

Peter Bjorklund

August 10, 2006

09006-ORC-J

## 35. HSC DLL TOOLS

HSC DLL Tools have been developed for Visual Basic 6 applications, however, they may also be used in other programming environments, such as VBA, Visual C++ and Delphi. However, HSC DLL Tools **have only been tested with Visual Basic 6.**

With HSC Chemistry 7.0 it is possible to access the HSC database or the Gibbs module directly from a user program through a number of available subroutines. This is achieved by first linking the **HSC7.dll** or **Gibbs5.dll** (Dynamic Link Library) file located in the Windows System directory. Any computer running a program utilizing the above DLL-files, **must have a legal version of HSC Chemistry 5.1 or above installed.**

If you create your own thermochemical applications with HSC DLL Tools which you would also like to pass on to others, please send these to Outotec:

Antti Roine  
Outotec Research Oy  
P.O. Box 69  
FIN-28101 PORI, FINLAND  
Fax: +358 - 2 - 626 - 5310  
Email: hsc@outotec.com

We may add your applications to future versions of HSC Chemistry, if these applications seem to have general interest.

**Important Note:** In new HSC Chemistry 7.0 the HSC5.DLL and Gibbs5.DLL have been combined into one file: HSC7.DLL.

### 35.1 Getting Started

The following procedure is the simplest way of including the HSC database and subroutines using Visual Basic:

1. Open the Visual Basic editor.
2. Select **“Project, References”** from the menu.
3. Select **“Browse...”** and locate either **HSC7.dll** or **Gibbs5.dll** from your HSC directory (for example C:\HSC7\HSC7.dll).

The subroutines as well as the HSC database are now available for the programmer. To initialize the subroutines of the HSC7.dll file, use for example the following line:

```
Dim HSC As New HSC7.HSC
```

To initialize the use of the subroutines of the Gibbs5.dll file, use for example the following line:

```
Dim Gibbs As New Gibbs5.Gibbs
```

Peter Bjorklund

August 10, 2006

09006-ORC-J

To call a subroutine, for example to retrieve the enthalpy (H) of the species "CO2(g)" at temperature 100 (Celsius or Kelvin), the H subroutine may be used:

Call HSC.H("CO2(g)", 100, H)

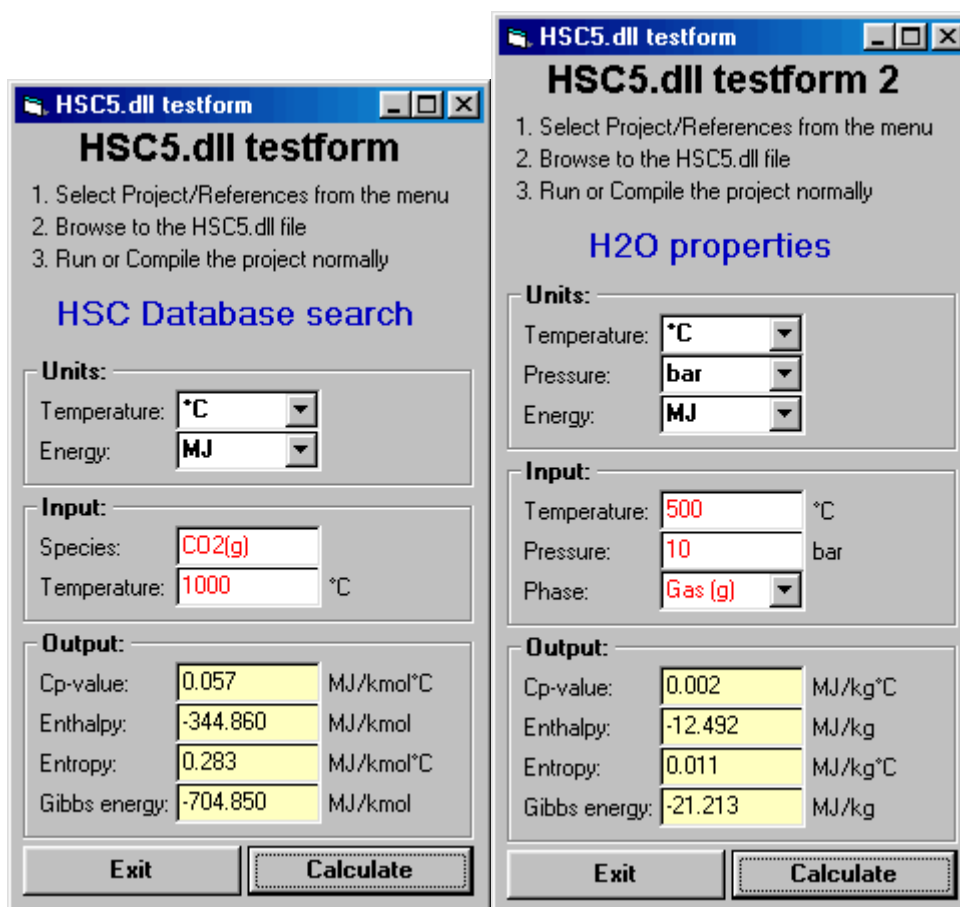
Changing the units to Kelvin, MPa and MJ is achieved by calling the Units subroutine:

Call HSC.Units("K", "MPa", "MJ")

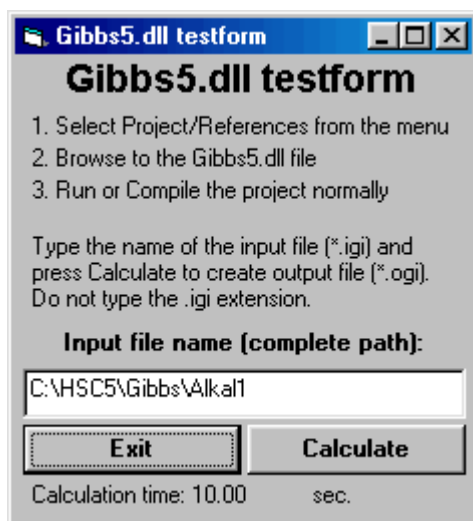
A complete description of the available subroutines is given in Chapter 35.3.

## 35.2 Example Files

There are three Visual Basic 6.0 examples included in the HSC Chemistry CD located in the **HSC DLL Tools\VB6** folder. These are **HSC\_Test1**, **HSC\_Test2** and **Gibbs\_Test1**. **HSC\_Test1** demonstrates how H, S, Cp and G data for a selected species may be displayed in an easy way. **HSC\_Test2** demonstrates the thermodynamic properties of water. **Gibbs\_Test1** demonstrates the use of the Gibbs module by creating .ogi files (output files used by the Pic program) from .igi files (input files created by the Equilibrium Module). Remember to follow the three steps given in Chapter 35.1 in order to test the examples.



**Figure 1:** Screenshots of the *HSCtest1* and *HSCtest2* VB programs (uses *HSC5.dll*).



**Figure 2:** Screenshot of the *Gibbs\_Test1* VB program (uses *Gibbs5.dll*).

The source code of *HSCTest1* is the following:

```

Dim HSC As New HSC5.HSC ' Initialize dll reference

Private Sub ButCalc_Click()

' This event calculates the outputs according to the inputs.
' It uses the file HSC5.dll and its subroutines to read the
' HSC database.
' Company:                Outokumpu Research Oy
' Product:                HSC Chemistry 5.1
' Author:                 Peter Björklund
' Last Changed:          19.9.2002

' Specify inputs
spec$ = Trim$(txtIn1.Text)      ' Input species
T# = Val(txtIn2.Text)          ' Input temperature value

' Specify units
unitT$ = cmbT.Text
unitE$ = cmbE.Text

' Change units
lblUnitT.Caption = unitT$
lblUnitE1.Caption = unitE$ & "/kmol" & unitT$
lblUnitE2.Caption = unitE$ & "/kmol"
lblUnitE3.Caption = unitE$ & "/kmol" & unitT$
lblUnitE4.Caption = unitE$ & "/kmol"

' Call dll subroutines
Call HSC.Units(unitT$, "MPa", unitE$) ' Select unit
Call HSC.CP(spec$, T#, CP#)         ' Cp-value
Call HSC.H(spec$, T#, H#)          ' Enthalpy
Call HSC.S(spec$, T#, S#)          ' Entropy
Call HSC.G(spec$, T#, G#)          ' Gibbs energy

' Print output
CP# = CP# / 1000                    ' k(unit) => M(unit)
S# = S# / 1000                      ' k(unit) => M(unit)
txtOut1.Text = Format(CP#, "0.000")
txtOut2.Text = Format(H#, "0.000")
txtOut3.Text = Format(S#, "0.000")
txtOut4.Text = Format(G#, "0.000")

End Sub

Private Sub butExit_Click()

End

End Sub

Private Sub Form_Load()

cmbT.ListIndex = 0                  ' Default temperature unit
cmbE.ListIndex = 0                  ' Default energy unit

End Sub

```

### 35.3 Description of available Subroutines

The optional argument *ErrorFlag* can be used for checking whether the species in the argument was found from the HSC database. If *ErrorFlag* is > 0 then the species was **not** found from the database.

Peter Bjorklund

August 10, 2006

09006-ORC-J

For the water subroutines it is up to the user to specify the correct phase according to the T and p arguments. If the phase is not correct the return values will simply be extrapolated from the given phase.

The Gibbs5.dll file includes the following subroutine:

Sub Calculatelgi(igi\_file As String, ogi\_file As String, \_  
guessFlag As Integer, fastFlag As Integer, dampenFlag As Integer, \_  
time As Double)

- Creates the output file ogi\_file\$ using the specified flags (true or false) and returns the calculation time in time.

The HSC7.dll file includes the following subroutines:

### General Subs

Sub BAL(Equation As String, Balanced As String)

- Balances equation

Sub Units(Tunit As String, Punit As String, Eunit As String)

- Changes units: "C" or "K", "bar" or "MPa", "Mcal" or "MJ" or "kWh"

Sub Species(DBNo As Double, DBPos As Double, Species As String)

- Returns Species in database DBNo and position DBPos

### Species Subs

Sub H(Species As String, T As Double, H As Double, Optional ErrorFlag as Integer)

- Returns the enthalpy (per kmol) of the species at T (temperature).

Sub S(Species As String, T As Double, S As Double, Optional ErrorFlag as Integer)

- Returns the entropy (per Mmol) of the species at T.

Sub CP(Species As String, T As Double, CP As Double, Optional ErrorFlag as Integer)

- Returns the heat cap. (per Mmol) of the species at T.

Sub G(Species As String, T As Double, G As Double, Optional ErrorFlag as Integer)

- Returns the Gibbs energy (per kmol) of the species at T.

Sub HKG(Species As String, T As Double, HKG As Double, Optional ErrorFlag as Integer)

- Returns the enthalpy (per kg) of the species at T.

Sub HCM(Species As String, T As Double, HCM As Double, Optional ErrorFlag as Integer)

- Returns the enthalpy (per Nm<sup>3</sup>) of the species at T.

Sub HLat(Species As String, T As Double, HLat As Double, Optional ErrorFlag as Integer)

- Returns the latent enthalpy (per kmol) of the species at T.

### Reaction Equation Subs

Sub H(Equation As String, T As Double, H As Double, Optional ErrorFlag as Integer)

Peter Bjorklund

August 10, 2006

09006-ORC-J

- Returns the enthalpy (per kmol) of the reaction equation at T.

Sub S(Equation As String, T As Double, S As Double, Optional ErrorFlag as Integer)

- Returns the entropy (per Mmol) of the reaction equation at T.

Sub CP(Equation As String, T As Double, CP As Double, Optional ErrorFlag as Integer)

- Returns the heat cap. diff. (per Mmol) of the reaction equation at T.

Sub G(Equation As String, T As Double, G As Double, Optional ErrorFlag as Integer)

- Returns the Gibbs energy (per kmol) of the reaction equation at T.

Sub HKG(Equation As String, T As Double, HKG As Double, Optional ErrorFlag as Integer)

- Returns the enthalpy (per kg) of the reaction equation at T.

Sub K(Equation As String, T As Double, K As Double, Optional ErrorFlag as Integer)

- Returns the equilibrium constant of the reaction equation at T.

#### Iteration (reverse) Subs

Sub TatCP(Species As String, CP As Double, TatCP As Double)

- Returns the temperature of the species at given enthalpy (per kmol).

Sub TatG(Species As String, G As Double, TatG As Double)

- Returns the temperature of the species at given entropy (per Mmol).

Sub TatH(Species As String, H As Double, TatH As Double)

- Returns the temperature of the species at given heat cap. (per Mmol).

Sub TatS(Species As String, S As Double, TatS As Double)

- Returns the temperature of the species at given Gibbs en. (per kmol).

#### Water Subs

Sub Critical\_H2O(T As Double, P As Double)

- Returns the temperature and pressure of the critical point.

Sub Triple\_H2O(T As Double, P As Double)

- Returns the temperature and pressure of the triple point.

Sub H\_H2O(T As Double, P As Double, Ph As String, H As Double, Optional ErrorFlag as Integer)

- Returns the enthalpy of water at P, T and phase (“g”, “l”, “s”).

Sub HKG\_H2O(T As Double, P As Double, Ph As String, H As Double, Optional ErrorFlag as Integer)

- Returns the enthalpy of water (per kg) at P, T and phase.

Sub S\_H2O(T As Double, P As Double, Ph As String, S As Double, Optional ErrorFlag as Integer)

- Returns the entropy of water at P, T and phase (“g”, “l”, “s”).

Sub SKG\_H2O(T As Double, P As Double, Ph As String, S As Double, Optional ErrorFlag as Integer)

- Returns the entropy of water (per kg) at P, T and phase.

Peter Bjorklund

August 10, 2006

09006-ORC-J

Sub CP\_H2O(T As Double, P As Double, Ph As String, CP As Double, Optional ErrorFlag as Integer)

- Returns the heat cap. of water at P, T and phase (“g”, “l”, “s”).

Sub CPKG\_H2O(T As Double, P As Double, Ph As String, CP As Double, Optional ErrorFlag as Integer)

- Returns the heat cap. of water (per kg) at P, T and phase (“g”, “l”, “s”).

Sub G\_H2O(T As Double, P As Double, Ph As String, G As Double, Optional ErrorFlag as Integer)

- Returns the Gibbs en. of water at P, T and phase (“g”, “l”, “s”).

Sub GKG\_H2O(T As Double, P As Double, Ph As String, G As Double, Optional ErrorFlag as Integer)

- Returns the Gibbs en. of water (per kg) at P, T and phase (“g”, “l”, “s”).

#### Temperature-independent Subs

Sub Struct(Species As String, Struct As String)

- Returns the structural formula of the species.

Sub ChName(Species As String, ChName As String)

- Returns the chemical name of the species.

Sub CoName(Species As String, CoName As String)

- Returns the common name of the species.

Sub CAN(Species As String, CAN As String)

- Returns the chemical abstract number of the species.

Sub MW(Species As String, MW As Double)

- Returns the molecular weight of the species.

Sub De(Species As String, De As Double)

- Returns the density of the species.

Sub MP(Species As String, MP As Double)

- Returns the melting point of the species.

Sub BP(Species As String, BP As Double)

- Returns the boiling point of the species.

Sub Phase(Species As String, Phase As String)

- Returns the phase of the species.

Sub RGBColor(Species As String, RGBColor As String)

- Returns the RGB color of the species.

Sub REF(Species As String, REF As String)

- Returns the database reference of the species.

Sub REL(Species As String, REL As Double)

Peter Bjorklund

August 10, 2006

09006-ORC-J

- Returns the reliability class of the species.

#### Percentage Subs

Sub MOLP(Species1 As String, Species2 As String, MOLP As Double)

- Returns the mol-% of Species1 in Species2

Sub WTP(Species1 As String, Species2 As String, WTP As Double)

- Returns the wt-% of Species1 in Species2.

#### Cp-data Subs

Sub CPFUNCTION(Species As String, CPFUNCTION As String)

- Returns the heat capacity pol. func. (as a string) of the given species.

Sub CPA(Species As String, CPA As Double)

- Returns the A-coefficient (as a number) of the  $c_p$ -function.

Sub CPB(Species As String, CPB As Double)

- Returns the B-coefficient (as a number) of the  $c_p$ -function.

Sub CPC(Species As String, CPC As Double)

- Returns the C-coefficient (as a number) of the  $c_p$ -function.

Sub CPD(Species As String, CPD As Double)

- Returns the D-coefficient (as a number) of the  $c_p$ -function.

Sub TMAX(Species As String, TMAX As Double)

- Returns the maximum valid temperature of the  $c_p$ -function.

Sub TMIN(Species As String, TMIN As Double)

- Returns the minimum valid temperature of the  $c_p$ -function.