OpenMI Association Technical Committee meeting no 40

See also: OATC Wiki Home

Date: 6. Januari 2011
Time: 13:00 - 14:00 CET
Venue: Skype Conference Call
Topic: From OpenMI 2.0 Beta to OpenMI 2.0

Table of contents

Participants

Adrian Harper, MWH Soft Ltd (adrian.harper@mwhsoft.com)
Stef Hummel, Deltares (stef.hummel@deltares.nl)
Standa Vanecek, DHI (s.vanecek@dhi.cz)
Peter Schade, Bundesanstalt fuer Wasserbau (peter.schade(at)baw.de)
Rob Knapen, Alterra, Wageningen UR (Rob.Knapen@wur.nl)

Apologies:
Jesper Grooss, DHI (jgr@dhigroup.com)
Unknown User (don), Deltares (gennadii.donchyts@deltares.nl)

1. OpenMI 1.4 64 bit

Problems with InfoWorks (OpenMI 1.4; Windows 7; 64bit)

Request from Johan van Assel (Aquafin):
"As far as I understand it, the problem lies in InfoWorks and not so much in Pipistrelle or in OpenMI 1.4, but in general I think it would be interesting if this information would be available to people. For the OpenMI Standard and for the standard (OATC/Pipistrelle) SDK and GUI, I suppose it could easily be mentioned. But for the individual softwares, might it be possible to have it somewhere in the compliance file ?"

-> A new Wiki-Page: "How to run OpenMI on 64 bit Windows systems"

Adrian: In 2.0 there will be a 32/64 bit check of the first level dll. Dlls of a lower level can not be checked.

It is recommended to compile separate 32bit and 64bit dlls, which is also planned for the GUI.

For developers with VS .NET Express: They should compile from command line.

Platform information is already part of the XML compliance file.

Additional note by Stef and a Deltares collegue (Hans van Putten):
After thorough testing it turned out that instead of rebuilding all OpenMI related dll's specifically for x86 systems it is enough to only rebuild the main application (i.e.Oatc.OpenMI.Gui.ConfigurationEditor.exe) specifically for the x86 platform. The executable will then treat the dll's which have been built with the "Any CPU" option as 32-bit dll's.

2. Signing of the standard assemblies

The standard is the only C# DLL and Java JAR that should be signed.

⚠ Rob will check out how to use public / private keys for the Java jar files.

⚠ Jesper will remove the keys from SourceForge.

There was an initial attempt to retrieve the private key used for signing the C# DLL but this did not appear to be possible (at least not in a trivial way). Since signing the DLL with a new key would mean having a new version of the DLL it was decided to sign the Java jar with a different key. Both key pairs will be kept by the chairman of the OATC.

Signing of the Java JAR file is possible using the keytool and jarsigner utilities from the JDK. A keystore (password-protected database) has to be created that holds the generated private key(s and certificates). Info from the keystore can than be used to sign the JAR and to export the public key that has to be published. For verification a user has to import the public key with the keytool and can then use jarsigner to verify the JAR. A certificate from a certification authority can be used for improved security.

Since the used .Net Strong Name Key was and is in a (public readable) SVN I think it should be considered "compromised" and not really suitable for authentication anymore. We probably have to discuss the whole signing issue and its purpose again at the next meeting before taking further steps.

3. Next Skype meetings

- Topic IValueSet: next Thursday, 13.1. at 9:00 CET (8:00 UTC)
- Regular meeting: Thursday, 20.1. at 9:00 CET (8:00 UTC)
- Open SDK issues: Thursday, 27.1. at 10:00 CET (9:00 UTC)