

```
1  # DUCKDB PARQUET EXTENSION USER GUIDE, March 29, 2025
2
3  ## For Non-Developers Using the Extension in Qlik Cloud
4
5  **March 29, 2025**
6
7  ---
8
9  ## 1. Introduction
10
11  The **DuckDB Parquet Extension** enables Qlik Cloud users to connect to public Parquet files, query them using DuckDB's in-browser SQL engine, and display the results in an interactive, paginated table. This extension seamlessly integrates Qlik's selection capabilities with DuckDB's high-performance data processing, providing a fast and user-friendly way to explore data directly in your browser.
12
13  ### Key Features
14  - **Load and Query Parquet Files**: Connect to publicly accessible Parquet files via URL and query them using DuckDB's SQL engine.
15  - **Interactive Filtering**: Filter data using Qlik field selections, a manual search box, or custom SQL queries.
16  - **Paginated Table**: Display results with customizable row counts and pagination controls.
17  - **Pivot Tables**: Group data by multiple fields with collapsible levels for hierarchical analysis.
18  - **Sorting**: Sort data by columns, either across the entire table or within the current selection.
19  - **Column Statistics**: Analyze data quality with a "Stats" button to view metrics like null counts, unique values, and outliers.
20  - **Refresh Data**: Use the "Refresh Data" button to reload data from the Parquet file on demand.
21  - **KPIs**: Display key metrics such as total records, matched records, and unmatched records.
22  - **Customizable Appearance**: Adjust table styles, layout, and messages via the properties panel.
23  - **Theme Integration**: Automatically applies Qlik Sense's theme for a consistent look and feel.
24  - **Accessibility**: Includes ARIA attributes (e.g., `aria-live`, `aria-label`) for better screen reader support.
25  - **Dedicated to Stuart Wannop**: A tribute to a Qlik Champion, with editable copyright and dedication text.
26
27  This extension has been professionally developed with rigorous Qlik verifications, security checks, and user-friendly controls to ensure reliability and performance.
28
29  ---
30
31  ## 2. Installation
32
33  ### Download the Extension
34  - Obtain the extension ZIP file from your administrator or the source repository. The ZIP file includes the `duckdb2.js` script, `duckdb2.qext` configuration file, and required dependencies (e.g., DuckDB WASM and Apache Arrow libraries).
35
36  ### Upload to Qlik Cloud
37  1. Log in to Qlik Cloud.
38  2. Navigate to the **Management Console** > **Extensions** section.
39  3. Click **Add**, then upload the ZIP file.
40  4. Verify that the extension appears in the list as "DuckDB Parquet Extension."
41
42  ### Add to a Sheet
43  1. Open a Qlik app in Qlik Cloud.
44  2. Edit a sheet and locate the **Custom Objects** panel.
```

```
45 3. Drag the **DuckDB Parquet Extension** onto the sheet.
46
47 ---
48
49 ## 3. Configuration
50
51 Configure the extension using the properties panel in Qlik Cloud. Below are the available sections and options, updated to reflect the
latest features and controls.
52
53 ### 3.1 Data Source
54 - **Data Files**:
55   - **File URL**: Enter the full URL to a publicly accessible file (e.g., `https://example.com/data.parquet`). Supports Parquet, CSV, JSON,
and Arrow formats.
56   - **File Type**: Select the file type (Parquet, CSV, JSON, or Arrow) from the dropdown.
57   - **File Name (optional)**: Specify a custom name for the file. If blank, it's derived from the URL.
58   - **Add Multiple Files**: Click "Add" to include additional files; the extension will union them into a single dataset.
59 - **Show Filtered Record Count**: Toggle to display counts of matched and unmatched records based on Qlik selections (default: On).
60
61 ### 3.2 Dependencies
62 - **DuckDB WASM URL**: URL to the DuckDB WASM file (e.g., `https://cdn.jsdelivr.net/npm/duckdb-wasm@1.0.0/dist/duckdb.wasm`). Must start
with `https://`.
63 - **DuckDB Worker URL**: URL to the DuckDB Worker script (e.g.,
`https://cdn.jsdelivr.net/npm/duckdb-wasm@1.0.0/dist/duckdb-browser.worker.js`). Must start with `https://`.
64 - **Arrow JS URL**: URL to the Apache Arrow library (e.g., `https://cdn.jsdelivr.net/npm/apache-arrow@12.0.0/arrow.min.js`). Must start
with `https://`.
65 - **Security Note**: The extension validates that all URLs use `https://` to ensure secure connections. Missing or invalid URLs will
display an error message: "Error: Missing DuckDB or Arrow library URLs. Please configure them in the Dependencies section."
66
67 ### 3.3 Settings
68 - **Query Timeout (ms)**: Set the maximum time for queries to run (default: 60,000 ms; range: 1,000-120,000 ms). Prevents long-running
queries from hanging.
69 - **Max RAM (GB)**: Limit the memory DuckDB can use (default: 4 GB; range: 1-32 GB). Files exceeding this limit trigger a confirmation
prompt.
70 - **Enable Debug Logging**: Toggle to enable detailed console logs for troubleshooting (default: Off). Recommended for administrators only.
71 - **Max Pivot Fields**: Set the maximum number of pivot fields (default: 3; range: 1-10).
72 - **Max Custom Query Rows**: Limit the number of rows returned by custom SQL queries (default: 0, meaning no limit).
73 - **Sort Scope**: Choose whether sorting applies to the entire table or only selected records (default: Selected Records).
74 - **Loading Message**: Customize the message shown during data loading (default: "Enjoy this extension. During data load Take a moment to
reflect whilst your data loads.>").
75
76 ### 3.4 Pagination
77 - **Rows per Page**: Set the default number of rows per table page (default: 25).
78 - **Note**: Pagination is disabled when using pivot tables.
79
80 ### 3.5 Appearance
81 - **General**:
82   - **Show Titles**: Toggle to enable Qlik Sense's native rendering of the title, subtitle, and footnote (default: On). When enabled, these
are displayed outside the extension object.
```

```
83 - **Title**: Set the title (default: "Data Explorer"). Supports Qlik expressions (e.g., `=Count(Distinct Region)`).
84 - **Subtitle**: Set the subtitle (default: blank). Supports Qlik expressions.
85 - **Footnote**: Set the footnote (default: blank). Supports Qlik expressions.
86 - **Styles**:
87   - **Container Padding**: Padding around the table (default: 10px).
88   - **KPI Margin Bottom**: Space below KPI stats (default: 10px).
89   - **KPI Font Weight**: Font weight for KPIs (default: bold).
90   - **KPI Color**: Text color for KPIs (default: #333).
91   - **Table Max Height**: Maximum table height (default: 600px).
92   - **Table Max Width**: Maximum table width (default: 100%).
93   - **Table Border**: Border style for the table (default: 1px solid #ccc).
94   - **Header Border**: Border for table headers (default: 1px solid black).
95   - **Header Padding**: Padding for headers (default: 8px).
96   - **Header Background**: Background color for headers (default: #f5f5f5).
97   - **Cell Border**: Border for table cells (default: 1px solid black).
98   - **Cell Padding**: Padding for cells (default: 8px).
99   - **Button Margin Top**: Space above pagination buttons (default: 10px).
100  - **Button Border**: Border style for buttons (default: 1px solid black).
101  - **Bold First Pivot Count**: Toggle to bold the count column in single-level pivot tables (default: On).
102
103 ### 3.6 Field Mappings
104 - **Qlik Field Name**: The field name in your Qlik app (e.g., `Region`).
105 - **Data Column Name**: The corresponding column in the file (e.g., `region_name`). Case-sensitive.
106 - **Add Multiple Mappings**: Click "Add Mapping" to link additional fields. If no mappings are defined, a warning appears: "No Qlik field mappings defined—running as standalone query tool."
107
108 ### 3.7 Search Settings
109 - **Search Placeholder**: Text shown in the search box (default: "Search or enter SQL query").
110
111 ### 3.8 Copyright & Dedication
112 - **Tribute**: Dedication message (default: "This Qlik Extension is shared in memory of Stuart Wannop").
113 - **Description**: Additional context (default: "A Qlik Champion who would have enjoyed seeing this type of embedded object").
114 - **Disclaimer**: Legal note (default: "Use of this extension is entirely at your own risk; no warranties are implied or inferred").
115 - **Note**: This section is displayed at the bottom of the extension, below the table.
116
117 ---
118
119 ## 4. Usage
120
121 ### Add the Extension
122 1. Drag the **DuckDB Parquet Extension** onto a sheet in your Qlik app.
123 2. Resize the object as needed to fit your layout.
124
125 ### Configure
126 1. In the properties panel, enter the file URL under **Data Source** (e.g., `https://example.com/data.parquet`).
127 2. Set the file type (e.g., Parquet, CSV, JSON, or Arrow).
128 3. Add field mappings to link Qlik fields to file columns (e.g., Qlik field `Region` to Parquet column `region_name`).
129 4. Adjust settings like rows per page, styles, and query timeout as needed.
```

```
130
131 ### Interact
132 - **Refresh Data Button**: Click to reload data from the file. The button temporarily changes to "Refreshing..." during the operation.
133 - **Search Box**:
134   - Type a search term and press Enter to filter across mapped fields (e.g., "West" to filter rows containing "West").
135   - Alternatively, enter a SQL query (e.g., `SELECT * FROM all_data WHERE region_name = 'West'`) and press Enter to execute it.
136 - **Pivot Tables**:
137   - Click "Add Pivot" to group data by a field (e.g., `region_name`).
138   - Add up to the maximum pivot fields (default: 3) to create a hierarchical view.
139   - Click the ►/▼ icons to expand or collapse levels.
140 - **Sorting**:
141   - Click a column header to sort by that column.
142   - Click again to toggle between ascending and descending order.
143   - Sorting scope (entire table or selected records) is set in the properties panel.
144 - **Stats Button**:
145   - Select "All Columns" or a specific column from the dropdown and click "Stats" to view data quality metrics (e.g., null count, unique values, min/max, outliers for numeric columns).
146 - **Pagination**:
147   - Use the "Back" and "Next" buttons to navigate through pages of results.
148   - Pagination is disabled when using pivot tables.
149 - **Qlik Selections**:
150   - Select values in mapped Qlik fields (e.g., select "West" in the `Region` field) to filter the table.
151   - KPIs update to show matched and unmatched record counts.
152 - **View Results**:
153   - The table displays the data with KPIs above it (e.g., Total Records, Last Refreshed, Matched Records, Unmatched Records).
154   - Columns with active Qlik selections are bolded and underlined.
155
156 ---
157
158 ## 5. Professional Development and Verifications
159
160 This extension has been developed with a focus on reliability, security, and user experience. Below are the key verifications and controls implemented to ensure professional quality.
161
162 ### 5.1 Qlik Verifications and Updates
163 - **Syntax Validation**:
164   - The `duckdb2.js` script was validated using ESLint to ensure no syntax errors. A specific issue in the `initDuckDB` function (`CREATE OR REMOVE TABLE` was corrected to `CREATE OR REPLACE TABLE`) was fixed.
165 - **Loading Message Fix**:
166   - Addressed a flash of the "data loading" message after clicking the "Refresh Data" button by introducing a `justRefreshed` flag. This ensures the message only appears when necessary, improving the user experience.
167 - **Compatibility**:
168   - Tested with Qlik Sense versions up to March 2025. The extension uses modern JavaScript features (`Promise`, `async/await`, `fetch`) supported in recent Qlik Sense releases. For older versions (pre-2018), compatibility adjustments may be needed.
169 - **Error Handling**:
170   - Added detailed error messages for common issues (e.g., "Error: Missing DuckDB or Arrow library URLs," "No valid files could be loaded," "File type mismatch").
171   - Historical Qlik Community issues (e.g., 442 error in QlikView 11 SR1) were reviewed, and similar issues in Qlik Sense were mitigated
```

```
    through proper extension loading checks.
172
173 ### 5.2 Security Checks and Controls
174 - **URL Validation**:
175   - All dependency URLs (DuckDB WASM, Worker, Arrow JS) must start with `https://` to ensure secure connections. Invalid URLs trigger an
    error: "Error: Missing DuckDB or Arrow library URLs."
176 - **File Type Validation**:
177   - The extension checks if the declared file type (e.g., Parquet) matches the URL extension (e.g., `.parquet`). Mismatches result in an
    error: "File type mismatch: [file URL] (declared as [type], appears to be [inferred type])."
178 - **File Size Limits**:
179   - Files exceeding the Max RAM setting (default: 4 GB) trigger a confirmation prompt: "File size ([size] MB) exceeds RAM limit ([maxRam]
    GB). Continue loading [file URL]?"
180   - Skipped files are listed in a warning: "Warning: Skipped [count] file(s) due to size exceeding [maxRam]GB: [file names]."
```

```
181 - **Network and CORS Checks**:
182   - The extension performs a `HEAD` request to verify file accessibility before loading. Failed requests result in a warning: "Skipped
    [file URL] (HEAD failed)."
```

```
183   - Fetch failures (e.g., due to CORS or network issues) display an error: "Error: Failed to fetch file [file URL]. Check the URL or
    network."
```

```
184 - **Query Timeout**:
185   - A configurable query timeout (default: 60,000 ms) prevents long-running queries from hanging the browser. Timeout errors display as: "
    [File type] load timeout for [file name]."
```

```
186 - **Trusted Sources**:
187   - Only use trusted file URLs, as data is processed in the browser. The extension does not execute server-side code, reducing server-side
    security risks.
188
189 ### 5.3 Performance Controls
190 - **Max RAM Setting**: Limits memory usage to prevent browser crashes (default: 4 GB).
191 - **Max Query Rows**: Caps the number of rows returned by custom SQL queries to avoid performance issues (default: 0, meaning no limit).
192 - **Pagination**: Reduces rendering load by displaying data in pages (default: 25 rows per page).
193 - **Debug Logging**: Optional debug logs provide detailed timing and error information for administrators (e.g., "Total load time: [time]
    ms").
194
195 ### 5.4 Warning and Error Messages
196 - **Missing Dependencies**: "Error: Missing DuckDB or Arrow library URLs. Please configure them in the Dependencies section."
197 - **No Files Selected**: "Please select at least one file to display in the Data Source section."
198 - **No Valid Files Loaded**: "No valid files could be loaded. Check file URLs and permissions."
199 - **File Type Mismatch**: "Error: File type mismatch: [file URL] (declared as [type], appears to be [inferred type])."
```

```
200 - **Fetch Failure**: "Error: Failed to fetch file [file URL]. Check the URL or network."
```

```
201 - **No Mappings**: "No Qlik field mappings defined—running as standalone query tool."
```

```
202 - **Skipped Files**: "Warning: Skipped [count] file(s) due to size exceeding [maxRam]GB: [file names]."
```

```
203 - **SQL Errors**: "SQL Error: [error message]" (displayed in the table when a custom SQL query fails).
204
205 ---
206
207 ## 6. Troubleshooting
208
209 ### "Error Loading Data" Message
210 - **Check the File URL**:
```

```
211 - Ensure the URL is correct and publicly accessible (e.g., test in a browser or with `curl`).
212 - Verify the URL uses `https://` and returns a valid file.
213 - CORS Issues:
214 - Confirm the server hosting the file allows CORS access from Qlik Cloud. Contact your administrator to adjust CORS headers if needed.
215 - Network Issues:
216 - Check your internet connection. Open the browser's developer console (F12) and look for failed network requests in the "Network" tab.
217
218 Table Not Showing
219 - Field Mappings:
220 - Ensure field mappings match the file's column names (case-sensitive).
221 - If no mappings are defined, the extension runs in standalone mode, and a warning appears: "No Qlik field mappings defined—running as standalone query tool."
222 - Empty File:
223 - Verify the file contains data. Empty files display: "No matching records found."
224 - File Type Mismatch:
225 - Check for a "File type mismatch" error in the extension. Correct the file type in the properties panel to match the URL extension.
226
227 Slow Performance
228 - Reduce Rows per Page:
229 - Lower the "Rows per Page" setting (e.g., from 25 to 10) for faster rendering.
230 - Use Smaller Files:
231 - If possible, use a smaller file to reduce memory usage.
232 - Adjust Max RAM:
233 - Increase the "Max RAM" setting if your browser can handle more memory (e.g., from 4 GB to 8 GB).
234 - Query Timeout:
235 - If queries time out, increase the "Query Timeout" setting (e.g., from 60,000 ms to 90,000 ms).
236
237 Duplicate Title/Subtitle/Footnote
238 - This issue was fixed in the March 29, 2025 update. The extension now uses Qlik Sense's native rendering for the title, subtitle, and footnote (controlled by the "Show Titles" property). Ensure "Show Titles" is enabled to display these elements outside the object.
239
240 "Data Loading" Message Flashing
241 - This issue was resolved in the March 29, 2025 update. The "Refresh Data" button now prevents the loading message from flashing by caching the data and skipping unnecessary reloads.
242
243 Console Logs
244 - Normal Operation: Minimal logs are shown unless "Enable Debug Logging" is turned on. You may see warnings or errors if issues occur (e.g., "Fetch failed: [error]").
245 - Debug Mode: When enabled, detailed logs provide timing and error information (e.g., "Total load time: [time]ms"). Contact your administrator to review these logs if needed.
246
247 Extension Not Loading (442 Error)
248 - This issue was fixed in the March 29, 2025 update. A syntax error in the `duckdb2.qext` file was corrected, ensuring the extension loads properly. If the error persists:
249 - Verify the extension was uploaded correctly in the Qlik Management Console.
250 - Check the Qlik Sense server logs for more details (e.g., `C:\ProgramData\Qlik\Sense\Log\Engine`).
251
252 ---
```

```
253
254 ## 7. Technical Notes
255
256 ### Dependencies
257 - **DuckDB WASM**: Embedded SQL engine for querying files in the browser. Provides high-performance data processing.
258 - **Apache Arrow**: Data format for efficient in-memory processing, enabling fast data transfers and queries.
259
260 ### Browser Support
261 - Works in modern browsers (Chrome, Edge, Firefox, Safari, Opera) with WebAssembly support. Ensure your browser is up to date for the best
  experience.
262
263 ### Security
264 - **Trusted Sources**: Only use trusted file URLs, as data is processed in the browser. The extension does not execute server-side code.
265 - **HTTPS Enforcement**: All dependency URLs must use `https://` to ensure secure connections.
266 - **CORS**: Files must be hosted on servers that allow CORS access from Qlik Cloud.
267
268 ### Performance
269 - The extension uses DuckDB's in-browser processing, which is memory-intensive for large files. Adjust the "Max RAM" and "Rows per Page"
  settings to optimize performance.
270
271 ### Browser Performance Hints
272 The DuckDB Parquet Extension relies on WebAssembly, in-memory processing, and JavaScript execution, which can be resource-intensive. Below
  are browser-specific tips to maximize performance in Microsoft Edge, Google Chrome, and Opera.
273
274 ##### Microsoft Edge
275 - **Enable Hardware Acceleration**:
276   - Go to `Settings` > `System and performance`.
277   - Ensure "Use hardware acceleration when available" is turned on. This leverages your GPU for faster WebAssembly and rendering
    performance.
278   - Restart Edge after enabling.
279 - **Optimize Memory Usage**:
280   - Open `edge://flags` in the address bar.
281   - Search for "Memory Saver" and enable it. This frees up memory from inactive tabs, ensuring more resources are available for the
    extension.
282   - Restart Edge to apply the change.
283 - **Disable Unnecessary Extensions**:
284   - Go to `Settings` > `Extensions`.
285   - Disable or remove extensions that consume significant CPU or memory (e.g., ad blockers, video downloaders), as they can interfere with
    DuckDB's processing.
286 - **Enable Performance Mode**:
287   - Go to `Settings` > `System and performance`.
288   - Turn on "Efficiency mode" to reduce CPU usage, but if performance is still slow, switch to "Performance" mode (available in Edge
    versions 112+).
289 - **Clear Cache Regularly**:
290   - Go to `Settings` > `Privacy, search, and services` > `Choose what to clear` under "Clear browsing data."
291   - Select "Cached images and files" and clear. A large cache can slow down JavaScript execution.
292
293 ##### Google Chrome
```

```
294 - **Enable Hardware Acceleration**:
295   - Go to `Settings` > `System`.
296   - Ensure "Use hardware acceleration when available" is enabled. This improves WebAssembly performance and table rendering.
297   - Restart Chrome after enabling.
298 - **Use Memory Saver**:
299   - Go to `Settings` > `Performance`.
300   - Enable "Memory Saver" to free up memory from inactive tabs, ensuring DuckDB has enough resources for in-memory processing.
301   - Set to "High" for maximum memory efficiency.
302 - **Optimize JavaScript Execution**:
303   - Open `chrome://flags` in the address bar.
304   - Search for "JavaScript" and enable "JavaScript compiler optimizations" (if available). This can speed up the extension's
    JavaScript-heavy operations.
305   - Restart Chrome to apply the change.
306 - **Disable Resource-Heavy Extensions**:
307   - Go to `Settings` > `Extensions`.
308   - Disable extensions that use significant CPU or memory (e.g., script blockers, VPNs), as they can slow down WebAssembly and JavaScript
    execution.
309 - **Enable Preload Pages**:
310   - Go to `Settings` > `Privacy and security` > `Cookies and other site data`.
311   - Turn on "Preload pages for faster browsing and searching." This can speed up file fetching and dependency loading.
312 - **Clear Cache and Reduce Memory Footprint**:
313   - Go to `Settings` > `Privacy and security` > `Clear browsing data`.
314   - Select "Cached images and files" and clear. A bloated cache can impact performance.
315
316 ##### Opera
317 - **Enable Hardware Acceleration**:
318   - Go to `Settings` (or `opera://settings`).
319   - Scroll to `System` and ensure "Use hardware acceleration when available" is enabled. This optimizes WebAssembly and rendering
    performance.
```