Viewing BSON Document Data

When connected to data sources that return structured document data (such as JSON or BSON), a regular two-dimensional grid is not ideal for viewing the data. For this type of data, you get a panel that contains three different views where you can choose the one that serves your purpose best based on what data you're looking at. As a rule of thumb, the **Nested View** will yield the best overview, the **Tree View** will be the most flexible and efficient when viewing complex structures and/or large data sets, and the **Text View** will present the most details.

In the first release, some of the features of the regular data grids are missing. In particular, you cannot reorder columns or export data from the grid. We hope to provide these capabilities in the near future.

Nested View

Click the nested view button (**A**) to show a view that organizes data in expandable column header groups based on the data structure. You can expand /collapse headers (**B**) to focus on the data you are interested in, either by double-clicking the header or via the popup menu (right-click the header). Select a cell in the grid (**C**) to show the details as a key-value tree representation in the right-hand panel. Click any cell in the tree to show its details in the text panel below (**D**).

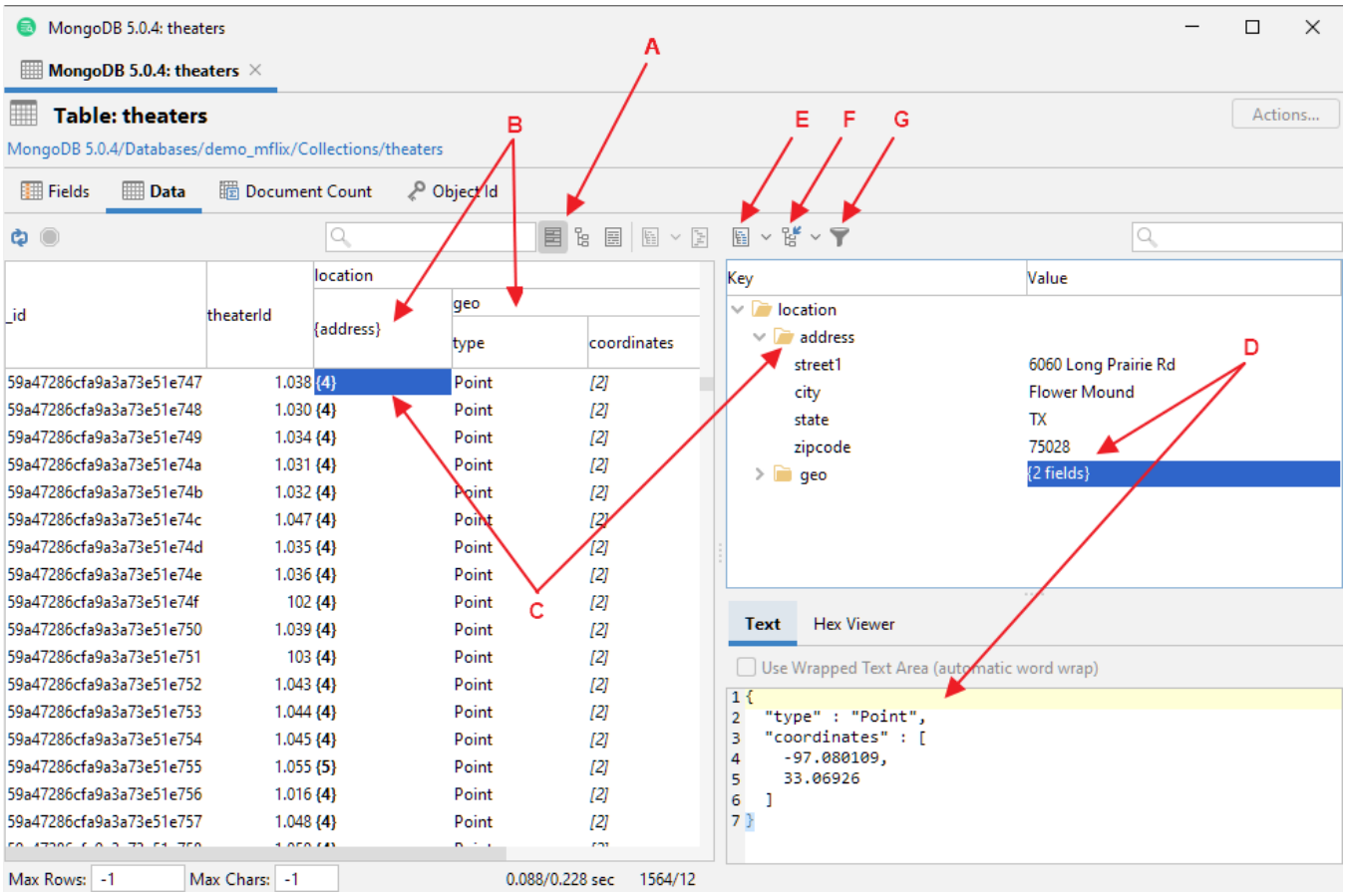
Choose how to show the details in the tree view:

- **Collapsed Rows (E)**
  Choose to see collapsed rows as the number of elements it contains, the JSON representation (abridged), or nothing (blank).
- **Scope (F)**
  Choose if you want to see the entire **row**, all columns belonging to the same **root** column group, or all columns belonging to the **closest** singular column group. If you choose the **closest** group, the actual group varies with the structure; the root of the tree will be the closest column group that is not part of an array. In the picture below, the tree starts at the **root** column (*location*).
- **Auto Expand (F)**
  Choose to automatically expand the tree to the level of the grid cell you selected.
- **Auto Filter (G)**
  Choose to filter on the selected cell, the tree will only show the cell you selected. If the cell is part of an array, you will see all occurrences of the cell.

In the text panel (**D**), a leaf cell is presented as plain text and you can choose to word-wrap the text to make it readable, whereas a container cell (a JSON structure) is automatically formatted based on the JSON syntax. You cannot word-wrap the formatted data.

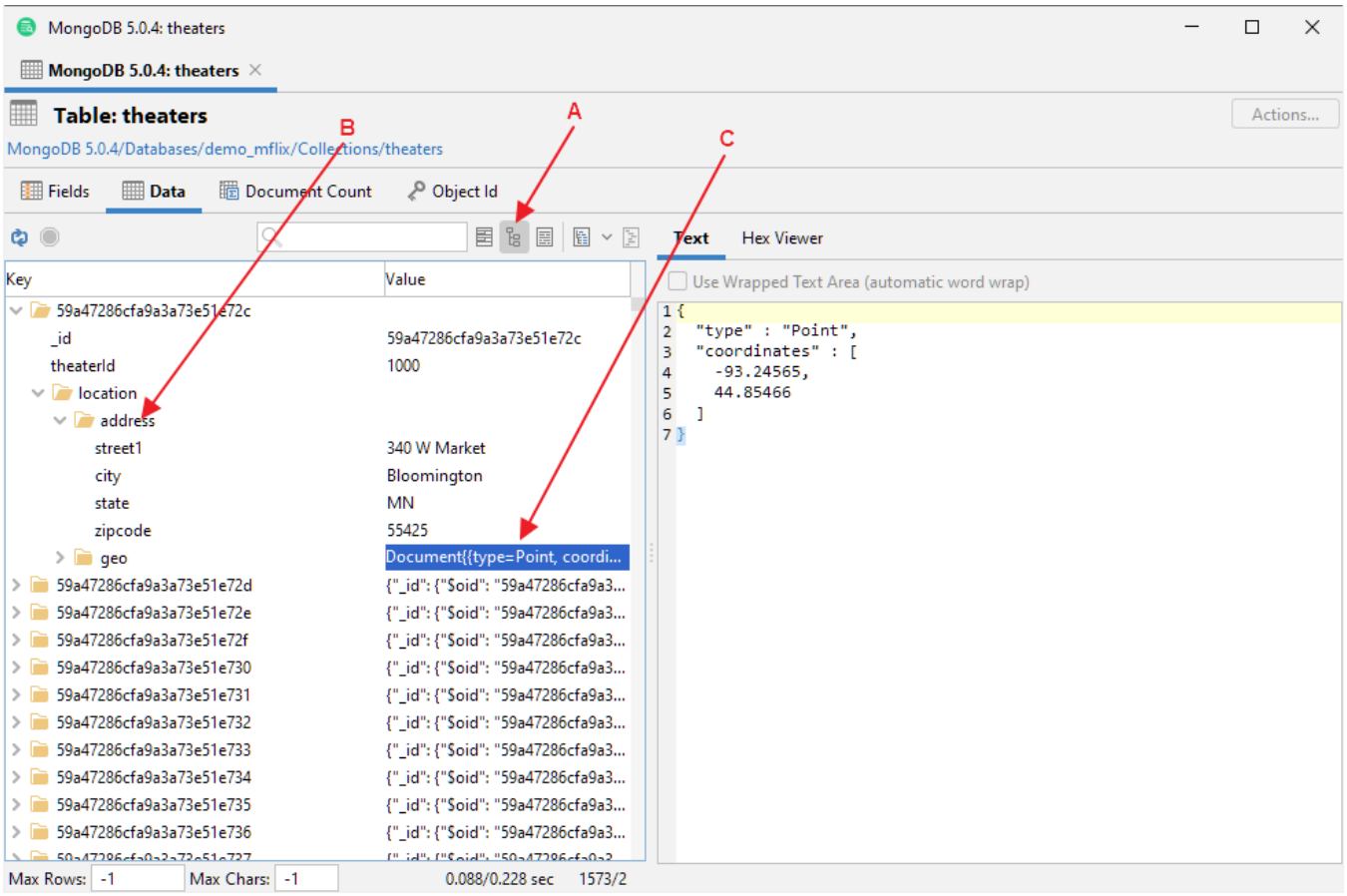
In both cases, you can view the hexadecimal representation of the data.

Showing complex structures in the nested view may be slow since it places a heavy load on the algorithm for calculating row and column sizes.
You may have to confirm the action before opening this view on some data.



Tree View

Click the tree view button (**A**) to show the data in a vertical fashion where each row represents an element. This view is the most efficient way to browse complex data since the structure adapts to each row. Like the details tree in the nested view, you can expand/collapse the rows (**B**), choose how to present collapsed rows (**C**), and select a cell in the tree to show its contents in the text panel to the right (**D**).

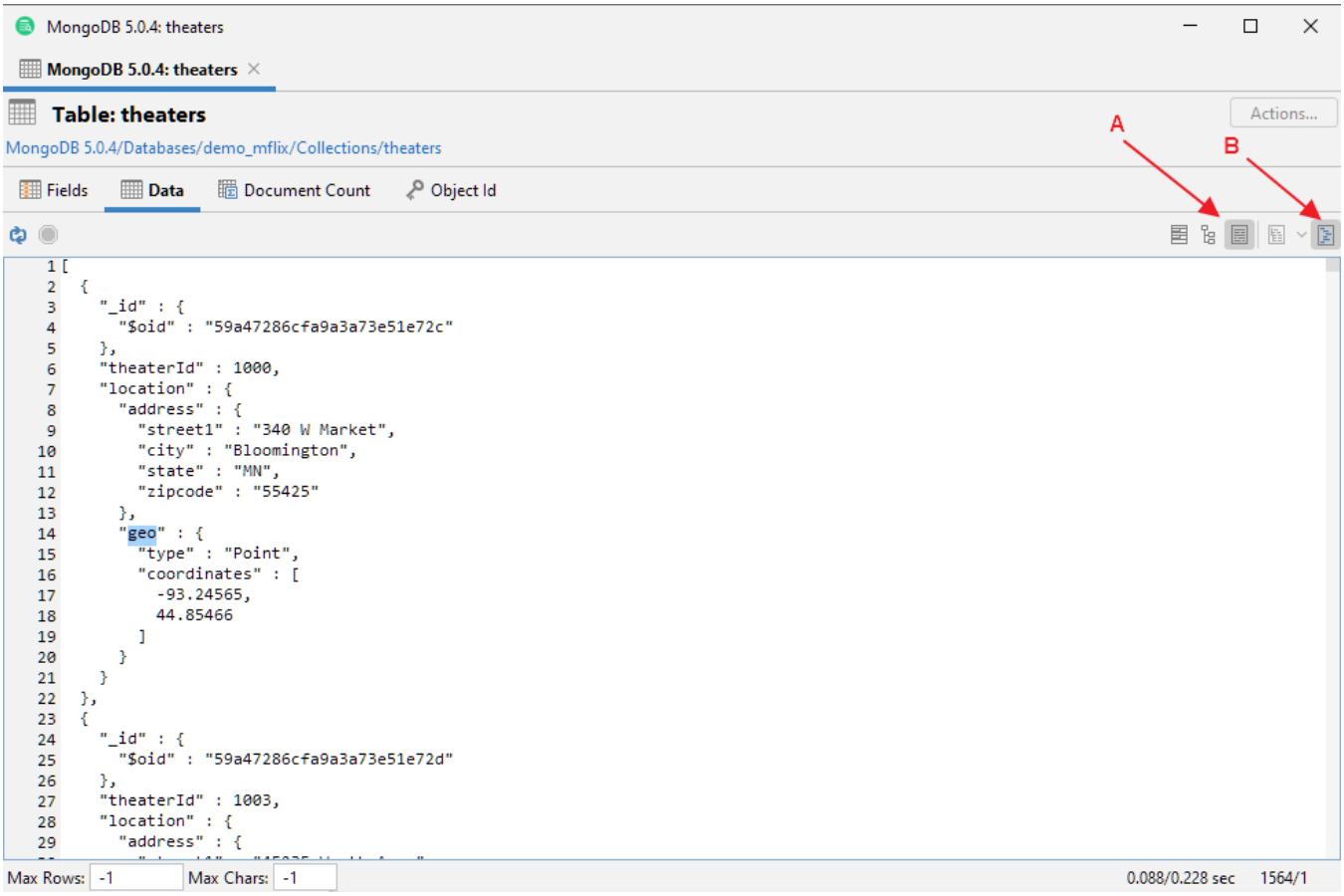


Text View

Click the text view button (**A**) to view the data as returned by the database, one row for each document.

Click the **format** button (**B**) to format/unformat the data based on JSON syntax.

A document data set is often larger than an average relational table; loading a complete data set may consume a lot of memory, and formatting /unformatting a large data set may be slow.


MongoDB 5.0.4: theaters
MongoDB 5.0.4: theaters
Table: theaters
MongoDB 5.0.4/Databases/demo_mflix/Collections/theaters
Actions...
A
B
Fields
Data
Document Count
Object Id
[
{
"_id" : {
"$oid" : "59a47286cfa9a3a73e51e72c"
},
"theaterId" : 1000,
"location" : {
"address" : {
"street1" : "340 W Market",
"city" : "Bloomington",
"state" : "MN",
"zipcode" : "55425"
},
"geo" : {
"type" : "Point",
"coordinates" : [
-93.24565,
44.85466
]
}
}
},
{
"_id" : {
"$oid" : "59a47286cfa9a3a73e51e72d"
},
"theaterId" : 1003,
"location" : {
"address" : {
Max Rows: -1
Max Chars: -1
0.088/0.228 sec
1564/1