1. 下载下来解压缩，按正常安装过程进行编译、安装。

# tar zxvf gsozp\_2.7.12.tar.gz

# cd gsoap\_2.7.12

# ./configure –-prefix=/usr/local/gSOAP（指定安装路径）

# make

# make install

1. 安装编译工具：

$sudo apt-get install build-essential

为了成功编译gSOAP，您需要安装GTK+的开发文件和GLib库(libraries)。

$sudo apt-get install libgtk2.0-dev libglib2.0-dev

安装Checkinstall以便管理您系统中直接由源代码编译安装的软件。

$sudo apt-get install checkinstall

1. 没有安装：yacc(Yet Another Compiler Compiler)，是Unix/Linux上一个用来生成编译器的编译器（编译器代码生成器）。

sudo apt-get install flex bison

1. make[5]: \*\*\* [wsdl2h] 错误 1 //是缺少openssl的相应的头文件

sudo apt-get install openssl  
为了安装OpenSSL通常的库文件，首先使用以下命令来确定在Ubuntu系统中可获得的库文件的应用版本：  
 sudo apt-cache search libssl | grep SSL   
 sudo apt-get install libssl-dev

1. 如果不知道被安装在哪个目录，可以用命令查询：find / -name wsdl2h

用wsdl2h或者soapcpp2查看gSOAP是否已经安装成功:

# wsdl ---> wsdl    wsdl2   wsdl2h

# soap ---> soapcpp2  soapsuds

1. wsdl2h -o onvif.h -c -s -t ./typemap.dat <http://www.onvif.org/onvif/ver10/device/wsdl/devicemgmt.wsdl> <http://www.onvif.org/onvif/ver10/media/wsdl/media.wsdl> <http://www.onvif.org/onvif/ver10/event/wsdl/event.wsdl> <http://www.onvif.org/onvif/ver10/display.wsdl> <http://www.onvif.org/onvif/ver10/deviceio.wsdl> <http://www.onvif.org/onvif/ver20/imaging/wsdl/imaging.wsdl> <http://www.onvif.org/onvif/ver20/ptz/wsdl/ptz.wsdl> <http://www.onvif.org/onvif/ver10/receiver.wsdl> <http://www.onvif.org/onvif/ver10/recording.wsdl> <http://www.onvif.org/onvif/ver10/search.wsdl> <http://www.onvif.org/onvif/ver10/network/wsdl/remotediscovery.wsdl> <http://www.onvif.org/onvif/ver10/replay.wsdl> <http://www.onvif.org/onvif/ver20/analytics/wsdl/analytics.wsdl> <http://www.onvif.org/onvif/ver10/analyticsdevice.wsdl> <http://www.onvif.org/ver10/actionengine.wsdl> <http://www.onvif.org/ver10/pacs/accesscontrol.wsdl> <http://www.onvif.org/ver10/pacs/doorcontrol.wsdl>
2. soapcpp2 -c onvif.h -x -I import路径 -I gsoap路径

soapcpp2 -c onvif.h -x -I /home/kk/test/gsoap-2.8/gsoap/import -I /home/kk/test/gsoap-2.8/gsoap/

1. wsdl2h常用选项

-o 文件名，指定输出头文件

-n 名空间前缀 代替默认的ns

-c 产生纯C代码，否则是C++代码

-s 不要使用STL代码

-t 文件名，指定type map文件，默认为typemap.dat

-e 禁止为enum成员加上名空间前缀

-C 仅生成客户端代码

-S 仅生成服务器端代码

-L 不要产生soapClientLib.c和soapServerLib.c文件

-c 产生纯C代码，否则是C++代码(与头文件有关)

-I 指定import路径（见上文）

-x 不要产生XML示例文件

-i 生成C++封装(代理)，客户端为xxxxProxy.h(.cpp)，服务器端为xxxxService.h(.cpp)。

1. 其它