Package ‘dataRetrieval’

October 13, 2022

Type           Package
Title          Retrieval Functions for USGS and EPA Hydrologic and Water Quality Data
Version        2.7.11
License        CC0
Copyright      This software is in the public domain because it contains materials that originally came from the United States Geological Survey, an agency of the United States Department of Interior.
Depends        R (>= 3.5.0)
Imports        httr (>= 1.0.0), curl, lubridate (>= 1.5.0), stats, utils, xml2, readr (>= 1.0.0), jsonlite
Suggests       covr, dplyr, ggplot2, DT, gridExtra, knitr, rmarkdown, sf, testthat
Encoding       UTF-8
BuildVignettes true
VignetteBuilder knitr
BugReports     https://github.com/USGS-R/dataRetrieval/issues
URL            https://pubs.usgs.gov/tm/04/a10/
RoxygenNote    7.1.2
NeedsCompilation no
Author         Laura DeCicco [aut, cre] (https://orcid.org/0000-0002-3915-9487), Robert Hirsch [aut] (https://orcid.org/0000-0002-4534-075X),
David Lorenz [aut],
Jordan Read [ctb],
Jordan Walker [ctb],
Lindsay Carr [ctb],
David Watkins [aut] (<https://orcid.org/0000-0002-7544-0700>),
David Blodgett [ctb],
Mike Johnson [aut] (<https://orcid.org/0000-0002-5288-8350>)

Maintainer  Laura DeCicco <ldecicco@usgs.gov>
Repository  CRAN
Date/Publication  2022-02-18 18:20:02 UTC

R topics documented:

addWaterYear ...................................................... 3
calcWaterYear .................................................... 4
constructNWISURL ............................................... 4
constructUseURL ................................................ 6
constructWQPURL ................................................. 7
countyCd .......................................................... 8
countyCdLookup .................................................. 8
dataRetrieval .................................................... 9
findNLDI ............................................................ 9
getQuerySummary .................................................. 11
getWebServiceData ............................................... 12
get_nldi_sources ................................................ 13
importNGWMN ..................................................... 13
importRDB1 ........................................................ 14
importWaterML1 .................................................. 16
importWaterML2 .................................................. 18
importWQP ......................................................... 19
is_dataRetrieval_user ......................................... 20
parameterCdFile .................................................. 21
readNGWMNdata ................................................... 21
readNGWMNlevels ............................................... 22
readNGWMNsites .................................................. 23
readNWISdata ..................................................... 24
readNWISdv ....................................................... 27
readNWISgwl .................................................... 29
readNWISmeas .................................................... 31
readNWISpCode ................................................... 33
readNWISpeak .................................................... 34
readNWISqw ....................................................... 35
readNWISrating ................................................... 38
readNWISsite ..................................................... 39
readNWISstat ..................................................... 41
readNWISsuse .................................................... 43
readNWISuv ....................................................... 44
addWaterYear

Description
Add a column to the dataRetrieval data frame with the water year. WQP queries will return a water year column for the start and end dates of the data.

Usage
addWaterYear(rawData)

Arguments
rawData
the daily- or unit-values dataset retrieved from NWISweb. Must have at least one of the following columns to add the new water year columns: 'dateTime', 'Date', 'ActivityStartDate', or 'ActivityEndDate'. The date column(s) can be character, POSIXct, Date. They cannot be numeric.

Value
data.frame with an additional integer column with "WY" appended to the date column name. For WQP, there will be 2 columns: 'ActivityStartDateWY' and 'ActivityEndDateWY'.

Examples

```r
nwisData <- readNWISdv('04085427','00060','2012-01-01','2012-06-30')
nwisData <- addWaterYear(nwisData)

wqpData <- readWQPqw('USGS-01594440','01075','','')
wqpData <- addWaterYear(wqpData)
```
calcWaterYear  Extract WY from a date

Description
Determine the correct water year based on a calendar date.

Usage
calcWaterYear(dateVec)

Arguments
dateVec  vector of dates as character ("YYYY-DD-MM"), Date, or POSIXct. Numeric
does not work.

Details
This function calculates a water year based on the USGS definition that a water year starts on October 1 of
the year before, and ends on September 30. For example, water year 2015 started on 2014-10-01 and ended on

Value
numeric vector indicating the water year

Examples
x <- seq(as.Date("2010-01-01"), as.Date("2010-12-31"), by="month")
calcWaterYear(x)

calcWaterYear(y)

constructNWISURL  Construct NWIS url for data retrieval

Description
Imports data from NWIS web service. This function gets the data from here: https://nwis.waterdata.usgs.gov/nwis/qwdata A list of parameter codes can be found here: https://nwis.waterdata.usgs.gov/nwis/pmcodes/ A list of statistic codes can be found here: https://nwis.waterdata.usgs.gov/nwis/help/?read_file=stat&format=table
constructNWISURL

Usage

constructNWISURL(
  siteNumbers,
  parameterCd = "00060",
  startDate = "",
  endDate = "",
  service,
  statCd = "00003",
  format = "xml",
  expanded = TRUE,
  ratingType = "base",
  statReportType = "daily",
  statType = "mean"
)

Arguments

siteNumbers string or vector of strings USGS site number. This is usually an 8 digit number
parameterCd string or vector of USGS parameter code. This is usually an 5 digit number.
startDate character starting date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the earliest possible record.
endDate character ending date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the latest possible record.
service string USGS service to call. Possible values are "dv" (daily values), "uv" (unit/instantaneous values), "qw" (water quality data), "gwlevels" (groundwater), and "rating" (rating curve), "peak", "meas" (discrete streamflow measurements), "stat" (statistics web service BETA).
statCd string or vector USGS statistic code only used for daily value service. This is usually 5 digits. Daily mean (00003) is the default.
format string, can be "tsv" or "xml", and is only applicable for daily and unit value requests. "tsv" returns results faster, but there is a possibilit that an incomplete file is returned without warning. XML is slower, but will offer a warning if the file was incomplete (for example, if there was a momentary problem with the internet connection). It is possible to safely use the "tsv" option, but the user must carefully check the results to see if the data returns matches what is expected. The default is therefore "xml".
expanded logical defaults to TRUE. If TRUE, retrieves additional information, only applicable for qw data.
ratingType can be "base", "corr", or "exsa". Only applies to rating curve data.
statReportType character Only used for statistics service requests. Time division for statistics: daily, monthly, or annual. Default is daily. Note that daily provides statistics for each calendar day over the specified range of water years, i.e. no more than 366 data points will be returned for each site/parameter. Use readNWISdata or readNWISdv for daily averages. Also note that 'annual' returns statistics for the calendar year. Use readNWISdata for water years. Monthly and yearly provide statistics for each month and year within the range individually.
constructUseURL

Construct URL for NWIS water use data service

Description
Reconstructs URLs to retrieve data from here: https://waterdata.usgs.gov/nwis/wu

Usage
constructUseURL(years, stateCd, countyCd, categories)

Arguments
years integer Years for data retrieval. Must be years ending in 0 or 5, or "ALL", which retrieves all available years.
stateCd could be character (full name, abbreviation, id), or numeric (id)
countyCd could be numeric (County IDs from countyCdLookup) or character ("ALL")
categories character Two-letter category abbreviation(s)
constructWQPURL

Value
url string

Examples
url <- constructUseURL(years=c(1990,1995),stateCd="Ohio",countyCd = c(1,3), categories = "ALL")

Description
Construct WQP url for data retrieval. This function gets the data from here: https://www.waterqualitydata.us

Usage
constructWQPURL(siteNumbers, parameterCd, startDate, endDate, zip = TRUE)

Arguments
siteNumbers  string or vector of strings USGS site number. This is usually an 8 digit number
parameterCd  string or vector of USGS parameter code. This is usually an 5 digit number.
startDate  character starting date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the earliest possible record.
endDate  character ending date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the latest possible record.
zip  logical to request data via downloading zip file. Default set to TRUE.

Value
url string

Examples
site_id <- '01594440'
startDate <- '1985-01-01'
endDate <- '

pCode <- c("00060","00010")
url_wqp <- constructWQPURL(paste("USGS",site_id,sep="-"),
c('01075','00029','00453'),
startDate,endDate)
url_wqp

charNames <- c("Temperature",
"Temperature, sample",
"Temperature, water",
"Temperature, water,
"Temperature, water,
"Temperature, water,
"Temperature, water,
"Temperature, water,
"Temperature, water,
"Temperature, water,


countyCd <- constructWQPURL(siteNumbers = c("IIDFG-41WSSPAHS", "USGS-02352560"), parameterCd = charNames, startDate,"")

countyCd

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STUSAB</td>
<td>character</td>
<td>State abbreviation</td>
</tr>
<tr>
<td>STATE</td>
<td>character</td>
<td>two-digit ANSI code</td>
</tr>
<tr>
<td>COUNTY</td>
<td>character</td>
<td>three-digit county code</td>
</tr>
<tr>
<td>COUNTY_NAME</td>
<td>character</td>
<td>County full name</td>
</tr>
<tr>
<td>COUNTY_ID</td>
<td>character</td>
<td>County id</td>
</tr>
</tbody>
</table>

Examples

head(countyCd)

countyCdLookup

Description

Function to simplify finding county and county code definitions. Used in readNWISdata and readNWISuse.

Usage

countyCdLookup(state, county, outputType = "id")

countyCdLookup

Description

Function to simplify finding county and county code definitions. Used in readNWISdata and readNWISuse.

Usage

countyCdLookup(state, county, outputType = "id")
Arguments

state could be character (full name, abbreviation, id), or numeric (id)
county could be character (name, with or without "County") or numeric (id)
outputType character can be "fullName", "tableIndex", "id", or "fullEntry".

Examples

id <- countyCdLookup(state = "WI", county = "Dane")
name <- countyCdLookup(state = "OH", county = 13, output = "fullName")
index <- countyCdLookup(state = "Pennsylvania", county = "ALLEGHENY COUNTY", output = "tableIndex")
fromIDs <- countyCdLookup(state = 13, county = 5, output = "fullName")
already_correct <- countyCdLookup(county = "51001")

Description

Package: dataRetrieval
Type: Package
License: Unlimited for this package, dependencies have more restrictive licensing.
Copyright: This software is in the public domain because it contains materials that originally came from the United States Government.
LazyLoad: yes

Details

Retrieval functions for USGS and EPA hydrologic and water quality data.

Please see https://pubs.er.usgs.gov/publication/tm4A10 for more information.

Author(s)

Laura De Cicco <ldecicco@usgs.gov>

findNLDI R Client for the Network Linked Data Index

Description

Provides a formal client to the USGS Network Linked Data Index.
Usage

```r
findNLDI(
  comid = NULL,
  nwis = NULL,
  wqp = NULL,
  huc12 = NULL,
  location = NULL,
  origin = NULL,
  nav = NULL,
  find = c("flowlines"),
  distance_km = 100,
  no_sf = FALSE
)
```

Arguments

- **comid**: numeric or character. An NHDPlusV2 COMID
- **nwis**: numeric or character. A USGS NWIS surface water siteID
- **wqp**: numeric or character. A water quality point ID
- **huc12**: numeric or character. A WBD HUC12 unit ID
- **location**: numeric vector. Coordinate pair in WGS84 SRS ordered lng/lat (X,Y)
- **origin**: named list. Specifying a feature type and ID (e.g. list("comid" = 101))
- **nav**: character vector. where to navigate from the starting point. Options include along the upper mainstream (UM), upstream tributary (UT), downstream mainstream (DM) and downstream divergences (DD). You may select one or more of the abbreviations ("UM", "UT", "DM", "DD").
- **find**: character vector. Define what resources to find along the navigation path(s) (see `get_nldi_sources()`$source). Can also include 'basin' or 'flowline', which will return the upstream basin of the starting feature or flowlines along the navigation respectively. The default is "flowlines". If you provide any other resource, AND want flowlines, then flowlines must be explicitly requested.
- **distance_km**: numeric. Define how far to look along the navigation path in kilometers (default = 100)
- **no_sf**: if available, should ‘sf’ be used for parsing, defaults to ‘TRUE’ if ‘sf’ is locally installed

Details

The function is useful for topology and location based feature discovery. A user must specify an origin feature, optional navigation direction(s) along the network, as well as features to identify along the navigated paths. Valid starting options can be given by one of the following arguments: comid, nwis, huc12, wqp, location, and start.

Value

a list of data.frames if sf is not installed, a list of sf objects if it is
Examples

```r
# Find Features / Define origin features

## Find feature by COMID
findNLDI(comid = 101)

## Find feature by NWIS ID
findNLDI(nwis = '11120000')

## Find feature by WQP ID
findNLDI(wqp = 'USGS-04024315')

## Find feature by LOCATION
findNLDI(location = c(-115, 40))

## GENERAL ORIGIN: COMID
findNLDI(origin = list("comid" = 101))

## GENERAL ORIGIN: WaDE
findNLDI(origin = list("wade" = 'CA_45206'))

# Navigation (flowlines will be returned if find is unspecified)
# UPPER MAINSTEM of USGS-11120000
findNLDI(nwis = '11120000', nav = "UM")

# MULTI-REQUEST
# UPPER MAINSTEM and TRIBUTARY of USGS-11120000
findNLDI(nwis = '11120000', nav = c("UT", "UM"))

# Discover Features (flowlines will not be returned unless included in find)

## Find feature(s) on the upper tributary of USGS-11120000
findNLDI(nwis = '11120000', nav = "UT", find = c("nwis", "wqp"))

## Find upstream basin boundary and of USGS-11120000
findNLDI(nwis = '11120000', find = "basin")

# Control Distance
## Limit search to 50 km
findNLDI(comid = 101, nav = "DM", find = c("nwis", "wqp", "flowlines"), distance_km = 50)
```

Description

`getQuerySummary` getting header information from a WQP query
**Usage**

getWebServiceData

**Arguments**

url the query url

---

**getWebServiceData**  
*Function to return data from web services*

**Description**

This function accepts a url parameter, and returns the raw data. The function enhances GET with more informative error messages.

**Usage**

getWebServiceData(obs_url, ...)

**Arguments**

obs_url character containing the url for the retrieval

... information to pass to header request

**Value**

raw data from web services

**Examples**

```
siteNumber <- "02177000"
startDate <- "2012-09-01"
endDate <- "2012-10-01"
offering <- '00003'
property <- '00060'
obs_url <- constructNWISURL(siteNumber,property,startDate,endDate,'dv')

rawData <- getWebServiceData(obs_url)
```
### get_nldi_sources

**Get current NLDI offerings**

**Description**

Used to query the current resources available through the NLDI

**Usage**

```r
get_nldi_sources()
```

**Value**

data.frame

**Examples**

```r
get_nldi_sources()
```

### importNGWMN

**Function to return data from the National Ground Water Monitoring Network waterML2 format**

**Description**

This function accepts a url parameter for a WaterML2 getObservation. This function is still under development, but the general functionality is correct.

**Usage**

```r
importNGWMN(input, asDateTime = FALSE, tz = "UTC")
```

**Arguments**

- **input** character or raw, containing the url for the retrieval or a path to the data file, or raw XML.
- **asDateTime** logical, if TRUE returns date and time as POSIXct, if FALSE, character
- **tz** character to set timezone attribute of dateTime. Default is "UTC", and converts the date times to UTC, properly accounting for daylight savings times based on the data's provided time zone offset. Possible values to provide are "America/New_York", "America/Chicago", "America/Denver", "America/Los_Angeles", "America/Anchorage", as well as the following which do not use daylight savings time: "America/Honolulu", "America/Jamaica", "America/Managua", "America/Phoenix", and "America/Metlakatla". See also OlsonNames() for more information on time zones.
importRDB1

**Function to return data from the NWIS RDB 1.0 format**

**Description**

This function accepts a url parameter that already contains the desired NWIS site, parameter code, statistic, startdate and enddate. It is not recommended to use the RDB format for importing multi-site data.

**Usage**

```r
importRDB1(obs_url, asDateTime = TRUE, convertType = TRUE, tz = "UTC")
```

**Arguments**

- `obs_url`: character containing the url for the retrieval or a file path to the data file.
- `asDateTime`: logical, if TRUE returns date and time as POSIXct, if FALSE, Date
- `convertType`: logical, defaults to TRUE. If TRUE, the function will convert the data to dates, datetimes, numerics based on a standard algorithm. If false, everything is returned as a character

**Value**

A data frame with the following columns:

- mergedDF
  - a data frame
  - source, time, value, uom, uomTitle, comment, gmlID

**Examples**

```r
  "service=SOS","version=2.0.0",
  "observedProperty=urn:ogc:def:property:OGC:GroundWaterLevel",
  "responseFormat=text/xml",
  "featureOfInterest=VW_GWDP_GEOSERVER.USGS.403836085374401",sep="&")

data_returned <- importNGWMN(obs_url)
```
### importRDB1

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agency_cd</td>
<td>character</td>
<td>The NWIS code for the agency reporting the data</td>
</tr>
<tr>
<td>site_no</td>
<td>character</td>
<td>The USGS site number</td>
</tr>
<tr>
<td>datetime</td>
<td>POSIXct</td>
<td>The date and time of the value converted to UTC (if asDateTime = TRUE)</td>
</tr>
<tr>
<td>dt</td>
<td>character</td>
<td>or raw character string (if asDateTime = FALSE)</td>
</tr>
<tr>
<td>tz_cd</td>
<td>character</td>
<td>The time zone code for datetime</td>
</tr>
<tr>
<td>code</td>
<td>character</td>
<td>Any codes that qualify the corresponding value</td>
</tr>
<tr>
<td>value</td>
<td>numeric</td>
<td>The numeric value for the parameter</td>
</tr>
<tr>
<td>tz_cd_reported</td>
<td></td>
<td>The originally reported time zone</td>
</tr>
</tbody>
</table>

Note that code and value are repeated for the parameters requested. The names are of the form XD_P_S, where X is literal, D is an option description of the parameter, P is the parameter code, and S is the statistic code (if applicable). If a date/time (dt) column contained incomplete date and times, a new column of dates and time was inserted. This could happen when older data was reported as dates, and newer data was reported as a date/time.

There are also several useful attributes attached to the data frame:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>character</td>
<td>The url used to generate the data</td>
</tr>
<tr>
<td>queryTime</td>
<td>POSIXct</td>
<td>The time the data was returned</td>
</tr>
<tr>
<td>comment</td>
<td>character</td>
<td>Header comments from the RDB file</td>
</tr>
</tbody>
</table>

#### Examples

```r
site_id <- "02177000"
startDate <- "2012-09-01"
endDate <- "2012-10-01"
offering <- "00003"
property <- "00060"

obs_url <- constructNWISURL(site_id, property, startDate, endDate, "dv", format="tsv")
data <- importRDB1(obs_url)

urlMultiPcodes <- constructNWISURL("04085427", c("00060", "00010"), startDate, endDate, "dv", statCd=c("00003", "00001"), format="tsv")
multiData <- importRDB1(urlMultiPcodes)

unitDataURL <- constructNWISURL(site_id, property, "2020-10-30", "2020-11-01", "uv", format="tsv") #includes timezone switch
unitData <- importRDB1(unitDataURL, asDateTime=TRUE)
qwURL <- constructNWISURL(c("04024430", "04024000"),
```

```
```r
importWaterML1

Function to return data from the NWISWeb WaterML1.1 service

Description
This function accepts a url parameter that already contains the desired NWIS site, parameter code, statistic, startdate and enddate.

Usage
importWaterML1(obs_url, asDateTime = FALSE, tz = "UTC")

Arguments

obs_url character or raw, containing the url for the retrieval or a file path to the data file, or raw XML.
asDateTime logical, if TRUE returns date and time as POSIXct, if FALSE, Date

Value
A data frame with the following columns:
```
importWaterML1

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agency_cd</td>
<td>character</td>
<td>The NWIS code for the agency reporting the data</td>
</tr>
<tr>
<td>site_no</td>
<td>character</td>
<td>The USGS site number</td>
</tr>
<tr>
<td>POSIXct</td>
<td>character</td>
<td>The date and time of the value converted to UTC (if asDateTime = TRUE), or raw character string (if asDateTime = FALSE)</td>
</tr>
<tr>
<td>tz_cd</td>
<td>character</td>
<td>The time zone code for</td>
</tr>
<tr>
<td>code</td>
<td>character</td>
<td>Any codes that qualify the corresponding value</td>
</tr>
<tr>
<td>value</td>
<td>numeric</td>
<td>The numeric value for the parameter</td>
</tr>
</tbody>
</table>

Note that code and value are repeated for the parameters requested. The names are of the form X_D_P_S, where X is literal, D is an option description of the parameter, P is the parameter code, and S is the statistic code (if applicable).

There are also several useful attributes attached to the data frame:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>character</td>
<td>The url used to generate the data</td>
</tr>
<tr>
<td>siteInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested sites</td>
</tr>
<tr>
<td>variableInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested parameters</td>
</tr>
<tr>
<td>statisticInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested statistics on the data</td>
</tr>
<tr>
<td>queryTime</td>
<td>POSIXct</td>
<td>The time the data was returned</td>
</tr>
</tbody>
</table>

See Also

renameNWISColumns

Examples

```r
site_id <- "02177000"
startDate <- "2012-09-01"
endDate <- "2012-10-01"
offering <- '00003'
property <- '00060'
obs_url <- constructNWISURL(site_id,property,startDate,endDate,'dv')
data <- importWaterML1(obs_url, asDateTime=TRUE)

groundWaterSite <- "431049071324301"
startGW <- "2013-10-01"
endGW <- "2014-06-30"
groundwaterExampleURL <- constructNWISURL(groundWaterSite, NA, startGW,endGW, service="gwlevels")
groundWater <- importWaterML1(groundwaterExampleURL)
groundWater2 <- importWaterML1(groundwaterExampleURL, asDateTime=TRUE)

unitDataURL <- constructNWISURL(site_id,property,"2013-11-03","2013-11-03","uv")
unitData <- importWaterML1(unitDataURL,TRUE)
```
importWaterML2

Parse the WaterML2 timeseries portion of a waterML2 file

Description

Returns data frame columns of all information with each time series measurement; Anything defined as a default, is returned as an attribute of that data frame.

Usage

importWaterML2(input, asDateTime = FALSE, tz = "UTC")
**Arguments**

- **input** XML with only the `wml2:MeasurementTimeseries` node and children
- **asDateTime** logical, if TRUE returns date and time as POSIXct, if FALSE, character
- **tz** character to set timezone attribute of datetime. Default is an empty quote, which converts the datetimes to UTC (properly accounting for daylight savings times based on the data’s provided time zone offset). Possible values are "America/New_York", "America/Chicago", "America/Denver", "America/Los_Angeles", "America/Anchorage", "America/Honolulu", "America/Jamaica", "America/Managua", "America/Phoenix", and "America/Metlakatla"

**Examples**

```r
baseURL <- "https://waterservices.usgs.gov/nwis/dv/?format=waterml,2.0"
URL <- paste(baseURL, "sites=01646500",
  "startDT=2014-09-01",
  "endDT=2014-09-08",
  "statCd=00003",
  "parameterCd=00060",sep="&")

timesereies <- importWaterML2(URL, asDateTime=TRUE, tz="UTC")
```

**Description**

Imports data from the Water Quality Portal based on a specified url.

**Usage**

```r
importWQP(obs_url, zip = TRUE, tz = "UTC", csv = FALSE)
```

**Arguments**

- **obs_url** character URL to Water Quality Portal
- **zip** logical to request data via downloading zip file. Default set to TRUE.
- **tz** character to set timezone attribute of datetime. Default is UTC (properly accounting for daylight savings times based on the data’s provided tz_cd column). Possible values include "America/New_York", "America/Chicago", "America/Denver", "America/Los_Angeles", "America/Anchorage", "America/Honolulu", "America/Jamaica", "America/Managua", "America/Phoenix", and "America/Metlakatla"
is_dataRetrieval_user

csv logical. Is the data coming back with a csv or tsv format. Default is FALSE. Currently, the summary service does not support tsv, for other services tsv is the safer choice.

Value

return dataframe raw data returned from the Water Quality Portal. Additionally, a POSIXct dateTime column is supplied for start and end times, and converted to UTC. See https://www.waterqualitydata.us/portal_userguide/ for more information.

See Also

readWQPdata, readWQPqw, whatWQPsites

Examples

# These examples require an internet connection to run

## Examples take longer than 5 seconds:

rawSampleURL <- constructWQPURL('USGS-01594440','01075', '', '')
rawSample <- importWQP(rawSampleURL)

rawSampleURL_NoZip <- constructWQPURL('USGS-01594440','01075', '', '', zip=FALSE)
rawSample2 <- importWQP(rawSampleURL_NoZip, zip=FALSE)

STORETdata <- importWQP(STORETdata)
**ParameterCdFile**

*List of USGS parameter codes*

**Description**

Complete list of USGS parameter codes as of Dec. 20, 2021.

**Value**

ParameterData data frame with information about USGS parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>parameter_cd</td>
<td>character</td>
<td>5-digit USGS parameter code</td>
</tr>
<tr>
<td>parameter_group_nm</td>
<td>character</td>
<td>USGS parameter group name</td>
</tr>
<tr>
<td>parameter_nm</td>
<td>character</td>
<td>USGS parameter name</td>
</tr>
<tr>
<td>casrn</td>
<td>character</td>
<td>Chemical Abstracts Service (CAS) Registry Number</td>
</tr>
<tr>
<td>srsname</td>
<td>character</td>
<td>Substance Registry Services Name</td>
</tr>
<tr>
<td>parameter_units</td>
<td>character</td>
<td>Parameter units</td>
</tr>
</tbody>
</table>

**Examples**

```r
head(parameterCdFile[,1:2])
```

---

**readNGWMNdata**

*import data from the National Groundwater Monitoring Network*

[https://cida.usgs.gov/ngwmn/](https://cida.usgs.gov/ngwmn/)

**Description**

Only water level data and site locations and names are currently available through the web service.

**Usage**

```r
readNGWMNdata(service, ..., asDateTime = TRUE, tz = "UTC")
```

**Arguments**

- **service** char Service for the request - "observation" and "featureOfInterest" are implemented.
- **...** Other parameters to supply, namely siteNumbers or bbox
- **asDateTime** logical if TRUE, will convert times to POSIXct format. Currently defaults to FALSE since time zone information is not included.
readNGWMNlevels

Character to set timezone attribute of dateTime. Default is "UTC", and converts the date times to UTC, properly accounting for daylight savings times based on the data's provided time zone offset. Possible values to provide are "America/New_York", "America/Chicago", "America/Denver", "America/Los_Angeles", "America/Anchorage", as well as the following which do not use daylight savings time: "America/Honolulu", "America/Jamaica", "America/Managua", "America/Phoenix", and "America/Metlakatla". See also OlsonNames() for more information on time zones.

Examples

#one site
site <- "USGS.430427089284901"
# oneSite <- readNGWMNdata(siteNumbers = site, service = "observation")

#multiple sites
sites <- c("USGS.272838082142201", "USGS.404159100494601", "USGS.401216080362703")
# Very slow:
# multiSiteData <- readNGWMNdata(siteNumbers = sites, service = "observation")
# attributes(multiSiteData)

#non-USGS site
# accepts colon or period between agency and ID
site <- "MBMG:702934"
# data <- readNGWMNdata(siteNumbers = site, service = "featureOfInterest")

#site with no data returns empty data frame
noDataSite <- "UTGS.401544112060301"
# noDataSite <- readNGWMNdata(siteNumbers = noDataSite, service = "observation")

#bounding box
#bboxSites <- readNGWMNdata(service = "featureOfInterest", bbox = c(30, -102, 31, 99))
# retrieve sites. Set asDateTime to false since one site has an invalid date
# Very slow:
#bboxData <- readNGWMNdata(service = "observation", siteNumbers = bboxSites$site[1:3],
# asDateTime = FALSE)

---

readNGWMNlevels

Retrieve groundwater levels from the National Ground Water Monitoring Network https://cida.usgs.gov/ngwmn/.

Description

Retrieve groundwater levels from the National Ground Water Monitoring Network https://cida.usgs.gov/ngwmn/. 
readNGWMNsites


**Usage**

```r
readNGWMNlevels(siteNumbers, asDateTime = TRUE, tz = "UTC")
```

**Arguments**

- `siteNumbers` character Vector of feature IDs formatted with agency code and site number separated by a period or semicolon, e.g. USGS.404159100494601.
- `asDateTime` logical Should dates and times be converted to date/time objects, or returned as character? Defaults to TRUE. Must be set to FALSE if a site contains non-standard dates.
- `tz` character to set timezone attribute of dateTime. Default is "UTC", and converts the date times to UTC, properly accounting for daylight savings times based on the data’s provided time zone offset. Possible values to provide are "America/New_York", "America/Chicago", "America/Denver", "America/Los_Angeles", "America/Anchorage", as well as the following which do not use daylight savings time: "America/Honolulu", "America/Jamaica", "America/Managua", "America/Phoenix", and "America/Metlakatla". See also OlsonNames() for more information on time zones.

**Examples**

```r
# one site
site <- "USGS.430427089284901"
#oneSite <- readNGWMNlevels(siteNumbers = site)

# multiple sites
sites <- c("USGS:272838082142201", "USGS:404159100494601", "USGS:401216080362703")
#multiSiteData <- readNGWMNlevels(sites)

# non-USGS site
site <- "MBMG.103306"
#data <- readNGWMNlevels(siteNumbers = site, asDateTime = FALSE)

# site with no data returns empty data frame
noDataSite <- "UTGS.401544112060301"
#noDataSite <- readNGWMNlevels(siteNumbers = noDataSite)
```

---

**Description**

Usage

readNGWMNsites(siteNumbers)

Arguments

siteNumbers character Vector of feature IDs formatted with agency code and site number separated by a period or semicolon, e.g. USGS.404159100494601.

Value

A data frame the following columns: #'

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>site</td>
<td>char</td>
<td>Site FID</td>
</tr>
<tr>
<td>description</td>
<td>char</td>
<td>Site description</td>
</tr>
<tr>
<td>dec_lat_va, dec_lon_va</td>
<td>numeric</td>
<td>Site latitude and longitude</td>
</tr>
</tbody>
</table>

Examples

# one site
site <- "USGS.430427089284901"
oneSite <- readNGWMNsites(siteNumbers = site)

# non-USGS site
site <- "MBMG.103306"
siteInfo <- readNGWMNsites(siteNumbers = site)

readNWISdata General Data Import from NWIS

Description

Returns data from the NWIS web service. Arguments to the function should be based on https://waterservices.usgs.gov service calls. See examples below for ideas of constructing queries.

Usage

readNWISdata(..., asDateTime = TRUE, convertType = TRUE, tz = "UTC")

Arguments

... see https://waterservices.usgs.gov/rest/Site-Service.html for a complete list of options. A list of arguments can also be supplied. One important argument to include is 'service'. Possible values are "iv" (for instantaneous), "iv_recent" (for instantaneous values within the last 120 days), "dv"
readNWISdata

(for daily values), "gwelevels" (for groundwater levels), "site" (for site service), "qw" (water-quality), "measurement", and "stat" (for statistics service). Note: "qw" and "measurement" calls go to: https://nwis.waterdata.usgs.gov/usa/nwis for data requests, and use different call requests schemes. The statistics service has a limited selection of arguments (see https://waterservices.usgs.gov/rest/Statistics-Service-Test-Tool.html).

asDateTime  logical, if TRUE returns date and time as POSIXct, if FALSE, Date
convertType  logical, defaults to TRUE. If TRUE, the function will convert the data to dates, datetimes, numerics based on a standard algorithm. If false, everything is returned as a character
tz  character to set timezone attribute of dateTime. Default is "UTC", and converts the date times to UTC, properly accounting for daylight savings times based on the data's provided tz_cd column. Possible values to provide are "America/New_York", "America/Chicago", "America/Denver", "America/Los_Angeles", "America/Anchorage", as well as the following which do not use daylight savings time: "America/Honolulu", "America/Jamaica", "America/Managua", "America/Phoenix", and "America/Metlakatla". See also OlsonNames() for more information on time zones.

Value

A data frame with the following columns:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agency</td>
<td>character</td>
<td>The NWIS code for the agency reporting the data</td>
</tr>
<tr>
<td>site</td>
<td>character</td>
<td>The USGS site number</td>
</tr>
<tr>
<td>dateTime</td>
<td>POSIXct</td>
<td>The date and time (if applicable) of the measurement, converted to UTC for unit value data. R only all time zones are converted to UTC solves this problem. For daily data, the time zone attribute is the time zone of the first returned measurement.</td>
</tr>
<tr>
<td>tz_cd</td>
<td>character</td>
<td>The time zone code for dateTime column</td>
</tr>
<tr>
<td>code</td>
<td>character</td>
<td>Any codes that qualify the corresponding value</td>
</tr>
<tr>
<td>value</td>
<td>numeric</td>
<td>The numeric value for the parameter</td>
</tr>
</tbody>
</table>

Note that code and value are repeated for the parameters requested. The names are of the form X_D_P_S, where X is literal, D is an option description of the parameter, P is the parameter code, and S is the statistic code (if applicable).

There are also several useful attributes attached to the data frame:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>character</td>
<td>The url used to generate the data</td>
</tr>
<tr>
<td>siteInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested sites</td>
</tr>
<tr>
<td>variableInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested parameters</td>
</tr>
<tr>
<td>statisticInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested statistics on the data</td>
</tr>
<tr>
<td>queryTime</td>
<td>POSIXct</td>
<td>The time the data was returned</td>
</tr>
</tbody>
</table>
Examples

```r
# Examples not run for time considerations

dataTemp <- readNWISdata(stateCd="OH",parameterCd="00010", service="dv")
instFlow <- readNWISdata(sites="05114000", service="iv",
parameterCd="00060",
startDate="2014-05-01T00:00Z",endDate="2014-05-01T12:00Z")

instFlowCDT <- readNWISdata(sites="05114000", service="iv",
parameterCd="00060",
startDate="2014-05-01T00:00",endDate="2014-05-01T12:00",
tz="America/Chicago")

#Empty:
multiSite <- readNWISdata(sites=c("04025000","04072150"), service="iv",
parameterCd="00010")

#Not empty:
multiSite <- readNWISdata(sites=c("04025500","040263491"),
service="iv", parameterCd="00060")
bBoxEx <- readNWISdata(bBox=c(-83,36.5,-81,38.5), parameterCd="00010")

startDate <- as.Date("2013-10-01")
endDate <- as.Date("2014-09-30")
waterYear <- readNWISdata(bBox=c(-83,36.5,-82.5,36.75), parameterCd="00010",
service="dv", startDate=startDate, endDate=endDate)
siteInfo <- readNWISdata(stateCd="WI", parameterCd="00010",
hasDataTypeCd="iv", service="site")
temp <- readNWISdata(bBox=c(-83,36.5,-82.5,36.75), parameterCd="00010", service="site",
seriesCatalogOutput=TRUE)

wiGWL <- readNWISdata(stateCd = "WI", service = "gwlevels")
meas <- readNWISdata(state_cd = "WI", service = "measurements",
format = "rdb_expanded")

waterYearStat <- readNWISdata(site = c("01646500"),
service = "stat",
statReportType="annual",
statYearType = "water",
missingData = "on")

monthlyStat <- readNWISdata(site=c("01646500"),
service="stat",
statReportType="monthly")

dailyStat <- readNWISdata(site = c("01646500"),
service = "stat",
statReportType = "daily",
statType = c("p25","p50","p75","min","max"),
"mean","median","stddev","max","min","percentile")
```

See Also

renameNWISColumns, importWaterML1, importRDB1
arg.list <- list(site = "03111548", 
    statReportType = "daily", 
    statType = c("p25","p50","p75","min","max"), 
    parameterCd = "00060")

allDailyStats_2 <- readNWISdata(arg.list, service="stat")

# use county names to get data
dailyStaffordVA <- readNWISdata(stateCd = "Virginia", 
    countyCd="Stafford", 
    parameterCd = "00060", 
    startDate = "2015-01-01", 
    endDate = "2015-01-30")

va_counties <- c("51001","51003","51005","51007","51009","51011","51013","51015")
va_counties_data <- readNWISdata(startDate = "2015-01-01", endDate = "2015-12-31", 
    parameterCd = "00060", countycode = va_counties)

site_id <- '01594440'

rating_curve <- readNWISdata(service = "rating", site_no = site_id, file_type="base")

all_sites_base <- readNWISdata(service = "rating", file_type="base")

all_sites_core <- readNWISdata(service = "rating", file_type="corr")

all_sites_exsa <- readNWISdata(service = "rating", file_type="exsa")

all_sites_24hrs <- readNWISdata(service = "rating", file_type="exsa", period = 24)

peak_data <- readNWISdata(service = "peak", 
    site_no = c("01594440","040851325"),
    range_selection = "data_range")

---

## readNWISdv

**Daily Value USGS NWIS Data Retrieval**

### Description

Imports data from NWIS web service. This function gets the data from here: [https://waterservices.usgs.gov/](https://waterservices.usgs.gov/)

### Usage

```r
readNWISdv(
    siteNumbers, 
    parameterCd, 
    startDate = "", 
    endDate = "", 
    statCd = "00003"
)
```
Arguments

- **siteNumbers**: character USGS site number. This is usually an 8 digit number. Multiple sites can be requested with a character vector.
- **parameterCd**: character of USGS parameter code(s). This is usually an 5 digit number.
- **startDate**: character starting date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the earliest possible record. Date arguments are always specified in local time.
- **endDate**: character ending date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the latest possible record. Date arguments are always specified in local time.
- **statCd**: character USGS statistic code. This is usually 5 digits. Daily mean (00003) is the default.

Value

A data frame with the following columns:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agency</td>
<td>character</td>
<td>The NWIS code for the agency reporting the data</td>
</tr>
<tr>
<td>site</td>
<td>character</td>
<td>The USGS site number</td>
</tr>
<tr>
<td>Date</td>
<td>Date</td>
<td>The date of the value</td>
</tr>
<tr>
<td>code</td>
<td>character</td>
<td>Any codes that qualify the corresponding value</td>
</tr>
<tr>
<td>value</td>
<td>numeric</td>
<td>The numeric value for the parameter</td>
</tr>
</tbody>
</table>

Note that code and value are repeated for the parameters requested. The names are of the form X_D_P_S, where X is literal, D is an option description of the parameter, P is the parameter code, and S is the statistic code (if applicable).

There are also several useful attributes attached to the data frame:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>character</td>
<td>The url used to generate the data</td>
</tr>
<tr>
<td>siteInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested sites</td>
</tr>
<tr>
<td>variableInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested parameters</td>
</tr>
<tr>
<td>statisticInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested statistics on the data</td>
</tr>
<tr>
<td>queryTime</td>
<td>POSIXct</td>
<td>The time the data was returned</td>
</tr>
</tbody>
</table>

See Also

- renameNWISColumns, importWaterML1

Examples

```r
site_id <- '04085427'
startDate <- '2012-01-01'
```
readNWISgwl

Groundwater level measurements retrieval from USGS (NWIS)

Description

Reads groundwater level measurements from NWISweb. Mixed date/times come back from the service depending on the year that the data was collected. See https://waterdata.usgs.gov/usa/nwis/gw for details about groundwater. By default the returned dates are converted to date objects, unless convertType is specified as FALSE. Sites with non-standard date formats (i.e. lacking a day) can be affected (see examples). See https://waterservices.usgs.gov/rest/GW-Levels-Service.html for more information.

Usage

readNWISgwl(
  siteNumbers,
  startDate = '',
  endDate = '',
  parameterCd = NA,
  convertType = TRUE,
  tz = 'UTC'
)

Arguments

- siteNumbers: character USGS site number (or multiple sites). This is usually an 8 digit number
- startDate: character starting date for data retrieval in the form YYYY-MM-DD. Default is '' which indicates retrieval for the earliest possible record.
readNWISgwl

**endDate** character ending date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the latest possible record.

**parameterCd** character USGS parameter code. This is usually an 5 digit number. Default is "".

**convertType** logical, defaults to TRUE. If TRUE, the function will convert the data to dates, datetimes, numerics based on a standard algorithm. If false, everything is returned as a character

**tz** character to set timezone attribute of dateTime. Default is "UTC", and converts the date times to UTC, properly accounting for daylight savings times based on the data’s provided tz_cd column. Possible values to provide are "America/New_York", "America/Chicago", "America/Denver", "America/Los_Angeles", "America/Anchorage", as well as the following which do not use daylight savings time: "America/Honolulu", "America/Jamaica", "America/Managua", "America/Phoenix", and "America/Metlakatla". See also OlsonNames() for more information on time zones.

**Value**

A data frame with the following columns:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agency_cd</td>
<td>character</td>
<td>The NWIS code for the agency reporting the data</td>
</tr>
<tr>
<td>site_no</td>
<td>character</td>
<td>The USGS site number</td>
</tr>
<tr>
<td>site_tp_cd</td>
<td>character</td>
<td>Site type code</td>
</tr>
<tr>
<td>lev_dt</td>
<td>Date</td>
<td>Date level measured</td>
</tr>
<tr>
<td>lev_tm</td>
<td>character</td>
<td>Time level measured</td>
</tr>
<tr>
<td>lev_tz_cd</td>
<td>character</td>
<td>Time datum</td>
</tr>
<tr>
<td>lev_va</td>
<td>numeric</td>
<td>Water level value in feet below land surface</td>
</tr>
<tr>
<td>sl_lev_va</td>
<td>numeric</td>
<td>Water level value in feet above specific vertical datum</td>
</tr>
<tr>
<td>lev_status_cd</td>
<td>character</td>
<td>The status of the site at the time the water level was measured</td>
</tr>
<tr>
<td>lev_agency_cd</td>
<td>character</td>
<td>The agency code of the person measuring the water level</td>
</tr>
</tbody>
</table>

There are also several useful attributes attached to the data frame:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>character</td>
<td>The url used to generate the data</td>
</tr>
<tr>
<td>queryTime</td>
<td>POSIXct</td>
<td>The time the data was returned</td>
</tr>
<tr>
<td>comment</td>
<td>character</td>
<td>Header comments from the RDB file</td>
</tr>
<tr>
<td>siteInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested sites</td>
</tr>
</tbody>
</table>

**See Also**

constructNWISURL, importRDB1
Examples

```r
site_id <- "434400121275801"

data <- readNWISgwl(site_id)
sites <- c("434400121275801", "375907091432201")
data2 <- readNWISgwl(site_id, "", "")
data3 <- readNWISgwl("420125073193001", "", "")

# handling of data where date has no day

data4 <- readNWISgwl("425957088141001", startDate = "1980-01-01")

data5 <- readNWISgwl("263819081585801", parameterCd = "72019")
```

---

**readNWISmeas**

*Surface-water measurement data retrieval from USGS (NWIS)*

**Description**


**Usage**

```r
readNWISmeas(
  siteNumbers,
  startDate = "",
  endDate = "",
  tz = "UTC",
  expanded = FALSE,
  convertType = TRUE
)
```

**Arguments**

- `siteNumbers` character USGS site number (or multiple sites). This is usually an 8 digit number.
- `startDate` character starting date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the earliest possible record.
- `endDate` character ending date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the latest possible record.
- `tz` character to set timezone attribute of dateTime. Default is "UTC", and converts the date times to UTC, properly accounting for daylight savings times based on the data’s provided tz_cd column. Possible values to provide are "America/New_York", "America/Chicago", "America/Denver", "America/Los_Angeles", "America/Phoenix", etc.
"America/Anchorage", as well as the following which do not use daylight savings time: "America/Honolulu", "America/Jamaica", "America/Managua", "America/Phoenix", and "America/Metlakatla". See also OlsonNames() for more information on time zones.

expanded logical. Whether or not (TRUE or FALSE) to call the expanded data.

convertType logical, defaults to TRUE. If TRUE, the function will convert the data to dates, datetimes, numerics based on a standard algorithm. If false, everything is returned as a character

Value

A data frame with at least the following columns:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agency_cd</td>
<td>character</td>
<td>The NWIS code for the agency reporting the data</td>
</tr>
<tr>
<td>site_no</td>
<td>character</td>
<td>The USGS site number</td>
</tr>
<tr>
<td>measurement_dt</td>
<td>POSIXct</td>
<td>The date and time (in POSIXct) of the measurement. Unless specified with the tz parameter, this column is an incomplete, a measurement_dt_date and measurement_dt_time column are added to the returned data frame.</td>
</tr>
<tr>
<td>tz_cd</td>
<td>character</td>
<td>The time zone code for the measurement_dt column</td>
</tr>
</tbody>
</table>


There are also several useful attributes attached to the data frame:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>character</td>
<td>The url used to generate the data</td>
</tr>
<tr>
<td>queryTime</td>
<td>POSIXct</td>
<td>The time the data was returned</td>
</tr>
<tr>
<td>comment</td>
<td>character</td>
<td>Header comments from the RDB file</td>
</tr>
<tr>
<td>siteInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested sites</td>
</tr>
<tr>
<td>tz_cd_reported</td>
<td>character</td>
<td>The originally reported time zone</td>
</tr>
</tbody>
</table>

See Also

constructNWISURL, importRDB1

Examples

```r
site_ids <- c('01594440','040851325')
data <- readNWISmeas(site_ids)
Meas05316840 <- readNWISmeas("05316840")
Meas05316840.ex <- readNWISmeas("05316840",expanded=TRUE)
Meas07227500.ex <- readNWISmeas("07227500",expanded=TRUE)
Meas07227500.exRaw <- readNWISmeas("07227500",expanded=TRUE, convertType = FALSE)
```
Description

Imports data from NWIS about measured parameter based on user-supplied parameter code or codes. This function gets the data from here: https://nwis.waterdata.usgs.gov/nwis/pmcodes

Usage

readNWISpCode(parameterCd)

Arguments

parameterCd  character of USGS parameter codes (or multiple parameter codes). These are 5 digit number codes, more information can be found here: https://help.waterdata.usgs.gov/. To get a complete list of all current parameter codes in the USGS, use "all" as the input.

Value

parameterData data frame with the following information:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>parameter_cd</td>
<td>character</td>
<td>5-digit USGS parameter code</td>
</tr>
<tr>
<td>parameter_group_nm</td>
<td>character</td>
<td>USGS parameter group name</td>
</tr>
<tr>
<td>parameter_nm</td>
<td>character</td>
<td>USGS parameter name</td>
</tr>
<tr>
<td>casrn</td>
<td>character</td>
<td>Chemical Abstracts Service (CAS) Registry Number</td>
</tr>
<tr>
<td>srsname</td>
<td>character</td>
<td>Substance Registry Services Name</td>
</tr>
<tr>
<td>parameter_units</td>
<td>character</td>
<td>Parameter units</td>
</tr>
</tbody>
</table>

See Also

importRDB1

Examples

paramINFO <- readNWISpCode(c('01075','00060','00931'))
paramINFO <- readNWISpCode(c('01075','00060','00931', NA))

all_codes <- readNWISpCode("all")
Description

Reads peak flow from NWISweb. Data is retrieved from https://waterdata.usgs.gov/nwis. In some cases, the specific date of the peak data is not known. This function will default to converting complete dates to a "Date" object, and converting incomplete dates to "NA". If those incomplete dates are needed, set the 'asDateTime' argument to FALSE. No dates will be converted to R Date objects.

Usage

```r
readNWISpeak(
  siteNumbers,
  startDate = "",
  endDate = "",
  asDateTime = TRUE,
  convertType = TRUE
)
```

Arguments

- `siteNumbers` character USGS site number(or multiple sites). This is usually an 8 digit number.
- `startDate` character starting date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the earliest possible record.
- `endDate` character ending date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the latest possible record.
- `asDateTime` logical default to TRUE. When TRUE, the peak_dt column is converted to a Date object, and incomplete dates are removed. When FALSE, no columns are removed, but no dates are converted.
- `convertType` logical, defaults to TRUE. If TRUE, the function will convert the data to dates, datetimes, numerics based on a standard algorithm. If false, everything is returned as a character

Value

A data frame with the following columns:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agency_cd</td>
<td>character</td>
<td>The NWIS code for the agency reporting the data</td>
</tr>
<tr>
<td>site_no</td>
<td>character</td>
<td>The USGS site number</td>
</tr>
<tr>
<td>peak_dt</td>
<td>Date</td>
<td>Date of peak streamflow</td>
</tr>
<tr>
<td>peak_tm</td>
<td>character</td>
<td>Time of peak streamflow as character</td>
</tr>
<tr>
<td>peak_va</td>
<td>numeric</td>
<td>Annual peak streamflow value in cfs</td>
</tr>
<tr>
<td>peak_cd</td>
<td>character</td>
<td>Peak Discharge-Qualification codes (see comment for more information)</td>
</tr>
</tbody>
</table>
readNWISqw

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gage_ht</td>
<td>numeric</td>
<td>Gage height for the associated peak streamflow in feet</td>
</tr>
<tr>
<td>gage_ht_cd</td>
<td>character</td>
<td>Gage height qualification codes</td>
</tr>
<tr>
<td>year_last_pk</td>
<td>numeric</td>
<td>Peak streamflow reported is the highest since this year</td>
</tr>
<tr>
<td>ag_dt</td>
<td>Date</td>
<td>Date of maximum gage-height for water year (if not concurrent with peak)</td>
</tr>
<tr>
<td>ag_tm</td>
<td>character</td>
<td>Time of maximum gage-height for water year (if not concurrent with peak)</td>
</tr>
<tr>
<td>ag_gage_ht</td>
<td>numeric</td>
<td>maximum Gage height for water year in feet (if not concurrent with peak)</td>
</tr>
<tr>
<td>ag_gage_ht_cd</td>
<td>character</td>
<td>maximum Gage height code</td>
</tr>
</tbody>
</table>

There are also several useful attributes attached to the data frame:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>character</td>
<td>The url used to generate the data</td>
</tr>
<tr>
<td>queryTime</td>
<td>POSIXct</td>
<td>The time the data was returned</td>
</tr>
<tr>
<td>comment</td>
<td>character</td>
<td>Header comments from the RDB file</td>
</tr>
<tr>
<td>siteInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested sites</td>
</tr>
</tbody>
</table>

See Also

constructNWISURL, importRDB1

Examples

```r
site_ids <- c('01594440','040851325')

data <- readNWISpeak(site_ids)
data2 <- readNWISpeak(site_ids, asDateTime=FALSE)
stations<-c("06011000")
peakdata<-readNWISpeak(stations,convertType=FALSE)
```

Description

Imports data from NWIS web service. This function gets the data from here: https://nwis.waterdata.usgs.gov/nwis/qwdata A list of parameter codes can be found here: https://nwis.waterdata.usgs.gov/nwis/pmcodes/ A list of statistic codes can be found here: https://nwis.waterdata.usgs.gov/nwis/help/?read_file=stat&format=table
Usage

```r
readNWISqw(
  siteNumbers,
  parameterCd,
  startDate = "",
  endDate = "",
  expanded = TRUE,
  reshape = FALSE,
  tz = "UTC"
)
```

Arguments

- **siteNumbers**: character of USGS site numbers. This is usually an 8 digit number.
- **parameterCd**: character that contains the code for a parameter group, or a character vector of 5-digit parameter codes. See Details.
- **startDate**: character starting date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the earliest possible record. Date arguments are always specified in local time.
- **endDate**: character ending date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the latest possible record. Date arguments are always specified in local time.
- **expanded**: logical defaults to TRUE. If TRUE, retrieves additional information. Expanded data includes remark_cd (remark code), result_va (result value), val_qual_tx (result value qualifier code), meth_cd (method code), dqi_cd (data-quality indicator code), rpt_lev_va (reporting level), and rpt_lev_cd (reporting level type). If FALSE, only returns remark_cd (remark code) and result_va (result value). Expanded = FALSE will not give sufficient information for unbiased statistical analysis.
- **reshape**: logical, reshape the expanded data. If TRUE, then return a wide data frame with all water-quality in a single row for each sample. If FALSE (default), then return a long data frame with each water-quality result in a single row. This argument is only applicable to expanded data. Data requested using expanded=FALSE is always returned in the wide format.
- **tz**: character to set timezone attribute of dateTime. Default is "UTC", and converts the date times to UTC, properly accounting for daylight savings times based on the data’s provided tz_cd column. Possible values to provide are "America/New_York","America/Chicago","America/Denver","America/Los_Angeles", "America/Anchorage", as well as the following which do not use daylight savings time: "America/Honolulu","America/Jamaica","America/Managua","America/Phoenix", and "America/Metlakatla". See also OlsonNames() for more information on time zones.

Details

Valid parameter code groups are "All," or group codes:
If more than one parameter group is requested, only sites that data for all requested groups are returned.

Value

A data frame with at least the following columns:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agency_cd</td>
<td>character</td>
<td>The NWIS code for the agency reporting the data</td>
</tr>
<tr>
<td>site_no</td>
<td>character</td>
<td>The USGS site number</td>
</tr>
<tr>
<td>sample_dt</td>
<td>Date</td>
<td>The date the sample was collected</td>
</tr>
<tr>
<td>sample_tm</td>
<td>character</td>
<td>The reported sample collection time</td>
</tr>
<tr>
<td>startDateT</td>
<td>POSIXct</td>
<td>Combining sample_dt and sample_tm, a date/time column is created, and converted into UTC (unless the tz argument specifies a different time zone)</td>
</tr>
<tr>
<td>endDateTime</td>
<td>POSIXct</td>
<td>If any sample_end_dt and sample_end_tm exist, this column is created similar to startDateTime</td>
</tr>
</tbody>
</table>

Further columns will be included depending on the requested output format (expanded = TRUE or FALSE). Columns that end in "_reported" are the originally reported timezones, but the "tz_cd" column defines the timezone of any POSIXct columns.

There are also several useful attributes attached to the data frame:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>character</td>
<td>The url used to generate the data</td>
</tr>
<tr>
<td>queryTime</td>
<td>POSIXct</td>
<td>The time the data was returned</td>
</tr>
<tr>
<td>comment</td>
<td>character</td>
<td>Header comments from the RDB file</td>
</tr>
<tr>
<td>siteInfo</td>
<td>data frame</td>
<td>A data frame containing information on the requested sites</td>
</tr>
</tbody>
</table>
variableInfo  data frame  A data frame containing information on the requested parameters

See Also

readWQPdata, whatWQPsites, readWQPqw, constructNWISURL

---

readNWISrating  Rating table for an active USGS streamgage retrieval

Description

Reads current rating table for an active USGS streamgage from NWISweb. Data is retrieved from https://waterdata.usgs.gov/nwis.

Usage

readNWISrating(siteNumber, type = "base", convertType = TRUE)

Arguments

siteNumber  character  USGS site number. This is usually an 8 digit number

type  character  can be "base", "corr", or "exsa"

convertType  logical, defaults to TRUE. If TRUE, the function will convert the data to dates, datetimes, numerics based on a standard algorithm. If false, everything is returned as a character

Value

A data frame. If type is "base," then the columns are INDEP, typically the gage height, in feet; DEP, typically the streamflow, in cubic feet per second; and STOR, where "*" indicates that the pair are a fixed point of the rating curve. If type is "exsa," then an additional column, SHIFT, is included that indicates the current shift in the rating for that value of INDEP. If type is "corr," then the columns are INDEP, typically the gage height, in feet; CORR, the correction for that value; and CORRINDEP, the corrected value for CORR.

If type is "base," then the data frame has an attribute called "RATING" that describes the rating curve is included.

There are also several useful attributes attached to the data frame:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>character</td>
<td>The url used to generate the data</td>
</tr>
<tr>
<td>queryTime</td>
<td>POSIXct</td>
<td>The time the data was returned</td>
</tr>
<tr>
<td>comment</td>
<td>character</td>
<td>Header comments from the RDB file</td>
</tr>
<tr>
<td>siteInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested sites</td>
</tr>
<tr>
<td>RATING</td>
<td>character</td>
<td>Rating information</td>
</tr>
</tbody>
</table>
**Note**

Not all active USGS streamgages have traditional rating curves that relate flow to stage.

**See Also**

constructNWISURL, importRDB1

**Examples**

```r
site_id <- '01594440'

data <- readNWISrating(site_id, "base")
attr(data, "RATING")
```

---

**readNWISsite**  
**USGS Site File Data Retrieval**

**Description**

Imports data from USGS site file site. This function gets data from here: https://waterservices.usgs.gov/ usgs.gov/

**Usage**

`readNWISsite(siteNumbers)`

**Arguments**

- `siteNumbers` character USGS site number (or multiple sites). This is usually an 8 digit number

**Value**

A data frame with at least the following columns:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agency_cd</td>
<td>character</td>
<td>The NWIS code for the agency reporting the data</td>
</tr>
<tr>
<td>site_no</td>
<td>character</td>
<td>The USGS site number</td>
</tr>
<tr>
<td>station_nm</td>
<td>character</td>
<td>Site name</td>
</tr>
<tr>
<td>site_tp_cd</td>
<td>character</td>
<td>Site type</td>
</tr>
<tr>
<td>lat_va</td>
<td>numeric</td>
<td>DMS latitude</td>
</tr>
<tr>
<td>long_va</td>
<td>numeric</td>
<td>DMS longitude</td>
</tr>
<tr>
<td>dec_lat_va</td>
<td>numeric</td>
<td>Decimal latitude</td>
</tr>
<tr>
<td>dec_long_va</td>
<td>numeric</td>
<td>Decimal longitude</td>
</tr>
<tr>
<td>coord_meth_cd</td>
<td>character</td>
<td>Latitude-longitude method</td>
</tr>
<tr>
<td>coord_acy_cd</td>
<td>character</td>
<td>Latitude-longitude accuracy</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>coord_datum_cd</td>
<td>character</td>
<td>Latitude-longitude datum</td>
</tr>
<tr>
<td>dec_coord_datum_cd</td>
<td>character</td>
<td>Decimal Latitude-longitude datum</td>
</tr>
<tr>
<td>district_cd</td>
<td>character</td>
<td>District code</td>
</tr>
<tr>
<td>state_cd</td>
<td>character</td>
<td>State code</td>
</tr>
<tr>
<td>county_cd</td>
<td>character</td>
<td>County code</td>
</tr>
<tr>
<td>country_cd</td>
<td>character</td>
<td>Country code</td>
</tr>
<tr>
<td>land_net_ds</td>
<td>character</td>
<td>Land net location description</td>
</tr>
<tr>
<td>map_nm</td>
<td>character</td>
<td>Name of location map</td>
</tr>
<tr>
<td>map_scale_fc</td>
<td>character</td>
<td>Scale of location map</td>
</tr>
<tr>
<td>alt_va</td>
<td>numeric</td>
<td>Altitude of Gage/land surface</td>
</tr>
<tr>
<td>alt_meth_cd</td>
<td>character</td>
<td>Method altitude determined</td>
</tr>
<tr>
<td>alt_acy_va</td>
<td>numeric</td>
<td>Altitude accuracy</td>
</tr>
<tr>
<td>alt_datum_cd</td>
<td>character</td>
<td>Altitude datum</td>
</tr>
<tr>
<td>huc_cd</td>
<td>character</td>
<td>Hydrologic unit code</td>
</tr>
<tr>
<td>basin_cd</td>
<td>character</td>
<td>Drainage basin code</td>
</tr>
<tr>
<td>topo_cd</td>
<td>character</td>
<td>Topographic setting code</td>
</tr>
<tr>
<td>instruments_cd</td>
<td>character</td>
<td>Flags for instruments at site</td>
</tr>
<tr>
<td>construction_dt</td>
<td>character</td>
<td>Date of first construction</td>
</tr>
<tr>
<td>inventory_dt</td>
<td>character</td>
<td>Date site established or inventoried</td>
</tr>
<tr>
<td>drain_area_va</td>
<td>numeric</td>
<td>Drainage area</td>
</tr>
<tr>
<td>contrib_drain_area_va</td>
<td>numeric</td>
<td>Contributing drainage area</td>
</tr>
<tr>
<td>tz_cd</td>
<td>character</td>
<td>Time Zone abbreviation</td>
</tr>
<tr>
<td>local_time_fg</td>
<td>character</td>
<td>Site honors Daylight Savings Time</td>
</tr>
<tr>
<td>reliability_cd</td>
<td>character</td>
<td>Data reliability code</td>
</tr>
<tr>
<td>gw_file_cd</td>
<td>character</td>
<td>Data-other GW files</td>
</tr>
<tr>
<td>nat_aqfr_cd</td>
<td>character</td>
<td>National aquifer code</td>
</tr>
<tr>
<td>aqfr_cd</td>
<td>character</td>
<td>Local aquifer code</td>
</tr>
<tr>
<td>aqfr_type_cd</td>
<td>character</td>
<td>Local aquifer type code</td>
</tr>
<tr>
<td>well_depth_va</td>
<td>numeric</td>
<td>Well depth</td>
</tr>
<tr>
<td>hole_depth_va</td>
<td>numeric</td>
<td>Hole depth</td>
</tr>
<tr>
<td>depth_src_cd</td>
<td>character</td>
<td>Source of depth data</td>
</tr>
<tr>
<td>project_no</td>
<td>character</td>
<td>Project number</td>
</tr>
</tbody>
</table>

There are also several useful attributes attached to the data frame:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>character</td>
<td>The url used to generate the data</td>
</tr>
<tr>
<td>queryTime</td>
<td>POSIXct</td>
<td>The time the data was returned</td>
</tr>
<tr>
<td>comment</td>
<td>character</td>
<td>Header comments from the RDB file</td>
</tr>
</tbody>
</table>

**Examples**

```r
siteINFO <- readNWISsite('05114000')
siteINFOMulti <- readNWISsite(c('05114000','09423350'))
```
**readNWISstat**

*Site statistics retrieval from USGS (NWIS)*

**Description**


**Usage**

```r
readNWISstat(
  siteNumbers, parameterCd, startDate = "", endDate = "",
  convertType = TRUE, statReportType = "daily", statType = "mean"
)
```

**Arguments**

- `siteNumbers`: character USGS site number (or multiple sites). This is usually an 8 digit number.
- `parameterCd`: character USGS parameter code. This is usually a 5 digit number.
- `startDate`: character starting date for data retrieval in the form YYYY, YYYY-MM, or YYYY-MM-DD. Dates cannot be more specific than the statReportType, i.e. startDate for monthly statReportTypes cannot include days, and annual statReportTypes cannot include days or months. Months and days are optional for the daily statReportType. Default is "" which indicates retrieval for the earliest possible record. For daily data, this indicates the start of the period the statistics will be computed over.
- `endDate`: character ending date for data retrieval in the form YYYY, YYYY-MM, or YYYY-MM-DD. Default is "" which indicates retrieval for the latest possible record. For daily data, this indicates the end of the period the statistics will be computed over. The same restrictions as startDate apply.
- `convertType`: logical, defaults to TRUE. If TRUE, the function will convert the data to numerics based on a standard algorithm. Years, months, and days (if applicable) are also returned as numerics in separate columns. If convertType is false, everything is returned as a character.
readNWISstat

statReportType character time division for statistics: daily, monthly, or annual. Default is daily. Note that daily provides statistics for each calendar day over the specified range of water years, i.e. no more than 366 data points will be returned for each site/parameter. Use readNWISdata or readNWISdv for daily averages. Also note that 'annual' returns statistics for the calendar year. Use readNWISdata for water years. Monthly and yearly provide statistics for each month and year within the range individually.

statType character type(s) of statistics to output for daily values. Default is mean, which is the only option for monthly and yearly report types. See the statistics service documentation at https://waterservices.usgs.gov/rest/Statistics-Service.html for a full list of codes.

Value

A data frame with the following columns:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agency_cd</td>
<td>character</td>
<td>The NWIS code for the agency reporting the data</td>
</tr>
<tr>
<td>site_no</td>
<td>character</td>
<td>The USGS site number</td>
</tr>
<tr>
<td>parameter_cd</td>
<td>character</td>
<td>The USGS parameter code</td>
</tr>
</tbody>
</table>

Other columns will be present depending on statReportType and statType

See Also

constructNWISURL, importRDB1

Examples

```r
x1 <- readNWISstat(siteNumbers=c("02319394"),
                     parameterCd=c("00060"),
                     statReportType="annual")

#all the annual mean discharge data for two sites
x2 <- readNWISstat(siteNumbers=c("02319394","02171500"),
                     parameterCd=c("00010","00060"),
                     statReportType="annual")

#Request p25, p75, and mean values for temperature and discharge for the 2000s
#Note that p25 and p75 were not available for temperature, and return NAs
x <- readNWISstat(siteNumbers=c("02171500"),
                    parameterCd=c("00010","00060"),
                    statReportType="daily",
                    statType=c("mean","median"),
                    startDate="2000",endDate="2010")
```
**readNWISuse**  
*Water use data retrieval from USGS (NWIS)*

**Description**
Retrieves water use data from USGS Water Use Data for the Nation. See [https://waterdata.usgs.gov/nwis/wu](https://waterdata.usgs.gov/nwis/wu) for more information. All available use categories for the supplied arguments are retrieved.

**Usage**
```r
readNWISuse(
  stateCd,  
  countyCd,  
  years = "ALL",  
  categories = "ALL",  
  convertType = TRUE,  
  transform = FALSE
)
```

**Arguments**
- **stateCd**
  could be character (full name, abbreviation, id), or numeric (id). Only one is accepted per query.
- **countyCd**
  could be character (name, with or without "County", or "ALL"), numeric (id), or codeNULL, which will return state or national data depending on the stateCd argument. ALL may also be supplied, which will return data for every county in a state. Can be a vector of counties in the same state.
- **years**
  integer Years for data retrieval. Must be years ending in 0 or 5. Default is all available years.
- **categories**
  character categories of water use. Defaults to ALL. Specific categories must be supplied as two-letter abbreviations as seen in the URL when using the NWIS water use web interface. Note that there are different codes for national and state level data.
- **convertType**
  logical defaults to TRUE. If TRUE, the function will convert the data to numerics based on a standard algorithm. Years, months, and days (if applicable) are also returned as numerics in separate columns. If convertType is false, everything is returned as a character.
- **transform**
  logical only intended for use with national data. Defaults to FALSE, with data being returned as presented by the web service. If TRUE, data will be transformed and returned with column names, which will reformat national data to be similar to state data.

**Value**
A data frame with at least the year of record, and all available statistics for the given geographic parameters. County and state fields will be included as appropriate.
Examples

```
#All data for a county
allegheny <- readNWISuse(stateCd = "Pennsylvania", countyCd = "Allegheny")

#Data for an entire state for certain years
ohio <- readNWISuse(years=c(2000,2005,2010),stateCd = "OH", countyCd = NULL)

#Data for an entire state, county by county
pr <- readNWISuse(years=c(2000,2005,2010),stateCd = "PR", countyCd="ALL")

#All national-scale data, transforming data frame to named columns from named rows
national <- readNWISuse(stateCd = NULL, countyCd = NULL, transform = TRUE)

#Washington, DC data
dc <- readNWISuse(stateCd = "DC", countyCd = NULL)

#data for multiple counties, with different input formatting
paData <- readNWISuse(stateCd = "42", countyCd = c("Allegheny County", "BUTLER", 1, "031"))

#retrieving two specific categories for an entire state
ks <- readNWISuse(stateCd = "KS", countyCd = NULL, categories = c("IT","LI"))
```

---

**readNWISuv**

*Instantaneous value data retrieval from USGS (NWIS)*

Description

Imports data from NWIS web service. This function gets the data from here: https://waterservices.usgs.gov/ A list of parameter codes can be found here: https://nwis.waterdata.usgs.gov/nwis/pmcodes/ A list of statistic codes can be found here: https://nwis.waterdata.usgs.gov/nwis/help/?read_file=stat&format=table. More information on the web service can be found here: https://waterservices.usgs.gov/rest/IV-Service.html.

Usage

```
readNWISuv(siteNumbers, parameterCd, startDate = "", endDate = "", tz = "UTC")
```

Arguments

- `siteNumbers` character USGS site number (or multiple sites). This is usually an 8 digit number.
- `parameterCd` character USGS parameter code. This is usually an 5 digit number.
- `startDate` character starting date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the earliest possible record. Simple date arguments are specified in local time. See more information here: https://waterservices.usgs.gov/rest/IV-Service.html.
endDate character ending date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the latest possible record. Simple date arguments are specified in local time. See more information here: https://waterservices.usgs.gov/rest/IV-Service.html.

tz character to set timezone attribute of dateTime. Default is "UTC", and converts the date times to UTC, properly accounting for daylight savings times based on the data’s provided tz_cd column. Possible values to provide are "America/New_York", "America/Chicago", "America/Denver", "America/Los_Angeles", "America/Anchorage", as well as the following which do not use daylight savings time: "America/Honolulu", "America/Jamaica", "America/Managua", "America/Phoenix", and "America/Metlakatla". See also OlsonNames() for more information on time zones.

**Value**

A data frame with the following columns:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agency_cd</td>
<td>character</td>
<td>The NWIS code for the agency reporting the data</td>
</tr>
<tr>
<td>site_no</td>
<td>character</td>
<td>The USGS site number</td>
</tr>
<tr>
<td>dateTime</td>
<td>POSIXxct</td>
<td>The date and time of the value converted to UTC</td>
</tr>
<tr>
<td>tz_cd</td>
<td>character</td>
<td>The time zone code for dateTime</td>
</tr>
<tr>
<td>code</td>
<td>character</td>
<td>Any codes that qualify the corresponding value</td>
</tr>
<tr>
<td>value</td>
<td>numeric</td>
<td>The numeric value for the parameter</td>
</tr>
</tbody>
</table>

Note that code and value are repeated for the parameters requested. The names are of the form: X_D_P_S, where X is literal, D is an option description of the parameter, P is the parameter code, and S is the statistic code (if applicable).

There are also several useful attributes attached to the data frame:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>character</td>
<td>The url used to generate the data</td>
</tr>
<tr>
<td>siteInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested sites</td>
</tr>
<tr>
<td>variableInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested parameters</td>
</tr>
<tr>
<td>statisticInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested statistics on the data</td>
</tr>
<tr>
<td>queryTime</td>
<td>POSIXxct</td>
<td>The time the data was returned</td>
</tr>
</tbody>
</table>

**See Also**

renameNWISColumns, importWaterML1

**Examples**

```r
site_id <- '05114000'
parameterCd <- '00060'
```
```r
startDate <- "2014-10-10"
endDate <- "2014-10-10"

rawData <- readNWISuv(site_id,parameterCd,startDate,endDate)
rawData_today <- readNWISuv(site_id, parameterCd, Sys.Date(),Sys.Date())
timeZoneChange <- readNWISuv(c("04024430","04024000"),parameterCd, 
   "2013-11-03","2013-11-03")

centralTime <- readNWISuv(site_id,parameterCd,
   "2014-10-10T12:00", "2014-10-10T23:59",
   tz="America/Chicago")

# Adding 'Z' to the time indicates to the web service to call the data with UTC time:
GMTdata <- readNWISuv(site_id,parameterCd, 
   "2014-10-10T00:00Z", "2014-10-10T23:59Z")
```

**readWQPdata**

*General Data Import from Water Quality Portal*

**Description**

Imports data from Water Quality Portal web service. This function gets the data from here: [https://www.waterqualitydata.us](https://www.waterqualitydata.us) because it allows for other agencies rather than the USGS.

**Usage**

```r
readWQPdata(..., querySummary = FALSE, tz = "UTC", ignore_attributes = FALSE)
```

**Arguments**

- `...`: see [https://www.waterqualitydata.us/webservices_documentation](https://www.waterqualitydata.us/webservices_documentation) for a complete list of options. A list of arguments can also be supplied.
- `querySummary`: logical to ONLY return the number of records and unique sites that will be returned from this query. This argument is not supported via the combined list from the ...argument
- `tz`: character to set timezone attribute of dateTime. Default is "UTC", and converts the date times to UTC, properly accounting for daylight savings times based on the data’s provided tz_cd column. Possible values to provide are "America/New_York","America/Chicago","America/Denver","America/Los_Angeles", "America/Anchorage", as well as the following which do not use daylight savings time: "America/Honolulu","America/Jamaica","America/Managua","America/Phoenix", and "America/Metlakatla". See also OlsonNames() for more information on time zones.
ignore_attributes

logical to choose to ignore fetching site and parameter attributes. Default is FALSE.

Value

A data frame with at least the following columns:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OrganizationIdentifier</td>
<td>character</td>
<td>A designator used to uniquely identify a unique business establishment.</td>
</tr>
<tr>
<td>OrganizationFormalName</td>
<td>character</td>
<td>The legal designator (i.e. formal name) of an organization.</td>
</tr>
<tr>
<td>ActivityIdentifier</td>
<td>character</td>
<td>Designator that uniquely identifies an activity within an organization.</td>
</tr>
<tr>
<td>ActivityTypeCode</td>
<td>character</td>
<td>The text describing the type of activity.</td>
</tr>
<tr>
<td>ActivityMediaName</td>
<td>character</td>
<td>Name or code indicating the environmental medium which the sample is collected.</td>
</tr>
<tr>
<td>ActivityStartDate</td>
<td>character</td>
<td>The calendar date on which the field activity is scheduled to begin.</td>
</tr>
<tr>
<td>ActivityEndTime.Time</td>
<td>character</td>
<td>The calendar date on which the activity is completed.</td>
</tr>
<tr>
<td>ActivityStartTime.TimeZoneCode</td>
<td>character</td>
<td>The time zone for which the time of day is reported.</td>
</tr>
<tr>
<td>ActivityStartTime.Time</td>
<td>character</td>
<td>The time of day that is reported when the field activity is scheduled to begin.</td>
</tr>
<tr>
<td>ActivityEndTime.TimeZoneCode</td>
<td>character</td>
<td>The time zone for which the time of day is reported.</td>
</tr>
<tr>
<td>ActivityEndTime.Time</td>
<td>character</td>
<td>The time of day that is reported when the field activity is completed.</td>
</tr>
<tr>
<td>ActivityDepthHeightMeasure.MeasureValue</td>
<td>character</td>
<td>A measurement of the vertical location (measured from the source reference)</td>
</tr>
<tr>
<td>ActivityDepthHeightMeasure.MeasureUnitCode</td>
<td>character</td>
<td>The code that represents the unit for measuring the item.</td>
</tr>
<tr>
<td>ActivityTopDepthHeightMeasure.MeasureValue</td>
<td>character</td>
<td>The code that represents the unit for measuring the upper vertical location.</td>
</tr>
<tr>
<td>ActivityBottomDepthHeightMeasure.MeasureValue</td>
<td>character</td>
<td>The code that represents the unit for measuring the lower vertical location.</td>
</tr>
<tr>
<td>ActivityTopDepthHeightMeasure.MeasureUnitCode</td>
<td>character</td>
<td>A measurement of the upper vertical location of a vertical location.</td>
</tr>
<tr>
<td>ActivityBottomDepthHeightMeasure.MeasureUnitCode</td>
<td>character</td>
<td>A measurement of the lower vertical location of a vertical location.</td>
</tr>
<tr>
<td>ProjectIdentifier</td>
<td>character</td>
<td>A designator uniquely identifies a project within an organization.</td>
</tr>
<tr>
<td>ActivityConductingOrganizationText</td>
<td>character</td>
<td>A name of the organization conducting the activity.</td>
</tr>
<tr>
<td>MonitoringLocationIdentifier</td>
<td>character</td>
<td>A designator used to uniquely identify a monitoring location.</td>
</tr>
<tr>
<td>ActivityCommentText</td>
<td>character</td>
<td>General comments concerning the activity.</td>
</tr>
<tr>
<td>SampleAquifer</td>
<td>character</td>
<td>A code that designates the aquifer associated with groundwater samples.</td>
</tr>
<tr>
<td>HydrologicCondition</td>
<td>character</td>
<td>Hydrologic condition is the hydrologic condition that is represented by the sample.</td>
</tr>
<tr>
<td>HydrologicEvent</td>
<td>character</td>
<td>A hydrologic event that is represented by the sample.</td>
</tr>
<tr>
<td>SampleCollectionMethod.MethodIdentifier</td>
<td>character</td>
<td>The identification number or code assigned by the method publisher.</td>
</tr>
<tr>
<td>SampleCollectionMethod.MethodIdentifierContext</td>
<td>character</td>
<td>Identifies the source or data system that created or defines the method.</td>
</tr>
<tr>
<td>SampleCollectionMethod.MethodName</td>
<td>character</td>
<td>The title that appears on the method from the method publisher.</td>
</tr>
<tr>
<td>SampleCollectionEquipmentName</td>
<td>character</td>
<td>The name for the equipment used in collecting the sample.</td>
</tr>
<tr>
<td>ResultDepthHeightMeasure.MeasureValue</td>
<td>character</td>
<td>A measurement of the vertical location (measured from the source reference)</td>
</tr>
<tr>
<td>ResultDepthHeightMeasure.MeasureUnitCode</td>
<td>character</td>
<td>The code that represents the unit for measuring the item.</td>
</tr>
<tr>
<td>ResultTopDepthHeightMeasure.MeasureValue</td>
<td>character</td>
<td>The code that represents the unit for measuring the upper vertical location.</td>
</tr>
<tr>
<td>ResultTopDepthHeightMeasure.MeasureUnitCode</td>
<td>character</td>
<td>A measurement of the upper vertical location of a vertical location.</td>
</tr>
<tr>
<td>ResultBottomDepthHeightMeasure.MeasureUnitCode</td>
<td>character</td>
<td>A measurement of the lower vertical location of a vertical location.</td>
</tr>
<tr>
<td>CharacteristicName</td>
<td>character</td>
<td>The object, property, or substance which is evaluated or measured.</td>
</tr>
<tr>
<td>ResultSampleFractionText</td>
<td>character</td>
<td>The text name of the portion of the sample associated with results obtained.</td>
</tr>
<tr>
<td>ResultMeasureValue</td>
<td>numeric</td>
<td>The reportable measure of the result for the chemical, microbiological, or other characteristic being analyzed.</td>
</tr>
<tr>
<td>MeasureQualifierCode</td>
<td>character</td>
<td>A code used to identify any qualifying issues that affect the measurement.</td>
</tr>
<tr>
<td>ResultMeasure.MeasureUnitCode</td>
<td>character</td>
<td>The code that represents the unit for measuring the item.</td>
</tr>
<tr>
<td>ResultStatusIdentifier</td>
<td>character</td>
<td>Indicates the acceptability of the result with respect to QA/QC criteria.</td>
</tr>
<tr>
<td>StatisticalBaseCode</td>
<td>character</td>
<td>The code for the method used to calculate derived results.</td>
</tr>
<tr>
<td>ResultValueTypeName</td>
<td>character</td>
<td>A name that qualifies the process which was used in the test.</td>
</tr>
<tr>
<td>ResultWeightBasisText</td>
<td>character</td>
<td>The name that represents the form of the sample or portion.</td>
</tr>
</tbody>
</table>
ResultTimeBasisText character The period of time (in days) over which a measurement was made. For example, BOD can be measured as 5 day or 20 day BOD.
ResultTemperatureBasisText character The name that represents the controlled temperature at which the sample was maintained during analysis, e.g. 25 deg BOD analysis.
ResultParticleSizeBasisText character User defined free text describing the particle size class for which the associated result is defined.
PrecisionValue character A measure of mutual agreement among individual measurements of the same property usually under prescribed similar conditions.
ResultCommentText character Free text with general comments concerning the result.
USGSPCode * character 5-digit number used in the US Geological Survey computerized data system, National Water Information System (NWIS), to uniquely identify a specific constituent.
ResultDepthHeightMeasure.MeasureValue + character A measurement of the vertical location (measured from a reference point) at which a result occurred.
ResultDepthHeightMeasure.MeasureUnitCode + character The code that represents the unit for measuring the item.
ResultDepthAltitudeReferencePointText + character The reference used to indicate the datum or reference used to establish the depth/altitude of a result.
SubjectTaxonomicName character The name of the organism from which a tissue sample was taken.
SampleTissueAnatomyName * character The name of the anatomy from which a tissue sample was taken.
ResultAnalyticalMethod.MethodIdentifier character The identification number or code assigned by the method publisher.
ResultAnalyticalMethod.MethodIdentifierContext character Identifies the source or data system that created or defined the identifier.
ResultAnalyticalMethod.MethodName character The title that appears on the method from the method publisher.
MethodDescriptionText * character A brief summary that provides general information about the method.
LaboratoryName character The name of Lab responsible for the result.
AnalysisStartDate character The calendar date on which the analysis began.
ResultLaboratoryCommentText character Remarks which further describe the laboratory procedures.
DetectionQuantitationLimitTypeName character Text describing the type of detection or quantitation level used in the analysis of a characteristic.
DetectionQuantitationLimitMeasure.MeasureValue numeric Constituent concentration that, when processed through the specified method, is different from a blank. Measure value is given in the units stored in DetectionQuantitationLimitMeasure.MeasureUnitCode.
DetectionQuantitationLimitMeasure.MeasureUnitCode character The code that represents the unit for measuring the item.
PreparationStartDate character The calendar date when the preparation/extraction of the sample for analysis began.
ActivityStartDateTime POSIXct Activity start date and time converted to POSIXct UTC.
ActivityEndDateTime POSIXct Activity end date and time converted to POSIXct UTC.

* = elements only in NWIS + = elements only in STORET

There are also several useful attributes attached to the data frame:

Name Type Description
url character The url used to generate the data
siteInfo data.frame A data frame containing information on the requested sites
variableInfo data.frame A data frame containing information on the requested parameters
queryTime POSIXct The time the data was returned

Examples

nameToUse <- "pH"
pHData <- readWQPdata(siteid="USGS-04024315", characteristicName=nameToUse)
pHData_summary <- readWQPdata(bBox=c(-90.10,42.67,-88.64,43.35),
characteristicName=nameToUse, querySummary=TRUE)
startDate <- as.Date("2013-01-01")
secchi.names = c("Depth, Secchi disk depth",
"Depth, Secchi disk depth (choice list)",
"Secchi Reading Condition (choice list)",
"Water transparency, Secchi disc")
args <- list('startDateLo' = startDate,
              'startDateHi' = "2013-12-31",
              statecode = "WI",
              characteristicName = secchi.names)

wqp.data <- readWQPdata(args)

args_2 <- list('startDateLo' = startDate,
                'startDateHi' = "2013-12-31",
                statecode = "WI",
                characteristicName = secchi.names,
                querySummary = TRUE)

wqp.summary <- readWQPdata(args_2)

arg_3 <- list('startDateLo' = startDate,
              'startDateHi' = "2013-12-31")
arg_4 <- list(statecode = "WI",
              characteristicName = secchi.names)
wqp.summary <- readWQPdata(arg_3, arg_4, querySummary = TRUE)
wqp.summary_WI <- readWQPdata(arg_3, statecode = "WI",
                               characteristicName = secchi.names,
                               querySummary = TRUE)

# querying by county
DeWitt <- readWQPdata(statecode = "Illinois",
                      countycode = "DeWitt",
                      characteristicName = "Nitrogen")

# Data profiles: "Organization Data"
org_data <- readWQPdata(statecode = "WI",
                         countycode = "Dane",
                         service = "Organization")

# Data profiles: "Site Data Only"
site_data <- readWQPdata(statecode = "WI",
                         countycode = "Dane",
                         service = "Station")

# Data profiles: "Project Data"
project_data <- readWQPdata(statecode = "WI",
                            countycode = "Dane",
                            service = "Project")

# Data profiles: "Project Monitoring Location Weighting Data"
proj_mlwd <- readWQPdata(statecode = "WI",
                         countycode = "Dane",
                         service = "ProjectMonitoringLocationWeighting")

# Data profiles: "Sample Results (physical/chemical metadata)"
samp_data <- readWQPdata(siteid = "USGS-04024315",
                          dataProfile = "resultPhysChem")
# Data profiles: "Sample Results (biological metadata)"
samp_bio <- readWQPdata(siteid = "USGS-04024315",
                         dataProfile = "biological")

# Data profiles: "Sample Results (narrow)"
samp_narrow <- readWQPdata(siteid = "USGS-04024315",
                         dataProfile = "narrowResult")

# Data profiles: "Sampling Activity"
samp_activity <- readWQPdata(siteid = "USGS-04024315",
                           dataProfile = "activityAll")

# Data profile: "Sampling Activity Metrics"
act_metrics <- readWQPdata(statecode = "WI",
                           countycode = "Dane",
                           service = "ActivityMetric")

# Data profile: "Result Detection Quantitation Limit Data"
dl_data <- readWQPdata(siteid = "USGS-04024315",
                        service = "ResultDetectionQuantitationLimit")

---

**readWQPqw**

**Raw Data Import for Water Quality Portal**

**Description**

Imports data from the Water Quality Portal. This function gets the data from here: [https://www.waterqualitydata.us](https://www.waterqualitydata.us). There are four required input arguments: siteNumbers, parameterCd, startDate, and endDate. parameterCd can either be a USGS 5-digit code, or a characteristic name. The sites can be either USGS, or other Water Quality Portal offered sites. It is required to use the 'full' site name, such as 'USGS-01234567'.

**Usage**

```r
readWQPqw(
  siteNumbers,
  parameterCd,
  startDate = "",
  endDate = "",
  tz = "UTC",
  querySummary = FALSE
)
```

**Arguments**

- `siteNumbers` character site number. This needs to include the full agency code prefix.
readWQPqw

**Value**

A data frame with at least the following columns:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OrganizationIdentifier</td>
<td>character</td>
<td>A designator used to uniquely identify a unique business establishment within an organization.</td>
</tr>
<tr>
<td>OrganizationFormalName</td>
<td>character</td>
<td>The legal designator (i.e. formal name) of an organization.</td>
</tr>
<tr>
<td>ActivityIdentifier</td>
<td>character</td>
<td>Designator that uniquely identifies an activity within an organization.</td>
</tr>
<tr>
<td>ActivityTypeCode</td>
<td>character</td>
<td>The text describing the type of activity.</td>
</tr>
<tr>
<td>ActivityMediaName</td>
<td>character</td>
<td>Name or code indicating the environmental medium where the sample was taken.</td>
</tr>
<tr>
<td>ActivityMediaSubdivisionName</td>
<td>character</td>
<td>Name or code indicating the environmental media as a subdivision of the sample medium.</td>
</tr>
<tr>
<td>ActivityStartDate</td>
<td>character</td>
<td>The calendar date on which the field activity is started.</td>
</tr>
<tr>
<td>ActivityStartTime.Time</td>
<td>character</td>
<td>The time of day that the field activity began, based on a 24-hour timescale.</td>
</tr>
<tr>
<td>ActivityEndTime.TimeTimeZoneCode</td>
<td>character</td>
<td>The time zone for which the time of day is reported.</td>
</tr>
<tr>
<td>ActivityEndDate</td>
<td>character</td>
<td>The calendar date when the field activity is completed.</td>
</tr>
<tr>
<td>ActivityEndTime.Time</td>
<td>character</td>
<td>The time of day that the field activity ended.</td>
</tr>
<tr>
<td>ActivityEndTime.TimeTimeZoneCode</td>
<td>character</td>
<td>The time zone for which the time of day is reported.</td>
</tr>
<tr>
<td>ActivityDepthHeightMeasure.MeasureValue</td>
<td>character</td>
<td>A measurement of the vertical location (measured from a reference point).</td>
</tr>
<tr>
<td>ActivityDepthHeightMeasure.MeasureUnitCode</td>
<td>character</td>
<td>The code that represents the unit for measuring the item.</td>
</tr>
<tr>
<td>ActivityDepthAltitudeReferencePointText</td>
<td>character</td>
<td>The reference used to indicate the datum or reference used to establish depth or altitude.</td>
</tr>
<tr>
<td>ActivityTopDepthHeightMeasure.MeasureValue</td>
<td>character</td>
<td>A measurement of the upper vertical location of a vertical location.</td>
</tr>
<tr>
<td>ActivityTopDepthHeightMeasure.MeasureUnitCode</td>
<td>character</td>
<td>The code that represents the unit for measuring the item.</td>
</tr>
<tr>
<td>ActivityBottomDepthHeightMeasure.MeasureValue</td>
<td>character</td>
<td>A measurement of the lower vertical location of a vertical location.</td>
</tr>
<tr>
<td>ActivityBottomDepthHeightMeasure.MeasureUnitCode</td>
<td>character</td>
<td>The code that represents the unit for measuring the item.</td>
</tr>
<tr>
<td>ProjectIdentifier</td>
<td>character</td>
<td>A designator used to uniquely identify a data collection project.</td>
</tr>
<tr>
<td>ActivityConductingOrganizationText</td>
<td>character</td>
<td>A name of the Organization conducting an activity.</td>
</tr>
<tr>
<td>MonitoringLocationIdentifier</td>
<td>character</td>
<td>A designator used to describe the unique name, number, or code assigned to identify the monitoring location.</td>
</tr>
<tr>
<td>ActivityCommentText</td>
<td>character</td>
<td>General comments concerning the activity.</td>
</tr>
<tr>
<td>SampleAquifer</td>
<td>character</td>
<td>A code that designates the aquifer associated with groundwater sampling.</td>
</tr>
</tbody>
</table>
HydrologicCondition * character Hydrologic condition is the hydrologic condition that is represented by the sample collected (i.e. normal, falling, rising, peak stage).
HydrologicEvent * character A hydrologic event that is represented by the sample collected (i.e. storm, drought, snowmelt).
SampleCollectionMethod.MethodIdentifier character The identification number or code assigned by the method publisher.
SampleCollectionMethod.MethodIdentifierContext character Identifies the source or data system that created or defined the identifier.
SampleCollectionMethod.MethodName character The title that appears on the method from the method publisher.
SampleCollectionEquipmentName character The name for the equipment used in collecting the sample.
ResultDetectionConditionText character The textual descriptor of a result.
CharacteristicName character The object, property, or substance which is evaluated or enumerated by either a direct field measurement, a direct field observation, or by laboratory analysis of material collected in the field.
ResultSampleFractionText character The text name of the portion of the sample associated with results obtained from a physically-partitioned sample.
MeasureQualifierCode character A code used to identify any qualifying issues that affect the result.
ResultMeasureValue numeric The reportable measure of the result for the chemical, microbiological or other characteristic being analyzed. Measure value is given in the units stored in ResultMeasure/MeasureUnitCode.
ResultMeasure.MeasureUnitCode character The code that represents the unit for measuring the item.
ResultStatusIdentifier character Indicates the acceptability of the result with respect to QA/QC criteria.
StatisticalBaseCode character The code for the method used to calculate derived results.
ResultValueTypeName character A name that qualifies the process which was used in the determination of the result value (e.g., actual, estimated, calculated).
ResultWeightBasisText character The name that represents the form of the sample or portion of the sample which is associated with the result value (e.g., wet weight, dry weight, ash-free dry weight).
ResultTimeBasisText character The period of time (in days) over which a measurement was made. For example, BOD can be measured as 5 day or 20 day BOD.
ResultTemperatureBasisText character The name that represents the controlled temperature at which the sample was maintained during analysis, e.g. 25 deg BOD analysis.
ResultParticleSizeBasisText character User defined free text describing the particle size class for which the associated result is defined.
PrecisionValue character A measure of mutual agreement among individual measurements of the same property usually under prescribed similar conditions.
ResultCommentText character Free text with general comments concerning the result.
USGSPCode * character 5-digit number used in the US Geological Survey computerized data system, National Water Information System (NWIS), to uniquely identify a specific constituent.
ResultDepthHeightMeasure.MeasureValue + character A measurement of the vertical location (measured from a reference point) at which a result occurred.
ResultDepthHeightMeasure.MeasureUnitCode + character The code that represents the unit for measuring the item.
ResultDepthAltitudeReferencePointText + character The reference used to indicate the datum or reference used to establish the depth/altitude of a result.
SubjectTaxonomicName character The name of the organism from which a tissue sample was taken.
SampleTissueAnatomyName * character The name of the anatomy from which a tissue sample was taken.
ResultAnalyticalMethod.MethodIdentifier character The identification number or code assigned by the method publisher.
ResultAnalyticalMethod.MethodIdentifierContext character Identifies the source or data system that created or defined the identifier.
ResultAnalyticalMethod.MethodName character The title that appears on the method from the method publisher.
MethodDescriptionText * character A brief summary that provides general information about the method.
LaboratoryName character The name of Lab responsible for the result.
AnalysisStartDate character The calendar date on which the analysis began.
ResultLaboratoryCommentText character Remarks which further describe the laboratory procedures which produced the result.
DetectionQuantitationLimitTypeName character Text describing the type of detection or quantitation level used in the analysis of a characteristic.
DetectionQuantitationLimitMeasure.MeasureValue numeric Constituent concentration that, when processed through the preparation/extraction and analysis, is considered different from a blank. Measure value is given in the units stored in DetectionQuantitationLimitMeasure/MeasureUnitCode.
DetectionQuantitationLimitMeasure.MeasureUnitCode character The code that represents the unit for measuring the item.
PreparationStartDate character The calendar date when the preparation/extraction of the sample for analysis began.
ActivityStartDateTime POSIXct Activity start date and time converted to POSIXct UTC.
ActivityEndDateTime POSIXct Activity end date and time converted to POSIXct UTC.

* = elements only in NWIS  + = elements only in STORET

There are also several useful attributes attached to the data frame:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>character</td>
<td>The url used to generate the data</td>
</tr>
<tr>
<td>siteInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested sites</td>
</tr>
</tbody>
</table>
**readWQPsummary**

Summary of Data Available from Water Quality Portal

**Description**

Returns a list of sites from the Water Quality Portal web service. This function gets the data from: [https://www.waterqualitydata.us](https://www.waterqualitydata.us). Arguments to the function should be based on [https://www.waterqualitydata.us/webservices_documentation](https://www.waterqualitydata.us/webservices_documentation). The information returned from this function describes the available data at the WQP sites, and some metadata on the sites themselves.

**Usage**

readWQPsummary(...)

**Arguments**

...  

see [https://www.waterqualitydata.us/webservices_documentation](https://www.waterqualitydata.us/webservices_documentation) for a complete list of options. A list of arguments can also be supplied.

**Value**

A data frame with at least the following columns:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Provider&quot;</td>
<td>character</td>
<td>Providing database.</td>
</tr>
<tr>
<td>&quot;MonitoringLocationIdentifier&quot;</td>
<td>character</td>
<td>A designator used to describe the unique name, number, or code assigned to identify the monitoring location.</td>
</tr>
<tr>
<td>&quot;YearSummarized&quot;</td>
<td>numeric</td>
<td>The year of the summary</td>
</tr>
</tbody>
</table>
renameNWISColumns

renameColumns

"CharacteristicType" character CharacteristicType
"CharacteristicName" character The object, property, or substance which is evaluated or enumerated by either a direct field measurement, a direct field observation, or by laboratory analysis of material collected in the field.
"ActivityCount" numeric The number of times the location was sampled
"ResultCount" numeric The number of individual data results.
"LastResultSubmittedDate" Date Date when data was last submitted.
"OrganizationIdentifier" character A designator used to uniquely identify a unique business establishment.
"OrganizationFormalName" character The legal designator (i.e. formal name) of an organization.
"MonitoringLocationName" character MonitoringLocationName
"ResolvedMonitoringLocationTypeName" character MonitoringLocationTypeName
"HUCEightDigitCode" character 8-digit HUC id.
"MonitoringLocationUrl" character URL to monitoring location.
"CountyName" character County of sampling location.
"StateName" character State of sampling location.
"MonitoringLocationLatitude" numeric latitude of sampling location.
"MonitoringLocationLongitude" numeric longitude of sampling location.

Examples

# Summary of a single site for the last 5 years:
site_5 <- readWQPsummary(siteid="USGS-07144100",
                         summaryYears=5)

# Summary of a single site for the full period of record:
site_all <- readWQPsummary(siteid="USGS-07144100",
                          summaryYears="all")

# Summary of the data available from streams in a single county:
dane_county_data <- readWQPsummary(countycode = "US:55:025",
                                    summaryYears = 5,
                                    siteType = "Stream")

# Summary of the data all available from lakes in a single county:
lake_sites <- readWQPsummary(siteType = "Lake, Reservoir, Impoundment",
                              countycode = "US:55:025")

# Summary of the data available for the last 5 years in New Jersey:
state1 <- readWQPsummary(statecode="NJ",
                         summaryYears = 5,
                         siteType = "Stream")
renameNWISColumns

Description

Rename columns coming back from NWIS data retrievals. Daily and unit value columns have names derived from their data descriptor, parameter, and statistic codes. This function reads information from the header and the arguments in the call to rename those columns.

Usage

renameNWISColumns(
  rawData,
  p00010 = "Wtemp",
  p00045 = "Precip",
  p00060 = "Flow",
  p00065 = "GH",
  p00095 = "SpecCond",
  p00300 = "DO",
  p00400 = "pH",
  p62611 = "GWL",
  p63680 = "Turb",
  p72019 = "WLBS",
  ...
)

Arguments

rawData the daily- or unit-values datset retrieved from NWISweb.
p00010 the base name for parameter code 00010.
p00045 the base name for parameter code 00045.
p00060 the base name for parameter code 00060.
p00065 the base name for parameter code 00065.
p00095 the base name for parameter code 00095.
p00300 the base name for parameter code 00300.
p00400 the base name for parameter code 00400.
p62611 the base name for parameter code 62611.
p63680 the base name for parameter code 63680.
p72019 the base name for parameter code 72019.
... named arguments for the base name for any other parameter code. The form of the name must be like pXXXXX, where XXXXX is the parameter code.

Value

A dataset like data with selected columns renamed.
**Note**

The following statistics codes are converted by `renameNWISColumns`.

- **00000** Instantaneous Value, suffix: Inst
- **00001** Maximum value, suffix: Max
- **00002** Minimum value, suffix: Min
- **00003** Mean value, no suffix
- **00006** Sum of values, suffix: Sum
- **00007** Modal value, suffix: Mode
- **00008** Median value, suffix: Median
- **00012** Equivalent mean value, suffix: EqMean
- **00021** Tidal high-high value, suffix: HiHiTide
- **00022** Tidal low-high value, suffix: LoHiTide
- **00023** Tidal high-low value, suffix: HiLoTide
- **00024** Tidal low-low value, suffix: LoLoTide

**See Also**

`readNWISdv, readNWISuv`

**Examples**

```r
siteWithTwo <- '01480015'
startDate <- "2012-09-01"
endDate <- "2012-10-01"

twoResults <- readNWISdv(siteWithTwo, "00060", startDate, endDate)
names(twoResults)
renamedCols <- renameNWISColumns(twoResults)
names(renamedCols)
#Custom names:
newNames <- renameNWISColumns(twoResults, p00060="Discharge")
names(newNames)
```

---

**setAccess**

**Set data endpoint**

**Description**

*access* Indicate which dataRetrieval access code you want to use.

**Usage**

```r
setAccess(access = "public")
```
Arguments

access  code for data access. Options are: "public","internal","cooperator", or "USGS".
  • "internal" represents Access=3 ...for a single water science center
  • "USGS" represents Access=2 ...for all water science centers
  • "cooperator" represents Access=1
  • "public" represents Access=0, public access

Author(s)

Luke Winslow, Jordan S Read

Examples

setAccess('internal')
setAccess('public')

stateCd

US State Code Lookup Table

Description

Data originally pulled from https://www2.census.gov/geo/docs/reference/state.txt on April 1, 2015. On Feb. 11, 2022, the fields were updated with the file found in inst/extdata, which is used internally with NWIS retrievals.

Value

stateCd data frame.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATE</td>
<td>character</td>
<td>FIPS State Code</td>
</tr>
<tr>
<td>STUSAB</td>
<td>character</td>
<td>Official United States Postal Service (USPS) Code</td>
</tr>
<tr>
<td>STATE_NAME</td>
<td>character</td>
<td>State Name</td>
</tr>
<tr>
<td>STATENS</td>
<td>character</td>
<td>Geographic Names Information System Identifier (GNISID)</td>
</tr>
</tbody>
</table>

Examples

head(stateCd)
stateCdLookup  

**State code look up**

**Description**

Function to simplify finding state and state code definitions. Used in readNWISdata and readWQPdata.

**Usage**

```
stateCdLookup(input, outputType = "postal")
```

**Arguments**

- **input** could be character (full name, abbreviation, id), or numeric (id)
- **outputType** character can be "postal","fullName","tableIndex", or "id".

**Examples**

```
fullName <- stateCdLookup("wi", "fullName")
abbrev <- stateCdLookup("Wisconsin", "postal")
id <- stateCdLookup("WI", "id")
name <- stateCdLookup(55, "fullName")
index <- stateCdLookup("WI", "tableIndex")
stateCd[index,]
stateCdLookup(c("West Virginia", "Wisconsin", 200, 55, "MN"))
```

whatNWISdata  

**USGS data availability**

**Description**

Imports a table of available parameters, period of record, and count. See [https://waterservices.usgs.gov/rest/Site-Service.html](https://waterservices.usgs.gov/rest/Site-Service.html) for more information.

**Usage**

```
whatNWISdata(..., convertType = TRUE)
```

**Arguments**

- **...** see [https://waterservices.usgs.gov/rest/Site-Service.html](https://waterservices.usgs.gov/rest/Site-Service.html) for a complete list of options. A list of arguments can also be supplied.
- **convertType** logical, defaults to TRUE. If TRUE, the function will convert the data to dates, datetimes, numerics based on a standard algorithm. If false, everything is returned as a character.
Value

A data frame with the following columns:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agency_cd</td>
<td>character</td>
<td>The NWIS code for the agency reporting the data</td>
</tr>
<tr>
<td>site_no</td>
<td>character</td>
<td>The USGS site number</td>
</tr>
<tr>
<td>station_nm</td>
<td>character</td>
<td>Site name</td>
</tr>
<tr>
<td>site_tp_cd</td>
<td>character</td>
<td>Site type</td>
</tr>
<tr>
<td>dec_lat_va</td>
<td>numeric</td>
<td>Decimal latitude</td>
</tr>
<tr>
<td>dec_long_va</td>
<td>numeric</td>
<td>Decimal longitude</td>
</tr>
<tr>
<td>coord_acy_cd</td>
<td>character</td>
<td>Latitude-longitude accuracy</td>
</tr>
<tr>
<td>dec_coord_datum_cd</td>
<td>character</td>
<td>Decimal Latitude-longitude datum</td>
</tr>
<tr>
<td>alt_va</td>
<td>character</td>
<td>Altitude of Gage or land surface</td>
</tr>
<tr>
<td>alt_acy_va</td>
<td>character</td>
<td>Altitude accuracy</td>
</tr>
<tr>
<td>alt_datum_cd</td>
<td>character</td>
<td>Altitude datum</td>
</tr>
<tr>
<td>huc_cd</td>
<td>character</td>
<td>Hydrologic unit code</td>
</tr>
<tr>
<td>data_type_cd</td>
<td>character</td>
<td>Data type</td>
</tr>
<tr>
<td>parm_cd</td>
<td>character</td>
<td>Parameter code</td>
</tr>
<tr>
<td>stat_cd</td>
<td>character</td>
<td>Statistical code</td>
</tr>
<tr>
<td>dd_nu</td>
<td>character</td>
<td>Internal database key</td>
</tr>
<tr>
<td>loc_web_ds</td>
<td>character</td>
<td>Additional measurement description</td>
</tr>
<tr>
<td>medium_grp_cd</td>
<td>character</td>
<td>Medium group code</td>
</tr>
<tr>
<td>parm_grp_cd</td>
<td>character</td>
<td>Parameter group code</td>
</tr>
<tr>
<td>srs_id</td>
<td>character</td>
<td>SRS ID</td>
</tr>
<tr>
<td>access_cd</td>
<td>character</td>
<td>Access code</td>
</tr>
<tr>
<td>begin_date</td>
<td>Date</td>
<td>Begin date</td>
</tr>
<tr>
<td>end_date</td>
<td>Date</td>
<td>End date</td>
</tr>
<tr>
<td>count_nu</td>
<td>integer</td>
<td>Record count</td>
</tr>
<tr>
<td>parameter_group_nm</td>
<td>character</td>
<td>Parameter group name</td>
</tr>
<tr>
<td>parameter_nm</td>
<td>character</td>
<td>Parameter name</td>
</tr>
<tr>
<td>casrn</td>
<td>character</td>
<td>Chemical Abstracts Service (CAS) Registry Number</td>
</tr>
<tr>
<td>srsname</td>
<td>character</td>
<td>Substance Registry Services</td>
</tr>
<tr>
<td>parameter_units</td>
<td>character</td>
<td>Parameter units</td>
</tr>
</tbody>
</table>

There are also several useful attributes attached to the data frame:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>character</td>
<td>The url used to generate the data</td>
</tr>
<tr>
<td>comment</td>
<td>character</td>
<td>Header comments from the RDB file</td>
</tr>
<tr>
<td>queryTime</td>
<td>POSIXct</td>
<td>The time the data was returned</td>
</tr>
</tbody>
</table>

Examples
whatNWISsites <- whatNWISdata(siteNumber = '05114000')
# To find just unit value ('instantaneous') data:
uvData <- whatNWISdata(siteNumber = '05114000', service="uv")
uvDataMulti <- whatNWISdata(siteNumber = c('05114000', '09423350'), service=c("uv","dv"),
                           parameterCd = c("00060","00010"),
                           statCd = "00003")

## Site Data Import from NWIS

**Description**

Returns a list of sites from the NWIS web service. This function gets the data from: [https://waterservices.usgs.gov/rest/Site-Test-Tool.html](https://waterservices.usgs.gov/rest/Site-Test-Tool.html). Mapper format is used.

**Usage**

whatNWISsites(...)

**Arguments**

... see [https://waterservices.usgs.gov/rest/Site-Service.html](https://waterservices.usgs.gov/rest/Site-Service.html) for a complete list of options. A list (or lists) can also be supplied.

**Value**

A data frame with at least the following columns:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agency_cd</td>
<td>character</td>
<td>The NWIS code for the agency reporting the data</td>
</tr>
<tr>
<td>site_no</td>
<td>character</td>
<td>The USGS site number</td>
</tr>
<tr>
<td>station_nm</td>
<td>character</td>
<td>Station name</td>
</tr>
<tr>
<td>site_tp_cd</td>
<td>character</td>
<td>Site type code</td>
</tr>
<tr>
<td>dec_lat_va</td>
<td>numeric</td>
<td>Decimal latitude</td>
</tr>
<tr>
<td>dec_long_va</td>
<td>numeric</td>
<td>Decimal longitude</td>
</tr>
<tr>
<td>queryTime</td>
<td>POSIXct</td>
<td>Query time</td>
</tr>
</tbody>
</table>

There are also several useful attributes attached to the data frame:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>character</td>
<td>The url used to generate the data</td>
</tr>
<tr>
<td>queryTime</td>
<td>POSIXct</td>
<td>The time the data was returned</td>
</tr>
</tbody>
</table>
whatWQPdata

Examples

sitelistPhos <- whatNWISsites(stateCd="OH",parameterCd="00665")
oneSite <- whatNWISsites(sites="05114000")

whatWQPdata

Data Available from Water Quality Portal

Description

Returns a list of sites from the Water Quality Portal web service. This function gets the data from: https://www.waterqualitydata.us. Arguments to the function should be based on https://www.waterqualitydata.us/webservices_documentation. The information returned from this function describes the available data at the WQP sites, and some metadata on the sites themselves.

Usage

whatWQPdata(..., saveFile = tempfile())

Arguments

... see https://www.waterqualitydata.us/webservices_documentation for a complete list of options. A list of arguments can also be supplied.

saveFile path to save the incoming geojson output.

Value

A data frame with at least the following columns:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;type_a&quot;</td>
<td>character</td>
<td>Geojson type</td>
</tr>
<tr>
<td>&quot;features.type&quot;</td>
<td>character</td>
<td>Geojson feature type</td>
</tr>
<tr>
<td>&quot;type1&quot;</td>
<td>character</td>
<td>Geojson spatial type</td>
</tr>
<tr>
<td>&quot;coordinates&quot;</td>
<td>list</td>
<td>List of longitude/latitude</td>
</tr>
<tr>
<td>&quot;ProviderName&quot;</td>
<td>character</td>
<td>The name of the database that provided the data to the Water Quality Portal</td>
</tr>
<tr>
<td>&quot;OrganizationIdentifier&quot;</td>
<td>character</td>
<td>A designator used to uniquely identify a unique business establishment.</td>
</tr>
<tr>
<td>&quot;OrganizationFormalName&quot;</td>
<td>character</td>
<td>The legal designator (i.e. formal name) of an organization.</td>
</tr>
<tr>
<td>&quot;MonitoringLocationIdentifier&quot;</td>
<td>character</td>
<td>A designator used to describe the unique name, number, or code assigned to identify the monitoring location.</td>
</tr>
<tr>
<td>&quot;MonitoringLocationName&quot;</td>
<td>character</td>
<td>The designator specified by the sampling organization for the site at which sampling or other activities are conducted.</td>
</tr>
<tr>
<td>&quot;MonitoringLocationTypeName&quot;</td>
<td>character</td>
<td>The descriptive name for a type of monitoring location.</td>
</tr>
<tr>
<td>&quot;ResolvedMonitoringLocationTypeName&quot;</td>
<td>character</td>
<td>The 8 digit federal code used to identify the hydrologic unit of the monitoring location.</td>
</tr>
<tr>
<td>&quot;HUCEightDigitCode&quot;</td>
<td>character</td>
<td>The 8 digit federal code used to identify the hydrologic unit of the monitoring location.</td>
</tr>
<tr>
<td>&quot;siteUrl&quot;</td>
<td>character</td>
<td>URL to site information</td>
</tr>
<tr>
<td>&quot;activityCount&quot;</td>
<td>numeric</td>
<td></td>
</tr>
</tbody>
</table>
whatWQPsamples

Site Data Import from Water Quality Portal

Description

Returns a list of sites from the Water Quality Portal web service. This function gets the data from: https://www.waterqualitydata.us. Arguments to the function should be based on https://www.waterqualitydata.us/webservices_documentation. The return from this function returns the basic metadata on WQP sites. It is generally faster than the whatWQPdata function, but does not return information on what data was collected at the site.

Usage

whatWQPsamples(...)
whatWQPmetrics(...)
whatWQPsites(...)

Arguments

... see https://www.waterqualitydata.us/webservices_documentation for a complete list of options. A list of arguments can also be supplied.

Details

The readWQPsummary function has
## Value

A data frame with at least the following columns:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OrganizationIdentifier</td>
<td>character</td>
<td>A designator used to uniquely identify a unique business establishment.</td>
</tr>
<tr>
<td>OrganizationFormalName</td>
<td>character</td>
<td>The legal designator (i.e. formal name) of an organization.</td>
</tr>
<tr>
<td>MonitoringLocationIdentifier</td>
<td>character</td>
<td>A designator used to describe the unique name, number, or code.</td>
</tr>
<tr>
<td>MonitoringLocationName</td>
<td>character</td>
<td>The designator specified by the sampling organization.</td>
</tr>
<tr>
<td>MonitoringLocationTypeName</td>
<td>character</td>
<td>The descriptive name for a type of monitoring location.</td>
</tr>
<tr>
<td>MonitoringLocationDescriptionText</td>
<td>character</td>
<td>Text description of the monitoring location.</td>
</tr>
<tr>
<td>HUCEightDigitCode</td>
<td>character</td>
<td>The 8 digit federal code used to identify the hydrologic unit.</td>
</tr>
<tr>
<td>DrainageAreaMeasure/MeasureValue</td>
<td>character</td>
<td>The drainage basin of a lake, stream, wetland, or estuary.</td>
</tr>
<tr>
<td>DrainageAreaMeasure/MeasureUnitCode</td>
<td>character</td>
<td>The code that represents the unit for measuring the item.</td>
</tr>
<tr>
<td>ContributingDrainageAreaMeasure/MeasureValue</td>
<td>character</td>
<td>The contributing drainage area of a lake, stream, wetland, or estuary.</td>
</tr>
<tr>
<td>ContributingDrainageAreaMeasure/MeasureUnitCode</td>
<td>character</td>
<td>The code that represents the unit for measuring the item.</td>
</tr>
<tr>
<td>LatitudeMeasure</td>
<td>numeric</td>
<td>The measure of the angular distance on a meridian north of the equator.</td>
</tr>
<tr>
<td>LongitudeMeasure</td>
<td>numeric</td>
<td>The measure of the angular distance on a meridian east of the prime meridian.</td>
</tr>
<tr>
<td>SourceMapScaleNumeric</td>
<td>character</td>
<td>The number representing the proportional distance on the ground.</td>
</tr>
<tr>
<td>HorizontalAccuracyMeasure/MeasureValue</td>
<td>character</td>
<td>The horizontal measure of the relative accuracy of the latitude coordinates.</td>
</tr>
<tr>
<td>HorizontalAccuracyMeasure/MeasureUnitCode</td>
<td>character</td>
<td>The code that represents the unit for measuring the item.</td>
</tr>
<tr>
<td>HorizontalCoordinateReferenceSystemDatumName</td>
<td>character</td>
<td>The name that identifies the method used to determine the latitude and longitude coordinates.</td>
</tr>
<tr>
<td>VerticalMeasure/MeasureValue</td>
<td>character</td>
<td>The measure of the vertical distance above or below a reference datum.</td>
</tr>
<tr>
<td>VerticalMeasure/MeasureUnitCode</td>
<td>character</td>
<td>The code that represents the unit for measuring the item.</td>
</tr>
<tr>
<td>VerticalCoordinateReferenceSystemDatumName</td>
<td>character</td>
<td>The name of the reference datum used to determine the vertical distance.</td>
</tr>
<tr>
<td>CountryCode</td>
<td>character</td>
<td>A code designator used to identify a principal geopolitical administrative unit.</td>
</tr>
<tr>
<td>StateCode</td>
<td>character</td>
<td>A code designator used to identify a U.S. county or county equivalent.</td>
</tr>
<tr>
<td>CountyCode</td>
<td>character</td>
<td>Name of the county in which the well is completed.</td>
</tr>
<tr>
<td>AquiferName</td>
<td>character</td>
<td>Name of the aquifer in which the well is completed.</td>
</tr>
<tr>
<td>FormationTypeText</td>
<td>character</td>
<td>Name of the primary formation or soil unit, in which the well is completed.</td>
</tr>
<tr>
<td>AquiferTypeName</td>
<td>character</td>
<td>The type of aquifer, such as confined or unconfined.</td>
</tr>
<tr>
<td>ConstructionDateText</td>
<td>character</td>
<td>Date of construction when well was completed. May be year only.</td>
</tr>
<tr>
<td>WellDepthMeasure/MeasureValue</td>
<td>character</td>
<td>The depth below land surface datum (LSD) to the bottom of the well.</td>
</tr>
<tr>
<td>WellDepthMeasure/MeasureUnitCode</td>
<td>character</td>
<td>The code that represents the unit for measuring the item.</td>
</tr>
<tr>
<td>WellHoleDepthMeasure/MeasureValue</td>
<td>character</td>
<td>Depth below land surface datum (LSD) to the bottom of the borehole.</td>
</tr>
<tr>
<td>WellHoleDepthMeasure/MeasureUnitCode</td>
<td>character</td>
<td>The code that represents the unit for measuring the item.</td>
</tr>
<tr>
<td>queryTime</td>
<td>POSIXct</td>
<td>Query time</td>
</tr>
</tbody>
</table>

* element is only in NWIS

### See Also

whatNWISdata
Examples

```r
site1 <- whatWQPsamples(siteid="USGS-01594440")
type <- "Stream"
sites <- whatWQPsamples(countycode="US:55:025",siteType=type)

type <- "Stream"
sites <- whatWQPmetrics(countycode="US:55:025",siteType=type)
lakeSites <- whatWQPmetrics(siteType = "Lake, Reservoir, Impoundment", statecode = "US:55")
site1 <- whatWQPsites(siteid="USGS-01594440")
type <- "Stream"
sites <- whatWQPsites(countycode="US:55:025",
characteristicName = "Phosphorus",
siteType=type)
```

---

**zeroPad**

*Pad string with leading zeros*

**Description**

Function to pad a string with leading zeros. Useful for parameter codes and USGS site IDs.

**Usage**

`zeroPad(x, padTo)`

**Arguments**

- `x` character
- `padTo` number Final desired length of the character

**Value**

`x` character returned with leading zeros
Examples

```r
pCode <- '10'
correctPCode <- zeroPad(pCode,5)
pCodes <- c('100','1000','0','12345','1565465465465465')
correctPCodes <- zeroPad(pCodes,5)
pCodeNA <- c(1,2,NA)
padPCodeNA <- zeroPad(pCodeNA,4)
```
Index

* IO
  renameNWISColumns, 54
* USGS
  constructNWISURL, 4
  countyCd, 8
  readNWISdv, 27
  readNWISpCode, 33
  readNWISqw, 35
  readNWISsite, 39
  readNWISuv, 44
  readWQPqw, 50
  stateCd, 57
  whatNWISdata, 58
  zeroPad, 64
* WQP
  constructWQPURL, 7
  readWQPdata, 46
  whatWQPdata, 61
  whatWQPsamples, 62
* countyCd
  countyCd, 8
* data
  constructNWISURL, 4
  constructWQPURL, 7
  readNWISdv, 27
  readNWISpCode, 33
  readNWISqw, 35
  readNWISsite, 39
  readNWISuv, 44
  readWQPqw, 50
  whatNWISdata, 58
  whatWQPdata, 61
  whatWQPsamples, 62
  zeroPad, 64
* import
  constructNWISURL, 4
  constructWQPURL, 7
  readNWISdv, 27
  readNWISpCode, 33
  readNWISqw, 35
  readNWISsite, 39
  readNWISuv, 44
* nldi
  findNLDI, 9
  get_nldi_sources, 13
* service
  constructNWISURL, 4
  constructWQPURL, 7
  readNWISdv, 27
  readNWISpCode, 33
  readNWISqw, 35
  readNWISsite, 39
  readNWISuv, 44
  readWQPdata, 46
  readWQPqw, 50
  whatNWISdata, 58
  whatWQPdata, 61
  whatWQPsamples, 62
  zeroPad, 64
* stateCd
  stateCd, 57
* web
  constructNWISURL, 4
  constructWQPURL, 7
  readNWISdv, 27
  readNWISpCode, 33
  readNWISqw, 35
  readNWISsite, 39
  readNWISuv, 44
INDEX

readWQPdata, 46
readWQPqw, 50
whatNWISdata, 58
whatWQPdata, 61
whatWQPsamples, 62
zeroPad, 64

addWaterYear, 3
calcWaterYear, 4
constructNWISURL, 4, 30, 32, 35, 38, 39, 42
constructUseURL, 6
constructWQPURL, 7
countyCd, 8
countyCdLookup, 8
dataRetrieval, 9

findNLDI, 9

GET, 12
get_nldi_sources, 13
getQuerySummary, 11
getWebServiceData, 12

importNGWMN, 13
importRDB1, 14, 26, 30, 32, 33, 35, 39, 42
importWaterML1, 16, 26, 28, 45
importWaterML2, 18
importWQP, 19, 53
is_dataRetrieval_user, 20

parameterCdFile, 21

readNGWMNdata, 21
readNGWMNlevels, 22
readNGWMNsites, 23
readNWISdata, 24
readNWISdv, 27, 56
readNWISgwl, 29
readNWISmeas, 31
readNWISpCode, 33
readNWISpeak, 34
readNWISqw, 35, 53
readNWISrating, 38
readNWISsite, 39
readNWISstat, 41
readNWISuse, 43
readNWISuv, 44, 56
readWQPdata, 20, 38, 46, 53

readWQPqw, 20, 38, 50
readWQPsummary, 53
renameNWISColumns, 17, 26, 28, 45, 54

setAccess, 56
stateCd, 57
stateCdLookup, 58

whatNWISdata, 58
whatNWISsites, 60
whatWQPdata, 61, 62
whatWQPmetrics (whatWQPsamples), 62
whatWQPsamples, 62
whatWQPsites, 20, 38, 53
whatWQPsites (whatWQPsamples), 62

zeroPad, 64