Building Cantera 1.7 with Visual C++ .NET

D. G. Goodwin

Caltech

November 2006

This presentation…

- 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 distribution and skip everything described here

System Requirements

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

- Optional: MATLAB
  - Only needed if you want to build the Cantera MATLAB Toolbox
  - Version 7.0 or greater is required

Getting Ready

- If you are installing Cantera for the first time, you need to first
  - Install Python
  - Install numarray*
  - Install SUNDIALS*
  - Set environment variables
  - Set up MATLAB*

- If you are upgrading from a previous Cantera installation and have already done these steps before, there is no need to do them again, unless you want to install newer versions of Python or numarray

*optional

Installing Python

- Python 2.4 or greater is required, since earlier versions of Python were compiled with Visual C++ 6.0, not Visual C++ .NET
- If you have Python 2.4 already, you can keep it, or you can download the current version (Python 2.5) from www.python.org
- If you do update Python, be sure to uninstall any older version of Python first
- Click the link below to get the Python 2.5 installer now:
    - http://www.python.org/ftp/python/2.5/python-2.5.msi

Installing 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 or 2.5 from http://sourceforge.net/projects/numpy

- Note: get the Windows installer for **numarray**, not the newer **numpy** package that Cantera does not yet support

Setting environment variables

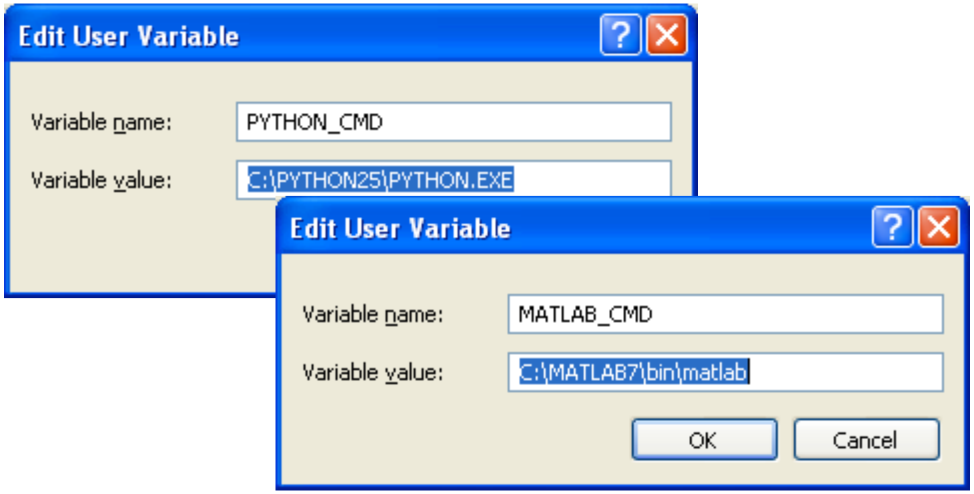
- The build process runs batch files that invoke Python and MATLAB

- 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 user environment variables
  - PYTHON_CMD
  - MATLAB_CMD

- Set each one to the path to the executable file



Edit User Variable

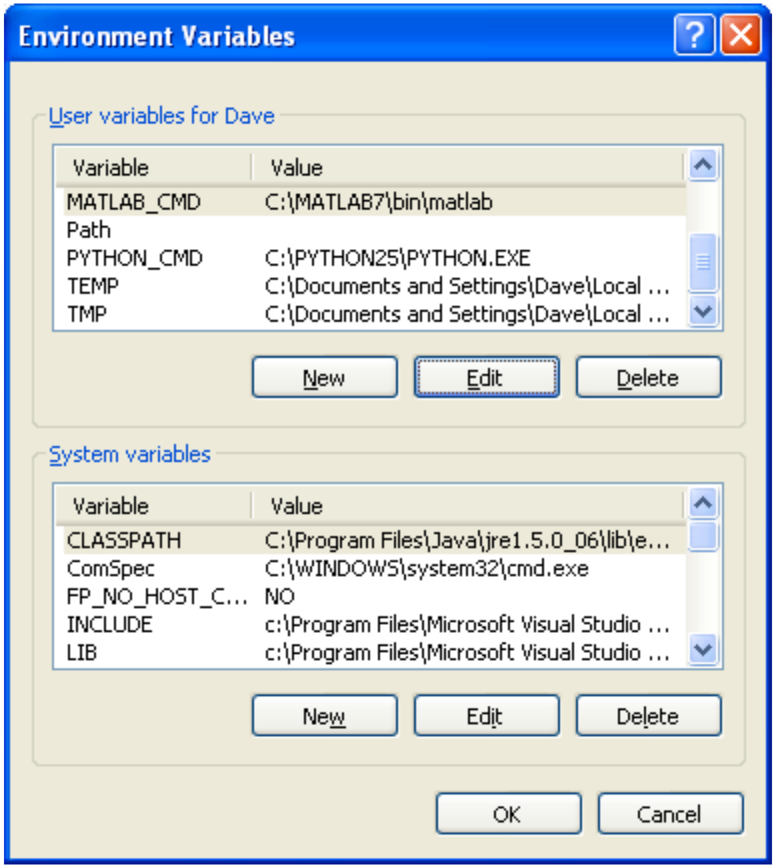
Variable name: PYTHON_CMD

Variable value: C:\PYTHON25\PYTHON.EXE

Edit User Variable

Variable name: MATLAB_CMD

Variable value: C:\MATLAB7\bin\matlab



Environment Variables

User variables for Dave

Variable	Value
MATLAB_CMD	C:\MATLAB7\bin\matlab
Path	
PYTHON_CMD	C:\PYTHON25\PYTHON.EXE
TEMP	C:\Documents and Settings\Dave\Local ...
TMP	C:\Documents and Settings\Dave\Local ...

System variables

Variable	Value
CLASSPATH	C:\Program Files\Java\jre1.5.0_06\lib\e...
ComSpec	C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_C...	NO
INCLUDE	c:\Program Files\Microsoft Visual Studio ...
LIB	c:\Program Files\Microsoft Visual Studio ...

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

Configuring MATLAB

- MATLAB needs to know which C/C++ compiler you want to use to build the Cantera MATLAB 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 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 in a zip file
  - **Pro**: Simple.
  - **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. (WinCVS is a good GUI CVS program.)
  - Tags inserted into the repository at each new release allow checking out the same code as in the zip file, even if later changes to the code in the repository have been made. Or you can get the latest development version.

Option 1: Downloading the source distribution

- At the Cantera Sourceforge site, select ‘Download Cantera’, then select the latest Windows source distribution (*-src.zip)

- Currently this is cantera-1.7.0-src.zip

- Extract the files in the zip archive into some 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, use the analogous procedures for that client

- First log in by typing

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

- Now check out module cantera:

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

Getting Cantera via CVS (cont'd)

- After the checkout procedure finishes, you will find a new folder called 'cantera'

- This folder contains the Cantera source code, and unix Makefiles, but not Visual C++ .NET project files. To get these, go into the 'cantera' folder/directory, and check out module 'win32':

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

- The end result should be that you now have a 'win32' folder within the 'cantera' folder.

Updating Cantera via CVS

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

  ```
  cvs update –d
  ```

  from the top Cantera directory. This will download all files in all subdirectories that have changed since you checked out or last updated Cantera

- If you want to specifically check out Cantera 1.7.0, do this:

  ```
  cvs update –d –r Release_1_7_0
  ```

SUNDIALS

- Cantera uses the CVODES ODE integrator from the “SUNDIALS” package to integrate reactor network ODES and to calculate sensitivity coefficients

- It is possible to build Cantera without SUNDIALS, since an older version of the CVODE integrator is included in the Cantera source distribution (but then you can’t do sensitivity analysis). For this option, use solution file Cantera_no_sundials

- The Cantera build procedure will build SUNDIALS too.

Getting SUNDIALS

- Go to www.llnl.gov/casc/sundials/ and download the latest source distribution (currently sundials-2.3.0.tar.gz)

- SUNDIALS is only distributed in unix “tar.gz” format, but most PC file compression utilities can handle this format.

- Get the entire SUNDIALS distribution, not the separate components

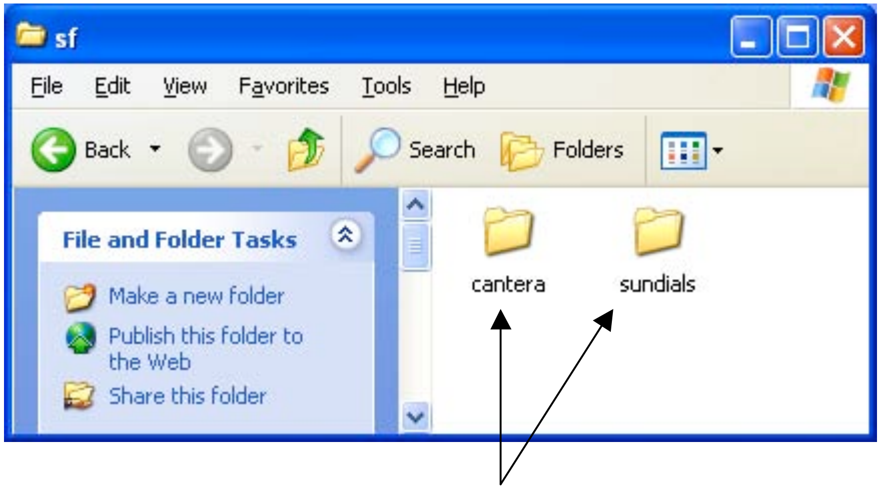
- Extract the files from the archive file; this should create a folder called sundials-2.3.0 (or similar)

Building SUNDIALS

- The Cantera Solution file contains projects to build those components of SUNDIALS required by Cantera, and they will be built automatically along with Cantera.

- All you need to do is:
  - Change the name of the SUNDIALS source folder to “sundials” (from e.g. “sundials-2.3.0”)

  - Drag the “sundials” folder into the folder where “cantera” is located (but not *into* the “cantera folder!)



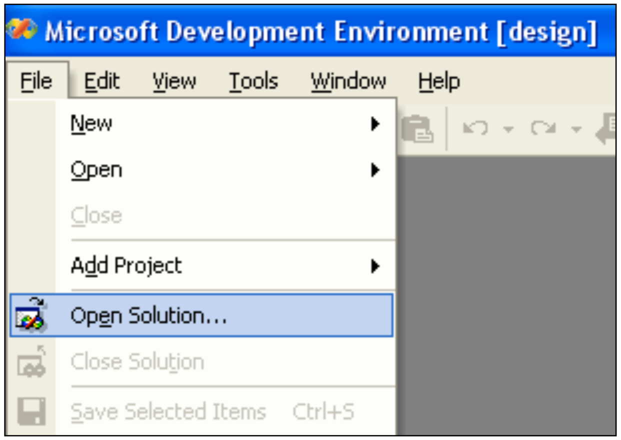
The Cantera and sundials source folders must both be in the same folder (here “sf”)

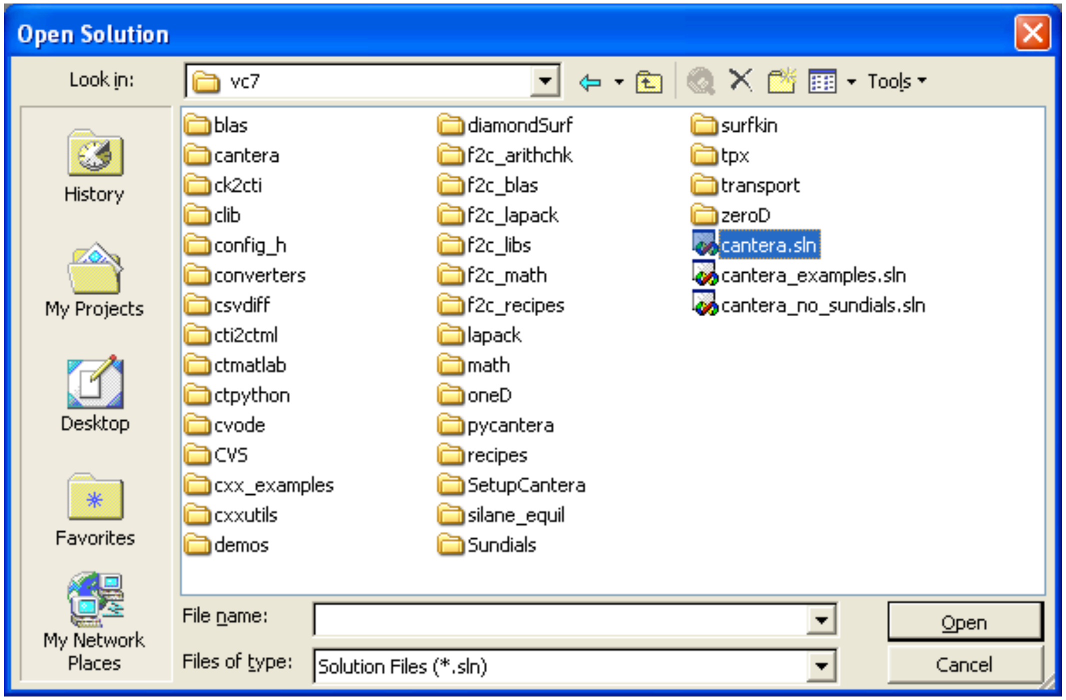
Open the Cantera solution in Visual Studio

- Now that we have the source code, and the VC++ .NET project files, we're ready to build.

- So start Visual Studio, and from the File menu select 'Open Solution'

- Navigate to 'cantera\win32\vc7' and open solution 'cantera'



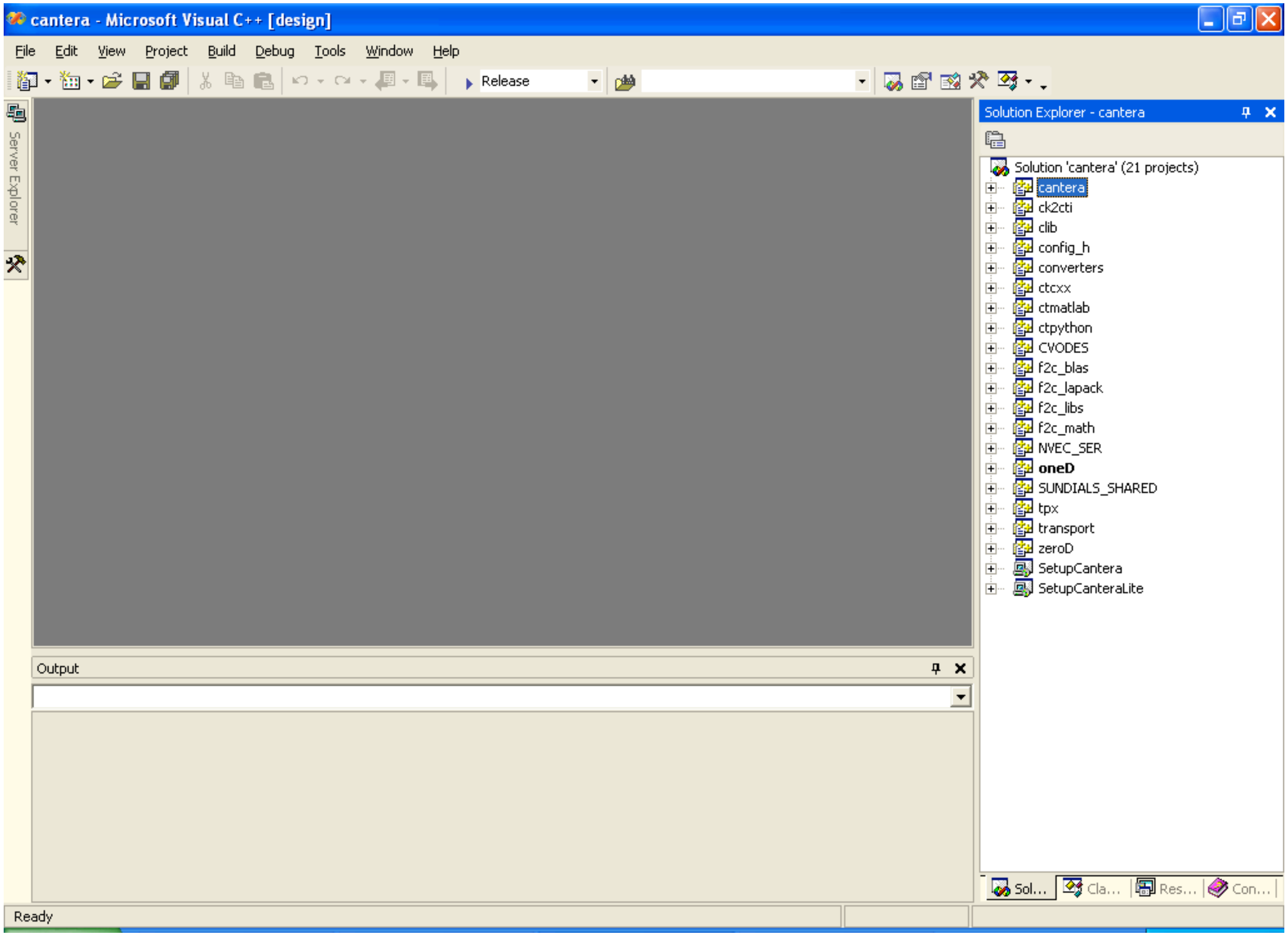


Look in: vc7

File name:

Files of type: Solution Files (*.sln)

You should see something like this:



Solution 'cantera' (21 projects)
- cantera
- ck2cti
- clib
- config_h
- converters
- ctcxx
- ctmatlab
- ctpython
- CVODES
- f2c_blas
- f2c_lapack
- f2c_libs
- f2c_math
- NVEC_SER
- **oneD**
- SUNDIALS_SHARED
- tpx
- transport
- zeroD
- SetupCantera
- SetupCanteraLite

Building Cantera

- Change the configuration to ‘Release’
- From the Build menu, select ‘Build Solution’
- This will build:
  - The Cantera static libraries
  - A few sundials static libraries
  - The ‘clib’ DLL used to access Cantera from Python and MATLAB
  - The Python and MATLAB interface modules

Source distribution layout

- The Cantera source distribution 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 directory.
  - Other directories contain application programs, tools for testing or generating documentation, etc.
  - Many things in the source distribution are useful only on unix-like systems (Makefiles, configure, etc.)

Installation Layout

- The build process installs Cantera files in a separate directory (folder)

- This directory 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

Building a Cantera Installer

- Running ‘Build Solution’ in Visual Studio skips two projects (SetupCantera and SetupCanteraLite)

- Right-click on SetupCantera, 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, and includes the static libraries needed to build C++ applications, and a set of C++ demonstration programs.

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!

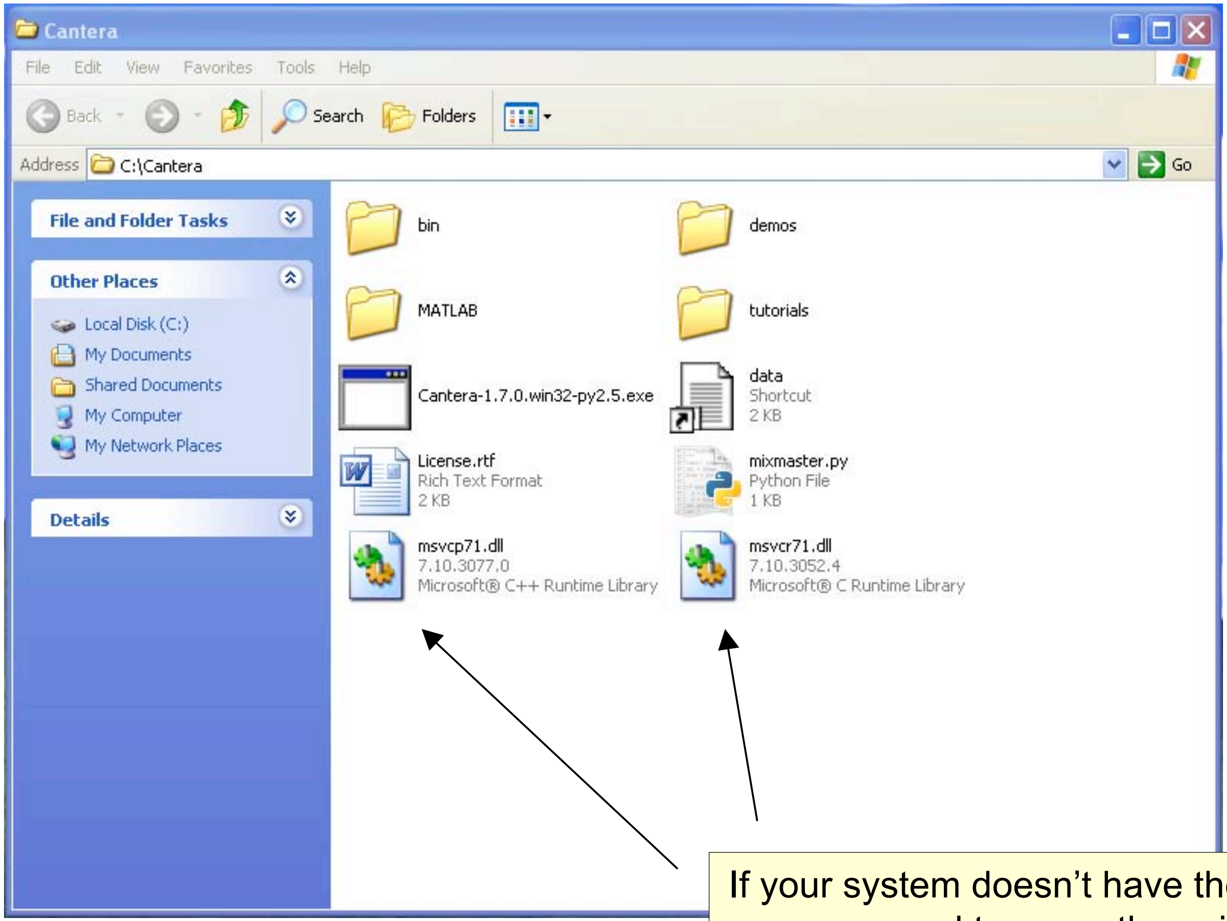
- If you have done a previous installation of Cantera, the installer will automatically remove it before installing the new installation. However, it won’t remove output files, so you might want to clean up the folder or remove it yourself before installing the new Cantera files.

SetupCantera and SetupCanteraLite

- SetupCanteraLite
  - Creates a Windows installer (Cantera.msi) file that can be used to install Cantera on any Windows PC, including ones without Visual C++
  - Installs in the Cantera installation folder:
    - An installer for the Cantera Python package
    - The MATLAB toolbox
    - Python and MATLAB demos
    - MixMaster
    - ck2cti.exe
  - Use this installer when Cantera will only be used from Python and/or MATLAB
- SetupCantera
  - Everything that SetupCanteraLite installs, but in addition installs
    - Cantera C++ header files
    - Cantera and SUNDIALS static libraries to use to link C++ programs
    - C++ demo programs
  - Install Cantera using this installer on machines where Cantera C++ programs need to be compiled
  - Note that it is not necessary to build Cantera from source on every machine – this installer provides everything needed to build Cantera C++ applications

The Installation Folder

- After running the Cantera.msi* installer, you should have a new folder that looks something like this:



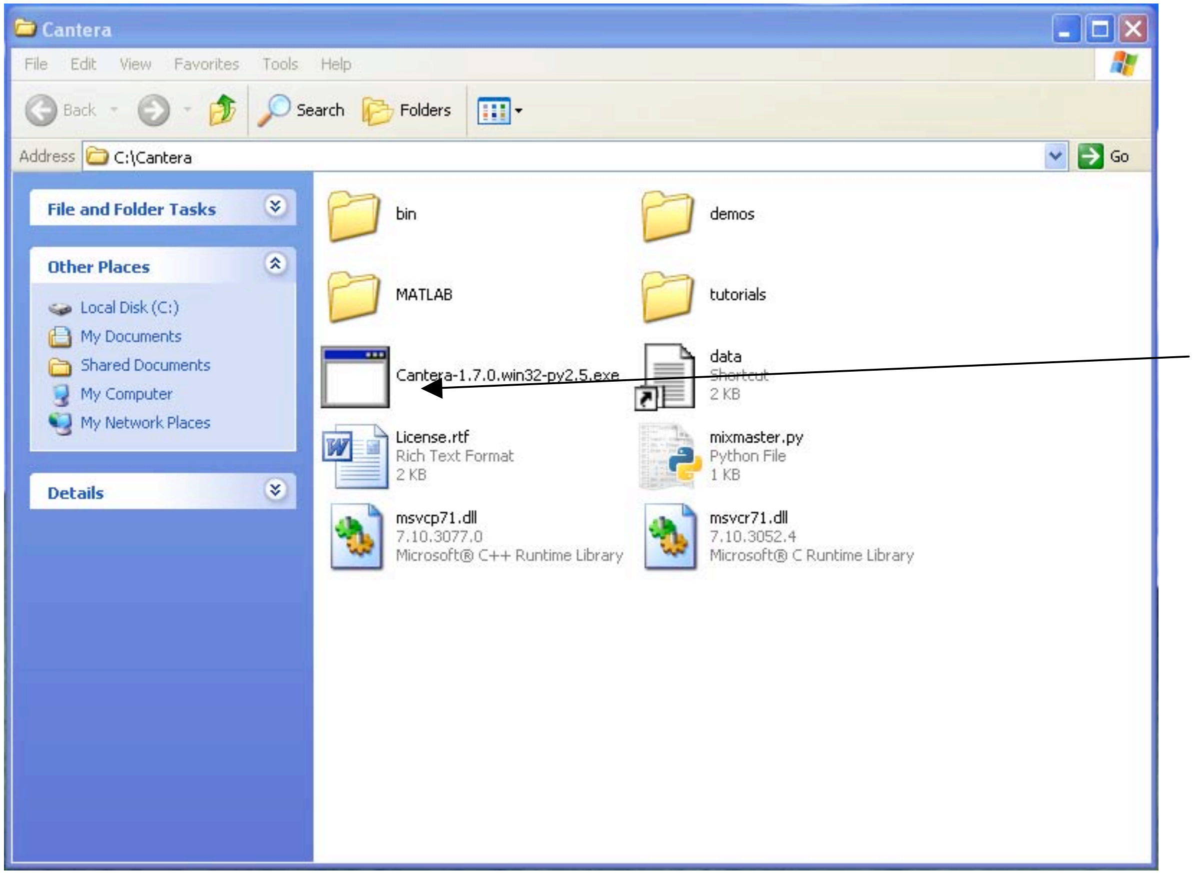
Address C:\Cantera

bin
demos
MATLAB
tutorials
Cantera-1.7.0.win32-py2.5.exe
data
Shortcut
2 KB
License.rtf
Rich Text Format
2 KB
mixmaster.py
Python File
1 KB
msvcp71.dll
7.10.3077.0
Microsoft® C++ Runtime Library
msvcr71.dll
7.10.3052.4
Microsoft® C Runtime Library

If your system doesn’t have these DLLs already, you may need to move these into the Windows system folder

*generated by SetupCanteraLite

Installing the Cantera Python Interface



Address C:\Cantera

- When you built project ‘ctpython’ in Visual Studio, the Cantera Python interface was installed as part of the build process
- But if you want to run Cantera.msi on other machines to install Cantera, you need to manually install the Python interface by executing this installer
- Note that you need this even if you plan to use Cantera from MATLAB or C++.

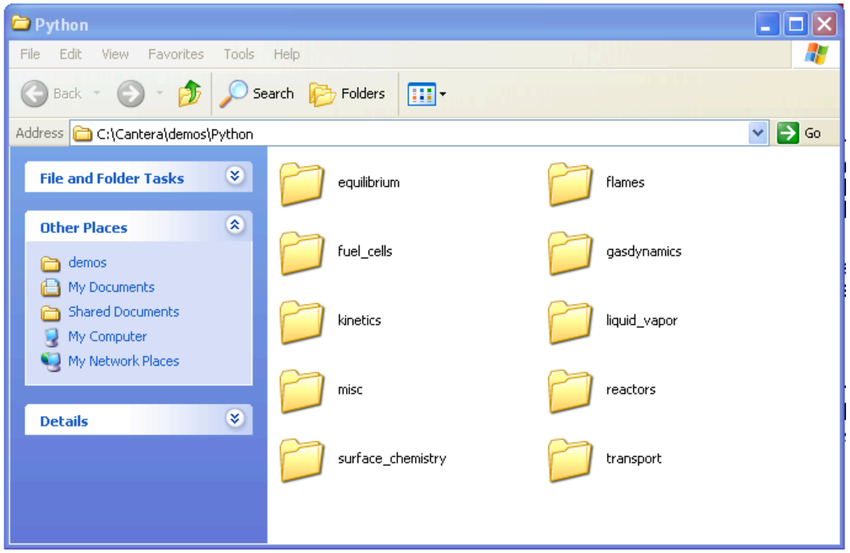
Testing the Installation: Running the Python Demos

- From the Start menu, or using the desktop shortcut, navigate to the Python demos folder within the installation folder

- The demos are grouped into folders

- Open any folder, and double-click on any script to verify it works

- Alternatively, right-click on the icon, and select “Edit with IDLE”. Click in the window showing the file contents, and press F5 or select “Run / Run Module”



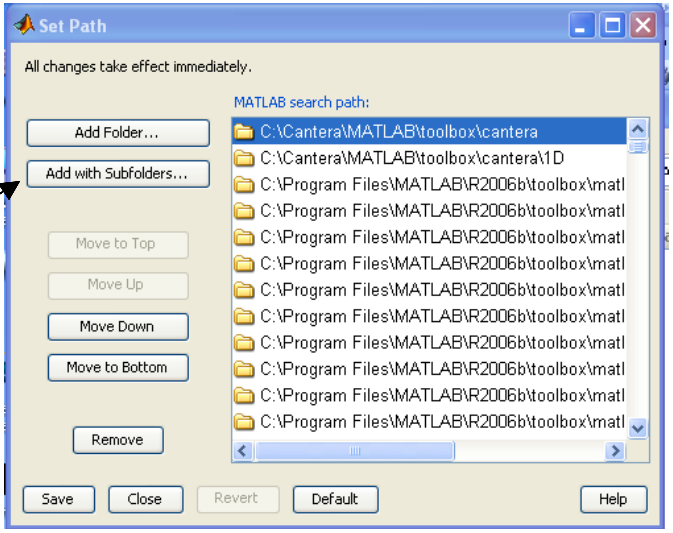
Configuring MATLAB

- Now that you have installed Cantera, you need to add the Cantera Toolbox to the MATLAB path, if you want to use Cantera from MATLAB.

- 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.



Set Path

All changes take effect immediately.

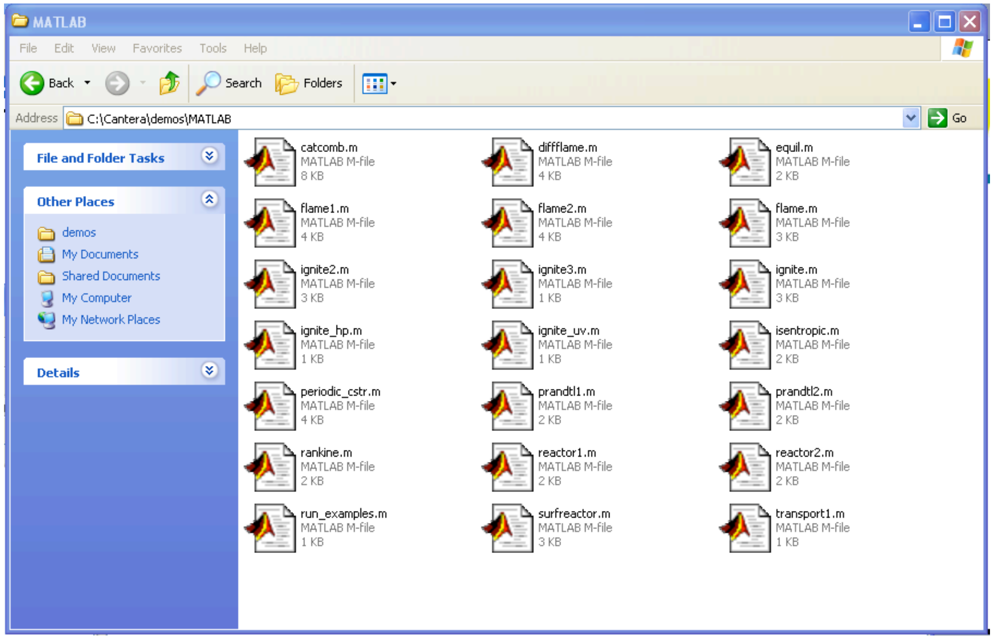
MATLAB search path:

C:\Cantera\MATLAB\toolbox\cantera
C:\Cantera\MATLAB\toolbox\cantera\1D
C:\Program Files\MATLAB\R2006b\toolbox\matl
C:\Program Files\MATLAB\R2006b\toolbox\matl
C:\Program Files\MATLAB\R2006b\toolbox\matl
C:\Program Files\MATLAB\R2006b\toolbox\matl
C:\Program Files\MATLAB\R2006b\toolbox\matl
C:\Program Files\MATLAB\R2006b\toolbox\matl
C:\Program Files\MATLAB\R2006b\toolbox\matl
C:\Program Files\MATLAB\R2006b\toolbox\matl
C:\Program Files\MATLAB\R2006b\toolbox\matl
C:\Program Files\MATLAB\R2006b\toolbox\matl

Running the MATLAB Demos

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

- Type the name of any m-file in this folder to run it, or type ‘run_examples’ to run them all.



C:\Cantera\demos\MATLAB

catcomb.m MATLAB M-file 8 KB
diffflame.m MATLAB M-file 4 KB
equil.m MATLAB M-file 2 KB
flame1.m MATLAB M-file 4 KB
flame2.m MATLAB M-file 4 KB
flame.m MATLAB M-file 3 KB
ignite2.m MATLAB M-file 3 KB
ignite3.m MATLAB M-file 1 KB
ignite.m MATLAB M-file 3 KB
ignite_hp.m MATLAB M-file 1 KB
ignite_uv.m MATLAB M-file 1 KB
isentropic.m MATLAB M-file 2 KB
periodic_cstr.m MATLAB M-file 4 KB
prandtl1.m MATLAB M-file 2 KB
prandtl2.m MATLAB M-file 2 KB
rankine.m MATLAB M-file 2 KB
reactor1.m MATLAB M-file 2 KB
reactor2.m MATLAB M-file 2 KB
run_examples.m MATLAB M-file 1 KB
surfreactor.m MATLAB M-file 3 KB
transport1.m MATLAB M-file 1 KB

Running the C++ Demos

- Go to the C++ demo folder, and double-click on CanteraDemos.vdproj
- This will start Visual Studio .NET
- Build the solution, and verify that the projects run.
- Note: if Cantera is not installed in C:\CANTERA, the project files may need to be edited before loding them.

Buiding your own C++ Cantera application

- To start a new project, the best thing to do is to *copy* one of the C++ demo project files (*.vcproj)

- The open it in Visual Studio, and change the source files to those for your application

- This process will insure that include and library paths are set correctly.

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

- 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, let us know so we can update the source distribution and CVS repository.