用户名不同的计算机之间的调用dcom

DCOM是依赖于RPC服务的，是一种分布式远程过程（进程）调用，调用代码在远程主机上执行，使用远程主机的系统资源，因此要有远程主机所属的访问权限，一般来说访问用户应是远程主机本地用户或主机所属的域用户，所以

第一步是在远程主机或其域上注册（获得）一个有足够权限的用户，

第二步是使用此用户令牌激活远程COM服务器，方法是使用CoInitializeSecurity & CoCreateInstanceEx函数(见文后例子)，成功激活后就得到相关请求接口的代理，此时代理并没有对远程服务器访问的权限，

第三步是为接口代理设置访问权限，方法是使用CoSetProxyBlanket函数，此后就可以使用代理指针进行方法调用了。见文后例子

以上是客户端要做的处理，至于服务器端，只要使用CoInitializeSecurity函数设置一下访问许可权限就可以了（有时客户端也要进行这一处理，比如使用了连接点），对于即存的服务器程序使用dcomcfg工具在注册表里设置也可以。

CoSetProxyBlanket函数调用例子：

****

原例子：

```
IRecordServerLink* pRSL=(IRecordServerLink*)mq[0].pItf;

hr = CoSetProxyBlanket( pRSL, RPC_C_AUTHN_WINNT, RPC_C_AUTHZ_NONE, NULL,
        RPC_C_AUTHN_LEVEL_DEFAULT, RPC_C_IMP_LEVEL_IMPERSONATE, &caid,
EOAC_NONE);
pRSL->Methods();
```

————————————————————————————————————

修改后的例子：

```
hr = CoSetProxyBlanket( pAccount, RPC_C_AUTHN_WINNT, RPC_C_AUTHZ_NONE,
NULL,
        RPC_C_AUTHN_LEVEL_DEFAULT, RPC_C_IMP_LEVEL_IMPERSONATE, &caid,
EOAC_NONE);
pAccount->Deposit(x);
```

※※

CoInitializeSecurity & CoCreateInstanceEx 函数调用例子：

****

例子 1:

```
HRESULT hr;
hr =  CoInitialize(NULL);
ASSERT(SUCCEEDED(hr));
MULTI_QI qi;
qi.pIID = &IID_ICRemoteTime;
qi.hr = NULL;
qi.pItf = NULL;
COAUTHIDENTITY authidentity;
authidentity.User = L"administrator";   ////你机子的帐号
authidentity.UserLength = wcslen(authidentity.User);
authidentity.Domain = NULL;
authidentity.DomainLength = 0;
authidentity.Password = L"";          ////密码
authidentity.PasswordLength = wcslen(authidentity.User);
authidentity.Flags = SEC_WINNT_AUTH_IDENTITY_UNICODE;
COAUTHINFO authinfo = {-1, 0, 0, RPC_C_AUTHN_LEVEL_DEFAULT,
      RPC_C_IMP_LEVEL_IMPERSONATE, &authidentity, 0};
COSERVERINFO servInf;
servInf.dwReserved1 = NULL;
servInf.dwReserved2 = NULL;
// servInf.pAuthInfo = NULL;
servInf.pAuthInfo = &authinfo;
USES_CONVERSION;
servInf.pwszName = L"127.0.0.1";
hr = CoInitializeSecurity(NULL, -1, NULL, NULL,
  RPC_C_AUTHN_LEVEL_NONE,       RPC_C_IMP_LEVEL_IDENTIFY,       NULL,
EOAC_ACCESS_CONTROL, NULL);
ASSERT(SUCCEEDED(hr));
hr = CoCreateInstanceEx(CLSID_CRemoteTime, NULL, CLSCTX_REMOTE_SERVER,
&servInf, 1, &qi);
if (FAILED(hr))
{
 TRACE(_T("CoCreateInstanceEx failed"));
 return false;;
}
```

```
if (FAILED(qi.hr))
{
 TRACE(_T("Failed to connect to server"));
 return false;;
}
//通过 IUnkonwn 指针去查询接口指针,返回 IAccount 指针
hr = pUnknown->QueryInterface(IID_ICRemoteTime,(void**)&pIRetime)
```

例子 2：

```
HRESULT hr = CoInitializeSecurity(NULL, -1, NULL, NULL,
  RPC_C_AUTHN_LEVEL_NONE, RPC_C_IMP_LEVEL_IDENTIFY, NULL, EOAC_NONE,
NULL);

 COAUTHIDENTITY us;

 us.User           = m_strName.AllocSysString();
 us.UserLength     = wcslen(us.User);
 us.Password       = m_strPassword.AllocSysString();
 us.PasswordLength = wcslen(us.Password);
 us.Domain         = m_strDomain.AllocSysString();
 us.DomainLength   = wcslen(us.Domain);
 us.Flags          = SEC_WINNT_AUTH_IDENTITY_UNICODE;

 COAUTHINFO auth;

 auth.dwAuthnSvc           = RPC_C_AUTHN_WINNT;
 auth.dwAuthzSvc           = RPC_C_AUTHZ_NONE;
 auth.pwszServerPrincName  = NULL;
 auth.dwAuthnLevel         = RPC_C_AUTHN_LEVEL_CONNECT;
 auth.dwImpersonationLevel = RPC_C_IMP_LEVEL_IMPERSONATE;
 auth.dwCapabilities       = EOAC_NONE;
 auth.pAuthIdentityData    = &us;

COSERVERINFO si;
 MULTI_QI     qi;

 CComBSTR bstr = strComputer;
 LPWSTR name   = bstr.m_str;
 si.dwReserved1 = 0;
 si.pwszName    = name;
 si.pAuthInfo   = m_bAccess ? &auth : NULL;
 si.dwReserved2 = 0;

 IID iid = __uuidof(m_pIRemoteControl);
 qi.pIID = &iid;
 qi.pItf = NULL;

 do
 {
  hr = CoCreateInstanceEx(__uuidof(RemoteControl), NULL, CLSCTX_SERVER, 
&si, 1, &qi);
  if(FAILED(hr) || FAILED(qi.hr))
   break ;

  m_pIRemoteControl = (IRemoteControl *)qi.pItf;
 }while(0);
```

————————————————————————————————————

例子 3：

```
HRESULT hr = CoInitializeSecurity(NULL, -1, NULL, NULL,
  RPC_C_AUTHN_LEVEL_NONE, RPC_C_IMP_LEVEL_IDENTIFY, NULL, EOAC_NONE, 
NULL);

 COAUTHIDENTITY us;

 us.User           = m_strName.AllocSysString();
 us.UserLength     = wcslen(us.User);
 us.Password       = m_strPassword.AllocSysString();
 us.PasswordLength = wcslen(us.Password);
 us.Domain         = m_strDomain.AllocSysString();
 us.DomainLength   = wcslen(us.Domain);
 us.Flags          = SEC_WINNT_AUTH_IDENTITY_UNICODE;

 COAUTHINFO auth;
 auth.dwAuthnSvc            = RPC_C_AUTHN_WINNT;
 auth.dwAuthzSvc            = RPC_C_AUTHZ_NONE;
 auth.pwszServerPrincName   = NULL;
 auth.dwAuthnLevel          = RPC_C_AUTHN_LEVEL_CONNECT;
 auth.dwImpersonationLevel  = RPC_C_IMP_LEVEL_IMPERSONATE;
 auth.dwCapabilities        = EOAC_NONE;
 auth.pAuthIdentityData     = &us;


 MULTI_QI MultiQi={&IID_IUnknown,NULL,NOERROR};
 COSERVERINFO si;

 CComBSTR bstr = strComputer;
 LPWSTR name   = bstr.m_str;

 si.dwReserved1 = 0;
 si.pwszName    = name;
 si.pAuthInfo   = m_bAccess ? &auth : NULL;
 si.dwReserved2 = 0;

  hr = CoCreateInstanceEx(__uuidof(RemoteControl), NULL, CLSCTX_SERVER,
&si, 1, &MultiQi);

 if(FAILED(hr))
 {
  MessageBox("创建对象实例失败！");
  return;
 }

  //通过 IUnkonwn 指针去查询接口指针,返回 IAccount 指针
 pUnknown = (IUnknown *) MultiQi.pItf;

 hr = pUnknown->QueryInterface(IID_IAccount,(void**)&pAccount);
 if(FAILED(hr))
 {
  MessageBox("没有查找的接口指针!");
  return false;
 }
 pUnknown->Release();
```