The query-astro package

v0.1.0

Valentin Dao*

April 20, 2026

The *query-astro* package provides a minimalist interface to mimic Python's Astroquery library, using cURL to send ADQL queries to the TAP services of major astronomical databases. It therefore requires **shell escape** to be enabled. Note that all queries are performed *synchronously*: a resource-intensive query (e.g. `SELECT TOP 100 *` against the Gaia catalogue) may significantly slow down compilation.[1] Simple queries are thus strongly recommended. For now, *query-astro* supports two databases (Simbad and Gaia) which should cover most use cases; future versions will allow users to register custom TAP service URLs. As of now, only raw ADQL queries are supported. No higher-level interface is provided, though this is planned for future versions. The package is still in active development, so some rough edges are to be expected. Finally, users are reminded that properly crediting the queried databases remains their own responsibility; this package is merely a convenience wrapper for LaTeX.

1 Functioning of the package

The package essentially works in three steps:

1. The ADQL query is written in a auxiliary file named `query.ADQL`.
2. A cURL configuration file named `curl.cfg` is generated.
3. The query is executed by calling cURL with the generated configuration file. The result is then stored in a auxiliary file named `query-result.txt`.

Every time you run a query, these files get overwritten.

*E-mail: vdao.texdev@gmail.com

[1] Asynchronous queries are not currently supported, as I am unaware of a straightforward way to implement async TAP services in TeX...if it is even possible at all.

2 Package macro

\ADQLquery `\ADQLquery {⟨macro⟩} {⟨ADQL query⟩} [⟨database⟩]`

This macro executes the ADQL request and stores the result in the provided `{⟨macro⟩}`. The nature of this macro depends on the number of queried keys with `SELECT`. If only one key was submitted, then the macro will be defined without arguments. If a list of keys was provided, then the individual results may be retrieved by specifying their index as an argument. In both cases, the macro will be expandable. Although `\ADQLquery` can support storing multiple columns, it *does not* support storing multiple rows. Hence, any query that goes along the line of `SELECT TOP...` will, in theory, work, but only the first row from the text file will be inspected. If you don't mind this limitation, you can safely ignore this warning. The optional argument `[⟨database⟩]` specifies the database to query. For now, here is the list of supported databases:

`simbad`	The Simbad astronomical database. [1]
`gaia`	The Gaia astronomical database. [2] [3]

If it isn't specified, Simbad will always be used. Here are a few examples of use:

```
\ADQLquery\SimbadVMag{
  SELECT V
  FROM allfluxes
  JOIN ident ON allfluxes.oidref = ident.oidref
  WHERE ident.id = 'Betelgeuse'
}
The V-band magnitude of Betelgeuse is \SimbadVMag.
```

The V-band magnitude of Betelgeuse is 0.42.

```
\ADQLquery\SimbadCoords{
  SELECT ROUND(ra, 5), ROUND(dec, 5)
  FROM basic
  WHERE main_id = 'M31'
}
Right ascension: \SimbadCoords{1}\par
Declination: \SimbadCoords{2}
```

Right ascension: 10.68471
Declination: 41.26875

```
\DeclareSIUnit\milliarcsecond{mas}
\ADQLquery\SiriusParallax{
  SELECT ROUND(parallax, 5), ROUND(parallax_error, 5)
  FROM gaiadr3.gaia_source_lite
  WHERE source_id = 2947050466531873024
}[gaia]
The parallax of Sirius is $\SiriusParallax{1} \pm \SiriusParallax{2} \unit{
    \milliarcsecond}$.
```

The parallax of Sirius is 374.48959 ± 0.23134mas.

Changelog

0.1.0 (20-04-2026) — Initial version.

References

[1] M Wenger et al. "The SIMBAD astronomical database." en. In: *Astron. Astrophys. Suppl. Ser.* 143.1 (Apr. 2000), pp. 9–22.

[2] Gaia Collaboration et al. "The Gaia mission." In: *A&A* 595 (2016), A1. DOI: 10.1051/0004-6361/201629272.

[3] Gaia Collaboration et al. "Gaia DR3: Summary of the contents and survey properties." In: *A&A* 674 (2023), A1. DOI: 10.1051/0004-6361/202243940.

3 Implementation

```
⟨*package⟩
⟨@@=query_astro⟩
\NeedsTeXFormat{LaTeX2e}

\def\queryastro@module{query-astro}
\def\queryastro@version{v0.1.0}
\def\queryastro@date{2026-04-20}
\def\queryastro@description{Querying astronomical databases with ADQL in LaTeX}

\ProvidesExplPackage
  \queryastro@module
  \queryastro@date
  \queryastro@version
  \queryastro@description

\sys_if_shell_unrestricted:F { \PackageError{query-astro}{query-astro~requires~shell~escape

\cs_generate_variant:Nn \seq_gset_split:NnV { cnV }

\iow_new:N \g__query_astro_ADQL_query_iow
\iow_new:N \g__query_astro_curl_config_iow
\ior_new:N \g__query_astro_ADQL_result_ior

\tl_new:N \l__query_astro_TAP_url_tl
\tl_const:Nn \c__query_astro_simbad_TAP_url_tl { https://simbad.u-strasbg.fr/simbad/sim-
tap/sync }
\tl_const:Nn \c__query_astro_gaia_TAP_url_tl { https://gea.esac.esa.int/tap-server/tap/sync

\cs_new_protected:Npn \__query_astro_write_curl_configuration:n #1
  {
    \iow_open:Nn \g__query_astro_curl_config_iow { curl.cfg }
    \iow_now:Ne \g__query_astro_curl_config_iow { #1 }
    \iow_close:N \g__query_astro_curl_config_iow
  }

\cs_new_protected:Npn \__query_astro_write_ADQL_query:n #1
  {
    \iow_open:Nn \g__query_astro_ADQL_query_iow { query.ADQL }
    \iow_now:Ne \g__query_astro_ADQL_query_iow { #1 }
    \iow_close:N \g__query_astro_ADQL_query_iow
  }

\cs_new_protected:Npn \__query_astro_execute_ADQL_query:n #1
  {
    \__query_astro_write_ADQL_query:n { #1 }
    \__query_astro_write_curl_configuration:n
      {
```

```
        data-urlencode~=~"QUERY@query.ADQL"^^J
        data~=~"REQUEST=doQuery"^^J
        data~=~"LANG=ADQL"^^J
        data~=~"FORMAT=csv"^^J
        url~=~"\l__query_astro_TAP_url_tl"^^J
        output~=~"query-result.txt"^^J
      }
    \sys_shell_now:n {curl~--config~curl.cfg}
  }

\cs_new_protected:Npn \__query_astro_retrieve_ADQL_result:n #1
  {
    \ior_open:Nn \g__query_astro_ADQL_result_ior { query-result.txt }
    \ior_get:NN \g__query_astro_ADQL_result_ior \l_tmpa_tl % ignore header
    \ior_get:NN \g__query_astro_ADQL_result_ior \l_tmpb_tl
    \seq_new:c { g__query_astro_ADQL_#1_result_seq }
    \seq_gset_split:cnV { g__query_astro_ADQL_#1_result_seq } { , } \l_tmpb_tl
    \ior_close:N \g__query_astro_ADQL_result_ior
    \int_compare:nNnTF { \seq_count:c { g__query_astro_ADQL_#1_result_seq } } > { 1 }
      { \cs_new:cpn { #1 } ##1 { \seq_item:cn { g__query_astro_ADQL_#1_result_seq } { ##1 } }
      { \cs_new:cpn { #1 } { \seq_item:cn { g__query_astro_ADQL_#1_result_seq } { 1 } } }
  }

\NewDocumentCommand\ADQLquery{ m m O{simbad} }
  {
    \str_case:nnF {#3}
      {
        {simbad} { \tl_set_eq:NN \l__query_astro_TAP_url_tl \c__query_astro_simbad_TAP_url_
        {gaia}   { \tl_set_eq:NN \l__query_astro_TAP_url_tl \c__query_astro_gaia_TAP_url_tl
      }
      { \PackageError{query-astro}{Unknown~database:~'#3'}{} }
    \__query_astro_execute_ADQL_query:n {#2}
    \exp_args:Ne \__query_astro_retrieve_ADQL_result:n { \cs_to_str:N #1 }
  }
```
⟨/package⟩