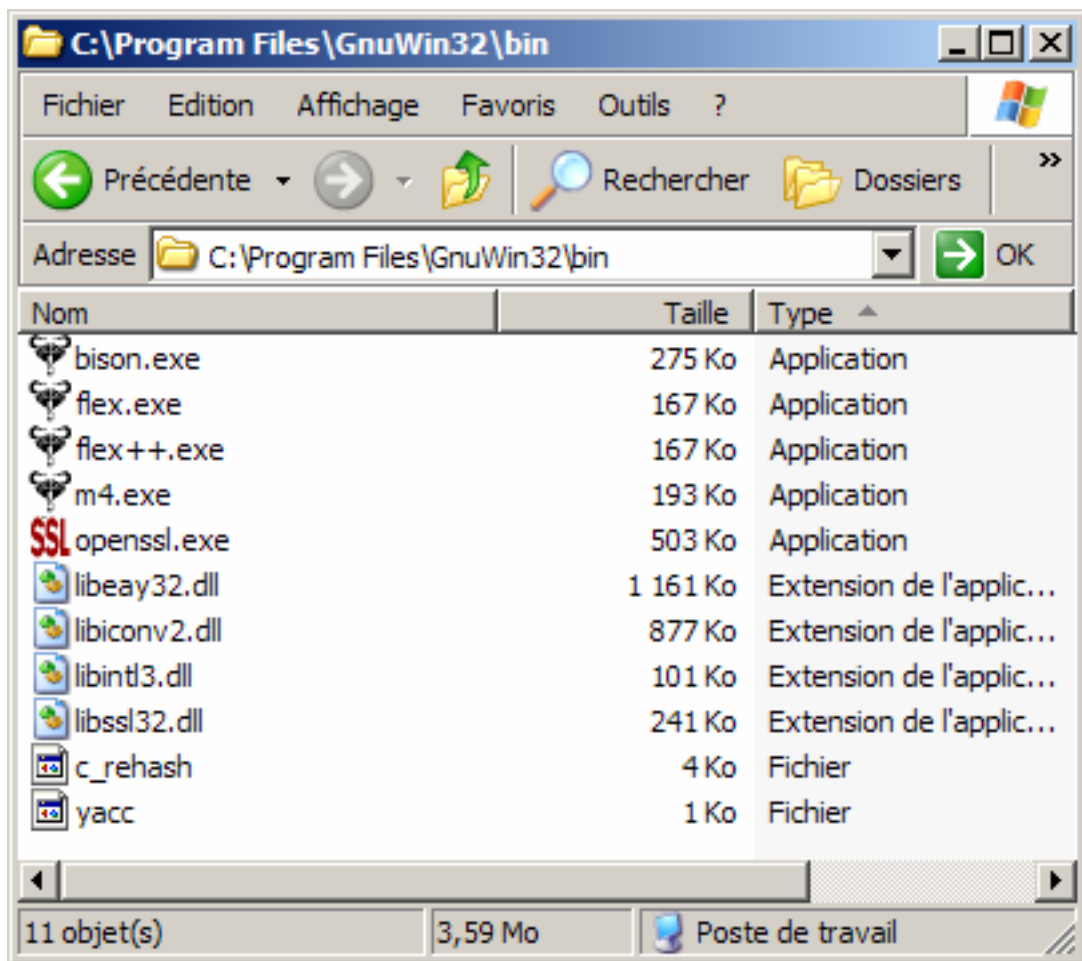


Note for development of CERTI (CERTI Savannah Registered Developer)

Guide for the implementation of CERTI from *Savannah*

1/ Tools for the development of CERTI

- CMake for Windows
<http://www.cmake.org/HTML/Download.html>



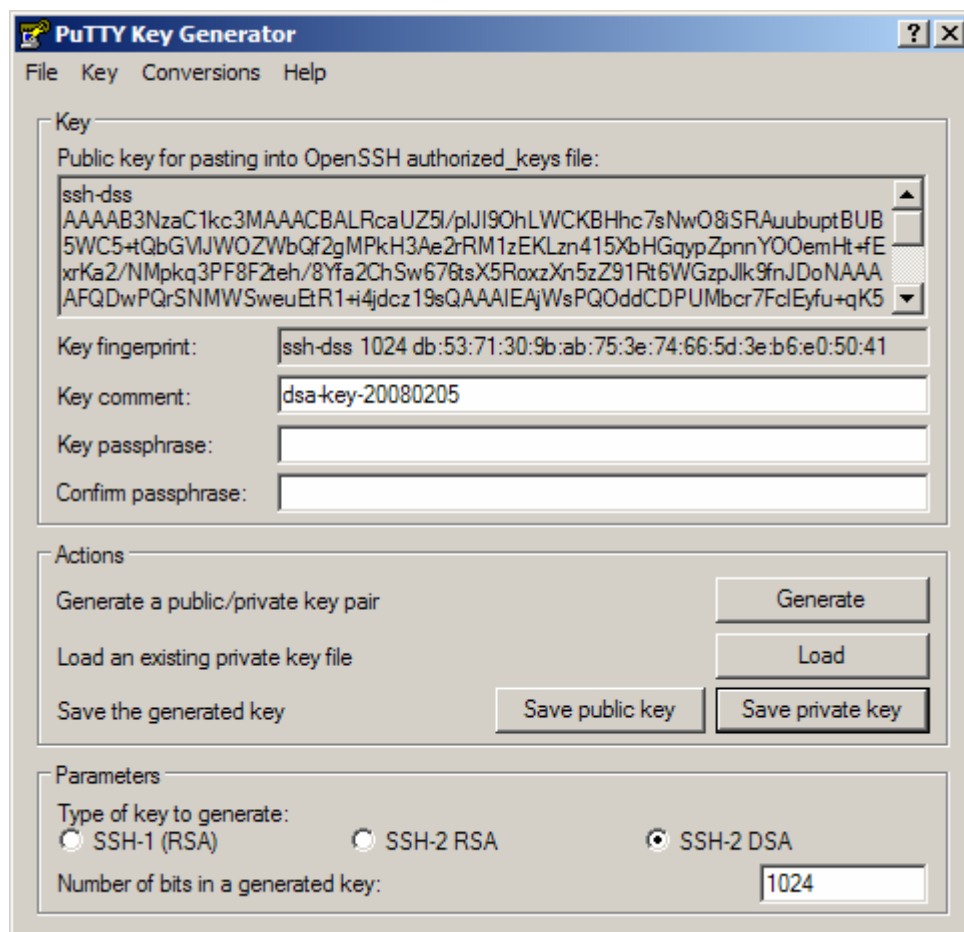
- TortoiseCVS (for example or other tools) for controlling the development version
<http://www.tortoisecvs.org/>
- Flex/Bison for Windows
<http://gnuwin32.sourceforge.net/packages/bison.htm>
<http://gnuwin32.sourceforge.net/packages/flex.htm>
- m4 : bison needs the installation of this software
<http://gnuwin32.sourceforge.net/packages/m4.htm>

- (option) NSIS for building an Installer:
http://nsis.sourceforge.net/Main_Page
- (option) libxml2 for Windows
<http://www.zlatkovic.com/libxml.en.html>
- PuTTY for defining the private-public keys (access to *Savannah*)
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
- (option) WinScp : tool for protocol SFTP
<http://winscp.net/eng/download.php>

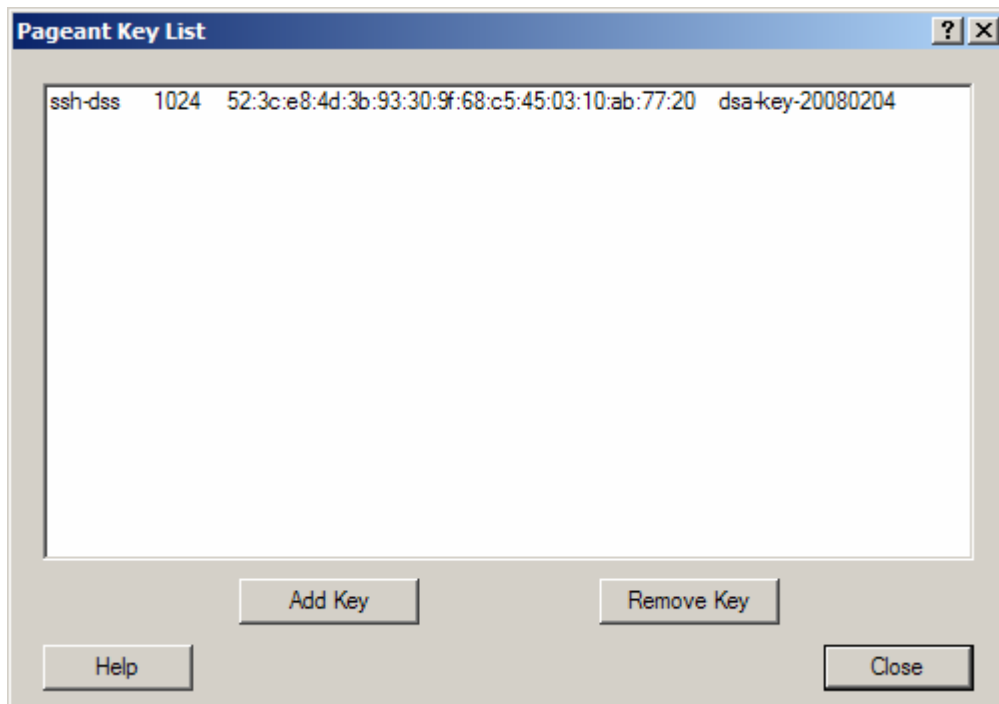
2/ public private Key

Building SSH 2 DSA keys with puttygen.exe; don't forget to move the mouse during the process for defining the keys!

Save the keys after defining the passphrase.



Add the private key to the key list with the program pageant.exe



Save the public key on *Savannah* after opening your account (My preference -> Modify public keys).

Mes préférences : Modifier vos clefs publiques

Clés Publiques SSH

Ce que vous devez copier/coller ressemble à ceci:

```
ssh-dss AAAAB.....[128 caractères].....DOQ== user@localhost.localdomain
```

Cela se trouve couramment dans `~/ .ssh/id_dsa.pub`.

Clefs publiques enregistrées :

Copiez, pour chaque clef que vous voulez enregistrer, la clef publique dans un champ. Après l'envoi du formulaire, vérifiez que le nombre de clefs recensées correspond à la valeur attendue.

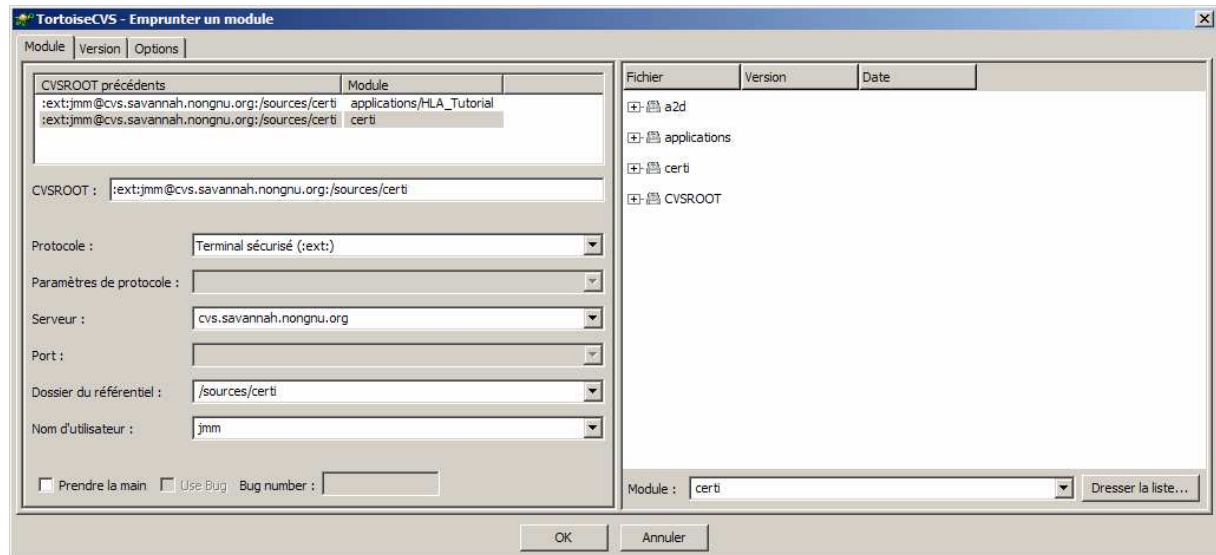
Take a look at <http://savannah.gnu.org/maintenance/SshAccess> for the building of keys

3/ Link to Savannah with SSH protocol

Run pageant.exe before giving the passphrase; this method avoids entering the passphrase for each transfer from/to *Savannah*.

4/ Create a folder for downloading the files

For example, define the folder '*certi*' before right clicking on it; Choose checkout!



Just enter the Protocol, the Server address, the folder for certi (/sources/certi), the user name, the module name (on the right)...

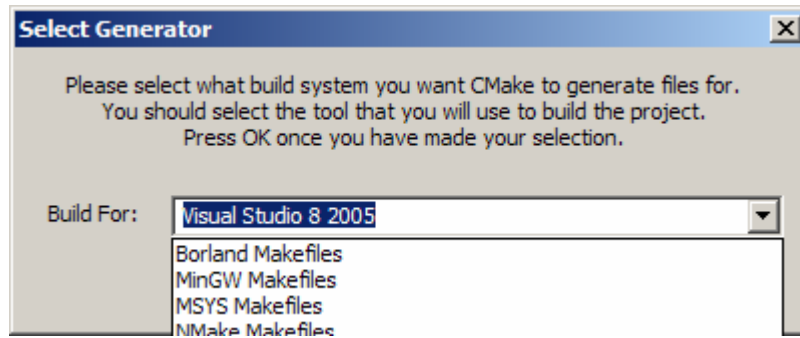
5/ Building : RTIG, RTIA, Federates

Enter the folder for the source code (here `\certi`, created during the previous step)

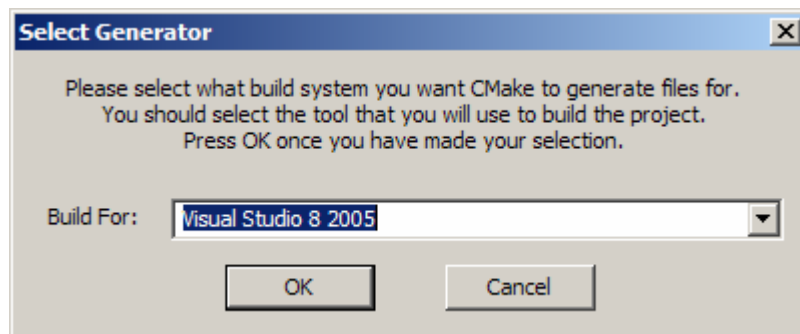
Enter the folder for building the programs (here `\tempo`)

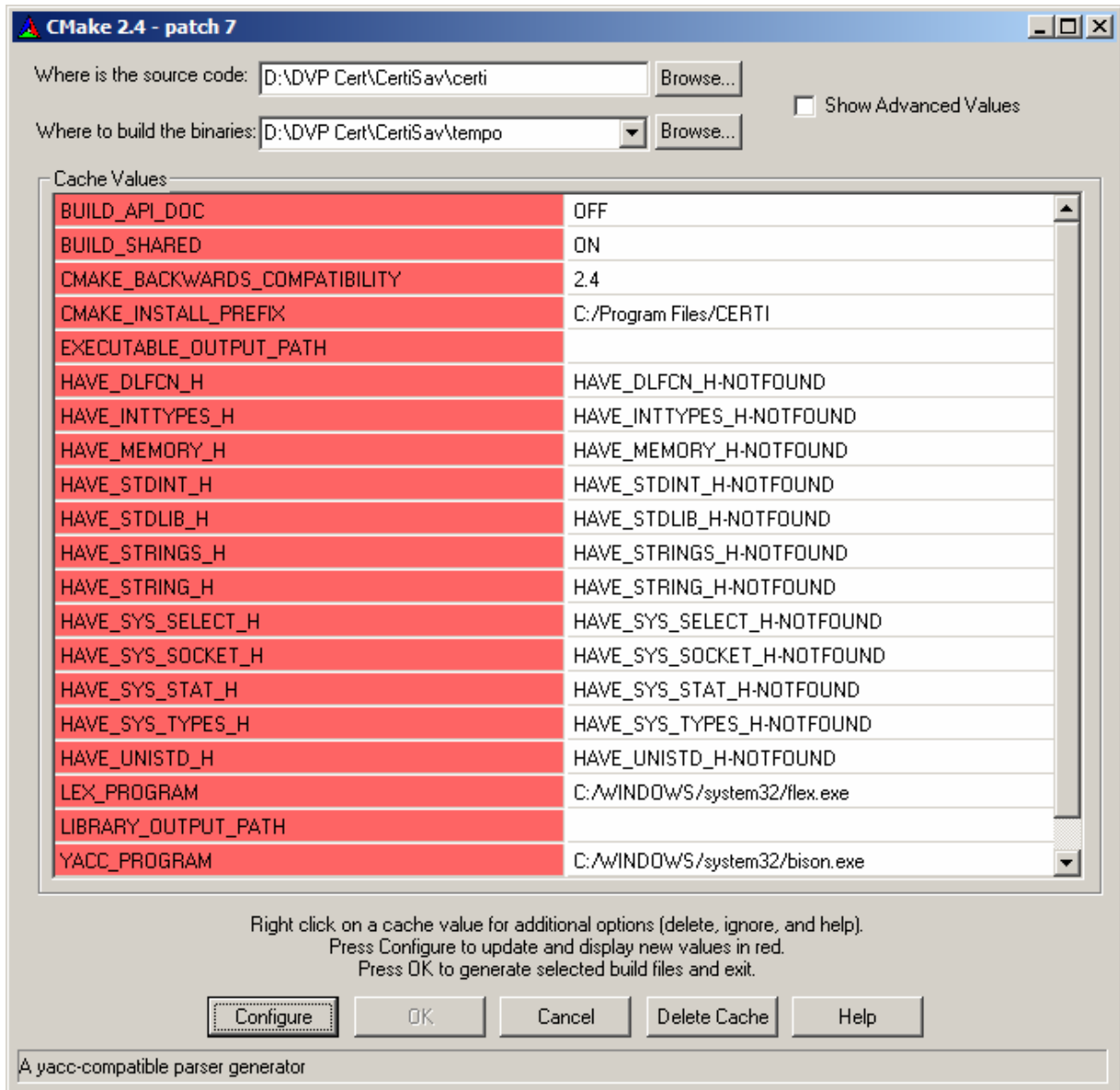


Configure...



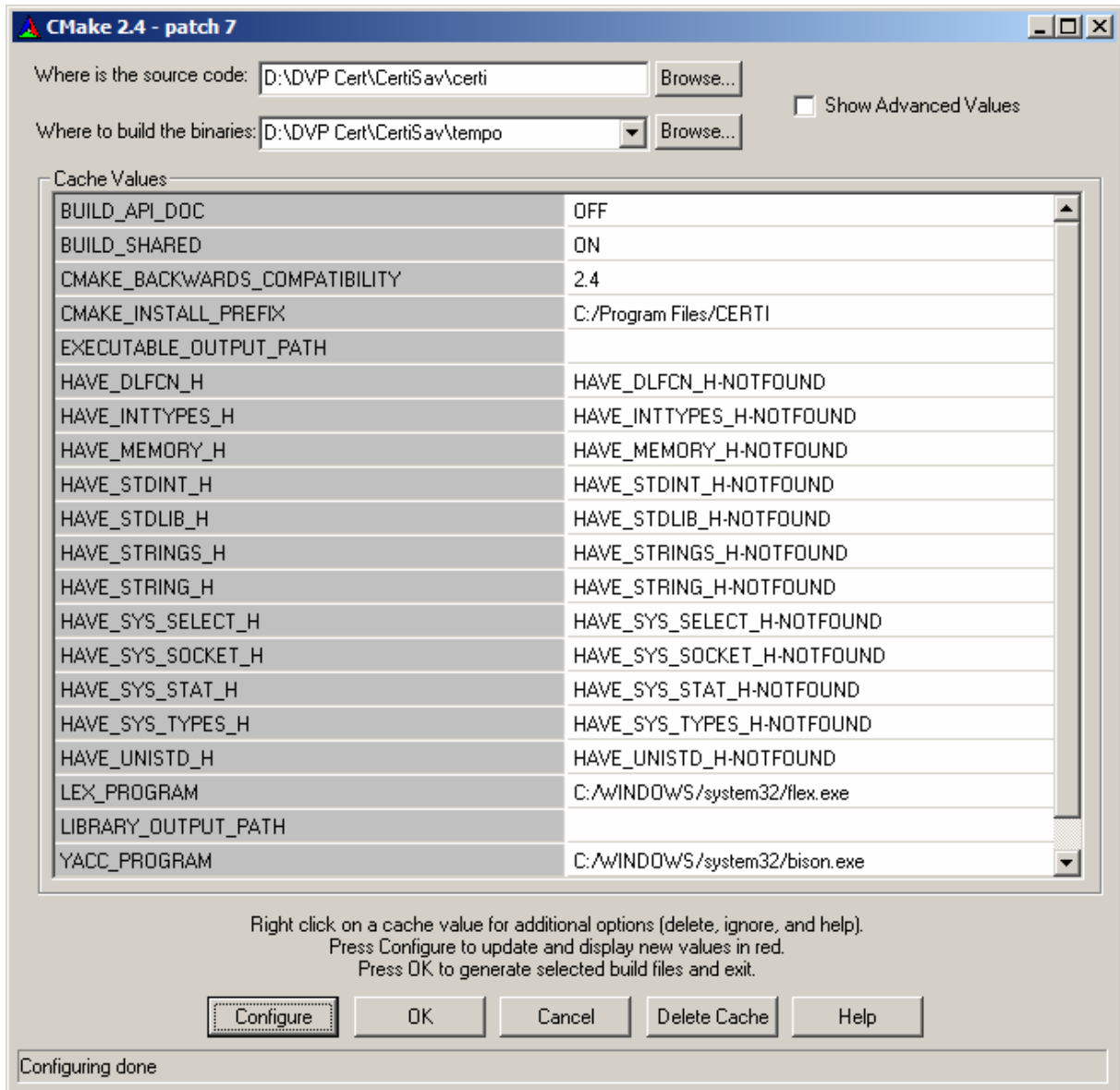
Choose your tool.



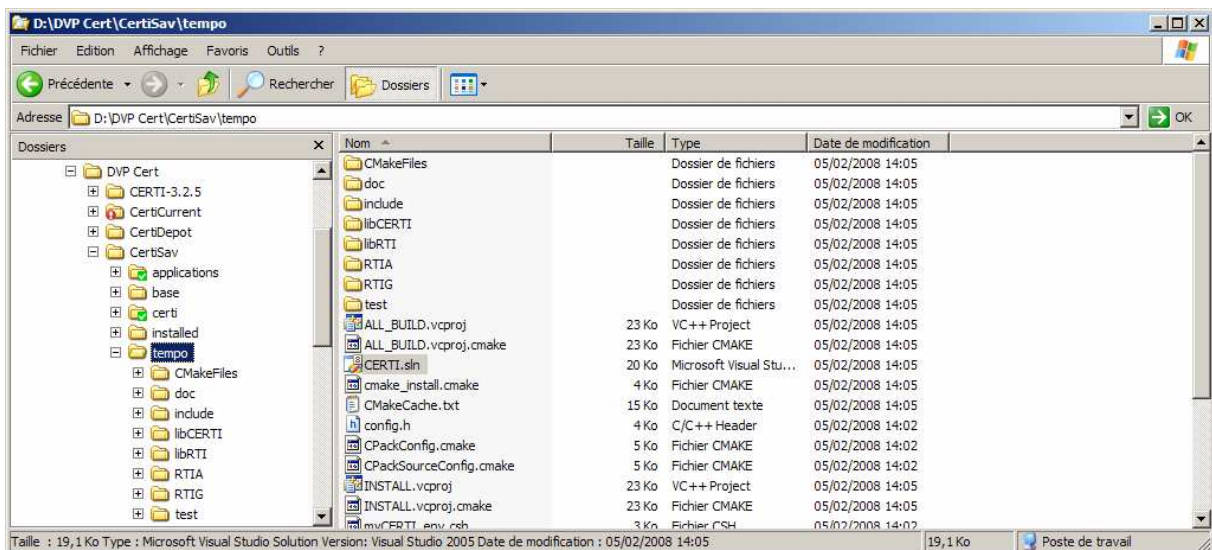


Verify the right addresses for the different tools (flex, bison..) before "Configure" again

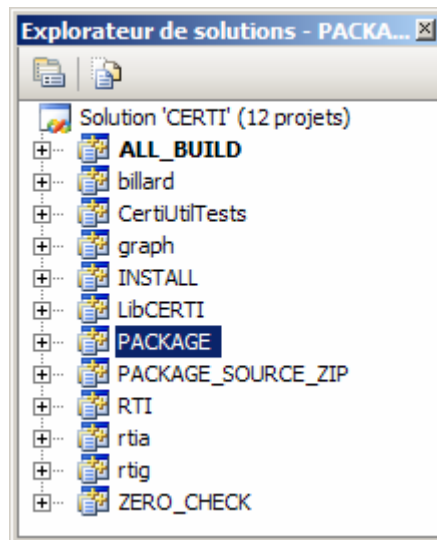
Repeat "Configure" for getting a valid OK box .



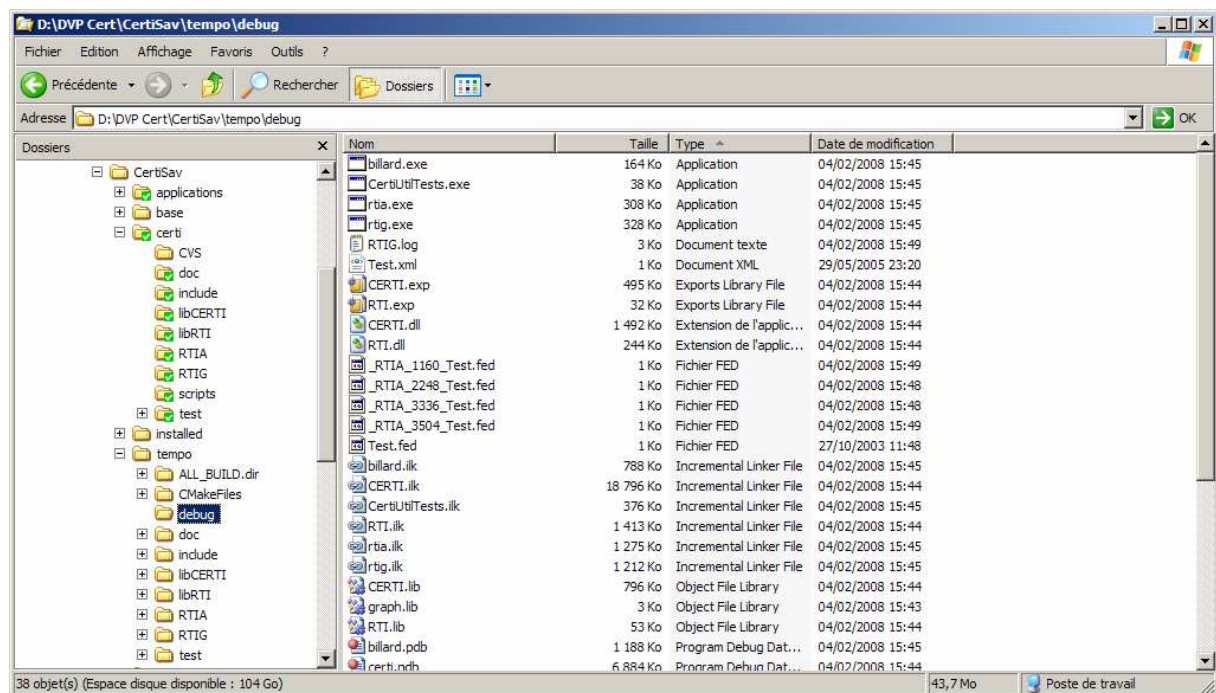
You get a right folder like that:



Launch CERTI.sln (here Visual C++ 2005 version)



Build all... Take a look at your folder ...



6) Check the test out

- open a windows command prompt before running RTIG
- open another windows command prompt before running the billard program.
- Don't forget the parameters : -n 1 fTest FTest.fed

```
C:\WINDOWS\system32\cmd.exe - rtig
D:\DUP Cert\CertiSav\base\debug>rtig
Updating : CERTI_HOME=rtig\
CERTI RTIG 3.2.6cvs - Copyright 2002-2006 ONERA
This is free software ; see the source for copying conditions. There is NO
warranty ; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

CERTI RTIG up and running ...
New federation: Test
Looking for FOM file...
Trying... Test.fed... opened.

<FED
  <federation "Test">
  <FEDversion "v1.3">
  <federate "fed" "Public">
  <spaces
    <space "Geo" <id 1>
      <dimension "X" <id 1>>
      <dimension "Y" <id 2>>>>
  <objects
    <class "ObjectRoot" <id 1>
      <attribute "privilegeToDelete" <id 1> reliable timestamp>
      <class "RTIprivate" <id 2>>
      <class "Bille" <id 3>
        <attribute "PositionX" <id 2> reliable timestamp>
        <attribute "PositionY" <id 3> reliable timestamp>
      <class "Boule" <id 4>
        <attribute "Color" <id 4> reliable timestamp>>>>
  <interactions
    <interaction "InteractionRoot" <id 1> best_effort receive>
    <interaction "RTIprivate" <id 2> best_effort receive>
    <interaction "Bing" <id 3> reliable timestamp>
      <sec_level "Public">
      <parameter "BoulNum" <id 1>>
      <parameter "DX" <id 2>>
      <parameter "DY" <id 3>>>>>

TCP Socket(RecevoirTCP) : No error
RTIG dropping client connection 1860.
TCP Socket 1860 : total = 358267b sent
TCP Socket 1860 : total = 889984b received
UDP Socket 1904 : total = 0b sent
UDP Socket 1904 : total = 0b received
```

```
C:\WINDOWS\system32\cmd.exe
D:\DUP Cert\CertiSav\base\debug>billard -n 1 -fTest -FTest.fed
CERTI Billard 3.2.6cvs
with TIMESTAMP. If you want without TIMESTAMP add -e option.

<FED
  <federation "Test">
  <FEDversion "v1.3">
  <federate "fed" "Public">
  <spaces
    <space "Geo" <id 1>
      <dimension "X" <id 1>>
      <dimension "Y" <id 2>>>>
  <objects
    <class "ObjectRoot" <id 1>
      <attribute "privilegeToDelete" <id 1> reliable timestamp>
      <class "RTIprivate" <id 2>>
      <class "Bille" <id 3>
        <attribute "PositionX" <id 2> reliable timestamp>
        <attribute "PositionY" <id 3> reliable timestamp>
      <class "Boule" <id 4>
        <attribute "Color" <id 4> reliable timestamp>>>>
  <interactions
    <interaction "InteractionRoot" <id 1> best_effort receive>
    <interaction "RTIprivate" <id 2> best_effort receive>
    <interaction "Bing" <id 3> reliable timestamp>
      <sec_level "Public">
      <parameter "BoulNum" <id 1>>
      <parameter "DX" <id 2>>
      <parameter "DY" <id 3>>>>>

Display(400, 25, 500, 100)
Press ENTER to start execution...

Declaration done.

RTIA: Received signal 2. Exiting peacefully.
Exit request received

Exiting.
```