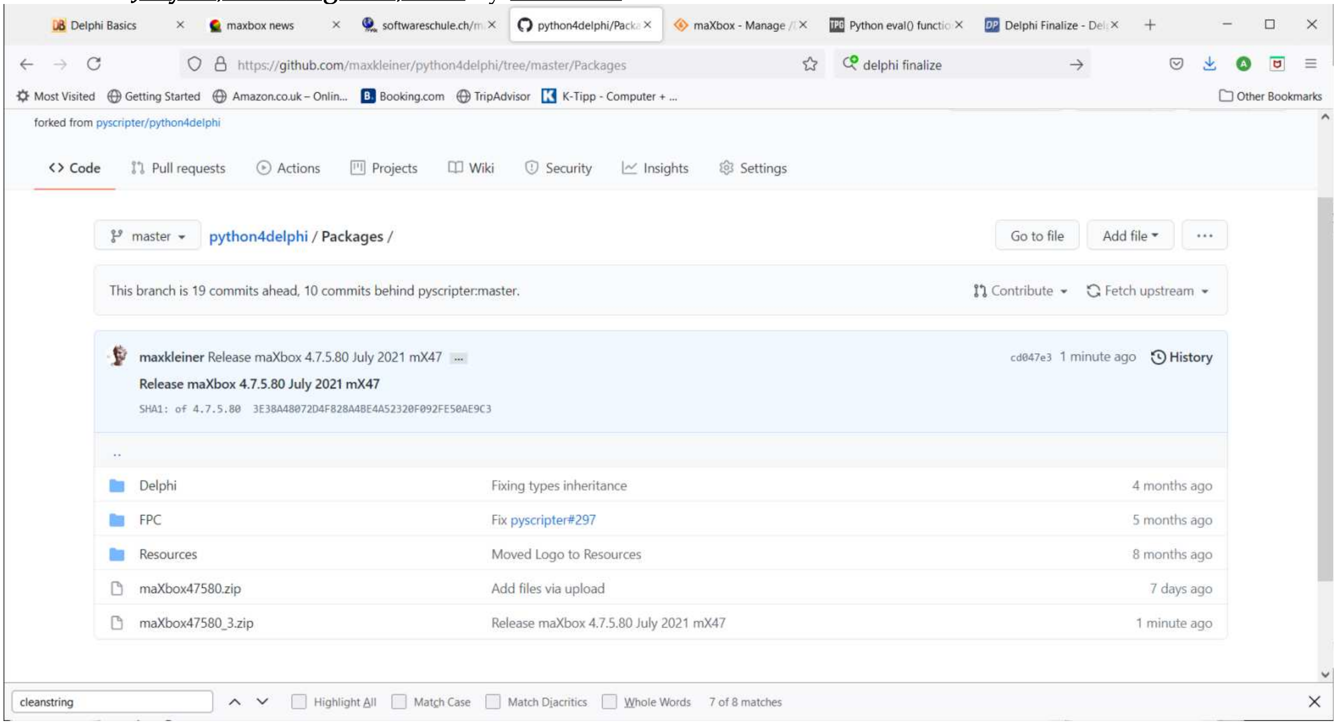
Welcome to maXbox Code (Blog)

maXbox

Python4maXbox Code

Posted on **July 28, 2021August 9, 2021** by **maxbox4**


https://github.com/maxkleiner/python4delphi/tree/master/Packages

forked from pyscripter/python4delphi

master python4delphi / Packages /

This branch is 19 commits ahead, 10 commits behind pyscripter:master.

maxkleiner Release maXbox 4.7.5.80 July 2021 mX47 cd047e3 1 minute ago History

Release maXbox 4.7.5.80 July 2021 mX47

SHA1: of 4.7.5.80 3E38A48072D4F828A4BE4A52320F092FE50AE9C3

..		
Delphi	Fixing types inheritance	4 months ago
FPC	Fix pyscripter#297	5 months ago
Resources	Moved Logo to Resources	8 months ago
maXbox47580.zip	Add files via upload	7 days ago
maXbox47580_3.zip	Release maXbox 4.7.5.80 July 2021 mX47	1 minute ago

cleanstring — Highlight All — Match Case — Match Diacritics — Whole Words — 7 of 8 matches

Thanks to Python4Delphi we now can evaluate (for expressions) or exec (for statements) some Python code in our scripts. This version 4.7.5.80 July 2021 allows us with the help of a Python Dll and an environment with modules in site-packages execute Py-functions. But the most is only available in a 32-bit space, possible also with 64-bit Python means the call of the external shell (ExecuteShell) with installed Python versions to choose from. By the way also a **Python4Lazarus** is available.

1413 unit uPSI_PythonEngine.pas _P4D_Beta
1414 unit uPSI_VclPythonGUIInputOutput;
1415 unit uPSI_VarPyth;
1416 unit JclUsesUtils;
1417 unit uPSI_cParameters;
1418 unit uPSI_WDCCMisc; (uPSI_cFileTemplates);
1419 uPSI_WDCCOleVariantEnum.pas
1420 unit uPSI_WDCCWinInet.pas _WDCC
1421 uPSI_PythonVersions.pas _P4D_
1422 unit uPSI_PythonAction.pas _P4D_

Imagine you need a 512bit hash and you don’t have the available function. SHA256 or SHA512 is a secure hash algorithm which creates a fixed length one way string from any input data. OK you start the Python-engine in your maXbox script and load the DLL. Most of the time you don’t need to install Python cause you find a DLL for example in the Wow64 subsystem and load it. **WoW64** (**W**indows 32-bit **o**n **W**indows **64**-bit) is a subsystem of the **Windows (https://en.wikipedia.org/wiki/Microsoft_Windows)** **operating system (https://en.wikipedia.org/wiki/Operating_system)** capable of running **32-bit (https://en.wikipedia.org/wiki/32-bit)** applications on 64-bit Windows.

To get a Dll that fits your size and space you can check with

```
writeln('is x64 '+botostr(Isx64('C:\maXbox\EKON24\python37.dll')));
```

Or best you install the environment with: **https://www.python.org/ftp/python/3.7.9/python-3.7.9.exe (https://www.python.org/ftp/python/3.7.9/python-3.7.9.exe)**

I provide also just a Dll one which we use most at:

https://sourceforge.net/projects/maxbox/files/Examples/EKON/P4D/python37.dll/download (https://sourceforge.net/projects/maxbox/files/Examples/EKON/P4D/python37.dll/download)

Search for registered versions is possible with the function *GetRegisteredPythonVersions : TPythonVersions');* On 64-bit Windows the 32-bit python27.dll is really in C:\Windows\sysWOW64. But if you try opening the C:\Windows\system32\python27.dll in a 32-bit process, it'll open just fine. If I'm not mistaken, WOW stands for Woodoo Of Windows. 😉

To make sure your path is the right test it with *OpenDll()*:

```
procedure TDynamicDll.LoadDll;
begin
  OpenDll( DllName );
end;

  eng.dllpath:= 'C:\maXbox\EKON25'
  eng.dllname:= 'python37.dll';
  eng.AutoLoad:= false;
  try
    eng.OpenDll('C:\maXbox\EKON25\python37.dll');
```

Or then you type the path and name of the Dll:

```
with TPythonEngine.create(self) do begin
 //Config Dll or Autoload
   Dllpath:= 'C:\Users\max\AppData\Local\Programs\Python\Python36-32\'
   Dllname:= 'python37_32.dll';
   LoadDll;
   writeln(pythonhome)
   writeln(ExecModule)
   pypara:= 'https://en.wikipedia.org/wiki/WoW64 (https://en.wikipedia.org/wiki/WoW64)';
   //pypara:= filetostring(exepath+'maXbox4.exe')
   try
     writeln(evalstr('__import__("math").sqrt(45)'));
     writeln(evalstr('__import__("hashlib").sha1(b"https://en.wikipedia.org/wiki/WoW64 (
     writeln(evalstr('__import__("hashlib").sha1(b"'+pypara+'").hexdigest().upper()'));
     writeln(evalstr('__import__("hashlib").sha256(b"'+pypara+'").hexdigest().upper()'))
     writeln(evalstr('__import__("hashlib").sha512(b"'+pypara+'").hexdigest().upper()'))
   finally
     free
   end;
 end;
```

So the last line is the sha512 and the result is: 8B94C64213CAD……and so on. The important thing is the evalstr() function. The `eval()` allows us to execute arbitrary strings as Python code. It accepts a source string and returns an object. But we can also import modules with the inbuilt syntax '**import**("hashlib").

The `eval()` is not just limited to simple expression. We can execute functions, call methods, reference variables and so on. So we use this by using the `__import__()` built-in function. Note also that the computed hash is converted to a readable hexadecimal string by hexdigest().upper()' and uppercase the hex-values in one line, amazing isn't it.

We step a bit further to exec a script in a script! If we call a file or an **const** Python command then we use ExecString(PYCMD); The script you can find at: **http://www.softwareschule.ch/examples/pydemo.txt (http://www.softwareschule.ch/examples/pydemo.txt)**

The essence is a bit of script:

```
const PYCMD = 'print("this is box")'+LB+
              'import sys'+LB+
              'f=open(r"1050pytest21.txt","w")'+LB+
              'f.write("Hello PyWorld_mX4, \n")'+LB+
              'f.write("This data will be written on the file.")'+LB+
              'f.close()';
```

The LB = CR+LF; is important cause we call it like a file or stream and exec() is cleaning (delete CR) and encoding the passing script afterwards:

```
writeln('ExecSynCheck1 '+botostr(eng.CheckExecSyntax(PYCMD)));
eng.ExecString(PYCMD);
```

We also check the syntax before to prevent an exception: Exception: Access violation at address 6BA3BA66 in module 'python36.dll'. or 'python37_32.dll' Read of address 000000AD. Free the engine means destroying it calls *Py_Finalize*, which frees all memory allocated by the Python Dll.

Or, if you're just using the Python API without the VCL wrappers like we do, you can probably just call *Py_NewInterpreter* on your *TPythonInterface* object to get a fresh execution environment without necessarily discarding everything done before!

By success of execute PYCMD a file (1050pytest21.txt) is written with some text so we executed line by line the **PYCMD**.

This is the whole tester Procedure PYLaz_P4D_Demo2; but key takeaway is that only use `eval()` with a trusted source.

https://thepythonguru.com/python-builtin-functions/eval/ (https://thepythonguru.com/python-builtin-functions/eval/)

https://github.com/maxkleiner/python4delphi/tree/master/Tutorials (https://github.com/maxkleiner/python4delphi/tree/master/Tutorials)

Eval can be evil if untrusted

```
Procedure PYLaz_P4D_Demo2;
//https://wiki.freepascal.org/Python4Delphi (https://wiki.freepascal.org/Python4Delphi)
var eng : TPythonEngine;
    out1: TPythonGUIInputOutput;
begin
  eng:= TPythonEngine.Create(Nil);
  out1:= TPythonGUIInputOutput.create(nil)
  out1.output:= pyMemo; //debugout.output;  //memo2;
  out1.RawOutput:= False;
  out1.UnicodeIO:= False;
  out1.maxlines:= 20;
  out1.displaystring('this string thing')
  //eng.IO:= Out1;
  Out1.writeline('draw the line');
  try
    eng.LoadDll;
    eng.IO:= Out1;
    if eng.IsHandleValid then begin
      writeln('DLLhandle: '+botostr(eng.IsHandleValid))
      WriteLn('evens: '+ eng.EvalStringAsStr('[x**2 for x in range(15)]'));
      WriteLn('gauss: '+ eng.EvalStringAsStr('sum([x for x in range(101)])'));
      WriteLn('gauss2: '+ eng.EvalStr('sum([x % 2 for x in range(10100)])'));
      WriteLn('mathstr: '+ eng.EvalStr('"py " * 7'));
      WriteLn('builtins: '+ eng.EvalStr('dir(__builtins__)'));
      WriteLn('upperstr: '+ eng.EvalStr('"hello again".upper()'));
      WriteLn('workdir: '+ eng.EvalStr('__import__("os").getcwd()'));
      writeln('syncheck '+
              botostr(eng.CheckEvalSyntax('print("powers:",[x**2 for x in range(10)])'))

      eng.ExecString('print("powers:",[x**2 for x in range(10)])');
      writeln('ExecSynCheck1 '+botostr(eng.CheckExecSyntax(PYCMD)));
      eng.ExecString(PYCMD);
      writeln('ExecSynCheck2 '+botostr(eng.CheckExecSyntax(myloadscript)));
      writeln('ExecSynCheck3 '+
            botostr(eng.CheckExecSyntax(filetostring(PYSCRIPT))));
      eng.ExecString(filetostring(PYSCRIPT));
      writeln(eng.Run_CommandAsString('print("powers:",[x**2 for x in range(10)])',eval_
      writeln(eng.Run_CommandAsString('sum([x for x in range(201)])',eval_input));
      pymemo.update;
    end
      else writeln('invalid library handle! '+Getlasterrortext);
    writeln('PythonOK: '+botostr(PythonOK));
  except
     eng.raiseError;
     writeln('PyErr '+ExceptionToString(ExceptionType, ExceptionParam));
  finally
     eng.free;
  end;
  out1.free;
  //pyImport(PyModule);
end;
```

The procedure raiseError helps to find errors for example:

Exception: : SRE module mismatch.
Make sure you do not have any mismatch between Python interpreter version used (like 3.7) and the ‘re’ python module (like 3.6.1).

The resolution of Dlls has changed in Python 3.8 for Windows.

```
New in version 3.8: Previous versions of CPython would resolve Dlls using the default beh
```

And here’s the reference output from *Procedure PYLaz_P4D_Demo2*:

```
DLLhandle: TRUE
evens: [0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196]
gauss: 5050
gauss2: 5050
mathstr: py py py py py py py
builtins: ['ArithmeticError', 'AssertionError', 'AttributeError', 'BaseException',
'BlockingIOError', 'BrokenPipeError', 'BufferError', 'BytesWarning',
'ChildProcessError', 'ConnectionAbortedError', 'ConnectionError',
'ConnectionRefusedError', 'ConnectionResetError', 'DeprecationWarning', 'EOFError',
'Ellipsis', 'EnvironmentError', 'Exception', 'False', 'FileExistsError',
'FileNotFoundError', 'FloatingPointError', 'FutureWarning', 'GeneratorExit',
'IOError', 'ImportError', 'ImportWarning', 'IndentationError', 'IndexError',
'InterruptedError', 'IsADirectoryError', 'KeyError', 'KeyboardInterrupt',
'LookupError', 'MemoryError', 'ModuleNotFoundError', 'NameError', 'None',
'NotADirectoryError', 'NotImplemented', 'NotImplementedError', 'OSError',
'OverflowError', 'PendingDeprecationWarning', 'PermissionError', 'ProcessLookupError',
'RecursionError', 'ReferenceError', 'ResourceWarning', 'RuntimeError',
'RuntimeWarning', 'StopAsyncIteration', 'StopIteration', 'SyntaxError',
'SyntaxWarning', 'SystemError', 'SystemExit', 'TabError', 'TimeoutError', 'True',
'TypeError', 'UnboundLocalError', 'UnicodeDecodeError', 'UnicodeEncodeError',
'UnicodeError', 'UnicodeTranslateError', 'UnicodeWarning', 'UserWarning',
'ValueError', 'Warning', 'WindowsError', 'ZeroDivisionError', 'build_class', 'debug',
'doc', 'import', 'loader', 'name', 'package', 'spec', 'abs', 'all', 'any', 'ascii',
'bin', 'bool', 'bytearray', 'bytes', 'callable', 'chr', 'classmethod', 'compile',
'complex', 'copyright', 'credits', 'delattr', 'dict', 'dir', 'divmod', 'enumerate',
'eval', 'exec', 'exit', 'filter', 'float', 'format', 'frozenset', 'getattr',
'globals', 'hasattr', 'hash', 'help', 'hex', 'id', 'input', 'int', 'isinstance',
'issubclass', 'iter', 'len', 'license', 'list', 'locals', 'map', 'max', 'memoryview',
'min', 'next', 'object', 'oct', 'open', 'ord', 'pow', 'print', 'property', 'quit',
'range', 'repr', 'reversed', 'round', 'set', 'setattr', 'slice', 'sorted',
'staticmethod', 'str', 'sum', 'super', 'tuple', 'type', 'vars', 'zip']
upperstr: HELLO AGAIN
workdir: C:\maXbox\maxbox3\maxbox3\maXbox3
syncheck TRUE
ExecSynCheck1 TRUE
ExecSynCheck2 TRUE
ExecSynCheck3 TRUE
None
20100
PythonOK: TRUE
 mX4 executed: 28/07/2021 18:23:31 Runtime: 0:0:1.609 Memload: 43% use
```

Conclusion: The **eval()** method parses the expression passed to it and runs python expression(code) (but no statements) within the program. For you and for me 5 functions are crucial:

```
Function CheckEvalSyntax(const str: AnsiString):Boolean');
Function CheckExecSyntax(const str: AnsiString):Boolean');
Procedure ExecString(const command: AnsiString);');
Procedure ExecString3(const command: AnsiString);');//alias
Procedure ExecStrings4(strings: TStrings);');
Function EvalStringAsStr(const command: AnsiString):string');//alias
Function EvalStr(const command: AnsiString): string');
```

Also, consider the situation when you have imported **os** module in your python program like above *WriteLn('workdir: '+ eng.EvalStr('**import**("os").getcwd()'));*. The os module provides portable way to use operating system functionalities like: read or write a file. A single command can delete all files in your system!

So `eval` expects an expression, `import` is a statement. That said, what you can trying is the following combination:

```
Println('exec as eval: '+eng.EvalStr('exec("import os as o")'));
Println('exec: '+eng.EvalStr('o.getcwd()'));
>>> exec as eval: None
>>> exec: C:\maXbox\mX47464\maxbox4
writeln('uuid: '+eng.evalstr('exec("import uuid") or str(uuid.uuid4())'));
>>> uuid: 3b2e10f9-0e31-4961-9246-00852fd508bd
writeln('wget: '+ eng.EvalStr('__import__("urllib")'));
writeln('wget_like: '+ eng.EvalStr('__import__("urllib.request").request.urlretrieve("htt

```

Compare to inline code

Lets do comparison between inline code and eval?

```
import json
import urllib.request

urlData = "http://api.openweathermap.org/data/2.5/weather?q=Kiruna (http://api.openweathe
webURL = urllib.request.urlopen(urlData)
data = webURL.read()
JSON_object = json.loads(data.decode('utf-8'))
```

We open an API, read it an convert to JSON now the eval:

```
execString('import urllib.request');
execString('import json');
Println('openurl: '+eng.EvalStr('json.loads(urllib.request.urlopen("'+
                                 urlData+'").read().decode("utf-8"))'));
```

You can use `exec` in eval instead if you intend to import the module or also *ExecString*: it depends on the global or local namespace you set, means also the second line knows the import statement from first line:

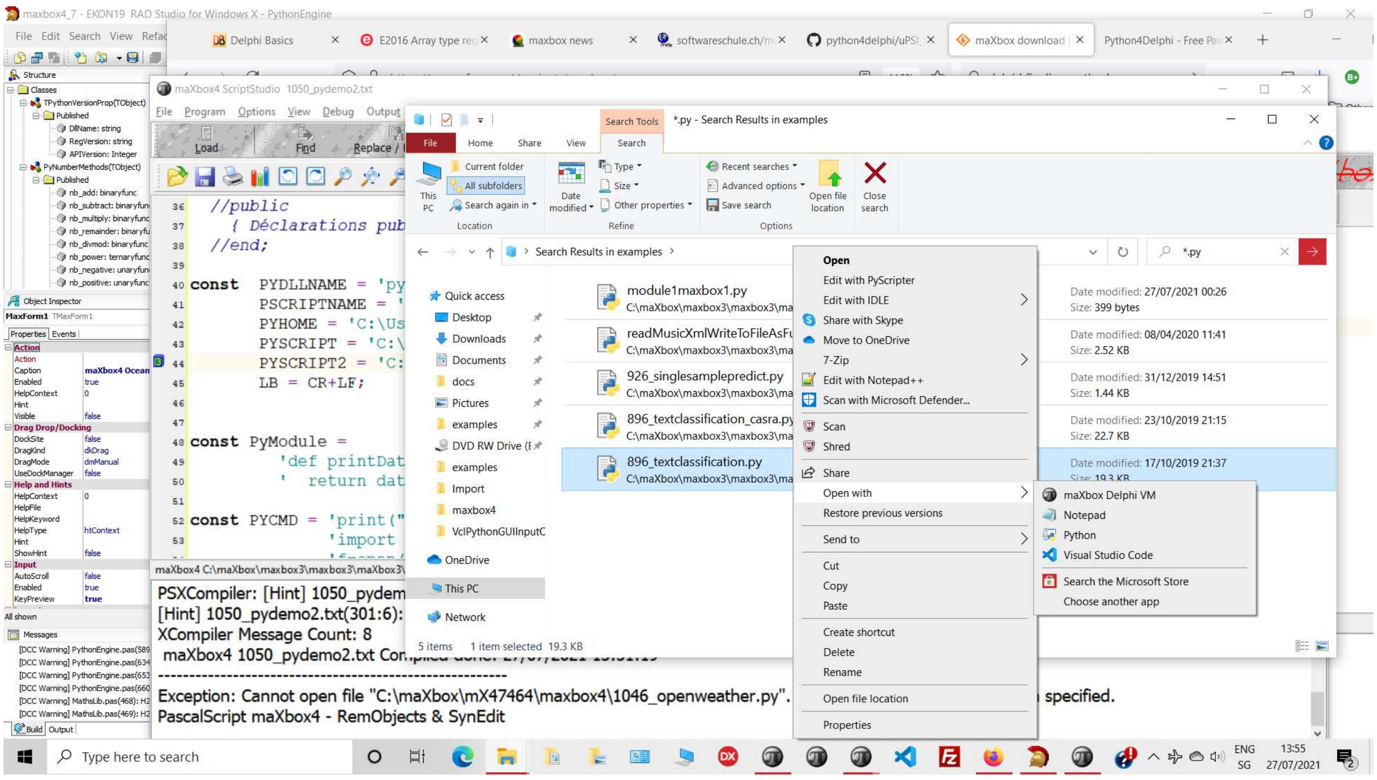
```
eng.ExecString('import math');
Println('evalexec: '+eng.EvalStr('dir(math)'));
```

When you use a float that doesn't have an exact binary float representation, the `Decimal` constructor cannot create an accurate decimal representation. For example:

```
import decimal
from decimal import Decimal

x = Decimal(0.1)
print(x)
```

And the same with an EvalExec:

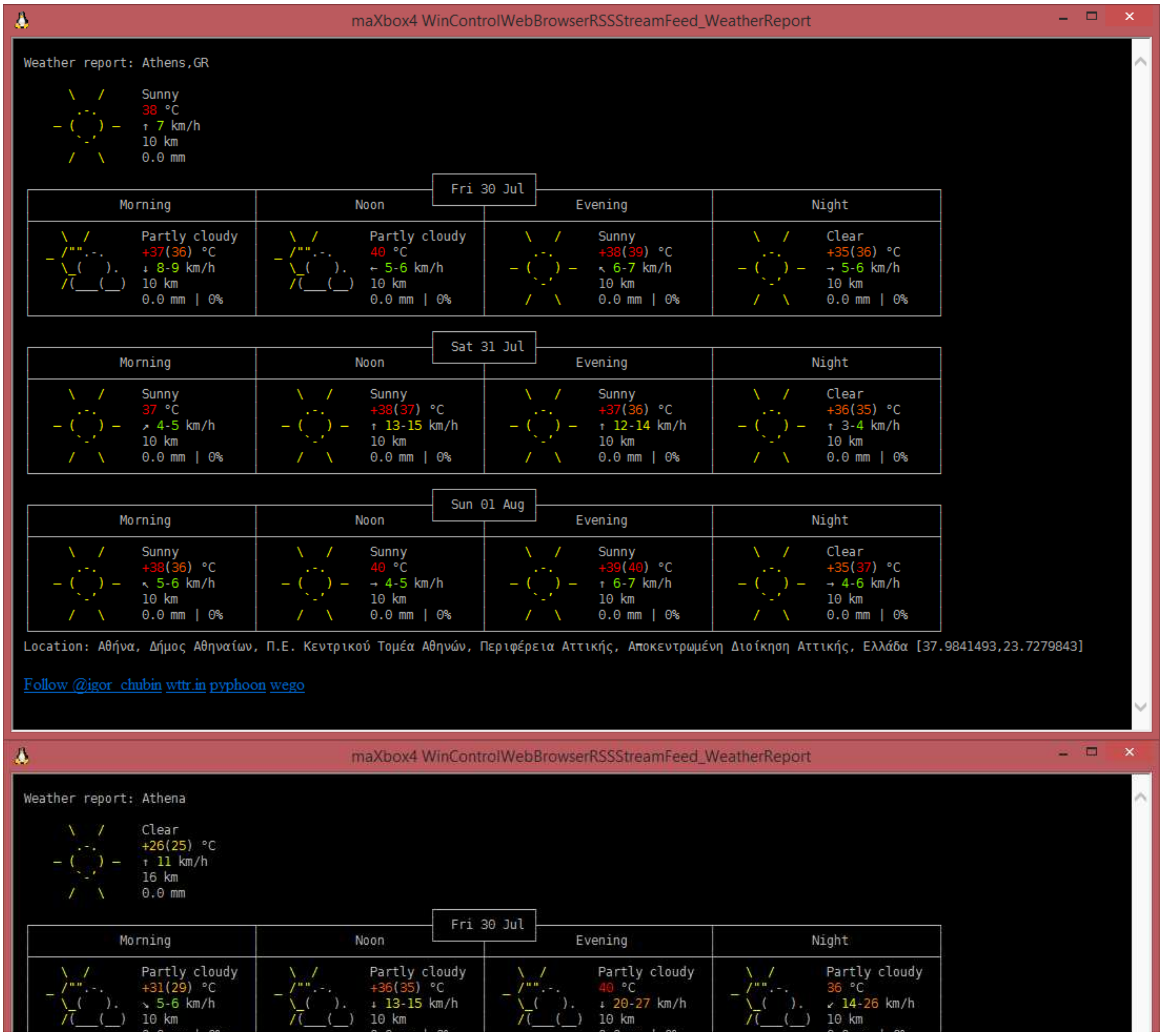
```
pymemo.lines.add('Decimal: '+
              eng.EvalStr('__import__("decimal").Decimal(0.1)'));

>>> 0.1000000000000000055511151231257827021181583404541015625
```



```
//public
  { Déclarations pub
//end;

const  PYDLLNAME = 'py
       PSCRIPTNAME = '
       PYHOME = 'C:\Us
       PYSCRIPT = 'C:\
       PYSCRIPT2 = 'C:
       LB = CR+LF;


const PyModule =
        'def printDat
        '  return dat

const PYCMD = 'print("
              'import
```

```
PSXCompiler: [Hint] 1050_pydem
[Hint] 1050_pydemo2.txt(301:6):
XCompiler Message Count: 8
 maXbox4 1050_pydemo2.txt Co
------------------------------------------------------
Exception: Cannot open file "C:\maXbox\mX47464\maxbox4\1046_openweather.py". specified.
PascalScript maXbox4 - RemObjects & SynEdit
```

Compile the Python Engine to run in maXbox and execute scripts in the code.

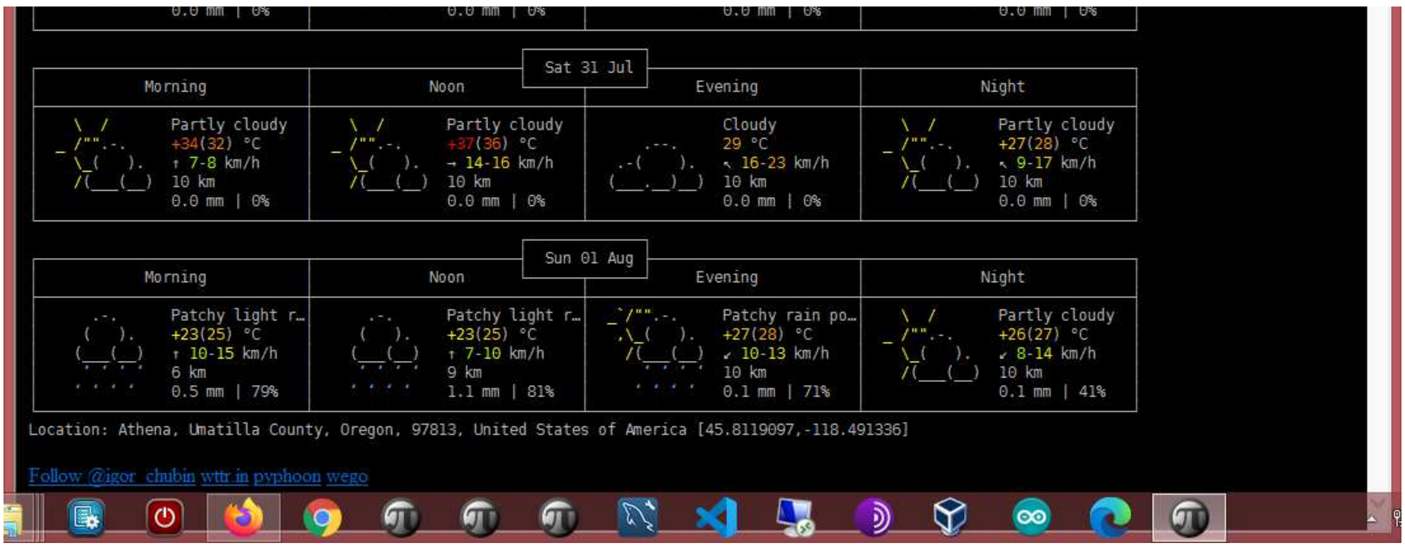


maXbox4 WinControlWebBrowserRSSStreamFeed_WeatherReport

```
Weather report: Athens,GR

     \   /     Sunny
      .-.      38 °C
   ― (   ) ―   ↑ 7 km/h
      `-'      10 km
     /   \     0.0 mm
                                                        ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤  Fri 30 Jul  ├───────────────────────┬──────────────────────────────┐
│            Morning           │             Noon      └──────┬──────┘     Evening           │             Night            │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────┼──────────────────────────────┤
│    \  /       Partly cloudy  │    \  /       Partly cloudy  │     \   /     Sunny          │     \   /     Clear          │
│  _ /"".-.     +37(36) °C     │  _ /"".-.     40 °C          │      .-.      +38(39) °C     │      .-.      +35(36) °C     │
│    \_(   ).   ↓ 8-9 km/h     │    \_(   ).   ← 5-6 km/h     │   ― (   ) ―   ↖ 6-7 km/h     │   ― (   ) ―   → 5-6 km/h     │
│    /(___(__)  10 km          │    /(___(__)  10 km          │      `-'      10 km          │      `-'      10 km          │
│               0.0 mm | 0%    │               0.0 mm | 0%    │     /   \     0.0 mm | 0%    │     /   \     0.0 mm | 0%    │
└──────────────────────────────┴──────────────────────────────┴──────────────────────────────┴──────────────────────────────┘
                                                        ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤  Sat 31 Jul  ├───────────────────────┬──────────────────────────────┐
│            Morning           │             Noon      └──────┬──────┘     Evening           │             Night            │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────┼──────────────────────────────┤
│     \   /     Sunny          │     \   /     Sunny          │     \   /     Sunny          │     \   /     Clear          │
│      .-.      37 °C          │      .-.      +38(37) °C     │      .-.      +37(36) °C     │      .-.      +36(35) °C     │
│   ― (   ) ―   ↗ 4-5 km/h     │   ― (   ) ―   ↑ 13-15 km/h   │   ― (   ) ―   ↑ 12-14 km/h   │   ― (   ) ―   ↑ 3-4 km/h     │
│      `-'      10 km          │      `-'      10 km          │      `-'      10 km          │      `-'      10 km          │
│     /   \     0.0 mm | 0%    │     /   \     0.0 mm | 0%    │     /   \     0.0 mm | 0%    │     /   \     0.0 mm | 0%    │
└──────────────────────────────┴──────────────────────────────┴──────────────────────────────┴──────────────────────────────┘
                                                        ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤  Sun 01 Aug  ├───────────────────────┬──────────────────────────────┐
│            Morning           │             Noon      └──────┬──────┘     Evening           │             Night            │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────┼──────────────────────────────┤
│     \   /     Sunny          │     \   /     Sunny          │     \   /     Sunny          │     \   /     Clear          │
│      .-.      +38(36) °C     │      .-.      40 °C          │      .-.      +39(40) °C     │      .-.      +35(37) °C     │
│   ― (   ) ―   ↖ 5-6 km/h     │   ― (   ) ―   → 4-5 km/h     │   ― (   ) ―   ↑ 6-7 km/h     │   ― (   ) ―   → 4-6 km/h     │
│      `-'      10 km          │      `-'      10 km          │      `-'      10 km          │      `-'      10 km          │
│     /   \     0.0 mm | 0%    │     /   \     0.0 mm | 0%    │     /   \     0.0 mm | 0%    │     /   \     0.0 mm | 0%    │
└──────────────────────────────┴──────────────────────────────┴──────────────────────────────┴──────────────────────────────┘
Location: Αθήνα, Δήμος Αθηναίων, Π.Ε. Κεντρικού Τομέα Αθηνών, Περιφέρεια Αττικής, Αποκεντρωμένη Διοίκηση Αττικής, Ελλάδα [37.9841493,23.7279843]

Follow @igor_chubin wttr.in pyphoon wego
```

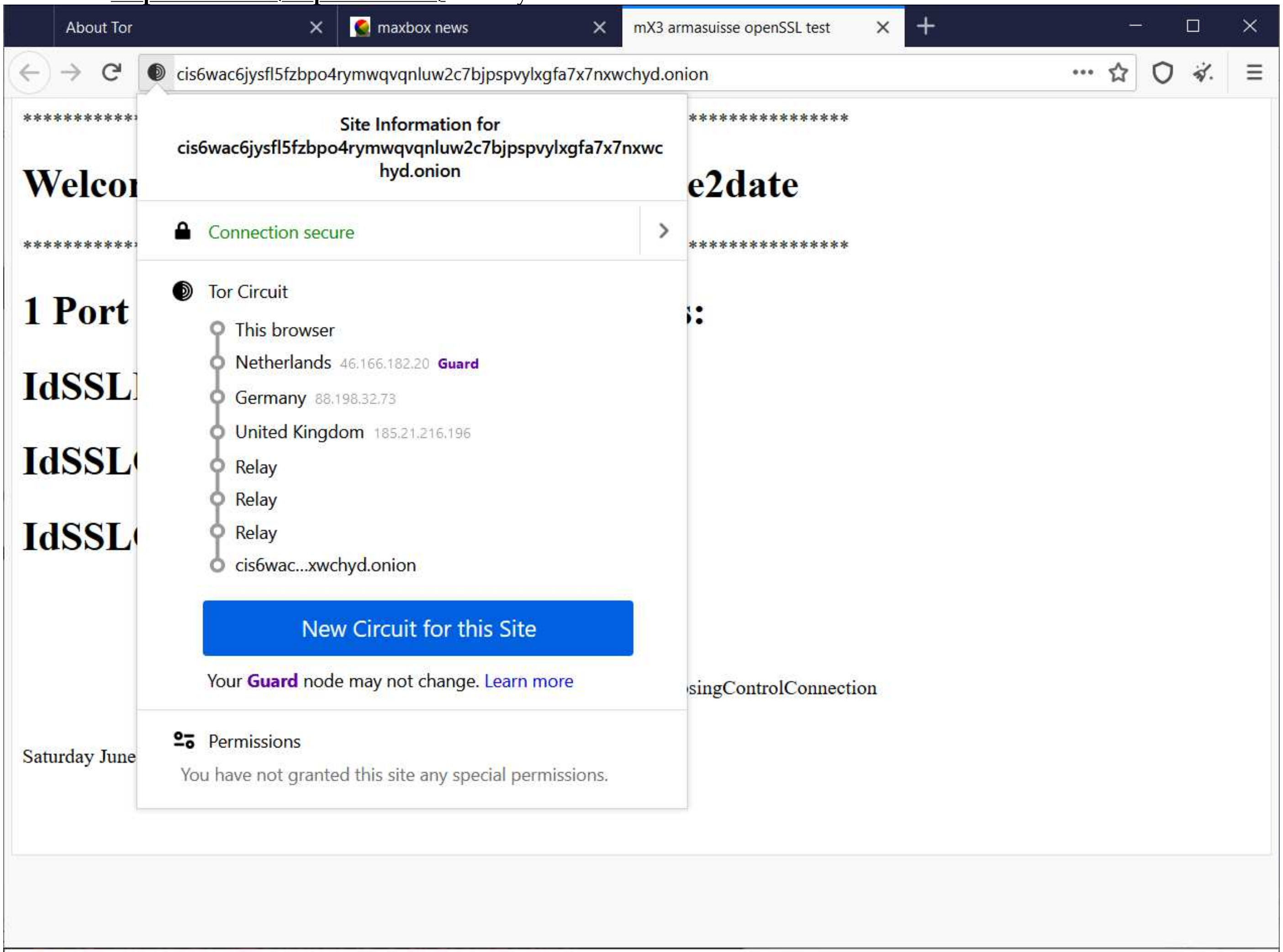
maXbox4 WinControlWebBrowserRSSStreamFeed_WeatherReport

```
Weather report: Athena

     \   /     Clear
      .-.      +26(25) °C
   ― (   ) ―   ↑ 11 km/h
      `-'      16 km
     /   \     0.0 mm
                                                        ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤  Fri 30 Jul  ├───────────────────────┬──────────────────────────────┐
│            Morning           │             Noon      └──────┬──────┘     Evening           │             Night            │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────┼──────────────────────────────┤
│    \  /       Partly cloudy  │    \  /       Partly cloudy  │    \  /       Partly cloudy  │    \  /       Partly cloudy  │
│  _ /"".-.     +31(29) °C     │  _ /"".-.     +36(35) °C     │  _ /"".-.     40 °C          │  _ /"".-.     36 °C          │
│    \_(   ).   ↘ 5-6 km/h     │    \_(   ).   ↓ 13-15 km/h   │    \_(   ).   ↓ 20-27 km/h   │    \_(   ).   ↙ 14-26 km/h   │
│    /(___(__)  10 km          │    /(___(__)  10 km          │    /(___(__)  10 km          │    /(___(__)  10 km          │
```



```
              0.0 mm | 0%          │              0.0 mm | 0%          │              0.0 mm | 0%          │              0.0 mm | 0%
                                                        ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤  Sat 31 Jul ├───────────────────────┬──────────────────────────────┐
│            Morning           │             Noon      └──────┬──────┘     Evening           │             Night            │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────┼──────────────────────────────┤
│    \  /       Partly cloudy  │    \  /       Partly cloudy  │               Cloudy         │    \  /       Partly cloudy  │
│  _ /"".-.     +34(32) °C     │  _ /"".-.     +37(36) °C     │      .--.     29 °C          │  _ /"".-.     +27(28) °C     │
│    \_(   ).   ↑ 7-8 km/h     │    \_(   ).   → 14-16 km/h   │   .-(    ).   ↖ 16-23 km/h   │    \_(   ).   ↖ 9-17 km/h    │
│    /(___(__)  10 km          │    /(___(__)  10 km          │  (___.__)__)  10 km          │    /(___(__)  10 km          │
│               0.0 mm | 0%    │               0.0 mm | 0%    │               0.0 mm | 0%    │               0.0 mm | 0%    │
└──────────────────────────────┴──────────────────────────────┴──────────────────────────────┴──────────────────────────────┘
                                                        ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤  Sun 01 Aug ├───────────────────────┬──────────────────────────────┐
│            Morning           │             Noon      └──────┬──────┘     Evening           │             Night            │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────┼──────────────────────────────┤
│     .-.       Patchy light r…│     .-.       Patchy light r…│ _`/"".-.      Patchy rain po…│    \  /       Partly cloudy  │
│    (   ).     +23(25) °C     │    (   ).     +23(25) °C     │  ,\_(   ).    +27(28) °C     │  _ /"".-.     +26(27) °C     │
│   (___(__)    ↑ 10-15 km/h   │   (___(__)    ↑ 7-10 km/h    │   /(___(__)   ↙ 10-13 km/h   │    \_(   ).   ↙ 8-14 km/h    │
│    ‘ ‘ ‘ ‘    6 km           │    ‘ ‘ ‘ ‘    9 km           │     ‘ ‘ ‘ ‘   10 km          │    /(___(__)  10 km          │
│   ‘ ‘ ‘ ‘     0.5 mm | 79%   │   ‘ ‘ ‘ ‘     1.1 mm | 81%   │    ‘ ‘ ‘ ‘    0.1 mm | 71%   │               0.1 mm | 41%   │
└──────────────────────────────┴──────────────────────────────┴──────────────────────────────┴──────────────────────────────┘
Location: Athena, Umatilla County, Oregon, 97813, United States of America [45.8119097,-118.491336]

Follow @igor_chubin wttr.in pyphoon wego
```

REST API of **https://wttr.in (https://wttr.in)** and PyOWM in combination out of maXbox



About Tor | maxbox news | mX3 armasuisse openSSL test

cis6wac6jysfl5fzbpo4rymwqvqnluw2c7bjpspvylxgfa7x7nxwchyd.onion

**************** ****************

Welco… …e2date

**************** ****************

1 Port … :

IdSSL…

IdSSL…

IdSSL…

…singControlConnection

Saturday June…

Site Information for cis6wac6jysfl5fzbpo4rymwqvqnluw2c7bjpspvylxgfa7x7nxwchyd.onion

Connection secure

Tor Circuit
- This browser
- Netherlands 46.166.182.20 Guard
- Germany 88.198.32.73
- United Kingdom 185.21.216.196
- Relay
- Relay
- Relay
- cis6wac...xwchyd.onion

New Circuit for this Site

Your **Guard** node may not change. Learn more

Permissions

You have not granted this site any special permissions.

Oracle VM VirtualBox Manager

Datei Maschine Hilfe

Werkzeuge

Neu Ändern Verwerfen Zeigen

linuxmint201 ausgeschaltet

kubuntu wird ausgeführt

Allgemein

Name: kubuntu
Betriebssystem: Ubuntu (64-bit)

System

Hauptspeicher: 1024 MB
Bootreihenfolge: Diskettenlaufwerk, Optisch, Platte
Beschleunigung: VT-x/AMD-V, Nested Paging, KVM-Paravirtualisierung

Anzeige

Vorschau

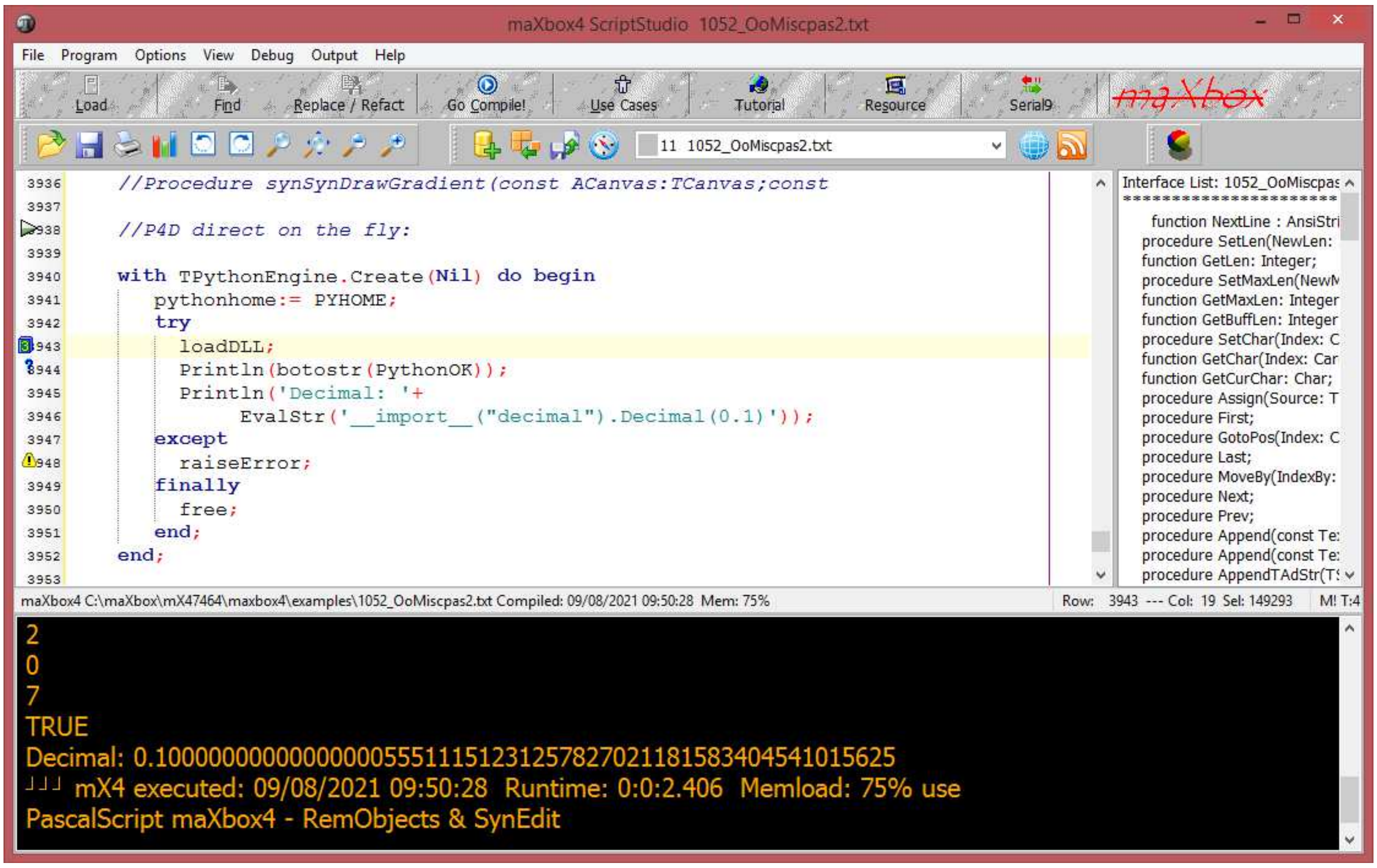
```
Grafikspeicher:    16 MB
Grafikcontroller:  VMSVGA
Fernsteuerung:     deaktiviert
Aufnahme:          deaktiviert

Massenspeicher
Controller: IDE
  Sekundärer Master:   [Optisches Laufwerk] leer
Controller: SATA
  SATA-Port 0:         Kubuntu 21.04 (64bit).vdi (normal, 500.00 GB)

Audio
```

Kubuntu 21 with maXbox and Python on Wine 5.1

Minimal Configuration Py on the fly

The minimal configuration depends on your Python-installation and the *UseLastKnownVersion* property in *TDynamicDll* but once known it goes like this with *raiseError* to get the Python exceptions:

```
with TPythonEngine.Create(Nil) do begin
  pythonhome:= PYHOME;
  try
    loadDLL;
    Println('Decimal: '+
          EvalStr('__import__("decimal").Decimal(0.1)'));
  except
    raiseError;
  finally
    free;
  end;
end;
```


maXbox4 ScriptStudio 1052_OoMiscpas2.txt
File Program Options View Debug Output Help
Load Find Replace / Refact Go Compile! Use Cases Tutorial Resource Serial9
11 1052_OoMiscpas2.txt

```
//Procedure synSynDrawGradient(const ACanvas:TCanvas;const

//P4D direct on the fly:

with TPythonEngine.Create(Nil) do begin
   pythonhome:= PYHOME;
   try
     loadDLL;
     Println(botostr(PythonOK));
     Println('Decimal: '+
          EvalStr('__import__("decimal").Decimal(0.1)'));
   except
     raiseError;
   finally
     free;
   end;
end;
```

```
Interface List: 1052_OoMiscpas
************************
  function NextLine : AnsiStri
procedure SetLen(NewLen:
function GetLen: Integer;
procedure SetMaxLen(NewM
function GetMaxLen: Integer
function GetBuffLen: Integer
procedure SetChar(Index: C
function GetChar(Index: Car
function GetCurChar: Char;
procedure Assign(Source: T
procedure First;
procedure GotoPos(Index: C
procedure Last;
procedure MoveBy(IndexBy:
procedure Next;
procedure Prev;
procedure Append(const Te:
procedure Append(const Te:
procedure AppendTAdStr(T:
```

```
maXbox4 C:\maXbox\mX47464\maxbox4\examples\1052_OoMiscpas2.txt Compiled: 09/08/2021 09:50:28 Mem: 75%    Row: 3943 --- Col: 19 Sel: 149293    M! T:4
```

```
2
0
7
TRUE
Decimal: 0.1000000000000000055511151231257827021181583404541015625
┘┘┘ mX4 executed: 09/08/2021 09:50:28  Runtime: 0:0:2.406  Memload: 75% use
PascalScript maXbox4 - RemObjects & SynEdit
```

Py on the Fly

Posted in **maXbox**, **Python** Tagged **Coding**, **Git** **10 Comments**

10 thoughts on “Python4maXbox Code”

1. **maxbox4** says:
   July 28, 2021 at 9:45 pm **Edit**
   ```
   function CleanString(const s : AnsiString; AppendLF : Boolean) : AnsiString;
   var
   i : Integer;
   begin
   result := s;
   if s = ” then
   Exit;
   i := Pos(AnsiString(CR),s);
   while i > 0 do
   begin
   Delete( result, i, 1 );
   i := PosEx(AnsiString(CR),result, i); //fix
   end;
   if AppendLF and (result[length(result)] LF) then
   Result := Result + LF;
   end;
   ```
   REPLY
   1. **maxbox4** says:
      July 30, 2021 at 3:07 pm **Edit**
      ```
      >>> mystr
      ‘hello world, how do i enter line breaks?’
      >>> mystr.replace(‘ ‘, ”)
      ‘helloworld,howdoienterlinebreaks?’
      ```
      In the example above, I replaced all spaces. The string ‘\n’ represents newlines. And \r represents carriage returns (if you’re on windows, you might be getting these and a second replace will handle them for you!).
      basically:

      ```
      # you probably want to use a space ‘ ‘ to replace `\n`
      mystring = mystring.replace(‘\n’, ‘ ‘).replace(‘\r’, ”)
      ```
      Note also, that it is a bad idea to call your variable string, as this shadows the module string. Another name I’d avoid but would love to use sometimes: file. For the same reason a semantic syntax checker.

      REPLY
      1. **maxbox4** says:
         July 30, 2021 at 3:13 pm **Edit**
         Create a string containing line breaks
         Newline code \n（LF）, \r\n（CR + LF）
         Triple quote ”’ or “””
         With indent
         Concatenate a list of strings on new lines
         Split a string into a list by line breaks: splitlines()
         Remove or replace line breaks
         Output with print() without a trailing newline
2. **breitsch breitsch** says:
   July 31, 2021 at 6:30 pm **Edit**
   Tested on Win10 and maXbox4
   Download installer: **https://www.python.org/ftp/python/3.7.9/python-3.7.9.exe**
   Set path:
   pydllpath= ‘C:\Users\breitsch\AppData\Local\Programs\Python\Python37-32\python37.dll’;

Load it:
pythonengine.openDll(pydllpath);
Test it:
PrintLn('builtins: '+ pythonengine.EvalStr('dir(__builtins__)'));

REPLY

breitsch breitsch says:
August 1, 2021 at 5:39 pm **Edit**
writeln('wget: '+ eng.EvalStr('__import__("urllib")'));
writeln('wget_like: '+ eng.EvalStr('__import__("urllib.request").request.urlretrieve("http://google.com/index.html", filename="indexg.html")'));

REPLY

maxbox4 says:
August 2, 2021 at 7:55 am **Edit**
If you get the error: Exception: : DLL load failed: %1 is not a valid Win32 application.
Solution set the pythonhome to 32bit:
PYHOME = 'C:\Users\max\AppData\Local\Programs\Python\Python36-32\';
eng.pythonhome:= PYHOME;
>>>wget:

3.
breitsch breitsch says:
August 5, 2021 at 6:54 pm **Edit**
Install from within script, e.g. numpy:

eng.ExecString('import subprocess');
eng.ExecString('subprocess.call(["pip", "install", "numpy"])')
eng.ExecString('import numpy');

REPLY

breitsch breitsch says:
August 6, 2021 at 4:35 pm **Edit**
Complete Scenario in maXbox:
1. install SymPy
eng.ExecString('import subprocess');
eng.ExecString('subprocess.call(["pip", "install", "sympy"])')
2. compute solver
avar:= '-2';
eng.ExecString('x=Symbol("x")');
eng.ExecString('x,y=symbols("x y")');
writeln(eng.EvalStr('solveset(Eq(x**2-9,0), x)'));
writeln(eng.EvalStr('solveset(Eq(x**2-3*x,'+avar+'),x)'));
writeln(eng.EvalStr('solveset(exp(x),x)'));
println(eng.EvalStr('linsolve([Eq(x-y,4),Eq(x + y ,1)], (x, y))'));
3. check results:
FiniteSet(-3, 3)
FiniteSet(1, 2)
EmptySet
FiniteSet((5/2, -3/2))

https://www.tutorialspoint.com/sympy/sympy_solvers.htm

REPLY

breitsch breitsch says:
August 6, 2021 at 4:47 pm **Edit**
Another 4 liner:
eng.ExecString('import subprocess');
eng.ExecString('subprocess.call(["pip", "install", "langdetect"])')
eng.ExecString('from langdetect import detect');
println('detect: '+eng.EvalStr('detect("bonjour mes ordinateurs")'));
>>> detect: fr

maxbox4 says:
August 8, 2021 at 12:47 pm **Edit**
Great, be sure that Pyhome and Pydll are of the same filespace when installing a package:

```
const
PYHOME = ‘C:\Users\max\AppData\Local\Programs\Python\Python36-32\’;
PYDLL = ‘C:\Users\max\AppData\Local\Programs\Python\Python36-32\python36.dll’;
eng.pythonhome:= PYHOME;
eng.opendll(PYDLL);
```

REPLY

Create a free website or blog at WordPress.com.