## Package ‘dataRetrieval’

**June 5, 2019**

**Type** Package

**Title** Retrieval Functions for USGS and EPA Hydrologic and Water Quality Data

**Version** 2.7.5

**Description** Collection of functions to help retrieve U.S. Geological Survey (USGS) and U.S. Environmental Protection Agency (EPA) water quality and hydrology data from web services. USGS web services are discovered from National Water Information System (NWIS) (<https://waterservices.usgs.gov/> and <https://waterdata.usgs.gov/nwis>). Both EPA and USGS water quality data are obtained from the Water Quality Portal (<https://www.waterqualitydata.us/>).

**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 (>= 3.0)

**Imports** httr (>= 1.0.0), curl, lubridate (>= 1.5.0), stats, utils, xml2, readr (>= 1.0.0), jsonlite

**Suggests** htmlTable, knitr, testthat

**BuildVignettes** true

**VignetteBuilder** knitr

**BugReports** https://github.com/USGS-R/dataRetrieval/issues

**URL** https://pubs.usgs.gov/tm/04/a10/

**RoxygenNote** 6.1.1

**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],
R topics documented:

addWaterYear .................................................. 3
calcWaterYear ................................................ 4
constructNWISURL ........................................... 4
constructUseURL ............................................. 6
constructWQPURL ............................................ 7
countyCd ...................................................... 8
countyCdLookup ............................................ 8
dataRetrieval ............................................... 9
goingQuerySummary ....................................... 9
getWebServiceData ......................................... 10
importNGWMN ................................................. 10
importRDB1 ................................................... 11
importWaterML1 ............................................. 13
importWaterML2 ............................................. 15
importWQP ................................................... 16
parameterCdFile ............................................ 17
readNGWMNdata ........................................... 18
readNGWMNlevels ......................................... 19
readNGWMNsites ........................................... 20
readNWISdata ............................................. 21
readNWISdv ................................................ ..24
readNWISgwl .................................................. 26
readNWISmeas ............................................... 27
readNWISpCode ............................................. 29
readNWISpeak ............................................... 30
readNWISqw ................................................... 31
readNWISrating ............................................. 34
readNWISsite ............................................... 35
readNWISstat ................................................ 37
readNWISuse ............................................... 38
readNWISuv .................................................. 40
readWQPdata ................................................ 43
readWQPqw .................................................... 46
renameNWISColumns ...................................... 49
setAccess ...................................................... 50
stateCd ......................................................... 51
stateCdLookup .............................................. 52
whatNWISdata ............................................. 52
whatNWISsites ............................................. 54
**`addWaterYear`**

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

```r
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 <- readWQPpw('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 2015-09-30. See the USGS definition at [https://water.usgs.gov/nwc/explain_data.html](https://water.usgs.gov/nwc/explain_data.html).

**Value**

numeric vector indicating the water year

**Examples**

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

---

`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](https://nwis.waterdata.usgs.gov/nwis/qwdata) A list of parameter codes can be found here: [https://nwis.waterdata.usgs.gov/nwis/pmcodes/](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](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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>siteNumbers</td>
<td>string or vector of strings USGS site number. This is usually an 8 digit number</td>
</tr>
<tr>
<td>parameterCd</td>
<td>string or vector of USGS parameter code. This is usually an 5 digit number.</td>
</tr>
<tr>
<td>startDate</td>
<td>character starting date for data retrieval in the form YYYY-MM-DD. Default is &quot;&quot;</td>
</tr>
<tr>
<td>endDate</td>
<td>character ending date for data retrieval in the form YYYY-MM-DD. Default is &quot;&quot;</td>
</tr>
<tr>
<td>service</td>
<td>string USGS service to call. Possible values are &quot;dv&quot; (daily values), &quot;uv&quot; (unit/instantaneous values), &quot;qw&quot; (water quality data), &quot;gwlevels&quot; (groundwater), and &quot;rating&quot; (rating curve), &quot;peak&quot;, &quot;meas&quot; (discrete streamflow measurements), &quot;stat&quot; (statistics web service BETA).</td>
</tr>
<tr>
<td>statCd</td>
<td>string or vector USGS statistic code only used for daily value service. This is usually 5 digits. Daily mean (00003) is the default.</td>
</tr>
<tr>
<td>format</td>
<td>string, can be &quot;tsv&quot; or &quot;xml&quot;, and is only applicable for daily and unit value requests. &quot;tsv&quot; returns results faster, but there is a possibility 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 &quot;tsv&quot; option, but the user must carefully check the results to see if the data returns matches what is expected. The default is therefore &quot;xml&quot;.</td>
</tr>
<tr>
<td>expanded</td>
<td>logical defaults to TRUE. If TRUE, retrieves additional information, only applicable for qw data.</td>
</tr>
<tr>
<td>ratingType</td>
<td>can be &quot;base&quot;, &quot;corr&quot;, or &quot;exsa&quot;. Only applies to rating curve data.</td>
</tr>
<tr>
<td>statReportType</td>
<td>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 readNWISdvd 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.</td>
</tr>
<tr>
<td>statType</td>
<td>character Only used for statistics service requests. 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 <a href="https://waterservices.usgs.gov/rest/Statistics-Service.html">https://waterservices.usgs.gov/rest/Statistics-Service.html</a> for a full list of codes.</td>
</tr>
</tbody>
</table>

Value

url string
Examples

```r
site_id <- '01594440'
startDate <- '1985-01-01'
endDate <- '
pcodes <- c("00003","00010")
url_daily <- constructNWISURL(site_id,pCode,
    startDate,endDate,'dv',statCd=c("00003","00001"))
url_unit <- constructNWISURL(site_id,pCode,2012-06-28,2012-06-30,'iv')

url_qw_single <- constructNWISURL(site_id,"01075",startDate,endDate,'qw')
url_qw <- constructNWISURL(site_id,c('01075','00029','00453'),
    startDate,endDate,'qw')
url_daily_tsv <- constructNWISURL(site_id,pCode,startDate,endDate,'dv',
    statCd=c("00003","00001"),format="tsv")
url_rating <- constructNWISURL(site_id,service="rating",ratingType="base")
url_peak <- constructNWISURL(site_id,service="peak")
url_meas <- constructNWISURL(site_id,service="meas")
urlQW <- constructNWISURL("45045609225801","70300",startDate="",endDate="","qw",expanded=TRUE)
```

---

**constructUseURL**  
*Construct URL for NWIS water use data service*

**Description**
Reconstructs URLs to retrieve data from here: [https://waterdata.usgs.gov/nwis/wu](https://waterdata.usgs.gov/nwis/wu)

**Usage**

```r
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)

**Value**

- **url string**

**Examples**

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

Construct WQP url for data retrieval

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

```r
site_id <- '01594440'
startDate <- '1985-01-01'
endDate <- '
pcodes <- c("00060","00010")
url_wqp <- constructWQPURL(paste("USGS",site_id,sep=""),
c('01075','00029','00453'),
startDate,endDate)
url_wqp
ccharNames <- c("Temperature",
"Temperature, sample",
"Temperature, water",
"Temperature, water, deg F")
obs_url_orig <- constructWQPURL(siteNumbers = c("IIDFG-4IWSSPAHS",
"USGS-02352560"),
parameterCd = charNames,
startDate,""
obs_url_orig
```
US County Code Lookup Table

Description

Data pulled from http://www2.census.gov/geo/docs/reference/codes/files/national_county.txt on April 1, 2015.

Value

countyCd data frame.

<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)

County code look up

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

```r
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")
```

dataRetrieval Retrieval functions for USGS and EPA data

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>

ggetQuerySummary getting header information from a WQP query

Description

getting header information from a WQP query

Usage

ggetQuerySummary(url)

Arguments

url the query url
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

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

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 <- '00000'
obs_url <- constructNWiSURL(siteNumber, property, startDate, endDate, 'dv')

rawData <- getWebServiceData(obs_url)
importRDB1

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.

Value

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

Examples

"service=SOS","version=2.0.0",
"observedProperty=urn:oog:OGC:GroundWaterLevel",
"responseFormat=text/xml",
"featureOfInterest=WW_GWDP_GEOSERVER.USGS.403836085374401",sep="&")
data <- importNWIS(obs_url)

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

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

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
- **tz**: character to set timezone attribute of datetime. Default converts the datetimes to UTC (properly accounting for daylight savings times based on the data’s provided tz_cd column). Recommended US values include "UTC","America/New_York","America/Chicago","America/Denver","America/Los_Angeles","America/Anchorage","America/Honolulu","America/Jamaica","America/Phoenix", and "America/Metlakatla". For a complete list, see https://en.wikipedia.org/wiki/List_of_tz_database_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>POSIXct</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 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>character</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"
start_date <- "2012-09-01"
end_date <- "2012-10-01"
offering <- "00003"
```
importWaterML

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

importWaterML(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
tz  character to set timezone attribute of datetime. Default converts the datetimes to UTC (properly accounting for daylight savings times based on the data’s provided tz_cd column). Recommended US values include "UTC","America/New_York","America/Chicago" "America/Denver","America/Los_Angeles","America/Anchorage","America/Honolulu","America/Jamaica", and "America/Metlakatla". For a complete list, see https://en.wikipedia.org/wiki/List_of_TZ_database_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>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)
```
import WaterML2

groundWaterSite <- "431049071324301"
startGW <- "2013-10-01"
endGW <- "2014-06-30"

groundwaterExampleURL <- constructNWISURL(groundWaterSite, NA,
startGW,endGW, service="gwlevels")
groundWater <- importWaterMLI(groundwaterExampleURL)
groundWater2 <- importWaterMLI(groundwaterExampleURL, asDateTime=TRUE)

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

# Two sites, two pcodes, one site has two data descriptors:
siteNumber <- c('01400015','04085427')
obsv_url <- constructNWISURL(siteNumber,c("00000","00010"),startDate,endDate,'dv')
data <- importWaterMLI(obs_url)
data$dateTime <- as.Date(data$dateTime)
data <- renameNWISColumns(data)
names(attributes(data))
attr(data, "url")
attr(data, "disclaimer")
inactiveSite <- "05212700"
inactiveSite <- constructNWISURL(inactiveSite, "00000", "2014-01-01", "2014-01-10","dv")
inactiveSite <- importWaterMLI(inactiveSite)
inactiveAndActive <- c("07334200","05212700")
inactiveAndActive <- constructNWISURL(inactiveAndActive, "00000", "2014-01-01", "2014-01-10","dv")
inactiveAndActive <- importWaterMLI(inactiveAndActive)

# Timezone change with specified local timezone:
tzURL <- constructNWISURL("04027000", c("00300","63680"), "2011-11-05", "2011-11-07","uv")
tzIssue <- importWaterMLI(tzURL, TRUE, "America/Chicago")

# raw XML
url <- constructNWISURL(service = "dv", siteNumber = '02319300', parameterCd = "00000",
startDate = "2014-01-01", endDate = "2014-01-10")
raw <- http::content(http::GET(url), as = 'raw')
rawParsed <- importWaterMLI(raw)

fullPath <- file.path(filePath, fileName)
importFile <- importWaterMLI(fullPath,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

baseURL <- "https://waterservices.usgs.gov/nwis/dv/?format=waterml2.0"
URL <- paste(baseURL, "sites=01646500”,
    "startDT=2014-09-01",
    "endDT=2014-09-08",
    "statCd=0003",
    "parameterCd=00060",sep="&")
timesereis <- importWaterML2(URL, asDateTime=TRUE, tz="UTC")
Arguments

obs_url character URL to Water Quality Portal # @keywords data import USGS web service
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"
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

retval 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)
STORETEx <- constructWQPURL('WIDNR_WQX-10032762','Specific conductance',' ',' ')
STORETdata <- importWQP(STORETEx)

parameterCdFile    List of USGS parameter codes

Description

Complete list of USGS parameter codes as of May 22, 2017.
readNGWMNdata

**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])
```

---

**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.
- **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/Anchoragae", 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.
readNGWMNlevels

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

Description

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

Usage

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.
readNGWMNsites

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 <- "USGS.401544112060301"
#noDataSite <- readNGWMNlevels(siteNumbers = noDataSite)
```

readNGWMNsites


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.
readNWISdata

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</td>
<td>numeric</td>
<td>Site latitude and longitude</td>
</tr>
</tbody>
</table>

Examples

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

# non-USGS site
site <- "MBMG.103306"
siteInfo <- readNWISsites(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" (for daily values), "gwlevels" (for groundwater levels), "site" (for site service), "qw" (water-quality), "measurement", and "stat" (for statistics service). Note: "qw" and "measurement" calls go to: https://mwi.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 returns in POSIXct. Use as.POSIXct() 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>

See Also

renameNWISColumns, importWaterML1, importRDB1

Examples

# Examples not run for time considerations
dataTemp <- readNWISdata(stateCd="OH", parameterCd="00010", service="dv")instFlow <- readNWISdata(sites="05114000", service="iv",parameterCd="00060",}
```
readNWISdata

startDate = "2014-05-01T00:00Z", endDate = "2014-05-01T12:00Z"

instFlowCDT <- readNWISdata(sites = c("05114000", "04072150"), 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("04025000", "040263491"),
service = "iv", parameterCd = "00060")
bBoxEx <- readNWISdata(bBox = c(-83.365, -81, 38.5), parameterCd = "00010")

startDate <- as.Date("2013-10-01")
endDate <- as.Date("2014-09-30")
waterYear <- readNWISdata(bBox = c(-83.365, -81, 38.5), parameterCd = "00010",
service = "dv", startDate = startDate, endDate = endDate)
siteInfo <- readNWISdata(stateCd = "WI", parameterCd = "00010",
hasDataTypeCd = "iv", service = "site")
qwData <- readNWISdata(bBox = c(-82.5, 41.52, -81, 41), startDate = as.Date("2000-01-01"),
drain_area_va_min = 50, qw_count_nu = 50, qw_attributes = "expanded",
wq.sample.wide = "wide", list.of_search_criteria = c("lat_long.bounding.box",
"drain_area.va", "obs_count_nu"), service = "qw")
temp <- readNWISdata(bBox = c(-83.365, -81, 38.5), parameterCd = "00010", service = "site",
seriesCatalogOutput = TRUE)
wigwL <- readNWISdata(stateCd = "WI", service = "gwlevels")
meas <- readNWISdata(state_cd = "WI", service = "measurements", format = "rdb_expanded")

waterYearStat <- readNWISdata(stateCd = "03112500", service = "stat", statReportType = "annual",
statYearType = "water", missingData = "on")
monthlyStat <- readNWISdata(stateCd = "03112500", "03111520"),
service = "stat",
statReportType = "monthly")
dailyStat <- readNWISdata(stateCd = "03112500", "03111520"),
service = "stat",
statReportType = "daily",
statType = c("p25", "p50", "p75", "min", "max"),
parameterCd = "00065")

allDailyStats <- readNWISdata(stateCd = "03111548"),
service = "stat",
statReportType = "daily",
statType = c("p25", "p50", "p75", "min", "max"),
parameterCd = "00060")

dailyWV <- readNWISdata(stateCd = "West Virginia", parameterCd = "00060")

arg.list <- list(stateCd = "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(state = "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)
today <- readNWISdata(service="iv", startDate = Sys.Date(),
    parameterCd = "00060", siteNumber = "05114000")

---

### 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'
endDate <- '2012-06-30'
pCode <- '00060'

rawDailyQ <- readNWIISdv(site_id,pCode, startDate, endDate)
rawDailyQAndTempMeanMax <- readNWIISdv(site_id,c('00010', '00060'),startDate, endDate, statCd=c('00001', '00003'))
rawDailyQAndTempMeanMax <- renameNWISColumns(rawDailyQAndTempMeanMax)
rawDailyMultiSites<- readNWIISdv(c("01491000", "01645000"),c('00010', '00060'), startDate, endDate, statCd=c('00001', '00003'))
# Site with no data:
x <- readNWIISdv("10258500", "00060", "2014-09-08", "2014-09-14")
names(attributes(x))
attr(x, "siteInfo")
attr(x, "variableInfo")

site <- "05212700"
notActive <- readNWIISdv(site, "00060", "2014-01-01", "2014-01-07")
```
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 = "", 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.
endDate character ending date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the latest possible record.
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>
</tbody>
</table>
readNWISmeas

lev_tz_cd character Time datum
lev_va numeric Water level value in feet below land surface
sl_lev_va numeric Water level value in feet above specific vertical datum
lev_status_cd character The status of the site at the time the water level was measured
lev_agency_cd character The agency code of the person measuring the water level

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")
```

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/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, these times are in the time zone provided by the Olsen Names database.</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></td>
<td>The originally reported time zone</td>
</tr>
</tbody>
</table>

See Also

- constructNWISURL
- importRDB1
readNWISpCode

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)
```

readNWISpCode USGS Parameter Data Retrieval

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

```r
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

```r
paramINFO <- readNWisSpeak(c('01075','00060','00931'))
paramINFO <- readNWisSpeak(c('01075','00060','00931',NA))
```

**readNWisSpeak**  
*Peak flow data from USGS (NWIS)*

**Description**

Reads peak flow from NWISweb. Data is retrieved from [https://waterdata.usgs.gov/nwis](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
readNWisSpeak(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>
<tr>
<td>gage_ht</td>
<td>numeric</td>
<td>Gage height for the associated peak streamflow in feet</td>
</tr>
</tbody>
</table>
### readNWISqw

**Description**

Imports data from NWIS web service. This function gets the data from here: [https://nwis.waterdata.usgs.gov/nwis/qwdata](https://nwis.waterdata.usgs.gov/nwis/qwdata) A list of parameter codes can be found here: [https://nwis.waterdata.usgs.gov/nwis/pmcodes/](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](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")
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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('06811000')
peakdata<--readNWISpeak(stations,convertType=FALSE)
```
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:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INF</td>
<td>Information</td>
</tr>
<tr>
<td>PHY</td>
<td>Physical</td>
</tr>
<tr>
<td>INM</td>
<td>Inorganics, Major, Metals (major cations)</td>
</tr>
<tr>
<td>INN</td>
<td>Inorganics, Major, Non-metals (major anions)</td>
</tr>
<tr>
<td>NUT</td>
<td>Nutrient</td>
</tr>
<tr>
<td>MBI</td>
<td>Microbiological</td>
</tr>
<tr>
<td>BIO</td>
<td>Biological</td>
</tr>
<tr>
<td>IMN</td>
<td>Inorganics, Minor, Non-metals</td>
</tr>
<tr>
<td>IMM</td>
<td>Inorganics, Minor, Metals</td>
</tr>
<tr>
<td>TOX</td>
<td>Toxicity</td>
</tr>
<tr>
<td>OPE</td>
<td>Organics, pesticide</td>
</tr>
</tbody>
</table>
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>startDateTime</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_dt 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).

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>variableInfo</td>
<td>data frame</td>
<td>A data frame containing information on the requested parameters</td>
</tr>
</tbody>
</table>

See Also

readWQPdata, whatWQPsites, readWQPqw, constructNWISURL

Examples

```r
site_ids <- c('04024430', '04024000')
startDate <- '2010-01-01'
endDate <- ''
parameterCd <- c('34247', '30234', '32104', '34220')

rawNWISqwData <- readNWISqw(site_ids, parameterCd, startDate, endDate)
rawNWISqwDataReshaped <- readNWISqw(site_ids, parameterCd, startDate, endDate, reshape=TRUE)
```
readNWISrating <- function() {
  parameterCd <- "all"
  rawNWISall <- readNWISqw(site_ids, parameterCd, startDate, endDate)
  pgroup <- c("NutB")
  rawNWISNutrients <- readNWISqw(site_ids, pgroup, startDate, endDate)
  groups <- c("Nut", "OpeB")
  rawNWISNutOpe <- readNWISqw(site_ids, groups, startDate, endDate)
  rawNWISOpe <- readNWISqw(site_ids, "Ope", startDate, endDate)
}

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 CORRINDEXDEP, 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>
</tbody>
</table>
readNWISsite

Description

Imports data from USGS site file site. This function gets data from here: https://waterservices.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>
</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")
```
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>
**readNWISstat**

**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.
- **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 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, importRDB

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 for more information. All available use categories for the supplied arguments are retrieved.
readNWISuse

Usage

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 code NULL, 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
pData <- 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**

**Usage**

```r
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](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](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:
readNWISu

<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</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>POSIXct</td>
<td>The time the data was returned</td>
</tr>
</tbody>
</table>

See Also

renameNWISColumns, importWaterML1

Examples

```r
site_id <- '05114000'
parameterCd <- '00060'
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, because it allows for other agencies rather than the USGS.

Usage

readWQPdata(..., querySummary = FALSE, tz = "UTC")

Arguments

... see 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.

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</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 matrix as a subdivision of the sample media.</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 is reported when the field activity began, based on a 24-hour timescale.</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>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 is reported when the field activity ended, based on a 24-hour timescale.</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>ActivityDepthHeightMeasure/MeasureValue</td>
<td>character</td>
<td>A measurement of the vertical location (measured from a reference point) at which an activity occurred.</td>
</tr>
<tr>
<td>ActivityDepthHeightMeasure/MeasureUnitCode</td>
<td>character</td>
<td>The code that represents the unit for measuring the item.</td>
</tr>
</tbody>
</table>
* = 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>
<tr>
<td>variableInfo</td>
<td>data.frame</td>
<td>A data frame containing information on the requested parameters</td>
</tr>
<tr>
<td>queryTime</td>
<td>POSIXct</td>
<td>The time the data was returned</td>
</tr>
</tbody>
</table>

**Examples**

```r
ameToUse <- "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")
nutrientDaneCounty <- readWQPdata(countycode="US:55:025",startDate=startDate,
                                  characteristicType="Nutrient")
secchi.names = c("Depth, Secchi disk depth",
                 "Depth, Secchi disk depth (choice list)",
                 "Secchi Reading Condition (choice list)",
                 "Secchi depth",
                 "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)
```

querySummary=TRUE)

# querying by county
dailyLexingtonVA <- readWQPdata(statecode = "Virginia",
    countycode="Lexington",
    parameterCd = "08010")

# Biological data:
bioData <- readWQPdata(statecode = "WI",
    countycode = "Dane",
    providers = "BIODATA")

---

**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.
- `parameterCd` vector of USGS 5-digit parameter code or characteristicNames. Leaving this blank will return all of the measured values during the specified time period.
- `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.
- `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", "America/Phoenix", "America/Phoenix", "America/Phoenix", "America/Phoenix", "America/Phoenix".
and "America/Metlakatla". See also OlsonNames() for more information on time zones.

querySummary logical to look at number of records and unique sites that will be returned from this query.

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>OrganizationIdentifier</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 activity occurs.</td>
</tr>
<tr>
<td>ActivityMediaSubdivisionName</td>
<td>character</td>
<td>Name or code indicating the environmental matrix as a subdivision of the sample media.</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/TimeZoneCode</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/TimeZoneCode</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 of measurement for the vertical location.</td>
</tr>
<tr>
<td>ActivityDepthAltitudeReferencePointText</td>
<td>character</td>
<td>The reference used to indicate the datum or reference used in the vertical location.</td>
</tr>
<tr>
<td>ActivityTopDepthHeightMeasure/MeasureValue</td>
<td>character</td>
<td>A measurement of the upper vertical location of a vertical feature.</td>
</tr>
<tr>
<td>ActivityTopDepthHeightMeasure/MeasureUnitCode</td>
<td>character</td>
<td>The code that represents the unit of measurement for the upper vertical location.</td>
</tr>
<tr>
<td>ActivityBottomDepthHeightMeasure/MeasureValue</td>
<td>character</td>
<td>A measurement of the lower vertical location of a vertical feature.</td>
</tr>
<tr>
<td>ActivityBottomDepthHeightMeasure/MeasureUnitCode</td>
<td>character</td>
<td>The code that represents the unit of measurement for the lower vertical location.</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 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 defined 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>ResultDetectionConditionText</td>
<td>character</td>
<td>The textual descriptor of a result.</td>
</tr>
<tr>
<td>CharacteristicName</td>
<td>character</td>
<td>The object, property, or substance which is evaluated or enumerated.</td>
</tr>
<tr>
<td>ResultSampleFractionText</td>
<td>character</td>
<td>The text name of the portion of the sample associated with the result.</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 result.</td>
</tr>
<tr>
<td>ResultMeasure/MeasureUnitCode</td>
<td>character</td>
<td>The code that represents the unit for measuring the result.</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 calculation of the result value.</td>
</tr>
</tbody>
</table>
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.
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 or 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 the laboratory 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 was used to show the sample was 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 began.
ActivityStartDateTime POSIXct Activity start date and time.
ActivityEndDateTime POSIXct Activity end date and time.

* = 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

See Also

readWQPdata, whatWQPsites, readNWISqw, and importWQP

Examples

rawPcode <- readWQPqw('USGS-01594440', '01075', '', '')
rawCharacteristicName <- readWQPqw('WIDNR_WQX-10032762', 'Specific conductance', '', '')
rawPHsites <- readWQPqw(c('USGS-05406450', 'USGS-05427949', 'WIDNR_WQX-133040'), 'pH', '', '')
wisEx <- readWQPqw('USGS-04024000', c('34247', '30234', '32104', '34220'), '', '2012-12-20')
wisEx.summary <- readWQPqw('USGS-04024000', c('34247', '30234', '32104', '34220'),)
renameNWISColumns


Arguments

- \texttt{rawData} the daily- or unit-values dataset retrieved from NWISweb.
- \texttt{p00010} the base name for parameter code 00010.
- \texttt{p00045} the base name for parameter code 00045.
- \texttt{p00060} the base name for parameter code 00060.
- \texttt{p00065} the base name for parameter code 00065.
- \texttt{p00095} the base name for parameter code 00095.
- \texttt{p00300} the base name for parameter code 00300.
- \texttt{p00400} the base name for parameter code 00400.
- \texttt{p62611} the base name for parameter code 62611.
- \texttt{p63680} the base name for parameter code 63680.
- \texttt{p72019} the base name for parameter code 72019.
- \ldots \text{ 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 \texttt{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 data retrieval access code you want to use: options: \c('public','internal')

**Usage**

`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

```r
setAccess('internal')
setAccess('public')
```

---

**stateCd** *US State Code Lookup Table*

**Description**

Data pulled from [https://www2.census.gov/geo/docs/reference/state.txt](https://www2.census.gov/geo/docs/reference/state.txt) on April 1, 2015.

**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**

```r
head(stateCd)
```
**stateCdLookup**  
*State code lookup*

**Description**

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

**Usage**

```r
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**

```r
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**

```r
whatNWISdata(...)
```

**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.

**Value**

A data frame with the following columns:
**Name**  | **Type**  | **Description**  
---|---|---  
agency_cd | character | The NWIS code for the agency reporting the data  
site_no | character | The USGS site number  
station_nm | character | Site name  
site_tp_cd | character | Site type  
dec_lat_va | numeric | Decimal latitude  
dec_long_va | numeric | Decimal longitude  
coord_acy_cd | character | Latitude-longitude accuracy  
dec_coord_datum_cd | character | Decimal Latitude-longitude datum  
alta_va | character | Altitude of Gage or land surface  
alta_acy_va | character | Altitude accuracy  
alta_datum_cd | character | Altitude datum  
huc_cd | character | Hydrologic unit code  
data_type_cd | character | Data type  
parm_cd | character | Parameter code  
stat_cd | character | Statistical code  
dd_nu | character | Internal database key  
loc_web_ds | character | Additional measurement description  
medium_grp_cd | character | Medium group code  
parm_grp_cd | character | Parameter group code  
srs_id | character | SRS ID  
access_cd | character | Access code  
begin_date | Date | Begin date  
end_date | Date | End date  
count_nu | integer | Record count  
parameter_group_nm | character | Parameter group name  
parm_nm | character | Parameter name  
casrn | character | Chemical Abstracts Service (CAS) Registry Number  
srsname | character | Substance Registry Services  
parameter_units | character | Parameter units  

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**

```r
availableData <- 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"))
flowAndTemp <- whatNWISdata(stateCd = "WI", service = "uv",
                          parameterCd = c("00060","00010"),
```
whatNWISsites

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. Mapper format is used.

Usage

whatNWISsites(...)

Arguments

... see 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>

Examples

sitelistPhos <- whatNWISsites(stateCd="OH",parameterCd="00665")
oneSite <- whatNWISsites(sites="85114000")
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>The designator specified by the sampling organization for the site at which monitoring took place.</td>
</tr>
<tr>
<td>&quot;MonitoringLocationName&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 descriptive name for a type of 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>
<tr>
<td>&quot;resultCount&quot;</td>
<td>numeric</td>
<td></td>
</tr>
<tr>
<td>&quot;StateName&quot;</td>
<td>character</td>
<td>State name</td>
</tr>
<tr>
<td>&quot;CountyName&quot;</td>
<td>character</td>
<td>County name</td>
</tr>
</tbody>
</table>

See Also

whatNWISsites
whatWQPsamples

Examples

```r
site1 <- whatWQPdata(siteid="USGS-01594440")

type <- "Stream"
sites <- whatWQPdata(countycode="US:55:025",siteType=type)

lakeSites <- whatWQPdata(siteType = "Lake, Reservoir, Impoundment", statecode = "US:55")
```

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

```r
whatWQPsamples(...)
whatWQPmetrics(...)
whatWQPsites(...)
readWQPsummary(...)
```

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</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</td>
</tr>
</tbody>
</table>
### whatWQPsamples

- **MonitoringLocationName** character: The designator specified by the sampling organization for the site at which sampling or other activities are conducted.
- **MonitoringLocationTypeName** character: The descriptive name for a type of monitoring location.
- **MonitoringLocationDescriptionText** character: Text description of the monitoring location.
- **HUCEightDigitCode** character: The 8 digit federal code used to identify the hydrologic unit of the monitoring location to the cataloging unit level of precision.
- **DrainageAreaMeasure/MeasureValue** * character: The drainage basin of a lake, stream, wetland, or estuary site. Measure value is given in the units stored in DrainageAreaMeasure/MeasureUnitCode.
- **DrainageAreaMeasure/MeasureUnitCode** * character: The code that represents the unit for measuring the item.
- **ContributingDrainageAreaMeasure/MeasureValue** * character: The contributing drainage area of a lake, stream, wetland, or estuary site. Measure value is given in the units stored in ContributingDrainageAreaMeasure/MeasureUnitCode.
- **ContributingDrainageAreaMeasure/MeasureUnitCode** * character: The code that represents the unit for measuring the item.
- **LatitudeMeasure** numeric: The measure of the angular distance on a meridian north or south of the equator.
- **LongitudeMeasure** numeric: The measure of the angular distance on a meridian east or west of the prime meridian.
- **SourceMapScaleNumeric** character: The number that represents the proportional distance on the ground for one unit of measure on the map or photo.
- **HorizontalAccuracyMeasure/MeasureValue** * character: The horizontal measure of the relative accuracy of the latitude and longitude coordinates. Measure value is given in the units stored in HorizontalAccuracyMeasure/MeasureUnitCode.
- **HorizontalAccuracyMeasure/MeasureUnitCode** * character: The code that represents the unit for measuring the item.
- **HorizontalCollectionMethodName** character: The name that identifies the method used to determine the latitude and longitude coordinates for a point on the earth.
- **HorizontalCoordinateReferenceSystemDatumName** character: The name that describes the reference datum used in determining latitude and longitude coordinates.
- **VerticalMeasure/MeasureValue** character: The measure of elevation (i.e., the altitude), above or below a reference datum. Measure value is given in the units stored in VerticalMeasure/MeasureUnitCode.
- **VerticalMeasure/MeasureUnitCode** character: The code that represents the unit for measuring the item.
- **VerticalAccuracyMeasure/MeasureValue** * character: The vertical measure of the relative accuracy of the latitude and longitude coordinates. Measure value is given in the units stored in VerticalAccuracyMeasure/MeasureUnitCode.
- **VerticalAccuracyMeasure/MeasureUnitCode** * character: The code that represents the unit for measuring the item.
- **VerticalCollectionMethodName** character: The name that identifies the method used to collect the vertical measure (i.e., the altitude).
- **VerticalCoordinateReferenceSystemDatumName** character: The name of the reference datum used to determine the vertical measure (i.e., the altitude).
- **CountryCode** character: A code designator used to identify a primary geopolitical unit of the world.
- **StateCode** character: A code designator used to identify a principal administrative subdivision of the United States, Canada, or Mexico.
- **CountyCode** character: A code designator used to identify a U.S. county or county equivalent.
- **AquiferName** * character: Name of the aquifer in which the well is completed.
- **FormationTypeText** * character: Name of the primary formation or soils unit, in which the well is completed.
- **AquiferTypeName** * character: The type of aquifer, such as confined or unconfined.
- **ConstructionDateText** * character: Date of construction when well was completed. May be year only.
- **WellDepthMeasure/MeasureValue** * character: Depth below land surface datum (LSD) to the bottom of the hole on completion of drilling. Measure value is given in the units stored in WellDepthMeasure/MeasureUnitCode.
- **WellDepthMeasure/MeasureUnitCode** * character: The code that represents the unit for measuring the item.
- **WellHoleDepthMeasure/MeasureValue** * character: Depth below land surface datum (LSD) to the bottom of the hole on completion of drilling. Measure value is given in the units stored in WellHoleDepthMeasure/MeasureUnitCode.
- **WellHoleDepthMeasure/MeasureUnitCode** * character: The code that represents the unit for measuring the item.
- **queryTime** POSIXct: Query time

* 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)
```
zeroPad

**Description**

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

**Usage**

```r
zeroPad(x, padTo)
```

**Arguments**

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

**Value**

- `x` character returned with leading zeros

**Examples**

```r
code <- '10'
correctCode <- zeroPad(code, 5)
codes <- c('100','1000','0','12345','1565465465465465')
correctCodes <- zeroPad(codes, 5)
codeNA <- c(1,2,NA)
padCodeNA <- zeroPad(codeNA, 4)
```
Index

*Topic **IO**
  renameNWISColumns, 49

*Topic **USGS**
  constructNWISURL, 4
  countyCd, 8
  readNWISdv, 24
  readNWISpCode, 29
  readNWISqw, 31
  readNWISsite, 35
  readNWISuv, 40
  readWQPqw, 46
  stateCd, 51
  whatNWISdata, 52
  zeroPad, 58

*Topic **WQP**
  constructWQPURL, 7
  readWQPdata, 43
  whatWQPdata, 55
  whatWQPsamples, 56

*Topic **countyCd**
  countyCd, 8

*Topic **data**
  constructNWISURL, 4
  constructWQPURL, 7
  readNWISdv, 24
  readNWISpCode, 29
  readNWISqw, 31
  readNWISsite, 35
  readNWISuv, 40
  readWQPdata, 43
  readWQPqw, 46
  whatNWISdata, 52
  whatWQPdata, 55
  whatWQPsamples, 56
  zeroPad, 58

*Topic **import**
  constructNWISURL, 4
  constructWQPURL, 7
  readNWISdv, 24
  readNWISpCode, 29
  readNWISqw, 31
  readNWISsite, 35
  readNWISuv, 40
  readWQPdata, 43
  readWQPqw, 46
  whatNWISdata, 52

*Topic **manip**
  renameNWISColumns, 49

*Topic **service**
  constructNWISURL, 4
  constructWQPURL, 7
  readNWISdv, 24
  readNWISpCode, 29
  readNWISqw, 31
  readNWISsite, 35
  readNWISuv, 40
  readWQPdata, 43
  readWQPqw, 46
  whatNWISdata, 52
  whatWQPdata, 55
  whatWQPsamples, 56
  zeroPad, 58

*Topic **stateCd**
  stateCd, 51

*Topic **web**
  constructNWISURL, 4
  constructWQPURL, 7
  readNWISdv, 24
  readNWISpCode, 29
  readNWISqw, 31
  readNWISsite, 35
  readNWISuv, 40
  readWQPdata, 43
  readWQPqw, 46
  whatNWISdata, 52
whatWQPdata, 55
whatWQPsamples, 56
zeroPad, 58
addWaterYear, 3
calcWaterYear, 4
constructNWISURL, 4, 27, 28, 31, 33, 35, 38
constructUseURL, 6
constructWQPURL, 7
countyCd, 8
countyCdlookup, 8
dataRetrieval, 9
dataRetrieval-package (dataRetrieval), 9

GET, 10
getQuerySummary, 9
getWebServiceData, 10

importNGWMN, 10
importRDBI, 11, 22, 27–29, 31, 35, 38
importWaterML1, 13, 22, 25, 42
importWaterML2, 15
importWQP, 16, 48

parameterCdFile, 17

readNGWMNdata, 18
readNGWMNlevels, 19
readNGWMNsites, 20
readNWISdata, 21
readNWISdv, 24, 50
readNWISgwl, 26
readNWISmeas, 27
readNWISpCode, 29
readNWISpeak, 30
readNWISqw, 31, 48
readNWISrating, 34
readNWISsite, 35
readNWISstat, 37
readNWISuse, 38
readNWISuv, 40, 50
readWQPdata, 17, 33, 43, 48
readWQPqw, 17, 33, 46
readWQPsummary (whatWQPsamples), 56
renameNWISCcolumns, 14, 22, 25, 42, 49

setAccess, 50
stateCd, 51

stateCdLookup, 52
whatNWISdata, 52
whatNWISSites, 54
whatWQPdata, 55, 56
whatWQPmetrics (whatWQPsamples), 56
whatWQPsamples, 56
whatWQPsites, 17, 33, 48
whatWQPsites (whatWQPsamples), 56
zeroPad, 58