

TAO工作室Stone Jiang的专栏

[欢迎 huapuyu6!](#) [退出](#) [我的博客](#) [配置](#) [写文章](#) [文章管理](#) [博客首页](#)

博客主站 <http://www.tao-studio.net/blog>

□□

空间

博客

好友

相册

留言

用户操作

[\[留言\]](#) [\[发消息\]](#) [\[加为好友\]](#)

订阅我的博客

78 位读者
POWERED BY FEEDSKY



+ 订阅到 鲜果

+ 订阅到 Google

+ 订阅到 抓虾

FocusOnACE的公告

欢迎大家与我交流

文章分类



1.C++&OOP



2.1 ACE Newbies' G

uide



2.ACE



3. OpenDDS



4.TAO



5.UML,UP,Design Pa

ttern



6 VxWorks



CORBA Programming with TAO - 5.Hello World (简单实例) [收藏](#)

CORBA Programming with TAO - 5.Hello World (简单实例)

摘要:

前面对CORBA编程的基础知识进行了简单介绍,本篇中,我们将实例讲解如何利用TAO中进行CORBA编程。

一、编写IDL

1、建一个空的Solution,然后在该Solution目录下新建一个名为idl的Win32 Static Library工程(对于一般的TAO工程,无需单独为idl建一个Project,但由于我们要重用该idl,所以单独为其建立一个工程比较便于管理)。然后新建一个文本文件,添加如下内容:

```
#ifndef __ECHO_IDL__
#define __ECHO_IDL__
```

```
interface Echo {
    string echoString(in string mesg);
};
```

```
#endif // __ECHO_IDL__
```

将该文件命名为echo.idl,保存在idl工程目录下,并将其添加到idl工程中。

2、在echo.idl上点击右键,选择Properties,将其编译工具由默认的midl改为Custom Build Tool,选择Apply,在Custom Build Step标签页Command栏内填入:

```
tao_idl -Sc $(InputFileName)
```

Outputs栏填入:

```
echoC.h
```



ACE/TAO讨论区

[1.ACE User group\(RSS\)](#)[2.ACE China\(RSS\)](#)[3.Huihoo ACE/TAO](#)

ACE/TAO网站

[1.ACE官方网站](#)[2.ACE Doxygen](#)

Blog

[1.My blog\(RSS\)](#)[2.ACE粉丝's Blog\(RSS\)](#)[3.孙向晖的blog\(RSS\)](#)[我的论坛\(RSS\)](#)[我的豆瓣\(RSS\)](#)

存档

[2010年01月\(1\)](#)[2009年12月\(2\)](#)[2009年11月\(3\)](#)[2009年10月\(1\)](#)[2009年08月\(2\)](#)[2009年07月\(4\)](#)[2009年06月\(3\)](#)[2009年05月\(2\)](#)[2009年02月\(4\)](#)[2009年01月\(1\)](#)[2008年12月\(9\)](#)[2008年11月\(9\)](#)

echoC.cpp

echoC.inl

echoS.h

echoS.cpp

echoS.inl

点击OK，保存设置，在echo.idl上点击Compile编译该idl文件，编译后，将在idl工程目录下产生上述各文件。将echoC.h/echoC.cpp/echoS.h/echoS.cpp添加到idl工程中，在idl工程上点击右键，选择Properties，在Librarian下添加如下Additional Dependencies：

aced.lib

taod.lib

TAO_AnyTypeCoded.lib

按F7编译上述工程。

二、直接通过*IOR*访问*Servant*

新建一个eg1_svr工程，添加到当前Solution，在该工程中添加一个名为eg1_svr.cpp的文件，其内容如下：

```
#include <iostream>
```

```
#include "../idl/echoS.h"
```

```
using namespace std;
```

```
class Echo_i : public POA_Echo,
public PortableServer::RefCountServantBase
```

```
{
```

```
public:
```

```
    inline Echo_i() {}
```

```
    virtual ~Echo_i() {}
```

```
    virtual char* echoString(const char* mesg);
```

```
};
```

```
char* Echo_i::echoString(const char* mesg)
```

```
{
```

```
    cerr << "Upcall " << mesg << endl;
```

```
    return CORBA::string_dup(mesg);
```

```
}
```

```
////////////////////////////////////
```

```
int main(int argc, char** argv)
```

2008年10月(11)

2008年09月(26)

2008年07月(1)

2007年07月(2)

2006年12月(1)

2006年11月(4)

2006年10月(1)

2006年08月(5)

2006年07月(7)

2006年06月(11)

```
{
    try {
        // initialize ORB
        CORBA::ORB_var orb = CORBA::ORB_init(argc, argv);
        // find RootPOA
        CORBA::Object_var obj = orb->resolve_initial_references("RootPOA");
        PortableServer::POA_var poa = PortableServer::POA::_narrow(obj);
        // create servant object and activate it
        Echo_i* myecho = new Echo_i();
        PortableServer::ObjectId_var myechoid = poa->activate_object(myecho);
        // Obtain a reference to the object, and print it out as a
        // stringified IOR.
        obj = myecho->_this();
        CORBA::String_var sior(orb->object_to_string(obj));
        cerr << (char*)sior << endl;
        myecho->_remove_ref();
        // find and activate POAManager
        PortableServer::POAManager_var pman = poa->the_POAManager();
        pman->activate();
        orb->run();
    }
    catch(CORBA::SystemException&) {
        cerr << "Caught CORBA::SystemException." << endl;
    }
    catch(CORBA::Exception&) {
        cerr << "Caught CORBA::Exception." << endl;
    }
    catch(...) {
        cerr << "Caught unknown exception." << endl;
    }
    return 0;
}
```

使该工程依赖于前面的工程idl。由于这是一个Server程序，需要链接TAO_PortableServerd.lib。
按F7，编译该工程。

新建一个eg1_cli工程，添加到当前Solution，在该工程中添加一个名为eg1_cli.cpp的文件，其内容如下：

```

#include <iostream>
#include "../idl/echoC.h"
using namespace std;

static void hello(Echo_ptr e)
{
    CORBA::String_var src = (const char*) "Hello!";
    CORBA::String_var dest = e->echoString(src);
    cerr << "I said, \"\" << (char*)src << "\".\" << endl
         << "The Echo object replied, \"\" << (char*)dest << "\".\" << endl;
}

////////////////////////////////////

int main(int argc, char* argv[]) {
    try {
        // initialize orb
        CORBA::ORB_var orb = CORBA::ORB_init(argc, argv);
        // check arguments
        if (argc != 2) {
            cerr << "Usage: eg1_cli <object reference>" << endl;
            throw 1;
        }
        // Obtain reference from servant IOR
        CORBA::Object_var obj = orb->string_to_object(argv[1]);
        Echo_var echoref = Echo::_narrow(obj);
        if( CORBA::is_nil(echoref) ) {
            cerr << "Can't narrow reference to type Echo (or it was nil)." << endl;
            return 1;
        }
        for (CORBA::ULong count=0; count<10; count++)
            // communicate with servant
            hello(echoref);
        orb->destroy();
    }
    catch(CORBA::COMM_FAILURE&) {
        cerr << "Caught system exception COMM_FAILURE -- unable to contact the "

```

```

        << "object." << endl;
    }
    catch(CORBA::SystemException&) {
        cerr << "Caught a CORBA::SystemException." << endl;
    }
    catch(CORBA::Exception&) {
        cerr << "Caught CORBA::Exception." << endl;
    }
    catch(...) {
        cerr << "Caught unknown exception." << endl;
    }

    return 0;
}

```

使该工程依赖于前面的工程idl，编译该工程。

代码比较简单，请自行参照注释阅读源代码。

三、运行

运行时应先启动eg1_svr，该程序会在标准输出上输出一长串字符，即servant的IOR信息，我们有了该信息就可以通过orb->string_to_object(...)连接到相应的servant了。

现在可以启动客户程序eg1_cli了，启动eg1_cli时应将上面获得的IOR串作为参数传递给该程序（注意，从Console上拷贝下来的时候会折行，将其拷贝到文本编辑器中去掉折行再拷出来即可）。

附：本文示例源代码

参考：

1. Duncan Grisby, Sai-Lai Lo, David Riddoch. The omniORB Version4.0 User's Guide. <http://omniorb.sourceforge.net/omni40/omniORB.pdf>

发表于 @ 2006年06月30日 13:34:00 | [评论\(1\)](#) | [举报](#) | [收藏](#)

旧一篇: [CORBA Programming with TAO - 4.Basic Functions](#)（常用基本方法解析） | 新一篇: [CORBA Programming with TAO - 6.Naming Service](#)（使用命名服务）

[shuiyuan2004](#) □□□Wednesday, May 23, 2007 09:48:07 □□ □□



□□□~
□□□□□□□□□□□□□□□□□□

□□□□Hello World

□□□□□□□□□□

ACE_DLL::open failed for TAO_PI: Error: check log for details.

Unable to find service: ORBInitializer_Registry

ACE_DLL::open failed for TAO_Codeset: Error: check log for details.

Unable to find service: TAO_Codeset

IOR:010000000d00000049444c3a4563686f3a312e300000000000100000000000004800000000101

02cd0a000000031302e322e312e383500a3111b00000014010f005253542992534687930300000000

000100000001000000cd0100000000000000800000001cdcdcd004f4154

□□□□□□□□

[发表评论](#)

表 情:



评论内容:

用 户 名: huapuyu6

匿名评论