//

Migrate to 64-bit-box

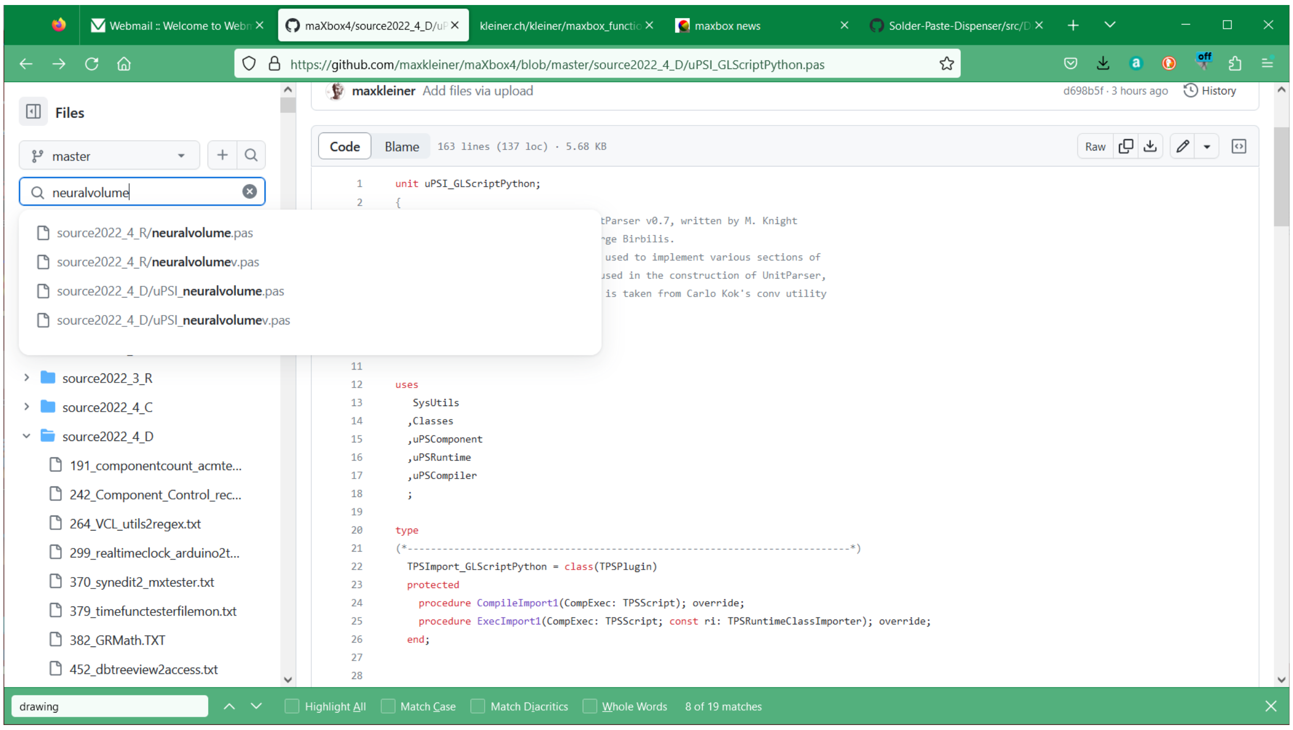
____

maXbox Starter 113 – Convert a 32-bit application to 64-bit from Delphi 2007 to Delphi 10.4 Sydney.

“Lost in translation – ghost in application” -Hardcore code.

Source: firstdemo_master11_cop21web.txt and my6 cannonball.txt blog

If you have a code base of 32-bit Windows Delphi applications that you want to convert to 64-bit Windows, you should first do a reorganisation of the sources to get an overview. The Source is organised in _C for Components and _R for Runtime (native Units) and _D for Design Units (script mapping imports) like the following graph of Package Neuralvolume of CAI NeuralNetwork shows as 4 files:


https://github.com/maxkleiner/maXbox4/blob/master/source2022_4_D/uPSI_GLScriptPython.pas

maxkleiner Add files via upload d698b5f · 3 hours ago History

Files

master

neuralvolume

source2022_4_R/neuralvolume.pas
source2022_4_R/neuralvolumev.pas
source2022_4_D/uPSI_neuralvolume.pas
source2022_4_D/uPSI_neuralvolumev.pas

source2022_3_R
source2022_4_C
source2022_4_D
191_componentcount_acmte...
242_Component_Control_rec...
264_VCL_utils2regex.txt
299_realtimeclock_arduino2t...
370_synedit2_mxtester.txt
379_timefunctesterfilemon.txt
382_GRMath.TXT
452_dbtreeview2access.txt

Code Blame 163 lines (137 loc) · 5.68 KB

```
unit uPSI_GLScriptPython;
{
                                        tParser v0.7, written by M. Knight
                                        ge Birbilis.
                                         used to implement various sections of
                                        used in the construction of UnitParser,
                                         is taken from Carlo Kok's conv utility

uses
   SysUtils
  ,Classes
  ,uPSComponent
  ,uPSRuntime
  ,uPSCompiler
  ;

type
(*----------------------------------------------------------------------------*)
  TPSImport_GLScriptPython = class(TPSPlugin)
  protected
    procedure CompileImport1(CompExec: TPSScript); override;
    procedure ExecImport1(CompExec: TPSScript; const ri: TPSRuntimeClassImporter); override;
  end;
```

drawing 8 of 19 matches

pic1: 1_sourceorganisation.png

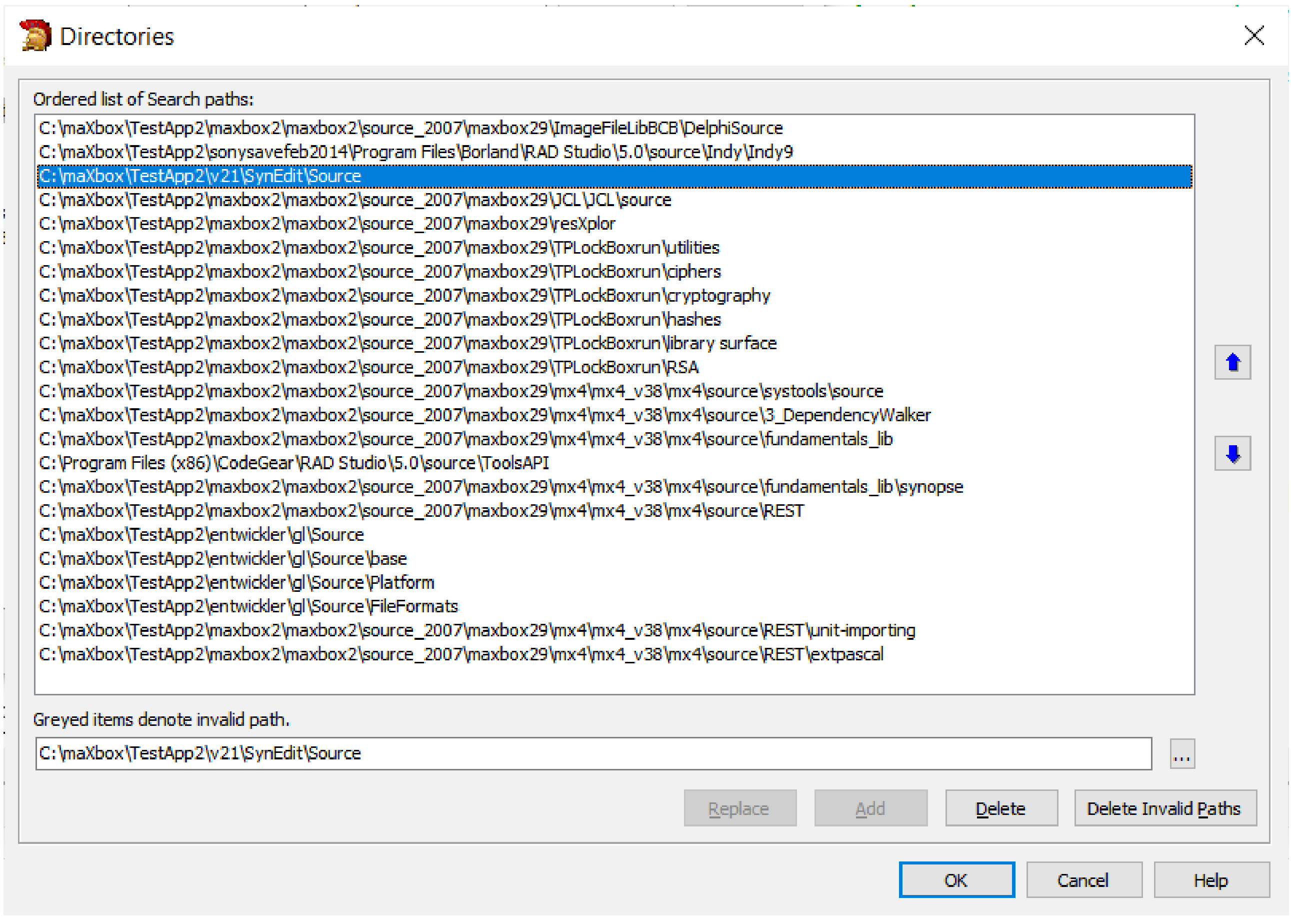
So its not that easy open your 32-bit application in the IDE, add and activate the 64-bit Windows target platform, and compile your application as a 64-bit Windows application.
While digging or diving through the source code of maXbox4 it seems to be impossible to migrate over 3300 units (exactly 3335) in a decent and proper way to maXbox5 aka 64.bit Version.

Source Organisation

First thing I must say is the missing support of any kind of the whole "HiRes/4K or DPI Awareness Resolution and Delphi forms" revolution;

There's no “Make my form look right on all resolutions" checkbox or a emulator which go through all forms. But you can drawback to the "don't support hi-DPI" setting. I know this is not the improvement we want, but this gives you the least headaches. As in mX4 and for the forthcoming mX5 the App is “out of the box” (self containment) and needs no installation nor registration. It has a independent system architecture (ISA).

So for the reorganisation of the sources I have the latest revision with patches from issue #202 (commit 86a057c) but I am unable to compile the files at first (Core_D26) that are part of the **PascalScript_Core_D27.dpk** for that platform for Linux64, Win64 nor MacOS64.



Directories

Ordered list of Search paths:

```
C:\maXbox\TestApp2\maxbox2\maxbox2\source_2007\maxbox29\ImageFileLibBCB\DelphiSource
C:\maXbox\TestApp2\sonysavefeb2014\Program Files\Borland\RAD Studio\5.0\source\Indy\Indy9
C:\maXbox\TestApp2\v21\SynEdit\Source
C:\maXbox\TestApp2\maxbox2\maxbox2\source_2007\maxbox29\JCL\JCL\source
C:\maXbox\TestApp2\maxbox2\maxbox2\source_2007\maxbox29\resXplor
C:\maXbox\TestApp2\maxbox2\maxbox2\source_2007\maxbox29\TPLockBoxrun\utilities
C:\maXbox\TestApp2\maxbox2\maxbox2\source_2007\maxbox29\TPLockBoxrun\ciphers
C:\maXbox\TestApp2\maxbox2\maxbox2\source_2007\maxbox29\TPLockBoxrun\cryptography
C:\maXbox\TestApp2\maxbox2\maxbox2\source_2007\maxbox29\TPLockBoxrun\hashes
C:\maXbox\TestApp2\maxbox2\maxbox2\source_2007\maxbox29\TPLockBoxrun\library surface
C:\maXbox\TestApp2\maxbox2\maxbox2\source_2007\maxbox29\TPLockBoxrun\RSA
C:\maXbox\TestApp2\maxbox2\maxbox2\source_2007\maxbox29\mx4\mx4_v38\mx4\source\systools\source
C:\maXbox\TestApp2\maxbox2\maxbox2\source_2007\maxbox29\mx4\mx4_v38\mx4\source\3_DependencyWalker
C:\maXbox\TestApp2\maxbox2\maxbox2\source_2007\maxbox29\mx4\mx4_v38\mx4\source\fundamentals_lib
C:\Program Files (x86)\CodeGear\RAD Studio\5.0\source\ToolsAPI
C:\maXbox\TestApp2\maxbox2\maxbox2\source_2007\maxbox29\mx4\mx4_v38\mx4\source\fundamentals_lib\synopse
C:\maXbox\TestApp2\maxbox2\maxbox2\source_2007\maxbox29\mx4\mx4_v38\mx4\source\REST
C:\maXbox\TestApp2\entwickler\gl\Source
C:\maXbox\TestApp2\entwickler\gl\Source\base
C:\maXbox\TestApp2\entwickler\gl\Source\Platform
C:\maXbox\TestApp2\entwickler\gl\Source\FileFormats
C:\maXbox\TestApp2\maxbox2\maxbox2\source_2007\maxbox29\mx4\mx4_v38\mx4\source\REST\unit-importing
C:\maXbox\TestApp2\maxbox2\maxbox2\source_2007\maxbox29\mx4\mx4_v38\mx4\source\REST\extpascal
```

Greyed items denote invalid path.

C:\maXbox\TestApp2\v21\SynEdit\Source

Pic2: 2_sourceorganisation2007.png

In Delphi, I can include a folder's source code by adding it to the project Search Path, or adding it to the Library Path. The Search Path applies only to the current project, while the Library Path applies to any project opened with the IDE. But hope other than that, is there any functional difference between the Search and Library paths?
The reason is I have a folder X with source used by project A. When I include that folder under Project A's search path, it says it cannot find a specific file in that folder. When I include it under the Library path, then test project A compiles fine.
As far as I know, browsing path is where the debugger should look for files when breaking/stepping into source files that's not in the library path. Lets say that you have a third-party component that you use. You point the library path to the directory where the pre-compiled dcu-files

of that component are. Your project will use these dcu-files when you compile. This is obvious, because it wont be recompiled every time you do a build. The default settings for the VCL show this. In library path they have put $(BSD)\Lib, and in the browsing path they have put $(BDS)\SOURCE\WIN32...

Here's some compiler output at first to compare using Delphi 10.3.2 Rio and then 10.4 dccosx64 or dcc64 compiler (similar results exist for dcclinux64):

```
[dcc] ./PascalScript_Core_D26.dpk
[dcc] ./uPSUtils.pas (730)
[dcc] Error: E2008 Incompatible types
[dcc] ./uPSCompiler.pas (1374)
[dcc] Fatal: F2063 Could not compile used unit 'uPSUtils.pas'
```

If I comment out that offending code then I get the following which is starting to look non-trivial...

```
[dcc] ./PascalScript_Core_D26.dpk
[dcc] ./uPSRuntime.pas (8923)
[dcc] Warning: W1057 Implicit string cast from 'AnsiString' to 'string'
[dcc] ./uPSRuntime.pas (11640)
[dcc] Error: E1025 Unsupported language feature: 'ASM'
[dcc] ./uPSRuntime.pas (11640)
[dcc] Error: E2029 ';' expected but 'ASM' found
[dcc] ./uPSRuntime.pas (11640)
[dcc] Warning: W1011 Text after final 'END.' - ignored by compiler
[dcc] ./uPSRuntime.pas (58)
[dcc] Error: E2065 Unsatisfied forward or external declaration:
'TPSProcRec.Create'
```

The reason of all problems in OSX64 (and Linux64, i think also) with PS is " The LongInt and LongWord Data Type are different on 64-bit POSIX platforms.
To keep interoperability between Delphi and POSIX API, for 64-bit POSIX platforms, the size of LongInt and LongWord types are changed to 64-bit. All 32-bit platforms and 64-bit Windows platforms keep 32-bit for the LongInt and LongWord types."

So, fixing can be very simple - change ALL LongInt type to Integer.
Files: uPSCompiler, uPSComponent, uPSDebugger, uPSRuntime, uPSUtils.

But don't change it by auto-replace from "Longint" to "Integer", because in this case declarations like AddTypeCopyN('Integer', 'LongInt');

and others – it as the reference as string literal will be broken.

I stubbed out 2 assembler routines which I hope could be translated to pure pascal by someone who understands their intent. I'm not really sure what the assembler is doing

```
procedure MyAllMethodsHandler;
procedure PutOnFPUStackExtended(ft: extended);
```

Perhaps as suggested we can just {$DEFINE empty_methods_handler} to avoid the assembler. But I don't know what its trying to do, so is a stub acceptable or do we need it to do something? As you can see the work for the 64bit box has begun but libs, maps, object-files, transpiler and registering is full of traps. If “Use Debug DCUs” option is not activated, and I debug our application, I can only single step through my own code. This is what we want in the most cases, because it is our code that is buggy, not Delphi’s code normally. It will be quite annoying to keep stepping into Delphi’s code.

Reorganisation

After restructured the source from Delphi 2007 (see pic 2) to Github and configured in Delphi 10.4 I begun to review and handle the following issues (mostly related to pointed operations in WinAPI issues, NativeInt size, and Assembly code):

If you pass pointers to SendMessage/PostMessage/TControl.Perform, the wParam and lParam parameters should be type-casted to the WPARAM/LPARAM type and not to Integer/Longint.

```
    Correct: SendMessage(hWnd, WM_SETTEXT, 0, LPARAM(@MyCharArray));

    Wrong: SendMessage(hWnd, WM_SETTEXT, 0, Integer(@MyCharArray));
```

Replace SetWindowLong/GetWindowLog with SetWindowLongPtr/GetWindowLongPtr for GWLP_HINSTANCE, GWLP_ID, GWLP_USERDATA, GWLP_HWNDPARENT and GWLP_WNDPROC as they return pointers and handles. Pointers that are passed to SetWindowLongPtr should be type-casted to LONG_PTR and not to Integer/Longint.

```
    Correct: SetWindowLongPtr(hWnd, GWLP_WNDPROC,
                                 LONG_PTR(@MyWindowProc));

    Wrong: SetWindowLong(hWnd, GWL_WNDPROC, Longint(@MyWindowProc));
```

In the **runtime library** several issues had to be done:

```
<?xml version="1.0" encoding="UTF-8"?>
{$IFNDEF WIN32}
  'This components are for 32bitDelphi only!'???›??
{$ENDIF}
```

Review the IFNDEF WIN32 in the sense: try to convert or leave!

In Delphi, strings as result values are treated like var parameters. In other words, a function like Foo is in fact compiled as:

```
function Foo(): String;
begin
  Result := 'foo';
  RaiseException('...');
end;

procedure Foo(var Result: string);
begin
  Result := 'Foo';
  RaiseException(...);
end;
```

So the chance to convert lines of WIN32 to WIN64 is possible with references or type-less references like procedure Foo(var Result;);

Overloads: For functions that took PChar, there are now PAnsiChar and PWideChar versions so the appropriate function gets called.

AnsiXXX functions are a consideration:

- SysUtils.AnsiXXXX functions, such as AnsiCompareStr:
- Remain declared with string and float to UnicodeString.
- Offer better backward compatibility (no need to change code).

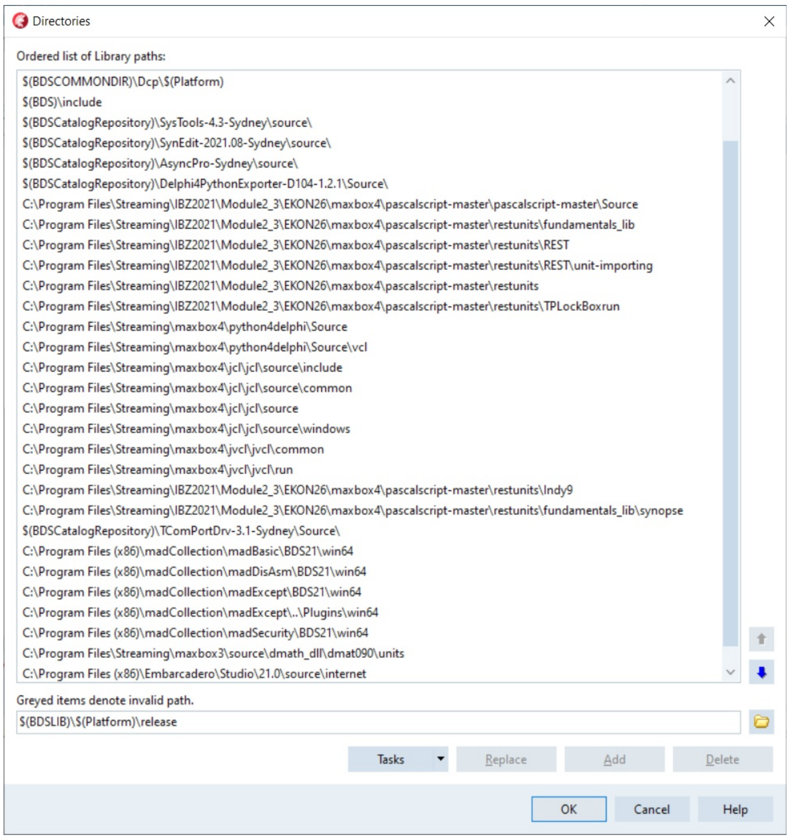
The AnsiStrings unit's AnsiXXXX functions offer the same capabilities as the SysUtils.AnsiXXXX functions, but work only for AnsiString. Also, the AnsiStrings.AnsiXXXX functions provide better performance for an AString than SysUtils.AnsiXXXX functions, which work for both AnsiString and UnicodeString, because no implicit conversions are performed.

String information functions:

- StringElementSize returns the actual data size.
- StringCodePage returns the code page of string data.
- System.StringRefCount returns the reference count.

RTL provides many helper functions that enable users to do explicit conversions between code pages and element size conversions. If developers are using the Move function on a character array, they cannot make assumptions about the element size. Much of this problem can be mitigated by making sure all RValue references generate the proper calls to RTL to ensure proper element sizes.

In the meantime I got the import and list in D10.4:



Ordered list of Library paths:

```
$(BDSCOMMONDIR)\Dcp\$(Platform)
$(BDS)\include
$(BDSCatalogRepository)\SysTools-4.3-Sydney\source\
$(BDSCatalogRepository)\SynEdit-2021.08-Sydney\source\
$(BDSCatalogRepository)\AsyncPro-Sydney\source\
$(BDSCatalogRepository)\Delphi4PythonExporter-D104-1.2.1\Source\
C:\Program Files\Streaming\IBZ2021\Module2_3\EKON26\maxbox4\pascalscript-master\pascalscript-master\Source
C:\Program Files\Streaming\IBZ2021\Module2_3\EKON26\maxbox4\pascalscript-master\restunits\fundamentals_lib
C:\Program Files\Streaming\IBZ2021\Module2_3\EKON26\maxbox4\pascalscript-master\restunits\REST
C:\Program Files\Streaming\IBZ2021\Module2_3\EKON26\maxbox4\pascalscript-master\restunits\REST\unit-importing
C:\Program Files\Streaming\IBZ2021\Module2_3\EKON26\maxbox4\pascalscript-master\restunits
C:\Program Files\Streaming\IBZ2021\Module2_3\EKON26\maxbox4\pascalscript-master\restunits\TPLockBoxrun
C:\Program Files\Streaming\maxbox4\python4delphi\Source
C:\Program Files\Streaming\maxbox4\python4delphi\Source\vcl
C:\Program Files\Streaming\maxbox4\jcl\jcl\source\include
C:\Program Files\Streaming\maxbox4\jcl\jcl\source\common
C:\Program Files\Streaming\maxbox4\jcl\jcl\source
C:\Program Files\Streaming\maxbox4\jcl\jcl\source\windows
C:\Program Files\Streaming\maxbox4\jvcl\jvcl\common
C:\Program Files\Streaming\maxbox4\jvcl\jvcl\run
C:\Program Files\Streaming\IBZ2021\Module2_3\EKON26\maxbox4\pascalscript-master\restunits\Indy9
C:\Program Files\Streaming\IBZ2021\Module2_3\EKON26\maxbox4\pascalscript-master\restunits\fundamentals_lib\synopse
$(BDSCatalogRepository)\TComPortDrv-3.1-Sydney\Source\
C:\Program Files (x86)\madCollection\madBasic\BDS21\win64
C:\Program Files (x86)\madCollection\madDisAsm\BDS21\win64
C:\Program Files (x86)\madCollection\madExcept\BDS21\win64
C:\Program Files (x86)\madCollection\madExcept\..\Plugins\win64
C:\Program Files (x86)\madCollection\madSecurity\BDS21\win64
C:\Program Files\Streaming\maxbox3\source\dmath_dll\dmat090\units
C:\Program Files (x86)\Embarcadero\Studio\21.0\source\internet
```

Greyed items denote invalid path.

```
$(BDSLIB)\$(Platform)\release
```

pic3: 3_sourceorg_pathlibrarylist2023.png

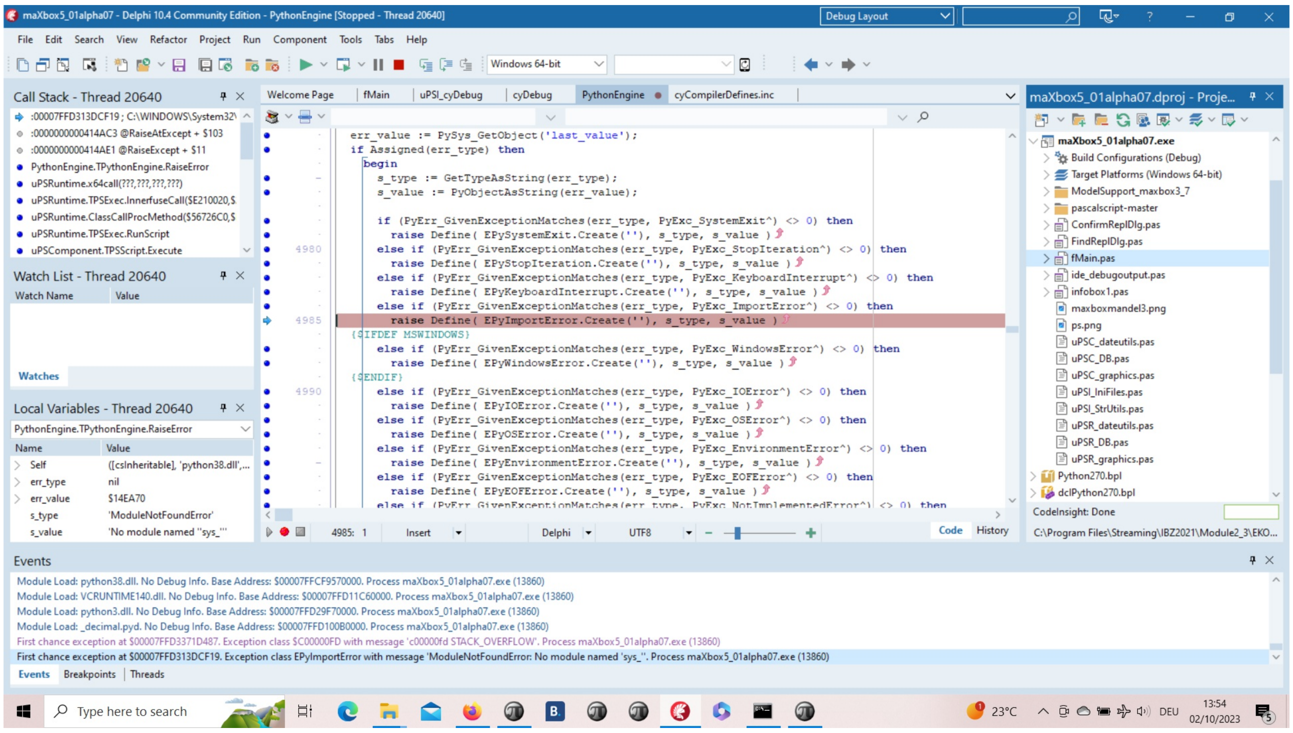
A big mess was or is the Tencoding (not finished yet) because Tencoding makes the real difference from ANSI to unicode:

- Defaults to users' active code page.
- Supports UTF-8.
- Supports UTF-16, big and little endian.
- Byte Order Mark (BOM) support.
- You can create descendent classes for user-specific encodings.

You need to perform these steps:

1. Review char- and string-related functions.
2. Rebuild the application.
3. Review surrogate pairs.
4. Review string payloads.

Night after night I got many AV of these type:
Exception code 0xc0000005 is an Access Violation. An AV at fault offset 0x00000000 means that something in your service's code is accessing a nil pointer. You will just have to debug the service while it is running to find out what it is accessing. If you cannot run it inside a debugger, then at least install a third-party exception logger framework, such as EurekaLog or MadExcept, to find out what your service was doing at the time of the AV. Most of the single time you get an AV in a unit with a initialisation section of e.g. a Setmem or some memory allocation stuff.


maXbox5_01alpha07 - Delphi 10.4 Community Edition - PythonEngine [Stopped - Thread 20640]
Debug Layout
File Edit Search View Refactor Project Run Component Tools Tabs Help
Windows 64-bit
Welcome Page | fMain | uPSI_cyDebug | cyDebug | PythonEngine | cyCompilerDefines.inc
maXbox5_01alpha07.dproj - Proje...
maXbox5_01alpha07.exe
Build Configurations (Debug)
Target Platforms (Windows 64-bit)
ModelSupport_maxbox3_7
pascalscript-master
ConfirmReplDlg.pas
FindReplDlg.pas
fMain.pas
ide_debugoutput.pas
infobox1.pas
maxboxmandel3.png
ps.png
uPSC_dateutils.pas
uPSC_DB.pas
uPSC_graphics.pas
uPSI_IniFiles.pas
uPSI_StrUtils.pas
uPSR_dateutils.pas
uPSR_DB.pas
uPSR_graphics.pas
Python270.bpl
dclPython270.bpl
CodeInsight: Done
4985: 1 | Insert | Delphi | UTF8 | Code | History
Type here to search
23°C DEU 13:54 02/10/2023

Call Stack - Thread 20640

```
:00007FFD313DCF19 ; C:\WINDOWS\System32\
:0000000000414AC3 @RaiseAtExcept + $103
:0000000000414AE1 @RaiseExcept + $11
PythonEngine.TPythonEngine.RaiseError
uPSRuntime.x64call(???,???,???,???)
uPSRuntime.TPSExec.InnerfuseCall($E210020,$
uPSRuntime.ClassCallProcMethod($56726C0,$
uPSRuntime.TPSExec.RunScript
uPSComponent.TPSScript.Execute
```

Watch List - Thread 20640

Watch Name	Value

Local Variables - Thread 20640
PythonEngine.TPythonEngine.RaiseError

Name	Value
Self	([csInheritable], 'python38.dll',...
err_type	nil
err_value	$14EA70
s_type	'ModuleNotFoundError'
s_value	'No module named ''sys_'''

```
  err_value := PySys_GetObject('last_value');
  if Assigned(err_type) then
    begin
      s_type := GetTypeAsString(err_type);
      s_value := PyObjectAsString(err_value);

      if (PyErr_GivenExceptionMatches(err_type, PyExc_SystemExit^) <> 0) then
        raise Define( EPySystemExit.Create(''), s_type, s_value )
      else if (PyErr_GivenExceptionMatches(err_type, PyExc_StopIteration^) <> 0) then
        raise Define( EPyStopIteration.Create(''), s_type, s_value )
      else if (PyErr_GivenExceptionMatches(err_type, PyExc_KeyboardInterrupt^) <> 0) then
        raise Define( EPyKeyboardInterrupt.Create(''), s_type, s_value )
      else if (PyErr_GivenExceptionMatches(err_type, PyExc_ImportError^) <> 0) then
        raise Define( EPyImportError.Create(''), s_type, s_value )
  {$IFDEF MSWINDOWS}
      else if (PyErr_GivenExceptionMatches(err_type, PyExc_WindowsError^) <> 0) then
        raise Define( EPyWindowsError.Create(''), s_type, s_value )
  {$ENDIF}
      else if (PyErr_GivenExceptionMatches(err_type, PyExc_IOError^) <> 0) then
        raise Define( EPyIOError.Create(''), s_type, s_value )
      else if (PyErr_GivenExceptionMatches(err_type, PyExc_OSError^) <> 0) then
        raise Define( EPyOSError.Create(''), s_type, s_value )
      else if (PyErr_GivenExceptionMatches(err_type, PyExc_EnvironmentError^) <> 0) then
        raise Define( EPyEnvironmentError.Create(''), s_type, s_value )
      else if (PyErr_GivenExceptionMatches(err_type, PyExc_EOFError^) <> 0) then
        raise Define( EPyEOFError.Create(''), s_type, s_value )
      else if (PyErr_GivenExceptionMatches(err_type, PyExc_NotImplementedError^) <> 0) then
```

Events

```
Module Load: python38.dll. No Debug Info. Base Address: $00007FFCF9570000. Process maXbox5_01alpha07.exe (13860)
Module Load: VCRUNTIME140.dll. No Debug Info. Base Address: $00007FFD11C60000. Process maXbox5_01alpha07.exe (13860)
Module Load: python3.dll. No Debug Info. Base Address: $00007FFD29F70000. Process maXbox5_01alpha07.exe (13860)
Module Load: _decimal.pyd. No Debug Info. Base Address: $00007FFD100B0000. Process maXbox5_01alpha07.exe (13860)
First chance exception at $00007FFD3371D487. Exception class $C00000FD with message 'c00000fd STACK_OVERFLOW'. Process maXbox5_01alpha07.exe (13860)
First chance exception at $00007FFD313DCF19. Exception class EPyImportError with message 'ModuleNotFoundError: No module named 'sys_''. Process maXbox5_01alpha07.exe (13860)
```

Events | Breakpoints | Threads

Pic: 4_buildexceptioncatchinfo_mX5.png

This means that an exception was thrown, but there's no catch handler for it anywhere. This is most likely a programming error, probably on your part. It looks like you called an OpenCV or API function which failed by throwing a CV::Exception, but you're not catching it.

This would normally lead to a crash, but since you're running inside a debugger, you get the option to ignore this exception. That's what the Continue button will do on this dialogue. So instead of throwing an exception, the code will just continue executing as if nothing had happened. This is likely to fail eventually, as an error condition has now been ignored.
A real horror was to convert the Format() Function not in Delphi but to work in PascalScript64 at runtime in a script, so here's the function in Delphi in system RTL:

```
Function Format(fmt : String; params : array of const) : String;
var
  pdw1, pdw2 : PDWORD;
  i : integer;
  pc : PChar;
begin
  pdw1 := nil;
  if length(params) > 0 then GetMem(pdw1, length(params) *
                                                     sizeof(Pointer));
  pdw2 := pdw1;
  for i := 0 to high(params) do begin
    pdw2^ := DWORD(PDWORD(@params[i])^);
    inc(pdw2);
  end;
  GetMem(pc, 1024 - 1);
  try
    SetString(Result, pc, wvsprintf(pc, PwideCHAR(fmt),
                             PwideCHAR(pdw1)));    //fix from pchar?
  except
    Result := #0;
  end;
  if (pdw1 <> nil) then FreeMem(pdw1);
  if (pc <> nil) then FreeMem(pc);
end;
```

At the core its a wvsprintf but the question was unicode available or not? The Byte Order Mark (BOM) should be added to files to indicate their encoding and the function returns a string. After reimport and rebuild and cast to PwideCHAR it works now.

Compatibility Compilation

When running a 32bit process, on a 64bit version of windows, and having the large address aware flag set, pointers in the 2-4GB range are valid. In that case the request of a varInt64 could, if I'm not mistaken, result in positive values in the 2-4GB range being returned. If NativeInt is a signed 32bit int, that would then result in a range violation. I'm not sure if that NativeInt / NativeUInt cast in between here is needed at all. NativeInt and NativeUInt are signed/unsigned and their size is of the targeted platform, hence 32 or 64 bits.
That's why I believe the cast changes are not needed.

Most of the times you deal with pointers or assembler code like:

```
function StrToWord(const Value: String): Word;
begin
  Result:= Word(pointer(@Value[1])^);
end;

function WordToStr(const Value: Word): WordStr;
begin
  SetLength(Result, SizeOf(Value));
  Move(Value, Result[1], SizeOf(Value));
end;
```

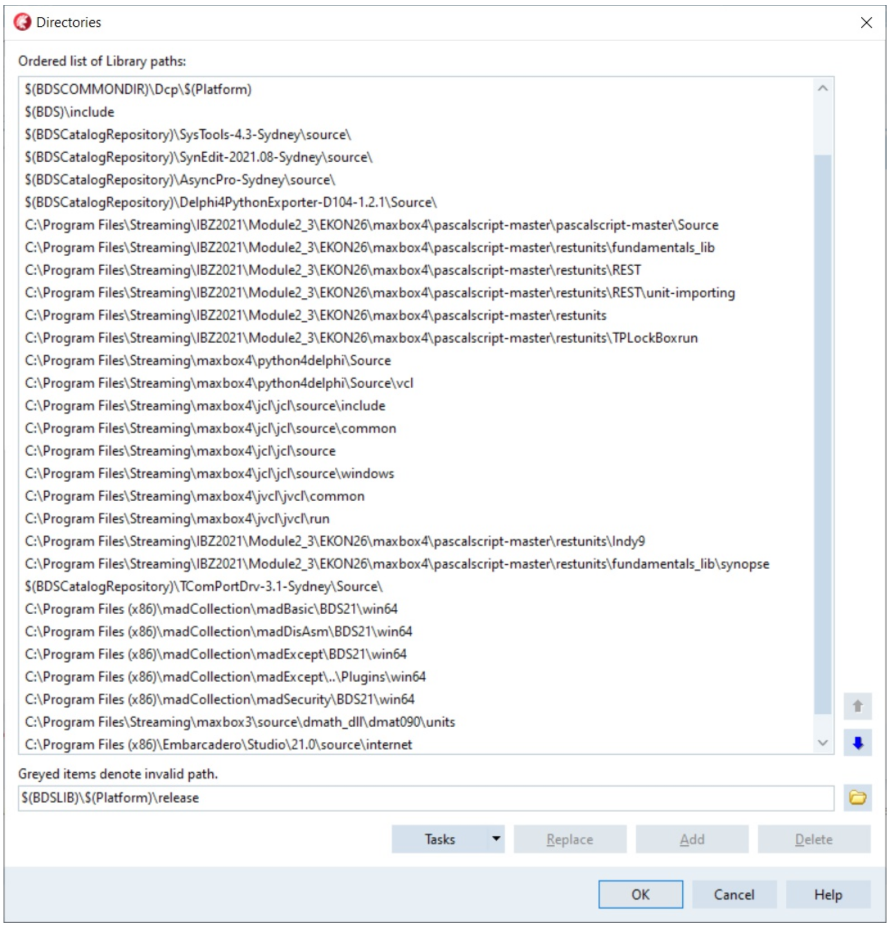
Or you can switch from assembler to Pascal code:

```
{$define GEOMETRY_NO_ASM}

procedure DivMod(dividend : Integer; divisor: Word; var result, remainder
: Word);
{$ifndef GEOMETRY_NO_ASM}
asm
   push  ebx
   mov   ebx, edx
   mov   edx, eax
   shr   edx, 16
   div   bx
   mov   ebx, remainder
   mov   [ecx], ax
   mov   [ebx], dx
   pop   ebx
{$else}
begin
   Result:=Dividend div Divisor;
   Remainder:=Dividend mod Divisor;
{$endif}
end;
```

The error "unit is compiled with a different version of..." is an annoying one. It occurs in a situation like below:

```
     +--------+
     | unit A |
     +--------+
      |      |
      |      |
      V      |
  +--------+ |
  | unit B | |
  +--------+ |
      |      |
      |      |
      V      V
     +--------+
     | unit C |
     +--------+
```

Both unit A and B use unit C and unit B uses C. Unit B and C are compiled and for some reason the source of unit B is not available. Now Unit C is changed (any change will do and is recompiled) And the dcu of unit C differs from the unit C used by unit B, so unit B needs to be recompiled too. But unfortunately, the source is not available so the compiler gives up.



```
$(BDSCOMMONDIR)\Dcp\$(Platform)
$(BDS)\include
$(BDSCatalogRepository)\SysTools-4.3-Sydney\source\
$(BDSCatalogRepository)\SynEdit-2021.08-Sydney\source\
$(BDSCatalogRepository)\AsyncPro-Sydney\source\
$(BDSCatalogRepository)\Delphi4PythonExporter-D104-1.2.1\Source\
C:\Program Files\Streaming\IBZ2021\Module2_3\EKON26\maxbox4\pascalscript-master\pascalscript-master\Source
C:\Program Files\Streaming\IBZ2021\Module2_3\EKON26\maxbox4\pascalscript-master\restunits\fundamentals_lib
C:\Program Files\Streaming\IBZ2021\Module2_3\EKON26\maxbox4\pascalscript-master\restunits\REST
C:\Program Files\Streaming\IBZ2021\Module2_3\EKON26\maxbox4\pascalscript-master\restunits\REST\unit-importing
C:\Program Files\Streaming\IBZ2021\Module2_3\EKON26\maxbox4\pascalscript-master\restunits
C:\Program Files\Streaming\IBZ2021\Module2_3\EKON26\maxbox4\pascalscript-master\restunits\TPLockBoxrun
C:\Program Files\Streaming\maxbox4\python4delphi\Source
C:\Program Files\Streaming\maxbox4\python4delphi\Source\vcl
C:\Program Files\Streaming\maxbox4\jcl\jcl\source\include
C:\Program Files\Streaming\maxbox4\jcl\jcl\source\common
C:\Program Files\Streaming\maxbox4\jcl\jcl\source
C:\Program Files\Streaming\maxbox4\jcl\jcl\source\windows
C:\Program Files\Streaming\maxbox4\jvcl\jvcl\common
C:\Program Files\Streaming\maxbox4\jvcl\jvcl\run
C:\Program Files\Streaming\IBZ2021\Module2_3\EKON26\maxbox4\pascalscript-master\restunits\Indy9
C:\Program Files\Streaming\IBZ2021\Module2_3\EKON26\maxbox4\pascalscript-master\restunits\fundamentals_lib\synopse
$(BDSCatalogRepository)\TComPortDrv-3.1-Sydney\Source\
C:\Program Files (x86)\madCollection\madBasic\BDS21\win64
C:\Program Files (x86)\madCollection\madDisAsm\BDS21\win64
C:\Program Files (x86)\madCollection\madExcept\BDS21\win64
C:\Program Files (x86)\madCollection\madExcept\..\Plugins\win64
C:\Program Files (x86)\madCollection\madSecurity\BDS21\win64
C:\Program Files\Streaming\maxbox3\source\dmath_dll\dmat090\units
C:\Program Files (x86)\Embarcadero\Studio\21.0\source\internet
```

Greyed items denote invalid path.

$(BDSLIB)\$(Platform)\release

Pic: 3_sourceorg_pathlibrarylist2023_2.png //pathlibrarylist2023.png

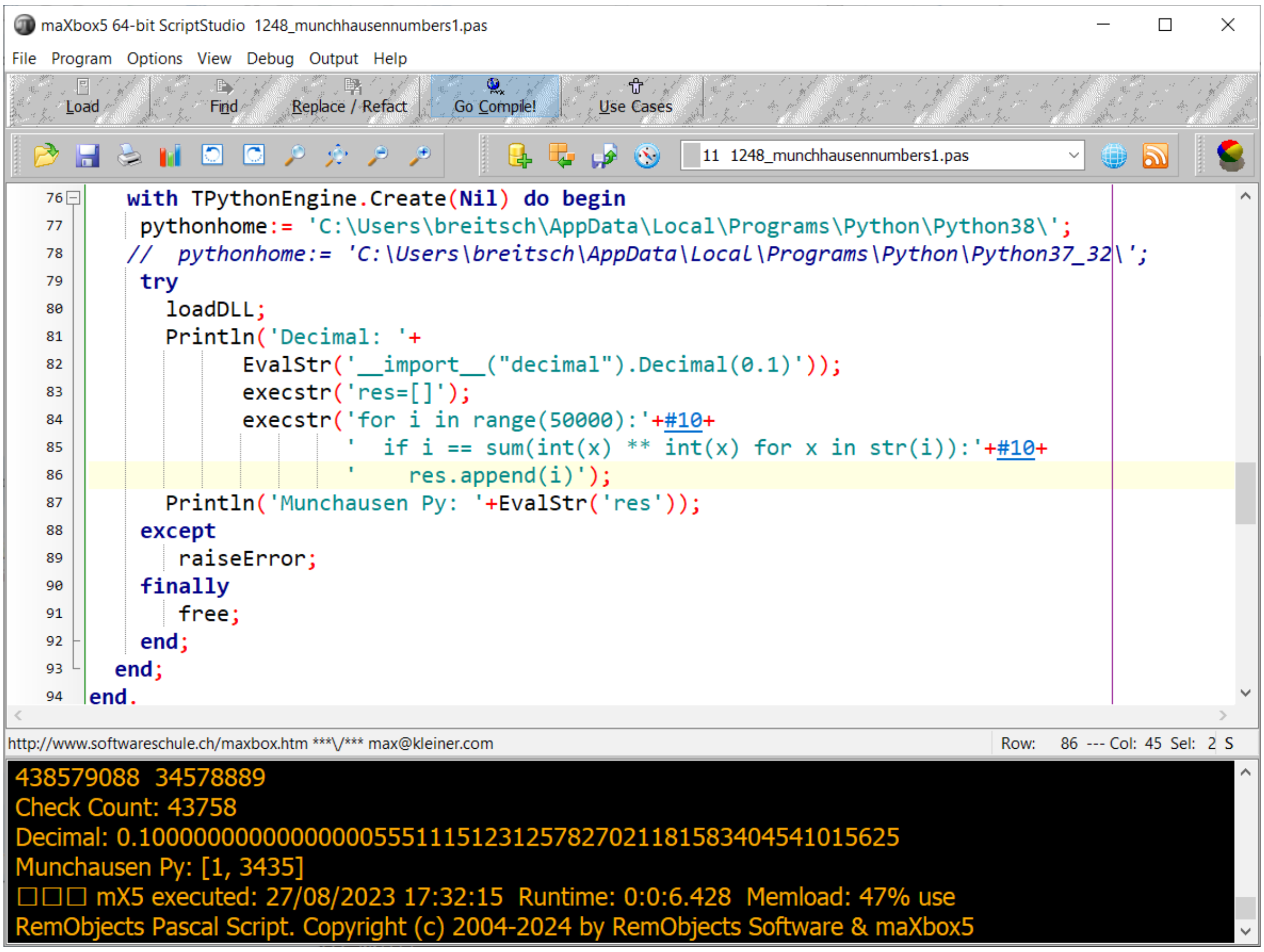
Progress has been made in that I've compiled the TestApplication sample with CrossVCL 1.27 for Mac64 and Linux64.

When I choose a second compile in a CrossVCL from the menu I receive the following error.

```
First chance exception at $0000000100419E9E. Exception class
EAccessViolation with message 'Access violation at address
0000000100419E9E, accessing address 00000009017241F8'. Process
TestApplication (5741)
Source Breakpoint at : C:\Program
Files\Streaming\IBZ2021\Module2_3\EKON26\maxbox4\pascalscript-
master\pascalscript-master\Source\uPSRuntime.pas line 2060. Process
TestApplication (5741)

Upsruntime.TPSExec.Clear()(0x00000002017350d0)
Upsdebugger.TPSCustomDebugExec.Clear()(0x00000002017350d0)
```

```
Upscomponent.TPSScript.Compile()(0x0000000201734c20)
Fmain.TForm1.Compile1Click(System.TObject*)(0x00007ffeefbfe038)
Vcl.Menus.TMenuItem.Click()(0x0000000201734960)
Vcl.Menus.TMenu.DispatchCommand(unsigned short)(0x0000000201734340,2)
Vcl.Forms.TCustomForm.WMCommand(Winapi.Messages.TWMCommand&)
(0x0000000205039ff0,0x00007ffeefbfe878)
:000000010001132B System::TObject::Dispatch(void*)
```

And then maybe by intuition I made a build and the AD has gone away and I got my first screen, compiled and script executed:


maXbox5 64-bit ScriptStudio 1248_munchhausennumbers1.pas
File Program Options View Debug Output Help
Load
Find
Replace / Refact
Go Compile!
Use Cases
11 1248_munchhausennumbers1.pas

```
with TPythonEngine.Create(Nil) do begin
  pythonhome:= 'C:\Users\breitsch\AppData\Local\Programs\Python\Python38\';
//  pythonhome:= 'C:\Users\breitsch\AppData\Local\Programs\Python\Python37_32\';
  try
    loadDLL;
    Println('Decimal: '+
          EvalStr('__import__("decimal").Decimal(0.1)'));
          execstr('res=[]');
          execstr('for i in range(50000):'+#10+
                  '  if i == sum(int(x) ** int(x) for x in str(i)):'+#10+
                  '    res.append(i)');
    Println('Munchausen Py: '+EvalStr('res'));
  except
    raiseError;
  finally
    free;
  end;
 end;
end.
```

http://www.softwareschule.ch/maxbox.htm ***\/*** max@kleiner.com   Row: 86 --- Col: 45 Sel: 2 S

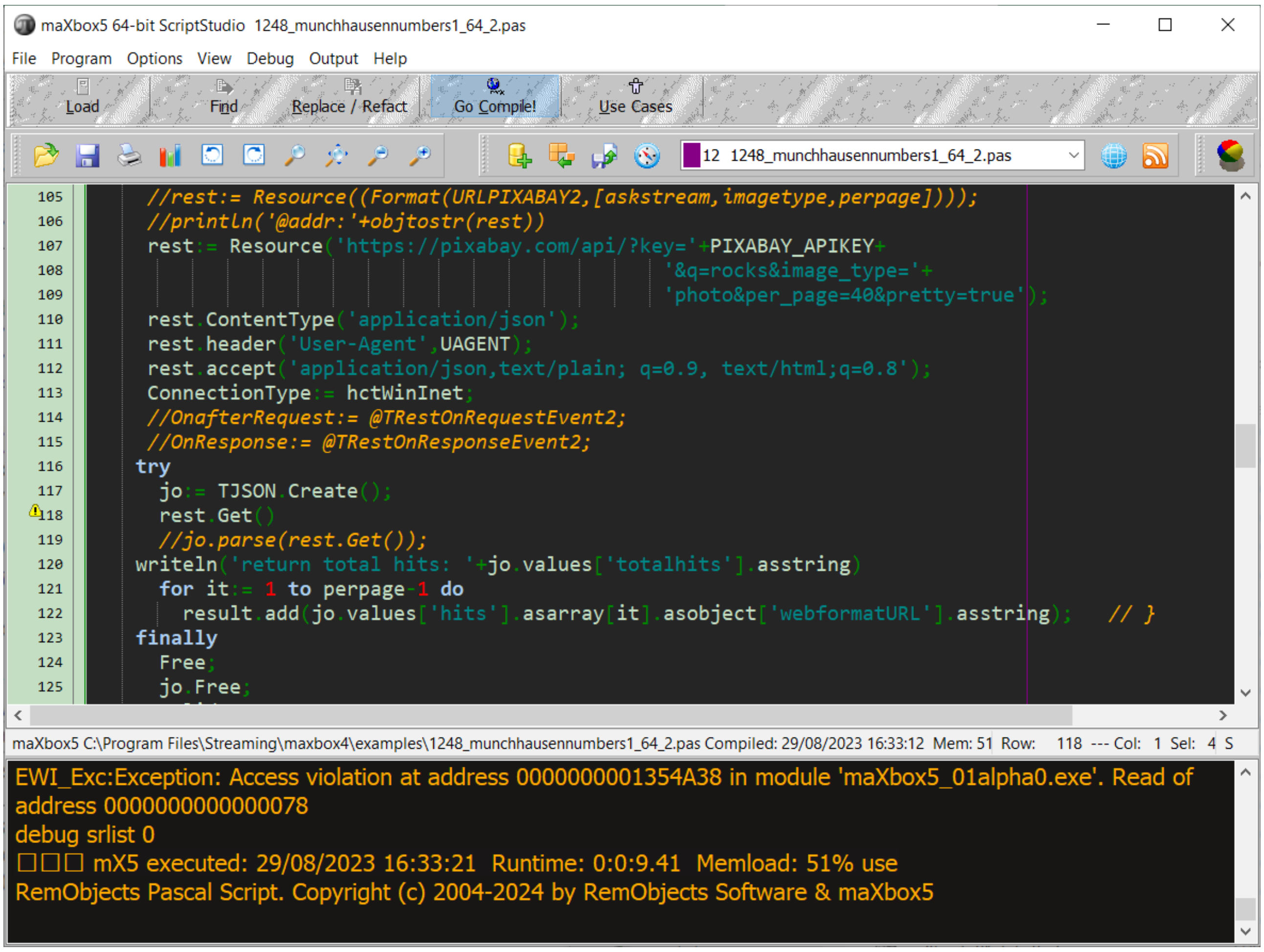
```
438579088  34578889
Check Count: 43758
Decimal: 0.1000000000000000055511151231257827021181583404541015625
Munchausen Py: [1, 3435]
□□□ mX5 executed: 27/08/2023 17:32:15  Runtime: 0:0:6.428  Memload: 47% use
RemObjects Pascal Script. Copyright (c) 2004-2024 by RemObjects Software & maXbox5
```

Pic5: 5_firstgui64bit.png

One of the unsolved problems is to catch an Access-violation instead of crash the app. Its like the code in uPSRuntime could not catch cause of halt or exit like the following:

```
procedure TdynamicDll.Quit;
begin
  if not( csDesigning in ComponentState ) then begin
{$IFDEF MSWINDOWS}
    MessageBox( GetActiveWindow, PChar(GetQuitMessage), 'Error',
                                              MB_TASKMODAL or MB_ICONSTOP );
    ExitProcess( 1 );
{$ELSE}
    WriteLn(ErrOutput, GetQuitMessage);
    Halt( 1 );
{$ENDIF}
  end;
end;
```

The weird thing was in one of the previous alpha versions (see below) the catch of the AV was present but in the meantime the app stuck and exits:


maXbox5 64-bit ScriptStudio 1248_munchhausennumbers1_64_2.pas
File Program Options View Debug Output Help
Load
Find
Replace / Refact
Go Compile!
Use Cases
12 1248_munchhausennumbers1_64_2.pas

```
      //rest:= Resource((Format(URLPIXABAY2,[askstream,imagetype,perpage])));
      //println('@addr:'+objtostr(rest))
      rest:= Resource('https://pixabay.com/api/?key='+PIXABAY_APIKEY+
                                                  '&q=rocks&image_type='+
                                                  'photo&per_page=40&pretty=true');
      rest.ContentType('application/json');
      rest.header('User-Agent',UAGENT);
      rest.accept('application/json,text/plain; q=0.9, text/html;q=0.8');
      ConnectionType:= hctWinInet;
      //OnafterRequest:= @TRestOnRequestEvent2;
      //OnResponse:= @TRestOnResponseEvent2;
     try
       jo:= TJSON.Create();
       rest.Get()
       //jo.parse(rest.Get());
     writeln('return total hits: '+jo.values['totalhits'].asstring)
       for it:= 1 to perpage-1 do
         result.add(jo.values['hits'].asarray[it].asobject['webformatURL'].asstring);   // }
     finally
       Free;
       jo.Free;
```

maXbox5 C:\Program Files\Streaming\maxbox4\examples\1248_munchhausennumbers1_64_2.pas Compiled: 29/08/2023 16:33:12 Mem: 51 Row: 118 --- Col: 1 Sel: 4 S

```
EWI_Exc:Exception: Access violation at address 0000000001354A38 in module 'maXbox5_01alpha0.exe'. Read of
address 0000000000000078
debug srlist 0
□□□ mX5 executed: 29/08/2023 16:33:21  Runtime: 0:0:9.41  Memload: 51% use
RemObjects Pascal Script. Copyright (c) 2004-2024 by RemObjects Software & maXbox5
```

pic6: 6_firstgui64bit2except.png

After debugging I realized it is a first chance exception which works as long the debugger is running with break or continue but without debugger the app disappears without forwarding the AV on the output like AV at address xyz read of address 000.
You can tell the debugger to ignore certain kinds of exceptions. Figure 3 shows Delphi's language-exception options. Add an exception class to the list, and all exceptions of that type and of any descendant types will pass through to your program without Delphi interfering. You can use Delphi's "advanced breakpoints" to disable exception handling around a region of code. To begin, set a breakpoint on the line of code where you want the IDE to ignore exceptions.

Now let's have a last look at a selected test app/script below with individual texts from your own data to translate. We wrote two useful functions. The first one returns text translated with a target language. The second one accepts one sentence as an argument with language detection as a param "auto". Then it will show text in JSON or as file.

Unit Unicode Testing

This app allows you to translate or detect text from many different languages and to test mX5 with Unicode. That's why I want this endpoint to be seamlessly integrated into googletrans, with it switching between endpoints if one is facing 4xx/5xx errors.

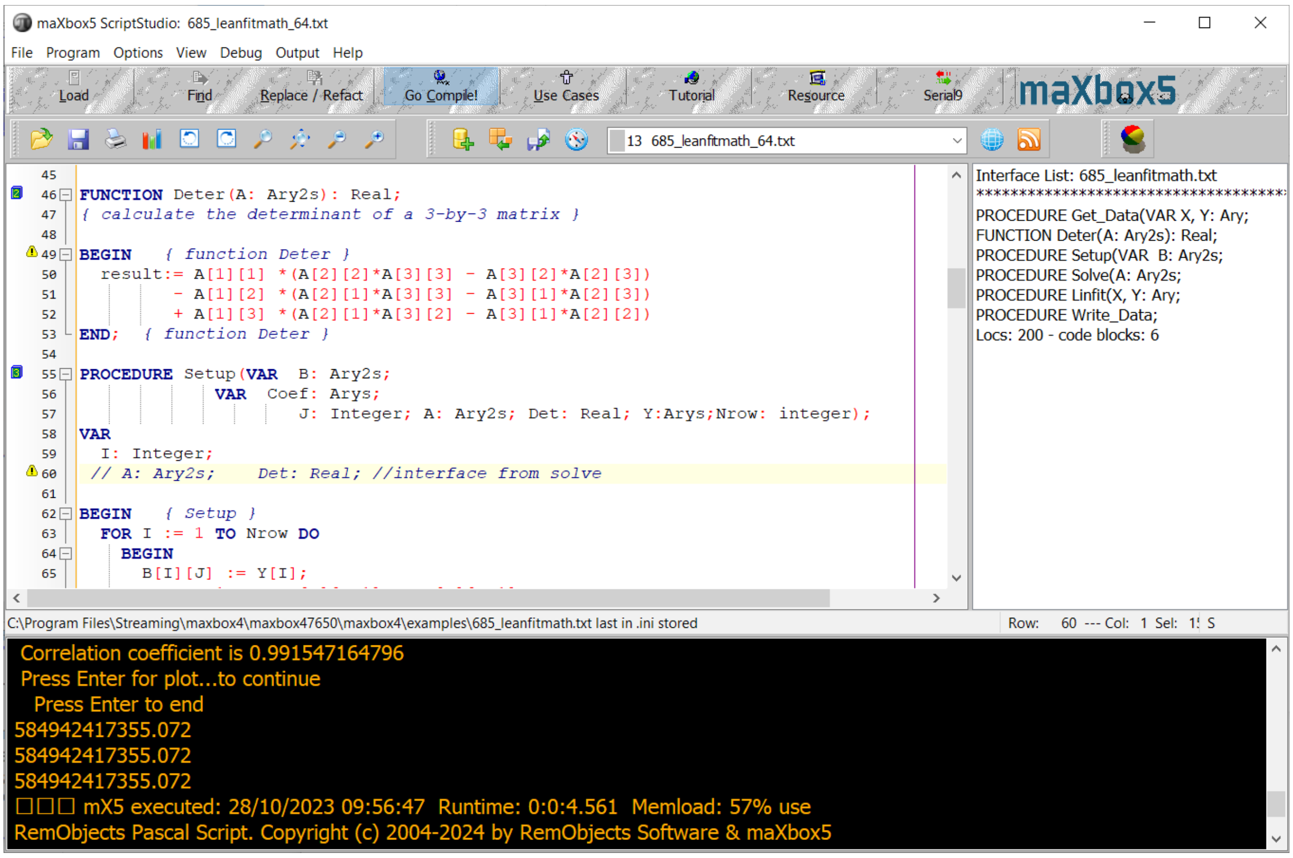
```
Const AURLS = 'https://clients5.google.com/translate_a/t?client=dict-
chrome-ex&sl=%s&tl=%s&q=%s';

function Text_to_traslate_API5(AURL, aclient,langorig,langtarget,atext:
                                                             string):string;
var httpq: THttpConnectionWinInet;
    rets: TStringStream;
    heads: TStrings; iht: IHttpConnection;
    jo: TJSON; jarr: TJsonArray2;
begin
  httpq:= THttpConnectionWinInet.Create(true);
  rets:= TStringStream.create('');
   try
     httpq.Get(Format(AURLS,[langorig,langtarget,atext]),rets);
     writeln('server: '+Httpq.GetResponseHeader('server'));

     jo:= TJSON.Create();
     jo.parse(rets.datastring)
     jarr:= jo.JsonArray;
     if httpq.getresponsecode=200 Then result:=jarr[0].stringify
       else result:='Failed:'+
             itoa(Httpq.getresponsecode)+Httpq.GetResponseHeader('message');
   except
      writeln('EWI_HTTP: '+ExceptiontoString(exceptiontype,exceptionparam));
   finally
     httpq.free;
     httpq:= Nil;
     rets.Free;
     jo.free;
   end;
end;
```

Google's service, offered free of charge, instantly translates words, phrases, text and web pages between English and over 100 other languages. That's how we call the function:

```
atext:= 'bonjour mes amis da la ville';
writeln(utf8ToAnsi(Text_to_traslate_API2(AURL,'dict-chrome-
                                                       ex','auto','es',atext)));
```

and the result: **server: ESF**
["Hola mis amigos en la ciudad","fr"]

Google Translate is now a form of augmented reality and is adapted for educational purposes. This application provides users with tools to translate between languages and they now include an image option; users take a photograph of a sign, piece of paper, or other form of written text and receive a translation in the language of their choice.


maXbox5 ScriptStudio: 685_leanfitmath_64.txt
File Program Options View Debug Output Help
Load
Find
Replace / Refact
Go Compile!
Use Cases
Tutorial
Resource
Serial9
maXbox5
13 685_leanfitmath_64.txt

```
FUNCTION Deter(A: Ary2s): Real;
{ calculate the determinant of a 3-by-3 matrix }

BEGIN    { function Deter }
  result:= A[1][1] *(A[2][2]*A[3][3] - A[3][2]*A[2][3])
         - A[1][2] *(A[2][1]*A[3][3] - A[3][1]*A[2][3])
         + A[1][3] *(A[2][1]*A[3][2] - A[3][1]*A[2][2])
END;  { function Deter }

PROCEDURE Setup(VAR  B: Ary2s;
             VAR  Coef: Arys;
                  J: Integer; A: Ary2s; Det: Real; Y:Arys;Nrow: integer);
VAR
  I: Integer;
 // A: Ary2s;    Det: Real; //interface from solve

BEGIN   { Setup }
  FOR I := 1 TO Nrow DO
    BEGIN
      B[I][J] := Y[I];
```

```
Interface List: 685_leanfitmath.txt
************************************
PROCEDURE Get_Data(VAR X, Y: Ary;
FUNCTION Deter(A: Ary2s): Real;
PROCEDURE Setup(VAR  B: Ary2s;
PROCEDURE Solve(A: Ary2s;
PROCEDURE Linfit(X, Y: Ary;
PROCEDURE Write_Data;
Locs: 200 - code blocks: 6
```

```
C:\Program Files\Streaming\maxbox4\maxbox47650\maxbox4\examples\685_leanfitmath.txt last in .ini stored    Row:  60 --- Col:  1 Sel:  1! S
```

```
 Correlation coefficient is 0.991547164796
 Press Enter for plot...to continue
   Press Enter to end
5849424173555.072
5849424173555.072
5849424173555.072
□□□ mX5 executed: 28/10/2023 09:56:47  Runtime: 0:0:4.561  Memload: 57% use
RemObjects Pascal Script. Copyright (c) 2004-2024 by RemObjects Software & maXbox5
```

Pic7: 7_mX5_64bitGUI.png

This visual technique above used to help with the understanding about what individual texts represent is called semantic analysis. About the topic: https://en.wikipedia.org/wiki/Semantic_analysis_(linguistics)

I found another endpoint to test with unicode within the source code of one of the google translate extensions on VSCode too.

```
"https://translate.googleapis.com/translate_a/single?client=gtx&dt=t + params"
// where the params are:
{
  "sl": source language,
  "tl": destination language,
  "q": the text to translate
}
```

The results looks something like this:

```
[[["こんにちは、今日はお元気ですか？","Hello, how are you today?",null,null,3,null,null,[[]
,[[["9588ca5d94759e1e85ee26c1b641b1e3","kgmt_en_ja_2020q3.md"]
]]
]]
,null,"en",null,null,null,null,[]
]
```

for the query: https://translate.googleapis.com/translate_a/single?client=gtx&dt=t&sl=en&tl=ja&q=Hello, how are you today?

And something like this:

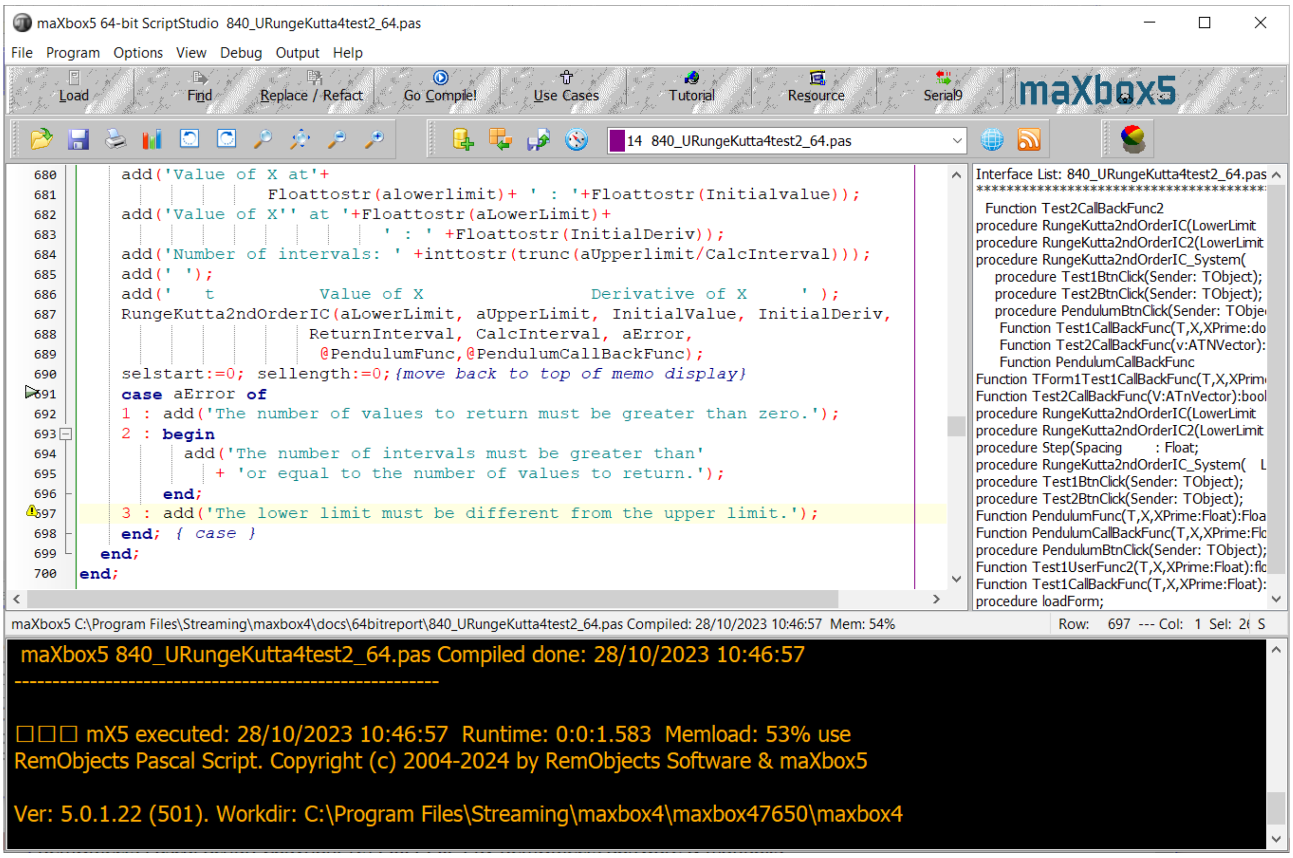
```
[[["Bonjour","Hello",null,null,1]]
,null,"en",null,null,null,null,[]
]
```

String unicode (\uxxx) encoding and decoding.

After some testing with request headers and F12 tools – Inspect (see below), I found the solution for the garbled text it can be. Simply set the User-Agent header to the one that Google Chrome uses.

Example:

```
import requests
word = 'لماذا تفعل هذا'
url = "https://clients5.google.com/translate_a/t?client=dict-chrome-
ex&sl=auto&tl=en&q=" + word
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36(KHTML, like Gecko) Chrome/88.0.4324.104 Safari/537.36'}

try:
    request_result = requests.get(url, headers=headers).json()
    print(request_result)
    print('[In English]: ' + request_result['alternative_translations'][0]
['alternative'][0]['word_postproc'])
    print('[Language Dectected]: ' + request_result['src'])
except:
     pass
```


maXbox5 64-bit ScriptStudio 840_URungeKutta4test2_64.pas

```
    add('Value of X at'+
                 Floattostr(alowerlimit)+ ' : '+Floattostr(Initialvalue));
    add('Value of X'' at '+Floattostr(aLowerLimit)+
                           ' : ' +Floattostr(InitialDeriv));
    add('Number of intervals: ' +inttostr(trunc(aUpperlimit/CalcInterval)));
    add(' ');
    add('   t           Value of X                 Derivative of X     ' );
    RungeKutta2ndOrderIC(aLowerLimit, aUpperLimit, InitialValue, InitialDeriv,
                     ReturnInterval, CalcInterval, aError,
                      @PendulumFunc,@PendulumCallBackFunc);
    selstart:=0; sellength:=0;{move back to top of memo display}
    case aError of
    1 : add('The number of values to return must be greater than zero.');
    2 : begin
          add('The number of intervals must be greater than'
            + 'or equal to the number of values to return.');
        end;
    3 : add('The lower limit must be different from the upper limit.');
    end; { case }
  end;
end;
```

```
Interface List: 840_URungeKutta4test2_64.pas
**************************************
  Function Test2CallBackFunc2
procedure RungeKutta2ndOrderIC(LowerLimit
procedure RungeKutta2ndOrderIC2(LowerLimit
procedure RungeKutta2ndOrderIC_System(
    procedure Test1BtnClick(Sender: TObject);
    procedure Test2BtnClick(Sender: TObject);
    procedure PendulumBtnClick(Sender: TObje
     Function Test1CallBackFunc(T,X,XPrime:do
     Function Test2CallBackFunc(v:ATNVector):
     Function PendulumCallBackFunc
Function TForm1Test1CallBackFunc(T,X,XPrim
Function Test2CallBackFunc(V:ATnVector):bool
procedure RungeKutta2ndOrderIC(LowerLimit
procedure RungeKutta2ndOrderIC2(LowerLimit
procedure Step(Spacing       : Float;
procedure RungeKutta2ndOrderIC_System(   L
procedure Test1BtnClick(Sender: TObject);
procedure Test2BtnClick(Sender: TObject);
Function PendulumFunc(T,X,XPrime:Float):Floa
Function PendulumCallBackFunc(T,X,XPrime:Flo
procedure PendulumBtnClick(Sender: TObject);
Function Test1UserFunc2(T,X,XPrime:Float):flo
Function Test1CallBackFunc(T,X,XPrime:Float):
procedure loadForm;
```

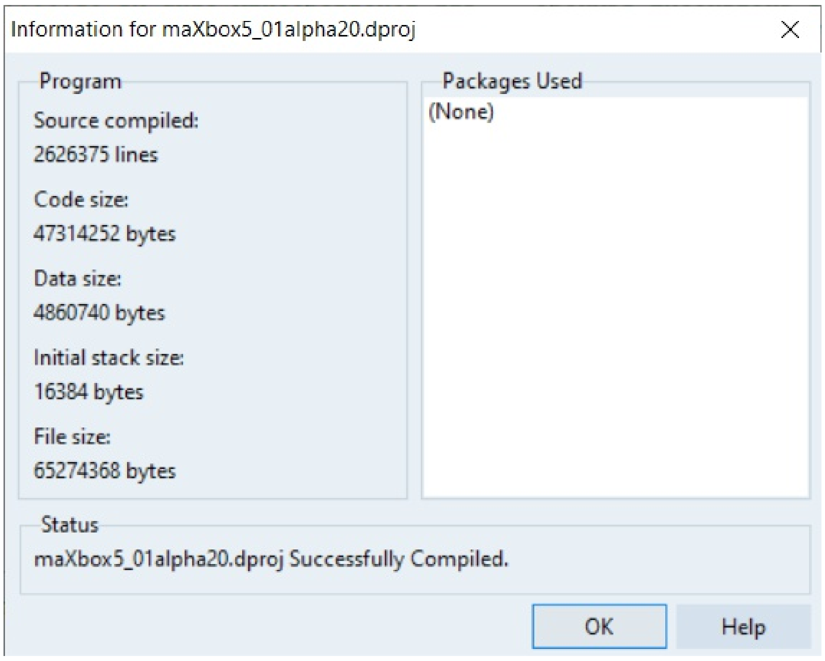
```
maXbox5 C:\Program Files\Streaming\maxbox4\docs\64bitreport\840_URungeKutta4test2_64.pas Compiled: 28/10/2023 10:46:57  Mem: 54%     Row:   697 --- Col:  1  Sel: 2( S
```

```
maXbox5 840_URungeKutta4test2_64.pas Compiled done: 28/10/2023 10:46:57
--------------------------------------------------------

□□□ mX5 executed: 28/10/2023 10:46:57  Runtime: 0:0:1.583  Memload: 53% use
RemObjects Pascal Script. Copyright (c) 2004-2024 by RemObjects Software & maXbox5

Ver: 5.0.1.22 (501). Workdir: C:\Program Files\Streaming\maxbox4\maxbox47650\maxbox4
```

Pic8: 8_mX5_64bitGUI2.png

Conclusion:

Of course the 64-bit-box is not finished yet. The current version 5.0.1.22 is an early beta Version. As a next step method pointers (func pointers) and TEncoding are on the list. Also on discovering a function marked with the overload directive, it prompts for a new function name, and then generates wrapper code that maps the new method name to the original version, but in a redirection we get an AV.
In general, there is unlikely to be much benefit beyond an 32-bit, possibly, a small speed increase, more memory and more registers. Don't rely on 64-bit to speed up a slow application though: you will still need to make algorithmic changes for large speed boosts.
64-bit isn't a magic bullet. Other than that, you only need to change if you've already encountered one of the limits imposed by 32-bit or you want to develop plugins for 64-bit app or just be compatible with a 64-bit operation system.


Information for maXbox5_01alpha20.dproj

Program
Source compiled:
2626375 lines
Code size:
47314252 bytes
Data size:
4860740 bytes
Initial stack size:
16384 bytes
File size:
65274368 bytes

Packages Used
(None)

Status
maXbox5_01alpha20.dproj Successfully Compiled.

Pic9: 9_buildinfo21_informX5.png

References:

Compiled Project:
https://github.com/maxkleiner/maXbox4/releases/download/V4.2.4.80/maxbox5.zip

Preparation:
https://stackoverflow.com/questions/4051603/how-should-i-prepare-my-32-bit-delphi-programs-for-an-eventual-64-bit-compiler

Doc and Tool: https://maxbox4.wordpress.com

Max Kleiner 28/10/2023

maXbox4exe Digital unterschrieben von maXbox4exe Datum: 2023.10.28 18:23:25 +02'00'