser Studio formats such as Microsoft Exchange specific formats not available with the default LogParser 2.2 install.

Currently supported output types

CSV and TXT are the currently supported output file types.

Log File Manager



The log file manager contains multiple features for working with log files. To add log files choose add file or add folder and select the folder or files you wish to include. If you are adding a folder you must select at least one file so that LPS knows which type of file it is working with. In some cases you may have multiple folders inside a master folder and you need to select the master folder but it contains no logs. You can bypass this problem by typing “dummy.log” (without the quotes) in the folder dialog window and click open. *Dummy.log does not actually need to exist; we are only informing log parser what folder to use and what file type we expect.*

When the file paths have been loaded into the Log File Manager it will automatically attempt to calculate how many logs were found and their total file size. This is helpful so that you are aware of the files/sizes you will be querying. You can also disable this behavior in Options > Preferences.

Enabling and Disabling Logs

Logs can be enabled/disabled from being queried even though they are in the Log File Manager list. This is a very handy feature when you want to load up logs from two different servers and run two instances of the same query, one against each log. Or say you want to load up several log types and only enable them as you switch between queries that need them. To enable/disable log files just check or uncheck the check box to the left of the file. By using this method you can get more work done faster without having to constantly load and unload files you wish to query. Anything unchecked will not be queried. Log Parser Studio remembers these choices for you and will reload them the next time your run LPS!

NOTE: If you are running any queries that do not need physical files such as those that query the registry or event logs and so on, the Log File Manager is completely bypassed and not used. Meaning that you don’t have to make any changes to the logs being searched if you are executing a query that doesn’t require physical logs.

Custom Log Types

LPS contains a number of custom written log types that aren’t available outside of Log Parser Studio. These are built in and operate on various log types such as Exchange 2010 SMTPSend, SMTPReceive, Exchange RPC Client Access logs, Exchange 2010 Address book logs and more.

Troubleshooting

Much work has been done to reduce issues that may arise when running queries. However, Log Parser Studio and LogParser 2.2 are only as stable as the queries they are executing. Since all queries are 100% editable, errors or anomalies in a query can in turn cause either LPS or LogParser 2.2 to return an error. Whenever you get an error that appears to be unexplainable try the following steps first:

1. Check the log file(s) path via File/Folder manager to insure the logs you wish to query actually exist and that you have permissions to access them. If LogParser can't find or read the files, it cannot query them.

2. Run a known "good" query against a known good log that you can use for troubleshooting. If this query is successful then we know the query engine is working and the query throwing the error probably has a mistake or unknown issue.

3. Make sure the query matches the log type selected in the Log File Manager. Log Parser Studio can query multiple log types. For example if you attempt to run an IIS query against SMTP logs you will receive an error or wildly inaccurate results. You can either change the files to the correct type or if the query is of the wrong type, select the query in the library and press F2. From the dialog that appears you can modify query metadata (name, description etc.) as well as the default log type. However, the log type set in the query must match the actual query itself. It is very easy to be running various queries and not realize that you have suddenly loaded a query or logs that mismatch. If you get unexpected errors when running a query, consider this as a possible culprit.

   When LogParser 2.2 evaluates queries it does so from top to bottom. If an error is returned, the error message usually gives a clue as to the exact point where the error occurred. For example, if you have a query and you notice the error message mentions the ORDER BY clause, chances are everything preceding up to the last line before the ORDER BY clause likely passed inspection.

4. Log Parser Studio keeps an error log of most errors. This log lists any queries passed to LogParser 2.2 that failed and resulted in an error. The format of the query in this log is identical to what LogParser 2.2 attempted to process which may give clues to the problem that may not be seen in the query window. For example the log file will show the literal path of the log file(s) being queried instead of the log file token. To open the log press CTRL+ALT+E.

5. Log Parser Studio needs a healthy library. If it cannot load the library for some reason such as it is corrupt or missing you will be prompted to create a new blank library or cancel and import an existing known good library. You can also, choose cancel then Help > Recover Library and reinstall the default installation library. Remember that recovering the library normally overwrites any previous queries but in this case LPS can't even load the default library so there is nothing to overwrite.

About the Queries

These queries are as-is and not warranted for accuracy, performance or any other expected or unexpected result in any way. These are strictly "use at your own risk" queries.