//

Code Signing

maXbox Starter 118 – Get a code signing certificate.

"Time behaves like space – timeless.

Source: 1033_signtool_simplebatch_solution_mX4Cert_second64.txt

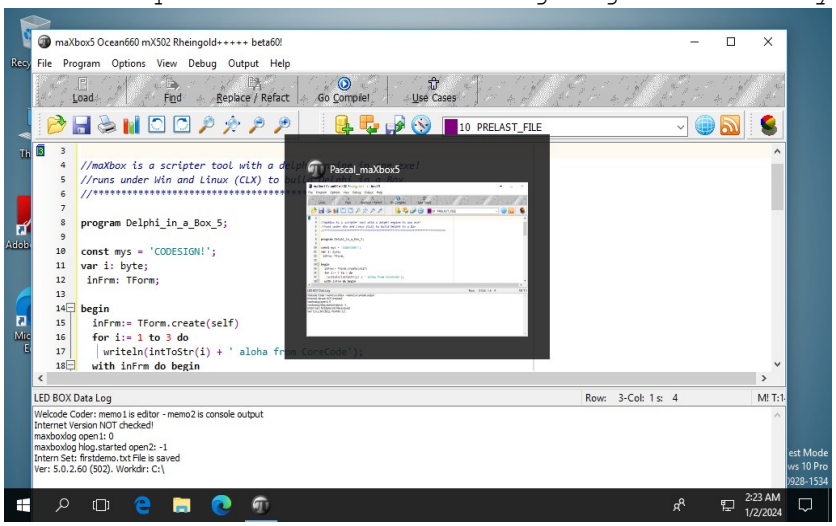
If you have a code base of a 64-bit Windows executable you can sign this Exe. Code signing is the process of digitally signing executables and scripts to confirm the software author and guarantee that the code has not been altered or corrupted since it was signed for the first time.

```
{-------------------------------------------------------------------------}
{-                                                                        -}
{-  Tool> signtool sign /f certs/tcertificate4.p12 /p belplan /t         -}
{-                        http://timestamp.digicert.com m85covid6.png     -}
{-Done Adding Additional Store                                            -}
SignTool Error: This file format cannot be signed because it is not
recognized. SignTool Error: An error occurred while attempting to sign:
{-                                  m85covid6.png                         -}
{-------------------------------------------------------------------------}
```

As you can see, you can't sign a png or jpg file, just exe, jar, js and bat and a few others. The ***.p12** contains both the private and the public key, and also information about the owner (name, email, location, address, etc.) all being certified by a third party. With such certificate, a user can identify himself and authenticate himself to any organization trusting the third party.
You should be able to see the content of the p12 file with

openssl pkcs12 -info -in filename.p12

provided openssl is installed in your system.
The executable is organised in a project-, a manifest- and a resource unit. This article will be the default article after the implementation of the new Minimum Requirements for Code Signing on February 1, 2017.



pic1: tutor118_signobjectscreen_6.png

You can either sign files out of a working directory, or you can place them in your Windows SDK\bin folder.

Source Organisation Steps

1. Open the Command Prompt: Windows 7: Start > Run > cmd, or for Windows 8-10, press the Windows Key, then type cmd and press enter.

2. Navigate to the directory with signtool.exe.

3. Use the following command to sign your file:
4. signtool sign /a /tr http://timestamp.globalsign.com/tsa/r6advanced1 /td SHA256 /fd SHA256 c:/path/to/your/file.exe

5. To verify the successful signature use the following commands: Authenticode: signtool verify /v /pa

Alternatively, you can batch that in a script:

```
Const TOOLPATH =
   'C:\maxbox\maxbox51\examples\signtool\';
   //'from C:\Program Files (x86)\Windows Kits\10\bin\10.0.19041.0\x64\
signtool.exe';
const CERTFILE =
             'C:\maxbox\maxbox51\examples\signtool\certs\maxbox4exe.pfx';

writeln(GETDOSOutput('signtool.exe sign /f'
            +' certs/maxbox4exe.pfx /p '+PASSTOKENfromfile
          //+' /t http://timestamp.digicert.com '+TOSIGNFILE
            +' /tr http://timestamp.globalsign.com/tsa/r6advanced1 /td
                                          SHA256 /fd SHA256 '
                                       +TOSIGNFILE  ,TOOLPATH));//}
```

Enter your Token Password. If the signing is successful you will see a prompt informing you so for the signing process.
→ Done Adding Additional Store

- **/tr**– Specify an RFC 3161 compliant trusted time stamp server.*Recommended*
- **/td SHA256**– Must be called after "/tr", this command specifies the TimeStamp digest Algorithm. *Recommended*
- **/ac**– Specify an Additional Certificate.

Note: Timestamping your Code is extremely important and is highly recommended for every piece of code that you sign.
This timestamp will allow the file that you sign to remain valid long after the certificate itself has expired.

So for the verify of the sign get the latest revision of signtool with patches as from issues and it goes like this:

```
WinExec32('cmd /C signtool.exe verify /v /pa '+TOSIGNFILE+' >
                                              signresult3445.txt',1);
```

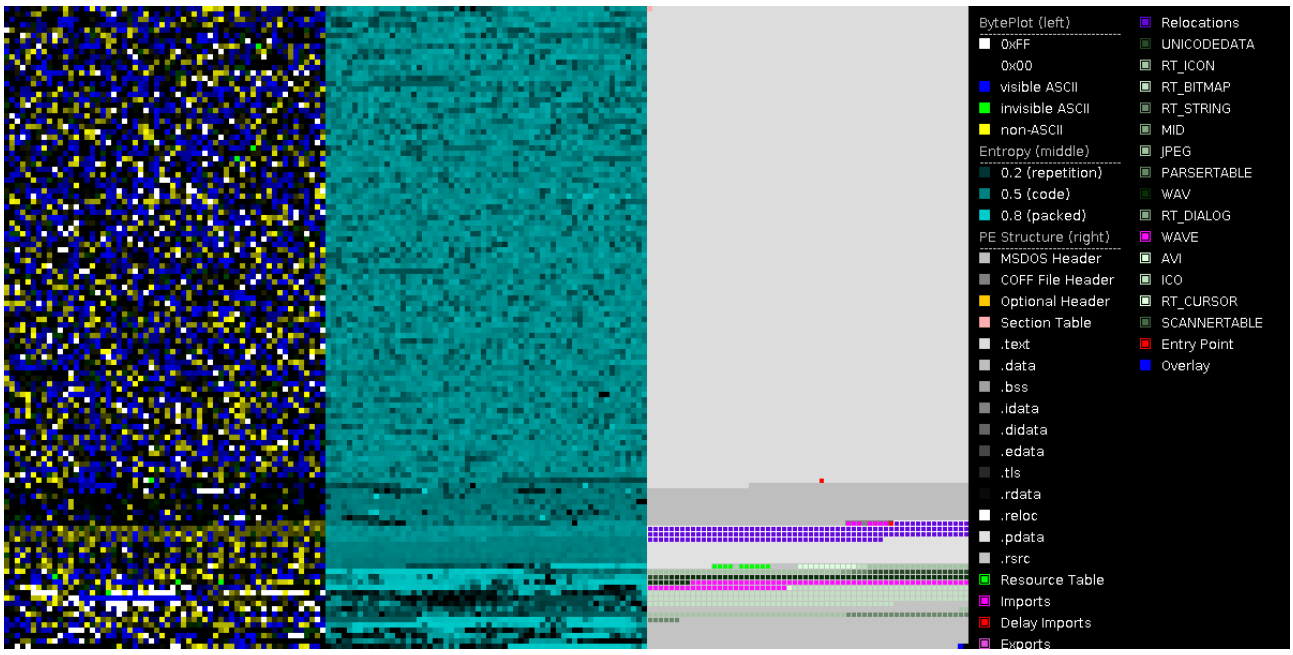
Verifying: PointInSpace5_64.exe

```
Signature Index: 0 (Primary Signature)
Hash of file (sha256):

139E1FBBEB8BA664CA08EB6B3B8CEFD1E59AC87A7A97B6DACE406FF489828BBB

Signing Certificate Chain:
    Issued to: maXboxCertAuth
    Issued by: maXboxCertAuth
    Expires:   Sun Jan 01 00:59:59 2040
    SHA1 hash: 6F83207B500DCC0E32A719599CBC6BD7E6B2A04D

        Issued to: maXbox4signer
        Issued by: maXboxCertAuth
        Expires:   Sun Jan 01 00:59:59 2040
        SHA1 hash: 6A89501B76D47C189A60BF1070BAA2FBFD38D7D7

            Issued to: maXbox4exe
            Issued by: maXbox4signer
            Expires:   Sun Jan 01 00:59:59 2040
            SHA1 hash: F0EB0CA218C5707FAC78921F81092CECA12AD0E9

The signature is timestamped: Tue Jan 02 19:31:40 2024
Timestamp Verified by:
    Issued to: GlobalSign
    Issued by: GlobalSign
    Expires:   Sun Dec 10 01:00:00 2034
    SHA1 hash: 8094640EB5A7A1CA119C1FDDD59F810263A7FBD1

        Issued to: GlobalSign Timestamping CA - SHA384 - G4
        Issued by: GlobalSign
        Expires:   Sun Dec 10 01:00:00 2034
        SHA1 hash: F585500925786F88E721D235240A2452AE3D23F9

            Issued to: Globalsign TSA for CodeSign1 - R6
            Issued by: GlobalSign Timestamping CA - SHA384 - G4
            Expires:   Sun May 08 08:45:38 2033
            SHA1 hash: CA3E8CFD7CFD329A99359A9A38F86185F0B01C4A


Successfully verified: PointInSpace5_64.exe

Number of files successfully Verified: 1
Number of warnings: 0
Number of errors: 0
```

The interesting point is to know where the certificate with the hash is stored in the executable itself. Validate a PE certificate. Is the signature valid or not. It should work when signature is embedded in PE executable and when the signature is in a security catalog.
For example the exe maXbox5.exe with the hash:

```
SHA-1: ddf3fa4e3ccb0835082c8d8bbd9ddd98a5b5c7b5
SHA-256: da34199785ae5371e2cf8a23a12b68295f7c968ba0c8a24f367baf0c5f091439
```

The embedded cert can be found at the end of an executable as a PE layer, we can visualize such a structure of a PE executable, look at the blue section called overlay at the very end:



Pic2: tutor118_maxbox5_visualized_samplesections.png

In Delphi or maXbox, I can include a folder's source code by adding it to the project Search Path or define as an include file, or adding it to the Library Path. The Search Path applies for the *UMatrix.pas* only to the current project, while the Library Path applies to any project opened with the IDE.

From DetectItEasy
PE64 Linker: Turbo Linker (8.0) [GUI64,signed] Compiler: Embarcadero Delphi (11.0 Alexandria) [Standard] Sign tool: Windows Authenticode (2.0) [PKCS #7]
File size 59.92 MB (62832288 bytes)

X509 Certificates

maXbox4signer

maXbox4exe

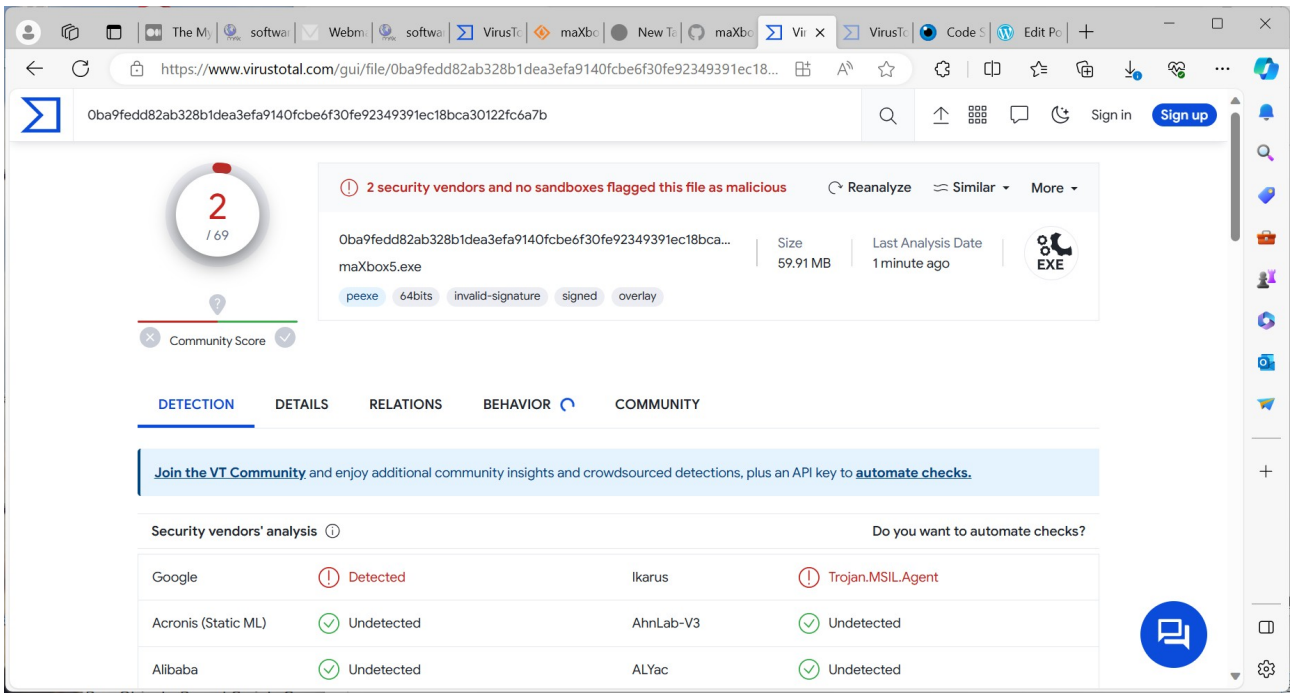
Globalsign TSA for CodeSign1 - R6

GlobalSign Timestamping CA - SHA384 - G4

GlobalSign

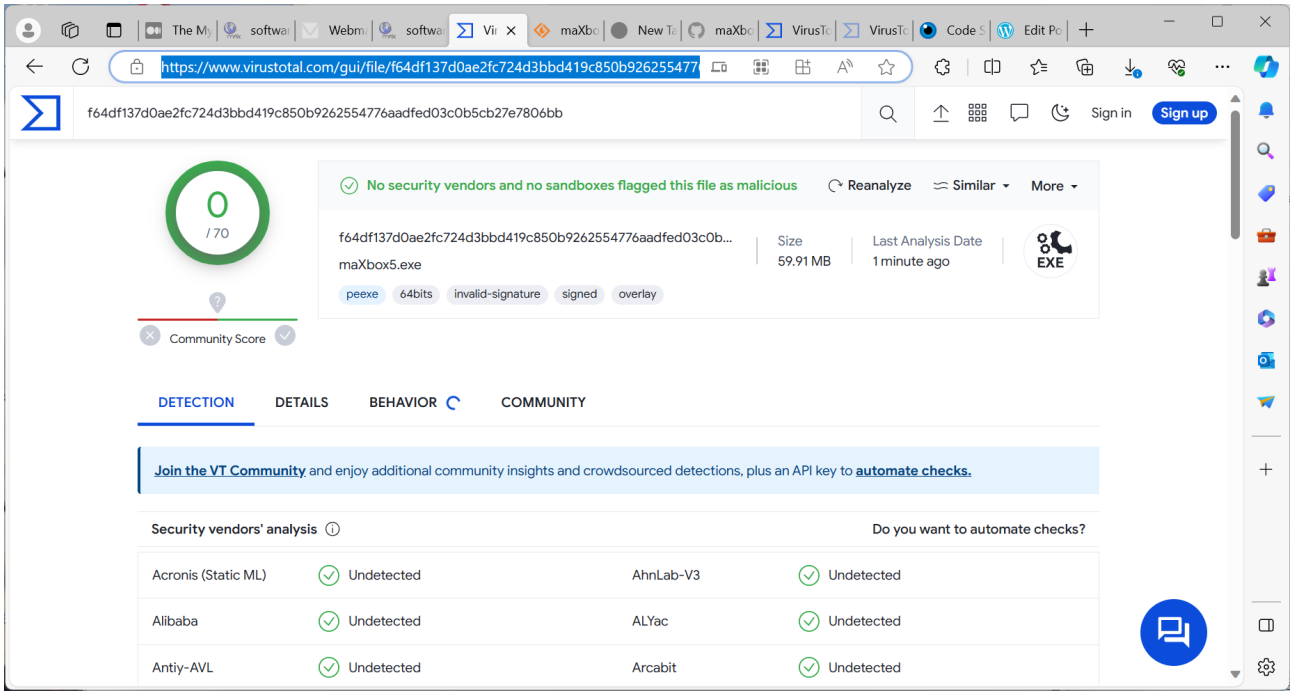
The Mystery of VirusTotal

Also please specify which certificate kind is the correct one. Most sites only mention "code signing" and talk about signing applications that are actually compiled by the user. This is actually the case for me. I got a last compile of this year 2023 for the multi-installer of Python4Delphi and as usual signed this with my code signing certificate as usual but VirusTotal showed 2 detections:



So what to do, checked the MSIL.Agent read something about misused code signing so I decided to test it without code signing and no flag this time, so it has to do with the signing process! (which is an extension of the underlaying Executable as overlay). I **replaced** the authenticode time stamp countersignature from digicert to globalsign an it worked with VirusTotal!:

```
 +' /tr http://timestamp.globalsign.com/tsa/r6advanced1 /td SHA256 /fd SHA256 '
           +TOSIGNFILE  ,TOOLPATH));  //}
```

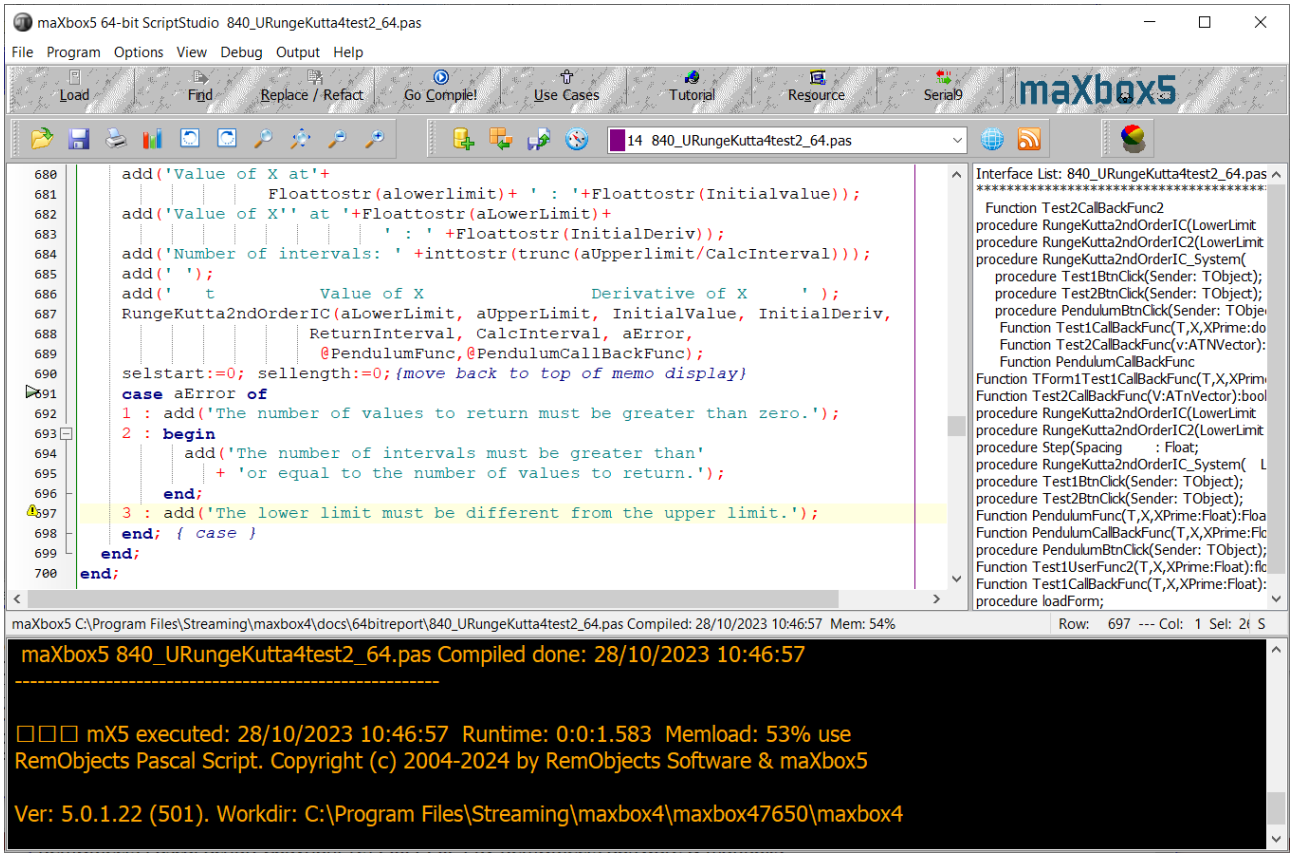


Pic3&4: tutor118_VTotal_secondtest.png

When we compare the sign part of VirusTotal in Details we can see that SHA-256 orders additionally use the R1-R3 Cross Certificate - default March 31, 2014 & after. (The R1-R3 Cross Certificate will need to be installed on the signing computer but not specified as an additional certificate during the signing procedure):

The old timestamp one X509 Certificates from DigiCert and the new one from GlobalSign:

VirusTotal - File - da34199785ae5371e2cf8a23a12b68295f7c968ba0c8a24f367baf0c5f091439



Pic5: 8_mX5_64bitGUI2.png

Conclusion:

To sign a Windows executable file, you will need a code signing certificate from a Certificate Authority (CA) like Verisign or a self signed certificate with OpenSSL or instantssl.com1. Once you have the certificate, you can use Microsoft's SignTool to sign your app. You can download SignTool as part of the Windows SDK1.
You download it as part of the Windows SDK. Note that it's also possible to install SignTool without installing the entire SDK.
How to install SignTool.exe for Windows 10 - Stack Overflow

Once installed you can use SignTool from the command line or a script.

Script: softwareschule.ch/examples/maxbox_starter115.txt

References:

Compiled Project:

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

Free Automated Malware Analysis Service - powered by Falcon Sandbox (hybrid-analysis.com)

Topic:

certificate - Signing a Windows EXE file - Stack Overflow

Preparation:

How to install SignTool.exe for Windows 10 - Stack Overflow

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

Appendix

Unit UMatrix

```
Procedure Determinant( Dimen : integer; Data : TNmatrix; var Det : Float; var Error : byte)');
Procedure Inverse2( Dimen : integer; Data : TNmatrix; var Inv : TNmatrix; var Error : byte)');
 Procedure Gaussian_Elimination(Dimen:integer;Coefficients:TNmatrix;Constants:TNvector;var Solution:TNvector;var
Error:byte);
 Procedure Partial_Pivoting(Dimen:integer; Coefficients:TNmatrix;Constants:TNvector; var Solution:TNvector;var
Error:byte);
 Procedure LU_Decompose(Dimen:integer;Coefficients:TNmatrix;var Decomp:TNmatrix;var Permute:TNmatrix;var
Error:byte);
 Procedure LU_Solve(Dimen: integer;var Decomp TNmatrix;Constants:TNvector;var Permute:TNmatrix;var
Solution:TNvector;var Error:byte);
 Procedure Gauss_Seidel( Dimen : integer; Coefficients : TNmatrix; Constants : TNvector; Tol : Float; MaxIter :
integer; var Solution : TNvector; var Iter : integer; var Error : byte)'); end;

 CL.AddTypeS('TNvector', 'array[1..30] of Extended');
 //TNvector = array[1..TNArraySize] of Float;
 CL.AddTypeS('TNmatrix', 'array[1..30] of TNvector');
 //TNmatrix = array[1..TNArraySize] of TNvector;

 CL.AddConstantN('TNNearlyZero','Extended').setExtended( 1E-07);
 CL.AddConstantN('TNArraySize','LongInt').SetInt( 30);
 CL.AddDelphiFunction('Procedure Gaussian_Elimination( Dimen : integer; Coefficients : TNmatrix; Constants :
TNvector; var Solution : TNvector; var Error : byte)');
 CL.AddDelphiFunction('Procedure Partial_Pivoting( Dimen : integer; Coefficients : TNmatrix; Constants : TNvector;
var Solution : TNvector; var Error : byte)');
 CL.AddDelphiFunction('Procedure LU_Decompose( Dimen : integer; Coefficients : TNmatrix; var Decomp : TNmatrix;
var Permute : TNmatrix; var Error : byte)');
 CL.AddDelphiFunction('Procedure LU_Solve2( Dimen : integer; var Decomp : TNmatrix; Constants : TNvector; var
Permute : TNmatrix; var Solution : TNvector; var Error : byte)');
 CL.AddDelphiFunction('Procedure Gauss_Seidel( Dimen : integer; Coefficients : TNmatrix; Constants : TNvector; Tol
: Float; MaxIter : integer; var Solution : TNvector; var Iter : integer; var Error : byte)');
end;

Solution 1: 5.00 5.00 5.00
  Distance to sphere 1 is 6.928 (radius 6.928)
  Distance to sphere 2 is 6.403 (radius 6.403)
  Distance to sphere 3 is 6.403 (radius 6.403)
Solution 2: 5.00 -3.00 5.00
  Distance to sphere 1 is 6.928 (radius 6.928)
  Distance to sphere 2 is 6.403 (radius 6.403)
  Distance to sphere 3 is 6.403 (radius 6.403)

Sum of coordinate differences:
  Solution 1: 1.75172, Solution #2: 24.05681
Using Solution 1 set
Solution 1: 5.00 5.00 5.00
  Distance to sphere 1 is 6.92800 (vs. measured 6.92800)
  Distance to sphere 2 is 6.40300 (vs. measured 6.40300)
  Distance to sphere 3 is 6.40300 (vs. measured 6.40300)
  Distance to sphere 4 is 3.46390 (vs. measured 3.46400)
//}
```

Max Kleiner 02/01/2024

maXbox4exe Digitally signed by maXbox4exe
Date: 2024.01.02 21:34:59 +01'00'