MarkLogic Server

Query Console User Guide

MarkLogic 9
May, 2017

Last Revised: 9.0-5, May 2018

Table of Contents

Query Console User Guide

1.0 Introduction to Query Console

Query Console is an interactive web-based query development tool for writing and executing ad-hoc queries in XQuery, Server-Side JavaScript, SQL and SPARQL. Query Console enables you to quickly test code snippets, debug problems, profile queries, and run administrative XQuery scripts.

The following terms and definitions cover the primary Query Console components:

Term	Definition
query	Any executable block of XQuery, Server-Side JavaScript, SQL, or SPARQL. When you run a query in Query Console, you can view the results in your choice of formats.
workspace	A collection of queries. Use workspaces to organize your queries. You can create multiple workspaces, but only one is active at a time.
history	A record of previously executed versions of a query. Each time you execute a query in Query Console, the query text is saved in the history. Use the history to restore a query to a previous state.

Using Query Console, you can:

- Create queries in JavaScript, XQuery, SQL, and SPARQL.
- Create, modify, rename, and delete queries.
- Run a query and view the output in multiple formats.
- Profile query performance (XQuery and JavaScript only).
- Explore the contents of a database.
- Create, delete, copy, rename, and reorganize workspaces to improve query organization.
- Restore a query to a previous version from the saved history.
- Export and import workspaces for easy sharing among users or across MarkLogic Server instances.

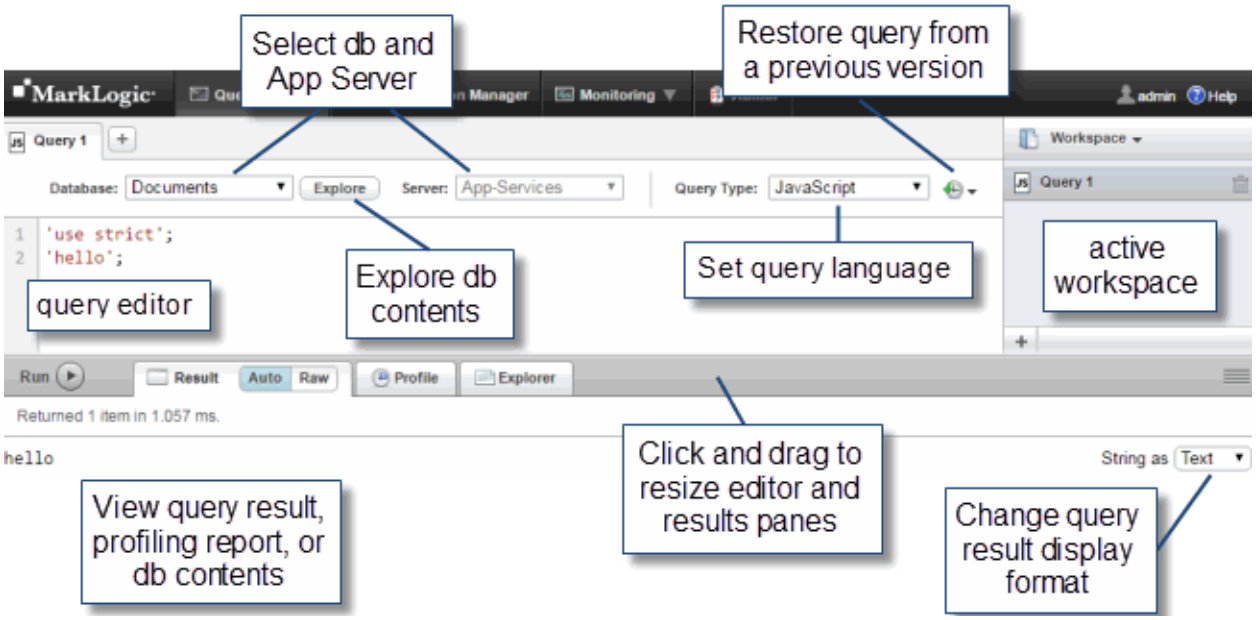
The query editor in Query Console includes features such as

- Syntax coloring
- Typeahead suggestions
- Pop-up function reference documentation
- Automatic closing and highlighting of quotes, parentheses, braces, and other grouping characters

The workspaces and queries created in Query Console are stored in MarkLogic Server, so they are available to you from any computer with access to your MarkLogic Server instance. For example, you can create workspaces and queries on your desktop computer and use them from a lab machine with access to the same MarkLogic Server instance.

> **Note:** You should only have one Query Console session active at a time for any given MarkLogic user. Query Console saves state to MarkLogic Server. If a user has multiple Query Console sessions active concurrently, the state can become inconsistent.

The picture below summarizes key Query Console UI features. For more information on using specific features, see the “Query Console Walkthrough” on page 5.



2.0 Query Console Walkthrough

This chapter provides a quick introduction to using the core Query Console features.

2.1 Accessing Query Console

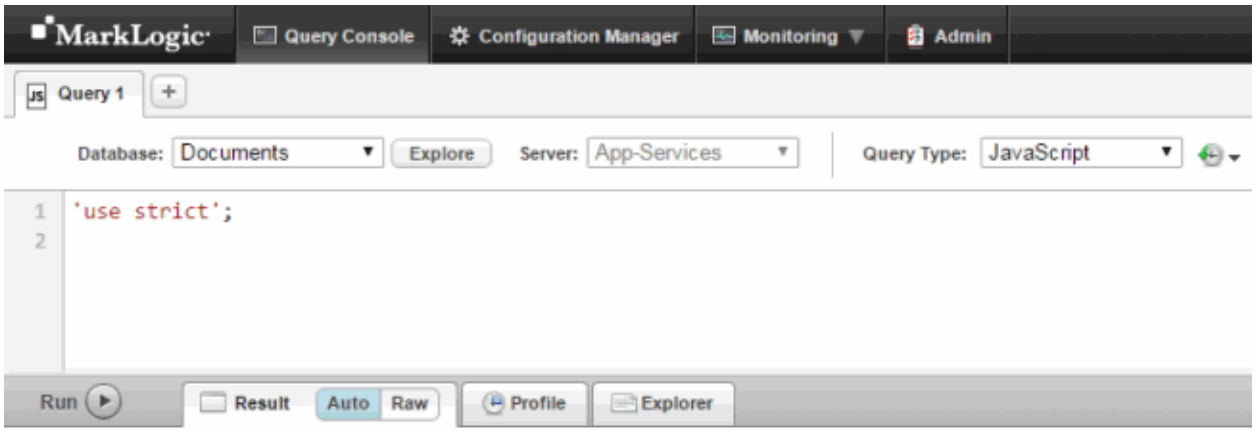
To begin using Query Console, open a browser and enter the URL:

```
http://your_host:8000
```

Note: If the application does not appear, you may not have sufficient privileges. To use Query Console, you must be a member of the `qconsole-user` role. If your privileges are insufficient, contact your MarkLogic Server administrator.

Note: Query Console does not grant extra access to databases or documents. To perform operations such as document insertion or deletion or database exploration from Query Console, you must have appropriate security privileges.

The first time you launch Query Console, you should see a page similar to the following:



You should only have one Query Console session active at a time for any given MarkLogic user. Query Console saves state to MarkLogic Server. If a user has multiple Query Console sessions active concurrently, the state can become inconsistent. For example, do not log into Query Console as the same user in multiple browsers or browser tabs.

2.2 Creating a Query

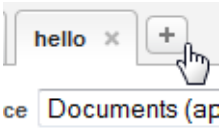
This section walks you through creating a new query.

The following example assumes an empty workspace named `Workspace`, populated only with the default initial XQuery query, named `Query 1`. This is the configuration you see the first time you launch Query Console.

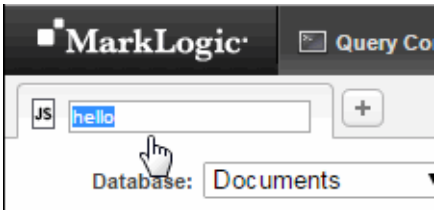
To create and run a query:

1. If the current workspace is not Workspace, click on the workspace dropdown on the upper right and select the workspace named Workspace.

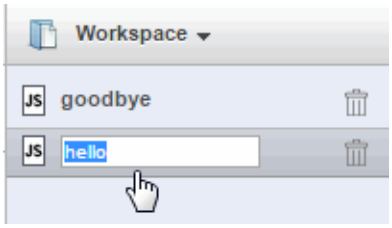
2. Click on the "+" at the top of the query editor to the right of the open query tabs. A new query is created and opened in the editor. The new query also appears in the workspace panel on the right.



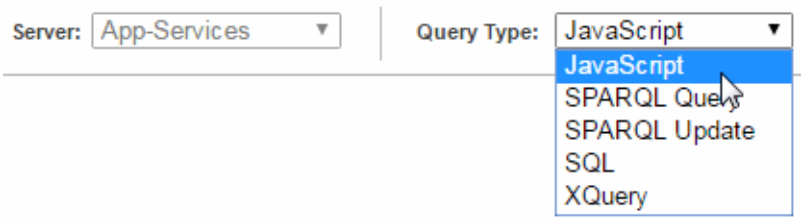
3. Double-click on the query name in the tab at the top of the editor and type in a meaningful name for the query, such as `hello`. Notice the name changes in the tab and in the workspace panel on the right.



You can also rename a query by double-clicking its name in the workspace panel.



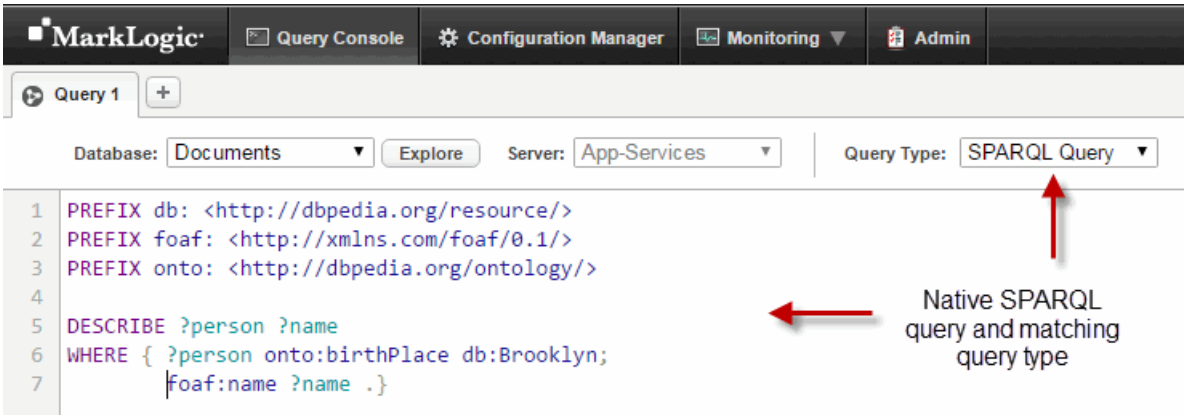
4. Choose the query type by clicking on the Query Type dropdown to the right of the Explore button. For this example, select XQuery.



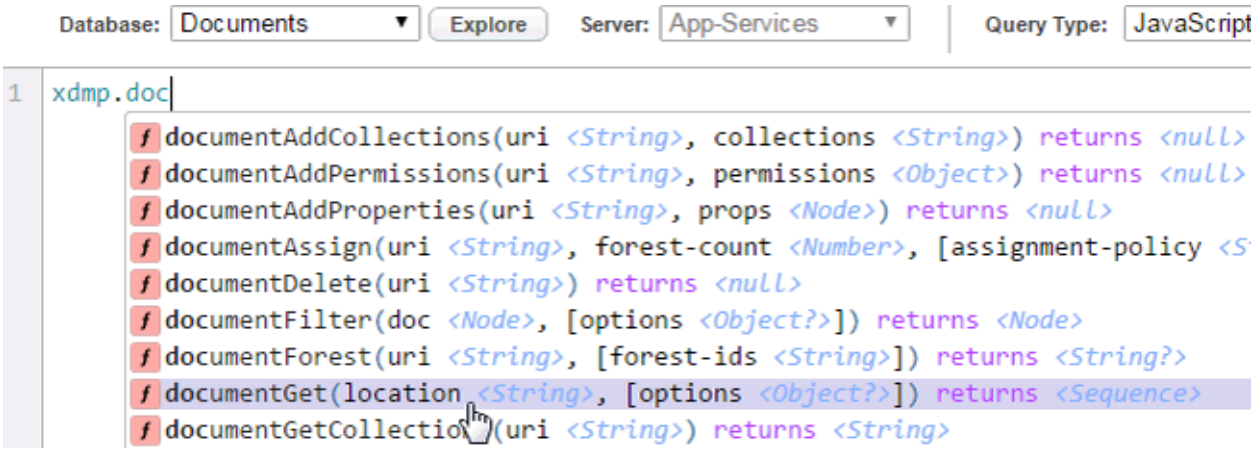
The Query Type determines the evaluation context for your query when you run it. The supported query langages are:

- Server-Side JavaScript
- XQuery
- SQL
- SPARQL Query
- SPARQL Update

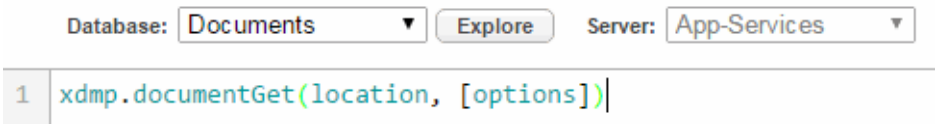
The following example shows a native SPARQL query with the Query Type set appropriately.



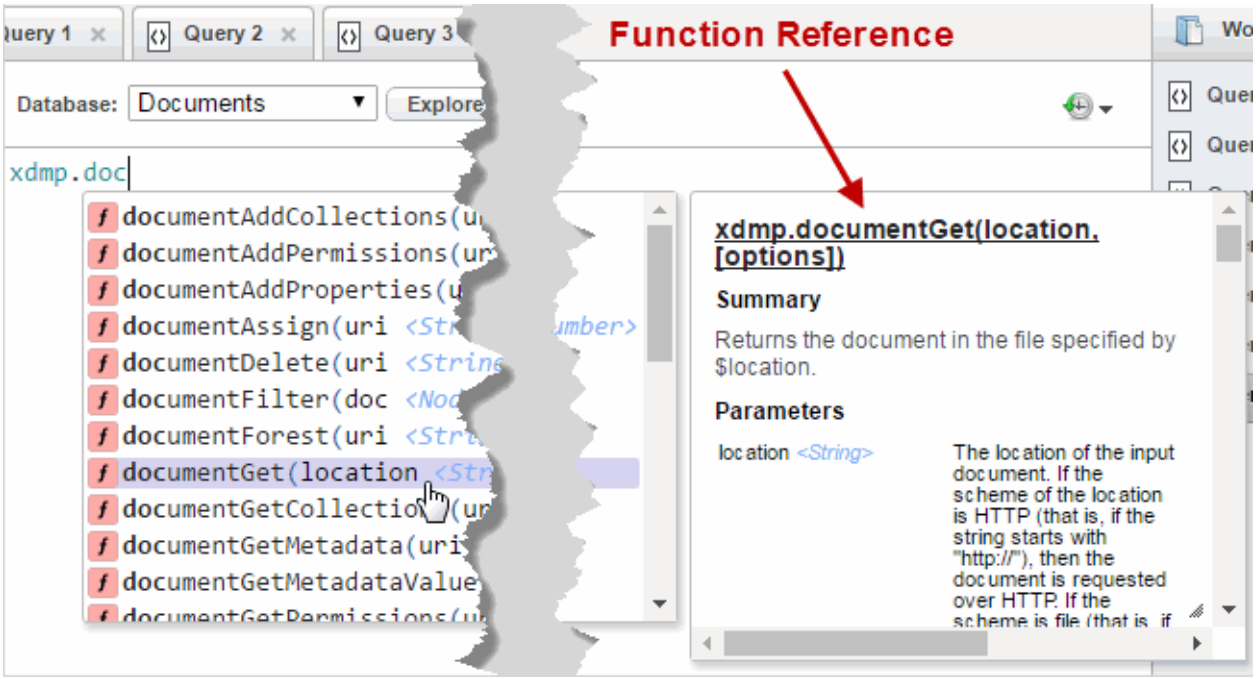
In XQuery and JavaScript, Query Console displays suggestions for matching functions, variables, and keywords as you type. For example:



If you click on a suggestion, Query Console adds the suggestion text to your query. For example, if you select `documentGet` in the list above, then the following text is entered into your query. You can replace the suggestion parameters with values appropriate to your task.



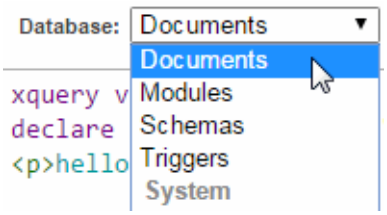
For a function, Query Console also display reference documentation to the right of the suggestions. For example:



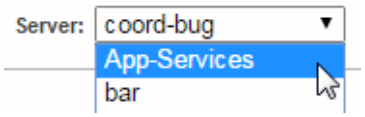
2.3 Running a Query

Follow this procedure to evaluate a query and view the results. You should already have entered your query in the query editor and selected the appropriate Query Type. If not, refer to Creating a Query.

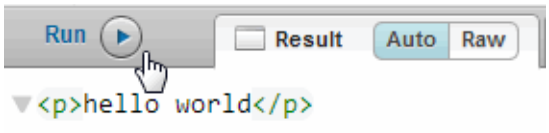
1. Click on the Database dropdown at the top of the editor to select the content database against which to run the query. For this example, you can use any database.



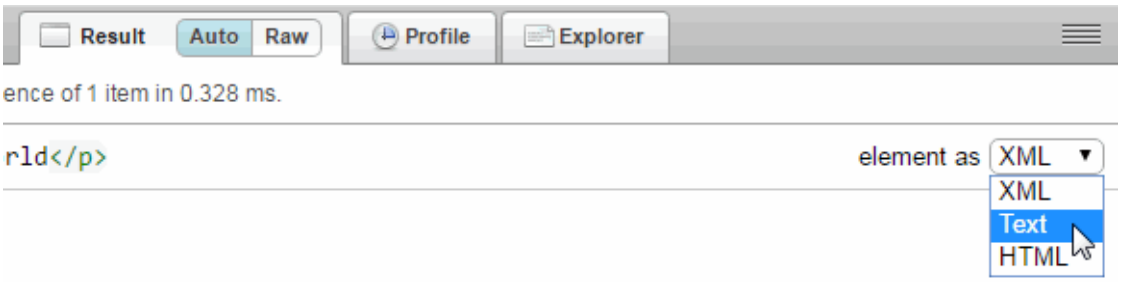
2. Click on the Server dropdown at the top of the editor to select the App Server against which to run the query. For this example, you can use any App Server.



3. Click the Run button to evaluate the query. In this case, the default "hello world" query. The prettyprinted results display in the output pane at the bottom of the page.



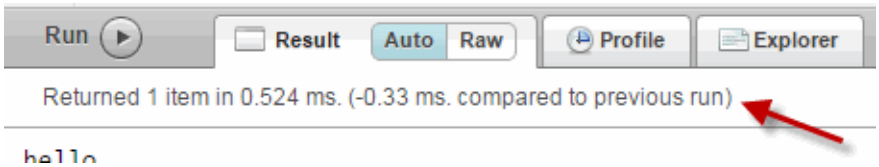
4. To view the query results as plain text, click the result format dropdown on the far right of the results pane and select Text. Your query results display as plain text.



5. To view the query results without prettyprint formatting, click the Raw button on the Result tab. Your raw query results display in the results pane at the bottom of the page. For details, see “Changing the Query Output Format” on page 12.

Each query has its own Results display tab, so you can switch between queries without losing the results of a run.

Each time you run a particular query, the results tab reports how long the query took and the time difference between the current and previous run. For example:



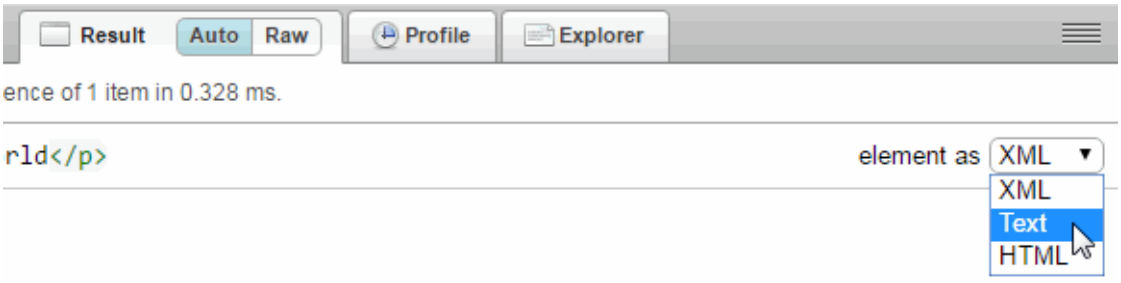
If your query returns a sequence of values, Query console reports the number of items in the sequence so you can distinguish between returning a sequence of one item versus a single item (not in a sequence).

2.4 Changing the Query Output Format

Query Console supports two modes for displaying query results, Auto and Raw. The default mode is Auto. In Auto mode, your query results are formatted for readability based on the query and the output type. For example:

- XML and JSON query results are displayed with syntax coloring and UI elements that allow you to expand and collapse the element tree. Sequences are unrolled to line items with individual formatting controls.
- Results from a SQL query (run in SQL mode) are formatted as a table.
- Results from a SPARQL query display matching IRIs.

In Auto mode, you can override the default rendering using the format dropdown at the far right of the results pane.



For example, strings are rendered as text by default, but if you know the string contains serialized JSON, you can change the rendering to JSON to get syntax highlighting and tree controls. The choices on the format dropdown depend on the type of data returned by your query.

Raw mode always displays plain text, but it is not necessarily the query results exactly as returned from MarkLogic Server. Slight formatting changes are still applied to improve readability. For example, even in Raw mode, an XQuery sequence or JavaScript Sequence displays as line items.

2.5 Using the Query Execution History

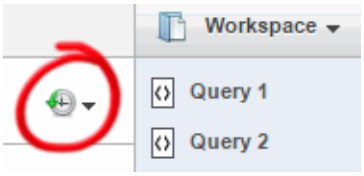
Each time you modify a query and evaluate it, Query Console saves the contents and time of execution in the Query History. Query Console maintains a separate history for each query.

Query Console adds a history entry for each unique version of a query. If the query text is unchanged between runs or if the changes create a duplicate of an existing history entry for the query, Query Console does not create a new entry.

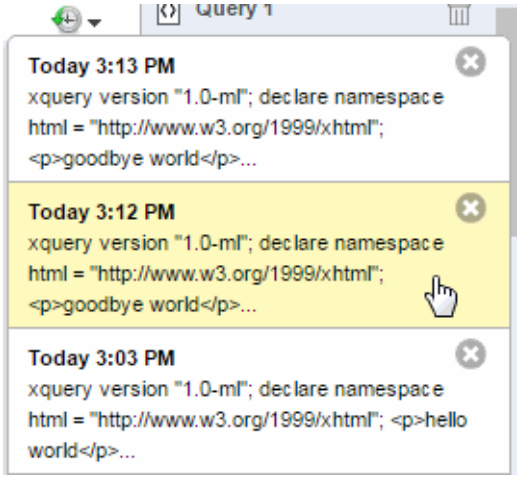
Query Console saves at most 50 history entries.

To use the query history:

1. Click the Query History dropdown on the upper right. The history appears, with the most recent runs at the top of the list.



2. To revert the query to a previous state, click on a history entry. The selected query version is restored in the editor.



To remove a history entry, click the delete (X) button in the upper right corner of the entry.

To close the history dropdown, click on the Query History dropdown again, or simply move the mouse outside the dropdown.

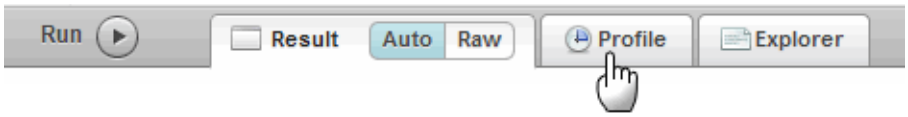
2.6 Profiling a Query

You can use Query Console to profile the performance of an XQuery or Server-Side JavaScript query. For XQuery, Query Console profiles your query as if you passed your query to `prof:invoke`, and then displays a performance report in the results pane.

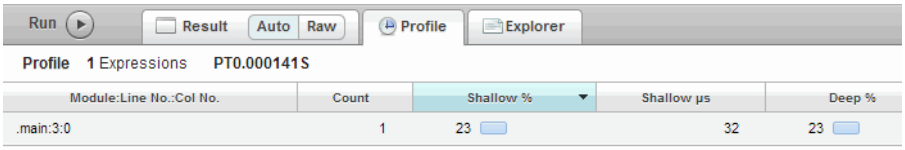
> **Note:** Profiling must be enabled on an App Server before you can profile a query. It is enabled by default when you create an App Server. For details, see Enabling Profiling on an App Server in the *Query Performance and Tuning Guide*.

Use the following procedure to profile a query:

1. Click the Profile tab at the top of the result panel. The profile tab is brought to the front.



2. Click the Run button to evaluate your query. A profiling report appears. If no profiling report appears, profiling may not be enabled for your App Server.

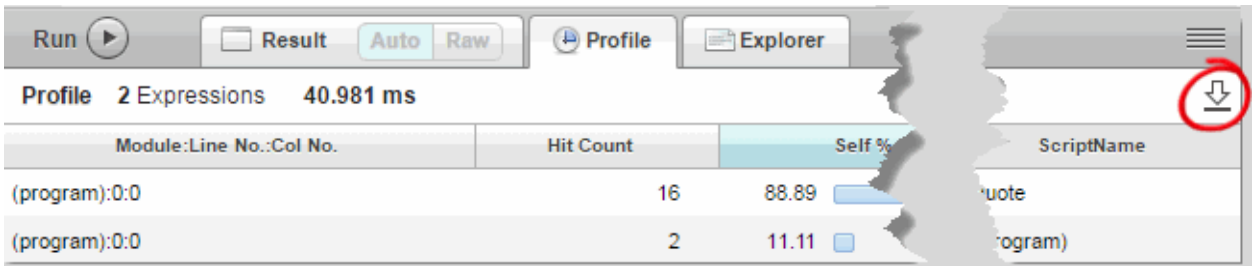


   In XQuery, your Query Console query appears as the `.main` module in the profiling report. In JavaScript, your query appears as `(program)`.

3. Click on a profiling report column header to sort the profiling data by a particular column. Each time you click a column, the order toggles between ascending and descending.

4. Click on the Result tab to view the output from your query.

When profiling a JavaScript query, you can click on the download icon in the upper right of the profiling report to save your profiling data in format that can be imported into the `Profiles` tab of the Chrome browser developer tools.



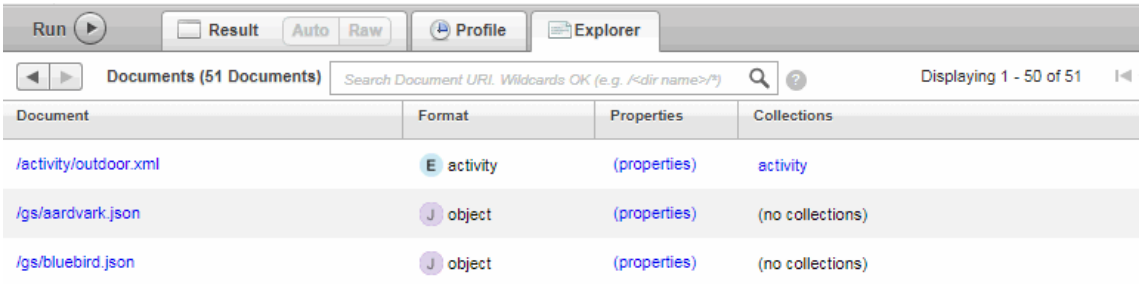
For details on profiling queries and the meaning of the profile report columns, see Profiling Requests to Evaluate Performance in *Query Performance and Tuning Guide*.

2.7 Exploring a Database

Use the Explore feature to browse the contents of a database. To explore a database:

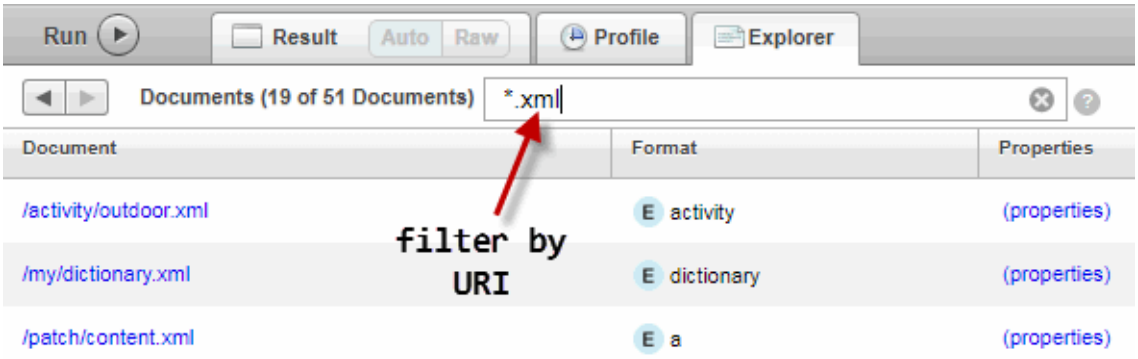
1. Select a database from the Database dropdown at the top of the current query.

2. Click Explore, to the right of the Database dropdown. Query Console displays a list of the documents in the selected database in the Explorer. For example:

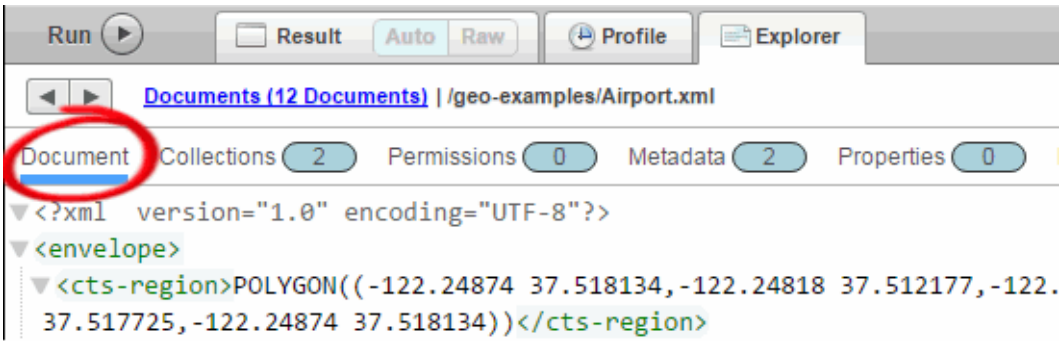


For each document in the database, the summary includes the document URI, the type and name of the root node, a link to the document properties, and a link to any collections to which the document belongs.

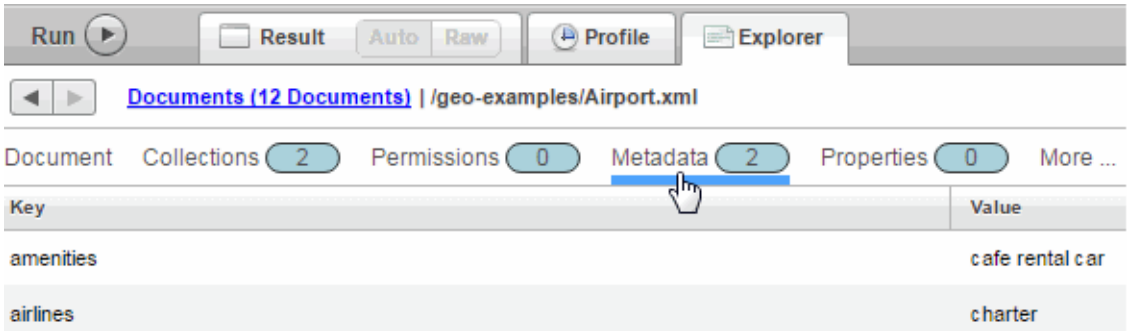
3. Optionally, enter a wildcard expression in the search box to filter the list of documents by URI. For more details, see "Filtering the Explorer View by URI" on page 16.



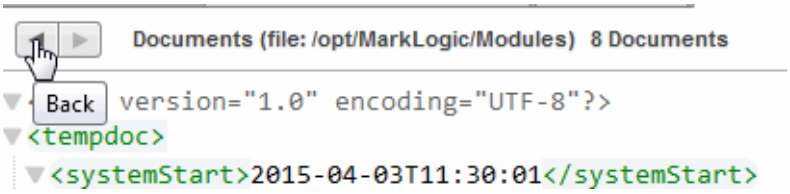
4. Click on a document URI to explore the document contents and metadata. The content is displayed by default.



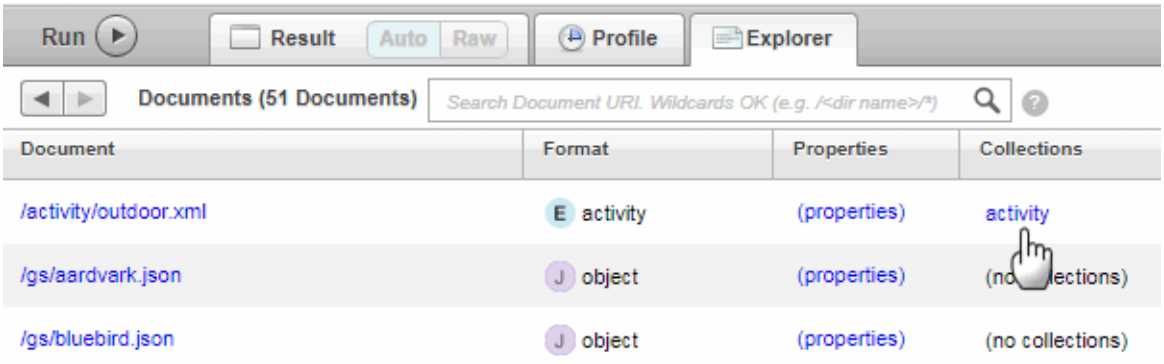
5. Click on one of the metadata categories above the content to explore details about that category. For example, click on Metadata to explore the document's key-value metadata:



6. Use the forward and back buttons in the upper left of the Explorer to navigate as you drill down into document contents and metadata. For example, to return to the database content summary after clicking on a document's URI, click the back arrow.



7. To explore the collections or properties of a document from the document list, click the link in the Collections or Properties column.



2.8 Filtering the Explorer View by URI

Use the search box on the Explorer tab to explore only a specific document or the documents that match a wildcard expression. The match uses the same semantics as the XQuery function `cts:uri-match` or the Server-Side JavaScript function `cts.uriMatch`.

Note: The URI lexicon must be enabled on the database you are exploring. You will get an error if the URI lexicon is not enabled.

The following semantics apply to filtering by URI:

- Your regular expression can include the following wildcard characters. These characters can occur multiple times and in any position.
  - "*" : Matches zero or more non-space characters.
  - "?": Matches exactly one non-space character.
- If all the non-wildcard characters in your expression are lower case, then the matching is case insensitive. Otherwise, the matching is case sensitive.
- If your regular expression contains no diacritics, then the matching is diacritic insensitive. Otherwise, the matching is diacritic sensitive.
- The URI filter is combined with any collection filter. For example if you enter "*.xml" in the search box and then click on the name of a collection in the Collections column of the Explorer, you see documents in that collection that end with ".xml".

The following table contains some examples of filtering expressions:

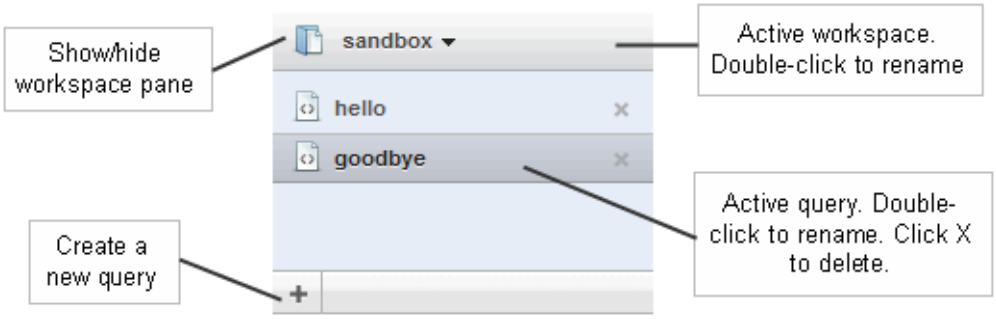
Filter Pattern	Explanation
/my/dir/doc	Matches only the document with exact URI "/my/dir/doc".
/my/dir/doc*	Matches URIs beginning with "/my/dir/doc", such as "/my/dir/doc", "/my/dir/doc.xml", "/my/dir/documents/example.json".
/my/dir/doc?.xml	Matches URIs such as "/my/dir/doc1.xml" or "/my/dir/docA.xml". Does not match URIs such as "/my/dir/doc.xml" or "/my/dir/doc10.xml".
/my/dir/doc??.json	Matches URIs such as "/my/dir/doc10.json" or "/my/dir/docAB.json". Does not match documents such as "/my/dir/doc1.xml" or "/my/dir/doc100.xml".
/my/dir/doc?*.xml	Matches documents such as "/my/dir/doc1.xml", "/my/dir/doc10.xml", "/my/dir/doc100.xml". Does not match URIs such as "/my/dir/doc.xml".
/my/dir?/doc?.*	Matches documents such as "/my/dir1/docA.xml" or "/my/dirA/doc1.json".

2.9 Organizing Queries with Workspaces

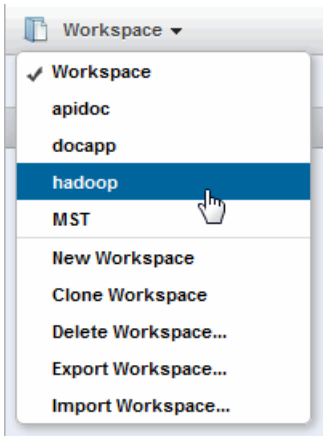
In Query Console, you organize your queries in workspaces. You can create multiple workspaces. However, only one workspace is active at a time. When you create a new query, Query Console automatically saves it in the active workspace.

2.9.1 Workspace Overview

Use the workspace panel on the upper right of the page to interact with or change the active workspace. The workspace panel shows the name of the active workspace and lists the queries it contains:

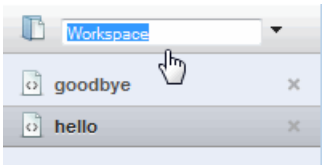


To see a list of available workspaces or to create, clone, delete, import or export a workspace, use the dropdown menu to the right of the workspace name:



2.9.2 Renaming a Workspace

To rename a workspace, double-click on the workspace name at the top of the workspace panel:



2.9.3 Reorganizing a Workspace

You can re-order the list of queries in a workspace by dragging and dropping the query items in the workspace panel. You can also reorder queries in the editor panel by dragging and dropping the query tabs.

2.9.4 Copying a Workspace

To create a new workspace that contains the same queries as an existing workspace:

1. If the source workspace is not the active workspace, make it the active workspace by selecting it in the workspace menu.

2. Click Clone Workspace in the workspace menu. A new workspace named "Clone of *workspace_name*" is created and becomes the active workspace.

3. To rename the new workspace, double-click on the name at the top of the workspace panel.

When you clone a workspace, all the queries in the original workspace are copied to the new workspace. Query histories are not copied.

2.9.5 Deleting a Workspace

To delete a workspace and all of the queries it contains:

1. If the workspace to delete is not the active workspace, make it the active workspace by selecting it in the workspace menu.

2. Click Delete Workspace in the workspace menu.

3. Click OK in the confirmation dialog box to confirm deletion of the workspace.

If you delete the last workspace, Query Console automatically creates a workspace with the default initial contents.

2.9.6 Exporting a Workspace

Export a workspace to share it with another user or use it on a different MarkLogic Server instance. Exporting a workspace saves the workspace and queries to an external file which can be imported back into Query Console. Query history is not exported.

To export a workspace:

1. If the workspace to export is not the active workspace, make it the active workspace by selecting it in the workspace menu.

2. Click Export Workspace in the workspace menu. The workspace is saved as an external XML file, using your browser download capability.

By default, the exported file is named *`workspace_name`*`.xml`.

2.9.7 Importing a Workspace

To import a previously exported workspace into Query Console:

1. Click Import Workspace in the workspace menu. The "Import a Workspace" dialog box appears.

2. In the dialog box, click the Choose File button to select an exported workspace XML file.

   To cancel the import, click anywhere outside the dialog box.

3. Click `Import` to load the workspace. A loading progress window displays.

4. When loading completes, the imported workspace becomes the active workspace.

If a workspace already exists with same name as the imported workspace, the imported workspace name is modified by appending a unique number to the name.

3.0 Keyboard Shortcuts

Query Console provides keyboard shortcuts for controlling the UI minimal mouse interaction.

Where there are combination key sequences in the table below, press and hold the modifier key or keys while pressing the final key. For example, `Ctrl R` means press and hold the `Ctrl` key and then press the (lower case) `R` key.

Operation	Windows/Linux	Macintosh
Run query and show results in the currently selected output format	Ctrl Enter	Ctrl Enter
Run query and show results in Auto mode	Ctrl Shift O	Ctrl Shift O
Run query and show results in Raw mode	Ctrl Shift R	Ctrl Shift R
Profile query	Ctrl Alt Shift Enter	Ctrl Option Shift Enter
Create a new query	Alt =	Ctrl =
Close the current query	Alt -	Ctrl -
Change the relative height of the query editor and results pane. (Cycles through positions).	Double Click, or Ctrl Shift Space	Double Click, or Ctrl Shift Space
Maximize editor and hide workspace pane	Ctrl Alt ’	Ctrl Option ’
Expand/collapse query execution history	Ctrl Alt H	Ctrl Option H
Create a new workspace	Ctrl Alt W	Ctrl Option W
Clone the current workspace	Ctrl Alt Shift W	Ctrl Option Shift W
Select query above the current selection, when the workspace panel has the focus	Up Arrow	Up Arrow
Select query below the current selection, when the workspace panel has the focus	Down Arrow	Down Arrow

4.0 Administering Query Console

This chapter covers tasks specific to administering Query Console on your MarkLogic Server.

4.1 Controlling Access to Query Console

Query Console stores per user information about workspaces and queries in MarkLogic Server. Query Console uses the following pre-defined security roles:

- `qconsole-user`
- `qconsole-internal`

Users also require normal privileges to any databases or documents they access through Query Console.

For details about the MarkLogic Server security model and about configuring users and roles, see the *Security Guide* and Security Administration in the *Administrator's Guide*.

4.1.1 qconsole-user

The `qconsole-user` role is a minimally privileged role that is needed to use Query Console. You must grant this role to all users who are allowed to use Query Console.

The `qconsole-user` role has the following execute privileges:

- qconsole (http://marklogic.com/xdmp/privileges/qconsole)

4.1.2 qconsole-internal

The `qconsole-internal` role is used by Query Console to amp certain functions that Query Console performs. You should not explicitly grant the `qconsole-internal` role to any user; it is only for internal use by Query Console.

4.2 Removing a User's Data From the Server

When Query Console users create workspaces and queries, the data is saved on the server in the App-Services database. If you need to remove all of a user's Query Console state information from the server, use a script similar to the following script.

For further assistance, contact MarkLogic Technical Support.

```
xquery version "1.0-ml";

declare namespace qc="http://marklogic.com/appservices/qconsole";

(: find the user id associated with a user name :)
declare function local:get-user-id($user-name as xs:string)
{
  let $eval :=
    fn:concat(
```

```
      'xquery version "1.0-ml";
       import module namespace
           sec="http://marklogic.com/xdmp/security"
                  at "/MarkLogic/security.xqy";
       sec:uid-for-name("', $user-name, '")')
  let $options :=
    <options xmlns="xdmp:eval">
      <database>{xdmp:database("Security")}</database>
    </options>
  return
  xdmp:eval($eval, (), $options)
};

(: retrieve all workspace URI's for a named user :)
declare function local:get-workspace-uris(
  $user-name as xs:string)
{
  let $user-id := local:get-user-id($user-name)
  return
    if (fn:empty($user-id))
    then ()
    else
    for $d in fn:doc()/qc:workspace/qc:security[qc:userid eq $user-id]
      return base-uri($d)
};

(: retrieve id's for all queries in a given workspace :)
declare function local:get-query-ids(
  $workspace-uri as xs:string)
{
  for $qid in
    fn:doc($workspace-uri)/qc:workspace/qc:queries/qc:query/*:id
  return $qid
};

(: Retrieve id's of all history entries associated with a query id :)
declare function local:get-query-history(
  $qid as xs:unsignedLong)
{
  for $d in fn:doc()
  where $d/qc:history/qc:query[qc:id eq $qid]
  return base-uri($d)
};

let $user-name := xdmp:get-request-field("username")
let $user-documents :=
  for $ws in local:get-workspace-uris($user-name)
  return (
    for $qid in local:get-query-ids($ws)
    return
      (
        fn:concat("/queries/", $qid, ".txt"),
        local:get-query-history($qid)
      ),
```

```
      $ws
    )
for $d in $user-documents
return xdmp:document-delete($d)
```

5.0 Technical Support

MarkLogic provides technical support according to the terms detailed in your Software License Agreement or End User License Agreement.

We invite you to visit our support website at http://help.marklogic.com to access information on known and fixed issues, knowledge base articles, and more. For licensed customers with an active maintenance contract, see the Support Handbook for instructions on registering support contacts and on working with the MarkLogic Technical Support team.

Complete product documentation, the latest product release downloads, and other useful information is available for all developers at http://developer.marklogic.com. For technical questions, we encourage you to ask your question on Stack Overflow.

6.0 Copyright

MarkLogic Server 9.0 and supporting products.
Last updated: April 28, 2018

COPYRIGHT

This technology is protected by U.S. Patent No. 7,127,469B2, U.S. Patent No. 7,171,404B2, U.S. Patent No. 7,756,858 B2, and U.S. Patent No 7,962,474 B2, US 8,892,599, and US 8,935,267.

The MarkLogic software is protected by United States and international copyright laws, and incorporates certain third party libraries and components which are subject to the attributions, terms, conditions and disclaimers set forth below.

For all copyright notices, including third-party copyright notices, see the Combined Product Notices for your version of MarkLogic.