gSoap文档(翻译)

# 快速用户指南

原则上，如果gSoap客户端-服务端程序使用同样的方式构建，并且不与其他SOAP实现进行通信，那么SOAP客户端和SOAP WEB服务可以使用gSoap的soapcpp2工具编译为C/C++代码，不需要详细了解gSoap协议。

本节旨在说明与其他SOAP实现(如Apache Axis，SOAP::Lite，.NET)进行连接和互操作的gSoap服务端和客户端实现。这需要了解SOAP和WSDL协议的一些细节。

## 如何构建SOAP/XML客户端

通常，SOAP客户端应用程序的实现需要客户端调用的每个服务操作的代理。代理的主要责任是编码参数、将带有参数的请求通过连接发送给指定的SOAP服务、等待响应，并在响应到达时解码参数。客户端应用程序调用服务操作，就像调用本地函数一样。使用C/C++手动编写代理任务相当乏味，尤其是输入输出参数包含复杂的数据结构，例如对象、结构、容器、数组、指针等。幸运的是，gSoap的WSDL解析器工具(wsdl2h)和代理/框架、序列化代码生成器工具(soapcpp2)可以自动开发SOAP/XML WEB服务和客户端程序。

可以使用wsdl2h将WSDL文件解析为.h头文件：

* wsdl2h –o calc.h <http://www.genivia.com/calc.wsdl>

生成的头文件包含用于操作服务的数据类型和消息，以及与WSDL和XML模式相关的元信息。

在calc.h文件中，SOAP服务方法被指定为函数原型。例如，两个浮点数求和的add函数：

int ns2\_\_add(double a, double b, double& result);

为防止名称冲突，ns2\_add函数使用XML名称空间前缀将其与其他命名空间的操作进行区分。将XML命名空间前缀添加到操作、类型、结构体和类成员的惯例被gSoap工具普遍使用，并由wsdl2h自动创建。但在将现有C/C++类型和操作转换为web服务时，不是必需的。因此，可以从头文件中定义的类型名称中省略前缀符号，以运行soapcpp2来创建交换现有数据类型的客户端和服务端。

这些函数原型由soapcpp2工具转换为远程调用的代理和存根。

soapStub.h 注释了输入定义的副本

soapH.h 序列化器

soapC.cpp 序列化器

soapClient.cpp 客户端调用存根

### 示例

### XML命名空间注意事项

### 示例

### 如何生成C++客户端代理类

### XSD类型编码注意事项

### 示例

### 如何更改响应元素名称

### 示例

### 如何指定多个输出参数

### 示例

### 如何指定struct/class类型的输出参数

### 示例

### 如何指定匿名参数名称

### 如何指定没有输入参数的方法

### 如何指定没有输出参数的方法

## 如何构建SOAP/XML WEB服务

### 示例

### MSVC++构建

### 如何创建独立服务器

### 如何创建多线程独立服务器

### 如何向服务的方法传递应用程序数据

### Web服务实现方面

### 如何生成C++服务对象类

### 如何链接C++服务类在同一端口接收消息

### 如何生成WSDL服务描述

### 示例

### 如何在服务中使用客户端功能

## 异步单向消息传递

## 实现通过HTTP的异步单向消息传递

## 如何使用SOAP的序列化和反序列化程序来保存和加载使用XML数据绑定的应用数据

### 使用wsdl2h讲XML模型映射为C/C++代码

### 使用soapcpp2讲C/C++代码映射到XML模型

### 将C/C++数据序列化为XML

### 从XML中反序列化C/C++数据

### 示例

### 序列化和反序列化类实例到流

### 如何为省略的数据指定默认值