ClearScript Examples

# Namespaces

The examples below use the following namespaces:

C#

using System;

using Microsoft.ClearScript;

using Microsoft.ClearScript.JavaScript;

using Microsoft.ClearScript.V8;

# Example code

The following examples demonstrate selected ClearScript features:

C#

// create a script engine

using (var engine = new V8ScriptEngine())

{

// expose a host type

engine.AddHostType("Console", typeof(Console));

engine.Execute("Console.WriteLine('{0} is an interesting number.', Math.PI)");

// expose a host object

engine.AddHostObject("random", new Random());

engine.Execute("Console.WriteLine(random.NextDouble())");

// expose entire assemblies

engine.AddHostObject("lib", new HostTypeCollection("mscorlib", "System.Core"));

engine.Execute("Console.WriteLine(lib.System.DateTime.Now)");

// create a host object from script

engine.Execute(@"

birthday = new lib.System.DateTime(2007, 5, 22);

Console.WriteLine(birthday.ToLongDateString());

");

// use a generic class from script

engine.Execute(@"

Dictionary = lib.System.Collections.Generic.Dictionary;

dict = new Dictionary(lib.System.String, lib.System.Int32);

dict.Add('foo', 123);

");

// call a host method with an output parameter

engine.AddHostObject("host", new HostFunctions());

engine.Execute(@"

intVar = host.newVar(lib.System.Int32);

found = dict.TryGetValue('foo', intVar.out);

Console.WriteLine('{0} {1}', found, intVar);

");

// create and populate a host array

engine.Execute(@"

numbers = host.newArr(lib.System.Int32, 20);

for (var i = 0; i < numbers.Length; i++) { numbers[i] = i; }

Console.WriteLine(lib.System.String.Join(', ', numbers));

");

// create a script delegate

engine.Execute(@"

Filter = lib.System.Func(lib.System.Int32, lib.System.Boolean);

oddFilter = new Filter(function(value) {

return (value & 1) ? true : false;

});

");

// use LINQ from script

engine.Execute(@"

oddNumbers = numbers.Where(oddFilter);

Console.WriteLine(lib.System.String.Join(', ', oddNumbers));

");

// use a dynamic host object

engine.Execute(@"

expando = new lib.System.Dynamic.ExpandoObject();

expando.foo = 123;

expando.bar = 'qux';

delete expando.foo;

");

// call a script function

engine.Execute("function print(x) { Console.WriteLine(x); }");

engine.Script.print(DateTime.Now.DayOfWeek);

// examine a script object

engine.Execute("person = { name: 'Fred', age: 5 }");

Console.WriteLine(engine.Script.person.name);

// read a JavaScript typed array

engine.Execute("values = new Int32Array([1, 2, 3, 4, 5])");

var values = (ITypedArray<int>)engine.Script.values;

Console.WriteLine(string.Join(", ", values.ToArray()));

}

# Example Output

The examples above generate the following output (some results are random or time-specific):

Console

3.14159265358979 is an interesting number.

0.715555223503874

5/11/2017 12:15:32 PM

Tuesday, May 22, 2007

True 123

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19

1, 3, 5, 7, 9, 11, 13, 15, 17, 19

Thursday

Fred

1, 2, 3, 4, 5