Installer paraview depuis les sources

03/06/2015

Tristan Agaesse

Contenu

[Introduction 1](#_Toc461194026)

[Mode d’emploi linux pour paraview-superbuild 2](#_Toc461194027)

[Configurer Cmake 2](#_Toc461194028)

[Annexe : mode d’emploi windows sans superbuild 4](#_Toc461194029)

[Prérequis 4](#_Toc461194030)

[Installer paraview 4](#_Toc461194031)

[Etape Cmake 5](#_Toc461194032)

[Etape Visual Studio 5](#_Toc461194033)

[Annexe : dépendances linux si on n’utilise pas paraview-superbuild 5](#_Toc461194034)

[Installer Qt 5](#_Toc461194035)

[Installer Mesa 6](#_Toc461194036)

# Introduction

Pour lier paraview à une version personnalisée de python il faut installer soi-même paraview depuis les sources. En effet ça permet de personnaliser les options d’installation, ce qui n’est pas possible avec l’installateur binaire de paraview.

Le but de la manœuvre est d’exécuter des scripts python dans paraview avec des librairies python installées dans sa distribution python personnelle (type anaconda). En effet la version standard de paraview utilise une version de python qui se trouve dans le répertoire d’installation de paraview. Cette version python et ses librairies (numpy etc…) sont incompatibles avec les librairies installées par anaconda si elles ne correspondent pas au même numéro de version.

Je recommande d’utiliser un paraview superbuild, qui contient déjà toutes les dépendances de paraview. Il n’y a donc pas besoin de les réinstaller, ça épargne beaucoup de travail.

# Mode d’emploi linux pour paraview-superbuild

<http://www.paraview.org/Wiki/ParaView:Build_And_Install>

Je fais toutes les manips d’installation dans une fenêtre xterm sur un nœud de aar sous l’OS sl6.

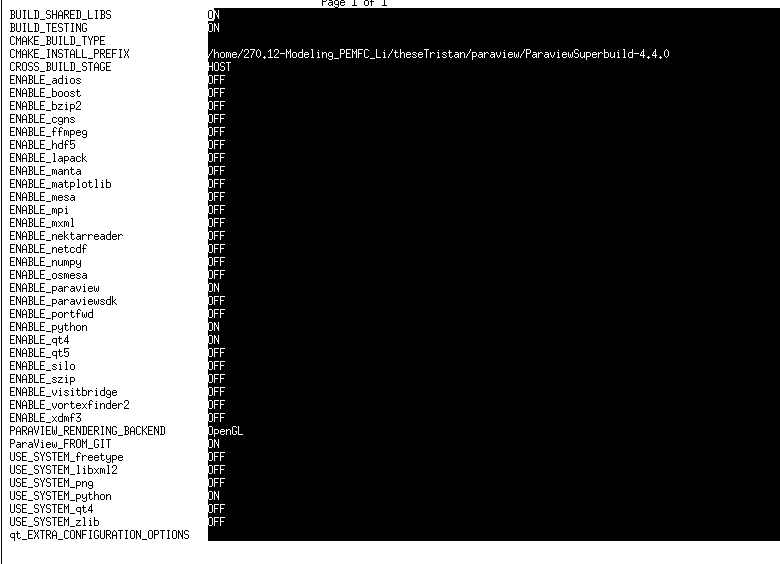
## Configurer Cmake

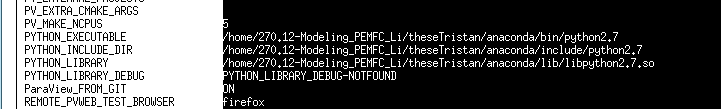
Images ci-dessous : cmake du paraviewSuperbuild-4.4.0

Pour récupérer ce superbuild :

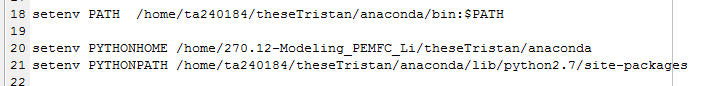
git clone git://paraview.org/ParaViewSuperbuild.git

git checkout v4.4.0





Une fois paraview installé, il faut indiquer les chemins suivants dans le fichier /home/.cshrc :



# Annexe : mode d’emploi windows sans superbuild

On présente ici une méthode pour compiller paraview sur windows sans superbuild. Notons que je n’ai pas réussi à aller au bout.

## Prérequis

Avoir installé la distribution de python anaconda

Télécharger et installer visual studio 2010 express (version gratuite) avec service pack 1:

<https://www.visualstudio.com/downloads/download-visual-studio-vs#DownloadFamilies_4>

puis télécharger et installer le service pack 1 :

<https://www.microsoft.com/fr-fr/download/details.aspx?id=23691>

Télécharger et installer qt :

<http://download.qt.io/archive/qt/4.8/4.8.2/>

Prendre [qt-win-opensource-4.8.2-vs2010.exe](http://download.qt.io/archive/qt/4.8/4.8.2/qt-win-opensource-4.8.2-vs2010.exe)

Télécharger et installer CMAKE

<http://www.cmake.org/files/v2.8/cmake-2.8.8-win32-x86.exe>

## Installer paraview

Le mode d’emploi est là : http://www.paraview.org/Wiki/ParaView:Build\_And\_Install

Deux étapes :

* Configurer la compilation avec Cmake. A cette étape on fait la liste des dépendances de paraview vis-à-vis des autres logiciels présents sur le PC.
* Compiler paraview avec Visual Studio.

### Etape Cmake

D’abord, indiquer le repertoire des sources Paraview et le repertoire build dans lequel seront compilées les sources (créer un répertoire vide différent du répertoire des sources).

La manipulation principale est d’indiquer que la version de python que Paraview doit utiliser est celle installée par vos soins, Anaconda.

Dans Cmake, cocher l’option PARAVIEW\_ENABLE\_PYTHON. Puis dans les options avancées, changer les variables PYTHON\_LIBRARY, PYTHON\_INCLUDE\_DIR, PYTHON\_EXECUTABLE :

* PYTHON\_LIBRARY = <anacondaPath>/anaconda/libs/python27.lib
* PYTHON\_INCLUDE\_DIR = <anacondaPath>/anaconda/include
* PYTHON\_EXECUTABLE = <anacondaPath>/anaconda/python.exe

Où <anacondaPath>/ est l’emplacement de votre anaconda. Dans mon cas c’est : C:\Users\ta240184\AppData\Local\Continuum

## Etape Visual Studio

Ajouter les dll QT à votre path : ouvrir la console windows en tapant *cmd* dans la fenêtre exécuter (*win+r*). Dans la console entrer : *set PATH=%PATH%;C\QT\4.8.2\bin*. Puis ouvrir VisualStudio depuis la console en tapant  *"<path\_to\_visual\_studio>\Common7\IDE\VCExpress.exe"* (c’est l’adresse de l’exécutable VisualStudio indiqué dans « tous les programmes » ).

Ouvrir avec visual studio le fichier ALL\_BUILD qui se trouve dans le dossier paraview build. Lancer la compilation (clic droit sur ALL\_BUILD dans VisualStudio) en mode debug.

# Annexe : dépendances linux si on n’utilise pas paraview-superbuild

## Installer Qt

ParaView uses Qt as its GUI library. Qt is required whenever the ParaView client is built. The LGPL of Qt can be found at <http://download.qt.io/archive/qt/4.8/4.8.2/>

For source code, use the latest stable version of qt-everywhere-opensource-src-VERSION.[tar.gz or zip or dmg]. If this gives you trouble, version 4.8.2 is known to work.

J’ai mis qt dans mon dossier perso/paraview. Extraction des sources dans paraview/tmp/qt-sources. Dossier build : paraview/tmp/qt-build. Répertoire d’installation : paraview/qt-4.8.2.

Procédure habituelle pour installer un logiciel depuis les sources : décompresser l’archive, puis ./configure –prefix myQtFolder (-qt-xcb éventuellement) , puis gmake, puis gmake install.

Plus d’infos : <http://doc.qt.io/qt-5/linux-building.html> , <http://doc.qt.io/qt-5/configure-options.html> <http://doc.qt.io/qt-5/linux-requirements.html>

## Installer Mesa

<http://www.paraview.org/Wiki/ParaView_And_Mesa_3D>

<http://www.mesa3d.org/install.html>