链滴

AWS CPP SDK 编译链接

作者：xiaojuna8
原文链接：https://ld246.com/article/1521768729152
来源网站：链滴

AWS scheme

modify date:2018.03.22

[TOC]

AWS 库编译

To use the AWS SDK for C++, you need:

- Visual Studio 2013 or later

 Note:Visual Studio 2013 doesn't provide default move constructors and operators. Later v rsions of Visual Studio provide a standards-compliant compiler.

- or GNU Compiler Collection (GCC) **4.9** or later
- or Clang 3.3 or later
- A minimum of 4 GB of RAM

(-dev packages) for libcurl, libopenssl, libuuid, zlib

cmake version > 3.0

unix编译依赖

- libz
- curl
- openssl
- libuuid

原文链接：AWS CPP SDK 编译链接

Ubuntu 安装开发库sudo apt-get install libcurl4-openssl-dev libssl-dev uuid-dev zlib1g-dev libulse-dev

openssl
uuid
zlib
curl

centos

安装新版本gcc和gdb

```
# 1. Install a package with repository for your system:
# On CentOS, install package centos-release-scl available in CentOS repository:
$ sudo yum install centos-release-scl

# On RHEL, enable RHSCL repository for you system:
$ sudo yum-config-manager --enable rhel-server-rhscl-7-rpms

# 2. Install the collection:
$ sudo yum install devtoolset-7

# 3. Start using software collections:
$ scl enable devtoolset-7 bash
```

安装cmake

```
yum install llvm-toolset-7
```

ccmake-命令行界面，交互模式。

安装libz，curl，openssl和libuuid的devel版

只编译部分模块

通过cmake配置生成模块，cmake /mnt/hd03.d/xdu.d/aws/source/aws-sdk-cpp -DBUILD_ONL="s3"

- BUILD_SHARED_LIBS，是否生成动态库。ON动态库，OFF静态库
- CMAKE_BUILD_TYPE，生成类型,Relaese,Debug -DCMAKE_BUILD_TYPE=Release

```
cmake /mnt/hd03.d/xdu.d/aws/source/aws-sdk-cpp  -DBUILD_ONLY="core;s3;transfer"  -DBULD_SHARED_LIBS="OFF" -DCMAKE_BUILD_TYPE=Release
```

动态库

```
cmake /mnt/hd03.d/xdu.d/aws/source/aws-sdk-cpp  -DBUILD_ONLY="core;s3;transfer"  -DBULD_SHARED_LIBS="ON" -DCMAKE_BUILD_TYPE=Release -DCMAKE_CXX_FLAGS=-fPIC -DCMKE_C_FLAGS=-fPIC
```

链接

原文链接：AWS CPP SDK 编译链接

动态链接需要定义USE_IMPORT_EXPORT

If you dynamically link to the SDK you will need to define the USE_IMPORT_EXPORT symbol f r all build targets using the SDK

GCC: what are the --start-group and --end-group command line options?

It is for resolving circular dependences between several libraries

Impossible to link static libraries with circular dependencies

在aws的静态库中，有**循环引用**问题。

In a project we have multiple static libraries (build using cdt), two of which have circular depe dencies to each other, that link together into an executable (also built using cdt).

With the "GCC C Linker" selected it's impossible to build that file. The linker complains about missing references when the static libraries are simply added in the "C/(C++ Build -> Settings -> GCC C Linker -> Libraries" tab, because references of the library that appears last on the g nerated command line has undefined references.

The solution would be to add "--start-group -lX -lY --end-group" to the linker command line fter the list of the object files. But that's simply not possible. The additional linker options (both he "Linker flags" and "Other options (-Xlinker [option])" fields) that can be specified on the " CC C Linker -> Miscellaneous" tab are added to the command line before the object, which h s essentially no effect.

N.B. In our project I have installed a (dirty) workaround: I have added the linker options (-Wl,- start-group,-lX,-lY,--end-group) to the "Other objects" field (and added a dummy make rule i makefile.init so make won't complain about it) to force CDT to put it after the list of objects. I works but is really ugly.

使用GCC可将链接参数传递给链接器使用-Wl,--start-group和-Wl,--end-group

```
-Wl,--start-group  -laws-cpp-sdk-core -laws-cpp-sdk-s3 -laws-cpp-sdk-transfer    -Wl,--end-g
oup
```

```
//linux makefile
CC = g++
CCFLAGS = -DNDEBUG  -std=c++11 -O0
INCLUDE := -I../
LDFLAGS := -L../aws/bin/linux/static -Wl,-rpath ./ -Wl,-rpath ../aws/bin/linux/dynamic
LIBS  := -lcurl -pthread -lcrypto -Wl,--start-group -laws-cpp-sdk-core -laws-cpp-sdk-s3 -laws-
pp-sdk-transfer -Wl,--end-group

make: main
	rm *.o
	@echo success
main:
	$(CC) $(CCFLAGS) $(INCLUDE) -c -o main.o main.cpp
	$(CC) $(CCFLAGS) $(INCLUDE) $(LDFLAGS) -o transfer main.o $(LIBS)
```

Mac下不需要-pthread -lcrypto

查看动态库依赖

readelf -d libbar.so

```
$ readelf -d ../aws/bin/linux/dynamic/libaws-cpp-sdk-transfer.so

Dynamic section at offset 0x29c38 contains 36 entries:
  Tag        Type                         Name/Value
 0x0000000000000001 (NEEDED)             Shared library: [libaws-cpp-sdk-s3.so]
 0x0000000000000001 (NEEDED)             Shared library: [libaws-cpp-sdk-core.so]
 0x0000000000000001 (NEEDED)             Shared library: [libpthread.so.0]
 0x0000000000000001 (NEEDED)             Shared library: [libcurl.so.4]
 0x0000000000000001 (NEEDED)             Shared library: [libssl.so.10]
 0x0000000000000001 (NEEDED)             Shared library: [libcrypto.so.10]
 0x0000000000000001 (NEEDED)             Shared library: [libz.so.1]
 0x0000000000000001 (NEEDED)             Shared library: [libstdc++.so.6]
 0x0000000000000001 (NEEDED)             Shared library: [libm.so.6]
 0x0000000000000001 (NEEDED)             Shared library: [libgcc_s.so.1]
 0x0000000000000001 (NEEDED)             Shared library: [libc.so.6]
 0x000000000000000e (SONAME)              Library soname: [libaws-cpp-sdk-transfer.so]
```

从静态库合并为动态库

```
gcc -shared -o libmerge.so -Wl,--whole-archive libaws-cpp-sdk-core.a libaws-cpp-sdk-s3.a li
aws-cpp-sdk-transfer.a -Wl,--no-whole-archive
```

windows 编译链接aws lib

使用cmake生成工程，到生成库目录下，此处打开PowerShell然后运行指令

```
& "E:\Software Install\cmake-3.11.0-rc3-win64-x64\cmake-3.11.0-rc3-win64-x64\bin\cmake" E
\libs\aws-sdk-cpp-1.4.11 -DBUILD_ONLY="transfer" -DBUILD_SHARED_LIBS=ON -DSTATIC_L
NKING=1 -DTARGET=WINDOWS -G "Visual Studio 15 2017"
```

windows 静态链接aws static lib 需要依赖于几个系统静态库

```
winhttp.lib
wininet.lib
bcrypt.lib
userenv.lib
Version.lib
```

静态链接出现GetObjectA/GetObjectW错误，由于wingdi.h中定义了

```
//wingdi.h
#ifdef UNICODE
#define GetObject  GetObjectW
#else
#define GetObject  GetObjectA
#endif // !UNICODE
```

在S3Client.h被包含之前取消**GetObject**定义

```
//on windows visual studio
//undef GetObject to avoid marco conflict
```

```
#if defined _WIN32 && defined GetObject
#undef GetObject
#endif
```