SLANGJS - A SLANG (Simple Language) to JavaScript Translator

The Slang4.net (http://slangfordotnet.codeplex.com) is a open source compiler written to demonstrate Compiler Construction. The base Code of the compiler was written to implement a Domain Specific Language (DSL) to process Excel Spreadsheet.

The Compiler Infrastructure has got

A) Tree Walking Interpreter
B) A .NET IL backend
C) A Node.js (Google V8) compatible JavaScript/JSON generator

This documents shows how to use the compiler. One is supposed to have .NET 4.0 runtime installed on the machine to run this.

The following files are there in the Folder …

```
G:\TO_UST>dir
 Volume in drive G has no label.
 Volume Serial Number is 8CC1-D237

 Directory of G:\TO_UST

09/24/2012  03:41 AM    <DIR>          .
09/24/2012  03:41 AM    <DIR>          ..
01/29/2010  11:28 AM               451 fact.sl
01/29/2010  11:41 AM               539 Fib.sl
01/29/2010  11:38 AM               446 fibrec.sl
01/29/2010  11:34 AM                54 helloworld.sl
01/29/2010  11:30 AM               204 onetohundred.sl
01/29/2010  10:29 AM               559 Quad.sl
09/24/2012  02:34 AM             5,632 SLANGJS.exe
09/24/2012  02:34 AM            43,520 SLANG_DOT_NET.dll
               8 File(s)         51,405 bytes
               2 Dir(s)   5,071,437,824 bytes free
```

Given below is the list of example programs available in the SLANG4.net distribution

Program Name	**Description**
Helloworld.sl	Spits Hello World in the console
Onetohundred.sl	Prints from 1 to 100 (demonstrates loop)
Quad.sl	Program to determine whether a solution exists for a Quadratic Equation

Fib.sl	Compute Fibonacci series (Iteratively)
Fact.sl	Program to compute Factorial of a number recursively
Fibrec.sl	Example for Tree Recursion

A) Program To demonstrate HelloWorld (HelloWorld.sl)

```
FUNCTION BOOLEAN MAIN()
PRINT "Hello World";

END
```

The Compiler generated the following code and node.js can execute it

```

G:\TO_UST>slangjs helloworld.sl > helloworld.js

G:\TO_UST>type helloworld.js
//--- invoke the main method ...
MAIN()

//---- Generated JavaScript from SLANG Script
function MAIN(
 ) {
console.log('Hello World');

 }
//----End  Generated JavaScript

G:\TO_UST>node helloworld.js
Hello World

G:\TO_UST>
```

B) A Program to calculate Fibonacci series recursively

```


G:\TO_UST>type Fibrec.sl
/////////////////////////////////
//
// Recursive Fibonacci series routine
//
//
```

```
FUNCTION NUMERIC FIB( NUMERIC n )
      IF ( n <= 1 ) then
           return 1;
      ELSE
         RETURN FIB(n-1) + FIB(n-2);
      ENDIF

END



////////////////////////////////////////
//
//
//  Main routine
//
//

FUNCTION BOOLEAN MAIN()
NUMERIC d;
d=0;
While ( d <= 10 )
   PRINTLINE FIB(d);
   d = d+1;
Wend
END

G:\TO_UST>slangjs Fibrec.sl > Fibrec.js

G:\TO_UST>type Fibrec.js
//--- invoke the main method ...
MAIN()

//---- Generated JavaScript from SLANG Script
function FIB(
N ) {
 if ( N<=1 ) {
 return 1;
 }
 else {
 return FIB( N-1)+FIB( N-2);

 }


 }
function MAIN(
 ) {
```

```
var D;
D=0;
while ( D<=10) {
console.log(FIB( D));
D=D+1;

}

 }
//----End  Generated JavaScript

G:\TO_UST>node Fibrec.js
1
1
2
3
5
8
13
21
34
55
89

G:\TO_UST>
```

C) A Program to Compute the first 100 numbers

```

G:\TO_UST>type onetohundred.sl
//////////////////////////////
//
// Program to Print One To Hundred
// STEP 7 and above
//
FUNCTION BOOLEAN MAIN()
NUMERIC d;
d=0;
While ( d <= 100 )
   PRINTLINE d;
   d = d+1;
Wend
END

G:\TO_UST>slangjs onetohundred.sl > onetohundred.js

G:\TO_UST>type onetohundred.js
```

```
//--- invoke the main method ...
MAIN()

//---- Generated JavaScript from SLANG Script
function MAIN(
 ) {
var D;
D=0;
while ( D<=100) {
console.log(D);
D=D+1;

}

 }
//----End  Generated JavaScript

G:\TO_UST>node onetohundred.js
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
```

[redacted]

```
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

G:\TO_UST>
```

D) A Program to recursively compute Factorial of a number

```

G:\TO_UST>type fact.sl
////////////////////
//
//  Recursive Routine to Print Factorial of a number
//
//  STEP 7 and above ...test case for recursion ...
//

FUNCTION NUMERIC FACT( NUMERIC d )
   IF ( d <= 0 ) THEN
        return 1;
   ELSE
        return d*FACT(d-1);
   ENDIF

END

////////////////////
```

```
//
//
//  Entry Point
//
//
FUNCTION BOOLEAN MAIN()
NUMERIC d;
d=0;
While ( d <= 10 )
    PRINTLINE FACT(d);
    d = d+1;
Wend
END

G:\TO_UST>slangjs fact.sl > fact.js

G:\TO_UST>type fact.js
//--- invoke the main method ...
MAIN()

//---- Generated JavaScript from SLANG Script
function FACT(
D ) {
 if ( D<=0 ) {
 return 1;
 }
 else {
 return D*FACT( D-1);

 }


 }
function MAIN(
 ) {
var D;
D=0;
while ( D<=10) {
console.log(FACT( D));
D=D+1;

}

 }
//----End  Generated JavaScript

G:\TO_UST>node fact.js
1
```

```
1
2
6
24
120
720
5040
40320
362880
3628800

G:\TO_UST>
```