官网的方法Nginx编译方法，思路是一致的，只是有几个细节地方需要调整。

需要注意以下问题：

1. Win10 操作系统下，貌似在编译OpenSSL的时候有问题。
2. 使用sm for Windows 作为微软的masm 的替代。

一：下载以下工具，并依次安装。

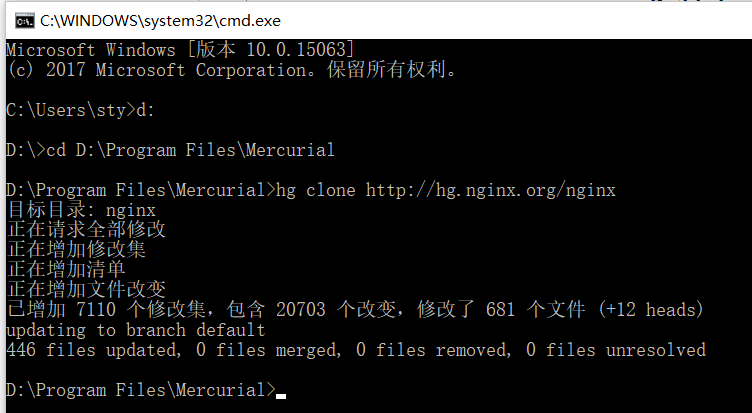
1. 下载 [VS2010](http://download.microsoft.com/download/E/0/4/E0427BB8-8490-4C7F-A05B-AFEA0FC3EA80/X16-60997VS2010UltimTrialCHS.iso)+ [VS2010 SP1](http://221.226.179.163/VS2010SP1dvd1.iso?fid=mGZF6ApSkJMmHHLTO9R-JWIazEEAIOleAAAAAGY1*sp5G0FqwT9enODmpXu5yeey&mid=666&threshold=150&tid=509A7D23589523AFD4D6B5DDEAE5E170&srcid=119&verno=1) （C++编译环境）VS2015也可以。
2. MSYS-1.0.11.exe
3. ActivePerl-5.24.2.2403-MSWin32-x64-403863.exe
4. mercurial-4.3.1-x64.msi
5. nasm-2.12.02rc9-installer-x64.exe
6. sed-4.2.1-setup.exe
7. 下载配套的所需库代码[PCRE](http://www.pcre.org/), [zlib](http://zlib.net/) and [OpenSSL](http://www.openssl.org/) libraries sources.

并且需要确保以上工具加入到了环境变量。

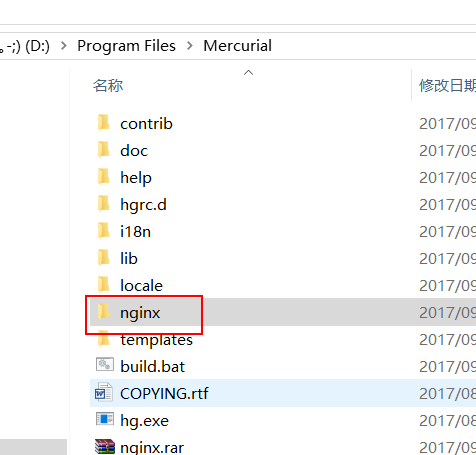
二、获取Nginx源代码

打开CMD命令，切换到Mercurial安装根目录，执行命令：

hg clone http://hg.nginx.org/nginx



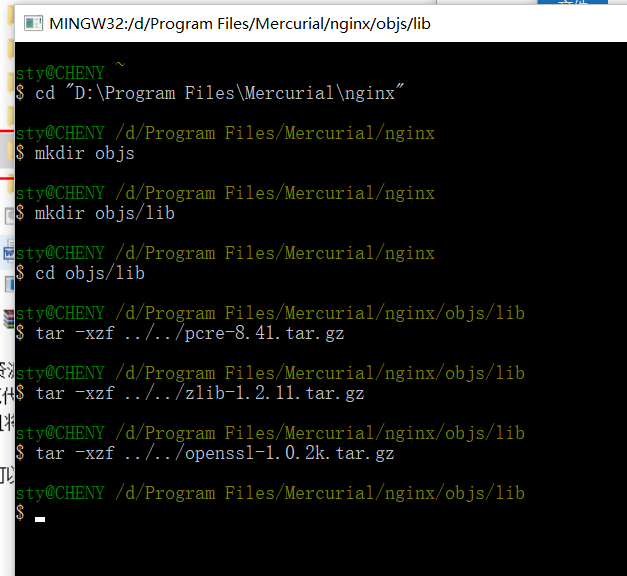
下载成功后会在Mercurial安装目录下，创建nginx源代码文件夹。



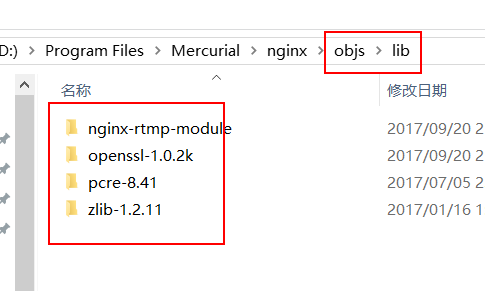
三、添加依赖资源包

在nginx源代码分别创建objs\lib 文件夹，此处如此命名只是约定的习惯，可以不是这两个名称。并且将下载好的nginx-rtmp-module-master、openssl、pcre、zlib分别解压在lib文件夹下面。

以上动作可以按官网教程一样，使用MSYS使用CUI的方式。



结果如下图：



四、生成makefile文件，用于C++编译器

继续使用MSYS工具，返回到nginx源代码根目录，执行以下脚本：

auto/configure --with-cc=cl --builddir=objs --prefix= \

--conf-path=conf/nginx.conf --pid-path=logs/nginx.pid \

--http-log-path=logs/access.log --error-log-path=logs/error.log \

--sbin-path=nginx.exe --http-client-body-temp-path=temp/client\_body\_temp \

--http-proxy-temp-path=temp/proxy\_temp \

--http-fastcgi-temp-path=temp/fastcgi\_temp \

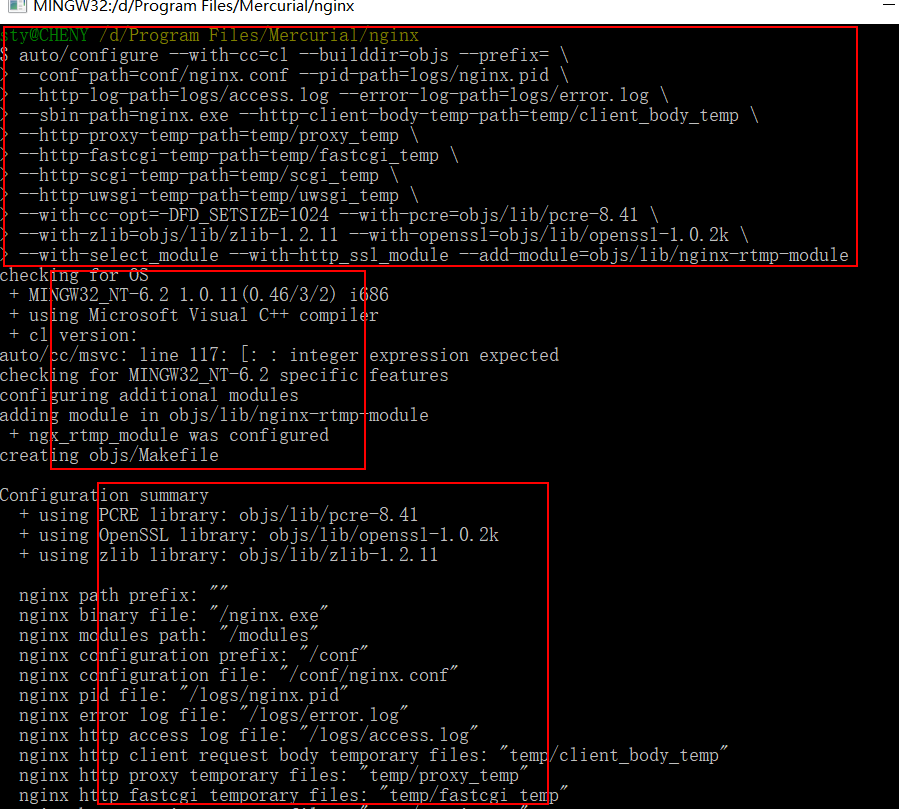
--http-scgi-temp-path=temp/scgi\_temp \

--http-uwsgi-temp-path=temp/uwsgi\_temp \

--with-cc-opt=-DFD\_SETSIZE=1024 --with-pcre=objs/lib/pcre-8.41 \

--with-zlib=objs/lib/zlib-1.2.11 --with-openssl=objs/lib/openssl-1.0.2k \

--with-select\_module --with-http\_ssl\_module --add-module=objs/lib/nginx-rtmp-module



有如上三部分输出

关于上述命令行报错： auto/cc/msvc: line 117: [: : integer expression expected

原因是编译时用的是 Visual Studio 2015，在脚本文件 nginx\auto\cc\msvc 中变量 NGX\_MSVC\_VER 的值计算有误

具体代码是：

NGX\_MSVC\_VER=`$NGX\_WINE $CC 2>&1 | grep 'Compiler Version' 2>&1 \

| sed -e 's/^.\* Version \(.\*\)/\1/'`

echo " + cl version: $NGX\_MSVC\_VER"

# MSVC 2005 supports C99 variadic macros

if [ "$ngx\_msvc\_ver" -ge 14 ]; then

have=NGX\_HAVE\_C99\_VARIADIC\_MACROS . auto/have

fi

用到了变量NGX\_MSVC\_VER, 用于比较这个值是否大于等于14, 而提示报错的也就是这一句话

由于对Configure的写法不是很了解(初步猜想是因为我用得是中文版本的原因), 可以简单的根据自己的Visual Studio的版本, 直接设定这个值, 参考

# MSVC 6.0 SP2 cl 12.00

# MSVC Toolkit 2003 (7.1) cl 13.10

# MSVC 2005 Express Edition SP1 (8.0) cl 14.00

# MSVC 2008 Express Edition (9.0) cl 15.00

# MSVC 2010 (10.0) cl 16.00

# MSVC 2015 (14.0) cl 19.00

我编译时用的是 Visual Studio 2015, 于是将代码改为：

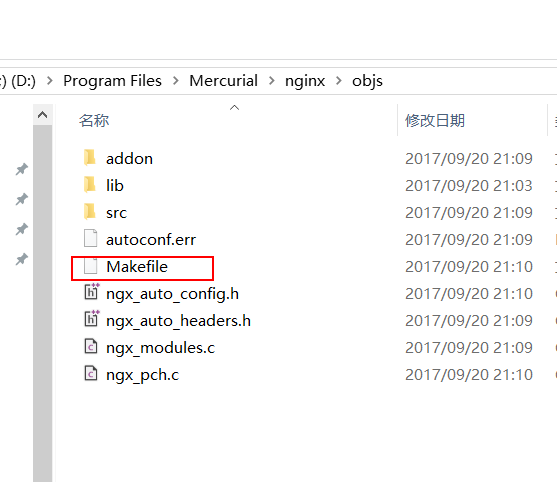
NGX\_MSVC\_VER=`$NGX\_WINE $CC 2>&1 | grep 'Compiler Version' 2>&1 \

| sed -e 's/^.\* Version \(.\*\)/\1/'`

NGX\_MSVC\_VER=19.00

echo " + cl version: $NGX\_MSVC\_VER"

然后重新在msys中执行 auto/Configure … 成功执行没有报错



最终结果，会在objs目录下生成Makefile文件，以及其他所依赖的代码。

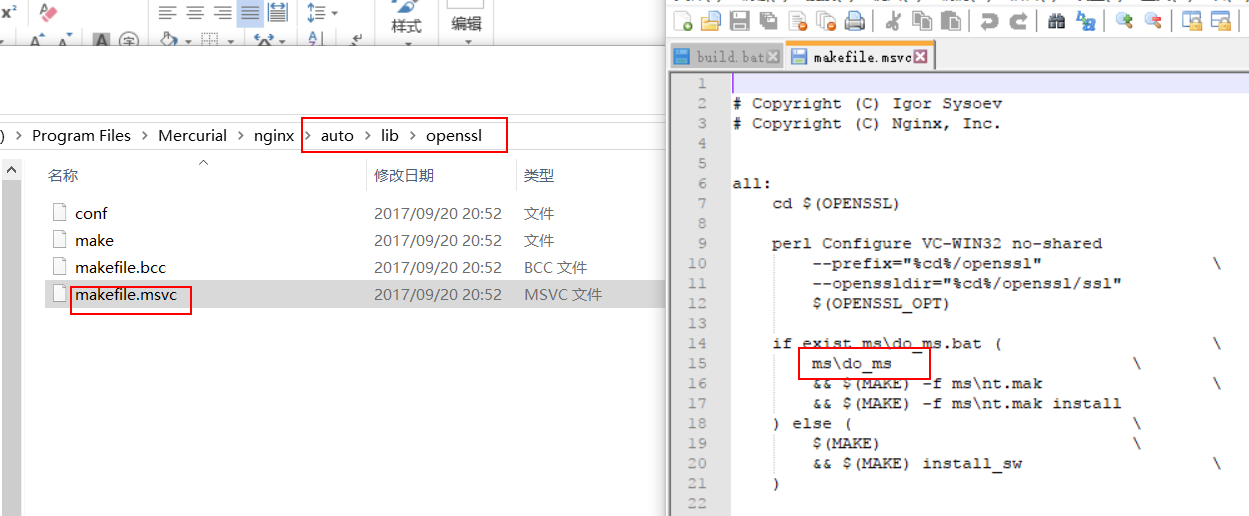
五、OpenSSL的坑，

需要修改 nginx\auto\lib\openssl\makefile.msvc 文件。

将ms\do\_ms 改为ms\do\_nasm。

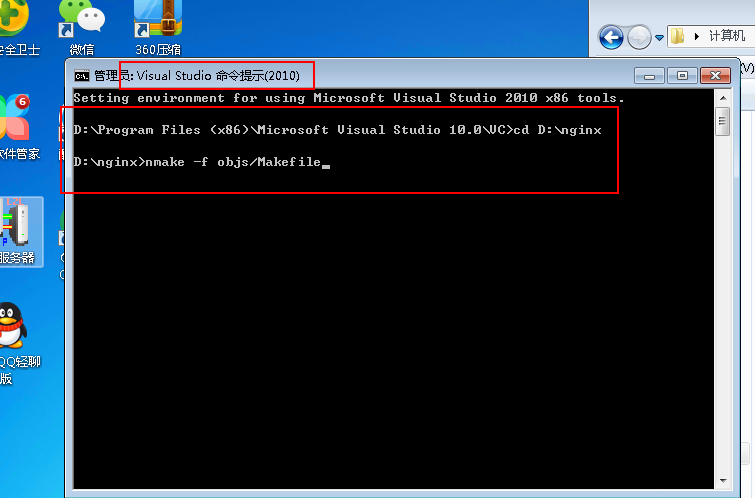
具体原因参看篇文章。

http://blog.csdn.net/felixking/article/details/51981794?utm\_source=itdadao&utm\_medium=referral



六、编译Nginx源码

使用VS2010命令工具，切换到Nginx根目录，执行nmake -f objs/Makefile 命令。

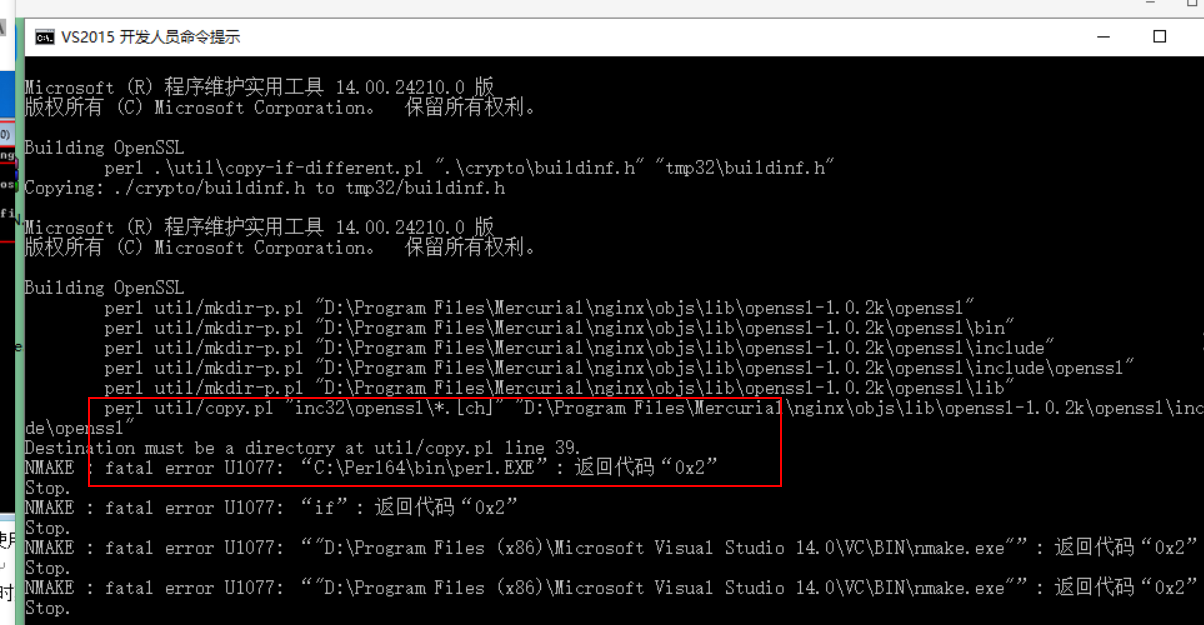


【我一开始在Wi10上使用VS2015的开发者工具，结果依然有问题，具体原因没有分析出来，核心错误如下：

perl util/copy.pl "inc32\openssl\\*.[ch]" "D:\Program Files\Mercurial\nginx\objs\lib\openssl-1.0.2k\openssl\include\openssl"

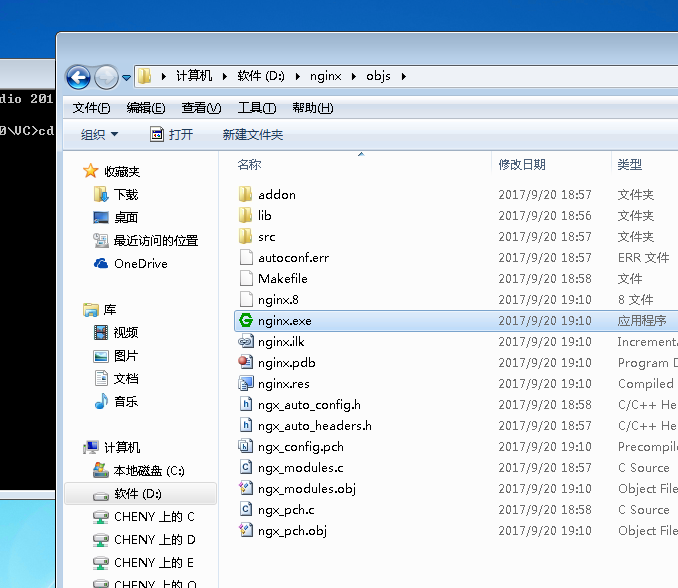
Destination must be a directory at util/copy.pl line 39.

推测是Win10版本都有这个异常】



最后改为Win7+VS2010。

不算漫长的等待之后，激动人心的时刻就到来了。编译成功。



注：使用VS2015命令工具。

# 修改错误级别为3级，d:\git\nginx\auto\cc\msvc的83行，把 -W4 修改为 -W3

# 否则，编译时(nmake -f objs/Makefile)，会出现如下错误：

# build/lib/nginx-rtmp-module/ngx\_rtmp\_core\_module.c(611): error C2220: 警告被视为错误 - 没有生成“object”文件

# build/lib/nginx-rtmp-module/ngx\_rtmp\_core\_module.c(611): warning C4456: “sa”的声明隐藏了上一个本地声明

# build/lib/nginx-rtmp-module/ngx\_rtmp\_core\_module.c(611): note: 要简化迁移，请考虑暂时对用于生成且不引发警告的编译器版本 使用 /Wv:18 标记

# build/lib/nginx-rtmp-module/ngx\_rtmp\_core\_module.c(506): note: 参见“sa”的声明

# NMAKE : fatal error U1077: “"C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\BIN\cl.EXE"”: 返回代码“0x2”

# Stop.

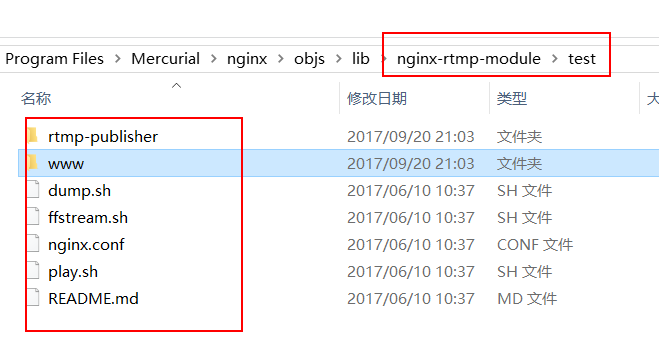
d:\git\nginx\auto\cc\msvc

CFLAGS="$CFLAGS -W4" ==> CFLAGS="$CFLAGS -W3"

然后 切换到Nginx根目录，执行nmake -f objs/Makefile 命令，在win10+VS2015的环境下编译成功。

七、创建运行基础文件夹

在nginx.exe所在文件夹下创建创建conf,logs,temp,tmp目录，可将nginx-rtmp-module\test下的nginx.conf文件复制到conf目录下，启动nginx，基础的RTMP流媒体服务搭建完成。



<http://nginx.org/en/docs/howto_build_on_win32.html>

## Building nginx on the Win32 platform with Visual C

|  |
| --- |
| [Build steps](http://nginx.org/en/docs/howto_build_on_win32.html#build_steps) [See also](http://nginx.org/en/docs/howto_build_on_win32.html#see_also) |

#### Prerequisites

To build nginx on the Microsoft Win32® platform you need:

* Microsoft Visual C compiler. Microsoft Visual Studio® 8 and 10 are known to work.
* [MSYS](http://www.mingw.org/wiki/MSYS).
* Perl, if you want to build OpenSSL® and nginx with SSL support. For example [ActivePerl](http://www.activestate.com/activeperl) or [Strawberry Perl](http://strawberryperl.com/).
* [Mercurial](https://www.mercurial-scm.org/) client.
* [PCRE](http://www.pcre.org/), [zlib](http://zlib.net/) and [OpenSSL](http://www.openssl.org/) libraries sources.

#### Build steps

Ensure that paths to Perl, Mercurial and MSYS bin directories are added to PATH environment variable before you start build. To set Visual C environment run vcvarsall.bat script from Visual C directory.

To build nginx:

* Start MSYS bash.
* Check out nginx sources from the hg.nginx.org repository. For example:
* hg clone http://hg.nginx.org/nginx
* Create a build and lib directories, and unpack zlib, PCRE and OpenSSL libraries sources into lib directory:
* mkdir objs
* mkdir objs/lib
* cd objs/lib
* tar -xzf ../../pcre-8.41.tar.gz
* tar -xzf ../../zlib-1.2.11.tar.gz
* tar -xzf ../../openssl-1.0.2k.tar.gz
* Run configure script:
* auto/configure --with-cc=cl --builddir=objs --prefix= \
* --conf-path=conf/nginx.conf --pid-path=logs/nginx.pid \
* --http-log-path=logs/access.log --error-log-path=logs/error.log \
* --sbin-path=nginx.exe --http-client-body-temp-path=temp/client\_body\_temp \
* --http-proxy-temp-path=temp/proxy\_temp \
* --http-fastcgi-temp-path=temp/fastcgi\_temp \
* --with-cc-opt=-DFD\_SETSIZE=1024 --with-pcre=objs/lib/pcre-8.41 \
* --with-zlib=objs/lib/zlib-1.2.11 --with-openssl=objs/lib/openssl-1.0.2k \
* --with-select\_module --with-http\_ssl\_module
* Run make:
* nmake -f objs/Makefile

#### See also