Linking GoldSim with the PHREEQC Geochemical Model with a Dynamic Link Library Element

Ted Eary

MWH Americas, Inc.

Fort Collins, Colorado

(ted.eary@mwhglobal.com or ted_eary@yahoo.com)

GoldSim 2007 User Conference, San Francisco

Abstract

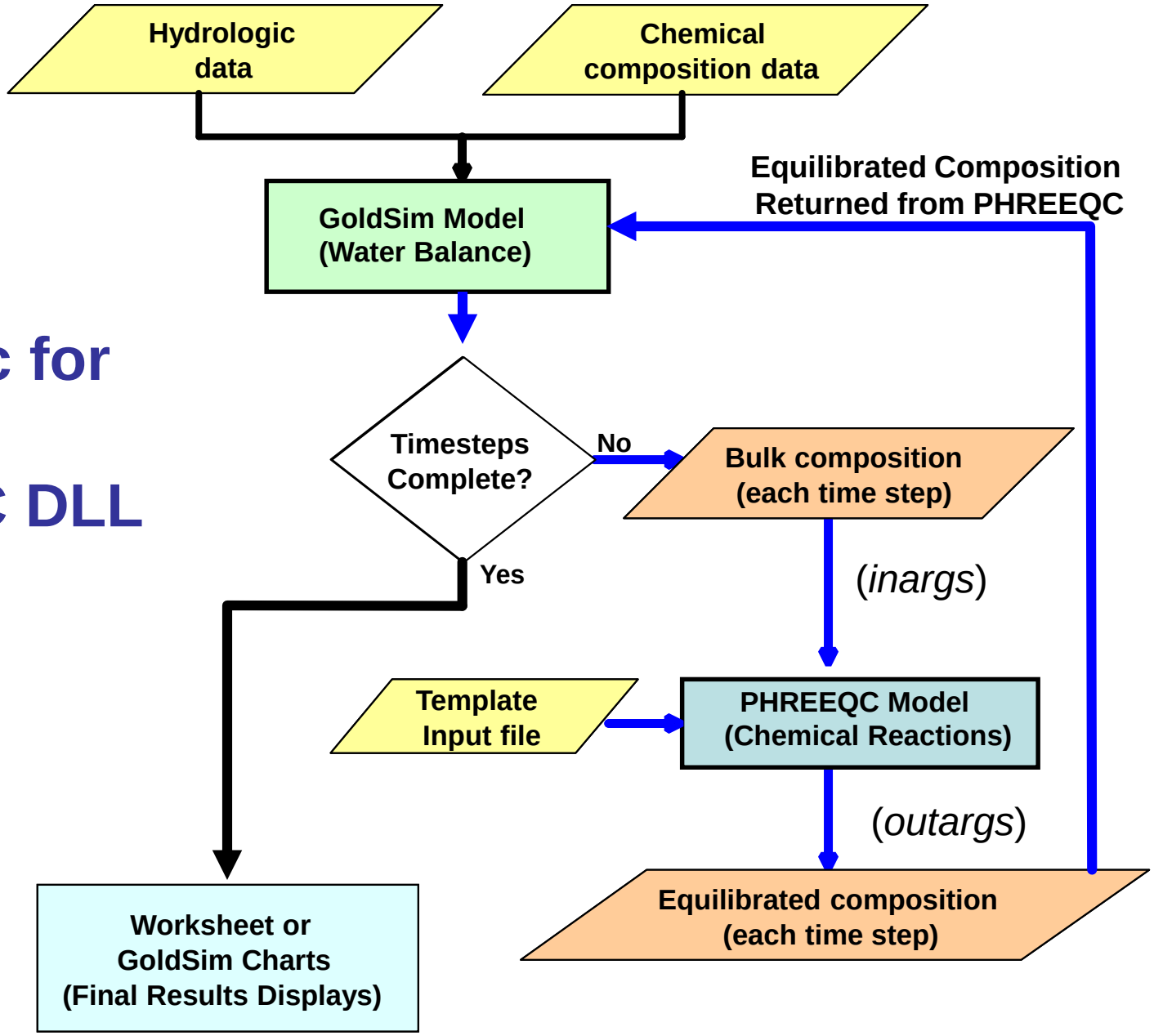
The GoldSim contaminant transport module provides capabilities for simulating chemical processes such as partitioning and solubilities. While flexible and easily configurable, these simple processes do not represent complete reaction paths, hence additional chemical relationships are sometimes needed to model complex hydrochemical systems. A versatile approach for representing complex chemical processes of chemical equilibration and aqueous speciation is to rely on GoldSim's dynamic link library (DLL) element. The purpose of this poster is to describe how to set up and use the DLL element to work with the PHREEQC geochemical model.

The approach for using an external DLL is based on the following steps for the flow of information to and from PHREEQC:

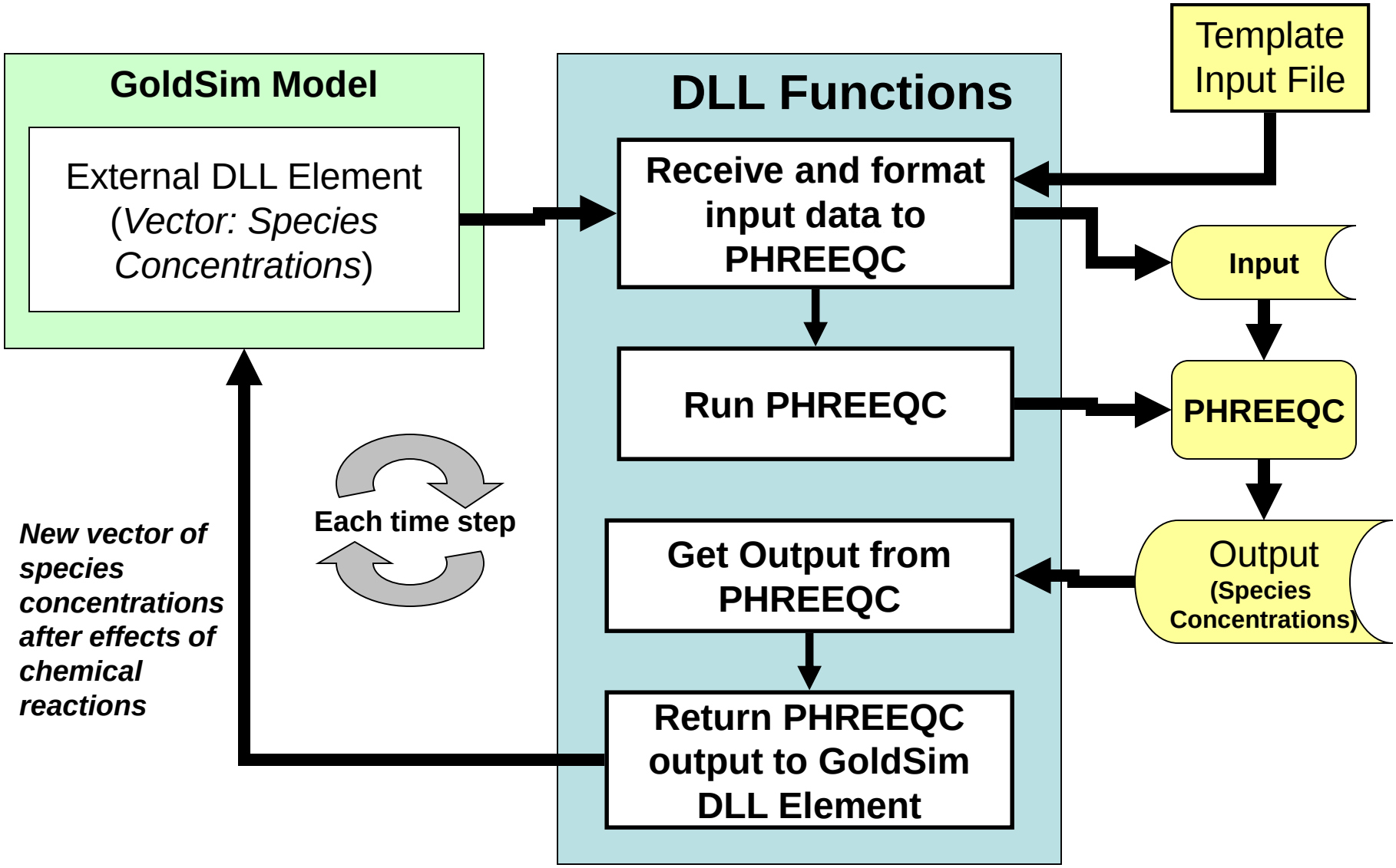
- A vector of species concentrations representing a complete solution is defined in the GoldSim model.
- The species vector is passed to GoldSim's DLL element as the input argument to the DLL interface.
- The GoldSim DLL element sends the species vector to the external DLL.
- The DLL receives the species vector and combines it with a template file to build an input file for PHREEQC.
- The DLL executes PHREEQC with the asynchronous CreateProcess method.
- The DLL captures the execution thread for PHREEQC and waits for it to run to completion.
- The DLL reads the PHREEQC output file and returns the new solution composition back to GoldSim through the output interface definition of the DLL element.
- The above steps are repeated for each time step.

The DLL is written in C++ and compiled with Microsoft Visual C++ Studio. This approach has the advantage of providing a generalized procedure to model complex chemical processes at each time step.

Data Flow Schematic for Using the PHREEQC DLL



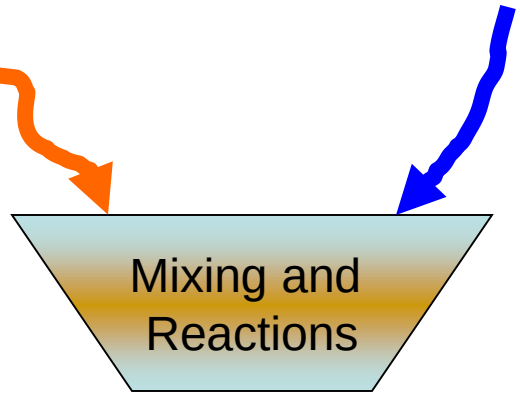
DLL Data Flow Between GoldSim and PHREEQC



Example: Solution Mixing Chemistry

Acidic pH,
High metal water,
Runoff (variable flow rate)

Alkaline pH,
Clean water,
Constant flow rate

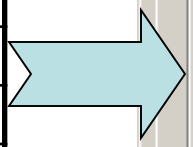


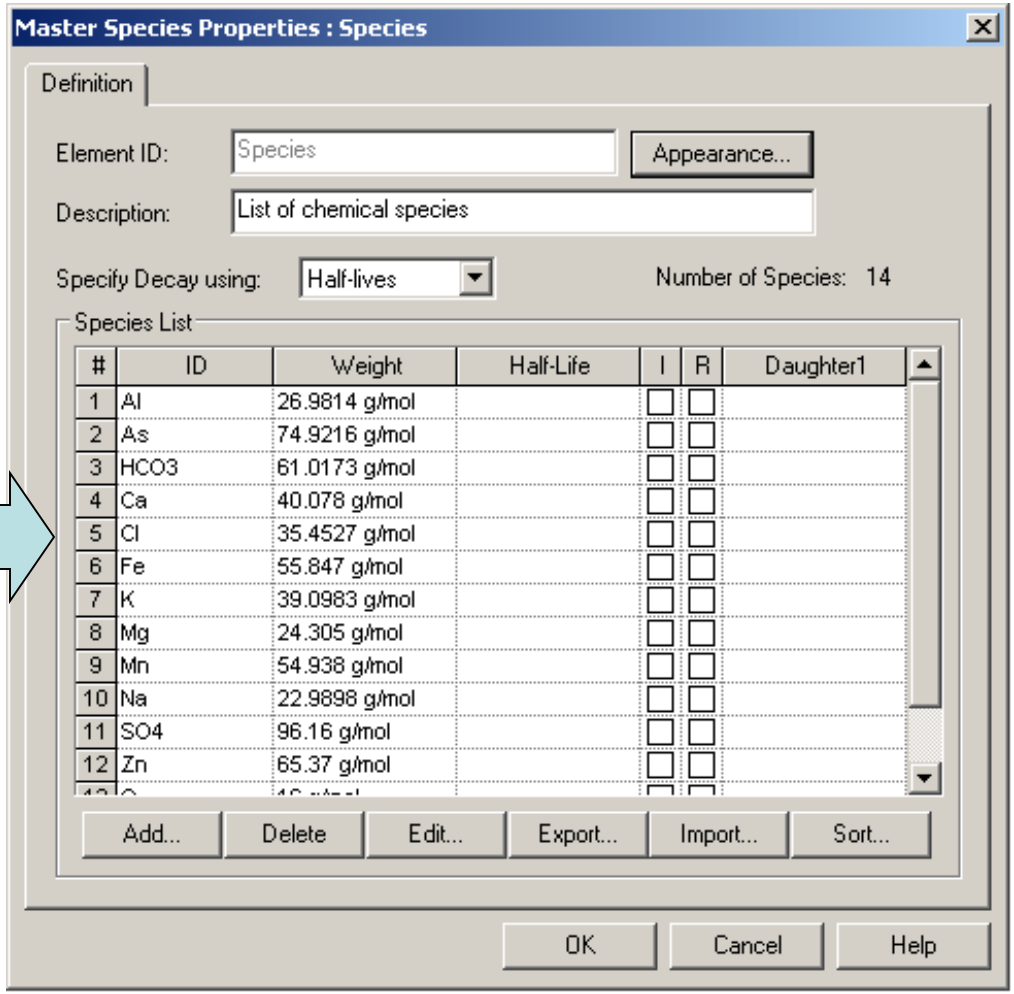
***Problem*: How to calculate solution chemistry after mixing and reactions?**
***Solution*: Use DLL to link to PHREEQC.**

Steps for Setup

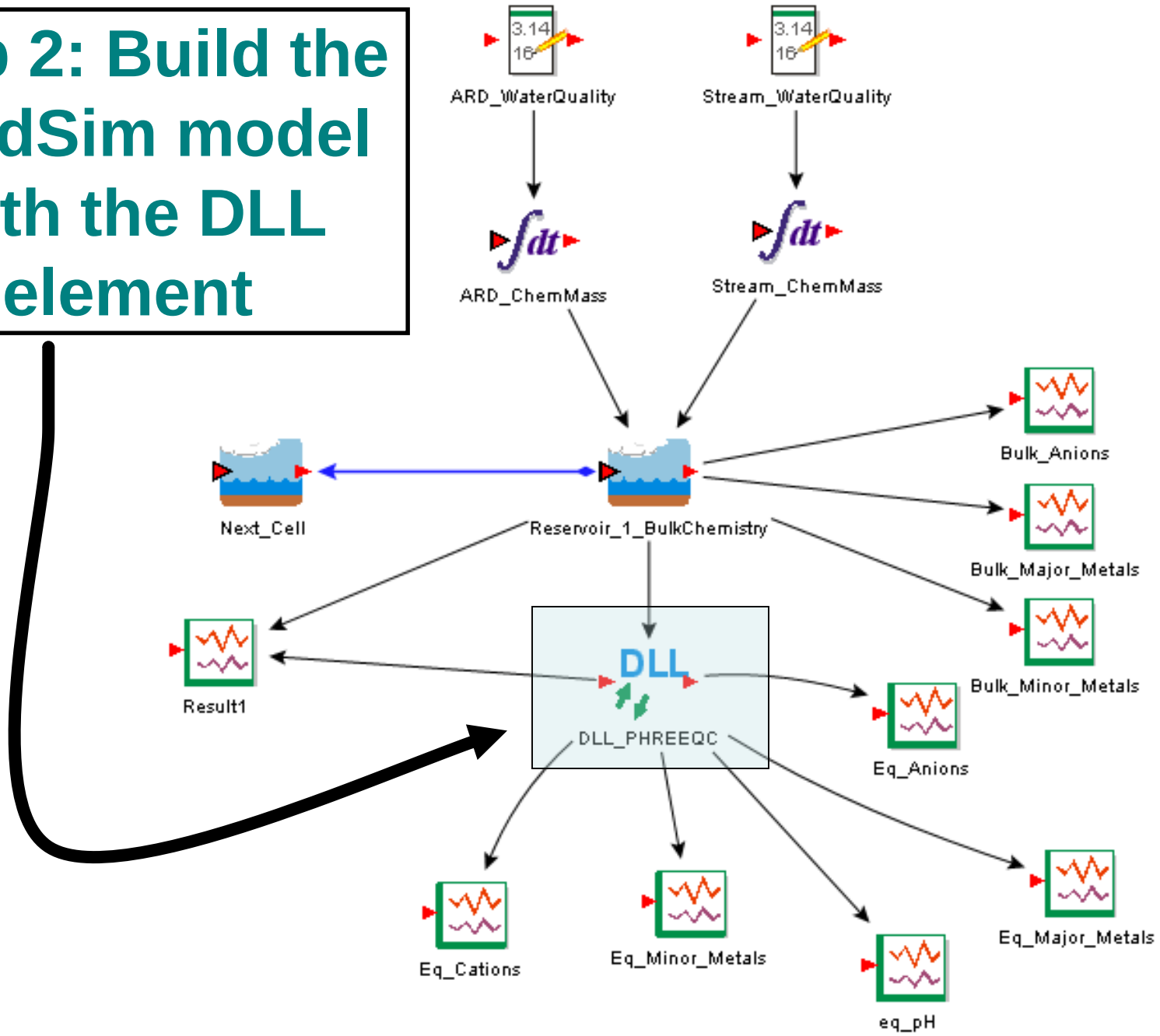
Step 1: Define species and starting solutions (must be charge-balanced)

Analyte	Acidic Runoff	Alkaline Streamflow
Al, mg/L	180	0.005
As, mg/L	0.5	0.0001
HCO_3, mg/L	0.001	130
Ca, mg/L	550	40
Cl, mg/L	50	5
Fe, mg/L	60	0.01
K, mg/L	40	8
Mg, mg/L	375	22
Mn, mg/L	8.4	0.2
Na, mg/L	60	12
SO_4, mg/L	4117	107.8
Zn, mg/L	2.1	0.0005
O, mg/L	8	8
pH, s.u.	3.09	8.36

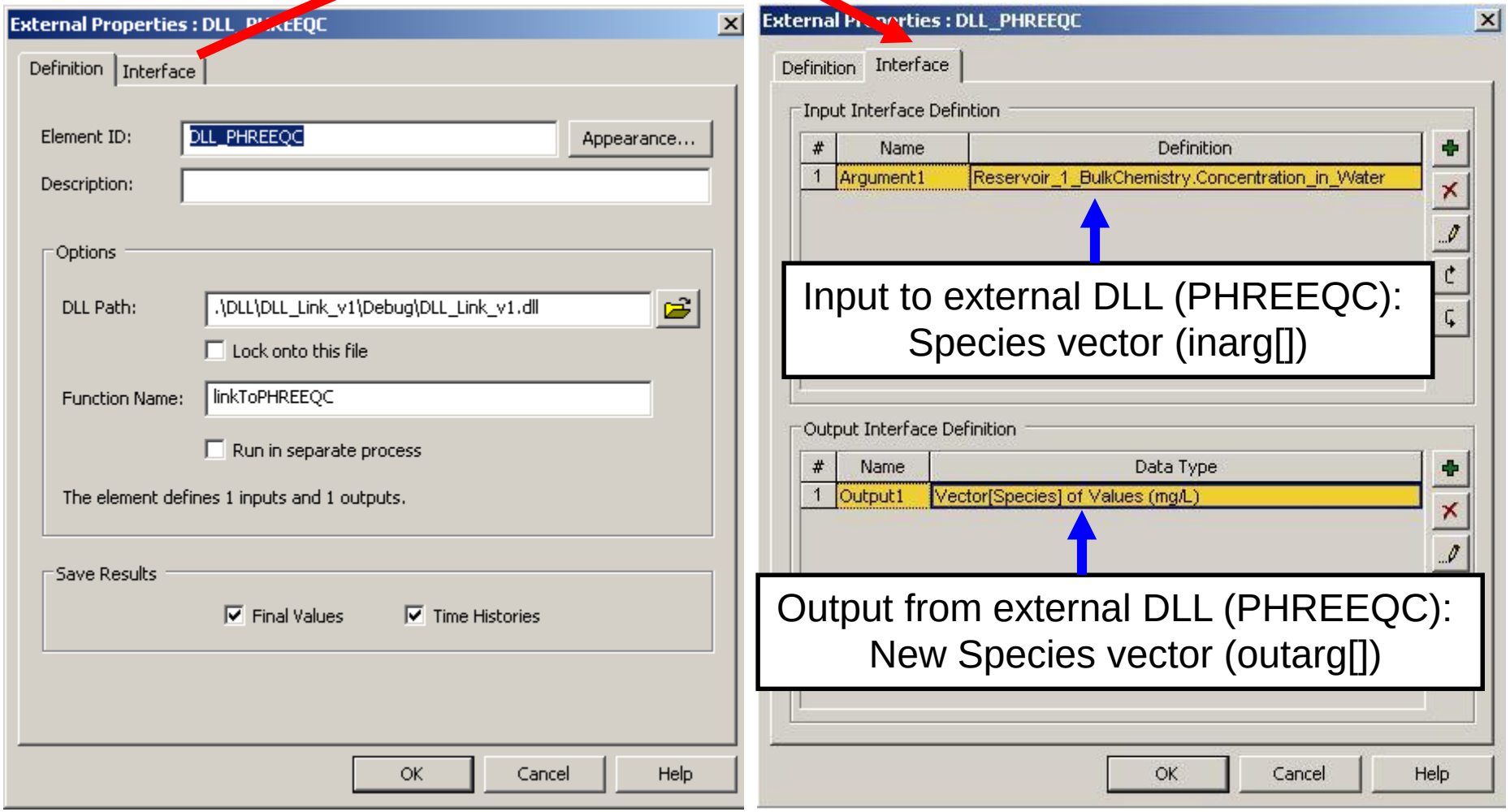




Step 2: Build the GoldSim model with the DLL element



Step 3: Define the DLL element properties



Step 4: Edit the DLL code to match the number of chemical species

```
// Exposed function for use in the GoldSim DLL.
extern "C" void __declspec(dllexport) linkToPHREEQC(int XFMethod, int *XFState, const double inarg[], double *outarg)
{
  int i;
  //Follow standard logic outlined in the GoldSim manual for c++, page 637-638.
  switch (XFMethod)
  {
    case XF_INITIALIZE:
       break;
    case XF_REP_VERSION:
       outarg[0] = VERSN;
       break;
    case XF_REP_ARGUMENTS:
       outarg[0] = 14;    //number of inargs (equal to the number of chemical species in GoldSim)
       outarg[1] = 14;    //number of outargs (equal to the number of chemical species returned to GoldSim)
       numArgs = 14;    //Define this as a global variable for use elsewhere
       break;
    case XF_CALCULATION:
       ReadTemplate(inarg, numArgs);  // Read the template file for PHREEQC input
       RunPHREEQC();                  // Run PHREEQC from here
       GetPHREEQCOutput();            // Get the PHREEQC output to return to GoldSim
       //The following loop sets the return values (outarg) equal to the equilibrated concentrations from PHREEQC output
       for(i = 0; i < numArgs; i++)
       {  outarg[i] = eqConcentration[i]; }   //eqConcentrations[] is a global variable
       break;
    case XF_CLEANUP:
       break;
  }
  return;
}
```

inarg[]: Species vector from GoldSim.
outarg[]: Species vector returned to GoldSim

DLL Functions

Step 5: Revise the PHREEQC Input Template Read by the DLL

```
TITLE  PHREEQC input file template for DLL Linkage
SELECTED_OUTPUT
        -file D:\PHREEQC\phreeqc_output.sel
        -ionic_strength True
        -charge_balance True
        -distance False
        -time False
        -mu False
        -step False
        -simulation False
        -temperature False
USER_PUNCH
        headings Al(mg/L) As(mg/L) HCO3(mg/L) Ca(mg/L) Cl(mg/L) Fe(mg/L) K(mg/L) Mg(mg/L)        Mn(mg/L) Na(mg/L) SO4(mg/L) Zn(mg/L) O(mg/L)
 -start
        5       Al_ppm = TOT("Al") *26.9814* 1000
        10      As_ppm = TOT("As") *74.9216* 1000
        15      HCO3_ppm = MOL("HCO3-")*61.0173* 1000
        20      Ca_ppm = TOT("Ca") *40.078* 1000
        25      Cl_ppm = TOT("Cl") *35.4527* 1000
        30      Fe_ppm = TOT("Fe") *55.847* 1000
        35      K_ppm = TOT("K") *39.0983* 1000
        40      Mg_ppm = TOT("Mg") *24.305* 1000
        45      Mn_ppm = TOT("Mn") *54.938* 1000
        50      Na_ppm = TOT("Na") *22.9898* 1000
        55      SO4_ppm = TOT("S")*96.16* 1000
        60      Zn_ppm = TOT("Zn") *65.37* 1000
        65      O_ppm = TOT("O(0)")*16* 1000
        75      PUNCH  Al_ppm
        80      PUNCH  As_ppm
        85      PUNCH  HCO3_ppm
        90      PUNCH  Ca_ppm
        95      PUNCH  Cl_ppm
        100     PUNCH  Fe_ppm
        105     PUNCH  K_ppm
        110     PUNCH  Mg_ppm
        115     PUNCH  Mn_ppm
        120     PUNCH  Na_ppm
        125     PUNCH  SO4_ppm
        130     PUNCH  Zn_ppm
        135     PUNCH  O_ppm
 -end
```

This line defines the PHREEQC output file from which the DLL returns results back to GoldSim

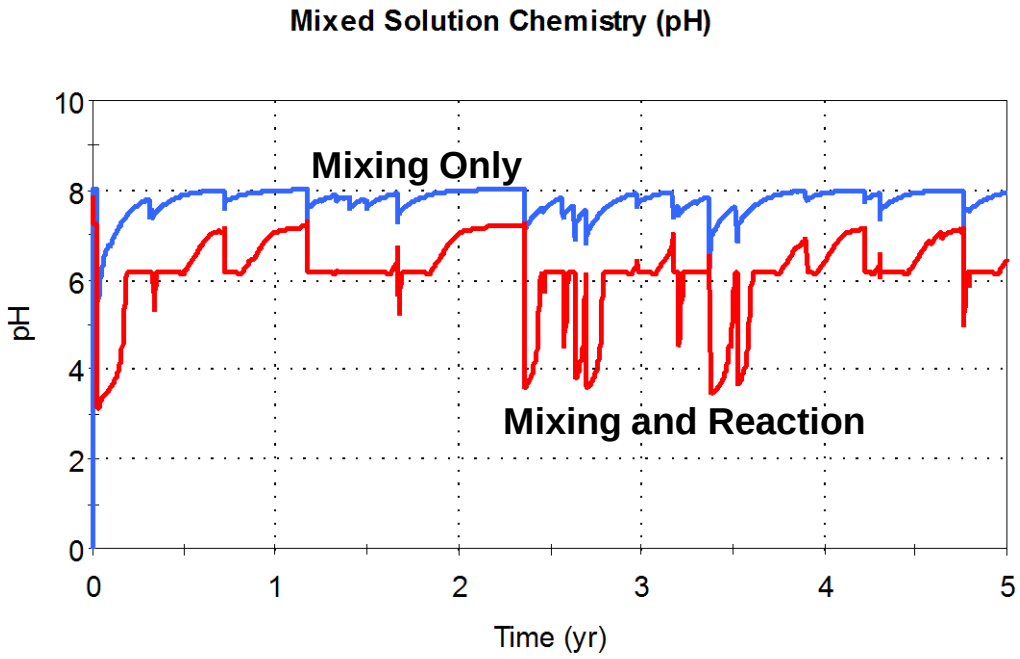
These lines convert PHREEQC outputs in mole/liter to mg/liter.

```
#-----------------------------------------------------------------------------
SURFACE 1
Hfo_sOH 0.000005 600 0.1
Hfo_wOH  0.0002
#-----------------------------------------------------------------------------
SOLUTION 1
  units mg/L
  temperature             12
  redox                   O(0)/O(-2)

-GoldSimStart
        Al                1.1255E+1
        As                3.1344E-2
        C(4)              1.2188E+2
        Ca                7.1875E+1
        Cl                7.8125E+0
        Fe                3.7594E+0
        K                 1.0E+1
        Mg                4.4063E+1
        Mn                7.125E-1
        Na                1.5E+1
        S(6)              3.5838E+2
        Zn                1.3172E-1
        O(0)              8.E+0
-GoldSimEnd

 pH    7.00   charge
EQUILIBRIUM_PHASES
  Gypsum                0                               0
  Calcite               0                               0
  CO2(g)                -2.9                            1
  O2(g)  -0.7                           1
  Ferrihydrite          0                               0
  Jurbanite             0                               0
  Al(OH)3(a)            0                               0
USE Surface 1
END
```

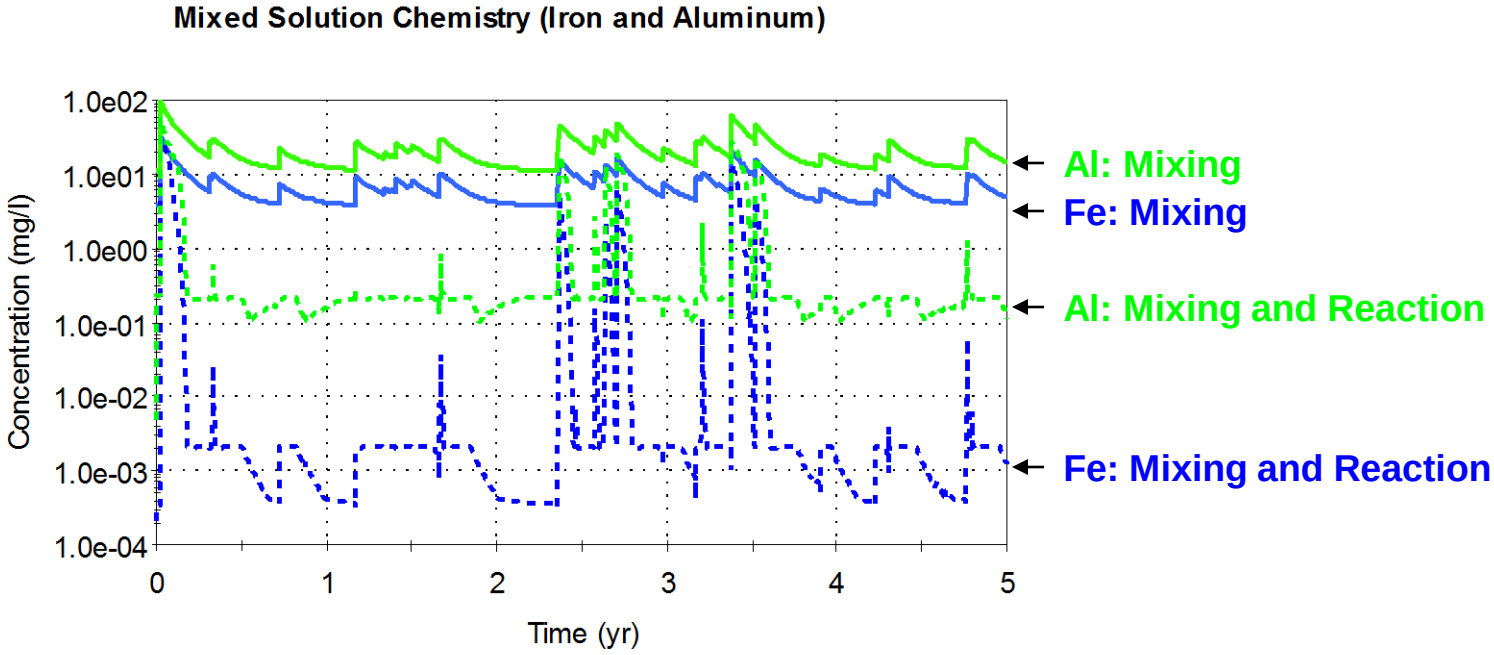
The DLL places concentrations from the species vector between the

"-GoldSimStart" and
"-GoldSimEnd" tags.

All the other lines are used unchanged from this template file to build the PHREEQC input file.



Example Results



Summary

***Advantages*:**

- **Complete integration of the PHREEQC geochemical model with a GoldSim model.**
- **Generalized procedure to model complex chemical processes.**
- **Provides the ability to include the effects of large number of different types of chemical processes on solution compositions on a per time-step basis.**

***Disadvantages*:**

- **Cannot not use the mass balance features of GoldSim.**
- **Run times increased because of overhead to run DLL, which in turn runs PHREEQC and returns output.**
- **Has not been extensively tested except for relatively simple systems, so robustness is still a question.**
- **Requires some knowledge of C++.**

Example available for download at: http://www.goldsim.com/ModuleSiteMap.asp
Model ID: 57 – Linking Geochemical Codes to GoldSim