Making Delphi work with DCOM (http://www.sandon.it/node/39)

Posted on: Sun, 02/01/2009 - 23:20 **By:** ldsandon

In the past few days I had many troubles making our application work when the client was installed on a PC outside the application domain. It was due to several Delphi 2007 issues with DCOM.

Usually, when a user was outside a domain we used TSocketConnection and the "Borland" Socket Server to establish the connection. "Outside" the domain usually meant remote PC, and the socket connection was more firewall-friendly. But it turned out that the socket connection wasn't correctly upgraded when new datatypes - i.e. Int64 - were introduced (see qc #69924 (http://qc.codegear.com/wc/qcmain.aspx?d=69924), for example) and that meant we could no longer use it until we get a fix, or rewrite the application to change the unsupported datatypes.

Thereby I was asked to make it work via TDCOMConnection even with users outside a domain. After a few searches, I found that there is a workaround. If the DCOM server has a **local** user with the same username and password of the client user, it will be authenticated and allowed to launch and call the server. Of course, some setup was needed.

- Because we were using a Windows 2003 server, security limits in DCOMCnfg ? DCOM Security were modified to allow our users' group to be enabled. Windows 2003 comes with a "Distributed COM User" group that could be used for that purpose, but custom groups will work as well.
- The users' group was added to the DCOM server permissions.

Everything worked fine as long as we used the "launching user" identity for the server. But this mode may have several drawbacks. First, one server instance is launched for every connected user, because each instance is run in the user security context. Second, if the server is called at the "impersonation" level (the default), it won't be allowed to access resources outside the machine it is running on. To obtain outside access, the client should ask for "delegation" level, and the server has to be trusted for delegation (this is a server property in Active Directory Users and Computers snap-in).

Unluckily, TDCOMConnection does not allow to set per-connection DCOM properties. Most of it was written in Windows 95/NT4 times, and never improved since them. Windows 95 had only a partial DCOM implementation (because it can't join a domain) and DCOM too added many features since NT 4. To set connection properties server defaults have to be changed.

Another solution is to set the server to run with a specified user. If per-user checks are not needed, it's a viable solution. That's how we usually configured the server when using TSocketConnection. But we found another issue. When the server was set to run with a specific user, the client was able to launch the server, but then the dreaded "Access denied" error was then returned, and the server hung, requiring to be killed. What was happening?

After much searching, it turned out it was due to qc #8814 (http://qc.codegear.com/wc/qcmain.aspx?d=8814), a bug filed almost five years ago and still in the "reported" state (note: it was opened later after I complained about it a lot). It looks Delphi does not register the DCOM server properly, some registry keys are missing. After applying the workaround suggested, the client was able to execute the server properly.

The last issue was due to some DCOM callbacks used to allow the client to be informed when a query returned data. Because when using callbacks the server and client roles are inverted, the server must be able to call the client. Thereby a user equal to the one created to run the server had to be added to clients, and allowed to be called via DCOM. Because a DCOM client doesn't get registered, default permission may be given.

Some other issues surfaced when a Linux VPN server was thrown in the mix. As long as VPN security was handled on the Linux box, the DCOM connection didn't work. The solution was to use RADIUS to authenticate the VPN user in the Active Directory domain. We used Windows' RADIUS server - Internet Authentication Server (IAS) - because it could use AD accounts easily and had no issues to work with the Linux VPN software.

- Log in (http://www.sandon.it/user/login?destination=/comment/reply/node/39/comment_node_blog%23comment-form) to post comments

Source URL: *http://www.sandon.it/node/39*