Package 'dataRetrieval'

March 11, 2020

Type Package

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

Version 2.7.6

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 7.0.2

NeedsCompilation no

Author Laura DeCicco [aut, cre] (<https://orcid.org/0000-0002-3915-9487>),
Robert Hirsch [aut] (<https://orcid.org/0000-0002-4534-075X>),
David Lorenz [aut],
Jordan Read [ctb],
Jordan Walker [ctb],

Lindsay Carr [ctb],
David Watkins [aut] (<https://orcid.org/0000-0002-7544-0700>)

Maintainer Laura DeCicco <ldecicco@usgs.gov>

Repository CRAN

Date/Publication 2020-03-11 06:30:13 UTC

R topics documented:

addWaterYear 3
calcWaterYear 4
constructNWISURL 4
constructUseURL 6
constructWQPURL 7
countyCd 8
countyCdLookup 8
dataRetrieval 9
getQuerySummary 9
getWebServiceData 10
importNGWMN 11
importRDB1 12
importWaterML1 13
importWaterML2 16
importWQP 17
parameterCdFile 18
readNGWMNdata 18
readNGWMNlevels 19
readNGWMNsites 20
readNWISdata 21
readNWISdv 24
readNWISgwl 26
readNWISmeas 28
readNWISpCode 30
readNWISpeak 31
readNWISqw 33
readNWISrating 36
readNWISsite 37
readNWISstat 38
readNWISuse 40
readNWISuv 42
readWQPdata 44
readWQPqw 47
renameNWISColumns 50
setAccess 52
stateCd 52
stateCdLookup 53
whatNWISdata 54
whatNWISsites 55

whatWQPdata 56
whatWQPsamples 57
zeroPad 60

Index **61**

addWaterYear *add a water year column*

Description

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

Usage

```
addWaterYear(rawData)
```

Arguments

rawData	the daily- or unit-values datset 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

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

wqpData <- readWQPqw('USGS-01594440','01075', '', '')
wqpData <- addWaterYear(wqpData)
```

calcWaterYear *Extract WY from a date*

Description

Determine the correct water year based on a calendar date.

Usage

```
calcWaterYear(dateVec)
```

Arguments

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

Details

This function calculates a water year based on the USGS definition that a water year starts on October 1 of the year before, and ends on September 30. For example, water year 2015 started on 2014-10-01 and ended on 2015-09-30. See the USGS definition at https://water.usgs.gov/nwc/explain_data.html.

Value

numeric vector indicating the water year

Examples

```
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 A list of parameter codes can be found here: https://nwis.waterdata.usgs.gov/nwis/pmcodes/ A list of statistic codes can be found here: https://nwis.waterdata.usgs.gov/nwis/help/?read_file=stat&format=table

Usage

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

Arguments

`siteNumbers` string or vector of strings USGS site number. This is usually an 8 digit number

`parameterCd` string or vector of USGS parameter code. This is usually an 5 digit number.

`startDate` character starting date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the earliest possible record.

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

`service` string USGS service to call. Possible values are "dv" (daily values), "uv" (unit/instantaneous values), "qw" (water quality data), "gwlevels" (groundwater),and "rating" (rating curve), "peak", "meas" (discrete streamflow measurements), "stat" (statistics web service BETA).

`statCd` string or vector USGS statistic code only used for daily value service. This is usually 5 digits. Daily mean (00003) is the default.

`format` string, can be "tsv" or "xml", and is only applicable for daily and unit value requests. "tsv" returns results faster, but there is a possiblitiy that an incomplete file is returned without warning. XML is slower, but will offer a warning if the file was incomplete (for example, if there was a momentary problem with the internet connection). It is possible to safely use the "tsv" option, but the user must carefully check the results to see if the data returns matches what is expected. The default is therefore "xml".

`expanded` logical defaults to `TRUE`. If `TRUE`, retrieves additional information, only applicable for qw data.

`ratingType` can be "base", "corr", or "exsa". Only applies to rating curve data.

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

`statType` 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 https://waterservices.usgs.gov/rest/Statistics-Service.html for a full list of codes.

Value

url string

Examples

```
site_id <- '01594440'
startDate <- '1985-01-01'
endDate <- ''
pCode <- c("00060","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("450456092225801","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

Usage

```
constructUseURL(years, stateCd, countyCd, categories)
```

Arguments

`years` integer Years for data retrieval. Must be years ending in 0 or 5, or "ALL", which retrieves all available years.

`stateCd` could be character (full name, abbreviation, id), or numeric (id)

`countyCd` could be numeric (County IDs from countyCdLookup) or character ("ALL")

`categories` character Two-letter cateogory abbreviation(s)

Value

url string

Examples

```
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

```
site_id <- '01594440'
startDate <- '1985-01-01'
endDate <- ''
pCode <- c("00060","00010")
url_wqp <- constructWQPURL(paste("USGS",site_id,sep="-"),
           c('01075','00029','00453'),
           startDate,endDate)
url_wqp
charNames <- c("Temperature",
               "Temperature, sample",
               "Temperature, water",
                "Temperature, water, deg F")
obs_url_orig <- constructWQPURL(siteNumbers = c("IIDFG-41WSSPAHS",
                                                "USGS-02352560"),
                                parameterCd = charNames,
                                startDate,"")
obs_url_orig
```

countyCd *US County Code Lookup Table*

Description

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

Value

countyCd data frame.

Name	Type	Description
STUSAB	character	State abbreviation
STATE	character	two-digit ANSI code
COUNTY	character	three-digit county code
COUNTY_NAME	character	County full name
COUNTY_ID	character	County id

Examples

```
head(countyCd)
```

countyCdLookup *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

```
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 G
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`>

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

Description

getting header information from a WQP query

Usage

```
getQuerySummary(url)
```

Arguments

url	the query url

---

getWebServiceData *Function to return data from web services*

Description

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

Usage

```
getWebServiceData(obs_url, ...)
```

Arguments

obs_url	character containing the url for the retrieval
...	information to pass to header request

Value

raw data from web services

Examples

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

rawData <- getWebServiceData(obs_url)
```

`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”)
```

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

```
obs_url <- paste(”https://cida.usgs.gov/ngwmn_cache/sos?request=GetObservation”,
”service=SOS”,”version=2.0.0”,
”observedProperty=urn:ogc:def:property:OGC:GroundWaterLevel”,
”responseFormat=text/xml”,
”featureOfInterest=VW_GWDP_GEOSERVER.USGS.403836085374401”,sep=”&”)
data <- importNGWMN(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")
```

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/Jama
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:

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

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:

Name	Type	Description
url	character	The url used to generate the data
queryTime	POSIXct	The time the data was returned
comment	character	Header comments from the RDB file

Examples

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

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

data <- importRDB1(obs_url)

urlMultiPcodes <- constructNWISURL("04085427",c("00060","00010"),
         startDate,endDate,"dv",statCd=c("00003","00001"),"tsv")
multiData <- importRDB1(urlMultiPcodes)
unitDataURL <- constructNWISURL(site_id,property,
         "2013-11-03","2013-11-03","uv",format="tsv") #includes timezone switch
unitData <- importRDB1(unitDataURL, asDateTime=TRUE)
qwURL <- constructNWISURL(c('04024430','04024000'),
          c('34247','30234','32104','34220'),
         "2010-11-03","","qw",format="rdb")
qwData <- importRDB1(qwURL, asDateTime=TRUE, tz="America/Chicago")
iceSite <- '04024000'
start <- "2015-11-09"
end <- "2015-11-24"
urlIce <- constructNWISURL(iceSite,"00060",start, end,"uv",format="tsv")
ice <- importRDB1(urlIce, asDateTime=TRUE)
iceNoConvert <- importRDB1(urlIce, convertType=FALSE)

# User file:
filePath <- system.file("extdata", package="dataRetrieval")
fileName <- "RDB1Example.txt"
fullPath <- file.path(filePath, fileName)
importUserRDB <- importRDB1(fullPath)
```

importWaterML1 *Function to return data from the NWISWeb WaterML1.1 service*

Description

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

Usage

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

Arguments

`obs_url` character or raw, containing the url for the retrieval or a file path to the data file, or raw XML.

`asDateTime` logical, if `TRUE` returns date and time as POSIXct, if `FALSE`, Date

`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/Jama
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:

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

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:

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
statisticInfo	data.frame	A data frame containing information on the requested statistics on the data
queryTime	POSIXct	The time the data was returned

See Also

renameNWISColumns

Examples

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

data <- importWaterML1(obs_url, asDateTime=TRUE)

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

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

# Two sites, two pcodes, one site has two data descriptors:
siteNumber <- c('01480015',"04085427")
obs_url <- constructNWISURL(siteNumber,c("00060","00010"),startDate,endDate,'dv')
data <- importWaterML1(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, "00060", "2014-01-01", "2014-01-10",'dv')
inactiveSite <- importWaterML1(inactiveSite)

inactiveAndAcitive <- c("07334200","05212700")
inactiveAndAcitive <- constructNWISURL(inactiveAndAcitive, "00060", "2014-01-01", "2014-01-10",'dv')
inactiveAndAcitive <- importWaterML1(inactiveAndAcitive)

# Timezone change with specified local timezone:
tzURL <- constructNWISURL("04027000", c("00300","63680"), "2011-11-05", "2011-11-07","uv")
tzIssue <- importWaterML1(tzURL, TRUE, "America/Chicago")

# raw XML
url <- constructNWISURL(service = 'dv', siteNumber = '02319300', parameterCd = "00060",
                         startDate = "2014-01-01", endDate = "2014-01-01")
raw <- httr::content(httr::GET(url), as = 'raw')
```

```
rawParsed <- importWaterML1(raw)

filePath <- system.file("extdata", package="dataRetrieval")
fileName <- "WaterML1Example.xml"
fullPath <- file.path(filePath, fileName)
importFile <- importWaterML1(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=waterml,2.0"
URL <- paste(baseURL, "sites=01646500",
     "startDT=2014-09-01",
     "endDT=2014-09-08",
     "statCd=00003",
     "parameterCd=00060",sep="&")

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

importWQP *Basic Water Quality Portal Data parser*

Description

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

Usage

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

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_An "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.

Value

parameterData data frame with information about USGS parameters.

Name	Type	Description
parameter_cd	character	5-digit USGS parameter code
parameter_group_nm	character	USGS parameter group name
parameter_nm	character	USGS parameter name
casrn	character	Chemical Abstracts Service (CAS) Registry Number
srsname	character	Substance Registry Services Name
parameter_units	character	Parameter units

Examples

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

readNGWMNdata *import data from the National Groundwater Monitoring Network https://cida.usgs.gov/ngwmn/.*

Description

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

Usage

```
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/Anchorage", as well as the following which do not use daylight savings time: "America/Honolulu", "America/Jamaica","America/Managua","America/Phoenix", and "America/Metlakatla". See also `OlsonNames()` for more information on time zones.

Examples

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

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

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

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

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

readNGWMNlevels

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

Description

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

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`.

`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

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

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

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

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

`readNGWMNsites` *Retrieve site data from the National Ground Water Monitoring Network* `https://cida.usgs.gov/ngwmn/`.

Description

Retrieve site data from the National Ground Water Monitoring Network `https://cida.usgs.gov/ngwmn/`.

Usage

```
readNGWMNsites(siteNumbers)
```

Arguments

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

Value

A data frame the following columns: #'

Name	Type	Description
site	char	Site FID
description	char	Site description
dec_lat_va, dec_lon_va	numeric	Site latitude and longitude

Examples

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

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

`readNWISdata` *General Data Import from NWIS*

Description

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

Usage

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

Arguments

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

(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://nwis.waterdata.usgs.gov/usa/nwis` for data requests, and use different call requests schemes. The statistics service has a limited selection of arguments (see `https://waterservices.usgs.gov/rest/Statistics-Service-Test-Tool.html`).

`asDateTime` logical, if `TRUE` returns date and time as POSIXct, if `FALSE`, Date

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

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

Value

A data frame with the following columns:

Name	Type	Description
agency	character	The NWIS code for the agency reporting the data
site	character	The USGS site number
dateTime	POSIXct	The date and time (if applicable) of the measurement, converted to UTC for unit value data. R only al
tz_cd	character	The time zone code for dateTime column
code	character	Any codes that qualify the corresponding value
value	numeric	The numeric value for the parameter

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:

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
statisticInfo	data.frame	A data frame containing information on the requested statistics on the data
queryTime	POSIXct	The time the data was returned

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

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

#Empty:
multiSite <- readNWISdata(sites=c("04025000","04072150"), service="iv",
                           parameterCd="00010")
#Not empty:
multiSite <- readNWISdata(sites=c("04025500","040263491"),
                           service="iv", parameterCd="00060")
bBoxEx <- readNWISdata(bBox=c(-83,36.5,-81,38.5), parameterCd="00010")

startDate <- as.Date("2013-10-01")
endDate <- as.Date("2014-09-30")
waterYear <- readNWISdata(bBox=c(-83,36.5,-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",
                qw_sample_wide="wide",list_of_search_criteria=c("lat_long_bounding_box",
                  "drain_area_va","obs_count_nu"),service="qw")
temp <- readNWISdata(bBox=c(-83,36.5,-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(site=c("03112500"),service="stat",statReportType="annual",
                 statYearType="water", missingData="on")
monthlyStat <- readNWISdata(site=c("03112500","03111520"),
                            service="stat",
                            statReportType="monthly")
dailyStat <- readNWISdata(site=c("03112500","03111520"),
                          service="stat",
                          statReportType="daily",
                          statType=c("p25","p50","p75","min","max"),
                          parameterCd="00065")
allDailyStats <- readNWISdata(site=c("03111548"),
                              service="stat",
                                  statReportType="daily",
                                  statType=c("p25","p50","p75","min","max"),
                                  parameterCd="00060")

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

arg.list <- list(site="03111548",
                 statReportType="daily",
                 statType=c("p25","p50","p75","min","max"),
                 parameterCd="00060")
allDailyStats_2 <- readNWISdata(arg.list, service="stat")

#' # use county names to get data
dailyStaffordVA <- readNWISdata(stateCd = "Virginia",
                                countyCd="Stafford",
                                parameterCd = "00060",
                                startDate = "2015-01-01",
                                endDate = "2015-01-30")
va_counties <- c("51001","51003","51005","51007","51009","51011","51013","51015")
va_counties_data <- readNWISdata(startDate = "2015-01-01", endDate = "2015-12-31",
parameterCd = "00060", countycode = va_counties)
site_id <- '01594440'
rating_curve <- readNWISdata(service = "rating", site_no = site_id, file_type="base")
all_sites_base <- readNWISdata(service = "rating", file_type="base")
all_sites_core <- readNWISdata(service = "rating", file_type="corr")
all_sites_exsa <- readNWISdata(service = "rating", file_type="exsa")
all_sites_24hrs <- readNWISdata(service = "rating", file_type="exsa", period = 24)

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/

Usage

```
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:

Name	Type	Description
agency	character	The NWIS code for the agency reporting the data
site	character	The USGS site number
Date	Date	The date of the value
code	character	Any codes that qualify the corresponding value
value	numeric	The numeric value for the parameter

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:

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
statisticInfo	data.frame	A data frame containing information on the requested statistics on the data
queryTime	POSIXct	The time the data was returned

See Also

`renameNWISColumns`, `importWaterML1`

Examples

```
site_id <- '04085427'
startDate <- '2012-01-01'
endDate <- '2012-06-30'
```

```
pCode <- '00060'

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

site <- "05212700"
notActive <- readNWISdv(site, "00060", "2014-01-01","2014-01-07")
```

`readNWISgwl` *Groundwater level measurements retrieval from USGS (NWIS)*

Description

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

Usage

```
readNWISgwl(
  siteNumbers,
  startDate = "",
  endDate = "",
  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:

Name	Type	Description
agency_cd	character	The NWIS code for the agency reporting the data
site_no	character	The USGS site number
site_tp_cd	character	Site type code
lev_dt	Date	Date level measured
lev_tm	character	Time level measured
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:

Name	Type	Description
url	character	The url used to generate the data
queryTime	POSIXct	The time the data was returned
comment	character	Header comments from the RDB file
siteInfo	data.frame	A data frame containing information on the requested sites

See Also

`constructNWISURL`, `importRDB1`

Examples

```
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")
```

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

Description

Reads surface-water measurement data from NWISweb. Data is retrieved from https://waterdata.usgs.gov/nwis. See https://waterdata.usgs.gov/usa/nwis/sw for details about surface water.

Usage

```
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:

Name	Type	Description
agency_cd	character	The NWIS code for the agency reporting the data
site_no	character	The USGS site number
measurement_dt	POSIXct	The date and time (in POSIXct) of the measurement. Unless specified with the tz parameter, th
tz_cd	character	The time zone code for the measurement_dt column

See https://waterdata.usgs.gov/usa/nwis/sw for details about surface water, and https://waterdata.usgs.gov/nwis/help?output_formats_help for help on the columns and codes.

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

Name	Type	Description
url	character	The url used to generate the data
queryTime	POSIXct	The time the data was returned
comment	character	Header comments from the RDB file
siteInfo	data.frame	A data frame containing information on the requested sites
tz_cd_reported	The originally reported time zone	

See Also

constructNWISURL, importRDB1

Examples

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

Usage

```
readNWISpCode(parameterCd)
```

Arguments

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

Value

parameterData data frame with the following information:

Name	Type	Description
parameter_cd	character	5-digit USGS parameter code
parameter_group_nm	character	USGS parameter group name
parameter_nm	character	USGS parameter name
casrn	character	Chemical Abstracts Service (CAS) Registry Number
srsname	character	Substance Registry Services Name
parameter_units	character	Parameter units

See Also

importRDB1

Examples

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

readNWISpeak *Peak flow data from USGS (NWIS)*

Description

Reads peak flow from NWISweb. Data is retrieved from https://waterdata.usgs.gov/nwis. In some cases, the specific date of the peak data is not know. 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

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

Arguments

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

Value

A data frame with the following columns:

Name	Type	Description
agency_cd	character	The NWIS code for the agency reporting the data
site_no	character	The USGS site number
peak_dt	Date	Date of peak streamflow
peak_tm	character	Time of peak streamflow as character
peak_va	numeric	Annual peak streamflow value in cfs
peak_cd	character	Peak Discharge-Qualification codes (see `comment` for more information)
gage_ht	numeric	Gage height for the associated peak streamflow in feet
gage_ht_cd	character	Gage height qualification codes
year_last_pk	numeric	Peak streamflow reported is the highest since this year
ag_dt	Date	Date of maximum gage-height for water year (if not concurrent with peak)
ag_tm	character	Time of maximum gage-height for water year (if not concurrent with peak)
ag_gage_ht	numeric	maximum Gage height for water year in feet (if not concurrent with peak)
ag_gage_ht_cd	character	maximum Gage height code

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

Name	Type	Description
url	character	The url used to generate the data
queryTime	POSIXct	The time the data was returned
comment	character	Header comments from the RDB file
siteInfo	data.frame	A data frame containing information on the requested sites

See Also

`constructNWISURL`, `importRDB1`

Examples

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

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

`readNWISqw` *Raw Data Import for USGS NWIS QW Data*

Description

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

Usage

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

Arguments

`siteNumbers` character of USGS site numbers. This is usually an 8 digit number

`parameterCd` character that contains the code for a parameter group, or a character vector of 5-digit parameter codes. See **Details**.

`startDate` character starting date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the earliest possible record. Date arguments are always specified in local time.

`endDate` character ending date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the latest possible record. Date arguments are always specified in local time.

`expanded` logical defaults to `TRUE`. If `TRUE`, retrieves additional information. Expanded data includes remark_cd (remark code), result_va (result value), val_qual_tx (result value qualifier code), meth_cd (method code), dqi_cd (data-quality indicator code), rpt_lev_va (reporting level), and rpt_lev_cd (reporting level type).

If `FALSE`, only returns remark_cd (remark code) and result_va (result value). Expanded = `FALSE` will not give sufficient information for unbiased statistical analysis.

`reshape` logical, reshape the expanded data. If `TRUE`, then return a wide data frame with all water-quality in a single row for each sample. If `FALSE` (default), then return a long data frame with each water-quality result in a single row. This argument is only applicable to expanded data. Data requested using `expanded=FALSE` is always returned in the wide format.

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

Details

Valid parameter code groups are "All," or group codes:

Code	Description
INF	Information
PHY	Physical
INM	Inorganics, Major, Metals (major cations)
INN	Inorganics, Major, Non-metals (major anions)
NUT	Nutrient
MBI	Microbiological
BIO	Biological
IMN	Inorganics, Minor, Non-metals
IMM	Inorganics, Minor, Metals
TOX	Toxicity
OPE	Organics, pesticide
OPC	Organics, PCBs
OOT	Organics, other
RAD	Radiochemical
SED	Sediment
POP	Population/community

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:

Name	Type	Description

agency_cd	character	The NWIS code for the agency reporting the data
site_no	character	The USGS site number
sample_dt	Date	The date the sample was collected
sample_tm	character	The reported sample collection time
startDateTime	POSIXct	Combining sample_dt and sample_tm, a date/time column is created, and converted into UTC (u
endDateTime	POSIXct	If any sample_end_dt and sample_end_dt exist, this column is created similar to startDateTime

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:

Name	Type	Description
url	character	The url used to generate the data
queryTime	POSIXct	The time the data was returned
comment	character	Header comments from the RDB file
siteInfo	data frame	A data frame containing information on the requested sites
variableInfo	data frame	A data frame containing information on the requested parameters

See Also

readWQPdata, whatWQPsites, readWQPqw, constructNWISURL

Examples

```
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)
parameterCd <- "all"
rawNWISall <- readNWISqw(site_ids,parameterCd,
          startDate,endDate)
pgroup <- c("NUT")
rawNWISNutrients <- readNWISqw(site_ids,pgroup,
          startDate,endDate)
groups <- c("NUT","OPE")
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 CORRINDEP, the corrected value for CORR.
If type is "base," then the data frame has an attribute called "RATING" that describes the rating curve is included.

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

Name	Type	Description
url	character	The url used to generate the data
queryTime	POSIXct	The time the data was returned
comment	character	Header comments from the RDB file
siteInfo	data.frame	A data frame containing information on the requested sites
RATING	character	Rating information

Note

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

See Also

constructNWISURL, importRDB1

Examples

```
site_id <- '01594440'

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

---

readNWISsite *USGS Site File Data Retrieval*

Description

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

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:

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
lat_va	numeric	DMS latitude
long_va	numeric	DMS longitude
dec_lat_va	numeric	Decimal latitude
dec_long_va	numeric	Decimal longitude
coord_meth_cd	character	Latitude-longitude method
coord_acy_cd	character	Latitude-longitude accuracy
coord_datum_cd	character	Latitude-longitude datum
dec_coord_datum_cd	character	Decimal Latitude-longitude datum
district_cd	character	District code
state_cd	character	State code
county_cd	character	County code
country_cd	character	Country code
land_net_ds	character	Land net location description
map_nm	character	Name of location map
map_scale_fc	character	Scale of location map
alt_va	numeric	Altitude of Gage/land surface

alt_meth_cd	character	Method altitude determined
alt_acy_va	numeric	Altitude accuracy
alt_datum_cd	character	Altitude datum
huc_cd	character	Hydrologic unit code
basin_cd	character	Drainage basin code
topo_cd	character	Topographic setting code
instruments_cd	character	Flags for instruments at site
construction_dt	character	Date of first construction
inventory_dt	character	Date site established or inventoried
drain_area_va	numeric	Drainage area
contrib_drain_area_va	numeric	Contributing drainage area
tz_cd	character	Time Zone abbreviation
local_time_fg	character	Site honors Daylight Savings Time
reliability_cd	character	Data reliability code
gw_file_cd	character	Data-other GW files
nat_aqfr_cd	character	National aquifer code
aqfr_cd	character	Local aquifer code
aqfr_type_cd	character	Local aquifer type code
well_depth_va	numeric	Well depth
hole_depth_va	numeric	Hole depth
depth_src_cd	character	Source of depth data
project_no	character	Project number

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

Name	Type	Description
url	character	The url used to generate the data
queryTime	POSIXct	The time the data was returned
comment	character	Header comments from the RDB file

Examples

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

readNWISstat *Site statistics retrieval from USGS (NWIS)*

Description

Retrieves site statistics from the USGS Statistics Web Service beta. See https://waterservices.usgs.gov/rest/Statistics-Service.html for more information.

Usage

```
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 appliccable) 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 or readNWISdv for daily averages. Also note that 'annual' returns statistics for the calendar year. Use readNWISdata for water years. Monthly and yearly provide statistics for each month and year within the range indivually.

`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:

Name	Type	Description
agency_cd	character	The NWIS code for the agency repor
site_no	character	The USGS site number
parameter_cd	character	The USGS parameter code
Other columns will be present depending on statReportType and statType		

See Also

constructNWISURL, importRDB1

Examples

```
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.

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 codeNULL, which will return state or national data depending on the stateCd argument. `ALL` may also be supplied, which will return data for every county in a state. Can be a vector of counties in the same state.
- `years` integer Years for data retrieval. Must be years ending in 0 or 5. Default is all available years.
- `categories` character categories of water use. Defaults to `ALL`. Specific categories must be supplied as two- letter abbreviations as seen in the URL when using the NWIS water use web interface. Note that there are different codes for national and state level data.
- `convertType` logical defaults to `TRUE`. If `TRUE`, the function will convert the data to numerics based on a standard algorithm. Years, months, and days (if appliccable) are also returned as numerics in separate columns. If convertType is false, everything is returned as a character.
- `transform` logical only intended for use with national data. Defaults to `FALSE`, with data being returned as presented by the web service. If `TRUE`, data will be transformed and returned with column names, which will reformat national data to be similar to state data.

Value

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

Examples

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

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

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

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

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

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

#retrieving two specific categories for an entire state
```

```
ks <- readNWISuse(stateCd = "KS", countyCd = NULL, categories = c("IT","LI"))
```

readNWISuv *Instantaneous value data retrieval from USGS (NWIS)*

Description

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

Usage

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

Arguments

siteNumbers	character USGS site number (or multiple sites). This is usually an 8 digit number
parameterCd	character USGS parameter code. This is usually an 5 digit number.
startDate	character starting date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the earliest possible record. Simple date arguments are specified in local time. See more information here: https://waterservices.usgs.gov/rest/IV-Service.html.
endDate	character ending date for data retrieval in the form YYYY-MM-DD. Default is "" which indicates retrieval for the latest possible record. Simple date arguments are specified in local time. See more information here: https://waterservices.usgs.gov/rest/IV-Service.html.
tz	character to set timezone attribute of dateTime. Default is "UTC", and converts the date times to UTC, properly accounting for daylight savings times based on the data's provided tz_cd column. Possible values to provide are "America/New_York","America/Chicago", "America/Denver","America/Los_Angeles", "America/Anchorage", as well as the following which do not use daylight savings time: "America/Honolulu", "America/Jamaica","America/Managua","America/Phoenix", and "America/Metlakatla". See also OlsonNames() for more information on time zones.

Value

A data frame with the following columns:

Name	Type	Description
agency_cd	character	The NWIS code for the agency reporting the data
site_no	character	The USGS site number

dateTime	POSIXct	The date and time of the value converted to UTC
tz_cd	character	The time zone code for dateTime
code	character	Any codes that qualify the corresponding value
value	numeric	The numeric value for the parameter

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:

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
statisticInfo	data.frame	A data frame containing information on the requested statistics on the data
queryTime	POSIXct	The time the data was returned

See Also

renameNWISColumns, importWaterML1

Examples

```
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", ignore_attributes = FALSE)
```

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.

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

Value

A data frame with at least the following columns:

Name	Type	Description
OrganizationIdentifier	character	A designator used to uniquely identify a unique busines
OrganizationFormalName	character	The legal designator (i.e. formal name) of an organizati
ActivityIdentifier	character	Designator that uniquely identifies an activity within ar
ActivityTypeCode	character	The text describing the type of activity.
ActivityMediaName	character	Name or code indicating the environmental medium wl
ActivityMediaSubdivisionName	character	Name or code indicating the environmental matrix as a
ActivityStartDate	character	The calendar date on which the field activity is started.
ActivityStartTime.Time	character	The time of day that is reported when the field activity
ActivityStartTime.TimeZoneCode	character	The time zone for which the time of day is reported. Ar
ActivityEndDate	character	The calendar date when the field activity is completed.
ActivityEndTime.Time	character	The time of day that is reported when the field activity

ActivityEndTime.TimeZoneCode	character	The time zone for which the time of day is reported. Ar
ActivityDepthHeightMeasure.MeasureValue	character	A measurement of the vertical location (measured from
ActivityDepthHeightMeasure.MeasureUnitCode	character	The code that represents the unit for measuring the iten
ActivityDepthAltitudeReferencePointText	character	The reference used to indicate the datum or reference u
ActivityTopDepthHeightMeasure.MeasureValue	character	A measurement of the upper vertical location of a vertic
ActivityTopDepthHeightMeasure.MeasureUnitCode	character	The code that represents the unit for measuring the iten
ActivityBottomDepthHeightMeasure.MeasureValue	character	A measurement of the lower vertical location of a vertic
ActivityBottomDepthHeightMeasure.MeasureUnitCode	character	The code that represents the unit for measuring the iten
ProjectIdentifier	character	A designator used to uniquely identify a data collection
ActivityConductingOrganizationText	character	A name of the Organization conducting an activity.
MonitoringLocationIdentifier	character	A designator used to describe the unique name, number
ActivityCommentText	character	General comments concerning the activity.
SampleAquifer *	character	A code that designates the aquifer associated with grou
HydrologicCondition *	character	Hydrologic condition is the hydrologic condition that is
HydrologicEvent *	character	A hydrologic event that is represented by the sample co
SampleCollectionMethod.MethodIdentifier	character	The identification number or code assigned by the meth
SampleCollectionMethod.MethodIdentifierContext	character	Identifies the source or data system that created or defi
SampleCollectionMethod.MethodName	character	The title that appears on the method from the method p
SampleCollectionEquipmentName	character	The name for the equipment used in collecting the sam
ResultDetectionConditionText	character	The textual descriptor of a result.
CharacteristicName	character	The object, property, or substance which is evaluated o
ResultSampleFractionText	character	The text name of the portion of the sample associated v
ResultMeasureValue	numeric	The reportable measure of the result for the chemical, r
MeasureQualifierCode	character	A code used to identify any qualifying issues that affec
ResultMeasure.MeasureUnitCode	character	The code that represents the unit for measuring the iten
ResultStatusIdentifier	character	Indicates the acceptability of the result with respect to
StatisticalBaseCode	character	The code for the method used to calculate derived resul
ResultValueTypeName	character	A name that qualifies the process which was used in th
ResultWeightBasisText	character	The name that represents the form of the sample or por
ResultTimeBasisText	character	The period of time (in days) over which a measuremen
ResultTemperatureBasisText	character	The name that represents the controlled temperature at
ResultParticleSizeBasisText	character	User defined free text describing the particle size class
PrecisionValue	character	A measure of mutual agreement among individual mea
ResultCommentText	character	Free text with general comments concerning the result.
USGSPCode *	character	5-digit number used in the US Geological Survey comp
ResultDepthHeightMeasure.MeasureValue +	character	A measurement of the vertical location (measured from
ResultDepthHeightMeasure.MeasureUnitCode +	character	The code that represents the unit for measuring the iten
ResultDepthAltitudeReferencePointText +	character	The reference used to indicate the datum or reference u
SubjectTaxonomicName	character	The name of the organism from which a tissue sample
SampleTissueAnatomyName *	character	The name of the anatomy from which a tissue sample v
ResultAnalyticalMethod.MethodIdentifier	character	The identification number or code assigned by the meth
ResultAnalyticalMethod.MethodIdentifierContext	character	Identifies the source or data system that created or defi
ResultAnalyticalMethod.MethodName	character	The title that appears on the method from the method p
MethodDescriptionText *	character	A brief summary that provides general information abo
LaboratoryName	character	The name of Lab responsible for the result.
AnalysisStartDate	character	The calendar date on which the analysis began.
ResultLaboratoryCommentText	character	Remarks which further describe the laboratory procedu
DetectionQuantitationLimitTypeName	character	Text describing the type of detection or quantitation lev

DetectionQuantitationLimitMeasure.MeasureValue	numeric	Constituent concentration that, when processed through
DetectionQuantitationLimitMeasure.MeasureUnitCode	character	The code that represents the unit for measuring the item
PreparationStartDate	character	The calendar date when the preparation/extraction of th
ActivityStartDateTime	POSIXct	Activity start date and time converted to POSIXct UTC
ActivityEndDateTime	POSIXct	Activity end date and time converted to POSIXct UTC.

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

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

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

Examples

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

wqp.data <- readWQPdata(args)

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

wqp.summary <- readWQPdata(args_2)

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

# querying by county
dailyLexingtonVA <- readWQPdata(statecode = "Virginia",
                                countycode="Lexington",
                                parameterCd = "00010")
```

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. 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

```
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", 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:

Name	Type	Description
OrganizationIdentifier	character	A designator used to uniquely identify a unique busines
OrganizationFormalName	character	The legal designator (i.e. formal name) of an organizati
ActivityIdentifier	character	Designator that uniquely identifies an activity within an
ActivityTypeCode	character	The text describing the type of activity.
ActivityMediaName	character	Name or code indicating the environmental medium wh
ActivityMediaSubdivisionName	character	Name or code indicating the environmental matrix as a
ActivityStartDate	character	The calendar date on which the field activity is started.
ActivityStartTime.Time	character	The time of day that is reported when the field activity
ActivityStartTime.TimeZoneCode	character	The time zone for which the time of day is reported. An
ActivityEndDate	character	The calendar date when the field activity is completed.
ActivityEndTime.Time	character	The time of day that is reported when the field activity
ActivityEndTime.TimeZoneCode	character	The time zone for which the time of day is reported. An
ActivityDepthHeightMeasure.MeasureValue	character	A measurement of the vertical location (measured from
ActivityDepthHeightMeasure.MeasureUnitCode	character	The code that represents the unit for measuring the item
ActivityDepthAltitudeReferencePointText	character	The reference used to indicate the datum or reference u
ActivityTopDepthHeightMeasure.MeasureValue	character	A measurement of the upper vertical location of a verti
ActivityTopDepthHeightMeasure.MeasureUnitCode	character	The code that represents the unit for measuring the item
ActivityBottomDepthHeightMeasure.MeasureValue	character	A measurement of the lower vertical location of a verti
ActivityBottomDepthHeightMeasure.MeasureUnitCode	character	The code that represents the unit for measuring the item
ProjectIdentifier	character	A designator used to uniquely identify a data collection
ActivityConductingOrganizationText	character	A name of the Organization conducting an activity.
MonitoringLocationIdentifier	character	A designator used to describe the unique name, number
ActivityCommentText	character	General comments concerning the activity.
SampleAquifer *	character	A code that designates the aquifer associated with grou
HydrologicCondition *	character	Hydrologic condition is the hydrologic condition that is
HydrologicEvent *	character	A hydrologic event that is represented by the sample co
SampleCollectionMethod.MethodIdentifier	character	The identification number or code assigned by the meth
SampleCollectionMethod.MethodIdentifierContext	character	Identifies the source or data system that created or defin
SampleCollectionMethod.MethodName	character	The title that appears on the method from the method p
SampleCollectionEquipmentName	character	The name for the equipment used in collecting the sam
ResultDetectionConditionText	character	The textual descriptor of a result.
CharacteristicName	character	The object, property, or substance which is evaluated o
ResultSampleFractionText	character	The text name of the portion of the sample associated w

ResultMeasureValue	numeric	The reportable measure of the result for the chemical, r
MeasureQualifierCode	character	A code used to identify any qualifying issues that affec
ResultMeasure.MeasureUnitCode	character	The code that represents the unit for measuring the iten
ResultStatusIdentifier	character	Indicates the acceptability of the result with respect to
StatisticalBaseCode	character	The code for the method used to calculate derived resul
ResultValueTypeName	character	A name that qualifies the process which was used in the
ResultWeightBasisText	character	The name that represents the form of the sample or por
ResultTimeBasisText	character	The period of time (in days) over which a measuremen
ResultTemperatureBasisText	character	The name that represents the controlled temperature at
ResultParticleSizeBasisText	character	User defined free text describing the particle size class
PrecisionValue	character	A measure of mutual agreement among individual mea
ResultCommentText	character	Free text with general comments concerning the result.
USGSPCode *	character	5-digit number used in the US Geological Survey comp
ResultDepthHeightMeasure.MeasureValue +	character	A measurement of the vertical location (measured from
ResultDepthHeightMeasure.MeasureUnitCode +	character	The code that represents the unit for measuring the iten
ResultDepthAltitudeReferencePointText +	character	The reference used to indicate the datum or reference u
SubjectTaxonomicName	character	The name of the organism from which a tissue sample
SampleTissueAnatomyName *	character	The name of the anatomy from which a tissue sample
ResultAnalyticalMethod.MethodIdentifier	character	The identification number or code assigned by the meth
ResultAnalyticalMethod.MethodIdentifierContext	character	Identifies the source or data system that created or defi
ResultAnalyticalMethod/MethodName	character	The title that appears on the method from the method p
MethodDescriptionText *	character	A brief summary that provides general information abo
LaboratoryName	character	The name of Lab responsible for the result.
AnalysisStartDate	character	The calendar date on which the analysis began.
ResultLaboratoryCommentText	character	Remarks which further describe the laboratory procedu
DetectionQuantitationLimitTypeName	character	Text describing the type of detection or quantitation lev
DetectionQuantitationLimitMeasure.MeasureValue	numeric	Constituent concentration that, when processed through
DetectionQuantitationLimitMeasure.MeasureUnitCode	character	The code that represents the unit for measuring the iten
PreparationStartDate	character	The calendar date when the preparation/extraction of th
ActivityStartDateTime	POSIXct	Activity start date and time converted to POSIXct UTC
ActivityEndDateTime	POSIXct	Activity end date and time converted to POSIXct UTC.

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

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

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

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','','')
nwisEx <- readWQPqw('USGS-04024000',c('34247','30234','32104','34220'),'','2012-12-20')
nwisEx.summary <- readWQPqw('USGS-04024000',c('34247','30234','32104','34220'),
    '','2012-12-20', querySummary=TRUE)
```

renameNWISColumns *renameColumns*

Description

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

Usage

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

Arguments

`rawData` the daily- or unit-values datset retrieved from NWISweb.

`p00010` the base name for parameter code 00010.

`p00045` the base name for parameter code 00045.

`p00060` the base name for parameter code 00060.

`p00065` the base name for parameter code 00065.

`p00095` the base name for parameter code 00095.

`p00300` the base name for parameter code 00300.

`p00400` the base name for parameter code 00400.

`p62611` the base name for parameter code 62611.

`p63680` the base name for parameter code 63680.

`p72019` the base name for parameter code 72019.

`...` named arguments for the base name for any other parameter code. The form of the name must be like pXXXXX, where XXXXX is the parameter code.

Value

A dataset like `data` with selected columns renamed.

Note

The following statistics codes are converted by `renameNWISColumns`.

00000 Instantaneous Value, suffix: Inst

00001 Maximum value, suffix: Max

00002 Minimum value, suffix: Min

00003 Mean value, no suffix

00006 Sum of values, suffix: Sum

00007 Modal value, suffix: Mode

00008 Median value, suffix: Median

00012 Equivalent mean value, suffix: EqMean

00021 Tidal high-high value, suffix: HiHiTide

00022 Tidal low-high value, suffix: LoHiTide

00023 Tidal high-low value, suffix: HiLoTide

00024 Tidal low-low value, suffix: LoLoTide

See Also

`readNWISdv`, `readNWISuv`

Examples

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

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

`setAccess` *Set data endpoint*

Description

access Indicate which dataRetrieval access code you want to use 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

```
setAccess('internal')

setAccess('public')
```

`stateCd` *US State Code Lookup Table*

Description

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

Value

stateCd data frame.

Name	Type	Description
STATE	character	FIPS State Code
STUSAB	character	Official United States Postal Service (USPS) Code
STATE_NAME	character	State Name
STATENS	character	Geographic Names Information System Identifier (GNISID)

Examples

```
head(stateCd)
```

stateCdLookup *State code look up*

Description

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

Usage

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

Arguments

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

Examples

```
fullName <- stateCdLookup("wi", "fullName")
abbriev <- 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 for more information.

Usage

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

Arguments

`...` see 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
alt_va	character	Altitude of Gage or land surface
alt_acy_va	character	Altitude accuracy
alt_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

parameter_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:

Name	Type	Description
url	character	The url used to generate the data
comment	character	Header comments from the RDB file
queryTime	POSIXct	The time the data was returned

Examples

```
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"),
                            statCd = "00003")
```

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:

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	Station name
site_tp_cd	character	Site type code
dec_lat_va	numeric	Decimal latitude
dec_long_va	numeric	Decimal longitude
queryTime	POSIXct	Query time

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

Name	Type	Description
url	character	The url used to generate the data
queryTime	POSIXct	The time the data was returned

Examples

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

whatWQPdata *Data Available from Water Quality Portal*

Description

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

Usage

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

Arguments

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

`saveFile` path to save the incoming geojson output.

Value

A data frame with at least the following columns:

Name	Type	Description
"type_a"	character	Geojson type
"features.type"	character	Geojson feature type
"type1"	character	Geojson spatial type
"coordinates"	list	List of longitude/latitude
"ProviderName"	character	The name of the database that provided the data to the Water Qaulity
"OrganizationIdentifier"	character	A designator used to uniquely identify a unique business establishme
"OrganizationFormalName"	character	The legal designator (i.e. formal name) of an organization.
"MonitoringLocationIdentifier"	character	A designator used to describe the unique name, number, or code assi
"MonitoringLocationName"	character	The designator specified by the sampling organization for the site at
"MonitoringLocationTypeName"	character	The descriptive name for a type of monitoring location.
"ResolvedMonitoringLocationTypeName"	character	
"HUCEightDigitCode"	character	The 8 digit federal code used to identify the hydrologic unit of the m
"siteUrl"	character	URL to site information
"activityCount"	numeric	
"resultCount"	numeric	
"StateName"	character	State name
"CountyName"	character	County name

See Also

whatNWISsites

Examples

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

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

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

whatWQPsamples *Site Data Import from Water Quality Portal*

Description

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

Usage

```
whatWQPsamples(...)

whatWQPmetrics(...)

whatWQPsites(...)

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:

Name	Type	Description
OrganizationIdentifier	character	A designator used to uniquely identify a unique busine
OrganizationFormalName	character	The legal designator (i.e. formal name) of an organizat
MonitoringLocationIdentifier	character	A designator used to describe the unique name, numbe
MonitoringLocationName	character	The designator specified by the sampling organization
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
DrainageAreaMeasure/MeasureValue *	character	The drainage basin of a lake, stream, wetland, or estuar
DrainageAreaMeasure/MeasureUnitCode *	character	The code that represents the unit for measuring the iten
ContributingDrainageAreaMeasure/MeasureValue *	character	The contributing drainage area of a lake, stream, wetlar
ContributingDrainageAreaMeasure/MeasureUnitCode *	character	The code that represents the unit for measuring the iten
LatitudeMeasure	numeric	The measure of the angular distance on a meridian nort
LongitudeMeasure	numeric	The measure of the angular distance on a meridian east
SourceMapScaleNumeric	character	The number that represents the proportional distance o
HorizontalAccuracyMeasure/MeasureValue *	character	The horizontal measure of the relative accuracy of the
HorizontalAccuracyMeasure/MeasureUnitCode *	character	The code that represents the unit for measuring the iten
HorizontalCollectionMethodName	character	The name that identifies the method used to determine
HorizontalCoordinateReferenceSystemDatumName	character	The name that describes the reference datum used in de
VerticalMeasure/MeasureValue	character	The measure of elevation (i.e., the altitude), above or b
VerticalMeasure/MeasureUnitCode	character	The code that represents the unit for measuring the iten
VerticalAccuracyMeasure/MeasureValue *	character	The vertical measure of the relative accuracy of the lati
VerticalAccuracyMeasure/MeasureUnitCode *	character	The code that represents the unit for measuring the iten
VerticalCollectionMethodName	character	The name that identifies the method used to collect the
VerticalCoordinateReferenceSystemDatumName	character	The name of the reference datum used to determine the
CountryCode	character	A code designator used to identify a primary geopolitic
StateCode	character	A code designator used to identify a principal administ

CountyCode	character	A code designator used to identify a U.S. county or cou
AquiferName *	character	Name of the aquifer in which the well is completed.
FormationTypeText *	character	Name of the primary formation or soils unit, in which t
AquiferTypeName *	character	The type of aquifer, such as confined or unconfined.
ConstructionDateText *	character	Date of construction when well was completed. May b
WellDepthMeasure/MeasureValue *	character	Depth below land surface datum (LSD) to the bottom c
WellDepthMeasure/MeasureUnitCode *	character	The code that represents the unit for measuring the iten
WellHoleDepthMeasure/MeasureValue *	character	Depth below land surface datum (LSD) to the bottom c
WellHoleDepthMeasure/MeasureUnitCode *	character	The code that represents the unit for measuring the iten
queryTime	POSIXct	Query time

* element is only in NWIS

See Also

whatNWISdata

Examples

```
site1 <- whatWQPsamples(siteid="USGS-01594440")

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



type <- "Stream"
sites <- whatWQPmetrics(countycode="US:55:025",siteType=type)
lakeSites <- whatWQPmetrics(siteType = "Lake, Reservoir, Impoundment", statecode = "US:55")


site1 <- whatWQPsites(siteid="USGS-01594440")

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


site1 <- readWQPsummary(siteid="USGS-07144100",
                        summaryYears=5,
                        dataProfile="periodOfRecord")
# Pretty slow:
#state1 <- readWQPsummary(statecode="NJ",
#                           dataProfile="periodOfRecord")
```

zeroPad *Pad string with leading zeros*

Description

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

Usage

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

Arguments

x	character
padTo	number Final desired length of the character

Value

x character returned with leading zeros

Examples

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

Index

∗Topic **IO**
- renameNWISColumns, 50

∗Topic **USGS**
- constructNWISURL, 4
- countyCd, 8
- readNWISdv, 24
- readNWISpCode, 30
- readNWISqw, 33
- readNWISsite, 37
- readNWISuv, 42
- readWQPqw, 47
- stateCd, 52
- whatNWISdata, 54
- zeroPad, 60

∗Topic **WQP**
- constructWQPURL, 7
- readWQPdata, 44
- whatWQPdata, 56
- whatWQPsamples, 57

∗Topic **countyCd**
- countyCd, 8

∗Topic **data**
- constructNWISURL, 4
- constructWQPURL, 7
- readNWISdv, 24
- readNWISpCode, 30
- readNWISqw, 33
- readNWISsite, 37
- readNWISuv, 42
- readWQPdata, 44
- readWQPqw, 47
- whatNWISdata, 54
- whatWQPdata, 56
- whatWQPsamples, 57
- zeroPad, 60

∗Topic **import**
- constructNWISURL, 4
- constructWQPURL, 7
- readNWISdv, 24
- readNWISpCode, 30
- readNWISqw, 33
- readNWISsite, 37
- readNWISuv, 42
- readWQPdata, 44
- readWQPqw, 47
- whatNWISdata, 54
- whatWQPdata, 56
- whatWQPsamples, 57
- zeroPad, 60

∗Topic **manip**
- renameNWISColumns, 50

∗Topic **service**
- constructNWISURL, 4
- constructWQPURL, 7
- readNWISdv, 24
- readNWISpCode, 30
- readNWISqw, 33
- readNWISsite, 37
- readNWISuv, 42
- readWQPdata, 44
- readWQPqw, 47
- whatNWISdata, 54
- whatWQPdata, 56
- whatWQPsamples, 57
- zeroPad, 60

∗Topic **stateCd**
- stateCd, 52

∗Topic **web**
- constructNWISURL, 4
- constructWQPURL, 7
- readNWISdv, 24
- readNWISpCode, 30
- readNWISqw, 33
- readNWISsite, 37
- readNWISuv, 42
- readWQPdata, 44
- readWQPqw, 47
- whatNWISdata, 54

whatWQPdata, 56
whatWQPsamples, 57
zeroPad, 60

addWaterYear, 3

calcWaterYear, 4
constructNWISURL, 4, *27*, *30*, *32*, *35*, *36*, *40*
constructUseURL, 6
constructWQPURL, 7
countyCd, 8
countyCdLookup, 8

dataRetrieval, 9

GET, *10*
getQuerySummary, 9
getWebServiceData, 10

importNGWMN, 11
importRDB1, 12, *23*, *27*, *30–32*, *36*, *40*
importWaterML1, 13, *23*, *25*, *43*
importWaterML2, 16
importWQP, 17, *49*

parameterCdFile, 18

readNGWMNdata, 18
readNGWMNlevels, 19
readNGWMNsites, 20
readNWISdata, 21
readNWISdv, 24, *51*
readNWISgwl, 26
readNWISmeas, 28
readNWISpCode, 30
readNWISpeak, 31
readNWISqw, 33, *49*
readNWISrating, 36
readNWISsite, 37
readNWISstat, 38
readNWISuse, 40
readNWISuv, 42, *51*
readWQPdata, *17*, *35*, 44, *49*
readWQPqw, *17*, *35*, 47
readWQPsummary *(*whatWQPsamples*)*, 57
renameNWISColumns, *15*, *23*, *25*, *43*, 50

setAccess, 52
stateCd, 52
stateCdLookup, 53

whatNWISdata, 54
whatNWISsites, 55
whatWQPdata, 56, *57*
whatWQPmetrics *(*whatWQPsamples*)*, 57
whatWQPsamples, 57
whatWQPsites, *17*, *35*, *49*
whatWQPsites *(*whatWQPsamples*)*, 57

zeroPad, 60