

# Kusto.Explorer installation and user interface

Article • 03/20/2023

Kusto.Explorer is a rich desktop application that enables you to explore your data using the Kusto Query Language in an easy-to-use user interface. This overview explains how to get started with setting up your Kusto.Explorer and explains the user interface you will use.

With Kusto.Explorer, you can:

- Query your data.
- Search your data across tables.
- Visualize your data in a wide variety of graphs.
- Share queries and results by email or using deep links.

## Installing Kusto.Explorer

- Download and install the Kusto.Explorer tool from:
  - <https://aka.ms/ke> (CDN location)
  - <https://aka.ms/ke-mirror> (Non-CDN location)
- Instead, access your Kusto cluster with your browser at: `https://<your_cluster>.<region>.kusto.windows.net`. Replace <your\_cluster> and <region> with your Azure Data Explorer cluster name and deployment region.

## Using Google Chrome and Kusto.Explorer

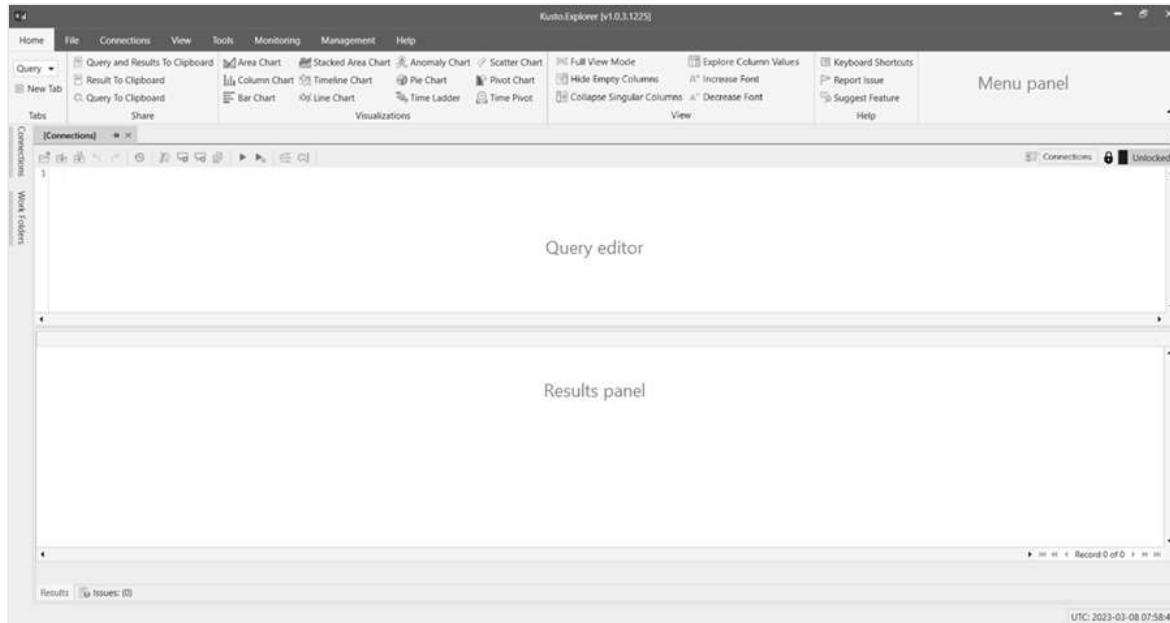
If Google Chrome is your default browser, installing the ClickOnce extension for Chrome is required. If your default browser is the Chromium-based Microsoft Edge, installing this extension *isn't* required.

<https://chrome.google.com/webstore/detail/clickonce-for-google-chro/kekahkplibinaibelipdcikofmedafmb/related?hl=en-US>

## Overview of the user interface

The Kusto.Explorer user interface is designed with a layout based on tabs and panels, similar to that of other Microsoft products:

- Navigate through the tabs on the menu panel to perform various operations
- Manage your connections in the connections panel
- Create scripts to run in the script panel
- View the results of the scripts in the results panel



## Connections panel

The Connections panel is a tree view showing the structure of configured cluster connections. The left sidebar has tabs for Connections and Work Folders. The main area shows a tree structure under the "Help" connection:
 

- Help
  - ContosoSales
  - FindMyPartner
  - SampleIoTData
  - SampleLogs
  - SampleMetrics
  - Samples
  - Trender

 A search bar at the top of the tree view allows filtering by name.

The Connections pane shows all the configured cluster connections. For each cluster the databases, tables, and attributes (columns) that they store are shown. Select items (which sets an implicit context for the search/query in the main panel), or double-click items to copy the name to the search/query panel.

If the actual schema is large (such as a database with hundreds of tables), you can search it by pressing **CTRL+F** and entering a substring (case-insensitive) of the entity name you're looking for.

Kusto.Explorer supports controlling the Connection panel from the query window, which is useful for scripts. For example, you can start a script file with a command that instructs Kusto.Explorer to connect to the cluster/database whose data is being queried by the script, by using the following syntax:

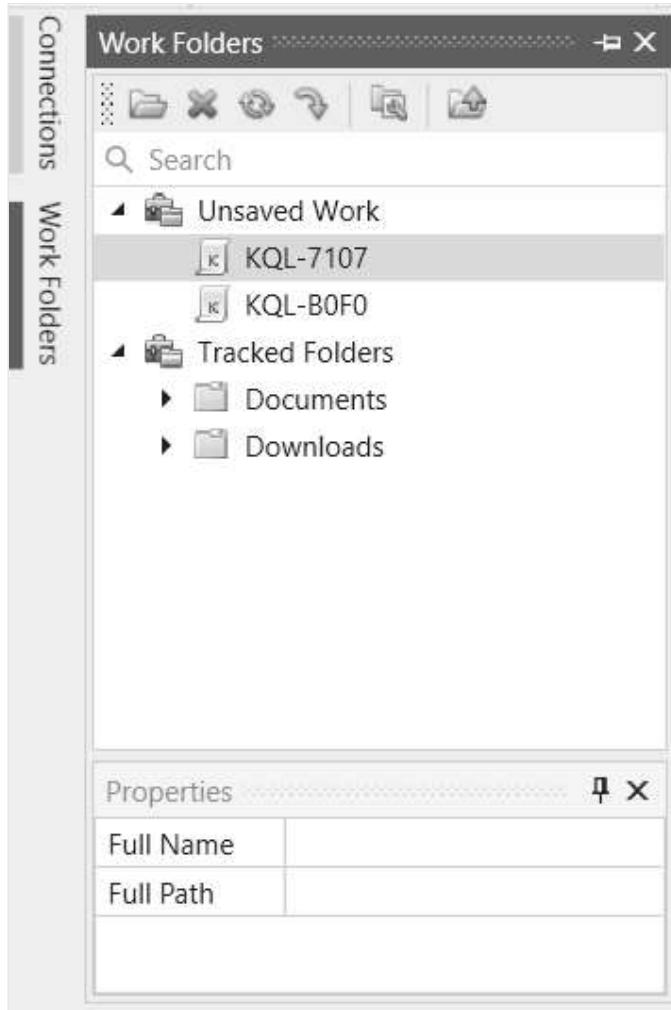
```
Kusto

#connect cluster('help').database('Samples')

StormEvents | count
```

Run each line using **F5**, or similar.

## Work Folders panel



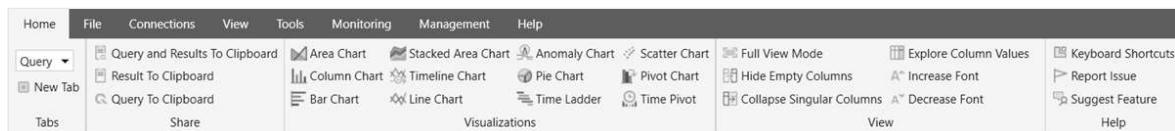
The Work folders pane organizes your work folders in one place to make navigating your work easier. There are two types of work folders:

- **Unsaved work:** lists folders for open query tabs that you may still be working on.

- **Tracked folders:** lists folders from your local device that you can add as KQL libraries for easier access and management.

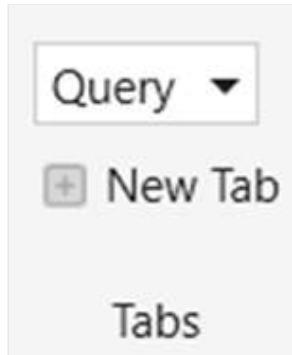
# Menu panel

## Home tab



The Home tab shows the most frequently used operations. It includes:

## Query section



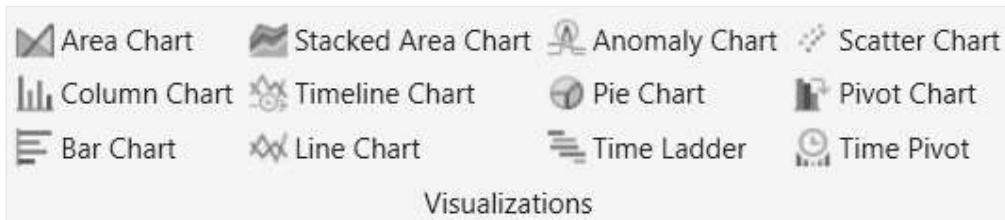
Menu	Behavior
Mode dropdown	<ul style="list-style-type: none"> <li>• Query mode: Switches the query editor into a query mode. Commands can be written and saved as queries (default)</li> <li>• Search mode: A single query mode where each command entered is processed immediately and presents a result in the result panel</li> <li>• Search++ mode: Allows searching for a term using search syntax across one or more tables. Learn more about using Search++ Mode</li> </ul>
New Tab	Opens a new tab for querying Kusto Query Language.

## Share section



Menu	Behavior
Data To Clipboard	Exports Query and data set to a clipboard. If a chart is presented, it exports the chart as bitmap
Result To Clipboard	Exports the data set to a clipboard. If a chart is presented, it exports the chart as bitmap
Query to Clipboard	Exports the Query to a clipboard

## Visualizations section

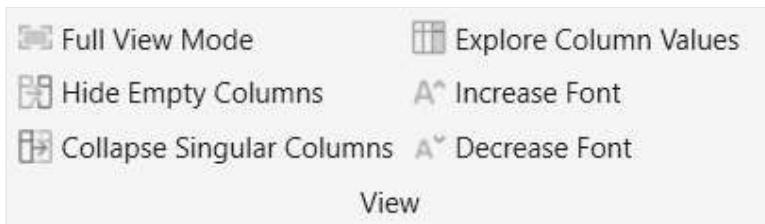


For variable visualizations, see the render operator.

Menu	Behavior
Area chart	Displays an area chart in which the X-axis is the first column (must be numeric). All numeric columns are mapped to different series (Y-axis).
Column Chart	Displays a column chart where all numeric columns are mapped to different series (Y-axis). The text column before numeric is the X-axis (can be controlled in the UI).
Bar Chart	Displays a bar chart where all numeric columns are mapped to different series (X-axis). The text column before numeric is the Y-axis (can be controlled in the UI).
Stacked Area Chart	Displays a stacked area chart in which the X-axis is the first column (must be numeric). All numeric columns are mapped to different series (Y-axis).
Time Chart	Displays a time chart in which the X-axis is the first column (must be datetime). All numeric columns are mapped to different series (Y-axis).
Line Chart	Displays a line chart in which the X-axis is the first column (must be numeric). All numeric columns are mapped to different series (Y-axis).
Anomaly Chart	Similar to Time Chart, but finds anomalies in time series data, using a machine learning anomaly detection algorithm. The data must be in fixed interval buckets. For anomaly detection, Kusto.Explorer uses the series_decompose_anomalies function.
Pie Chart	Displays a pie chart in which the color-axis is the first column. The theta-axis (must be a measure, converted to percent) is the second column.

Menu	Behavior
Time Ladder	Displays a ladder chart in which the X-axis is the last two columns (must be datetime). The Y-axis is a composite of the other columns.
Scatter Chart	Displays a point graph in which the X-axis is the first column (must be numeric). All numeric columns are mapped to different series (Y-axis).
Pivot Chart	Displays a pivot table and pivot chart that gives the full flexibility of selecting data, columns, rows, and various chart types.
Time Pivot	Interactive navigation over the events time-line (pivoting on time axis)

## View section



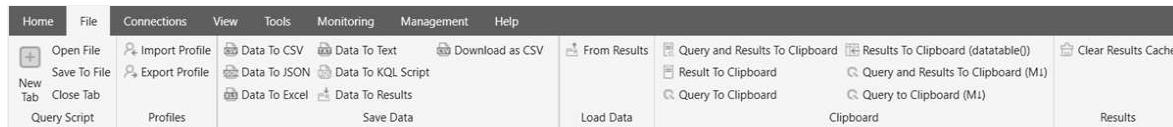
Menu	Behavior
Full View Mode	Maximizes the work space by hiding the ribbon menu and Connection Panel. Exit Full View Mode by selecting <b>Home &gt; Full View Mode</b> , or by pressing F11.
Hide Empty Columns	Removes empty columns from the data grid.
Collapse Singular Columns	Collapses columns with singular values.
Explore Column Values	Shows column values distribution
Increase Font	Increases the font size of the query tab and of the results data grid.
Decrease Font	Decreases the font size of the query tab and of the results data grid.

### ⓘ Note

Data View Settings:

Kusto.Explorer keeps track of what settings are used per unique set of columns. When columns are reordered or removed, the data view is saved and will be reused whenever the data with the same columns is retrieved. To reset the view to its defaults, in the View tab, select **Reset View**.

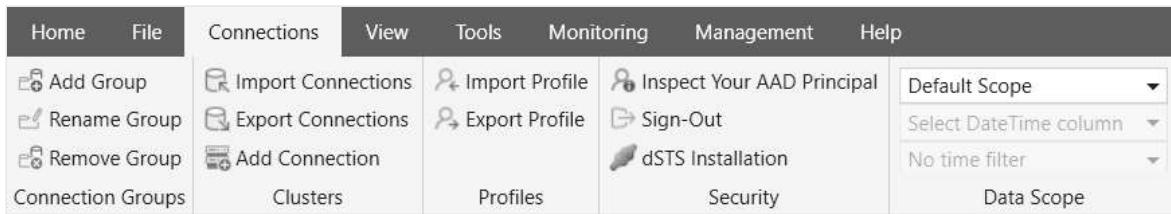
## File tab



Menu	Behavior
<i>-----Query Script-----</i>	
New Tab	Opens a new tab window for querying Kusto.
Open File	Loads data from a *.kql file to the active script panel.
Save To File	Saves the content of the active script panel to *.kql file.
Close Tab	Closes the current tab window
<i>-----Profiles-----</i>	
Import Profile	Import a Kusto.Explorer profile.
Export Profile	Export a Kusto.Explorer profile.
<i>-----Save Data-----</i>	
Data To CSV	Exports data to a CSV (comma-separated-values) file.
Data To JSON	Exports data to a JSON formatted file.
Data To Excel	Exports data to an XLSX (Excel) file.
Data To Text	Exports data to a TXT (text) file.
Data To KQL Script	Exports Query to a script file.
Data To Results	Exports Query and data to a Results (QRES) file.
Run Query Into CSV	Runs a query and saves the results to a local CSV file.
<i>-----Load Data-----</i>	
From Results	Loads Query and data from a Results (QRES) file.

Menu	Behavior
-----Clipboard-----	
Query and Results To Clipboard	Exports Query and data set to a clipboard. If a chart is presented, it exports the chart as a bitmap.
Result To Clipboard	Exports data set to a clipboard. If a chart is presented, it exports the chart as a bitmap.
Query To Clipboard	Exports the Query to a clipboard.
Results To Clipboard (datatable())	Exports query results to a clipboard. Maximal allowed cell value is 30000.
-----Results-----	
Clear results cache	Clears cached results of previously executed queries.

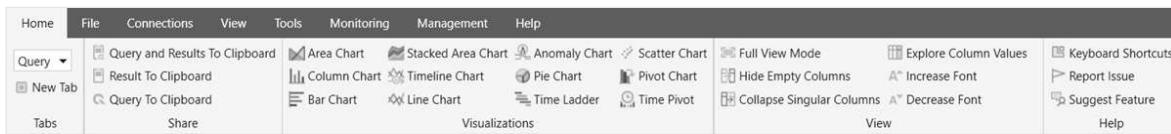
## Connections tab



Menu	Behavior
-----Groups-----	
Add Group	Adds a new Kusto Server group.
-----Clusters-----	
Import Connections	Imports connections from a file specifying connections.
Export Connections	Exports connections to a file.
Add Connection	Adds a new Kusto Server connection.
Edit Connection	Opens a dialog for Kusto Server connection properties editing.
Remove Connection	Removes the existing connection to Kusto Server.
Refresh	Refreshes properties of a Kusto server connection.
-----Profiles-----	
Import Profile	Import a Kusto.Explorer profile.

Menu	Behavior
Export Profile	Export a Kusto.Explorer profile.
	----- <i>Security</i> -----
Inspect Your ADD Principal	Shows currents active user details.
Sign-out	Signs-out the current user from the connection to AAD.
	----- <i>Data Scope</i> -----
Caching scope	<ul style="list-style-type: none"> <li>• Hot Data: Execute queries only on hot data cache</li> <li>• All Data: Execute queries on all available data (default).</li> </ul>
DateTime Column	Name of column which may be used for time pre-filter.
Time Filter	Value of time pre-filter.

## View tab



Menu	Behavior
	----- <i>Appearance</i> -----
Full View Mode	Maximizes the work space by hiding the ribbon menu and Connection Panel.
Increase Font	Increases the font size of the query tab and of the results data grid.
Decrease Font	Decreases the font size of the query tab and of the results data grid.
Reset Layout	Resets the layout of the tool's docking controls and windows.
Rename Document Tab	Rename the selected tab.
	----- <i>Data View</i> -----
Reset View	Resets data view settings to its defaults.
Explore Column Values	Shows column values distribution.

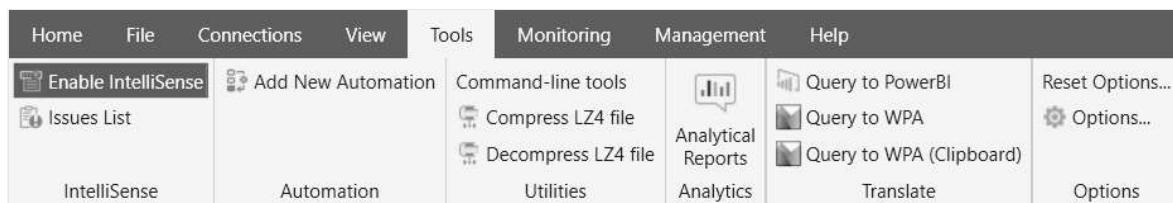
Menu	Behavior
Focus on Query Statistics	Changes the focus to query statistics instead of query results upon query completion.
Hide Duplicates	Toggles removal of the duplicate rows from the query results.
Hide Empty Columns	Toggles removal of empty columns from the query results.
Collapse Singular Columns	Toggles collapsing columns with singular value.
Row Selection	Enables selection of specific rows in the Results panel
Color By Column	Groups identical records in the first column by color.
Wrap Text	Formats cells to wrap the data to fit the column width.
----- <i>Data Filtering</i> -----	
Filter Rows In Search	Toggles the option to show only matching rows in query results search ( <b>Ctrl+F</b> ).
----- <i>Visualizations</i> -----	
Visualizations	See Visualizations, above.

### ⚠ Note

#### Data View Settings:

Kusto.Explorer keeps track of the settings used per unique set of columns. When columns are reordered or removed, the data view is saved and will be reused whenever the data with the same columns is retrieved. To reset the view to its defaults, in the **View** tab, select **Reset View**.

## Tools tab



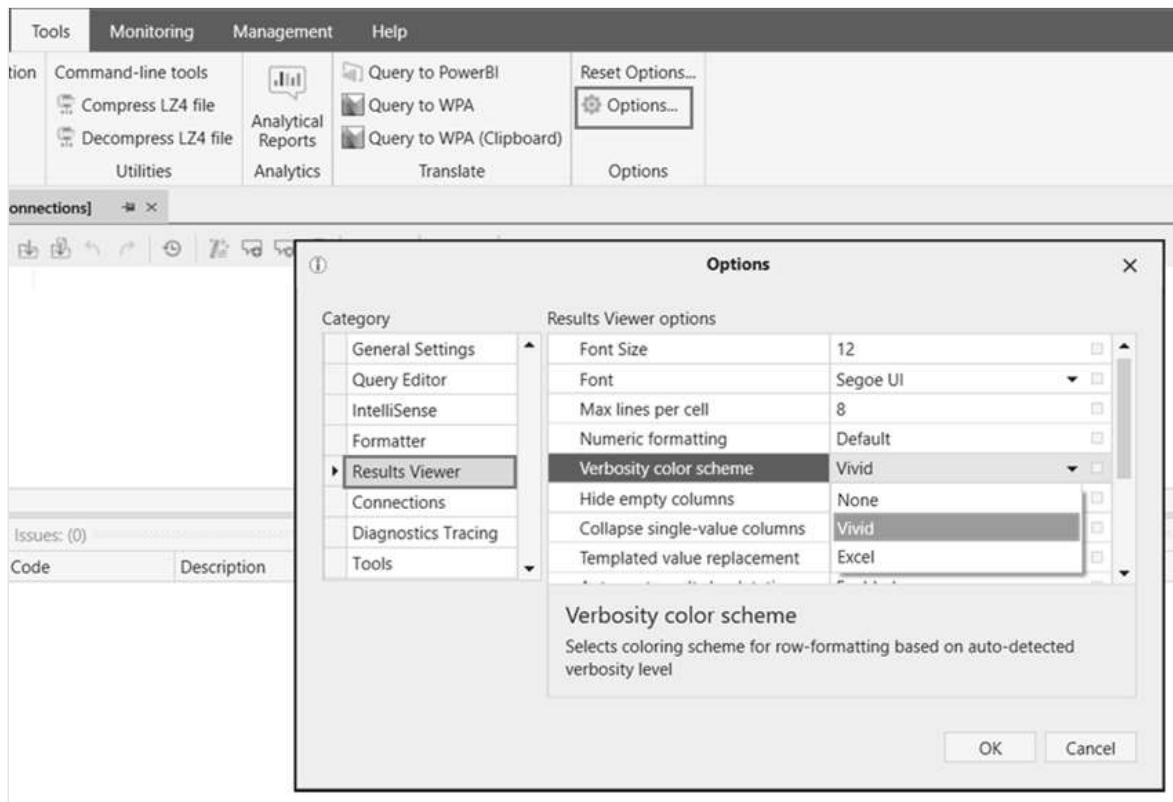
Menu	Behavior
----- <i>IntelliSense</i> -----	

Menu	Behavior
Enable IntelliSense	Enables and disables IntelliSense on the Script Panel.
Issues List	Lists issues in the Script panel.
	----- <i>Automation</i> -----
Add New Automation	Produces an analysis report that summarizes query results with additional insights.
	----- <i>Utilities</i> -----
Command-line tools	Opens the command prompt on your computer.
Compress LZ4 file	Compresses files using the LZ4 algorithm.
Decompress LZ4 file	Decompresses files using the LZ4 algorithm.
	----- <i>Analytics</i> -----
Analytical Reports	Opens a dashboard with multiple pre-built reports for data analysis.
	----- <i>Translate</i> -----
Query to Power BI	Translates a query to a format suitable for using in Power BI.
	----- <i>Options</i> -----
Reset Options	Sets application settings to default values.
Options	Opens a tool for configuring application settings. To learn more, see Kusto.Explorer options.

## Table row colors

Kusto.Explorer tries to interpret the severity or verbosity level of each row in the results panel and color them accordingly. It does this by matching the distinct values of each column with a set of known patterns ("Warning", "Error", and so on).

To modify the output color scheme, or turn this behavior off, from the **Tools** menu, select **Options > Results Viewer > Verbosity color scheme**.



Excel color scheme legend



Vivid color scheme legend

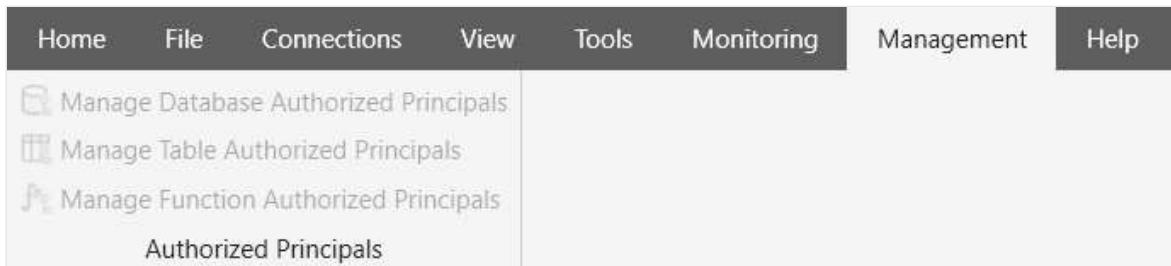


## Monitoring tab

The screenshot shows the Power BI ribbon with the 'Monitoring' tab selected. Below the ribbon, there are two cards: 'Latest Data: All tables' and 'Latest Data: Selected table'. At the bottom, there is a 'Monitor' button.

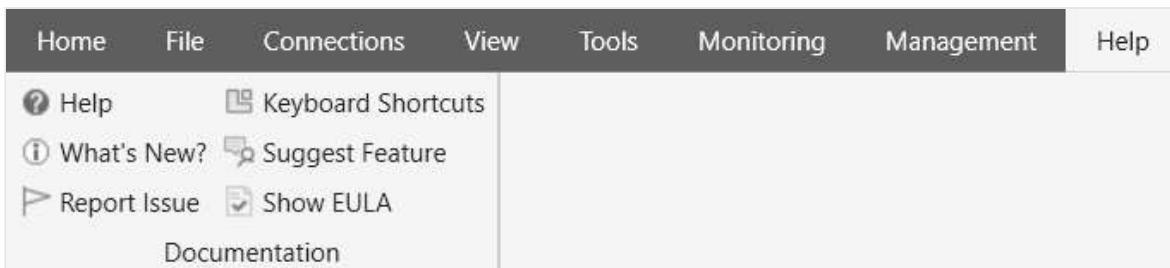
<b>Menu</b>	<b>Behavior</b>
----- <i>Monitor</i> -----	
Cluster Diagnostics	Shows a health summary for the Server Group currently selected in Connections Panel.
Latest data: All tables	Shows a summary of the latest data in all tables of the currently selected database.
Latest data: Selected table	Shows in the status bar the latest data in the selected table.
Cluster Health (Geneva)	Opens a link to view the state of your cluster.
Purge Metrics (Geneva)	Opens a link to view data purge metrics.

## Management tab



<b>Menu</b>	<b>Behavior</b>
----- <i>Authorized Principals</i> -----	
Manage Database Authorized Principals	Enables managing a database's principals for authorized users.
Manage Table Authorized Principals	Enables managing a table's principals for authorized users.
Manage Function Authorized Principals	Enables managing a function's principals for authorized users.

## Help tab



Menu	Behavior
<i>-----Documentation-----</i>	
Help	Opens a link to the Kusto online documentation.
What's new	Opens a document that lists all Kusto.Explorer changes.
Report Issue	Opens a dialog with two options: <ul style="list-style-type: none"><li>• Report issues related to service</li><li>• Report issues in the client application</li></ul>
Keyboard Shortcuts	Opens a link to the list of Kusto.Explorer keyboard shortcuts.
Suggest Feature	Opens a link to the Kusto feedback forum.
Show EULA	Opens a link to the Microsoft Azure Legal Information article.

## Control the user identity connecting to Kusto.Explorer

The default security model for new connections is AAD-Federated security.

Authentication is done through the Azure Active Directory using the default AAD user experience.

If you need finer control over the authentication parameters, you can expand the "Advanced: Connection Strings" edit box and provide a valid Kusto connection string value.

For example, users with a presence in multiple AAD tenants sometimes need to use a particular "projection" of their identities to a specific AAD tenant. Do this by providing a connection string, such as the one below (replace words IN CAPITALS with specific values):

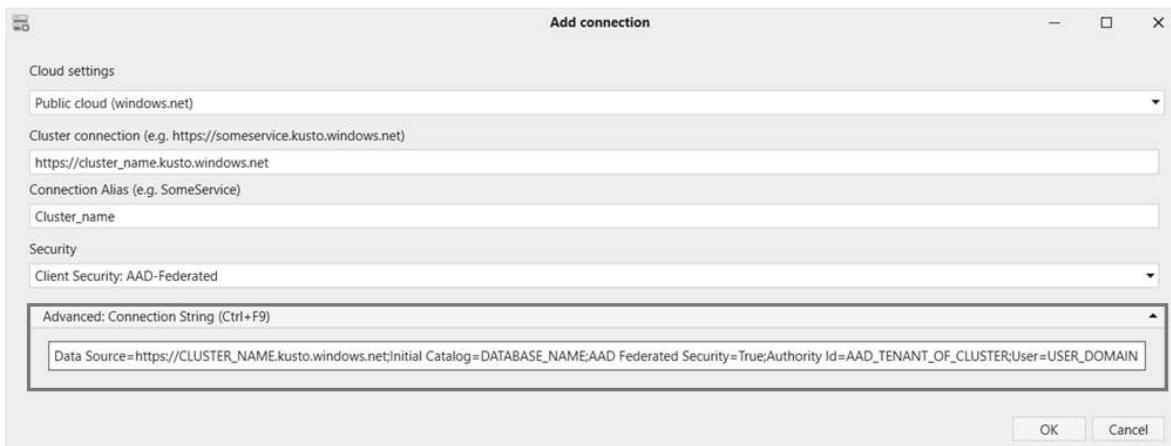
```
Kusto

Data Source=https://CLUSTER_NAME.kusto.windows.net;Initial Catalog=DATABASE_NAME;AAD Federated Security=True;Authority Id=AAD_TENANT_OF_CLUSTER;User=USER_DOMAIN
```

- `AAD_TENANT_OF_CLUSTER` is a domain name or AAD tenant ID (a GUID) of the AAD tenant in which the cluster is hosted. This is usually the domain name of the organization that owns the cluster, such as `contoso.com`.
- `USER_DOMAIN` is the identity of the user invited into that tenant (for example, `user@example.com`).

### ⓘ Note

The domain name of the user is not necessarily the same as that of the tenant hosting the cluster.



## Next steps

To learn more about working with Kusto.Explorer, see:

- Using Kusto.Explorer
- Kusto.Explorer keyboard shortcuts
- Kusto.Explorer options
- Troubleshooting Kusto.Explorer

To learn more about Kusto.Explorer tools and utilities, see:

- Kusto.Explorer code features
- Kusto Query Language (KQL)

## Feedback

Was this page helpful?

Yes

No



# Using Kusto.Explorer

Article • 03/20/2023

Kusto.Explorer is a desktop application that enables you to explore your data using the Kusto Query Language in an easy-to-use user interface. This article shows you how to use search and query modes, share your queries, and manage clusters, databases, and tables.

## Search++ mode

Search++ mode enables you to search for a term using search syntax across one or more tables.

1. In the **Query** dropdown on the **Home** tab, select **Search++**.
2. Select **Multiple tables**.
3. Under **Choose tables**, define which tables to search, then select **OK**.
4. In the edit box, enter your search phrase and select **Go**.

A heat-map of the table/time-slot grid shows which terms appear and where they appear.

The screenshot shows the Kusto Explorer application interface. The main area displays a heatmap grid of data values for multiple tables over time. The left sidebar shows the 'Workspaces' section with a list of tables: SamplePowerRe..., FHV\_Trips, GeoRegions, nyc\_taxi, Trips, PopulationData, StormEvents, demo\_clustering1, demo\_make\_ser..., and demo\_make\_ser... A dropdown menu 'Multiple tables' is open, showing 'KANSAS,TEXAS,OHIO,NEW,YORK,CRD,VTS'. The top navigation bar includes tabs like Home, File, Connections, View, Tools, Monitoring, Management, Help, and a 'Search ++' dropdown. The status bar at the bottom right shows 'UTC: 2023-03-14 08:16:20'.

5. Select a cell in the grid and select **View Details** to show the relevant entries in the results pane.

## Query mode

Kusto.Explorer includes a powerful query mode that enables you to write, edit, and run inline queries. The query mode comes with syntax highlighting and IntelliSense, so you can quickly ramp-up your knowledge of the Kusto Query Language.

This section describes how to run basic queries in Kusto.Explorer and how to add parameters to your queries.

## Basic queries

If you have table Logs, you can start exploring them:

```
Kusto

StormEvents | count
```

When your cursor is on this line, it's colored gray. Press F5 to run the query.

Here are some more example queries:

```
Kusto

// Take 10 lines from the table. Useful to get familiar with the data
StormEvents | take 10
```

```
Kusto

// Filter by EventType == 'Flood' and State == 'California' ( =~ means case
// insensitive)
// and take sample of 10 lines
StormEvents
```

```
| where EventType == 'Flood' and State =~ 'California'  
| take 10
```

The screenshot shows the Kusto Explorer interface with a query editor at the top containing the KQL command. Below the editor is a results table with columns: StartTime, EndTime, EpisodeId, EventId, State, EventType, InjuriesDirect, InjuriesIndirect, DeathsDirect, DeathsIndirect, DamageProperty, DamageCrops, Source, BeginLocation, EndLocation, and Beg. The table displays six records of event data from California. At the bottom of the interface, there are tabs for 'Results', 'Query Summary', and 'QueryCompletionInformation'. A status bar at the bottom right indicates 'Record 1 of 6'.

StartTime	EndTime	EpisodeId	EventId	State	EventType	InjuriesDirect	InjuriesIndirect	DeathsDirect	DeathsIndirect	DamageProperty	DamageCrops	Source	BeginLocation	EndLocation	Beg	
2007-02-19 00:00:00.0000000	2007-02-19 08:00:00.0000000	2021	9977	CALIFORNIA	Flood	0	0	0	0	0	0	0	Mesonet	TAFT	TEHACHAPI	35
2007-10-29 17:20:00.0000000	2007-10-29 21:00:00.0000000	11718	64439	CALIFORNIA	Flood	0	0	0	0	500000	0	0	Newspaper	FIGARDEN	FIG GARDEN	
2007-10-29 18:15:00.0000000	2007-10-29 22:00:00.0000000	11718	64442	CALIFORNIA	Flood	0	0	0	0	20000	0	0	Newspaper	CINOWITHS CORNER	PATTERSON TRACT	
2007-10-29 19:30:00.0000000	2007-10-29 21:30:00.0000000	11718	64443	CALIFORNIA	Flood	0	0	0	0	0	0	0	Park/Forest Service	YOSEMITE VLG	YOSEMITE NATL PARK	
2007-12-18 23:40:00.0000000	2007-12-19 01:00:00.0000000	13209	73291	CALIFORNIA	Flood	0	0	0	0	25000	0	0	Law Enforcement	HAMMOND	FRESNO	
2007-12-19 00:30:00.0000000	2007-12-19 02:00:00.0000000	13209	73292	CALIFORNIA	Flood	0	0	0	0	150000	0	0	Law Enforcement	HAMMOND	(FCH) CHANDLER AF	36

To learn more about the Kusto Query Language, see [Kusto Query Language](#).

### ⓘ Note

Blank lines in the query expression can affect which part of the query is executed.

If no text selected, it's assumed that the query or command is separated by empty lines. If text is selected, the selected text is run.

## Client-side query parameterization

### ⓘ Note

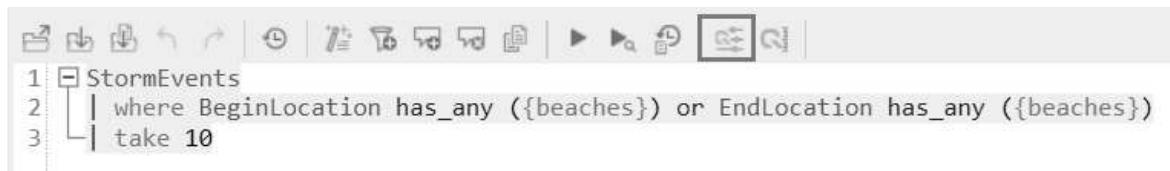
There are two types of query parametrization techniques in Kusto:

- **Language-integrated query parametrization** is implemented as part of the query engine and is meant to be used by applications that query the service programmatically. This method is not described in this document.
- Client-side query parametrization, described below, is a feature of the Kusto.Explorer application only. It's equivalent to using string-replace operations on the queries before sending them to be executed by the service. The syntax described below is not part of the query language itself and can't be used when sending queries to the service by means other than Kusto.Explorer.

If you use the same value in multiple queries or in multiple tabs, it's highly inconvenient to change that value in every place it's used. That's why Kusto.Explorer supports query

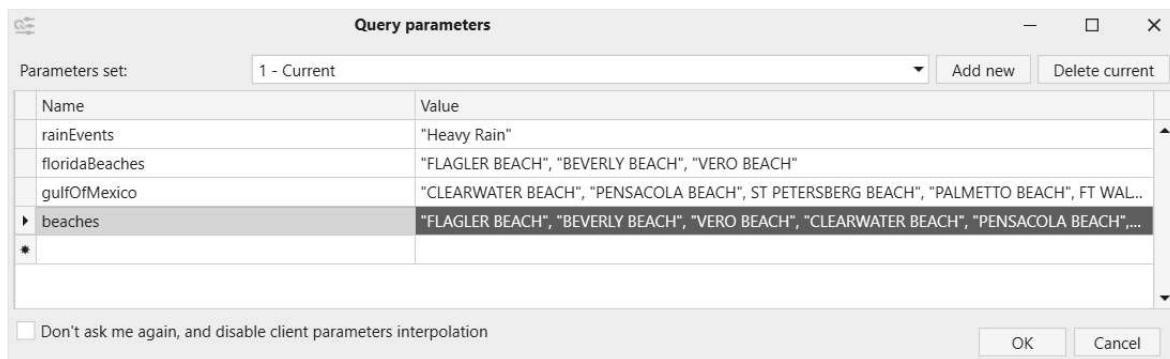
parameters. Query parameters are shared among tabs so that they can be easily reused. Parameters are denoted by {} brackets. For example, {parameter1}.

You can easily define and edit existing query parameters:



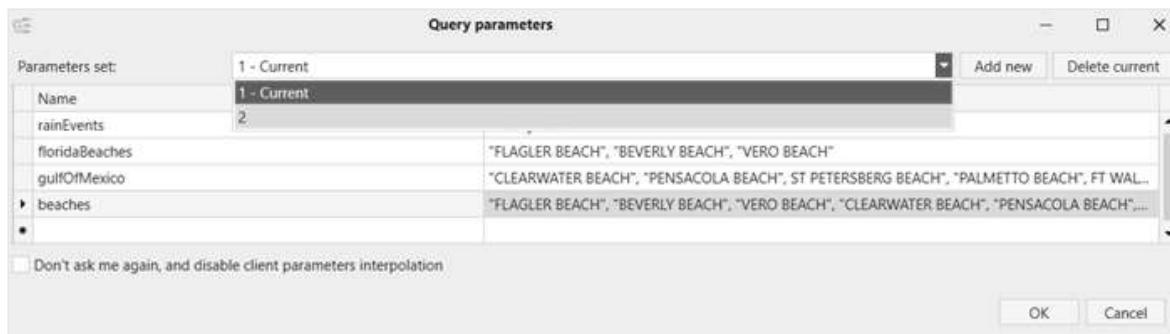
The screenshot shows the Kusto Explorer interface. At the top is a toolbar with various icons. Below it is a query editor window containing the following code:

```
1 | StormEvents
2 |   where BeginLocation has_any ({beaches}) or EndLocation has_any ({beaches})
3 |   take 10
```



You can have multiple sets of parameters (listed in the **Parameters Set** combo box).

Select **Add new** or **Delete current** to manipulate the list of parameter sets.



## Share queries and results

In Kusto.Explorer, you can share queries and results by email. You can also create deep links that open and run a query in the browser.

### Share queries and results by email

Kusto.Explorer provides a convenient way to share queries and query results by email.

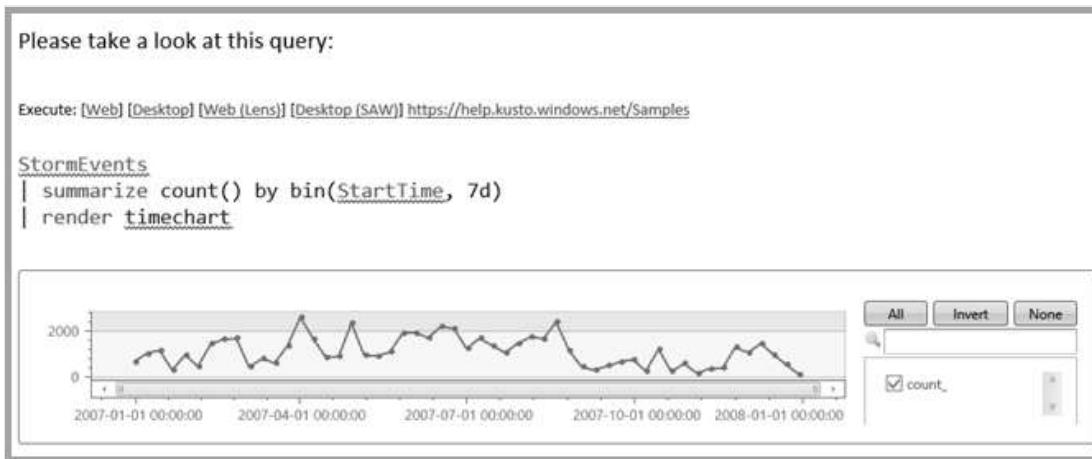
1. Run your query in Kusto.Explorer.
2. In the Home tab, in the Share section, select **Export to Clipboard** (or press **Ctrl+Shift+C**).



Kusto.Explorer copies the following to the clipboard:

- Your query
- The query results (table or chart)
- The connection details for the Kusto cluster and database
- A link that reruns the query automatically

3. Paste the contents of the clipboard into a new email message.



## Deep-linking queries

You can create a URI that, when opened in a browser, opens Kusto.Explorer locally and runs a specific query on a specified Kusto database.

### ! Note

For security reasons, deep-linking is disabled for control commands.

## Creating a deep-link

The easiest way to create a deep-link is to author your query in Kusto.Explorer and then use `Export to Clipboard` to copy the query (including the deep link and results) to the clipboard. You can then share it by email.

When copied to an email, the deep link is displayed in small font. For example:

<https://help.kusto.windows.net/Samples> [Run the query]

The first link opens Kusto.Explorer and sets the cluster and database context appropriately. The second link ([Run the query](#)) is the deep link. If you move the link to an email message and press CTRL+K, you can see the actual URL:

[https://help.kusto.windows.net/Samples?  
web=0&query=H4sIAAAAAAAEAsuyS%2fKdS1LzSspVuDIqIEoLs3NTSzKrEpVSM4vzSvR  
0FRIqlRlyszTCC5JLCoJycxN1VEwT9EEKS1KzUtJLVloAYoIZwAlFQCB3oo%2bTAaaaa%3d%  
3d](https://help.kusto.windows.net/Samples?web=0&query=H4sIAAAAAAAEAsuyS%2fKdS1LzSspVuDIqIEoLs3NTSzKrEpVSM4vzSvR0FRIqlRlyszTCC5JLCoJycxN1VEwT9EEKS1KzUtJLVloAYoIZwAlFQCB3oo%2bTAaaaa%3d%3d)

## Deep-links and parametrized queries

You can use parametrized queries with deep-linking.

1. Create a query to be formed as a parametrized query (for example, `KustoLogs | where Timestamp > ago({Period}) | count`)
2. Provide a parameter for every query parameter in the URI, such as:

```
https://<your\_cluster>.kusto.windows.net/MyDatabase?  
web=0&query=KustoLogs+%7c+where+Timestamp+>+ago\({Period}\)+%7c+count&Period=1h
```

Replace `<your_cluster>` with your Azure Data Explorer cluster name.

## Limitations

The queries are limited to ~2000 characters because of browser limitations, HTTP proxies, and tools that validate links, such as Microsoft Outlook. The limitation is approximate because it's dependent on the cluster and Database name length. For more information, see <https://support.microsoft.com/kb/208427>.

To reduce the chances of reaching the character limit, see Getting Shorter Links.

The format of the URI is:

```
https://<ClusterCname>.kusto.windows.net/<DatabaseName>web=0?query=<QueryToExecute>
```

For example: [https://help.kusto.windows.net/Samples?  
web=0&query=StormEvents+%7c+limit+10](https://help.kusto.windows.net/Samples?web=0&query=StormEvents+%7c+limit+10)

This URI will open Kusto.Explorer, connect to the `Help` Kusto cluster, and run the specified query on the `Samples` database. If there's an instance of Kusto.Explorer already

running, the running instance will open a new tab and run the query in it.

## Getting shorter links

Queries can become long. To reduce the chance the query exceeds the maximum length, use the `string`

`Kusto.Data.Common.Cs1CommandGenerator.EncodeQueryAsBase64Url(string query)` method available in Kusto Client Library. This method produces a more compact version of the query. The shorter format is also recognized by Kusto.Explorer.

<https://help.kusto.windows.net/Samples?web=0&query=H4sIAAAAAAAEAsuyS%2fKdS1LzSspVuDlqlEoLs3NTSzKrEpVSM4vzSvR0FRlqlRlyszTCC5JLCojYcxN1VEwT9EEKS1KzUtJLVloAYoIZwAIFQCB3oo%2bTAAAAA%3d%3d>

The query is made more compact by applying next transformation:

```
C#  
  
UrlEncode(Base64Encode(GZip(original query)))
```

## Kusto.Explorer command-line arguments

Command-line arguments are used to configure the tool to perform additional functions on start-up. For example, load a script and connect to a cluster. As such, command-line arguments aren't a replacement for any Kusto.Explorer functionality.

Command-line arguments are passed as part of the URL that's used to open the application, in a similar way to query deep-linking.

## Command-line argument syntax

Kusto.Explorer supports several command-line arguments in the following syntax (the order matters):

`[LocalScriptFile] [QueryString]`

- `LocalScriptFile` is the name of a script file on your local machine, which must have the extension `.kql`. If such a file exists, Kusto.Explorer automatically loads this file when it starts up.
- `QueryString` is a string that uses HTTP query string formatting. This method provides additional properties, as described in the table below.

For example, to start Kusto.Explorer with a script file called `c:\temp\script.kql` and configured to communicate with cluster `help`, database `Samples`, use the following command:

```
Kusto
Kusto.Explorer.exe c:\temp\script.kql
uri=https://help.kusto.windows.net/Samples;Fed=true&name=Samples
```

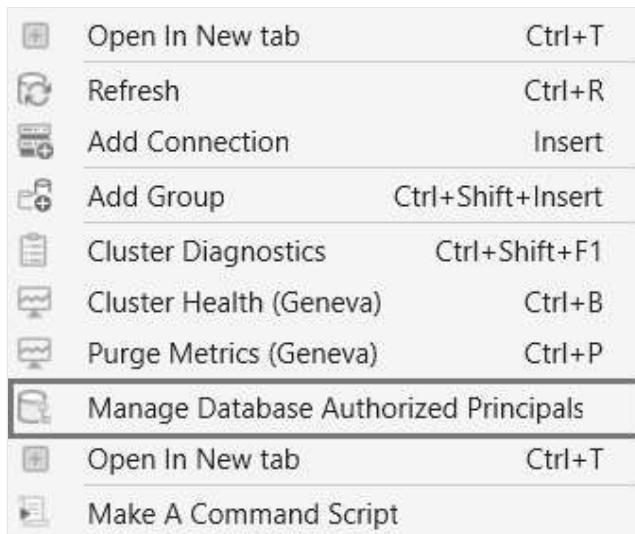
Argument	Description
<b>Query to execute</b>	
<code>query</code>	The query to execute (gzipped, then base64-encoded; see "Getting shorter links" above). If empty, use <code>querysrc</code> .
<code>querysrc</code>	The HTTP URL of a file/blob holding the query to execute (if <code>query</code> is empty).
<b>Connection to the Kusto cluster</b>	
<code>uri</code>	The connection string of the Kusto cluster to connect to.
<code>name</code>	The display name of the connection to the Kusto cluster.
<b>Connection group</b>	
<code>path</code>	The URL of a connection group file to download (URL-encoded).
<code>group</code>	The name of the connection group.
<code>filename</code>	The local file holding the connection group.

## Manage databases, tables, or function authorized principals

### i Important

Only admins can add or drop authorized principals in their own scope.

1. To view the list of authorized principals, right-click the target entity in the Connections panel, and select **Manage Database Authorized Principals**. (You can also select this option from the Management Menu.)



2. Select **Add principal** to add an authorized principal.

Principal Display Name	Role	Principal Type	Principal Object Id	Principal FQN
Cris Barros	Database Samples Admin	AAD User	12345678-1234-123a-bcd	aaduser=12345678-1234-123a-b- Ad

Drop principal    Add principal

3. Provide the principal details, then select **Add principal**.

Select principal type and role

Principal type: Kusto User

Role assignment: Database 'Samples' User

Can read all data and metadata of the database; additionally, can create tables (thus becoming the table admin for that table) and functions in the database.

Click here for additional information on Kusto access control

Principal identity

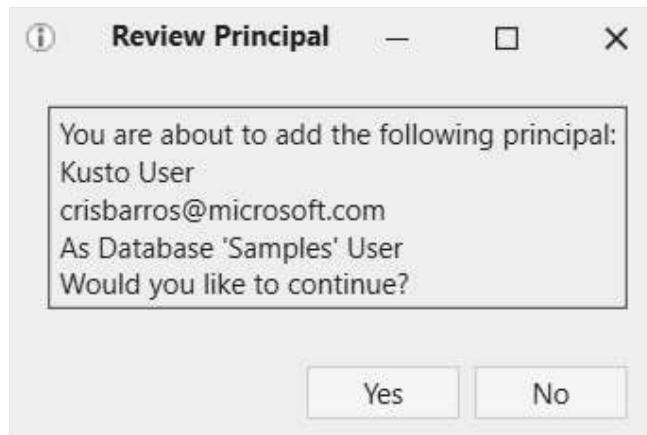
Identifier: crisbarros@microsoft.com

Enter the username.

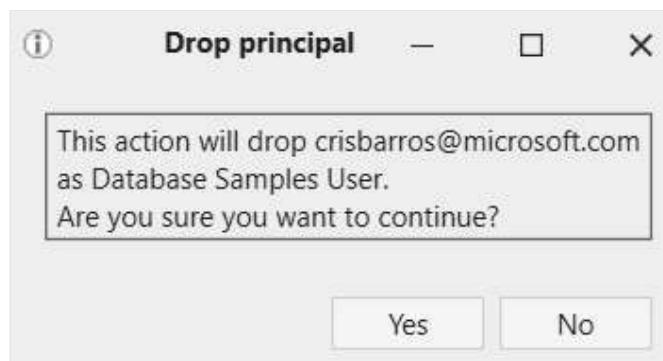
Add a note to the role assignment (optional)

Add principal

4. Confirm that you want to add the authorized principal.



To drop an existing authorized principal, select **Drop principal** and confirm the action.



## Next steps

- Kusto.Explorer keyboard shortcuts
- Kusto.Explorer options
- Troubleshooting Kusto.Explorer

To Learn more about Kusto.Explorer tools and utilities, See:

- Kusto.Explorer code features
- Kusto Query Language (KQL)

---

## Feedback

Was this page helpful?

Provide product feedback | Get help at Microsoft Q&A

# Kusto Explorer options

Article • 03/20/2023

The following tables describe the options for customizing the behavior of Kusto.Explorer from the **Tools > Options** dialog box.

## General Settings

Option	Description
Tool Mode	Enables beta, alpha and experimental application features. Default is none.
Extended Accessibility	When enabled, the application sends accessibility events used by accessibility tools. May affect performance if enabled. Default is disabled.
Allow sending telemetry data	When enabled, the application sends telemetry data when errors occur or the application crashes.
Welcome message	When enabled, the application shows the welcome message on start. Default is enabled.
Display theme	The application UI color scheme: light or dark.
Use legacy display themes	Enables and disables using legacy themes.
Multi-line tabs	Enables and disables multi-line arrangement of the query panel tabs. Default is enabled.

## Query editor

Option	Description
Queries auto-save	When enabled, the application automatically saves open documents. Changing this setting requires restarting the application to take effect. Enabled by default.
Track changes	When enabled, the application tracks changes made to query scripts on disk by other applications. If the query script is changed externally, you'll be notified and prompted for reloading it. Changing this setting requires restarting the application. Enabled by default.
Use Edit Sessions	When enabled, each application process owns its own set of documents. Disabled by default.
Font	The font used in the Query editor.

<b>Option</b>	<b>Description</b>
Font Size	The font size used in the query editor.
Select command background	The background color used to highlight the currently selected command.
Replace tabs with spaces	When enabled, tabs are automatically replaced with spaces.
Disable function parameter injection	When enabled, function parameter injection from the clipboard is disabled.
Disable query parameter injection	When enabled, query parameter injection is disabled.
Disable query run triggers except F5	When enabled, only the F5 key triggers queries to run.
Show help inside the application	When enabled, Help topics are shown inside the application. When disabled Help topics are opened inside a browser.
Query parameter length limit	The maximal length of a string that can be used as a query parameter. Setting this value to more than 128K may cause performance issues. Default is 64K.
Quick replacements	Invoke "Quick replacement" by pressing Ctrl+Space in the query editor when the caret is at the text to be replaced.

## IntelliSense

<b>Option</b>	<b>Description</b>
IntelliSense	Enables or disables IntelliSense. Default is enabled.
IntelliSense-Command Version	Chooses the IntelliSense version for control commands. Default: mixed.
Syntax Highlighting	Enables or disables syntax highlighting. Default is enabled.

Option	Description
Issues List	Displays the issues list for the current query or command in the editor.
Issue Options	Adjust which issues appear in the issues list.
Tool-tips	Enables or disables the tool-tips that appear when the mouse hovers over areas of the selected query. Default is enabled.

## Formatter

Option	Description
Version	The version of the formatter used when the Prettify Query tool is applied to the current query. <i>V1</i> is the classic formatter. <i>V2</i> is the modern formatter. Default is <i>V2</i> .
Indentation	The number of spaces for indented items.
Bracketing Style	The placement of braces, parentheses, and brackets.
Schema	The alignment of the parentheses around a schema declaration, when on separate lines.
Data Table Values	The alignment of brackets around data table values, when on separate lines.
Function Body	The alignment of function body brace. * when on separate lines.
Function Parameters	The alignment of function parameter parentheses, when on separate lines.
Function Arguments	The alignment of function argument parentheses, when on separate lines.
Pipe Operator	The placement of the pipe (bar) character that exists between tabular query operators. <i>New Line</i> places all pipe characters on a new line. <i>Smart</i> places all pipe characters on a new line if the query already spans more than one line. <i>None</i> leaves all pipe characters as is.
Expression	The placement of expressions in lists.
Statement	The placement of statements.
Semicolon	The placement of semicolons that end query statements. <i>New Line</i> places semicolons on a new line, indented. <i>Smart</i> places semicolons on a new line if the statement itself spans more than one line. <i>None</i> leaves semicolons as is.

Option	Description
Insert Missing Syntax	Adds missing punctuation like (such as commas and semicolons) that you're getting error messages for.

## Results Viewer

Option	Description
Font Size	The font size used by the data table grid, where query results are shown.
Font	Font family used to display results.
Max lines per cell	Maximum number of lines to show in each cell. Changing the setting to '-1' will result in unlimited lines per cell. Default is eight lines per cell.
Numeric formatting	Formatting used for numbers. Default is no formatting.
Verbosity color scheme	Selects the color scheme for row formatting based on autodetected verbosity level.
Hide empty columns	When enabled, empty columns in the view showing data will be hidden. Default is disabled.
Collapse single-value columns	When enabled, will autocollapse single-value columns in the view showing data. Columns with single nonempty values will be shown as groups. Default is disabled.
Templated value replacement	When enabled, you can interpolate the template column 'body:string' using the other columns, and the properties column 'env_properties:dynamic'. Interpolation is done for the pattern '{columnOrPropertyName}'. This option requires the result to have a 'body' and 'env_properties' columns. Default is disabled.
Auto-sort results by datetime columns	When enabled, rows will be auto-sorted by datetime columns. Default is enabled.
Column visible range	The maximal number of characters to be displayed in a column. Default is 2048.
Text wrapping	When enabled, text that doesn't fit into cell width will be wrapped according to the column width. Default is disabled.

<b>Option</b>	<b>Description</b>
Visualize JSON as text	When enabled, JSON values are visualized as single-row text. Default is disabled.
Maximum Text Details Size	The maximum number of characters that will be shown in the detailed information panel when the cell is double-clicked. Default is 32KB.
Play Completion Sound	Plays sounds when query or command is completed. Default is disabled.

## Connections

<b>Option</b>	<b>Description</b>
Additional Trusted Hosts	Lists additional trusted hosts that are considered safe to connect to. Use ';' to specify more than a single host, and '*' to specify a multiple-domain pattern.
Sort table columns alphabetically	When enabled, the columns appearing under the table nodes in the connections panel will be sorted alphabetically.
Show Views in 'Function' folder	When enabled, functions with the 'view=true' tag will appear under the Functions folder or Connections panel.
Query Server Timeout	The server timeout for query execution.
Admin Command Server Timeout	Specifies the server timeout for admin command execution.
Lazy schema exploration	When enabled, the connections panel will only fetch and display database schema when the database node is expanded.
Show hidden system objects	When enabled, hidden system objects will be shown if the user has appropriate permissions.
Query weak consistency	When enabled, weak consistency will be used for queries.
KQL parses version	Determines which version of the Kusto Query Language parser will be used when executing a query.

Option	Description
Allow unsafe connections	Allows using unsafe connection protocols for local dev/test environments.

## Diagnostic Tracing

Option	Description
Enable Tracing	Enables tracing.
Trace verbosity	Sets the verbosity of tracing.
Traces location	The location where traces are logged.
PlatformTraceVerbosity	Sets the verbosity of tracing for the platform.

## Tools

Option	Description
Export Visible Result Only	When enabled, exporting results to clipboard only exports the data from currently visible results tab.

## Feedback

Was this page helpful?



Yes



No

Provide product feedback | Get help at Microsoft Q&A

# Kusto.Explorer keyboard shortcuts (hot-keys)

Article • 03/20/2023

## Application-level

The following keyboard shortcuts can be used from any context:

Hot Key	Description
F1	Opens the help
F11	Toggle full-view mode
Ctrl + F1	Toggle ribbon appearance
Ctrl + +	Increases query and data results font
Ctrl + -	Decreases query and data results font
Ctrl + 0	Resets query and data results font
Ctrl + 1 .. 7	Switches to Query document panel with respective number (1..7)
Ctrl + F2	Renames header of the Query Editor panel
Ctrl + N	Opens a new query editor
Ctrl + O	Open query editor script in a new query editor
Ctrl + W	Closes active query editor
Ctrl + S	Saves query into a file
Shift + F3	Opens Analytical Report Gallery
Shift + F12	Toggles Light/Dark theme of the application
Ctrl + F12	Toggles Extended Accessibility events of the application (requires restart to take effect)
Ctrl + Shift + 0	Opens Kusto.Explorer options and settings dialog
Esc	Cancel running query
Shift + F5	Cancel running query

# Query and results view

You can use the following keyboard shortcuts with the query editor or when the context is in the results view:

Hot Key	Description
<code>Ctrl + Shift + C</code>	Copies query, deep-link, and data to the clipboard
<code>Alt + Shift + C</code>	Copies query and deep-link the clipboard in HTML format
<code>Alt + Shift + R</code>	Copies query, deep-link, and places data in the clipboard in Markdown format
<code>Alt + Shift + M</code>	Copies query and deep-link the clipboard in Markdown format
<code>Ctrl + ~</code>	Copies query and data to the clipboard in markdown format
<code>Ctrl + Shift + D</code>	Toggles mode of hiding duplicate rows in the data view
<code>Alt + Shift + H</code>	Toggles mode of hiding empty columns in the data view
<code>Ctrl + Shift + J</code>	Toggles mode of collapsing columns with single value in the data view
<code>Ctrl + Shift + A</code>	Opens a Query Analyzer tool in a new query panel
<code>Alt + C</code>	Renders Column chart over existing data
<code>Alt + T</code>	Renders Timeline chart over existing data
<code>Alt + A</code>	Renders Anomaly timeline chart over existing data
<code>Alt + P</code>	Renders Pie chart over existing data
<code>Alt + L</code>	Renders Ladder timeline chart over existing data
<code>Alt + V</code>	Renders Pivot chart over existing data
<code>Ctrl + Shift + V</code>	Shows Timeline pivot over existing data
<code>Ctrl + F9</code>	Toggles <code>show only matching rows</code> / <code>highlight matching rows</code> modes for client text search ( <code>Ctrl + F</code> ) behavior in data grid.
<code>Ctrl + F10</code>	Shows details panel - where selected row is presented as property grid
<code>Ctrl + F</code>	Shows search box for the panel that is currently in focus. Supported in <code>Connections</code> , <code>Data Results</code> , and <code>Query Editor</code> panels
<code>Ctrl + Tab</code>	Shows Query Editor document selector dialog. You can hold <code>ctrl</code> and switch between documents with <code>Tab</code>
<code>Ctrl + J</code>	Toggles appearance of the result panel

<b>Hot Key</b>	<b>Description</b>
<code>Ctrl + E</code>	Toggles appearance of the query editor and result panel in cycle of: <code>Query Editor and Results -&gt; Query Editor -&gt; Query Editor and Results -&gt; Results</code>
<code>Ctrl + Shift + E</code>	Toggles appearance of the query editor and result panel in cycle of: <code>Query Editor and Results -&gt; Results -&gt; Query Editor and Results -&gt; Query Editor</code>
<code>F6</code>	Toggles focus of the main application panel in cycle of: <code>Connections panel -&gt; Query Editor -&gt; Results</code>
<code>Ctrl + Shift + R</code>	Focuses on Results panel
<code>Ctrl + Shift + T</code>	Focuses on Connections panel
<code>Ctrl + Shift + Y</code>	Focuses on Query editor
<code>Ctrl + Shift + P</code>	Focuses on Chart panel
<code>Ctrl + Shift + I</code>	Focuses on Query Information panel
<code>Ctrl + Shift + S</code>	Focuses on Query Statistics panel
<code>Ctrl + Shift + K</code>	Focuses on Error panel
<code>Alt + Ctrl + L</code>	Locks current connection context to the Query Editor, so changing selected row in the Connection panel has no effect on the Query Editor context.

## Results Table Viewer

The following keyboard shortcuts can be used when results view (table) is in active keyboard focus:

<b>Hot Key</b>	<b>Description</b>
<code>Ctrl + Q</code>	Show current column context menu
<code>Ctrl + S</code>	Toggle current column sorting
<code>Ctrl + U</code>	Opens a panel showing current column values with client-side filtering
<code>Ctrl + F</code>	Shows search box for the results
<code>Ctrl + F3</code>	Toggles <code>show only matching rows/highlight matching rows</code> modes for client text search ( <code>Ctrl + F</code> ) behavior in data grid.

## Query editor

The following keyboard shortcuts can be used when editing a query in the query editor:

Hot Key	Description
F1	When cursor points to an operator or function - opens a help window with information about the operator or function. If the help topic isn't present - opens a help URL
F5	Run currently selected query
Shift + Enter	Run currently selected query
F8	Fetch query results from the local cache. If results aren't present - run currently selected query
Ctrl + F5	Preview results of the selected query (shows few results and total count)
Ctrl + Shift + Space	Insert data cell selections as filters into the query
Ctrl + Space	Force IntelliSense rules check. Possible options will be shown in any rule matched
Ctrl + Enter	Adds pipe symbol and moves to a new line
Ctrl + Z	Undo
Ctrl + Y	Redo
Ctrl + L	Deletes current line
Ctrl + D	Deletes current line
Ctrl + F	Opens Find dialog
Ctrl + Shift + F	Opens Find dialog (all tabs lookup)
Ctrl + H	Opens Replace dialog
Ctrl + Shift + H	Opens Replace dialog (all tabs lookup)
Ctrl + G	Opens Go-to line dialog
Ctrl + F8	Show my queries past 3 days
Ctrl + bracket	When cursor is at bracket symbols: ( , ) , [ , ] , { , } - moves cursor to the matching opening or closing bracket
Ctrl + Shift + Q	Prettify current query
Ctrl + Shift + L	Make current query or selection lower-case
Ctrl + Shift + U	Make current query or selection upper-case

<b>Hot Key</b>	<b>Description</b>
<code>Ctrl + Mouse wheel up</code>	Increases font of the query editor
<code>Ctrl + Mouse wheel down</code>	Decreases font of the query editor
<code>Alt + P</code>	Opens query parameters dialog
<code>F2</code>	Open current line / selected text in editor dialog
<code>Ctrl + F6</code>	Runs KQL static query analysis to detect common issues
<code>F12</code>	Navigate to the definition of the symbol
<code>Alt + F12</code>	Find all references of the current symbol
<code>Alt + Home</code>	Navigate to the definition of the symbol
<code>Alt + Ctrl + M</code>	Extract currently selected literal or tabular expression as let statement
<code>Ctrl + .</code>	Extract currently selected literal or tabular expression as let statement
<code>Ctrl + R, Ctrl + R</code>	Renames current symbol
<code>Ctrl + K, Ctrl + D</code>	Inserts current timestamp as datetime literal
<code>Ctrl + K, Ctrl + R</code>	Inserts <code>range x from 1 to 1 step 1</code> snippet
<code>Ctrl + K, Ctrl + C</code>	Comment current line or selected lines
<code>Ctrl + K, Ctrl + F</code>	Prettify current query
<code>Ctrl + K, Ctrl + V</code>	Duplicate current query (append it to the end of current query document)
<code>Ctrl + K, Ctrl + U</code>	Uncomment current line or selected lines
<code>Ctrl + K, Ctrl + S</code>	Turn current line or selected lines into multi-line string literal
<code>Ctrl + K, Ctrl + M</code>	Remove multi-line string literal marks (reverse of <code>Ctrl + K, Ctrl + S</code> )
<code>Ctrl + M, Ctrl + M</code>	Toggle outlining expansion of the current query
<code>Ctrl + M, Ctrl + L</code>	Toggle outlining expansion of all queries in the document

## JSON viewer

The following keyboard shortcuts can be used from within the results JSON viewer. They display if you double-click on a JSON-like value in the results view cell:

<b>Hot Key</b>	<b>Description</b>
<code>Ctrl + Up Arrow</code>	Navigate to parent
<code>Ctrl + Right Arrow</code>	Expand current node (one level)
<code>Ctrl + Left Arrow</code>	Collapse current node (one level)
<code>Ctrl + .</code>	Toggle expansion of the current node (all child levels expanded/collapsed)
<code>Ctrl + Shift + .</code>	Toggle expansion of the current node parent (all child levels expanded/collapsed)

## Connection panel

The following keyboard shortcuts can be used from within the results JSON viewer. They display if you double-click on a JSON-like value in the results view cell:

<b>Hot Key</b>	<b>Description</b>
<code>Ctrl + Up Arrow</code>	Navigate to parent
<code>Ctrl + Right Arrow</code>	Expand current node (one level)
<code>Ctrl + Left Arrow</code>	Collapse current node (one level)
<code>Ctrl + Shift + L</code>	Collapse all levels
<code>Ctrl + R</code>	Refresh currently selected connection
<code>Insert</code>	Add a new connection
<code>Del</code>	Delete current connection
<code>Ctrl + E</code>	Edit currently selected connection
<code>Ctrl + T</code>	Open a new query editor using currently selected connection

## Diagnostics and Monitoring

The following keyboard shortcuts are available from `Monitoring` ribbon.

<b>Hot Key</b>	<b>Description</b>
<code>Ctrl + Shift + F1</code>	Run cluster diagnostic flow

---

# Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# Kusto.Explorer code features

Article • 03/20/2023

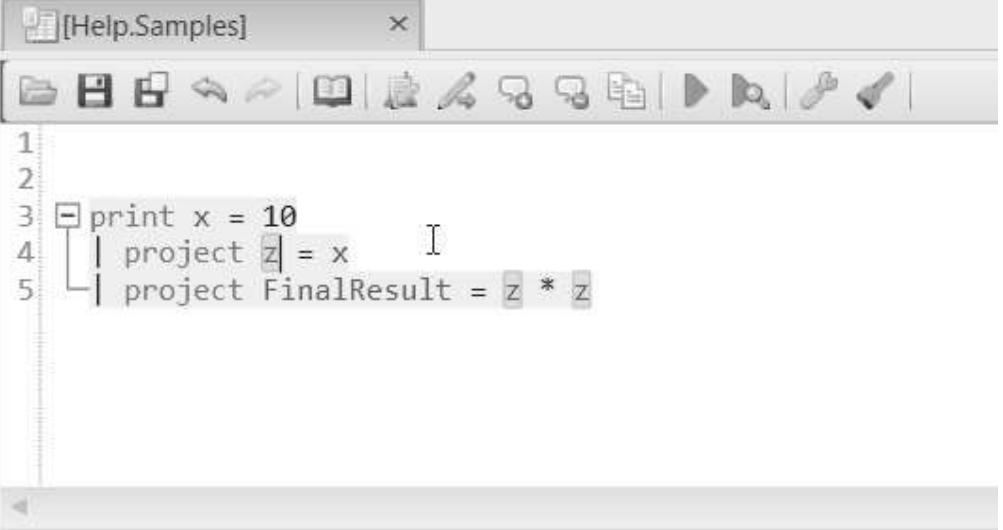
Similar to other IDEs, Kusto.Explorer offers a variety of code features, including Code refactoring, Code navigation, and a Code analyzer utility.

## Code refactoring

Use Kusto.Explorer's KQL query editing and refactoring features to rename variables and column names, and extract scalars and tabular statements as `let` expressions.

### Rename variable or column name

Rename selected symbols by clicking `Ctrl + R` in the query editor window.



```
[Help.Samples] x
[File] [Save] [New] [Open] [Save As] [Import] [Export] [Find] [Replace] [Copy] [Cut] [Paste] [Delete] [Run] [Stop] [Break] [Properties]
1
2
3 print x = 10
4 | project z=x
5 | project FinalResult = z * z
```

### Extract scalars as `let` expressions

To define selected literals as `let` expressions, press `Alt + Ctrl + M`.

```
1
2
3 let StormEvents =
4     | where StartTime > datetime(2015-01-01)
5     | where EndTime < datetime(2015-01-01) + 30d
6     | count
```

## Extract tabular statements as `let` expressions

To define tabular expressions as `let` statements, select the text, and then press

`Alt + Ctrl + M`.

```
1
2
3 let StormEvents =
4     | where StartTime > datetime(2015-01-01)
5     | where EndTime < datetime(2015-01-01) + 30d
6     | count
```

## Code navigation

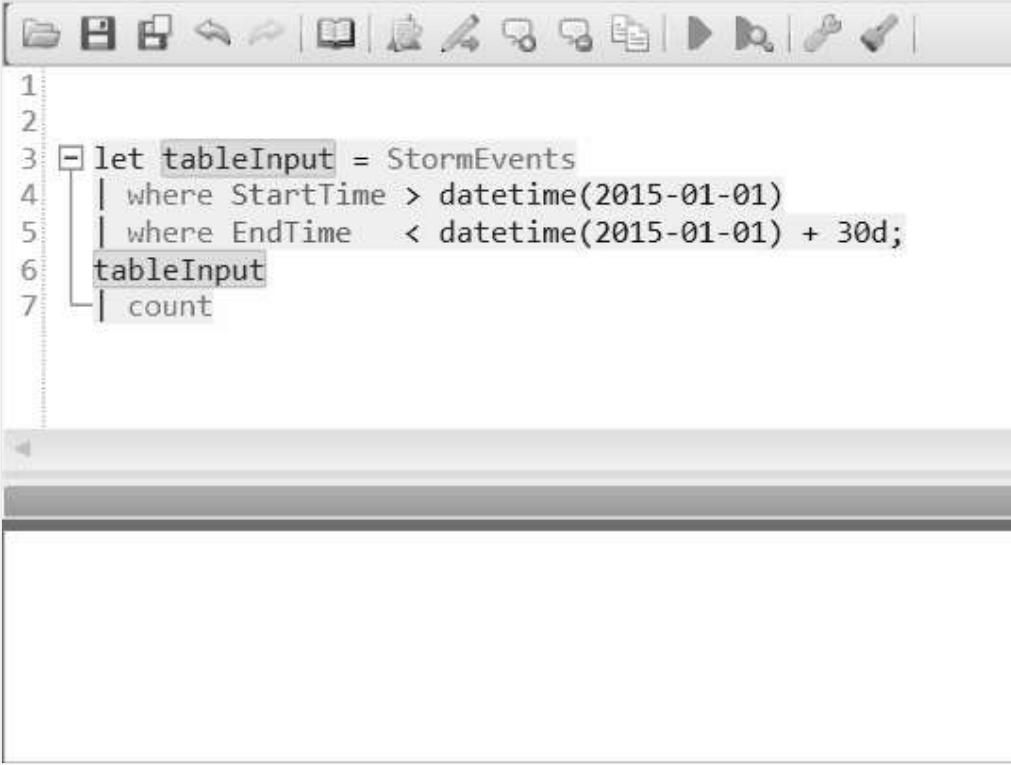
Kusto.Explorer provides several features for easy code navigation using query symbols information.

### Go-to symbol definition

You can navigate to the definition of the current symbol using `F12` or the `Alt + Home` shortcuts.

## List all references of a symbol

You can obtain all references of the current symbol using the **ctrl+F12** shortcut.



The screenshot shows the Kusto Explorer interface. At the top is a toolbar with various icons. The fourth icon from the left, which looks like a magnifying glass with a line, is highlighted with a red box. Below the toolbar is a code editor window containing the following Kusto query:

```
1
2
3 let tableInput = StormEvents
4 | where StartTime > datetime(2015-01-01)
5 | where EndTime < datetime(2015-01-01) + 30d;
6 tableInput
7 | count
```

For more information on keyboard shortcuts in Kusto.Explorer, see [Keyboard shortcuts](#).

## Code analyzer

Use Kusto.Explorer's code analyzer utility to automatically analyze the current query and output a set of applicable improvement recommendations.

To view improvement recommendations, at the bottom of the result grid, select the **Issues** tab.

```
1 StormEvents
2 | where
```

Issues: (1)

Code	Description	Category	Severity
KS006	Missing expression	General	Error

Issues found: 1

Results Issues: (1)

---

## Feedback

Was this page helpful?

Provide product feedback [↗](#) | Get help at Microsoft Q&A

# Troubleshooting

Article • 03/07/2022

This document provides common difficulties in running and using Kusto.Explorer, and offers solutions. This document also describes how to reset Kusto.Explorer.

## Kusto.Explorer fails to start

### Kusto.Explorer shows error dialog during or after start-up

#### Symptom

At start-up, Kusto.Explorer shows an `InvalidOperationException` error.

#### Possible solution

This error may suggest that the operating system became corrupted or is missing some of the essential modules. To check missing or corrupted system files, follow the steps described here:

[https://support.microsoft.com/help/929833/use-the-system-file-checker-tool-to-repair-missing-or-corrupted-system ↗](https://support.microsoft.com/help/929833/use-the-system-file-checker-tool-to-repair-missing-or-corrupted-system)

## Kusto.Explorer always downloads, even when there are no updates

#### Symptom

Every time you open Kusto.Explorer, you're prompted to install a new version.

Kusto.Explorer downloads the entire package, without updating the already-installed version.

#### Possible solution

This symptom could be a result of corruption in your local ClickOnce store. You can clear the local ClickOnce store, by running the following command, in an elevated command prompt.

## ⓘ Important

1. If there are any other instances of ClickOnce applications or of `dfsvc.exe`, terminate them before running this command.
2. All ClickOnce apps will reinstall automatically the next time you run them, as long as you have access to the original install location stored in the app shortcut. App shortcuts won't be deleted.

Kusto

```
rd /q /s %userprofile%\appdata\local\apps\2.0
```

Try installing Kusto.Explorer again from one of the installation mirrors.

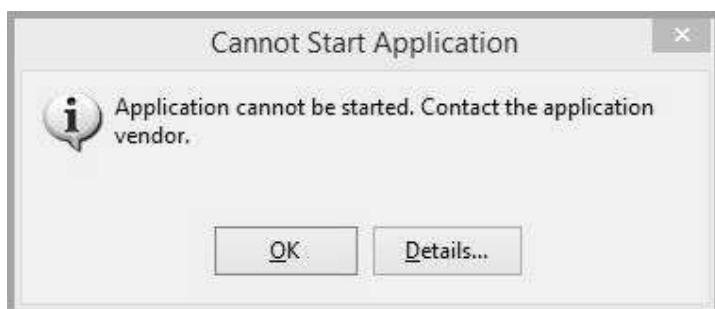
## ClickOnce error: Cannot Start Application

### Symptoms

The program fails to start and displays one of the following errors:

- External component has thrown an exception
- Value does not fall within the expected range
- The application binding data format is invalid.
- Exception from HRESULT: 0x800736B2
- The referenced assembly is not installed on your system. (Exception from HRESULT: 0x800736B3)

You can explore the error details by clicking `Details` in the following error dialog:



Kusto

```
Following errors were detected during this operation.  
* System.ArgumentException
```

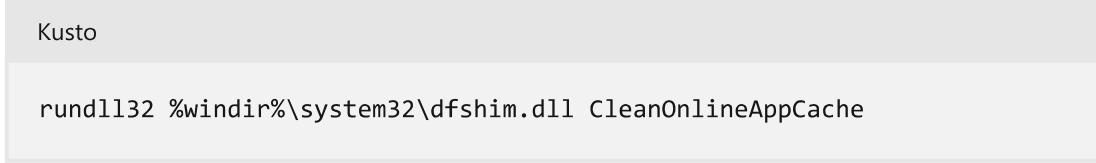
```

    - Value does not fall within the expected range.
    - Source: System.Deployment
    - Stack trace:
        at
System.Deployment.Application.NativeMethods.ClrLaunchApplication(UInt32
hostType, String applicationFullName, Int32 manifestPathsCount, String[]
manifestPaths, Int32 activationDataCount, String[] activationData,
PROCESS_INFORMATION processInformation)
        at
System.Deployment.Application.ComponentStore.ActivateApplication(DefinitionA
ppId appId, String activationParameter, Boolean useActivationParameter)
        at
System.Deployment.Application.SubscriptionStore.ActivateApplication(Definiti
onappId appId, String activationParameter, Boolean useActivationParameter)
        at
System.Deployment.Application.ApplicationActivator.Activate(DefinitionAppId
appId, AssemblyManifest appManifest, String activationParameter, Boolean
useActivationParameter)
        at
System.Deployment.Application.ApplicationActivator.ProcessOrFollowShortcut(S
tring shortcutFile, String& errorPageUrl, TempFile& deployFile)
        at
System.Deployment.Application.ApplicationActivator.PerformDeploymentActivati
on(Uri activationUri, Boolean isShortcut, String textualSubId, String
deploymentProviderUrlFromExtension, BrowserSettings browserSettings, String&
errorPageUrl)
        at
System.Deployment.Application.ApplicationActivator.ActivateDeploymentWorker(
Object state)

```

## Proposed solution steps

1. Uninstall Kusto.Explorer using `Programs and Features` (`appwiz.cpl`).
2. Try running `CleanOnlineAppCache`, and then try installing Kusto.Explorer again.  
From an elevated command-prompt:



```
rundll32 %windir%\system32\dfshim.dll CleanOnlineAppCache
```

Install Kusto.Explorer again from one of the installation mirrors.

3. If the application still doesn't start, delete the local ClickOnce store. All ClickOnce apps will reinstall automatically the next time you run them, as long as you have access to the original install location stored in the app shortcut. App shortcuts won't be deleted.

From an elevated command-prompt:

Kusto

```
rd /q /s %userprofile%\appdata\local\apps\2.0
```

Install Kusto.Explorer again from one of the installation mirrors

4. If the application still doesn't start:

- a. Remove temporary deployment files.
- b. Rename the Kusto.Explorer local AppData folder.

From an elevated command-prompt:

Kusto

```
rd /s/q %userprofile%\AppData\Local\Temp\Deployment  
ren %LOCALAPPDATA%\Kusto.Explorer Kusto.Explorer.bak
```

- c. Install Kusto.Explorer again from one of the installation mirrors
- d. To restore your connections from Kusto.Explorer.bak, from an elevated command-prompt:

Kusto

```
copy %LOCALAPPDATA%\Kusto.Explorer.bak\User*.xml  
%LOCALAPPDATA%\Kusto.Explorer
```

## Enabling ClickOnce verbose logging

1. If the application still doesn't start:

- a. Enable verbose ClickOnce logging by creating a LogVerbosityLevel string value of 1 under:

Kusto

```
HKEY_CURRENT_USER\Software\Classes\Software\Microsoft\Windows\CurrentVersion\Deployment
```

- b. Repro it again.
- c. Send the verbose output to KEBugReport@microsoft.com.

# ClickOnce error: Your administrator has blocked this application because it potentially poses a security risk to your computer

## Symptom

The application fails to install with either of the following errors:

- Your administrator has blocked this application because it potentially poses a security risk to your computer.
- Your security settings do not allow this application to be installed on your computer.

## Solution

This symptom could be because another application is overriding the default ClickOnce trust prompt behavior.

1. View your default configuration settings.
2. Compare your configuration settings to the actual ones on your machine.
3. Reset your configuration settings as necessary, as explained in this how-to article.

## Cleanup application data

Sometimes, when previous troubleshooting steps didn't help with getting Kusto.Explorer to start, cleaning data stored locally may help.

Data stored by Kusto.Explorer application can be found here: C:\Users\[your username]\AppData\Local\Kusto.Explorer.

### ⓘ Note

Cleaning the data will lead to a loss of opened tabs (Recovery folder), saved connections (Connections folder), and application settings (UserSettings folder).

## Reset Kusto.Explorer

If you need to, you can completely reset Kusto.Explorer. The following procedure describes how to progressively reset Kusto.Explorer, until it's removed from your

computer and must be installed from scratch.

1. In Windows, open **Change or remove a program** (also known as **Programs and Features**).
2. Select every item that starts with `Kusto.Explorer`.
3. Select **Uninstall**.

If this procedure fails to uninstall the application (a known issue with ClickOnce applications), see this article for instructions [↗](#).

4. Delete the folder `%LOCALAPPDATA%\Kusto.Explorer`, which removes all connections, history, and so on.
5. Delete the folder `%APPDATA%\Kusto`, which removes the Kusto.Explorer token cache. You'll need to reauthenticate to all clusters.

It's also possible to revert to a specific version of Kusto.Explorer:

1. Run `appwiz.cpl`.
2. Select **Kusto.Explorer** and select **Uninstall/Change**.
3. Select **Restore the application to its previous state**.

## Next steps

- Learn about the Kusto.Explorer user interface
- Learn about running Kusto.Explorer from the command line
- Learn about Kusto Query Language (KQL)

---

## Feedback

Was this page helpful?  Yes  No

Provide product feedback [↗](#) | Get help at Microsoft Q&A

# Kusto CLI

Article • 10/02/2022

Kusto.Cli is a command-line utility that is used to send requests to Kusto, and display the results. It can run in one of several modes:

- *REPL mode*: The user enters queries and commands, and the tool displays the results, then awaits the next user query/command. ("REPL" stands for "read/eval/print/loop".)
- *Execute mode*: The user enters one or more queries and commands to run as command-line arguments. The arguments are automatically run in sequence, and their results output to the console. Optionally, after all the input queries and commands have run, the tool goes into REPL mode.
- *Script mode*: Similar to execute mode, but with the queries and commands specified through a "script" file.

Kusto.Cli is primarily provided for automating tasks against a Kusto service that normally requires writing code. For example, a C# program or a PowerShell script.

## Get the tool

Kusto.Cli is part of the NuGet package `Microsoft.Azure.Kusto.Tools` that you can download for .NET. After you download the package, extract the package's `tools` folder to the target folder. No additional installation is required because it's xcopy-installable.

## Run the tool

Kusto.Cli requires at least one command-line argument to run. Usually, that argument is the connection string to the Kusto service that the tool should connect to. For more information, see Kusto connection strings. If you run the tool without command-line arguments, with an unknown set of arguments, or with the `/help` switch, a help message will display on the console.

For example, use the following command to run Kusto.Cli. The command will connect to the `help` Kusto service, and set the database context to the `Samples` database:

```
Kusto.Cli.exe "https://help.kusto.windows.net/Samples;Fed=true"
```

### ⓘ Note

Use double-quotes around the connection string to prevent shell applications such as PowerShell from mis-interpreting the semicolon ( ; ) and similar characters.

## Command-line arguments

`Kusto.Cli.exe ConnectionString [Switches]`

*ConnectionString*

- The Kusto connection string that holds all the Kusto connection information.  
Defaults to `net.tcp://localhost/NetDefaultDB`.

`-execute:QueryOrCommand`

- If specified, runs Kusto.Cli in execute mode and the specified query or command is run. This switch can repeat, and the queries/commands are run sequentially in order of appearance. This switch can't be used together with `-script` or `-scriptm1`.

`-keepRunning:EnableKeepRunning`

- If specified, as either `true` or `false`, it enables or disables REPL mode after all `-script` or `-execute` values have been processed.

`-script:ScriptFile`

- If specified, runs Kusto.Cli in script mode. The specified script file is loaded and the queries or commands in it are run sequentially. Newlines are used to delimit queries/commands, except when lines end with a `&` or `&&` combination, as explained below. This switch can't be used together with `-execute`.

`-scriptm1:ScriptFile`

- If specified, runs Kusto.Cli in script mode. The specified script file is loaded and the queries or commands in it are run sequentially. The entire script file is considered a single query or command. This switch can't be used together with `-execute`.

`-scriptQuitOnError`: *QuitOnFirstScriptError*

- If enabled, Kusto.Cli will quit if a command or query in a script results with an error.  
If disabled, script execution will continue despite errors. By default this switch is enabled.

`-echo`: *EnableEchoMode*

- If specified, as either `true` or `false`, it enables or disables echo mode. When echo mode is enabled, every query or command is repeated in the output.

`-transcript`: *TranscriptFile*

- If specified, writes program output to *TranscriptFile*.

`-logToConsole`: *EnableLogToConsole*

- If specified, as either `true` or `false`, it enables or disables displaying the program output on the console.

`-lineMode`: *EnableLineMode*

- If specified, switches between the default line input mode, when set to `true`, and the block input mode, when set to `false`. See below for an explanation of these two modes, which determine how newlines are treated.

## Example

```
Kusto.Cli.exe "https://help.kusto.windows.net/Samples;Fed=true" -  
script:"c:\mycommands.txt"
```

### ⓘ Note

There should be no space between the colon and the argument value

## Directives

Kusto.Cli runs a number of directives in the tool instead of sending them to the service for processing.

Directive	Description
-----------	-------------

Directive	Description
? #h #help	Get a short help message
q #quit #exit	Exit the tool
#a #abort	Exit the tool abortively
#clip	The results of the next query or command will be copied to the clipboard
#cls	Clear the console screen
#connect [ConnectionString]	Connects to a different Kusto service (if <i>ConnectionString</i> is omitted, the current one will be displayed)
#crp [Name [= Value]]	Sets the value of a client request property, or just displays it, or displays all values
#crp (-list   -doc) [Prefix]	Lists client request properties, by prefix, or all
#dbcontext [DatabaseName]	Changes the "context" database used by queries and commands to <i>DatabaseName</i> . If omitted, the current context displays
ke Text	Sends the specified text to a running Kusto.Explorer process
#loop Count Text	Runs the text a number of times
#qp [Name [= Value]]	Sets the value of a query parameter, or just displays it, or displays all values. Single/double quotes at beginning/end will be trimmed
#save Filename	The results of the next query or command will be saved to the indicated CSV file
#script Filename	Executes the indicated script
#scriptm1 Filename	Executes the indicated multiline script

## Line input mode and block input mode

By default, Kusto.Cli runs in **line input mode**. Each newline character is interpreted as a delimiter between queries/commands, and the line is immediately sent for execution.

In this mode, you can break a long query or command into multiple lines. The `&` character as the last character of a line, before the newline, causes Kusto.Cli to continue reading the next line. The `&&` character as the last character of a line, before the newline, causes Kusto.Cli to ignore the newline and continue reading the next line.

Kusto.Cli also supports running in **block input mode**. By using either the command-line switch `-lineMode:false`, or by using the directive `#blockmode`, you can instruct Kusto.Cli to assume every line is a continuation of the previous line, so that queries and commands are delimited by an empty input line only.

## Comments

Kusto.Cli interprets a `//` string that begins new line as a comment line. It ignores the rest of the line and continues reading the next line.

## Tool-only options

Commands	Effect	Currently
<code>#timeon #timeoff</code>	enable/disable option <code>timing</code> : Display the time requests took	TRUE
<code>#tableon #tableoff</code>	enable/disable option <code>tableView</code> : Format results sets as tables	TRUE
<code>#marson #marsoff</code>	enable/disable option <code>marsView</code> : Display the second-to-last result sets	FALSE
<code>#resultson #resultsoff</code>	enable/disable option <code>outputResultsSet</code> : Display the result sets	TRUE
<code>#prettyon #prettyoff</code>	enable/disable option <code>prettyErrors</code> : Clean up errors	TRUE
<code>#markdownon #markdownoff</code>	enable/disable option <code>markdownView</code> : Format tables as MarkDown	FALSE
<code>#progressiveon #progressiveoff</code>	enable/disable option <code>progressiveView</code> : Ask for and display progressive results	FALSE
<code>#linemode #blockmode</code>	enable/disable option <code>lineMode</code> : Single-line input mode	TRUE

Commands	Effect	Default

Commands	Effect	Default
#cridon #cridoff	(enable disable option <code>crid</code> : Display the ClientRequestId before sending the request)	FALSE
#csvheaderson #csvheadersoff	(enable disable option <code>csvHeaders</code> : Include headers in CSV output)	TRUE
#focuson #focusoff	(enable disable option <code>focus</code> : Remove all the extra fluff and focus on the right stuff)	FALSE
#linemode #blockmode	(enable disable option <code>lineMode</code> : Single-line input mode)	TRUE
#markdownon #markdownoff	(enable disable option <code>markdownView</code> : Format tables as MarkDown)	FALSE
#marson #marsoff	(enable disable option <code>marsView</code> : Display the second-to-last result sets)	FALSE
#prettyon #prettyoff	(enable disable option <code>prettyErrors</code> : Clean up errors)	TRUE
#querystreamingon #querystreamingoff	(enable disable option <code>queryStreaming</code> : Use the queryStreaming endpoint (Kusto team only))	FALSE
#resultson #resultsoff	(enable disable option <code>outputResultsSet</code> : Display the result sets)	TRUE
#tableon #tableoff	(enable disable option <code>tableView</code> : Format results sets as tables)	TRUE
#timeon #timeoff	(enable disable option <code>timing</code> : Display the amount of time that the requests took)	TRUE
#typeon #typeoff	(enable disable option <code>typeView</code> : Display the type of each column in table view. Forces Streaming=true)	TRUE
#v2protocolon #v2protocoloff	(enable disable option <code>v2protocol</code> : Use the v2 query protocol, not v1)	TRUE

## Use Kusto.Cli to export results as CSV

Kusto.Cli has a special client-side command, `#save` that exports the `next` query results to a local file in CSV format. For example, the following line will export 10 records out of the `StormEvents` table into the `help.kusto.windows.net` cluster, `Samples` database:

```
Kusto.Cli.exe @help/Samples -execute:"#save c:\temp\test.log" -  
execute:"StormEvents | take 10"
```

## Use Kusto.Cli to control a running instance of Kusto.Explorer

You can instruct Kusto.Cli to communicate with the "primary" instance of Kusto.Explorer running on the machine, and send it queries. This mechanism can be useful for programs that want to run a number of queries, but don't want to start the Kusto.Explorer process repeatedly. In the following example, Kusto.Cli is used to run a query against the help cluster:

```
#connect cluster('help').database('Samples')  
  
#ke StormEvents | count
```

The syntax is simple: `#ke`, followed by whitespace, and the query to run. The query is then sent to the primary instance of Kusto.Explorer, if one exists, with the current cluster/database set in Kusto.Cli.

---

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# Kusto (KQL) extension for Azure Data Studio (Preview)

Article • 03/09/2023

The Kusto (KQL) extension for Azure Data Studio enables you to connect and query to Azure Data Explorer clusters.

Users can write and run KQL queries and author notebooks with the Kusto kernel complete with IntelliSense.

By enabling the native Kusto (KQL) experience in Azure Data Studio, data engineers, data scientists, and data analysts can quickly observe trends and anomalies against massive amounts of data stored in Azure Data Explorer.

This extension is currently in preview.

## Prerequisites

If you don't have an Azure subscription, create a free Azure account [↗](#) before you begin.

The following prerequisites are also required:

- Azure Data Studio installed.
- An Azure Data Explorer cluster and database.

## Install the Kusto (KQL) extension

To install the Kusto (KQL) extension in Azure Data Studio, follow the steps below.

1. Open the extensions manager in Azure Data Studio. You can either select the extensions icon or select **Extensions** in the View menu.
2. Type in *Kusto* in the search bar.
3. Select the **Kusto (KQL)** extension and view its details.
4. Select **Install**.



## How to connect to an Azure Data Explorer cluster

### Find your Azure Data Explorer cluster

Find your Azure Data Explorer cluster in the Azure portal [↗](#), then find the URI for the cluster.

The screenshot shows the Azure portal interface for an Azure Data Explorer cluster named 'mydataexplorercluster'. The left sidebar has options like Overview, Activity log, Access control (IAM), Tags, and Diagnose and solve problems. The main pane displays cluster details: Location (West US), Subscription (change), Subscription ID, State (Running), URI (<https://mydataexplorercluster.westus.kusto.windows.net>), Data Ingestion URI (<https://ingest-mydataexplorercluster.westus.kusto.windows.net>), and Compute specifications (D13\_v2). The 'URI' field is highlighted with a yellow box.

However, you can get started immediately using the `help.kusto.windows.net` cluster.

For this article, we're using data from the `help.kusto.windows.net` cluster for samples.

### Connection details

To set up an Azure Data Explorer cluster to connect to, follow the steps below.

1. Select **New connection** from the **Connections** pane.
2. Fill in the **Connection Details** information.
  - a. For **Connection type**, select *Kusto*.
  - b. For **Cluster**, enter in your Azure Data Explorer cluster address (for example, <https://mydataexplorercluster.kusto.windows.net>).
  - c. For **Authentication Type**, use the default - *Azure Active Directory - Universal with MFA account*.
  - d. For **Account**, use your account information.

- e. For **Database**, use *Default*. If your account does not have access to the *Default* database, you can use any database that you have access to.
- f. For **Server Group**, use *Default*.
  - i. You can use this field to organize your servers in a specific group.
- g. For **Name (optional)**, leave blank.
  - i. You can use this field to give your server an alias.

**Connection Details**

Connection type	Azure Data Explorer (Kusto)	▼	
Cluster *	<a href="https://mydataexplorercluster.kusto.windows.net">https://mydataexplorercluster.kusto.windows.net</a>		
Authentication type	Azure Active Directory - Universal with MFA support		▼
Account	yourusername@contoso.com		▼
Azure AD tenant	Contoso		▼
Database	<Default>		▼
Server group	<Default>		▼
Name (optional)			
<a href="#">Advanced...</a>			
<a href="#" style="background-color: #54677A; color: white; padding: 5px 10px;">Connect</a> <a href="#" style="border: 1px solid #ccc; padding: 5px 10px;">Cancel</a>			

## How to query an Azure Data Explorer database in Azure Data Studio

Now that you have set up a connection to your Azure Data Explorer cluster, you can query your database(s) using Kusto (KQL).

To create a new query tab, you can either select **File > New Query**, use **Ctrl + N**, or right-click the database and select **New Query**.

Once you have your new query tab open, then enter your Kusto query.

Here are some samples of KQL queries:

Kusto

```
StormEvents  
| limit 1000
```

Kusto

```
StormEvents  
| where EventType == "Waterspout"
```

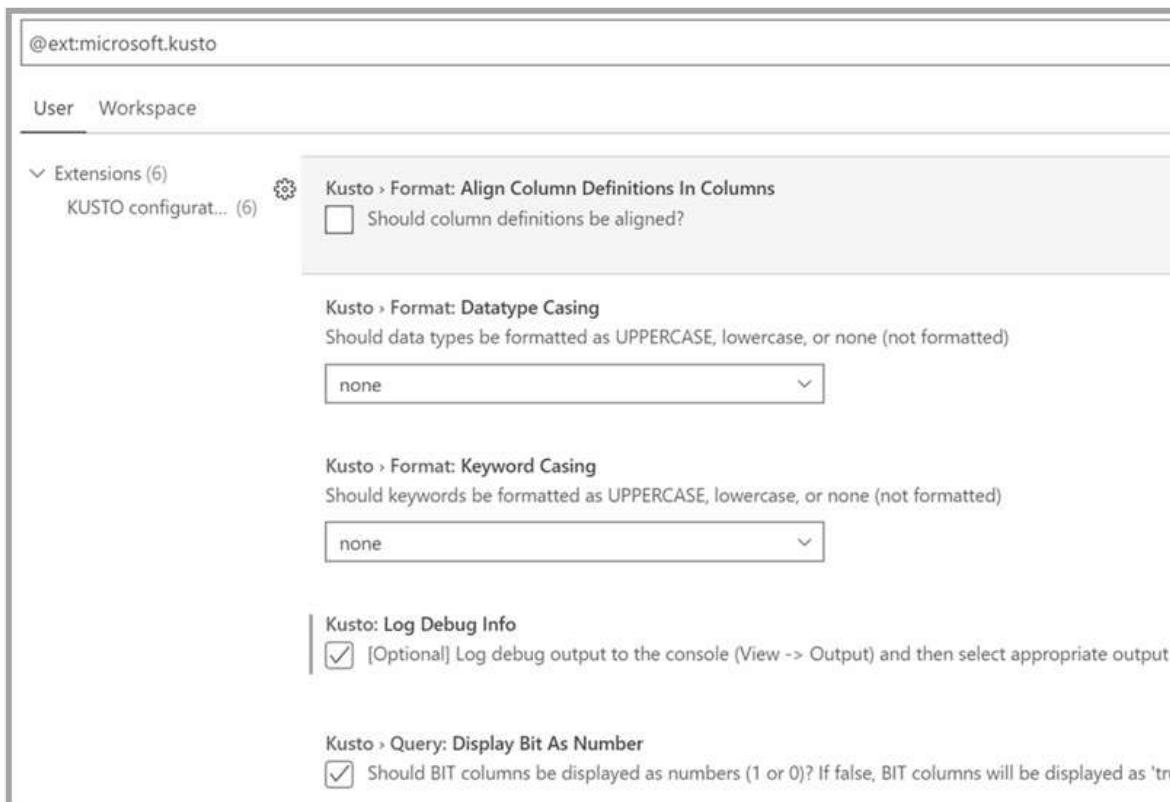
For more information about writing KQL queries, visit [Write queries for Azure Data Explorer](#)

## View extension settings

To change the settings for the Kusto extension, follow the steps below.

1. Open the extension manager in Azure Data Studio. You can either select the extensions icon or select **Extensions** in the View menu.
2. Find the **Kusto (KQL)** extension.
3. Select the **Manage** icon.
4. Select the **Extension Settings** icon.

The extensions settings look like this:



## SandDance visualization

The SandDance extension with the Kusto (KQL) extension in Azure Data Studio bring rich interactive visualization together. From the KQL query result set, select the **Visualizer** button to launch SandDance ↗.

The screenshot shows the Azure Data Studio interface with a KQL query running in a query editor window. The query is:

```

1 StormEvents
2 | extend Damage = DamageProperty + DamageCrops, EventTime = bin(StartTime, 7d)
3 | where State in ("TEXAS", "WASHINGTON", "FLORIDA")
4 | project EventTime, State, EventType, Damage
    
```

The results pane displays a table of event data:

	EventTime	State	EventType	Damage
1	2007-09-17 00:00:00	FLORIDA	Heavy Rain	0
2	2007-09-17 00:00:00	FLORIDA	Tornado	6200000
3	2007-12-10 00:00:00	FLORIDA	Thunderstorm Wind	5000
4	2007-12-10 00:00:00	FLORIDA	Thunderstorm Wind	0
5	2007-12-10 00:00:00	FLORIDA	Thunderstorm Wind	10000
6	2007-12-24 00:00:00	FLORIDA	Funnel Cloud	0
7	2007-12-10 00:00:00	FLORIDA	Thunderstorm Wind	6000
8	2007-12-17 00:00:00	FLORIDA	Tornado	15000
9	2007-12-10 00:00:00	TEXAS	Thunderstorm Wind	5000
10	2007-12-10 00:00:00	TEXAS	Tornado	12000
11	2007-12-17 00:00:00	TEXAS	Hail	0

## Known issues

Details	Workaround
In Kusto notebook, Changing a database connection on a saved alias connection is stuck after an error in code cell execution ↗	Close and reopen the Notebook, then connect to the right cluster with the database
In Kusto Notebook, changing a database connection on a non-saved alias connection doesn't work ↗	Create a new connection from Connection Viewlet and save it with an alias. Then create a new notebook and connect to the newly saved connection)
In Kusto Notebook, the database dropdown isn't populated when creating a new ADX connection ↗	Create a new connection from Connection Viewlet and save it with an alias. Then create a new notebook and connect to the newly saved connection)

You can file a feature request ↗ to provide feedback to the product team.

You can file a bug ↗ to provide feedback to the product team.

## Next steps

- Create and run a Kusto notebook
- Kqlmagic notebook in Azure Data Studio
- SQL to Kusto cheat sheet
- What is Azure Data Explorer?
- Using SandDance visualizations ↗

# Management commands overview

Article • 06/19/2023

This article describes the management commands, also known as control commands, used to manage Kusto. Management commands are requests to the service to retrieve information that is not necessarily data in the database tables, or to modify the service state, etc.

## Differentiating management commands from queries

Kusto uses three mechanisms to differentiate queries and management commands: at the language level, at the protocol level, and at the API level. This is done for security purposes.

At the language level, the first character of the text of a request determines if the request is a management command or a query. Management commands must start with the dot (.) character, and no query may start by that character.

At the protocol level, different HTTP/HTTPS endpoints are used for control commands as opposed to queries.

At the API level, different functions are used to send management commands as opposed to queries.

## Combining queries and management commands

Management commands can reference queries (but not vice-versa) or other management commands. There are several supported scenarios:

1. **AdminThenQuery**: A management command is executed, and its result (represented as a temporary data table) serves as the input to a query.
2. **AdminFromQuery**: Either a query or a `.show` admin command is executed, and its result (represented as a temporary data table) serves as the input to a management command.

Note that in all cases, the entire combination is technically a management command, not a query, so the text of the request must start with a dot (.) character, and the request must be sent to the management endpoint of the service.

Also note that query statements appear within the query part of the text (they can't precede the command itself).

### ⓘ Note

Don't run *AdminThenQuery* operations too frequently. *AdminThenQuery* pipes the result set of the management command and applies filters/aggregations on it.

- For example: `.show ... | where ... | summarize ...`
- When running something like: `.show cluster extents | count` (emphasis on the `| count`), Kusto first prepares a data table that holds all details of all extents in the cluster. The system then sends that in-memory-only table to the Kusto engine to do the count. The system actually works hard in an unoptimized path to give you such a trivial answer.

**AdminThenQuery** is indicated in one of two ways:

1. By using a pipe (`|`) character, the query therefore treats the results of the management command as if it were any other data-producing query operator.
2. By using a semicolon (`;`) character, which then introduces the results of the management command into a special symbol called `$command_results`, that one may then use in the query any number of times.

For example:

Kusto

```
// 1. Using pipe: Count how many tables are in the database-in-scope:  
.show tables  
| count  
  
// 2. Using semicolon: Count how many tables are in the database-in-scope:  
.show tables;  
$command_results  
| count  
  
// 3. Using semicolon, and including a let statement:  
.show tables;  
let useless=(n:string){strcat(n,'-', 'useless')};  
$command_results | extend LastColumn=useless(TableName)
```

**AdminFromQuery** is indicated by the `<|` character combination. For example, in the following we first execute a query that produces a table with a single column (named

`str` of type `string`) and a single row, and write it as the table name `MyTable` in the database in context:

Kusto

```
.set MyTable <|
let text="Hello, World!";
print str=text
```

---

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A