Building Cantera with Visual C++ .NET

D. G. Goodwin
Caltech
December 2004
This document...

- explains how to compile and install Cantera from source code on a PC running Windows

- If you just want to use Cantera from Python or MATLAB, you can alternatively get the binary Cantera installer and skip everything described here
System Requirements

- Windows 2000 or XP
- Visual C++ .NET

- MATLAB 7.0
  - Required only if the Cantera MATLAB toolbox will be built
  - MATLAB 6.x may work too
Getting Ready…
Before you can install Cantera, you need to:

- Install Python
- Install numarray
- Set environment variables
- Set up MATLAB

These steps only need to be done once

The next few slides will describe how to do these four steps in detail
Step 1: Install Python

- Python 2.4 or greater is required, since earlier versions of Python were compiled with Visual C++ 6.0

- Uninstall any older version of Python on your system

- Download and run the **Python 2.4 Windows Installer** from [http://www.python.org](http://www.python.org)

- Click the link below to get the Python installer now:
  - [http://www.python.org/ftp/python/2.4/python-2.4.msi](http://www.python.org/ftp/python/2.4/python-2.4.msi)
Step 2: Install numarray

- This step is only required if you plan to use Cantera from Python (but is highly recommended in any case, and only takes a minute).

- Get and run the numarray Windows binary installer for Python 2.4 from http://sourceforge.net/projects/numpy

- Click below to get the numarray installer now http://prdownloads.sourceforge.net/numpy/numarray-1.1.1.win32-py2.4.exe?download

(Note: you can also use the older ‘Numeric’ package instead of ‘numarray’, but if you do you will need to edit the file ‘config.h’ in the Cantera source code.)
Step 3: Set environment variables

- The build process runs batch files that invoke Python and MATLAB.

- These batch files use environment variables PYTHON_CMD and MATLAB_CMD.

- To set these:
  - From the control panel, select System.
  - Select the Advanced tab, and press the Environment Variables button.
Create new entries PYTHON_CMD and MATLAB_CMD

Set each one to the path to the executable file

Note: if you don’t have MATLAB, set MATLAB_CMD to REM
Step 4: Configure MATLAB

- MATLAB needs to know which C/C++ compiler you want to use to build the Cantera Toolbox

- Start MATLAB, and at the MATLAB prompt type:
  ```
  >> mex -setup
  ```

- Select Visual C++ from the list of available compilers

- Exit MATLAB
Getting the Cantera Source Code
The Cantera source code is maintained on Sourceforge at http://sourceforge.net/projects/cantera

You can either
- download the latest Windows source distribution, or
- Check out the source code using CVS
Which method should you use?

- **Downloading the source distribution**
  - **Pro:**
    - Relatively stable; demos and test programs are verified to build and run before a new source distribution is released.
    - Does not require a CVS client.
  - **Con:** Updated infrequently. New features or bug fixes are not available until the next source distribution is released.

- **Using CVS**
  - **Pro:** Access to latest Cantera code. New features, bug fixes available immediately.
  - **Con:** Requires a CVS client program.

- The choice is up to you!
Option 1: Downloading the source distribution

- At the Cantera Sourceforge site, select ‘Files’, then choose the latest Windows source distribution (cantera-1.x.x-src.zip)

- Alternatively, follow this link to download cantera-1.5.5-src.zip now:  
  [http://prdownloads.sourceforge.net/cantera/cantera-1.5.5-src.zip?download](http://prdownloads.sourceforge.net/cantera/cantera-1.5.5-src.zip?download)

- Extract the files in the zip archive into any temporary directory
Option 2: Getting Cantera via CVS

- These instructions assume you have a CVS client that runs from the command line. If you have a graphical CVS client like WinCVS, the procedures will be somewhat different.

- First log in by typing

  ```
cvs -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/cantera login
  ```

- Now check out module cantera:

  ```
cvs -z3 -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/cantera co cantera
  ```

- **Note**: if you define environment variable CVS_ROOT to be the long string beginning with pserver:; you can simply type ‘cvs -z3 co cantera’
Getting Cantera via CVS (cont’d)

- After the checkout procedure finishes, you will find a new directory called ‘cantera’

- Go into this directory and remove the directory named ‘win32’ within it. (This contains old Visual C++ 6.0 project files.)

- Now from the new ‘cantera’ directory, check out module ‘win32’:

  cd cantera
cvs -z3 -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/cantera co win32

- The end result should be that you now have a new ‘win32’ directory within ‘cantera’, replacing the old one you deleted.
Updating Cantera via CVS

- If you checked out Cantera using CVS, you can update it by typing
  
  `cvs update`

  from the top Cantera directory.

- This will download all files in all subdirectories that have changed since you checked out or last updated Cantera.

- To revert to Release 1.5.5 and undo any later changes that have been made to the repository, type
  
  `cvs update -r Release_1_5_5`
Building Cantera in Visual Studio
Open the Cantera solution in Visual Studio

- Start Visual Studio, and from the File menu select ‘Open Solution’ (or ‘Open Project’)
- Navigate down to the ‘cantera\win32\vc7’ folder and open solution ‘cantera’ (file ‘cantera.sln’).
You should see something like this:
Building Cantera

- Change the configuration to ‘Release’

- From the Build menu, select ‘Build Solution’

- This will build:
  - The Cantera static libraries
  - The ‘clib’ DLL used to access Cantera from Python and MATLAB
  - The Python and MATLAB interface modules

- Output files are put into a temporary location (folder ‘cantera/build’); you need to ‘install’ them (as described next) before they will be available for use.
Installing Cantera
Source and Installation Trees

- Cantera uses two different arrangements of files in directories (‘trees’)

- The source tree is what you have been working with, and is designed for use by developers.

- The installation tree is the final installed set of files, designed for use by users. This is what binary installers create (which is the next step in the process).
The Cantera source tree is laid out for the convenience of Cantera developers, not Cantera users / application programmers.

- The Cantera kernel is in ‘Cantera/src’, and the language interfaces each occupy a directory within ‘Cantera’

- All source, header, demo, and tutorial files for each language are within its own directory.

- Other directories contain utility programs, tools for testing or generating documentation, unix and Mac build tools, etc. Most of this is not relevant for end-users who only want to run Cantera applications.
Installation Tree Layout

- The installation tree is laid out so that
  - Utility programs (ck2cti.exe) are easy to find
  - Python and MATLAB demo scripts are easily accessible
  - Header and library files for C++ application programs are collected together in one place

- Data files and the clib DLL are put in special locations where Cantera can find them

- **Next step**: building an Installer to generate the installation tree.
Building a Cantera Installer

- Running ‘Build Solution’ in Visual Studio skipped one project (‘SetupCantera’)

- Now right-click on this project, and select ‘Build’

- This will create a binary Windows Installer file for Cantera named Cantera.msi

- This installer can be used on any Windows PC to install Cantera
Running the Installer

- To install Cantera, right-click again on ‘SetupCantera’ and select ‘Install’

- This will run the Cantera.msi installer you created, and will install Cantera at a location you specify. Make sure this is not the same as the source directory where you have built Cantera!

- You can install Cantera anywhere you like. The C++ demo project assumes Cantera is in C:\CANTERA, so if you want to test Cantera quickly without editing any project files, choose this installation directory.

- If you have done a previous installation of Cantera, the installer will automatically remove it before installing the new installation
Testing the Installation

- After running the installer, you should have
  - a ‘Cantera’ entry in the ‘Start’ menu
  - A shortcut to the installation folder on your desktop

- From the Start menu, or using the shortcut, navigate to the Python demos and double-click on any one of them to verify it runs.

- Then select ‘C++ Demos’ from the Cantera entry of the Start menu. This will open Visual Studio with a Cantera demo application. If it builds, then you are ready to start using Cantera to write your own C++ applications…
Setting the MATLAB Path

- To use the Cantera Toolbox in MATLAB, you need to add it to the MATLAB path.

- Start MATLAB, and from the File menu select ‘Set Path’.

- Select ‘Add with subfolders’ and navigate to folder ‘MATLAB\Toolbox\cantera’ in the installation directory.

- Press the ‘Save’ button to save this path.
Running the MATLAB Demos

- Within MATLAB, go to the ‘demos/MATLAB’ folder in the installation directory.

- Type the name of any m-file in this folder to run it, or type ‘run_demos’ to run them all.
If you have problems...

- Look at the messages at the Cantera User’s Group site to see if someone else has already dealt with the issue. [http://groups.yahoo.com/groups/cantera](http://groups.yahoo.com/groups/cantera)

- If not, send a message to the Cantera User’s Group describing the problem, and probably someone will be able to help.

- And if you run into a problem in the build procedure and fix it, please let us know the problem and fix!