

Forum:[Hack Forums](#)

Thread: [Lexiron - Front-End Script Engine \[2.1 Updated\]](#)

TheBarret - May 3, 2014, 10:32 am

I present you Lexiron or 'Lexi' and has entered the beta stage.

The purpose of Lexi is that you can extend your functionality of your application dynamicly. Adding your functions hooked into Lexi, making it agile and a versatile tool for your applications.

Ever wanted to let your end-users change the behavior of your application or perhaps you want to have a dynamic element on your own application to make it more functional?

Lexi could be your awnser.

Extended with scripts that can be build to serialized objects and even encrypted if you want it protected.

How does it work?

I'm not gonna explain every detail but it uses the LL(1) top down parsing method. You will find more details on this method here:

http://en.wikipedia.org/wiki/LL_parser

Lexi breaks down the character stream into tokens wich then are broken down to seperate AST elements, after this is done Lexi can save this state of your script into a serialized object.

As you can see here:

How can we use it?

You set a new reference of lexi in your project.

Declare the engine
Code Private Engine As Lexiron.Engine

Private Sub FrmMain_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

Engine = New Lexiron.Engine
End Sub

To load and build a script:

Code Engine.Parse(Filename)

If no errors occoured you can save it to a serialized object:

Code Engine.Save(Filename)

To execute your finalized script object:

```
Code Engine.Execute(Filename,Parameters())
```

This function also returns a value if the script makes use of the 'return()'

How can we check if any errors occurred?

You can check the engine for errors as follows

```
Code if (Engine.HasError) then
```

```
..code...
```

```
End If
```

If this is true, you wanna see what kind of errors occurred

```
Code For Each Err As ParseError In Engine.Errors
```

```
...show errors...
```

```
Next
```

Engine usefull routines

```
Code Public Sub Parse(Filename As String)
```

```
Public Sub AddDelegate(Reference As String, Delegate As ScriptDelegate)
```

```
Public Sub RemoveDelegate(Reference As String)
```

```
Public Sub Open(Filename As String, Optional Key As String = "")
```

```
Public Sub Save(Filename As String, Optional Key As String = "")
```

```
Public Function Execute(Filename As String, ParamArray Params() As Object) As Object
```

```
Public Function Execute(ParamArray Params() As Object) As Object
```

```
Public Function Evaluate(Expression As String, ParamArray Params() As Object) As Object
```

```
Public Sub GenerateTreeView(Tv As TreeView)
```

```
Public Sub SerializeObject(Obj As Object, OutputFile As String, Compression As Boolean, Optional  
Key As String = "")
```

```
Public Function Deserialize(Obj As Object, Inputfile As String, Optional Key As String = "") As  
Object
```

Properties:

```
Code Public ReadOnly Property Tokenizer As Tokenizer
```

```
Public ReadOnly Property Tokens As List(Of Token)
```

```
Public ReadOnly Property HasError As Boolean
```

```
Public ReadOnly Property Errors As ParseError()
```

```
Public ReadOnly Property AstTree As Statement
```

```
Public Property Break() As Boolean
```

Changelog

2.0 Initial Release

- Released

[Download link](#)

2.1

- Fixed: Return(Expression) will terminate any further execution and return to the caller

- Added: For-Loops

- Added: break() Will exit from the current active for-loop

- Added: Switch Case (With '..else' support)

- Added: Engine.Evaluate() will evaluate and execute direct input and return if any

- Added: Engine.Execute() Another execute signature for in combination with Engine.Open()

- Added: Engine.Break() Will forcefully stop a script execution

[Download link](#)

Questions and Feedback are welcome.

If you need help using my software you can pm me for skype.

(if ur planning to decompile my engine, go ahead, but your better of asking me how it works as i can explain to you how it works)

- Barret

Washi - May 3, 2014, 10:44 am

Been talking about this already for quite a while with barret, as we work on similar projects :3.

Glad to see it released. Really reminds me of LUA, as it is also a script language that extends a program. This one however has a C-ish syntax, which I like actually. As far as I can see it's really easy to add functions, which is pretty sweet!

Some questions: Does it also support some kind of start-up arguments ? How about declaring functions inside the script ? Maybe you can add a possibility to execute these functions as well from the host application, so people can write for instance callbacks for events. That would be pretty cool.

Temploit - May 3, 2014, 10:52 am

You seriously have too much free time, to do so much in so little time. How much coffee do you consume on a daily basis?

I'm really curious on how that syntax tree works :3

Great job as always ^^

TheBarret - May 3, 2014, 10:55 am

(May 3, 2014 10:44 AM)TheUnknownProgrammer Wrote: [\[CITE\]](#) Been talking about this already for quite a while with barret, as we work on similar projects :3.

Glad to see it released. Really reminds me of LUA, as it is also a script language that extends a program. This one however has a C-ish syntax, which I like actually. As far as I can see it's really easy to add functions, which is pretty sweet!

Some questions: Does it also support some kind of start-up arguments ? How about declaring functions inside the script ? Maybe you can add a possibility to execute these functions as well from the host application, so people can write for instance callbacks for events. That would be pretty cool.

Arguments

Yes it does

Code Engine.Executes(Filename,Arguments)

Declaring external function

Yes it does

Code Engine.AddDelegate("print", New Engine.ScriptDelegate(AddressOf Me.ScriptMessage))
and for that delegate signature:

Code Private Function ScriptMessage(Params() As Object) As Object

 If (Params.Length > 0) Then

 ScriptLog.AppendText(String.Join("", Params) & ControlChars.CrLf)

 ScriptLog.SelectionStart = DebugLog.Text.Length - 1

 ScriptLog.ScrollToCaret()

 End If

 Return String.Empty

End Function
- Barret

SIM-1676753 - May 3, 2014, 11:10 am

This is truly awesome bro! I've seen you post this in another thread and thought it was pretty cool.

SIM-1456245 - May 3, 2014, 11:30 am

This is just amazing. Ability to dynamically execute code in your app in another language and use it in the actual app, now that is epic. Also I was going to ask for the Help thingy, but it's already there. The three is amazing and the whole thing is amazing.

jeiceei - May 3, 2014, 3:13 pm

this is really some good shit, I appreciate

TheBarret - May 3, 2014, 3:28 pm

Next update is coming soon!

To be expected new features in lexi:

- For Loops
Can be escaped with break()
Nested loops possible with the use of a Loopstack

```
Code for x = 1 to 10 {  
    print("Hello!");  
    if (x=5) { break(); }  
}
```

- Select Case

```
Code Switch x {  
    :1 { }  
    :2 { }  
    :3 { }  
}
```

- Fixed: Return() will terminate the script execution and return to the caller.
- Added: Syntax tree will show and mark where "escapes" are set for better overview.

- Barret

Washi - May 3, 2014, 3:29 pm

(May 3, 2014 10:55 AM)TheBarret Wrote: [\[CITE\]](#) Arguments

Yes it does

Code Engine.Executes(Filename,Arguments)

Declaring external function

Yes it does

Code Engine.AddDelegate("print", New Engine.ScriptDelegate(AddressOf Me.ScriptMessage))
and for that delegate signature:

```
Code Private Function ScriptMessage(Params() As Object) As Object  
    If (Params.Length > 0) Then  
        ScriptLog.AppendText(String.Join("", Params) & ControlChars.CrLf)  
        ScriptLog.SelectionStart = DebugLog.Text.Length - 1  
        ScriptLog.ScrollToCaret()
```

```
End If
Return String.Empty
End Function
```

- Barret

I was actually not talking about external functions, but internal functions, which are declared inside the script and not by the host. This allows people to for instance set up event listeners or callbacks that will get executed when an event occurs. Suppose I have a button in the host application and want to add a callback to the click event of that button in my script.

The script would look something like below.

PHP Code

```
<?php
addhandler(button_click_event, click_callback);

function click_callback()
{
    print("This script has just handled a click event from the host application!");
}
```

The addhandler would be specified by the host application of course, as that is an external function. It will subscribe a function declared inside the script to an event declared outside the script. And on the host, I'd put in my button handler something like below, which will dispatch the event and calls all functions that are subscribing to this event.

PHP Code

```
<?php
Public Sub Button1_Click(...) Handles Button1.Click
    Engine.DispatchEvent(button_click_event)
End Sub
```

If you could implement that, then this script engine would be really useful.

TheBarret - May 3, 2014, 3:35 pm

(May 3, 2014 03:29 PM)TheUnknownProgrammer Wrote: [\[CITE\]](#) If you could implement that, then this script engine would be really useful.

Hmmm ye i think i can make that possible.

Let me see what i can do tomorrow, got some new updates waiting already (forloops are working tho)

- Barret

Weakky. - May 3, 2014, 4:01 pm

Impressive man. That would be much better if you were sharing the source code though. I'm really interesting to see how you handled the parsing. Amazing job anyway.

TheBarret - May 4, 2014, 12:51 am

New update,

- Forloops

- Switch

And a few fixes..

Smelly Shoes - May 4, 2014, 7:24 pm

<http://sharpdox.de/en/> - Software for creating documentation for C# projects. I love it.

TheBarret - May 5, 2014, 2:51 am

(May 4, 2014 07:24 PM)Smelly Shoes Wrote: [\[CITE\]](#) <http://sharpdox.de/en/> - Software for creating documentation for C# projects. I love it.

Nice find, really usefull thank you! ?

ManDoR - May 5, 2014, 9:59 am

(May 3, 2014 03:28 PM)TheBarret Wrote: [\[CITE\]](#) Next update is coming soon!

To be expected new features in lexi:

- For Loops

Can be escaped with break()

Nested loops possible with the use of a Loopstack

```
Code for x = 1 to 10 {  
    print("Hello!");  
    if (x=5) { break(); }  
}
```

- Select Case

```
Code Switch x {  
    :1 {}  
    :2 {}  
    :3 {}  
}
```

- Fixed: Return() will terminate the script execution and return to the caller.

- Added: Syntax tree will show and mark where "escapes" are set for better overview.

- Barret

In my opinion, you shouldn't be using "Break()" or "Return()", but instead "Break" and "Return". In fact, those aren't methods or functions, so you shouldn't be using their syntax.

Nice work though.

TheBarret - May 5, 2014, 10:20 am

(May 5, 2014 09:59 AM)ManDoR Wrote: [\[CITE\]](#) In my opinion, you shouldn't be using "Break()" or "Return()", but instead "Break" and "Return". In fact, those aren't methods or functions, so you shouldn't be using their syntax.

Nice work though.

Well thanks for telling me, altho i know what im doing, after im done with the fine tweaking, i used actual functions for quick testing to implement break and returning, after im done with that tweaking ill hard code them as keywords.

More geuston, critism?