# Using a DataSet in VisualAPL

This example provides the source code for a simple VisualAPL Console Project built in Visual Studio 2008 which illustrates the use of a DataSet.

A “using System.Data;” directive has been included so that DataSet, DataTable, DataRow and DataColumn will be defined.

A single example table (#0) is defined in the DataSet and two columns (“ID” and “Name”) are defined in the table.

The first loop sets the five rows of table #0 in the DataSet.

The second loop illustrates the use of the “foreach{…}” iterator to display the values of the rows of table #0 of the DataSet.

The third loop illustrates the use of the “for{…}” iterator to display the same values. Note that the traditional APL “:FOR … :ENDFOR” control structure could have been used in place of the .Net-style “for{…}” loop control structure. The third loop also illustrates the assignment of APL variables with values in the rows of table #0 of the DataSet.

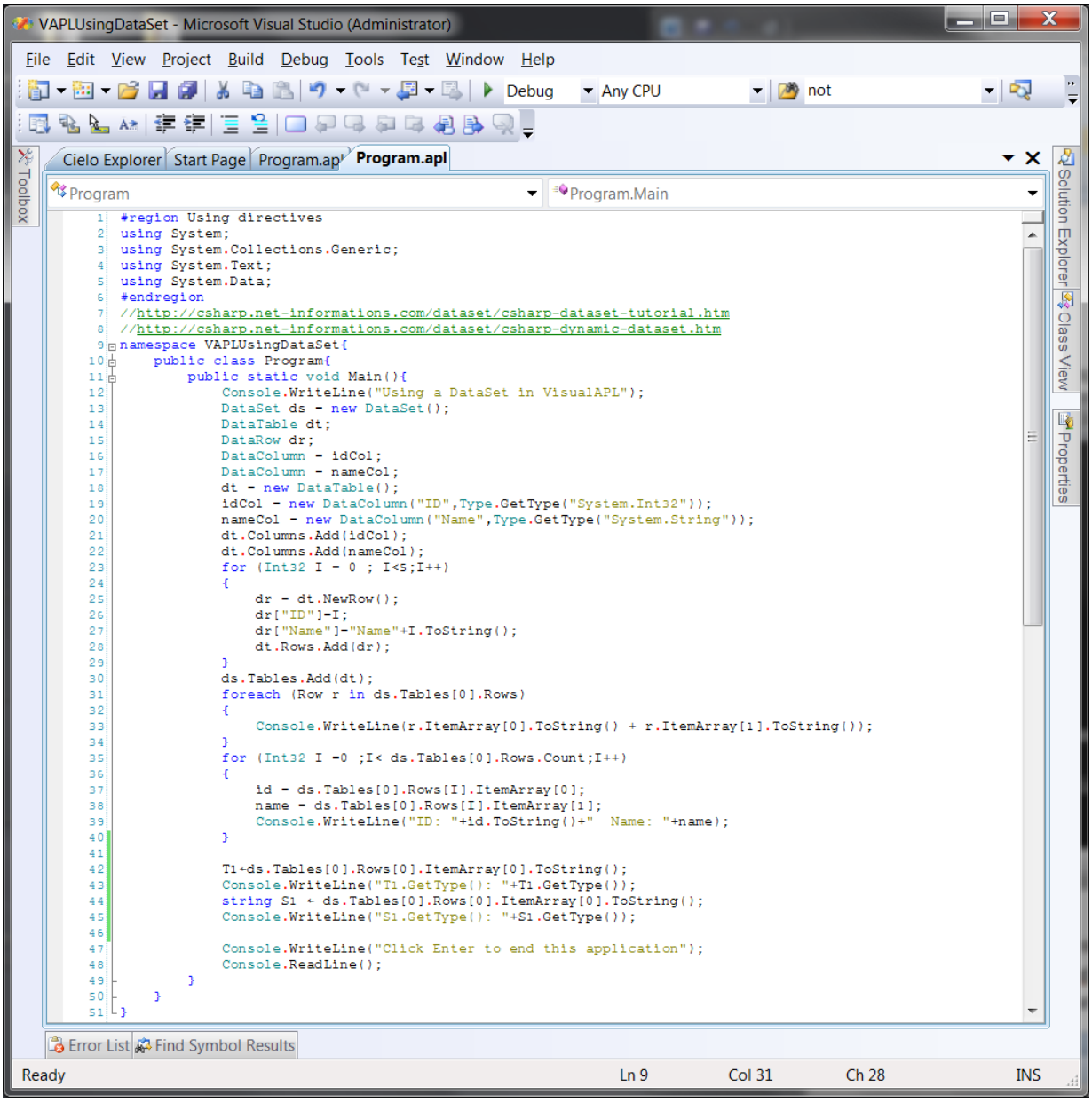
The assignment of T1 and S1 illustrate the use of dynamically- and strongly-typed variables respectively.

The source code for this simple example was obtained from:

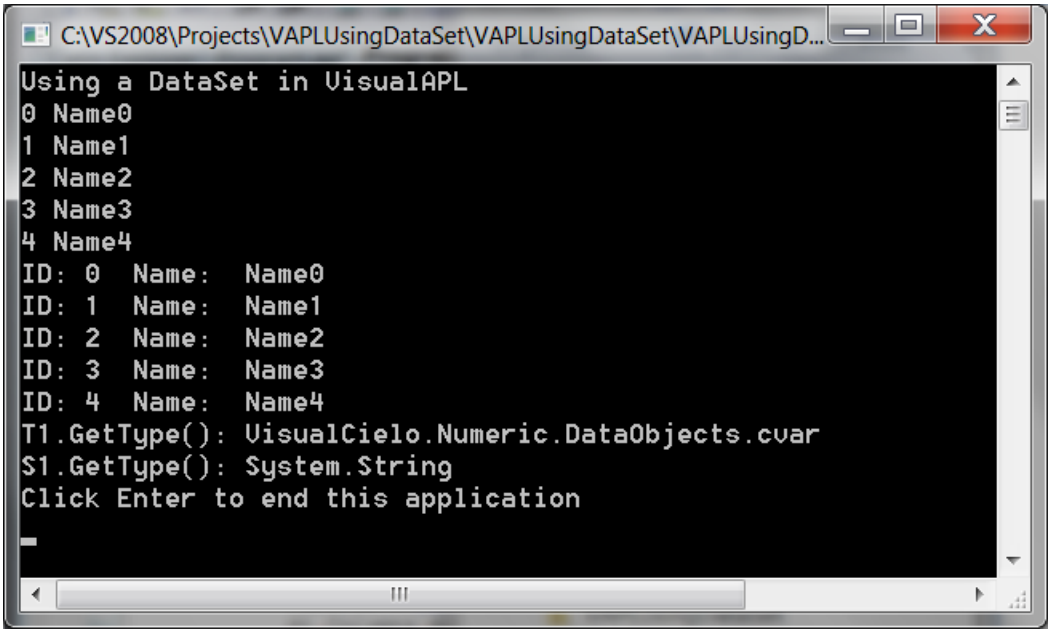
http://csharp.net-informations.com/dataset/csharp-dataset-tutorial.htm

http://csharp.net-informations.com/dataset/csharp-dynamic-dataset.htm

There is a wealth of documentation on .Net DataSet usage on the Internet. Try searching for “C# DataSet” for more links of interest.


VAPLUsingDataSet - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Tools Test Window Help
Debug
Any CPU
not
Cielo Explorer
Start Page
Program.ap
Program.apl
Program
Program.Main
Toolbox
Solution Explorer
Class View
Properties
#region Using directives
using System;
using System.Collections.Generic;
using System.Text;
using System.Data;
#endregion
//http://csharp.net-informations.com/dataset/csharp-dataset-tutorial.htm
//http://csharp.net-informations.com/dataset/csharp-dynamic-dataset.htm
namespace VAPLUsingDataSet{
public class Program{
public static void Main(){
Console.WriteLine("Using a DataSet in VisualAPL");
DataSet ds = new DataSet();
DataTable dt;
DataRow dr;
DataColumn idCol;
DataColumn nameCol;
dt = new DataTable();
idCol = new DataColumn("ID",Type.GetType("System.Int32"));
nameCol = new DataColumn("Name",Type.GetType("System.String"));
dt.Columns.Add(idCol);
dt.Columns.Add(nameCol);
for (Int32 I = 0 ; I<5;I++)
{
dr = dt.NewRow();
dr["ID"]=I;
dr["Name"]="Name"+I.ToString();
dt.Rows.Add(dr);
}
ds.Tables.Add(dt);
foreach (Row r in ds.Tables[0].Rows)
{
Console.WriteLine(r.ItemArray[0].ToString() + r.ItemArray[1].ToString());
}
for (Int32 I =0 ;I< ds.Tables[0].Rows.Count;I++)
{
id = ds.Tables[0].Rows[I].ItemArray[0];
name = ds.Tables[0].Rows[I].ItemArray[1];
Console.WriteLine("ID: "+id.ToString()+"  Name: "+name);
}
T1←ds.Tables[0].Rows[0].ItemArray[0].ToString();
Console.WriteLine("T1.GetType(): "+T1.GetType());
string S1 ← ds.Tables[0].Rows[0].ItemArray[0].ToString();
Console.WriteLine("S1.GetType(): "+S1.GetType());
Console.WriteLine("Click Enter to end this application");
Console.ReadLine();
}
}
}
Error List
Find Symbol Results
Ready
Ln 9
Col 31
Ch 28
INS

When this project is run, the console display is as follows:




Notice that the T1 variable because it was not strongly-typed is instantiated as a dynamically-typed Cielo variable even though the .ToString() operator is used to establish its value. However the S1 variable with the same value is instantiated as a strongly-typed string variable.