

# Package: hydrocan (via r-universe)

June 8, 2026

**Title** Unified Access to Canadian Hydrometric Data

**Version** 0.0.1

**Description** Provides a unified interface for accessing diverse web-published hydrometric data sources across Canada, presenting users with consistent tabular output regardless of the underlying data source.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**URL** <https://github.com/HakaiInstitute/hydrocan>,  
<https://hakaiinstitute.github.io/hydrocan/>

**BugReports** <https://github.com/HakaiInstitute/hydrocan/issues>

**Depends** R (>= 4.2.0)

**Imports** cli, dplyr, httr2, tibble

**Suggests** httpptest2, knitr, rmarkdown, testthat (>= 3.0.0), withr

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Config/pak/sysreqs** libssl-dev

**Repository** <https://hakaiinstitute.r-universe.dev>

**Date/Publication** 2026-06-08 20:13:15 UTC

**RemoteUrl** <https://github.com/HakaiInstitute/hydrocan>

**RemoteRef** HEAD

**RemoteSha** dcfa071bb3998dbe690b1391ac61a5f2f7e6a4cf

## Contents

hc_citation . . . . .	2
hc_list_sources . . . . .	3
hc_read_daily_flows . . . . .	3
hc_read_daily_levels . . . . .	4
hc_read_flows . . . . .	5
hc_read_levels . . . . .	6
hc_read_stations . . . . .	6
new_hydrocan_adapter . . . . .	7
register_hydrocan_adapter . . . . .	9
<b>Index</b>	<b>10</b>

---

hc_citation	<i>Cite a hydrocan data source</i>
-------------	------------------------------------

---

### Description

Returns a citation for a registered data source, formatted as a `bibentry()` object. The output behaves like `citation()`: it prints a human-readable reference and a BibTeX entry, and can be passed to `toBibtex()`.

### Usage

```
hc_citation(source)
```

### Arguments

source	Single character string naming the data source to cite. See <code>hc_list_sources()</code> for available names.
--------	---

### Value

A `bibentry()` object.

### Examples

```
## Not run:
hc_citation("cehq")
toBibtex(hc_citation("hydroquebec"))

## End(Not run)
```

---

hc_list_sources	<i>List registered data sources</i>
-----------------	-------------------------------------

---

**Description**

Returns a summary of all data sources currently available via hydrocan, including their description and which data types they support. No network calls are made.

**Usage**

```
hc_list_sources()
```

**Value**

A tibble with columns name (chr), description (chr), has\_flows (lgl), has\_daily\_flows (lgl), has\_levels (lgl), has\_daily\_levels (lgl), has\_stations (lgl), license (chr), license\_url (chr), terms\_url (chr), and docs\_url (chr).

**Examples**

```
hc_list_sources()
```

---

hc_read_daily_flows	<i>Retrieve daily flow summaries</i>
---------------------	--------------------------------------

---

**Description**

Returns one value per station per calendar day as published by the data source. Not all data sources publish daily data; those that do not will produce a warning and return no rows for the affected stations.

**Usage**

```
hc_read_daily_flows(  
  station_id,  
  start_date,  
  end_date = Sys.Date(),  
  source = NULL  
)
```

**Arguments**

station_id	Character vector of station identifiers.
start_date	Start of the requested period (Date, or character coercible to Date).
end_date	End of the requested period (Date, or character coercible to Date). Defaults to today.
source	Optional single character string naming the data source to use directly. When NULL (default) the source is detected automatically from the station ID. See <a href="#">hc_list_sources()</a> for available names.

**Value**

A tibble with columns station\_id (chr), date (Date), value (dbl), parameter (chr), unit (chr), provider\_name (chr), quality\_code (chr), and qf\_desc (chr).

**Examples**

```
## Not run:
try(hc_read_daily_flows("030101", start_date = "2020-01-01", end_date = "2020-01-31"))

## End(Not run)
```

---

hc\_read\_daily\_levels *Retrieve daily water level summaries*

---

**Description**

Returns one water level value per station per calendar day as published by the data source. Not all data sources publish daily level data; those that do not will produce a warning and return no rows for the affected stations.

**Usage**

```
hc_read_daily_levels(
  station_id,
  start_date,
  end_date = Sys.Date(),
  source = NULL
)
```

**Arguments**

station_id	Character vector of station identifiers.
start_date	Start of the requested period (Date, or character coercible to Date).
end_date	End of the requested period (Date, or character coercible to Date). Defaults to today.
source	Optional single character string naming the data source to use directly. When NULL (default) the source is detected automatically from the station ID. See <a href="#">hc_list_sources()</a> for available names.

**Value**

A tibble with columns `station_id` (chr), `date` (Date), `value` (dbl), `parameter` (chr: "water\_level"), `unit` (chr), `provider_name` (chr), `quality_code` (chr), and `qf_desc` (chr).

**Examples**

```
## Not run:
try(hc_read_daily_levels("030101", start_date = "2020-01-01", end_date = "2020-01-31"))

## End(Not run)
```

---

<code>hc_read_flows</code>	<i>Retrieve sub-daily flow observations</i>
----------------------------	---

---

**Description**

Fetches sub-daily observations for one or more stations across the requested date range. The data source is determined automatically from the station ID, or fixed explicitly via `source`.

**Usage**

```
hc_read_flows(station_id, start_date, end_date = Sys.Date(), source = NULL)
```

**Arguments**

<code>station_id</code>	Character vector of station identifiers.
<code>start_date</code>	Start of the requested period (Date, or character coercible to Date).
<code>end_date</code>	End of the requested period (Date, or character coercible to Date). Defaults to today.
<code>source</code>	Optional single character string naming the data source to use directly. When NULL (default) the source is detected automatically from the station ID. See <a href="#">hc_list_sources()</a> for available names.

**Value**

A tibble with columns `station_id` (chr), `timestamp` (POSIXct UTC), `value` (dbl), `parameter` (chr), `unit` (chr), `provider_name` (chr), `quality_code` (chr), and `qf_desc` (chr).

**Examples**

```
## Not run:
try(hc_read_flows("703", start_date = Sys.Date() - 7))

## End(Not run)
```

---

hc\_read\_levels      *Retrieve sub-daily water level observations*

---

### Description

Fetches sub-daily water level observations for one or more stations across the requested date range. The data source is determined automatically from the station ID, or fixed explicitly via source.

### Usage

```
hc_read_levels(station_id, start_date, end_date = Sys.Date(), source = NULL)
```

### Arguments

station_id	Character vector of station identifiers.
start_date	Start of the requested period (Date, or character coercible to Date).
end_date	End of the requested period (Date, or character coercible to Date). Defaults to today.
source	Optional single character string naming the data source to use directly. When NULL (default) the source is detected automatically from the station ID. See <a href="#">hc_list_sources()</a> for available names.

### Value

A tibble with columns station\_id (chr), timestamp (POSIXct UTC), value (dbl), parameter (chr: "water\_level"), unit (chr), provider\_name (chr), quality\_code (chr), and qf\_desc (chr).

### Examples

```
## Not run:
try(hc_read_levels("703", start_date = Sys.Date() - 7))

## End(Not run)
```

---

hc\_read\_stations      *Retrieve station metadata*

---

### Description

Returns location and period-of-record information for all stations available across registered data sources. Data sources that do not publish station metadata are skipped with a warning.

### Usage

```
hc_read_stations(source = NULL)
```

## Arguments

source            Optional single character string naming the data source to query directly. When NULL (default) all registered data sources are queried. See [hc\\_list\\_sources\(\)](#) for available names.

## Value

A tibble with columns `station_id` (chr), `station_name` (chr), `provider_name` (chr), `longitude` (dbl), `latitude` (dbl), `elevation_m` (dbl), `period_start` (Date), `period_end` (Date), and `notes` (list).

## Examples

```
## Not run:  
try(hc_read_stations())  
try(hc_read_stations(source = "cehq"))  
  
## End(Not run)
```

---

new\_hydrocan\_adapter    *Create a hydrocan adapter*

---

## Description

Constructs a validated adapter object for a data source. At least one fetch function must be supplied.

## Usage

```
new_hydrocan_adapter(  
  name,  
  description,  
  list_stations_fn,  
  fetch_flows_fn = NULL,  
  fetch_daily_flows_fn = NULL,  
  fetch_levels_fn = NULL,  
  fetch_daily_levels_fn = NULL,  
  list_stations_meta_fn = NULL,  
  title = NULL,  
  publisher = NULL,  
  license = NULL,  
  license_url = NULL,  
  terms_url = NULL,  
  docs_url = NULL  
)
```

**Arguments**

name	Non-empty string identifying this source. Used as the registry key and as the provider_name column in output.
description	String describing the source and any known limitations (e.g. rolling data window). Shown by <code>hc_list_sources()</code> .
list_stations_fn	Function with no arguments returning a character vector of station IDs this source can serve.
fetch_flows_fn	Optional function(station_id, start_date, end_date) returning a tibble matching the flows schema (timestamp column). NULL if sub-daily flow data is not available.
fetch_daily_flows_fn	Optional function(station_id, start_date, end_date) returning a tibble matching the daily flows schema (date column). NULL if daily flow data is not available.
fetch_levels_fn	Optional function(station_id, start_date, end_date) returning a tibble matching the flows schema (timestamp column) with parameter = "water_level". NULL if sub-daily level data is not available.
fetch_daily_levels_fn	Optional function(station_id, start_date, end_date) returning a tibble matching the daily flows schema (date column) with parameter = "water_level". NULL if daily level data is not available.
list_stations_meta_fn	Optional function with no arguments returning a tibble matching the stations schema. NULL if station metadata is not available.
title	Optional string with the formal name of the dataset as published by the provider (used in citations).
publisher	Optional string naming the organization that publishes the data (used in citations).
license	Optional string naming the data license (e.g. "CC-BY 4.0").
license_url	Optional string with a URL to the license text.
terms_url	Optional string with a URL to the data provider's terms of use or data policy.
docs_url	Optional string with a URL to human-readable documentation about the data (field definitions, codes, data structure). A machine-readable metadata endpoint is acceptable if no human-readable page exists.

**Value**

A list with class "hydrocan\_adapter".

---

register\_hydrocan\_adapter

*Register a hydrocan adapter*

---

### **Description**

Adds an adapter to the package registry. Registering under an existing name overwrites it.

### **Usage**

```
register_hydrocan_adapter(adapter)
```

### **Arguments**

adapter            A "hydrocan\_adapter" object from [new\\_hydrocan\\_adapter\(\)](#).

### **Value**

adapter, invisibly.

# Index

`bibentry()`, 2

`citation()`, 2

`hc_citation`, 2

`hc_list_sources`, 3

`hc_list_sources()`, 2, 4–8

`hc_read_daily_flows`, 3

`hc_read_daily_levels`, 4

`hc_read_flows`, 5

`hc_read_levels`, 6

`hc_read_stations`, 6

`new_hydrocan_adapter`, 7

`new_hydrocan_adapter()`, 9

`register_hydrocan_adapter`, 9

`toBibtex()`, 2