Package ‘phreeqc’

January 10, 2025

Title R Interface to Geochemical Modeling Software

Version 3.8.6

License GPL-3

NeedsCompilation yes

Depends R (>= 3.5.0)

URL https://www.usgs.gov/software/phreeqc-version-3

Description A geochemical modeling program developed by the US Geological Survey that is designed to perform a wide variety of aqueous geochemical calculations, including speciation, batch-reaction, one-dimensional reactive-transport, and inverse geochemical calculations.

Maintainer S.R. Charlton <charlton@usgs.gov>

BugReports https://github.com/usgs-coupled/iphreeqc/issues

LazyData yes

Encoding UTF-8

RoxygenNote 7.3.2

Author S.R. Charlton [aut, cre],
D.L. Parkhurst [aut],
C.A.J. Appelo [aut],
with contributions from D. Gillespie for Chipmunk BASIC [aut],
S.D. Cohen [aut],
A.C. Hindmarsh [aut],
R. Serban [aut],
D. Shumaker [aut],
A.G. Taylor for CVODE/SUNDIALS [aut]

Repository CRAN

Date/Publication 2025-01-10 21:50:05 UTC

Contents

`phreeqc-package` *R interface to the PHREEQC geochemical modeling program.*

Description

An interface to PHREEQC (Version 3). PHREEQC is a geochemical modeling program developed by the US Geological Survey that is designed to perform a wide variety of aqueous geochemical calculations, including speciation, batch-reaction, one-dimensional reactive-transport, and inverse geochemical calculations.

Details

Package:	phreeqc
Type:	Package
Version:	3.8.6
Date:	2025-01-07
License:	BSD_3_clause + file LICENSE

Author(s)

S.R. Charlton, D.L. Parkhurst, and C.A.J. Appelo, with contributions from D. Gillespie for Chipmunk BASIC (p2c) and S.D. Cohen, A.C. Hindmarsh, R. Serban, D. Shumaker, and A.G. Taylor for CVODE (SUNDIALS)
Maintainer: S.R. Charlton <charlton@usgs.gov>

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf
https://computing.llnl.gov/projects/sundials

See Also

Useful links:

- https://www.usgs.gov/software/phreeqc-version-3
- Report bugs at https://github.com/usgs-coupled/iphreeqc/issues

Examples

```
##########################################################################
# Run ex2 and plot results
##########################################################################

# load the phreeqc.dat database
phrLoadDatabaseString(phreeqc.dat)

# run example 2
phrRunString(ex2)

# retrieve selected_output as a list of data.frame
so <- phrGetSelectedOutput()

# plot the results
attach(so$n1)
title  <- "Gypsum-Anhydrite Stability"
xlabel <- "Temperature, in degrees celsius"
ylabel <- "Saturation index"
plot(temp.C., si_gypsum, main = title, xlab = xlabel, ylab = ylabel,
     col = "darkred", xlim = c(25, 75), ylim = c(-0.4, 0.0))
points(temp.C., si_anhydrite, col = "darkgreen")
legend("bottomright", c("Gypsum", "Anhydrite"),
       col = c("darkred", "darkgreen"), pch = c(1, 1))
```

Amm.dat *The Amm.dat database.*

Description

Amm.dat is the same as phreeqc.dat, except that ammonia redox state has been decoupled from the rest of the nitrogen system; that is, ammonia has been defined as a separate component. The database has been reformatted for use by phrLoadDatabaseString.

Usage

```
Amm.dat  # phrLoadDatabaseString(Amm.dat)
```

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Databases: ColdChem.dat, Kinec.v2.dat, Kinec_v3.dat, PHREEQC_ThermoddemV1.10_15Dec2020.dat, Tipping_Hurley.dat, core10.dat, ex15.dat, frezchem.dat, iso.dat, llnl.dat, minteq.dat, minteq.v4.dat, phreeqc.dat, phreeqc_rates.dat, pitzer.dat, sit.dat, wateq4f.dat

ColdChem.dat *The ColdChem.dat database.*

Description

ColdChem.dat is a low-temperature thermodynamic model for the Na-K-Ca-Mg-Cl system incorporating new experimental heat capacities in KCl, MgCl2, and CaCl2 solutions. Jonathan D. Toner and David C. Catling The database has been reformatted for use by phrLoadDatabaseString.

Usage

```
ColdChem.dat  # phrLoadDatabaseString(ColdChem.dat)
```

Source

https://www.usgs.gov/software/phreeqc-version-3/

See Also

Other Databases: Amm.dat, Kinec.v2.dat, Kinec_v3.dat, PHREEQC_ThermoddemV1.10_15Dec2020.dat, Tipping_Hurley.dat, core10.dat, ex15.dat, frezchem.dat, iso.dat, llnl.dat, minteq.dat, minteq.v4.dat, phreeqc.dat, phreeqc_rates.dat, pitzer.dat, sit.dat, wateq4f.dat

core10.dat *The core10.dat database*

Description

core10.dat is a database derived from phreeqc.dat and llnl.dat, with careful data checking, use of SUPCRT for temperature dependence, and addition of molar volumes for pressure dependence. Contributed by Marc Neveu. The database has been reformatted for use by phrLoadDatabaseString.

Usage

```
core10.dat  # phrLoadDatabaseString(core10.dat)
```

Source

https://github.com/MarcNeveu/IcyDwarf/blob/master/IcyDwarf/PHREEQC-3.1.2/core10.dat

References

doi:10.1016/j.gca.2017.06.023

See Also

Other Databases: Amm.dat, ColdChem.dat, Kinec.v2.dat, Kinec_v3.dat, PHREEQC_ThermoddemV1.10_15Dec2020.dat, Tipping_Hurley.dat, ex15.dat, frezchem.dat, iso.dat, llnl.dat, minteq.dat, minteq.v4.dat, phreeqc.dat, phreeqc_rates.dat, pitzer.dat, sit.dat, wateq4f.dat

ex1 *Example 1–Speciation Calculation*

Description

This example calculates the distribution of aqueous species in seawater and the saturation state of seawater relative to a set of minerals. To demonstrate how to expand the model to new elements, uranium is added to the aqueous model defined by phreeqc.dat. The example can be run using the phrRunString routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: ex10, ex11, ex12, ex13a, ex14, ex15, ex16, ex17, ex18, ex19, ex2, ex20a, ex21, ex22, ex3, ex4, ex5, ex6, ex7, ex8, ex9

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex1)
phrGetOutputStrings()
```

---

ex10 *Example 10–Aragonite-Strontianite Solid Solution*

Description

PHREEQC has the capability to model multicomponent ideal and binary nonideal solid solutions. For ideal solid solutions, the activity of each end member solid is equal to its mole fraction. For nonideal solid solutions, the activity of each end member is the product of the mole fraction and an activity coefficient, which is determined from the mole fraction and Guggenheim excess free-energy parameters. Example 10 considers an aragonite (CaCO3)-strontianite (SrCO3) solid solution and demonstrates how the composition of the solid solution and the aqueous phase change as strontium carbonate is added to an initially pure calcium carbonate system. The example can be run using the phrRunString routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: ex1, ex11, ex12, ex13a, ex14, ex15, ex16, ex17, ex18, ex19, ex2, ex20a, ex21, ex22, ex3, ex4, ex5, ex6, ex7, ex8, ex9

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex10)
phrGetOutputStrings()
```

ex11 *Example 11–Transport and Cation Exchange*

Description

The following example simulates the chemical composition of the effluent from a column containing a cation exchanger (Appelo and Postma, 2005). Initially, the column contains a sodium-potassium-nitrate solution in equilibrium with the exchanger. The column is flushed with three pore volumes of calcium chloride solution. Calcium, potassium, and sodium react to equilibrium with the exchanger at all times. The problem is run two ways–by using the ADVECTION data block, which models only advection, and by using the TRANSPORT data block, which simulates advection and dispersive mixing. The example can be run using the `phrRunString` routine.

Source

`https://www.usgs.gov/software/phreeqc-version-3/`

References

`https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf`

See Also

Other Examples: `ex1`, `ex10`, `ex12`, `ex13a`, `ex14`, `ex15`, `ex16`, `ex17`, `ex18`, `ex19`, `ex2`, `ex20a`, `ex21`, `ex22`, `ex3`, `ex4`, `ex5`, `ex6`, `ex7`, `ex8`, `ex9`

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex11)
phrGetOutputStrings()
```

ex12 *Example 12–Advective and Diffusive Flux of Heat and Solutes*

Description

This example demonstrates the capability of PHREEQC to calculate transient transport of heat and solutes in a column or along a 1D flowline. A column is initially filled with a dilute KCl solution at 25 degrees C in equilibrium with a cation exchanger. A KNO3 solution then advects into the column and establishes a new temperature of 0 degrees C. Subsequently, a sodium chloride solution at 24 degrees C is allowed to diffuse from both ends of the column, assuming no heat is lost through the column walls. At one end, a constant boundary condition is imposed, and at the other end, the final cell is filled with the sodium chloride solution and a closed boundary condition is prescribed. For the column end with a constant boundary condition, an analytical solution is compared with PHREEQC results, for unretarded Cl- (R = 1.0) and retarded Na+ and temperature (R = 3.0). Finally, the second-order accuracy of the numerical method is verified by increasing the number of cells by a factor of three and demonstrating a decrease in the error of the numerical solution by approximately one order of magnitude relative to the analytical solution. The example can be run using the phrRunString routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: ex1, ex10, ex11, ex13a, ex14, ex15, ex16, ex17, ex18, ex19, ex2, ex20a, ex21, ex22, ex3, ex4, ex5, ex6, ex7, ex8, ex9

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex12)
phrGetOutputStrings()
```

ex13a *Example 13–Aragonite-Strontianite Solid Solution*

Description

PHREEQC has the capability to model multicomponent ideal and binary nonideal solid solutions. For ideal solid solutions, the activity of each end member solid is equal to its mole fraction. For nonideal solid solutions, the activity of each end member is the product of the mole fraction and an activity coefficient, which is determined from the mole fraction and Guggenheim excess free-energy parameters. Example 10 considers an aragonite ($CaCO3$)-strontianite ($SrCO3$) solid solution and demonstrates how the composition of the solid solution and the aqueous phase change as strontium carbonate is added to an initially pure calcium carbonate system. The example can be run using the phrRunString routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: ex1, ex10, ex11, ex12, ex14, ex15, ex16, ex17, ex18, ex19, ex2, ex20a, ex21, ex22, ex3, ex4, ex5, ex6, ex7, ex8, ex9

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex13a)
phrGetOutputStrings()
```

---

ex14 *Example 14–Advective Transport, Cation Exchange, Surface Complexation, and Mineral Equilibria*

Description

This example uses the phase-equilibrium, cation-exchange, and surface-complexation reaction capabilities of PHREEQC in combination with advective-transport capabilities to model the evolution of water in the Central Oklahoma aquifer. The geochemistry of the aquifer has been described by Parkhurst and others (1996). Two predominant water types occur in the aquifer: a calcium magnesium bicarbonate water with pH in the range of 7.0 to 7.5 in the unconfined part of the aquifer and a sodium bicarbonate water with pH in the range of 8.5 to 9.2 in the confined part of the aquifer. In addition, marine-derived sodium chloride brines exist below the aquifer and presumably in fluid inclusions and dead-end pore spaces within the aquifer. Large concentrations of arsenic, selenium, chromium, and uranium occur naturally within the aquifer. Arsenic is associated almost exclusively with the high-pH, sodium bicarbonate water type. The example can be run using the phrRunString routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: ex1, ex10, ex11, ex12, ex13a, ex15, ex16, ex17, ex18, ex19, ex2, ex20a, ex21, ex22, ex3, ex4, ex5, ex6, ex7, ex8, ex9

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex14)
phrGetOutputStrings()
```

---

ex15 — *Example 15–1D Transport: Kinetic Biodegradation, Cell Growth, and Sorption*

Description

A test problem for advective-dispersive-reactive transport was developed by Tebes-Stevens and Valocchi (1997) and Tebes-Stevens and others (1998). Although based on relatively simple speciation chemistry, the solution to the problem demonstrates several interacting chemical processes that are common to many environmental problems: bacterially mediated degradation of an organic substrate; bacterial cell growth and decay; metal sorption; and aqueous speciation, including metal-ligand complexation. In this example, the test problem is solved with PHREEQC, which produces results almost identical to those of Tebes-Stevens and Valocchi (1997) and Tebes-Stevens and others (1998). The example can be run using the `phrRunString` routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: `ex1`, `ex10`, `ex11`, `ex12`, `ex13a`, `ex14`, `ex16`, `ex17`, `ex18`, `ex19`, `ex2`, `ex20a`, `ex21`, `ex22`, `ex3`, `ex4`, `ex5`, `ex6`, `ex7`, `ex8`, `ex9`

Examples

```
# this example takes longer than 5 seconds
phrLoadDatabaseString(ex15.dat)
phrSetOutputStringsOn(TRUE)
## Not run: phrRunString(ex15)
phrGetOutputStrings()
```

ex15.dat *The ex15.dat database*

Description

ex15.dat is a database used by example 15 (ex15). The database has been reformatted for use by phrLoadDatabaseString.

Usage

```
ex15.dat  # phrLoadDatabaseString(ex15.dat)
```

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Databases: Amm.dat, ColdChem.dat, Kinec.v2.dat, Kinec_v3.dat, PHREEQC_ThermoddemV1.10_15Dec2020.dat, Tipping_Hurley.dat, core10.dat, frezchem.dat, iso.dat, llnl.dat, minteq.dat, minteq.v4.dat, phreeqc.dat, phreeqc_rates.dat, pitzer.dat, sit.dat, wateq4f.dat

ex16 *Example 16–Inverse Modeling of Sierra Spring Waters*

Description

This example repeats the inverse modeling calculations of the chemical evolution of spring-water compositions in the Sierra Nevada that are described in a classic paper by Garrels and Mackenzie (1967). The same example is described in the manual for the inverse-modeling program NETPATH (Plummer and others, 1991 and 1994). The example uses two spring-water compositions, one from an ephemeral spring, which is less chemically evolved, and one from a perennial spring, which probably has had a longer residence time in the subsoil. The differences in composition between the ephemeral spring and the perennial spring are assumed to be caused by reactions between the water and the minerals and gases it contacts. The object of inverse modeling in this example is to find sets of minerals and gases that, when reacted in appropriate amounts, account for the differences in composition between the two solutions. The example can be run using the phrRunString routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: ex1, ex10, ex11, ex12, ex13a, ex14, ex15, ex17, ex18, ex19, ex2, ex20a, ex21, ex22, ex3, ex4, ex5, ex6, ex7, ex8, ex9

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex16)
phrGetOutputStrings()
```

---

ex17 *Example 17–Inverse Modeling With Evaporation*

Description

Evaporation is handled in the same manner as other heterogeneous reactions for inverse modeling. To model evaporation (or dilution), it is necessary to include a phase with the composition "H2O". The important concept in modeling evaporation is the water mole-balance equation (see Parkhurst and Appelo, 1999, "Equations and Numerical Method for Inverse Modeling"). The moles of water in the initial solutions times their mixing fractions, plus water gained or lost by dissolution or precipitation of phases, plus water gained or lost through redox reactions, must equal the moles of water in the final solution. The equation is still approximate because it does not include the moles of water gained or lost in hydrolysis and complexation reactions in the solutions. The results of inverse modeling are compared with a forward model using Pitzer equations to calculate the sequence of salts that precipitate during evaporation. The example can be run using the phrRunString routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: ex1, ex10, ex11, ex12, ex13a, ex14, ex15, ex16, ex18, ex19, ex2, ex20a, ex21, ex22, ex3, ex4, ex5, ex6, ex7, ex8, ex9

Examples

```
phrLoadDatabaseString(pitzer.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex17)
phrGetOutputStrings()
```

---

ex18 *Example 18–Inverse Modeling of the Madison Aquifer*

Description

In this example, inverse modeling, including isotope mole-balance modeling, is applied to the evolution of water in the Madison aquifer in Montana. Plummer and others (1990) used mole-balance modeling to quantify the extent of dedolomitization at locations throughout the aquifer. In the dedolomitization process, anhydrite dissolution causes the precipitation of calcite and dissolution of dolomite. Additional reactions identified by mole-balance modeling include sulfate reduction, cation exchange, and halite and sylvite dissolution (Plummer and others, 1990). Del 13C and del 34S data were used to corroborate the mole-balance models and carbon-14 was used to estimate groundwater ages (Plummer and others, 1990). Initial and final water samples were selected from a flow path that extends from north-central Wyoming northeast across Montana (Plummer and others, 1990, flow path 3). This pair of water samples was selected specifically because it was one of the few pairs that showed a relatively large discrepancy between previous mole-balance approaches and the mole-balance approach of PHREEQC, which includes uncertainties; results for most sample pairs were not significantly different between the two approaches. In addition, this pair of samples was selected because it was modeled in detail in Plummer and others (1990) to determine the sensitivity of mole-balance results to various model assumptions and was used as an example in the NETPATH manual (Plummer and others, 1994, example 6). Results of PHREEQC calculations are compared to NETPATH calculations. This example is also discussed in Parkhurst (1997). The example can be run using the `phrRunString` routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: `ex1`, `ex10`, `ex11`, `ex12`, `ex13a`, `ex14`, `ex15`, `ex16`, `ex17`, `ex19`, `ex2`, `ex20a`, `ex21`, `ex22`, `ex3`, `ex4`, `ex5`, `ex6`, `ex7`, `ex8`, `ex9`

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex18)
phrGetOutputStrings()
```

`ex19` *Example 19–Modeling Cd+2 Sorption With Linear, Freundlich, and Langmuir Isotherms, and With a Deterministic Distribution of Sorption Sites for Organic Matter, Clay Minerals, and Iron Oxyhydroxides*

Description

Sorption of heavy metals and organic pollutants on natural materials can be described by linear, Freundlich, or Langmuir isotherms. All three isotherms can be calculated by PHREEQC, as shown in this example for Cd+2 sorbing on a loamy soil (Christensen, 1984; Appelo and Postma, 2005). A more mechanistic approach, also illustrated here, is to model the distribution of Cd+2 over the sorbing components in the soil, in this case, in and on organic matter, clay minerals, and iron oxyhydroxides. The example can be run using the `phrRunString` routine.

Source

`https://www.usgs.gov/software/phreeqc-version-3/`

References

`https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf`

See Also

Other Examples: `ex1`, `ex10`, `ex11`, `ex12`, `ex13a`, `ex14`, `ex15`, `ex16`, `ex17`, `ex18`, `ex2`, `ex20a`, `ex21`, `ex22`, `ex3`, `ex4`, `ex5`, `ex6`, `ex7`, `ex8`, `ex9`

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex19)
phrGetOutputStrings()
```

ex2 *Example 2–Equilibration With Pure Phases*

Description

This example shows how to calculate the solubility and relative thermodynamic stability of two minerals, gypsum and anhydrite. The example can be run using the phrRunString routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: ex1, ex10, ex11, ex12, ex13a, ex14, ex15, ex16, ex17, ex18, ex19, ex20a, ex21, ex22, ex3, ex4, ex5, ex6, ex7, ex8, ex9

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex2)
phrGetOutputStrings()
```

ex20a *Example 20–Distribution of Isotopes Between Water and Calcite*

Description

The database iso.dat implements the approach to isotope reactions described by Thorstenson and Parkhurst (2002, 2004), in which minor isotopes are treated as individual thermodynamic components. The aqueous and solid species of minor isotopes have slightly different equilibrium constants than those of the major isotopes, which account for fractionation processes. The treatment of isotopes in gases requires a separate species for each isotopic variant of a gas; for example, the isotopic variants of carbon dioxide are CO2, C18OO, C18O2, 13CO2, 13C18OO, and 13C18O2. Similarly, every isotopic variant of a mineral must be included as a component of a solid solution to represent completely the isotopic composition of the solid. The equilibrium constants in iso.dat are derived from empirical fractionation factors, from statistical mechanical theory, or, where no data are available (the most common case), by assuming no fractionation. However, the database is a framework that can be expanded as additional isotopic thermodynamic data become available. The example can be run using the phrRunString routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: ex1, ex10, ex11, ex12, ex13a, ex14, ex15, ex16, ex17, ex18, ex19, ex2, ex21, ex22, ex3, ex4, ex5, ex6, ex7, ex8, ex9

Examples

```
phrLoadDatabaseString(iso.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex20a)
phrGetOutputStrings()
```

---

ex21 *Example 21–Modeling Diffusion of HTO, 36Cl-, 22Na+, and Cs+ in a Radial Diffusion Cell*

Description

This example illustrates how PHREEQC version 3 can simulate a diffusion experiment, as is now often performed for assessing the properties of a repository for nuclear waste in a clay formation. A sample is cut from a core of clay, enveloped in filters, and placed in a diffusion cell (see Van Loon and others, 2004, for details). Solutions with tracers are circulated at the surfaces of the filters, the tracers diffuse into and out of the clay, and the solutions are sampled and analyzed regularly in time. The concentration changes are interpreted with Fick's diffusion equations to obtain transport parameters for modeling the rates of migration of elements away from a waste repository. Transport in clays is mainly diffusive because of the low hydraulic conductivity, and solutes are further retarded by sorption (cations) and by exclusion from part of the pore space (anions). The example can be run using the phrRunString routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: ex1, ex10, ex11, ex12, ex13a, ex14, ex15, ex16, ex17, ex18, ex19, ex2, ex20a, ex22, ex3, ex4, ex5, ex6, ex7, ex8, ex9

Examples

```
phrLoadDatabaseString(phreeqc.dat)
# example 21 requires the selected_output file to be turned on
phrSetSelectedOutputFileOn(1, TRUE)
phrSetOutputStringsOn(TRUE)
# this takes longer than 5 seconds
## Not run: phrRunString(ex21)
phrGetOutputStrings()
```

---

ex22 *Example 22–Modeling Gas Solubilities: CO2 at High Pressures*

Description

PHREEQC calculates the fugacity coefficient with the Peng-Robinson equation of state (Peng and Robinson, 1976) from the critical pressure and temperature, and the acentric factor of the gas in a gas mixture to obtain the limiting volume and the attraction factor in the Van der Waals equation. The fugacity coefficient is close to 1 when the total pressure of the gas phase is less than about 10 atm, and it can be neglected in the solubility calculation. At higher pressures, the effect can be substantial. At low pressures, the concentration of CO2 increases near-linearly with pressure. At 25 degrees C and pressures higher than 62 atm, the concentration increases more gradually because the fugacity coefficient drops rapidly. The example can be run using the `phrRunString` routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: `ex1`, `ex10`, `ex11`, `ex12`, `ex13a`, `ex14`, `ex15`, `ex16`, `ex17`, `ex18`, `ex19`, `ex2`, `ex20a`, `ex21`, `ex3`, `ex4`, `ex5`, `ex6`, `ex7`, `ex8`, `ex9`

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex22)
phrGetOutputStrings()
```

ex3 *Example 3–Mixing*

Description

This example demonstrates the capabilities of PHREEQC to perform a series of geochemical simulations, with the final simulations relying on results from previous simulations within the same run. The example investigates diagenetic reactions that may occur in zones where seawater mixes with carbonate groundwater. The example is divided into five simulations, labeled part A through part E. (A) Carbonate groundwater is defined by equilibrating pure water with calcite at a of 10-2.0 atm. (B) Seawater is defined by using the major-ion data given in table.9. (C) The two solutions are mixed together in the proportions 70 percent groundwater and 30 percent seawater. (D) The mixture is equilibrated with calcite and dolomite. (E) The mixture is equilibrated with calcite only to investigate the chemical evolution if dolomite precipitation is assumed to be negligible. The example can be run using the `phrRunString` routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: `ex1`, `ex10`, `ex11`, `ex12`, `ex13a`, `ex14`, `ex15`, `ex16`, `ex17`, `ex18`, `ex19`, `ex2`, `ex20a`, `ex21`, `ex22`, `ex4`, `ex5`, `ex6`, `ex7`, `ex8`, `ex9`

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex3)
phrGetOutputStrings()
```

ex4 *Example 4–Evaporation and Homogeneous Redox Reactions*

Description

Evaporation is accomplished by removing water from the chemical system. Water can be removed by several methods: (1) water can be specified as an irreversible reactant with a negative reaction coefficient in the REACTION keyword input, (2) the solution can be mixed with pure water which is given a negative mixing fraction in MIX, or (3) "H2O" can be specified as the alternative reaction in EQUILIBRIUM_PHASES keyword input, in which case water is removed or added to the aqueous phase to attain equilibrium with a specified phase. This example uses the first method; the REACTION data block is used to simulate concentration of rainwater by approximately 20-fold by removing 95 percent of the water. The resulting solution contains only about 0.05 kg of water. In a subsequent simulation, the MIX keyword is used to generate a solution that has the same concentrations as the evaporated solution, but has a total mass of water of approximately 1 kg. The example can be run using the `phrRunString` routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: `ex1`, `ex10`, `ex11`, `ex12`, `ex13a`, `ex14`, `ex15`, `ex16`, `ex17`, `ex18`, `ex19`, `ex2`, `ex20a`, `ex21`, `ex22`, `ex3`, `ex5`, `ex6`, `ex7`, `ex8`, `ex9`

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex4)
phrGetOutputStrings()
```

---

ex5 *Example 5–Irreversible Reactions*

Description

This example demonstrates the irreversible reaction capabilities of PHREEQC in modeling the oxidation of pyrite. Oxygen (O2) and NaCl are added irreversibly to pure water in six amounts (0.0, 0.001, 0.005, 0.01, 0.03, and 0.05 mol); the relative proportion of O2 to NaCl in the irreversible reaction is 1.0 to 0.5. Pyrite, calcite, and goethite are allowed to dissolve to equilibrium and the carbon dioxide partial pressure is maintained at 10-3.5 (atmospheric partial pressure). In addition, gypsum is allowed to precipitate if it becomes supersaturated. The example can be run using the `phrRunString` routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: ex1, ex10, ex11, ex12, ex13a, ex14, ex15, ex16, ex17, ex18, ex19, ex2, ex20a, ex21, ex22, ex3, ex4, ex6, ex7, ex8, ex9

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex5)
phrGetOutputStrings()
```

---

ex6 *Example 6–Reaction-Path Calculations*

Description

In this example, the precipitation of phases as a result of incongruent dissolution of K-feldspar (microcline) is investigated. Only the four phases originally addressed by Helgeson and others (1969)–K-feldspar, gibbsite, kaolinite, and K-mica (muscovite)–are considered. The thermodynamic data for the phases (PHASES keyword) are derived from Robie and others (1978) and are the same as for test problem 5 in the PHREEQE manual (Parkhurst and others, 1980). The example can be run using the phrRunString routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: ex1, ex10, ex11, ex12, ex13a, ex14, ex15, ex16, ex17, ex18, ex19, ex2, ex20a, ex21, ex22, ex3, ex4, ex5, ex7, ex8, ex9

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex6)
phrGetOutputStrings()
```

ex7 *Example 7–Gas-Phase Calculations*

Description

This example demonstrates the capabilities of PHREEQC to model the evolution of gas compositions in equilibrium with a solution with a fixed (total) pressure or a fixed volume of the gas phase. In the case of a fixed-pressure gas phase, a gas bubble forms as soon as the sum of the partial pressures of the component gases exceeds the specified pressure of the gas phase. Once the bubble forms, its volume and composition will vary with the extent of reactions. This case applies to gas bubbles forming in surface water or groundwater at a given depth, where the total pressure is constant. With a fixed-volume gas phase, the aqueous solution is in contact with a head space of a fixed volume, which is typical for a laboratory experiment with a closed bottle. The gas phase always exists in this head space, but its pressure and composition will vary with the reactions. Another way to model gas-liquid reactions in PHREEQC is to maintain a fixed partial pressure by using the EQUILIBRIUM_PHASES data block. This fixed-partial-pressure approach is illustrated in this example by fixing the CO2 pressure for a SOLUTION. The example can be run using the `phrRunString` routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: `ex1`, `ex10`, `ex11`, `ex12`, `ex13a`, `ex14`, `ex15`, `ex16`, `ex17`, `ex18`, `ex19`, `ex2`, `ex20a`, `ex21`, `ex22`, `ex3`, `ex4`, `ex5`, `ex6`, `ex8`, `ex9`

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex7)
phrGetOutputStrings()
```

ex8 *Example 8–Surface Complexation*

Description

In all surface complexation models, sorption is a function of both chemical and electrostatic energy as described by the free energy relationship. Sorption is stronger when the Gibbs energy decreases. Thus, a counter-ion that carries a charge opposite to the surface charge tends to be sorbed electrostatically, while a co-ion that carries a charge with the same sign as the surface tends to be rejected. The example can be run using the phrRunString routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: ex1, ex10, ex11, ex12, ex13a, ex14, ex15, ex16, ex17, ex18, ex19, ex2, ex20a, ex21, ex22, ex3, ex4, ex5, ex6, ex7, ex9

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
# example 8 requires the selected_output file to be turned on
phrSetSelectedOutputFileOn(1, TRUE)
phrRunString(ex8)
phrGetOutputStrings()
# cleanup
unlink(c("ex8.sel", "Zn1e_7", "Zn1e_4"))
```

ex9 *Example 9–Kinetic Oxidation of Dissolved Ferrous Iron With Oxygen*

Description

Kinetic rate expressions can be defined in a completely general way in PHREEQC by using Basic statements in the RATES data block. The rate expressions can be used in batch-reaction and transport calculations with the KINETICS data block. For transport calculations (ADVECTION or TRANSPORT), kinetic reactions can be defined cell by cell by the number range following the KINETICS keyword (KINETICS m-n). The rate expressions are integrated with an embedded (up to)

5th-order Runge-Kutta-Fehlberg algorithm, or with a stiff, variable-order, variable-step multistep solver (Cohen and Hindmarsh, 1996). Equilibrium is calculated before a kinetic calculation is initiated and again when a kinetic reaction increment is added. Equilibrium includes solution species equilibrium; exchange-, equilibrium-phase-, solid-solution-, and surface-assemblage equilibrium; and gas-phase equilibrium. A check is performed to ensure that the difference between estimates of the integrated rate over a time interval is smaller than a user-defined tolerance. If the tolerance is not satisfied, then the integration over the time interval is automatically restarted with a smaller time interval. The example can be run using the `phrRunString` routine.

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Examples: `ex1`, `ex10`, `ex11`, `ex12`, `ex13a`, `ex14`, `ex15`, `ex16`, `ex17`, `ex18`, `ex19`, `ex2`, `ex20a`, `ex21`, `ex22`, `ex3`, `ex4`, `ex5`, `ex6`, `ex7`, `ex8`

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
phrRunString(ex9)
phrGetOutputStrings()
```

`frezchem.dat` *The frezchem.dat database*

Description

frezchem.dat is a database from Jonathon Toner and the developers for the frezchem code (Toner and Sletten (2013), Marion and coworkers). Frezchem is a Pitzer model for temperatures of 0 C and below. `phrLoadDatabaseString`.

Usage

```
frezchem.dat  # phrLoadDatabaseString(frezchem.dat)
```

See Also

Other Databases: `Amm.dat`, `ColdChem.dat`, `Kinec.v2.dat`, `Kinec_v3.dat`, `PHREEQC_ThermoddemV1.10_15Dec2020.dat`, `Tipping_Hurley.dat`, `core10.dat`, `ex15.dat`, `iso.dat`, `llnl.dat`, `minteq.dat`, `minteq.v4.dat`, `phreeqc.dat`, `phreeqc_rates.dat`, `pitzer.dat`, `sit.dat`, `wateq4f.dat`

iso.dat *The iso.dat database.*

Description

iso.dat is a partial implementation of the individual component approach to isotope calculations as described by Thorstenson and Parkhurst. The database has been reformatted for use by phrLoadDatabaseString.

Usage

```
iso.dat  # phrLoadDatabaseString(iso.dat)
```

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Databases: Amm.dat, ColdChem.dat, Kinec.v2.dat, Kinec_v3.dat, PHREEQC_ThermoddemV1.10_15Dec2020.dat, Tipping_Hurley.dat, core10.dat, ex15.dat, frezchem.dat, llnl.dat, minteq.dat, minteq.v4.dat, phreeqc.dat, phreeqc_rates.dat, pitzer.dat, sit.dat, wateq4f.dat

Kinec.v2.dat *Thermodynamic and rates database from Oelkers and coworkers.*

Description

Kinec.v2.dat contains the parameters for calculating mineral dissolution rates for primary and secondary silicate minerals using the equations and parameters reported by Hermanska et al. (2022, 2023), and dissolution rates for other non)-silicate mineral systems using the equations and parameters reported by Oelkers and Addassi (2024, in preparation).

Usage

```
Kinec.v2.dat  # phrLoadDatabaseString(Kinec.v2.dat)
```

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

Hermanska et al. (2022, 2003) and Oelkers and Addassi (2024, in preparation).

See Also

Other Databases: Amm.dat, ColdChem.dat, Kinec_v3.dat, PHREEQC_ThermoddemV1.10_15Dec2020.dat, Tipping_Hurley.dat, core10.dat, ex15.dat, frezchem.dat, iso.dat, llnl.dat, minteq.dat, minteq.v4.dat, phreeqc.dat, phreeqc_rates.dat, pitzer.dat, sit.dat, wateq4f.dat

---

Kinec_v3.dat *Thermodynamic and rates database from Oelkers and coworkers.*

Description

Kinec_v3.dat contains the parameters for calculating mineral dissolution rates for primary and secondary silicate minerals using the equations and parameters reported by Hermanska et al. (2022, 2023), and dissolution rates for other non)-silicate mineral systems using the equations and parameters reported by Oelkers and Addassi (2024, in preparation).

Usage

```
Kinec_v3.dat  # phrLoadDatabaseString(Kinec_v3.dat)
```

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

Hermanska et al. (2022, 2003) and Oelkers and Addassi (2024, in preparation).

See Also

Other Databases: Amm.dat, ColdChem.dat, Kinec.v2.dat, PHREEQC_ThermoddemV1.10_15Dec2020.dat, Tipping_Hurley.dat, core10.dat, ex15.dat, frezchem.dat, iso.dat, llnl.dat, minteq.dat, minteq.v4.dat, phreeqc.dat, phreeqc_rates.dat, pitzer.dat, sit.dat, wateq4f.dat

---

llnl.dat *The llnl.dat database.*

Description

llnl.dat is a database derived from databases for EQ3/6 and Geochemist's Workbench that uses thermodynamic data compiled by the Lawrence Livermore National Laboratory. The database has been reformatted for use by phrLoadDatabaseString.

Usage

```
llnl.dat  # phrLoadDatabaseString(llnl.dat)
```

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Databases: Amm.dat, ColdChem.dat, Kinec.v2.dat, Kinec_v3.dat, PHREEQC_ThermoddemV1.10_15Dec2020.dat, Tipping_Hurley.dat, core10.dat, ex15.dat, frezchem.dat, iso.dat, minteq.dat, minteq.v4.dat, phreeqc.dat, phreeqc_rates.dat, pitzer.dat, sit.dat, wateq4f.dat

---

minteq.dat *The minteq.dat database.*

Description

minteq.dat is a database derived from the databases for the program MINTEQA2. The database has been reformatted for use by phrLoadDatabaseString.

Usage

```
minteq.dat  # phrLoadDatabaseString(minteq.dat)
```

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Databases: Amm.dat, ColdChem.dat, Kinec.v2.dat, Kinec_v3.dat, PHREEQC_ThermoddemV1.10_15Dec2020.dat, Tipping_Hurley.dat, core10.dat, ex15.dat, frezchem.dat, iso.dat, llnl.dat, minteq.v4.dat, phreeqc.dat, phreeqc_rates.dat, pitzer.dat, sit.dat, wateq4f.dat

`minteq.v4.dat` *The minteq.v4.dat database.*

Description

minteq.v4.dat is a database derived from MINTEQA2 version 4. The database has been reformatted for use by `phrLoadDatabaseString`.

Usage

```
minteq.v4.dat  # phrLoadDatabaseString(minteq.v4.dat)
```

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Databases: `Amm.dat`, `ColdChem.dat`, `Kinec.v2.dat`, `Kinec_v3.dat`, `PHREEQC_ThermoddemV1.10_15Dec2020.dat`, `Tipping_Hurley.dat`, `core10.dat`, `ex15.dat`, `frezchem.dat`, `iso.dat`, `llnl.dat`, `minteq.dat`, `phreeqc.dat`, `phreeqc_rates.dat`, `pitzer.dat`, `sit.dat`, `wateq4f.dat`

`phrAccumulateLine` *Accumulate line(s) for input to phreeqc.*

Description

Appends a line of text to the input buffer in order to be run using `phrRunAccumulated`.

Usage

```
phrAccumulateLine(line)
```

Arguments

`line`	the line(s) to add for input to phreeqc.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Accumulate: phrClearAccumulatedLines(), phrGetAccumulatedLines(), phrRunAccumulated()

Examples

```
# this example loads the phreeqc.dat database, accumulates input, and
# runs it
phrLoadDatabaseString(phreeqc.dat)
phrAccumulateLine("TITLE Example 2.--Temperature dependence of solubility")
phrAccumulateLine("                  of gypsum and anhydrite")
phrAccumulateLine("SOLUTION 1 Pure water")
phrAccumulateLine("        pH      7.0")
phrAccumulateLine("        temp    25.0")
phrAccumulateLine("EQUILIBRIUM_PHASES 1")
phrAccumulateLine("        Gypsum          0.0     1.0")
phrAccumulateLine("        Anhydrite       0.0     1.0")
phrAccumulateLine("REACTION_TEMPERATURE 1")
phrAccumulateLine("        25.0 75.0 in 51 steps")
phrAccumulateLine("SELECTED_OUTPUT")
phrAccumulateLine("        -file   ex2.sel")
phrAccumulateLine("        -temperature")
phrAccumulateLine("        -si     anhydrite  gypsum")
phrAccumulateLine("END")
phrSetOutputFileOn(TRUE)
phrSetOutputFileName(file.path(tempdir(), "ex2.output"))
if (is.null(phrRunAccumulated())) {
  cat(paste("see ", phrGetOutputFileName(), ".\n", sep = ""))
}
```

---

phrClearAccumulatedLines

Clear the accumulated input buffer.

Description

Clears the accumulated input buffer. The input buffer is accumulated from calls to the phrAccumulateLine method.

Usage

```
phrClearAccumulatedLines()
```

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Accumulate: phrAccumulateLine(), phrGetAccumulatedLines(), phrRunAccumulated()

Examples

```
# This example loads some keyword input, clears the input, and displays
# the results.
phrAccumulateLine("SOLUTION 1")
phrAccumulateLine("END")
cat("The accumulated input is:", phrGetAccumulatedLines(), sep = "\n")
phrClearAccumulatedLines()
cat("The accumulated input now is:\n", phrGetAccumulatedLines(), sep = "\n")
```

phreeqc.dat *The phreeqc.dat database*

Description

phreeqc.dat is a phreeqc database file derived from PHREEQE, which is consistent with wateq4f.dat, but has a smaller set of elements and aqueous species. The database has been reformatted for use by `phrLoadDatabaseString`.

Usage

```
phreeqc.dat  # phrLoadDatabaseString(phreeqc.dat)
```

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Databases: `Amm.dat`, `ColdChem.dat`, `Kinec.v2.dat`, `Kinec_v3.dat`, `PHREEQC_ThermoddemV1.10_15Dec2020.dat`, `Tipping_Hurley.dat`, `core10.dat`, `ex15.dat`, `frezchem.dat`, `iso.dat`, `llnl.dat`, `minteq.dat`, `minteq.v4.dat`, `phreeqc_rates.dat`, `pitzer.dat`, `sit.dat`, `wateq4f.dat`

phreeqc_rates.dat *Thermodynamic and rates database*

Description

Same as the phreeqc.dat database, but with new data blocks RATE_PARAMETERS_HERMANSKA, RATE_PARAMETERS_PK, and RATE_PARAMETERS_SVD that tabulate rate parameters from Hermanska and others (2023), Palandri and Kharaka (2004), and Sverdrup and others (2019). The Sverdrup parameters are only for two minerals as a demonstration. Basic functions RATE_HERMANSKA, RATE_PK, and RATE_SVD can be used to calculate rates using the corresponding parameters.

Usage

```
phreeqc_rates.dat  # phrLoadDatabaseString(phreeqc_rates.dat)
```

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

Hermanska and others (2023), Palandri and Kharaka (2004), and Sverdrup and others (2019).

See Also

Other Databases: Amm.dat, ColdChem.dat, Kinec.v2.dat, Kinec_v3.dat, PHREEQC_ThermoddemV1.10_15Dec2020.dat, Tipping_Hurley.dat, core10.dat, ex15.dat, frezchem.dat, iso.dat, llnl.dat, minteq.dat, minteq.v4.dat, phreeqc.dat, pitzer.dat, sit.dat, wateq4f.dat

PHREEQC_ThermoddemV1.10_15Dec2020.dat *Thermochemical Database from the BRGM institute (French Geological Survey)*

Description

Thermochemical Database from the BRGM institute (French Geological Survey)

Usage

```
PHREEQC_ThermoddemV1.10_15Dec2020.dat
# phrLoadDatabaseString(PHREEQC_ThermoddemV1.10_15Dec2020.dat)
```

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://thermoddem.brgm.fr/

See Also

Other Databases: Amm.dat, ColdChem.dat, Kinec.v2.dat, Kinec_v3.dat, Tipping_Hurley.dat, core10.dat, ex15.dat, frezchem.dat, iso.dat, llnl.dat, minteq.dat, minteq.v4.dat, phreeqc.dat, phreeqc_rates.dat, pitzer.dat, sit.dat, wateq4f.dat

---

phrGetAccumulatedLines

Retrieve the accumulated input.

Description

Returns the accumulated input as a character vector.

Usage

```
phrGetAccumulatedLines()
```

Value

A character vector containing the accumulated input.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Accumulate: phrAccumulateLine(), phrClearAccumulatedLines(), phrRunAccumulated()

Examples

```
# This example loads some keyword input and displays the contents.
phrAccumulateLine("SOLUTION 1")
phrAccumulateLine("END")
cat("The accumulated input is:", phrGetAccumulatedLines(), sep = "\n")
```

phrGetComponentList *Retrieve a list containing the current list of components.*

Description

Retrieve a list containing the current list of components.

Usage

```
phrGetComponentList()
```

Value

A list containing the names of the components defined in the current system.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

Examples

```
# This example runs the ex2 input file and echos the list of components.
phrLoadDatabaseString(phreeqc.dat)
phrRunString(ex2)
cat("components:\n")
for (comp in phrGetComponentList()) {
  cat(comp, "\n")
}
```

phrGetDumpFileName *Retrieve the name of the dump file.*

Description

Retrieves the name of the dump file. This file name is used if not specified within DUMP input. The default value is dump.0.out.

Usage

```
phrGetDumpFileName()
```

Value

The name of the dump file as a string.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Dump: phrGetDumpFileOn(), phrGetDumpStrings(), phrGetDumpStringsOn(), phrSetDumpFileName(), phrSetDumpFileOn(), phrSetDumpStringsOn()

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetDumpFileOn(TRUE)
phrSetDumpFileName(file.path(tempdir(), "dump.out"))

input <-              "SOLUTION 1 Pure water     \n"
input <- paste(input, "EQUILIBRIUM_PHASES 1      \n")
input <- paste(input, "    Calcite 0 10          \n")
input <- paste(input, "SAVE solution 1           \n")
input <- paste(input, "SAVE equilibrium_phases 1 \n")
input <- paste(input, "DUMP                      \n")
input <- paste(input, "    -solution 1           \n")
input <- paste(input, "    -equilibrium_phases  1\n")

if (!is.null(phrRunString(input))) {
  cat(phrGetErrorStrings())
}
cat(paste("see ", phrGetDumpFileName(), ".\n", sep = ""))
```

phrGetDumpFileOn *Retrieve the current value of the dump file switch.*

Description

Retrieve the current value of the dump file switch.

Usage

```
phrGetDumpFileOn()
```

Value

TRUE if output as defined by the DUMP data block are currently being written to file.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Dump: phrGetDumpFileName(), phrGetDumpStrings(), phrGetDumpStringsOn(), phrSetDumpFileName(), phrSetDumpFileOn(), phrSetDumpStringsOn()

phrGetDumpStrings *Retrieve DUMP strings.*

Description

Retrieves the string buffer containing DUMP output as a character vector.

Usage

```
phrGetDumpStrings()
```

Value

The dump output as a character vector.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Dump: phrGetDumpFileName(), phrGetDumpFileOn(), phrGetDumpStringsOn(), phrSetDumpFileName(), phrSetDumpFileOn(), phrSetDumpStringsOn()

Examples

```
phrLoadDatabaseString(phreeqc.dat)
phrSetDumpStringsOn(TRUE)

input <-              "SOLUTION 1 Pure water     \n"
input <- paste(input, "EQUILIBRIUM_PHASES 1      \n")
input <- paste(input, "    Calcite 0 10          \n")
input <- paste(input, "SAVE solution 1           \n")
input <- paste(input, "SAVE equilibrium_phases 1 \n")
input <- paste(input, "DUMP                      \n")
input <- paste(input, "    -solution 1           \n")
input <- paste(input, "    -equilibrium_phases 1 \n")

if (!is.null(phrRunString(input))) {
  cat(phrGetErrorStrings(), sep = "\n")
}
cat(phrGetDumpStrings(), sep = "\n")
```

phrGetDumpStringsOn *Retrieve the current value of the dump strings switch.*

Description

Retrieve the current value of the dump strings switch.

Usage

```
phrGetDumpStringsOn()
```

Value

TRUE if output as defined by the DUMP data block are currently being stored as a character vector.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Dump: phrGetDumpFileName(), phrGetDumpFileOn(), phrGetDumpStrings(), phrSetDumpFileName(), phrSetDumpFileOn(), phrSetDumpStringsOn()

phrGetErrorFileName *Retrieve the name of the error file.*

Description

Retrieves the name of the error file. The default value is phreeqc.0.err.

Usage

```
phrGetErrorFileName()
```

Details

The error file switch must be set using the phrSetErrorFileOn function.

Value

The name of the error file as a string.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Error: `phrGetErrorFileOn()`, `phrGetErrorStrings()`, `phrGetErrorStringsOn()`, `phrSetErrorFileName()`, `phrSetErrorFileOn()`, `phrSetErrorStringsOn()`

`phrGetErrorFileOn` *Retrieve the current value of the error file switch.*

Description

Retrieve the current value of the error file switch.

Usage

```
phrGetErrorFileOn()
```

Value

TRUE if error messages are currently being written to file.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Error: `phrGetErrorFileName()`, `phrGetErrorStrings()`, `phrGetErrorStringsOn()`, `phrSetErrorFileName()`, `phrSetErrorFileOn()`, `phrSetErrorStringsOn()`

`phrGetErrorStrings` *Retrieve error string messages.*

Description

Retrieves a character vector containing any error messages that were generated during the last invocation of the following methods: `phrAccumulateLine`, `phrLoadDatabase`, `phrLoadDatabaseString`, `phrRunAccumulated`, `phrRunFile`, `phrRunString`

Usage

```
phrGetErrorStrings()
```

Details

This routine is rarely needed when running interactively since the error string is displayed when it occurs.

Value

The error messages as a character vector.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Error: phrGetErrorFileName(), phrGetErrorFileOn(), phrGetErrorStringsOn(), phrSetErrorFileName(), phrSetErrorFileOn(), phrSetErrorStringsOn()

Examples

```
# loaddatabase should fail
n <- try(phrLoadDatabase("missing.dat"), silent = TRUE)
# if n is non-NULL display error string
if (!is.null(n)) phrGetErrorStrings()
```

phrGetErrorStringsOn *Retrieve the current value of the error strings switch.*

Description

Retrieve the current value of the error strings switch.

Usage

```
phrGetErrorStringsOn()
```

Value

TRUE if error messages are currently being stored as a character vector.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Error: phrGetErrorFileName(), phrGetErrorFileOn(), phrGetErrorStrings(), phrSetErrorFileName(), phrSetErrorFileOn(), phrSetErrorStringsOn()

phrGetLogFileName *Retrieve the name of the log file.*

Description

Retrieves the name of the log file. The default name is phreeqc.0.log.

Usage

```
phrGetLogFileName()
```

Value

The name of the log file as a string.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Log: phrGetLogFileOn(), phrGetLogStrings(), phrGetLogStringsOn(), phrSetLogFileName(), phrSetLogFileOn(), phrSetLogStringsOn()

Examples

```
# This example checks to see if the log file is turned on
# and prints the appropriate message
if (phrGetLogFileOn()) {
  cat("The name of the log file (is/will be):", phrGetLogFileName(), "\n")
} else {
  cat("The log file is not turned on\n")
}
```

phrGetLogFileOn *Retrieve the current value of the log file switch.*

Description

Retrieve the current value of the log file switch.

Usage

```
phrGetLogFileOn()
```

Value

TRUE if log messages are currently being written to file.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Log: phrGetLogFileName(), phrGetLogStrings(), phrGetLogStringsOn(), phrSetLogFileName(), phrSetLogFileOn(), phrSetLogStringsOn()

---

phrGetLogStrings *Retrieve log output.*

Description

Retrieves the string buffer containing phreeqc log output.

Usage

```
phrGetLogStrings()
```

Value

A character vector containing phreeqc log output.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Log: phrGetLogFileName(), phrGetLogFileOn(), phrGetLogStringsOn(), phrSetLogFileName(), phrSetLogFileOn(), phrSetLogStringsOn()

Examples

```
# This example equilibrates pure water with gypsum with the output file on.
phrLoadDatabaseString(phreeqc.dat)
phrSetLogStringsOn(TRUE)

input <- vector(mode="character")
input <- c(input, "SOLUTION 1 Pure water ")
input <- c(input, "EQUILIBRIUM_PHASES 1  ")
input <- c(input, "  Gypsum 0 10         ")
input <- c(input, "KNOBS                 ")
input <- c(input, "  -logfile TRUE       ")
```

```
if (is.null(phrRunString(input))) {
  log <- phrGetLogStrings()
}
```

phrGetLogStringsOn *Retrieve the current value of the log strings switch.*

Description

Retrieve the current value of the log strings switch.

Usage

```
phrGetLogStringsOn()
```

Value

TRUE if log messages are currently being stored as a character vector.

References

https://www.usgs.gov/software/phreeqc-version-3/

See Also

Other Log: phrGetLogFileName(), phrGetLogFileOn(), phrGetLogStrings(), phrSetLogFileName(), phrSetLogFileOn(), phrSetLogStringsOn()

phrGetOutputFileName *Retrieve the name of the output file.*

Description

Retrieves the name of the output file. The default name is phreeqc.0.out.

Usage

```
phrGetOutputFileName()
```

Value

The name of the output file as a string.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Output: phrGetOutputFileOn(), phrGetOutputStrings(), phrGetOutputStringsOn(), phrSetOutputFileName(), phrSetOutputFileOn(), phrSetOutputStringsOn()

Examples

```
# This example equilibrates pure water with gypsum with the output file on.
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputFileOn(TRUE)
phrSetOutputFileName(file.path(tempdir(), "gypsum.out"))

input <- vector(mode="character")
input <- c(input, "SOLUTION 1 Pure water ")
input <- c(input, "EQUILIBRIUM_PHASES 1  ")
input <- c(input, "  Gypsum 0 10         ")

if (is.null(phrRunString(input))) {
  output <- readLines(phrGetOutputFileName())
  unlink(phrGetOutputFileName())  # tidy up
}
```

---

phrGetOutputFileOn *Retrieve the current value of the output file switch.*

Description

Retrieve the current value of the output file switch.

Usage

```
phrGetOutputFileOn()
```

Value

TRUE if output messages are currently being written to file.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Output: phrGetOutputFileName(), phrGetOutputStrings(), phrGetOutputStringsOn(), phrSetOutputFileName(), phrSetOutputFileOn(), phrSetOutputStringsOn()

phrGetOutputStrings *Retrieve standard phreeqc output.*

Description

Retrieves the phreeqc output as a character vector.

Usage

```
phrGetOutputStrings()
```

Details

A NULL value is returned when there is no selected-output.

Value

A character vector containing phreeqc output.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Output: phrGetOutputFileName(), phrGetOutputFileOn(), phrGetOutputStringsOn(), phrSetOutputFileName(), phrSetOutputFileOn(), phrSetOutputStringsOn()

Examples

```
# This example equilibrates pure water with calcite and displays
# the results
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)

input <- vector(mode="character")
input <- c(input, "SOLUTION 1 Pure water ")
input <- c(input, "EQUILIBRIUM_PHASES 1  ")
input <- c(input, "  Gypsum 0 10         ")

if (is.null(phrRunString(input))) {
  cat(phrGetOutputStrings(), sep = "\n")
}
```

phrGetOutputStringsOn *Retrieve the current value of the output strings switch.*

Description

Retrieve the current value of the output strings switch.

Usage

```
phrGetOutputStringsOn()
```

Value

TRUE if output messages are currently being stored as a character vector.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Output: phrGetOutputFileName(), phrGetOutputFileOn(), phrGetOutputStrings(), phrSetOutputFileName(), phrSetOutputFileOn(), phrSetOutputStringsOn()

phrGetSelectedOutput *Returns the contents of the selected output as a list of data frames.*

Description

phrGetSelectedOutput returns a named list containing the resultant selected output blocks. The names of each data frame are creating by concatenating the letter 'n' and the user number of the selected output block.

Usage

```
phrGetSelectedOutput(allow_ = TRUE)
```

Arguments

allow_	used for compatibility with R prior to 1.9.0 (default is TRUE).

Details

phrGetSelectedOutput uses the make.names function to create valid column names. The allow_ variable is passed to make.names and is used for backward compatibility.

Value

Returns a named list of data frames containing the selected_output from the previous run.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Selected Output: `phrGetSelectedOutputFileName()`, `phrSetSelectedOutputFileName()`, `phrSetSelectedOutputFileOn()`

Examples

```
# Load database and run ex2
phrLoadDatabaseString(phreeqc.dat)
phrRunString(ex2)

# display a summary of the results
df <- phrGetSelectedOutput()
summary(df$n1)
```

---

phrGetSelectedOutputFileName

Retrieve the name of the selected_output file.

Description

Retrieves the name of the selected_output file. The default value is selected_{nuser}.0.out.

Usage

```
phrGetSelectedOutputFileName(nuser)
```

Arguments

`nuser`	the user number specified in the SELECTED_OUTPUT block.

Details

The selected_output file must be turned on using the `phrSetSelectedOutputFileOn` function.

Value

The name of the selected_output file as a string.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Selected Output: phrGetSelectedOutput(), phrSetSelectedOutputFileName(), phrSetSelectedOutputFileOn(

Examples

```
# This example equilibrates pure water with calcite at various temperatures
# and displays the name of the selected_output file.
phrLoadDatabaseString(phreeqc.dat)
phrSetSelectedOutputFileOn(1, TRUE)
phrSetSelectedOutputFileName(1, file.path(tempdir(), "ex2.sel"))

input <- c(
  'SOLUTION 1 Pure water     ',
  'EQUILIBRIUM_PHASES 1      ',
  '    Calcite    0.0   1.0  ',
  'REACTION_TEMPERATURE 1    ',
  '    25.0 75.0 in 51 steps ',
  'SELECTED_OUTPUT 1         ',
  '    -temperature          ',
  '    -si     calcite       '
  )


if (is.null(phrRunString(input))) {
  cat(paste("see ", phrGetSelectedOutputFileName(1), ".\n", sep = ""))
}
```

phrGetWarningStrings *Retrieve warning messages.*

Description

Returns a character vector containing any warning messages that were generated during the last invocation of the following methods: phrAccumulateLine, phrLoadDatabase, phrLoadDatabaseString, phrRunAccumulated, phrRunFile, phrRunString.

Usage

```
phrGetWarningStrings()
```

Details

A NULL value is returned if there are no warnings.

Value

A character vector containing warning messages or NULL.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

Examples

```
# This example loads the phreeqc.dat database and attempts to use the
# DATABASE keyword to set the database to wateq4f.dat.  A warning is
# displayed stating that the DATABASE keyword is ignored in the 'R'
# implementation.
phrLoadDatabaseString(phreeqc.dat)
phrAccumulateLine("DATABASE wateq4f.dat")
phrAccumulateLine("SOLUTION 1")
phrRunAccumulated()
if (!is.null(phrGetWarningStrings())) {
  cat(phrGetWarningStrings(), sep = "\n")
}
```

---

phrLoadDatabase *Load a phreeqc database file*

Description

Loads the given phreeqc database into phreeqc. Returns NULL if successful.

Usage

```
phrLoadDatabase(filename)
```

Arguments

filename	The name of the database file.

Value

This function returns NULL.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Load Database: phrLoadDatabaseString()

Examples

```
# create temporary database file
tf <- tempfile()
writeLines(phreeqc.dat, tf)

if (is.null(phrLoadDatabase(tf))) {
  cat("database ok\n")
} else {
  cat("database contains errors\n")
}

# delete temporary database file
unlink(tf)
```

phrLoadDatabaseString *Load a phreeqc database as a string or a list of strings.*

Description

Load the specified string(s) as a database into phreeqc. Returns NULL if successful.

Usage

```
phrLoadDatabaseString(input)
```

Arguments

input	String containing data to be used as the phreeqc database.

Details

All previous definitions are cleared.

Value

This function returns NULL.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Load Database: phrLoadDatabase()

Examples

```
# this example loads the phreeqc.dat database, turns on the
# output file and runs ex2 as a string
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputFileOn(TRUE)
phrSetOutputFileName(file.path(tempdir(), "ex2.load-database.out"))
if (is.null(phrRunString(ex2))) {
  cat(paste("see ", phrGetOutputFileName(), ".\n", sep = ""))
}
```

---

phrRunAccumulated *Runs the accumulated input.*

Description

Runs the input buffer as defined by calls to phrAccumulateLine.

Usage

```
phrRunAccumulated()
```

Details

After calling phrRunAccumulated phrGetAccumulatedLines can be used in case of errors. The accumulated lines will be cleared on the next call to phrAccumulateLine.

The phrAccumulateLine method cannot be called until a database has been successfully loaded by calls to either phrLoadDatabase or phrLoadDatabaseString.

Value

This function returns NULL on success.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Accumulate: phrAccumulateLine(), phrClearAccumulatedLines(), phrGetAccumulatedLines()

Other Run: phrRunFile(), phrRunString()

Examples

```
# turn on the output file
phrSetOutputFileOn(TRUE)
phrSetOutputFileName(file.path(tempdir(), "ex2.out"))

# load the phreeqc.dat database
phrLoadDatabaseString(phreeqc.dat)

# accumulate the input
phrAccumulateLine("TITLE Example 2.--Temperature dependence of solubility")
phrAccumulateLine("                  of gypsum and anhydrite")
phrAccumulateLine("SOLUTION 1 Pure water")
phrAccumulateLine("        pH      7.0")
phrAccumulateLine("        temp    25.0")
phrAccumulateLine("EQUILIBRIUM_PHASES 1")
phrAccumulateLine("        Gypsum          0.0     1.0")
phrAccumulateLine("        Anhydrite       0.0     1.0")
phrAccumulateLine("REACTION_TEMPERATURE 1")
phrAccumulateLine("        25.0 75.0 in 51 steps")
phrAccumulateLine("SELECTED_OUTPUT")
phrAccumulateLine("        -file   ex2.sel")
phrAccumulateLine("        -temperature")
phrAccumulateLine("        -si     anhydrite  gypsum")
phrAccumulateLine("END")

# run it and echo the name of the output file
if (is.null(phrRunAccumulated())) {
  cat(paste("see ", phrGetOutputFileName(), ".\n", sep = ""))
}
```

`phrRunFile` *Run phreeqc input file*

Description

phrRunFile executes a phreeqc run using a file as input

Usage

```
phrRunFile(filename)
```

Arguments

`filename`	The file name of the phreeqc input file.

Value

This function returns NULL on success.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Run: phrRunAccumulated(), phrRunString()

Examples

```
# load the phreeqc.dat database
phrLoadDatabaseString(phreeqc.dat)

# create ex2 if it doesn't exist
if (!file.exists("ex2")) writeLines(ex2, "ex2")

# run ex2
if (is.null(phrRunFile("ex2"))) {
  cat("use phrGetSelectedOutput() to see results.\n")
}

unlink("ex2")  # tidy up
```

phrRunString *Runs phreeqc using the given string as input.*

Description

Runs phreeqc using the given string as input. Returns the number of errors encountered during the run.

Usage

```
phrRunString(input)
```

Arguments

input	character vector containing phreeqc input

Details

The RunString method cannot be called until a database has been successfully loaded by one of the following the LoadDatabase methods phrLoadDatabase, phrLoadDatabaseString.

Value

This function returns NULL on success.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Run: phrRunAccumulated(), phrRunFile()

Examples

```
#
# This example accumulates phreeqc input into a character vector
# and runs it.
#

# load phreeqc.dat file
phrLoadDatabaseString(phreeqc.dat)

# create input
input <- vector()
input <- c(input, "SOLUTION 1")
input <- c(input, "  temp 25.0")
input <- c(input, "  pH    7.0")

# turn on output
phrSetOutputFileOn(TRUE)
phrSetOutputFileName(file.path(tempdir(), "pure_water.out"))

# run input
phrRunString(input)
cat(paste("see ", phrGetOutputFileName(), ".\n", sep=""))
```

phrSetDumpFileName *Set the name of the dump file.*

Description

Sets the name of the dump file. This file name is used if not specified within the DUMP keyword block. The default value is dump.0.out.

Usage

```
phrSetDumpFileName(filename)
```

Arguments

filename	the name of the file.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Dump: phrGetDumpFileName(), phrGetDumpFileOn(), phrGetDumpStrings(), phrGetDumpStringsOn(), phrSetDumpFileOn(), phrSetDumpStringsOn()

Examples

```
# This example equilibrates pure water with calcite and writes the
# dump results to file.
phrLoadDatabaseString(phreeqc.dat)
phrSetDumpFileOn(TRUE)
phrSetDumpFileName(file.path(tempdir(), "phreeqc.dump"))
input <- c(
  'SOLUTION 1 Pure water     ',
  'EQUILIBRIUM_PHASES 1      ',
  '    Calcite 0 10          ',
  'SAVE solution 1           ',
  'SAVE equilibrium_phases 1 ',
  'DUMP                      ',
  '    -solution 1           ',
  '    -equilibrium_phases 1 '
  )

if (is.null(phrRunString(input))) {
  cat(paste("see ", phrGetDumpFileName(), ".\n", sep = ""))
}
```

---

phrSetDumpFileOn *Set the dump file on/off.*

Description

Sets the dump file switch on or off. This switch controls whether or not phreeqc writes to the dump file. The initial setting is off.

Usage

```
phrSetDumpFileOn(value)
```

Arguments

value	if TRUE, captures output normally sent to the dump file into a buffer.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Dump: phrGetDumpFileName(), phrGetDumpFileOn(), phrGetDumpStrings(), phrGetDumpStringsOn(), phrSetDumpFileName(), phrSetDumpStringsOn()

Examples

```
# This example equilibrates pure water with calcite and writes the
# dump results to file.
phrLoadDatabaseString(phreeqc.dat)
phrSetDumpFileOn(TRUE)
input <- c(
  'SOLUTION 1 Pure water     ',
  'EQUILIBRIUM_PHASES 1      ',
  '    Calcite 0 10          ',
  'SAVE solution 1           ',
  'SAVE equilibrium_phases 1 ',
  'DUMP                      ',
  '    -solution 1           ',
  '    -equilibrium_phases 1 '
  )

if (is.null(phrRunString(input))) {
  cat(paste("see ", phrGetDumpFileName(), ".\n", sep = ""))
}
```

---

phrSetDumpStringsOn *Set dump strings on/off.*

Description

Sets the dump strings switch on or off. This switch controls whether or not the data normally sent to the dump file are stored in a buffer for retrieval. The initial setting is off.

Usage

```
phrSetDumpStringsOn(value)
```

Arguments

value	if TRUE, captures output normally sent to the error file into a buffer.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Dump: phrGetDumpFileName(), phrGetDumpFileOn(), phrGetDumpStrings(), phrGetDumpStringsOn(), phrSetDumpFileName(), phrSetDumpFileOn()

Examples

```
# This example equilibrates pure water with calcite and echos the
# dump strings.
phrLoadDatabaseString(phreeqc.dat)
phrSetDumpStringsOn(TRUE)
input <- c(
  'SOLUTION 1 Pure water     ',
  'EQUILIBRIUM_PHASES 1      ',
  '    Calcite 0 10          ',
  'SAVE solution 1           ',
  'SAVE equilibrium_phases 1 ',
  'DUMP                      ',
  '    -solution 1           ',
  '    -equilibrium_phases 1 '
  )

if (is.null(phrRunString(input))) {
  cat("Dump:\n")
  cat(phrGetDumpStrings(), sep = "\n")
}
```

phrSetErrorFileName *Set the name of the error file.*

Description

Sets the name of the error file. The default value is phreeqc.0.err.

Usage

```
phrSetErrorFileName(filename)
```

Arguments

filename	the name of the file.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Error: phrGetErrorFileName(), phrGetErrorFileOn(), phrGetErrorStrings(), phrGetErrorStringsOn(), phrSetErrorFileOn(), phrSetErrorStringsOn()

Examples

```
# This example attempts to run a input string, fails and writes
# the error message to the error file (no database is loaded).
phrSetErrorFileOn(TRUE)
phrSetErrorFileName(file.path(tempdir(), "phreeqc.errors"))
input <- c(
  'SOLUTION 1 Pure water ',
  'EQUILIBRIUM_PHASES 1  ',
  '    Calcite 0 10      '
  )

if (is.null(phrRunString(input))) {
  cat(paste("see ", phrGetErrorFileName(), ".\n", sep = ""))
}
```

---

phrSetErrorFileOn *Set error file on/off.*

Description

Sets the error file switch on or off. This switch controls whether or not phreeqc writes to the error file. The initial setting is off.

Usage

```
phrSetErrorFileOn(value)
```

Arguments

value	if TRUE, writes output to the the error file.

Details

The try is necessary to keep the error message from displaying immediately.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Error: phrGetErrorFileName(), phrGetErrorFileOn(), phrGetErrorStrings(), phrGetErrorStringsOn(), phrSetErrorFileName(), phrSetErrorStringsOn()

Examples

```
# This example attempts to run ex1, fails, and writes the error
# message to the error file (no database is loaded).
phrSetErrorFileOn(TRUE)
phrSetErrorFileName(file.path(tempdir(), "phreeqc.errors"))
if (!is.null(try(phrRunString(ex1), silent=TRUE))) {
  cat(paste("see ", phrGetErrorFileName(), ".\n", sep = ""))
}
```

---

phrSetErrorStringsOn *Set error strings on/off.*

Description

Sets the error strings switch on or off. This switch controls whether or not the data normally sent to the error file are stored in a buffer for retrieval. The initial setting is on.

Usage

```
phrSetErrorStringsOn(value)
```

Arguments

value	if TRUE, captures output normally sent to the error file into a buffer.

Details

The try is necessary to keep the error message from displaying immediately.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Error: phrGetErrorFileName(), phrGetErrorFileOn(), phrGetErrorStrings(), phrGetErrorStringsOn(), phrSetErrorFileName(), phrSetErrorFileOn()

Examples

```
# This example attempts to run ex1, fails, and displays the error message
# (no database is loaded).
phrSetErrorStringsOn(TRUE)
if (!is.null(try(phrRunString(ex1), silent=TRUE))) {
  cat(phrGetErrorStrings(), "\n")
}
```

phrSetLogFileName *Set the name of the log file.*

Description

Sets the name of the log file. The default value is phreeqc.0.log

Usage

```
phrSetLogFileName(filename)
```

Arguments

`filename`	the name of the file.

Details

Logging must be enabled through the use of the KNOBS -logfile option in order to receive any log messages.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Log: `phrGetLogFileName()`, `phrGetLogFileOn()`, `phrGetLogStrings()`, `phrGetLogStringsOn()`, `phrSetLogFileOn()`, `phrSetLogStringsOn()`

Examples

```
# This example equilibrates pure water with calcite and displays
# the log file name.
phrLoadDatabaseString(phreeqc.dat)
phrSetLogFileOn(TRUE)
phrSetLogFileName(file.path(tempdir(), "phreeqc.log"))
input <- c(
  'SOLUTION 1 Pure water ',
  'EQUILIBRIUM_PHASES 1  ',
  '    Calcite 0 10      ',
  'KNOBS                 ',
  '    -logfile true     '
  )

if (is.null(phrRunString(input))) {
  cat(paste("see ", phrGetLogFileName(), ".\n", sep = ""))
}
```

phrSetLogFileOn *Set log file on/off.*

Description

Sets the log file switch on or off. This switch controls whether or not phreeqc writes log messages to the log file. The initial setting is off.

Usage

```
phrSetLogFileOn(value)
```

Arguments

value	if TRUE, writes output to the the log file.

Details

Logging must be enabled through the use of the KNOBS -logfile option in order to receive an log messages.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Log: phrGetLogFileName(), phrGetLogFileOn(), phrGetLogStrings(), phrGetLogStringsOn(), phrSetLogFileName(), phrSetLogStringsOn()

Examples

```
# This example runs ex2 with the log file turned on.
phrLoadDatabaseString(phreeqc.dat)
phrSetLogFileOn(TRUE)
phrSetLogFileName(file.path(tempdir(), "ex2.log"))

# turn logging on
phrAccumulateLine("KNOBS; -logfile true")
phrRunAccumulated()

if (is.null(phrRunString(ex2))) {
  cat(paste("see ", phrGetLogFileName(), ".\n", sep = ""))
}
```

phrSetLogStringsOn *Set log strings on/off.*

Description

Sets the log strings switch on or off. This switch controls whether or not the data normally sent to the log file are stored in a buffer for retrieval. The initial setting is off.

Usage

```
phrSetLogStringsOn(value)
```

Arguments

value	if TRUE, captures output normally sent to the log file into a buffer.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Log: phrGetLogFileName(), phrGetLogFileOn(), phrGetLogStrings(), phrGetLogStringsOn(), phrSetLogFileName(), phrSetLogFileOn()

Examples

```
# This example runs ex2 with log strings turned on.
phrLoadDatabaseString(phreeqc.dat)
phrSetLogStringsOn(TRUE)

# turn logging on
phrAccumulateLine("KNOBS; -logfile true")
phrRunAccumulated()

if (is.null(phrRunString(ex2))) {
  cat(phrGetLogStrings(), sep = "\n")
}
```

phrSetOutputFileName *Set the name of the output file.*

Description

Sets the name of the output file. The default value is phreeqc.0.out.

Usage

```
phrSetOutputFileName(filename)
```

Arguments

filename	the name of the file.

Details

The output file must be turned on using the phrSetOutputFileOn function.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Output: phrGetOutputFileName(), phrGetOutputFileOn(), phrGetOutputStrings(), phrGetOutputStringsOn(), phrSetOutputFileOn(), phrSetOutputStringsOn()

Examples

```
# This example equilibrates pure water with calcite and displays
# the resulting file name.
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputFileOn(TRUE)
phrSetOutputFileName(file.path(tempdir(), "calcite.out"))
input <- c(
  'SOLUTION 1 Pure water ',
  'EQUILIBRIUM_PHASES 1  ',
  '    Calcite 0 10      '
  )

if (is.null(phrRunString(input))) {
  cat(paste("see ", phrGetOutputFileName(), ".\n", sep = ""))
}
```

phrSetOutputFileOn *Set output file on/off.*

Description

Sets the output file switch on or off. This switch controls whether or not phreeqc writes to the output file. This is the output normally generated when phreeqc is run. The initial setting is off.

Usage

```
phrSetOutputFileOn(value)
```

Arguments

value	if TRUE, writes output to the the output file.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Output: phrGetOutputFileName(), phrGetOutputFileOn(), phrGetOutputStrings(), phrGetOutputStringsOn(), phrSetOutputFileName(), phrSetOutputStringsOn()

Examples

```
# This example runs ex2 with the output file turned on.

# write temporary input file
tf <- tempfile()
writeLines(ex2, tf)

# load database and run input file
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputFileOn(TRUE)
phrSetOutputFileName(file.path(tempdir(), "ex2.out"))
if (is.null(phrRunFile(tf))) {
  cat(paste("see ", phrGetOutputFileName(), ".\n", sep = ""))
}

# delete temporary input file
unlink(tf)
```

phrSetOutputStringsOn *Set output strings on/off.*

Description

Set the output string switch on or off. This switch controls whether or not the data normally sent to the output file are stored in a buffer for retrieval. The initial setting is off.

Usage

```
phrSetOutputStringsOn(value)
```

Arguments

value	if TRUE, captures output normally sent to the output file into a buffer.

Details

The output strings setting is only used by the Run methods: phrRunAccumulated, phrRunFile, phrRunString.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Output: phrGetOutputFileName(), phrGetOutputFileOn(), phrGetOutputStrings(), phrGetOutputStringsOn(), phrSetOutputFileName(), phrSetOutputFileOn()

Examples

```
# This example equilibrates pure water with calcite and displays
# the results.
phrLoadDatabaseString(phreeqc.dat)
phrSetOutputStringsOn(TRUE)
input <- c(
  'SOLUTION 1 Pure water ',
  'EQUILIBRIUM_PHASES 1  ',
  '    Calcite 0 10      '
  )

if (is.null(phrRunString(input))) {
  cat(phrGetOutputStrings(), sep = "\n")
}
```

phrSetSelectedOutputFileName

Set the name of the selected_output file.

Description

Sets the name of the selected_output file. The default value is selected_{nuser}.0.out.

Usage

```
phrSetSelectedOutputFileName(nuser, filename)
```

Arguments

nuser	the user number specified in the SELECTED_OUTPUT block.
filename	the name of the selected_output file.

Details

The selected_output file must be turned on using the phrSetSelectedOutputFileOn function.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Selected Output: phrGetSelectedOutput(), phrGetSelectedOutputFileName(), phrSetSelectedOutputFileOn(

Examples

```
# This example equilibrates pure water with calcite at various temperatures
# and displays the name of the selected_output file.
phrLoadDatabaseString(phreeqc.dat)
phrSetSelectedOutputFileOn(1, TRUE)
phrSetSelectedOutputFileName(1, file.path(tempdir(), "ex2.sel"))

input <- c(
  'SOLUTION 1 Pure water     ',
  'EQUILIBRIUM_PHASES 1      ',
  '    Calcite    0.0   1.0  ',
  'REACTION_TEMPERATURE 1    ',
  '    25.0 75.0 in 51 steps ',
  'SELECTED_OUTPUT 1         ',
  '    -temperature          ',
  '    -si     calcite       '
  )
```

```
if (is.null(phrRunString(input))) {
  cat(paste("see ", phrGetSelectedOutputFileName(1), ".\n", sep = ""))
}
```

`phrSetSelectedOutputFileOn`

Set selected_output file on/off.

Description

Sets the output file switch on or off. This switch controls whether or not phreeqc writes to the output file. This is the output normally generated when phreeqc is run. The initial setting is off.

Usage

```
phrSetSelectedOutputFileOn(nuser, value)
```

Arguments

`nuser`	the user number specified in the SELECTED_OUTPUT block.
`value`	if TRUE, writes output to the the selected_output file.

References

https://water.usgs.gov/water-resources/software/PHREEQC/IPhreeqc.pdf

See Also

Other Selected Output: `phrGetSelectedOutput()`, `phrGetSelectedOutputFileName()`, `phrSetSelectedOutputFileNam`

Examples

```
# This example equilibrates pure water with calcite at various temperatures
# and displays the name of the selected_output file.
phrLoadDatabaseString(phreeqc.dat)
phrSetSelectedOutputFileOn(1, TRUE)
phrSetSelectedOutputFileName(1, file.path(tempdir(), "ex2.sel"))

input <- c(
  'SOLUTION 1 Pure water     ',
  'EQUILIBRIUM_PHASES 1      ',
  '    Calcite    0.0   1.0  ',
  'REACTION_TEMPERATURE 1    ',
  '    25.0 75.0 in 51 steps ',
  'SELECTED_OUTPUT 1         ',
  '    -temperature          ',
  '    -si     calcite       '
  )
```

```
if (is.null(phrRunString(input))) {
  cat(paste("see ", phrGetSelectedOutputFileName(1), ".\n", sep = ""))
}
```

pitzer.dat *The pitzer.dat database.*

Description

pitzer.dat is a database for the specific-ion-interaction model of Pitzer as implemented in PHRQPITZ. The database has been reformatted for use by phrLoadDatabaseString.

Usage

```
pitzer.dat  # phrLoadDatabaseString(pitzer.dat)
```

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Databases: Amm.dat, ColdChem.dat, Kinec.v2.dat, Kinec_v3.dat, PHREEQC_ThermoddemV1.10_15Dec2020.dat, Tipping_Hurley.dat, core10.dat, ex15.dat, frezchem.dat, iso.dat, llnl.dat, minteq.dat, minteq.v4.dat, phreeqc.dat, phreeqc_rates.dat, sit.dat, wateq4f.dat

sit.dat *The sit.dat database.*

Description

sit.dat is a database derived from databases for EQ3/6 and Geochemist's Workbench that uses thermodynamic data compiled by the Lawrence Livermore National Laboratory. The database has been reformatted for use by phrLoadDatabaseString.

Usage

```
sit.dat  # phrLoadDatabaseString(sit.dat)
```

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Databases: Amm.dat, ColdChem.dat, Kinec.v2.dat, Kinec_v3.dat, PHREEQC_ThermoddemV1.10_15Dec2020.dat, Tipping_Hurley.dat, core10.dat, ex15.dat, frezchem.dat, iso.dat, llnl.dat, minteq.dat, minteq.v4.dat, phreeqc.dat, phreeqc_rates.dat, pitzer.dat, wateq4f.dat

---

Tipping_Hurley.dat *The Tipping_Hurley.dat database*

Description

Tipping_Hurley.dat is a database for organic-ligand binding approximating WHAM by Tipping and Hurley. phrLoadDatabaseString.

Usage

```
Tipping_Hurley.dat  # phrLoadDatabaseString(Tipping_Hurley.dat)
```

See Also

Other Databases: Amm.dat, ColdChem.dat, Kinec.v2.dat, Kinec_v3.dat, PHREEQC_ThermoddemV1.10_15Dec2020.dat, core10.dat, ex15.dat, frezchem.dat, iso.dat, llnl.dat, minteq.dat, minteq.v4.dat, phreeqc.dat, phreeqc_rates.dat, pitzer.dat, sit.dat, wateq4f.dat

---

wateq4f.dat *The wateq4f.dat database.*

Description

wateq4f.dat is a database derived from WATEQ4F. The database has been reformatted for use by phrLoadDatabaseString.

Usage

```
wateq4f.dat  # phrLoadDatabaseString(wateq4f.dat)
```

Source

https://www.usgs.gov/software/phreeqc-version-3/

References

https://pubs.usgs.gov/tm/06/a43/pdf/tm6-A43.pdf

See Also

Other Databases: Amm.dat, ColdChem.dat, Kinec.v2.dat, Kinec_v3.dat, PHREEQC_ThermoddemV1.10_15Dec2020.dat, Tipping_Hurley.dat, core10.dat, ex15.dat, frezchem.dat, iso.dat, llnl.dat, minteq.dat, minteq.v4.dat, phreeqc.dat, phreeqc_rates.dat, pitzer.dat, sit.dat

Index