

DCMTK (MD版) 编译和安装+VS2015

原创 置顶 犹新 2017-10-11 09:51:20 10331 收藏 17

版权

分类专栏: C++医学图像处理 C++ DCMTK 文章标签: DCMTK3.6.2 visual studio cmake c++ windows

Visual Studio 2015+cmake编译安装MD版的DCMTK

下一章 DCMTK (MD版)、QT、VS2015编写浏览医学图像应用-- 读取模块
<https://blog.csdn.net/chaoenhu/article/details/101704350>

免编译, 直接 去中科院分子影像重点实验室Radiomics组官网下载编译好的文件

<http://www.radiomics.net.cn/platform/version/6>



一、编译环境

本文的编译机器是Windows10 64位系统, 使用VS2015和cmake3.7.2编译DCMTK源码为64位包 (运行时库为MD或MDd)。

注 (引用): /MT和/MTd表示采用多线程CRT库的静态lib版本。该选项会在编译时将运行时库以静态lib的形式完全嵌入。该选项生成的可执行文件运行时不需要运行时库dll的参加, 会获得轻微的性能提升, 但最终生成的二进制代码因链入庞大的运行时库实现而变得非常臃肿。当某项目以静态链接库的形式嵌入到多个项目, 则可能造成运行时库的内存管理有多份, 最终将导致致命的"Invalid Address specified to RtlValidateHeap"问题。另外托管C++和CLI中不再支持/MT和/MTd选项。

/MD和/MDd表示采用多线程CRT库的动态dll版本, 会使应用程序使用运行时库特定版本的多线程DLL。链接时将按照传统VC链接dll的方式将运行时库MSVCRxx.DLL的导入库MSVCRT.lib链接, 在运行时要求安装了相应版本的VC运行时库可再发行组件包 (当然把这些运行时库dll放在应用程序目录下也是可以的)。因/MD和/MDd方式不会将运行时库链接到可执行文件内部, 可有效减少可执行文件尺寸。当多项目以MD方式运作时, 其内部会采用同一个堆, 内存管理将被简化, 跨模块内存管理问题也能得到缓解。

结论: /MD和/MDd将是潮流所趋, /ML和/MLd方式请及时放弃, /MT和/MTd在非必要时最好也不要采用了。

二、下载文件

1) DCMTK包dcmtk-3.6.2.tar.gz下载

直接下载链接或下载最新版<http://www.dcmtk.org/dcmtk.php.en>

2) 支持库包dcmtk-3.6.2-win64-support_MD-msvc-14.0.zip下载

直接下载链接或下载最新版<http://www.dcmtk.org/dcmtk.php.en>

DCMTK 3.6.2 - source code and documentation (2017-07-14)

DCMTK can be downloaded as a gzip compressed tar archive or as a ZIP archive. The contents of the two archives are identical.



DCMTK 3.6.2 source code and documentation

DCMTK 3.6.2 source code and documentation



[dcmtk-3.6.2-win64-support_MD-msvc-14.0.zip](#)
11,292K

Pre-compiled libraries for Visual Studio 2015 (MSVC 14.0), 64 bit, with "MD" option

3) Cmake3.10 下载

直接下载，并解压，进入bin文件夹，点击cmake-gui.exe.

官网：<https://cmake.org/download/>

	cmake.exe	2017/1/13 9:56	应用程序	5,738 KB
	cmake-gui.exe	2017/1/13 9:56	应用程序	5,352 KB
	cmcldeps.exe	2017/1/13 9:55	应用程序	509 KB

三、用cmake构建VS2015工程

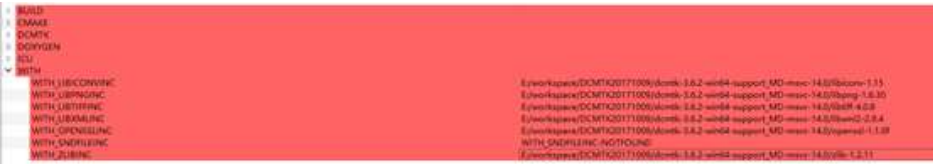
- 点击**cmake-gui.exe**打开cmake界面，关闭所有跟VS有关的程序。
- 在**where is the source code:**，选择dcmtk源代码的文件夹，即DCMTK3.6.2包解压路径：
E:/workspace/DCMTK20171009/dcmtk-3.6.2
- 在**where to build the binaries:**，选择你想存放build结果的文件夹（新建的空文件夹），
E:/workspace/DCMTK20171009/dcmtk-msvc15，勾选**Grouped**和**Advanced**



- 点击**Configure**，选择编译环境**Visual Studio 14 2015 Win64**，点击**Finish**，等待进度完成。



- 进度条完成后，修改参数，本文的如下图：



图中具体参数如下：

MD/MDd设置(重要) (防止产生错误三)

set "DCMTK_OVERWRITE _WIN32_COMPILER_FLAGS" to "OFF"

Xml support:

"DCMTK_WITH_XML" "ON"

"WITH_LIBXMLINC" 解压的支持库中xml的位置/dcmtd-3.6.2-win64-support_MD-msvc-14.0/libxml2-2.9.4

libpng support:

set "DCMTK_WITH_PNG" to "ON" and set "WITH_LIBPNGINC"

我的E:/workspace/DCMTK20171009/dcmk-3.6.2-win64-support_MD-msvc-14.0/libpng-1.6.30

libtiff support:

set "DCMTK_WITH_TIFF" to "ON" and set "WITH_LIBTIFFINC"

我的E:/workspace/DCMTK20171009/dcmk-3.6.2-win64-support_MD-msvc-14.0/libtiff-4.0.8

OpenSSL support:

set "DCMTK_WITH_OPENSSL" to "ON" and set "WITH_OPENSSLINC"

E:/workspace/DCMTK20171009/dcmk-3.6.2-win64-support_MD-msvc-14.0/openssl-1.1.0f

zlib support:

set "DCMTK_WITH_ZLIB" to "ON" and set "WITH_ZLIBINC"

我的E:/workspace/DCMTK20171009/dcmk-3.6.2-win64-support_MD-msvc-14.0/zlib-1.2.11

libiconv support:

set "DCMTK_WITH_ICONV" to "ON" and set "WITH_LIBICONVINC"

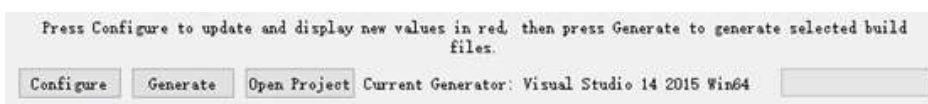
我的E:/workspace/DCMTK20171009/dcmk-3.6.2-win64-support_MD-msvc-14.0/libiconv-1.15

else

DCMTK_ENABLE_CHARSET_CONVERSION <disable>

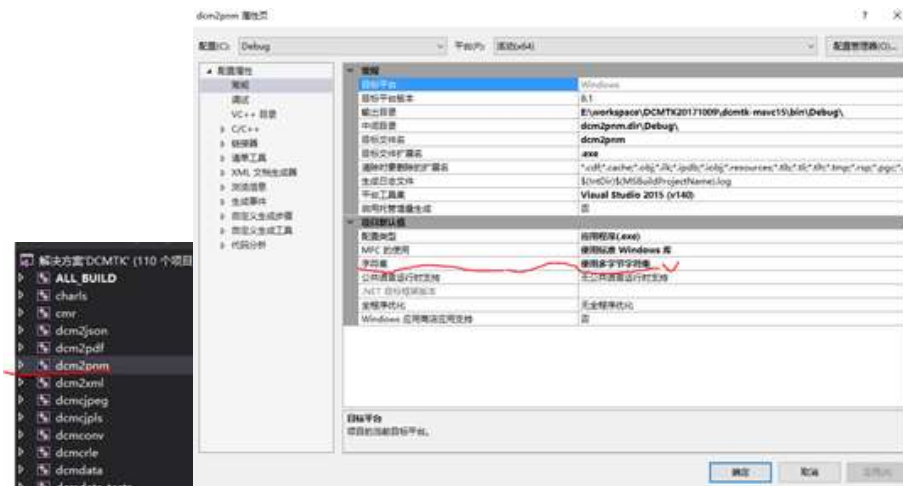
其他的支持库都关闭比如ICU。

- 修改参数完成后，点击"**configure**"，进度完成，点击**Generate**，完成后就在dcmk-msvc15文件夹下生产了VS工程文件，可以点击**Open Project** 打开。



四、编译和安装

1. 打开某个（如dcm2pnm）项目的属性页（右击属性），查看字符集（属性->配置属性->常规）是否为使用多字节字符集，和运行库（属性->配置属性->C/C++->代码生成）是否为MDd(Debug下)或MD (release下)，如果是证明生成项目成功。

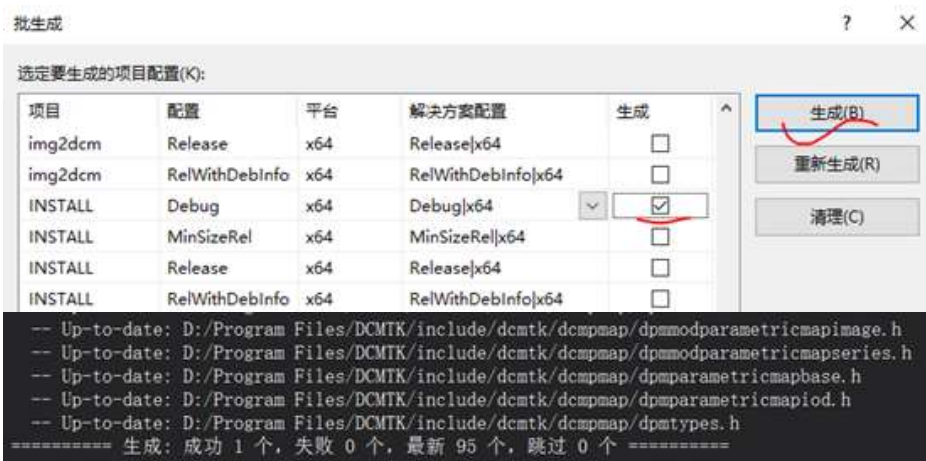


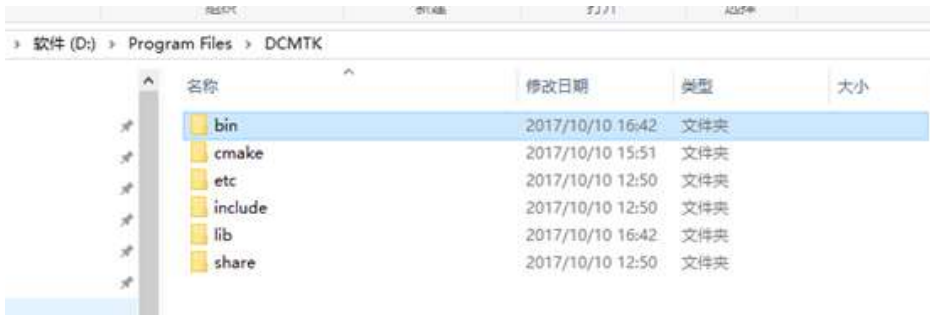
- 编译：生成->批生成->选中您需要配置方式的ALL_BUILD项目，点击右边的生成，下图是生成Debug版的，最后生成: 成功 95 个。



```
----- 已启动生成: 项目: ALL_BUILD, 配置: Debug x64 -----
Building Custom Rule E:\workspace\DCMTK20171009\dcmtk-3.6.2\CMakeLists.txt
CMake does not need to re-run because E:\workspace\DCMTK20171009\dcmtk-msvc15\CMakeFiles\generate.stamp is up-to-date.
***** 生成: 成功 95 个, 失败 0 个, 最新 0 个, 跳过 0 个 *****
```

- 为了防止产生错误一和错误二，请修改不同目录下的cmake_install.cmake文件，参考下面的错误一和错误二。
- 安装：生成->批生成->选中您需要配置方式的INSTALL项目，点击右边的生成，下图是安装Debug版的，默认安装在C:/ProgramFiles/DCMTK/，修改到其他盘，请参考错误一的解决方案二





五、错误集锦

错误一：生成INSTALL项目时，出现以下错误

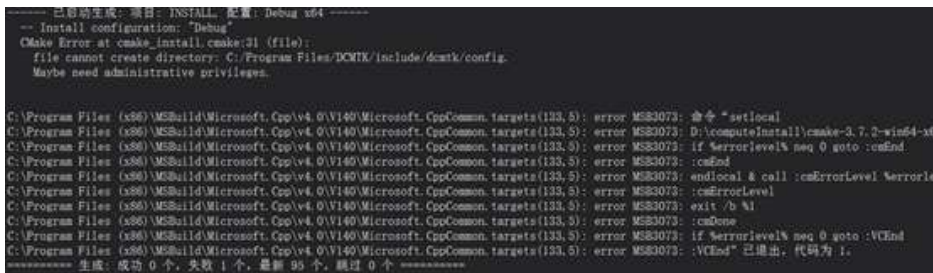
----- 已启动生成: 项目: INSTALL, 配置: Debug x64 -----

-- Install configuration: "Debug"

CMake Error at cmake_install.cmake:31 (file):

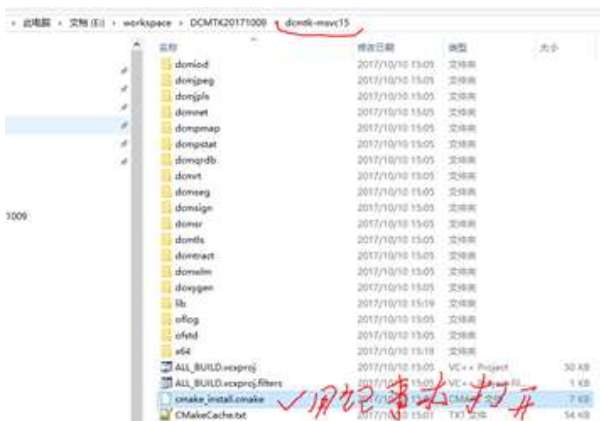
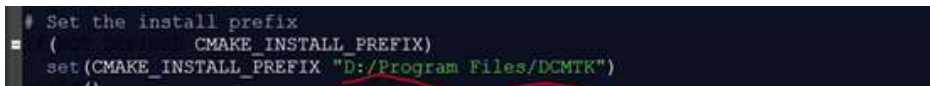
file cannot create directory: C:/Program Files/DCMTK/include/dcmtk/config.

Maybe need administrative privileges.



解决方案：

1. 以管理员方式启动VS2015,重新生成。
2. 修改解决方案路径下的cmake_install.cmake文件（用文本方式打开，我用的是NotePad++），把安装路径C:/Program Files/DCMTK/修改到D盘如下图,清理解决方案重新生成



错误二：生成INSTALL项目时，出现以下错误

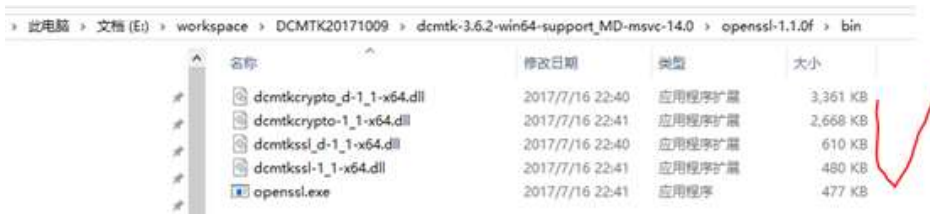
```

CMake Error at dcmnet/apps/cmake_install.cmake:139 (file):
  file INSTALL cannot find
  "E:/workspace/DCMTK20171009/dcmtk-3.6.2-win64-support_MD-msvc-14.0/openssl-1.1.0f/bin/dcmtkey.dll".
Call Stack (most recent call first):
  dcmnet/cmake_install.cmake:33 (include)
  cmake_install.cmake:100 (include)

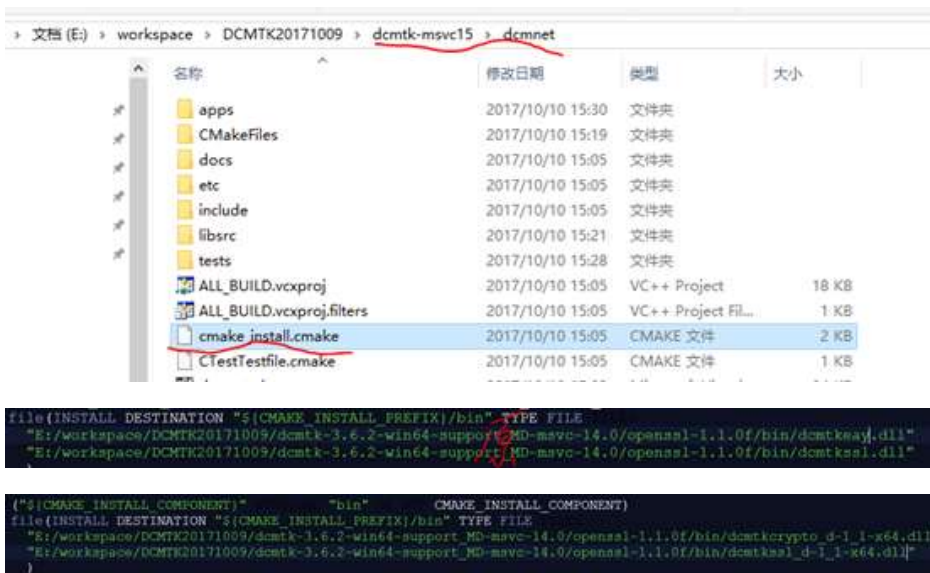
```

解决方案:

1. 把文件夹dcmtk-3.6.2-win64-support_MD-msvc-14.0\openssl-1.1.0f\bin下的dll复制到安装目录D:/Program Files/DCMTK/bin下。



2. 修改解决方案路径下的文件夹dcmnet/apps中的cmake_install.cmake文件, 把dcmtkey.dll修改为"dcmtcrypto_d-1_1-x64.dll", dcmtkssl.dll修改为" dcmtkssl_d-1_1-x64.dll"如下图



错误三运行时库 (MT MD) 冲突

导致各种错误, 如warning LNK4098: 默认库"MSVCRT"与其他库的使用冲突; 使用 /NODEFAULTLIB:library

如: LNK2001 无法解析的外部符号 __imp_*

解决方案: 运行时与支持库运行时匹配

六、测试例子

测试代码

```

#include"dcmtk/config/osconfig.h"

#include"dcmtk/dcndata/dctk.h"

#include<iostream>

```

```
using namespace std;

int main()
{
    DcmFileFormat fileformat;

    OFCondition oc = fileformat.loadFile("111.dcm");

    if (oc.good()) {
        OFString patientName;

        if (fileformat.getDataset()->findAndGetOFString(DCM_PatientName, patientName).good())
        {
            cout << "Patient Name:" << patientName.data() << endl;
        }
    }

    system("pause");

    return 0;
}
```

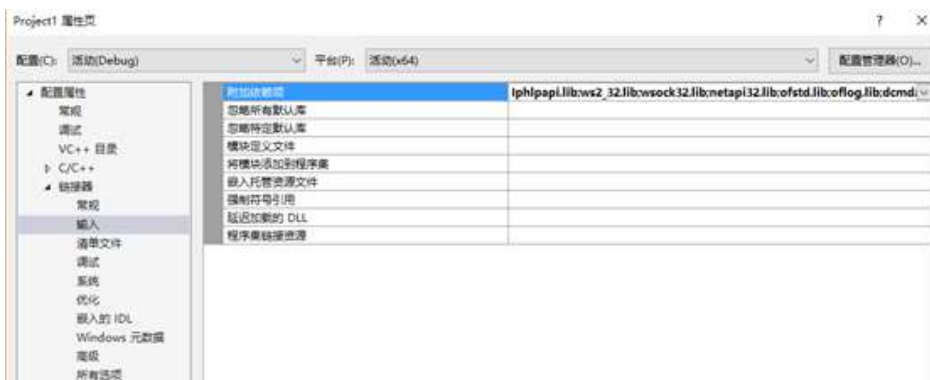
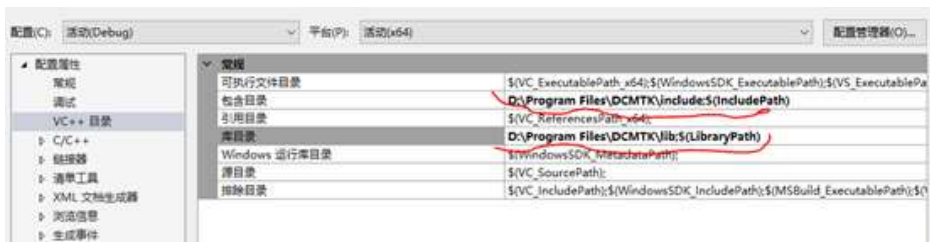


```
main.cpp 1 #include "dcmtdk/config/osconfig.h"
2 #include "dcmtdk/dcmdata/dctk.h"
3 #include <iostream>
4
5 using namespace std;
6 int main()
7 {
8     DcmFileFormat fileformat;
9     OFCondition oc = fileformat.loadFile("111.dcm");
10    if (oc.good()) {
11        OFString patientName;
12        if (fileformat.getDataset()->findAndGetOFString(DCM_PatientName, patientName).good())
13        {
14            cout << "Patient Name:" << patientName.data()<<endl;
15        }
16    }
17
18    system("pause");
19    return 0;
20 }
21 }
```

结果:

```
E:\workspace\test\Project1\64\Debug\Project1.exe
Patient Name:he ying hou
请按任意键继续. . .
```

配置环境





按顺序添加依赖项lphlpapi.lib; ws2_32.lib; wsock32.lib; netapi32.lib; ofstd.lib; oflog.lib; dcmtdata.lib; zlib_d.lib。

lphlpapi.lib; ws2_32.lib; wsock32.lib; netapi32.lib; 为系统库文件

ofstd.lib; oflog.lib; dcmtdata.lib; 为DCMTK生成的库文件

zlib_d.lib 支持包里的库文件,把它移动到DCMTK生成的库文件的位置

测试数据

测试数据 (111.dcm) 用下面链接下载

<http://www.casmi.science/index.php/s/mAn8XKYRd04FVIH>

说明

本文档仅用于交流学习, 编写于2017年10月10日星期二18: 06, 转载请引用, 有问题发邮件到 mitk@fingerepass.net.cn!

本文的PDF版下载 <http://download.csdn.net/download/chaoenhu/10015394>

免编译, 直接 去中科院分子影像重点实验室Radiomics组官网下载编译好的文件

<http://www.radiomics.net.cn/platform/version/6>

下一章 DCMTK (MD版) 、QT、VS2015编写浏览医学图像应用-- 读取模块
<https://blog.csdn.net/chaoenhu/article/details/101704350>

进一步学习

目录:

开始 《DCMTK (MD版) 编译和安装+VS2015》

第一章 《DCMTK (MD版) 、QT、 VS2015编写Dicom序列浏览应用程序-新建项目，配置环境》

第二章 《第二章 基于QT和DCMTK的Dicom 图像浏览器---界面设计》

第三章 《 基于QT和DCMTK的Dicom 图像浏览器---单个Dicom图像读取类》

第四章 《基于QT和DCMTK的Dicom 图像浏览器---检查文件夹下Dicom序列个数》

第五章 《基于QT和DCMTK的Dicom 图像浏览器---Dicom图像序列类》

第六章 《基于QT和DCMTK的Dicom 图像浏览器---Dicom视图类》

第七章 《基于QT和DCMTK的Dicom 图像浏览器---收尾》