




Process Engineer DCOM Settings

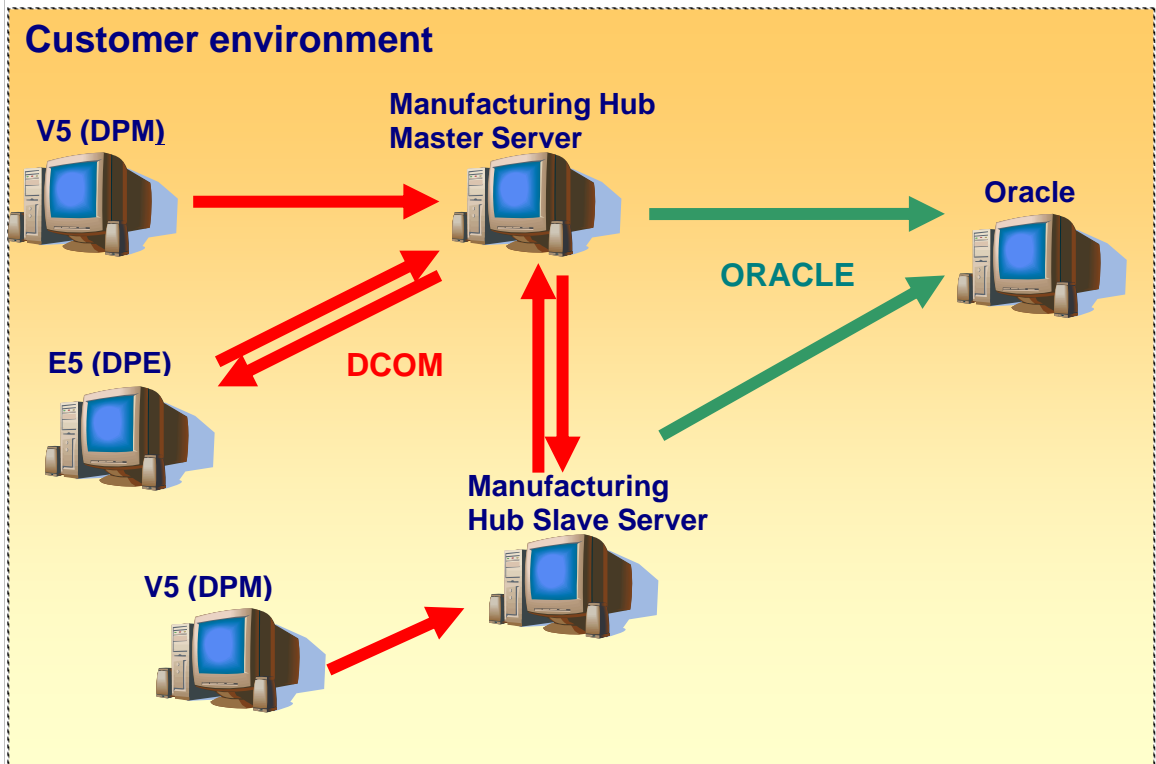
Process Engineer DCOM Settings	1
Overview	3
DCOM (Distribution Component Object Model)	3
Callbacks	4
DCOM and Firewalls	4
Prerequisites	5
Working with enabled DCOM Authentication	6
Server Settings	7
Configure the User Identity of the Server Processes	8
Set Process Launch Permissions:	9
Set Process Access Permissions:	10
Windows Server 2003 SP1 (and newer) Policies add-ons	11
Working across Network Domains	13
Working with disabled DCOM Authentication	14
When does it make sense to work with disabled Authentication?	14
Server Settings	14
Configure the User Identity of the Server Processes	14
Permissions on Server Processes:	14
Windows Server 2003 (and newer) Policies add-ons	17
Windows Server 2003 SP1 (and newer) Policies add-ons	18
Client Settings	21
Windows XP SP2 Operation System add-ons	21
Working with DCOM encryption	23
Which server processes to customize?	24
Assign local administrator rights to a user	25
How to Launch DCOM configuration Tool	26
On Windows XP and Windows Server 2003	26
On Windows 2000	27
How to Launch Local Policies configuration	28
How to enable DCOM?	29
Customize Windows Firewall	30



How to disable Windows Firewall?	30
Enable DPE Client to run on Windows XP SP2 with an enabled Windows Firewall	31
Enable Server Processes to run on Windows Server 2003 SP1 with an enabled Windows Firewall	34
Checklist for connection problems	35
DCOM settings for multiple clients on Novell networks without a Windows Domain Controller	37
Authentication	39
DCOM HTTP Tunneling	40
General	40
Setup	42
System requirements	42
Client and Server machine configuration:	42
Example	45

Overview

The Process Engineer solution consists of 3 different layers. These 3 layers are database, server applications and client applications. Part of the solution is the capability to run these 3 layers on different computer. The interaction between these layers is provided via technologies that provide network capabilities.



The solution used as database is Oracle. Oracle itself uses by default TCP/IP to communicate via network and therefore provides the technology that allows the communication between server and database.

This manual concentrates on explaining the setup of the other technology, which is used by the Server and Client part of the solution for network communication. This technology is called DCOM.

DCOM (Distribution Component Object Model)

DCOM is a technology that has been put on the market by Microsoft in the middle of the 90th. It has been developed on top of COM (Formerly called also OLE). COM itself is a technology that initial was used for inter module/inter process communication inside computer boundaries. DCOM enhanced this technology by the capability to communicate via network. Internal DCOM uses **Remote Procedure Calls (RPC)**, which itself again is implemented base on the TCP/IP network protocol.

What makes DCOM difficult to install and to administrate is the security model, which it provides. This manual will explain the settings that have to be customized on client and server side in order to enable the solution to work via network.

Callbacks

A callback represents in DCOM a connection that is build up backward from the server to the client. Since this type of connections are built up by DCOM in this way, in some scenarios customization effort on the Client OS is necessary. Server applications use these methods for asynchronous communication with the clients. In the context of DPE client (also called E5 client) callbacks are used in first instance for keeping the data on the client side up to date. Since DPM clients (also called V5 clients), compared to the DPE clients, ensure the consistency of the data in another way, they also don't need any callback mechanism.

The usage of callback, in our solution, is in majority the reason for client side customization. The manual will explain in which scenarios client side customization is required.

DCOM and Firewalls

The DCOM technology uses a dynamically assignment of TCP/IP ports for communication purposes. In theory DCOM could use all ports on an individual machine that are not used for other communication purposes. This fact and the fact that the solution is using the callback functionality of DCOM makes the setup of our solution over Firewall boundaries very complex. Further explanations on how to setup DCOM cross Firewall boundaries can be found on the following Microsoft URL:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndcom/html/msdn_dcomfirewall.asp

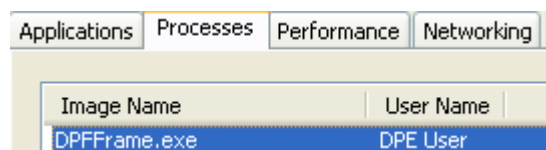
Prerequisites

The changes described further in this manual are recommendations. They base on Operation System installations where security policies are as they are shipped by Microsoft. In case the security policies have been changed by the customer further customization steps could be needed.

To execute the changes described in this manual you need to be logged in as **Local Administrator** on the machines on that changes have to be done.

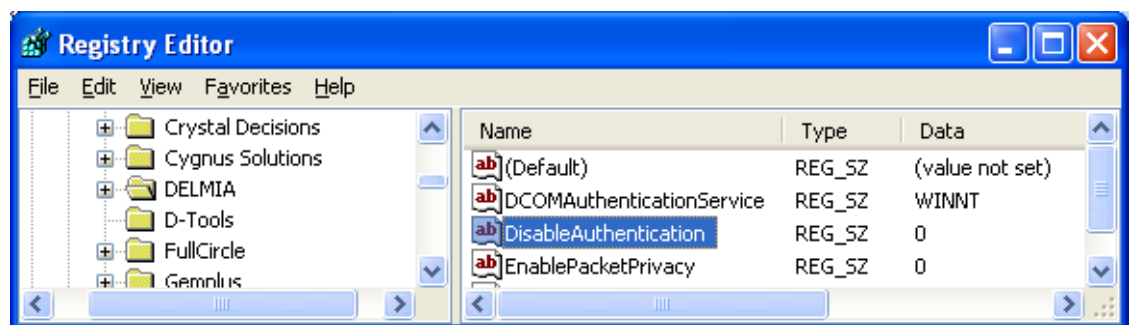
Working with enabled DCOM Authentication

Running the software with an enable DCOM Authentication means that a user authentication takes place when a connection is established. In order to authenticate the transfer of user contexts from one machine to another is performed. Per example the DPE Client, as in the following pictures shown, runs under the context of a user called **DPE User** (account of user under, which the log in to the client machine has been performed):



This user context is transferred to the server machine where first of all authentications takes place. In second instance security checks are done that determine if user has access permission.

Configuration of which mode the software uses happens via registry. By default the solution is installed to run with enabled DCOM Authentication. This means the **DisableAuthentication** value that is stored in the registry under the key **HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA** is set by default to **0**. The flag has to be consistent on all server and client machines.



Notes: We recommend the users to be logged in as a Domain User when working with the client software. Also we recommend the server process to run under a Domain User. The usage of a Domain User for the server process identity is suggested cause of the callback connections. The general use of Domain Users makes the administration and setting up of the solution easier.

Server Settings

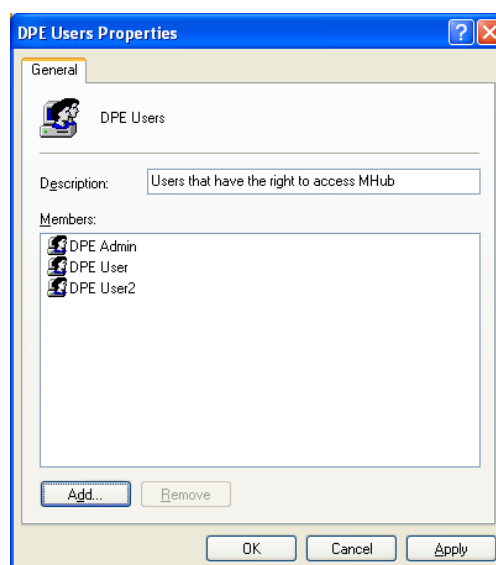
When setting up the system the first thing to do, is to define the user identity under which the server processes have to run. We recommend the use of a **Domain User** that has **Local Administrator Rights** (see chapter [Assign local administrator rights to a user](#)) for this purpose. Due to the callback connections the server has to authenticate on the client machines. This means the user identity used by the server processes has to be known on the client machines (applies only in case client is a DPE client).

Generally spoken, if non domain users are used either as client user identity or server process identity, the only option is to make the user known by creating a user with the same name/password combination on the opposite machine.

The following steps have to be done in order to setup DCOM Permissions in the right way. How to do these steps will be explained further on in own sub chapters:

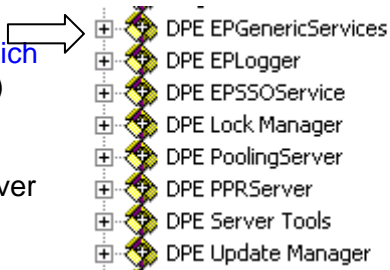
- The user identity has to be set for the Server Processes
- Set the Launch Permission for the Server Processes
- Set the Access Permission for the Server Processes
- Depending on the used server OS, machine policies have to be adapted.

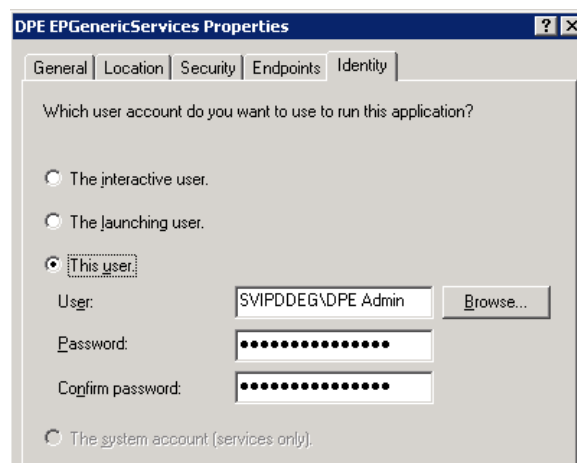
For setting up the permissions in a convenient way we recommend setting up a **Domain Group** containing all users that work with the solution and also **the user identity of the server processes**:



If only one server machine is set up, in that case a **Local Group** is sufficient.

Configure the User Identity of the Server Processes

- ➔ Start DCOM Configuration, for how see also chapter [How to Launch DCOM configuration Tool](#).
- ➔ Set DCOM Identity on required processes (see also chapter [Which server processes to customize?](#)) DPE Server processes.
 
- ➔ Select *Properties...* on each server process.
- ➔ Select the **Identity** tab:
- ➔ Check the radio button *This user* and enter the user name (DPE Admin) and user password.
- ➔ Confirm with OK

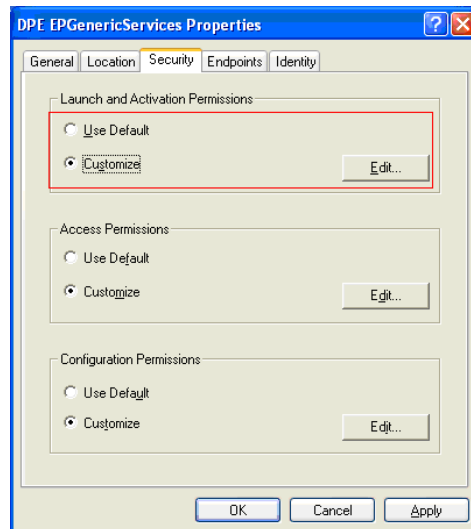


- ➔ The server processes from now should run under the **DPE Admin** user account.

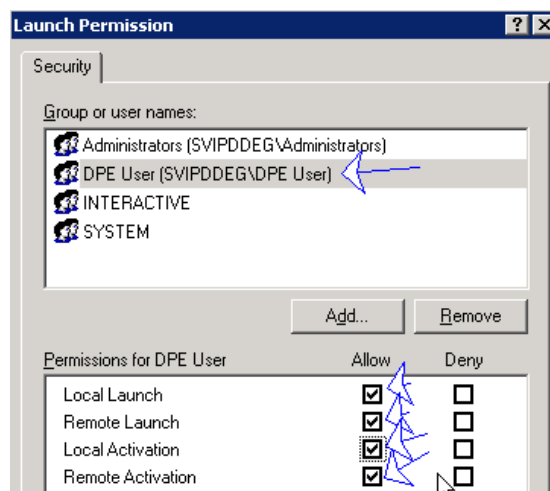
Applications	Processes	Performance	Networking	Users
Image Name	User Name	CPU	Mem Usage	
lockmng.exe	DPE Admin User	00	6,724 K	
EPServerTools.exe	DPE Admin User	00	22,916 K	
UpdateMng.exe	DPE Admin User	00	7,184 K	
eppoolingserver.exe	DPE Admin User	00	8,940 K	
IPDServer.exe	DPE Admin User	00	50,576 K	

Set Process Launch Permissions:

- Select **Properties...** on each server process (DPE...) in DCOM configuration Tool.
- Select **Customize** and **Edit** to set the Launch Permissions. Exit the dialog with **OK**.

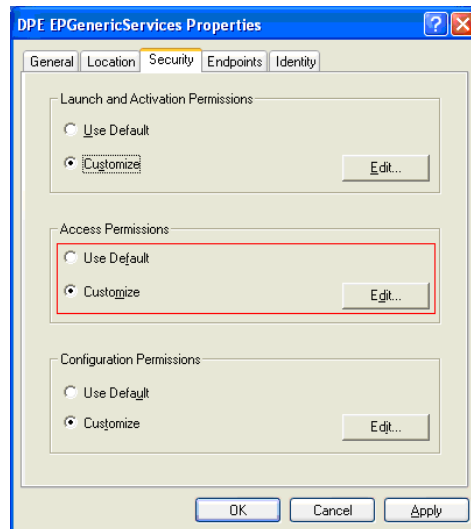


- Assign **DPE User** Group all Permissions and exit with **OK**

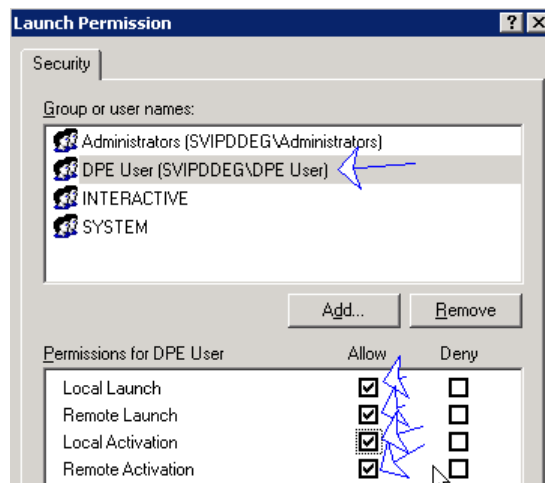


Set Process Access Permissions:

- Select **Properties...** on each server process (DPE...) in DCOM configuration Tool.
- Select **Customize** and **Edit** to set the Access Permissions. Exit the dialog with **OK**.



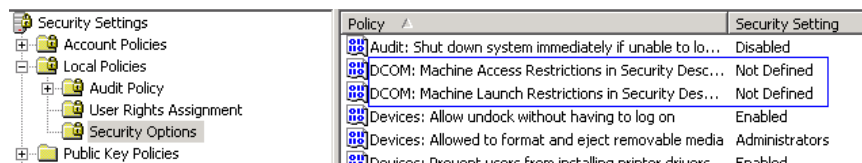
- Assign **DPE User Group** all Permissions and exit with **OK**.



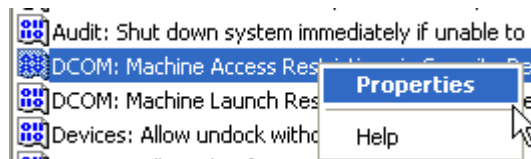
Windows Server 2003 SP1 (and newer) Policies add-ons

With the introduction of Windows Server 2003 SP1 also new DCOM policies have been introduced. These policies control on a machine level, which is allowed to launch and access DCOM processes. For setting up these in an adequate way please refer to the following steps.

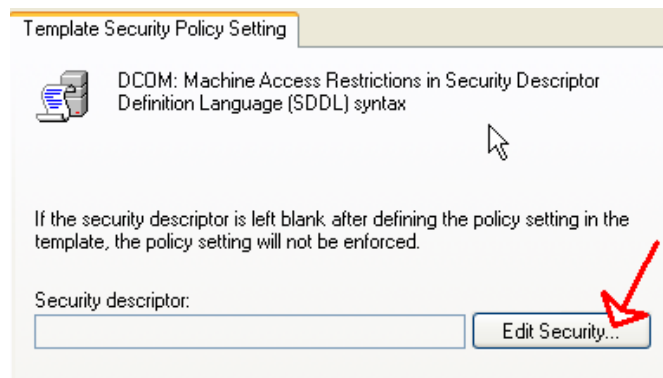
- ➔ Start Local Security settings. See also chapter [How to Launch Local Policies configuration](#).
- ➔ Select **Security Options** and edit **DCOM Machine Access** and **Launch Restrictions**.



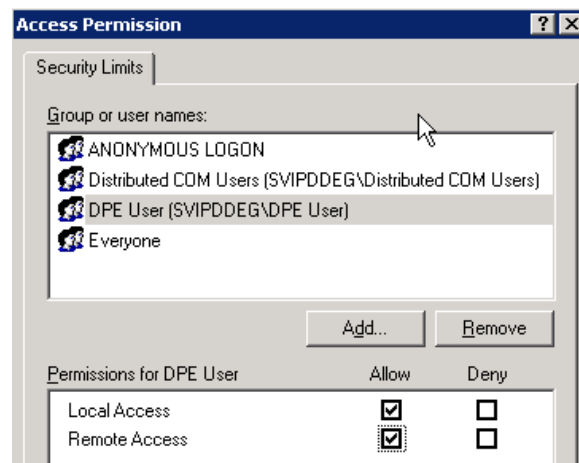
- ➔ Open **Properties** on **DCOM: Machine Access Restriction**.



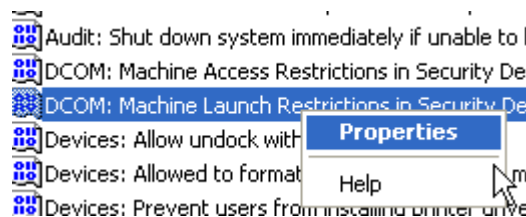
- ➔ Press Button **Edit Security...**



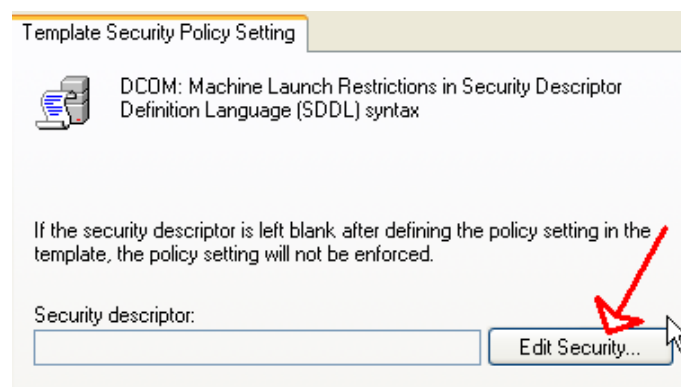
- Add group DPE User.
- Allow Access from Local/Remote.



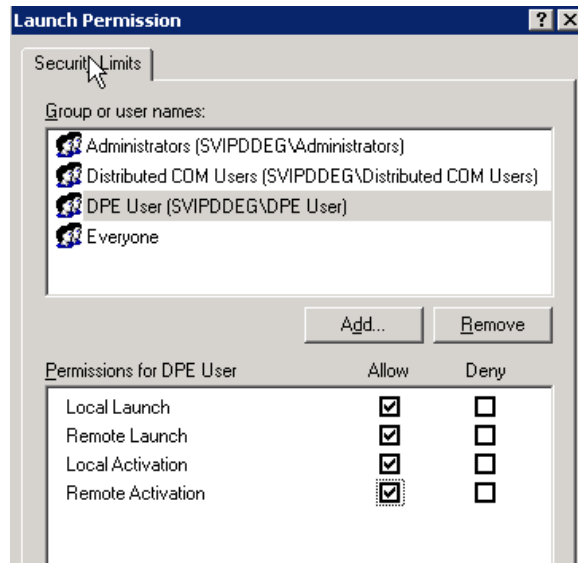
- Confirm with **OK** (two times).
- Open **Properties** DCOM: Machine Launch Restriction ...



- Select Button **Edit Security...**



- ➞ Add group **DPE User** to user list in dialog.



- ➞ Assign to **DPE User** group all permissions.
- ➞ Confirm with **OK** (two times).

Working across Network Domains

In case the solution should run across Network Domain boundaries the client user account has to be known on the server machine. The options here:

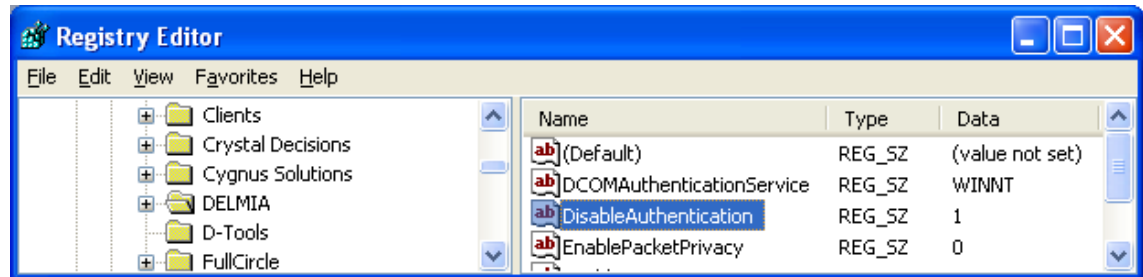
1. Work with trusted Network Domains.
2. Create a Domain user with the same name/password combination in the server domain.
3. Create a local user with the same name/password combination on the server machine.

In case a DPE Client is connecting to a server across Network Domain boundaries the same applies for the server process identity. It has to be known on the client machine.

Another alternative is not to work with enabled Authentication. See chapter [Working with disabled DCOM Authentication](#).

Working with disabled DCOM Authentication

First step for running DCOM with a disabled Authentication is to set the registry value **DisableAuthentication** under **HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA** to 1 (Default is 0). The flag has to be consistent on all server and client machines.



When does it make sense to work with disabled Authentication?

- Novell network. For further information have a look on the [DCOM settings for multiple clients on Novell networks without a Windows Domain Controller](#) chapter).
- When solution is setup over multiple network domains that run not as trusted domains. A disabled Authentication could provide in this case an easier way to administrate security through whole the solution.

Server Settings

Configure the User Identity of the Server Processes

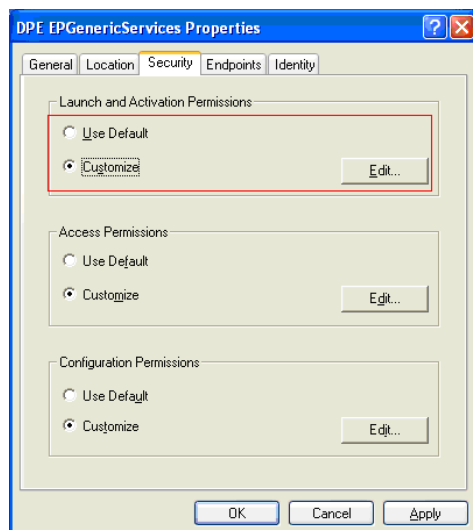
The process identity of the server process has to be set up in the same way as with enabled DCOM Authentication. Have a look [corresponding section](#) of the “Working with DCOM Authentication” chapter.

Permissions on Server Processes:

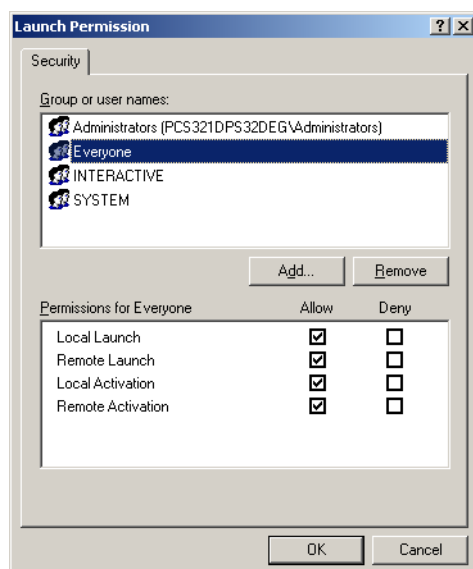
For connecting and accessing server processes by the client applications the Launch and Access Permissions have to be set up. For this purpose the DCOM Configuration Tool has to be started. See [How to Launch DCOM configuration Tool](#) section. In order to know which server processes to setup please look into section “[Which server processes to customize?](#)”

Set Process Launch Permissions:

- ➔ Select **Properties...** on each server process.
- ➔ Select **Customize** and **Edit** to set Permissions. Exit the dialog with OK.

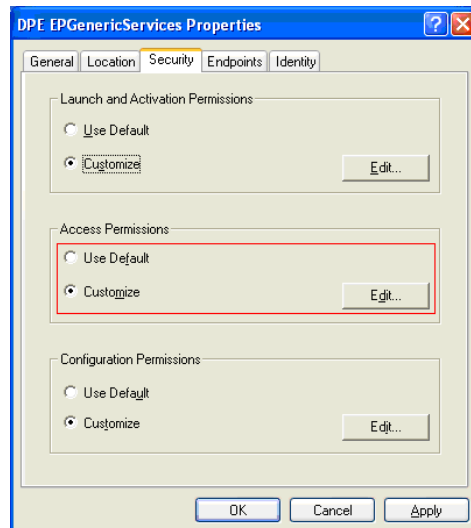


- ➔ Assign **Everyone** Group all Permissions:

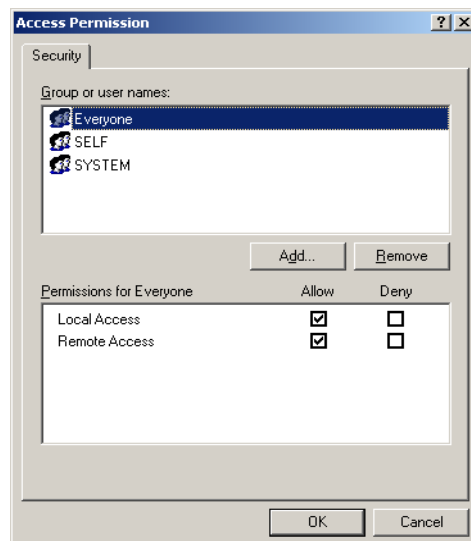


Set Process Access Permissions:

- ➔ Select **Properties...** on each server process
- ➔ Select **Customize** and **Edit** to set the Permissions. Exit the dialog with **OK**.



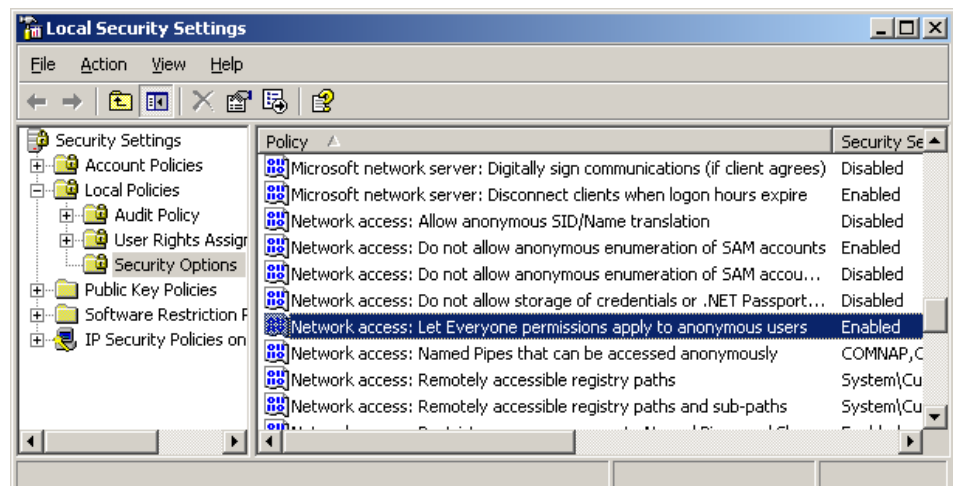
- ➔ Assign **Everyone** Group all Permissions:



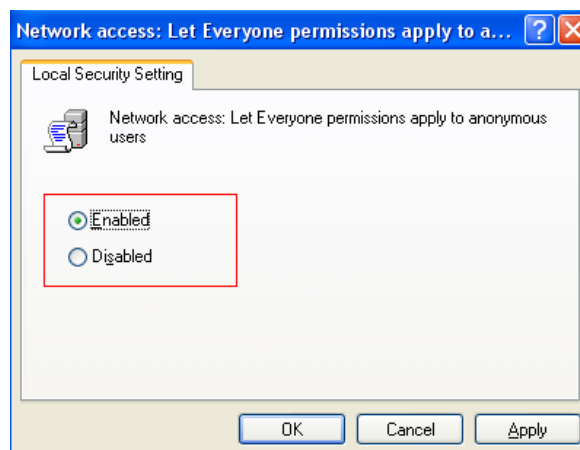
Windows Server 2003 (and newer) Policies add-ons

Configure Network Access Permission on machine level:

- Start Local Security settings. For further information's have a look at the chapter [How to Launch Local Policies configuration](#).
- Navigate in the tree and select the Policy **Network access: Let Everyone permissions apply to anonymous users**.



- Start **Properties** Dialog by selecting **Properties...** in the context menu on the selected item.
- Enable Security setting by checking the radio button. Exit the dialog with **OK**.

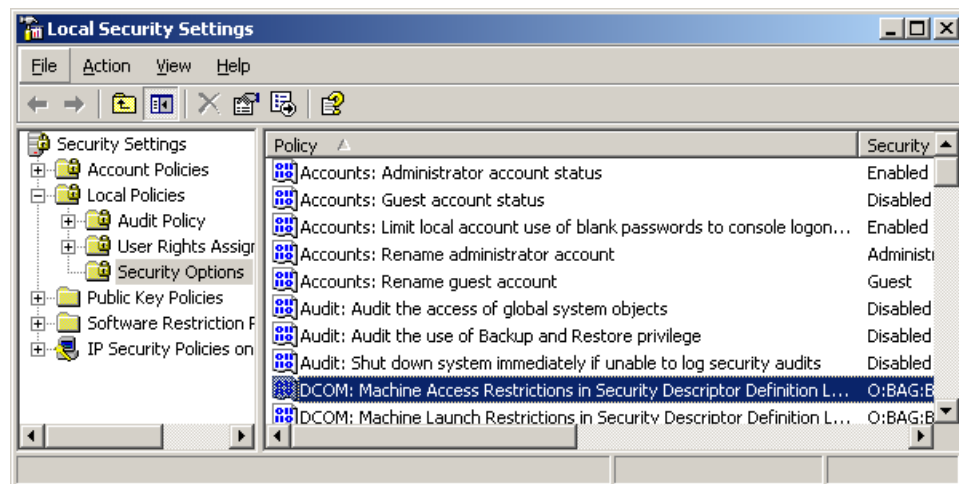


Windows Server 2003 SP1 (and newer) Policies add-ons

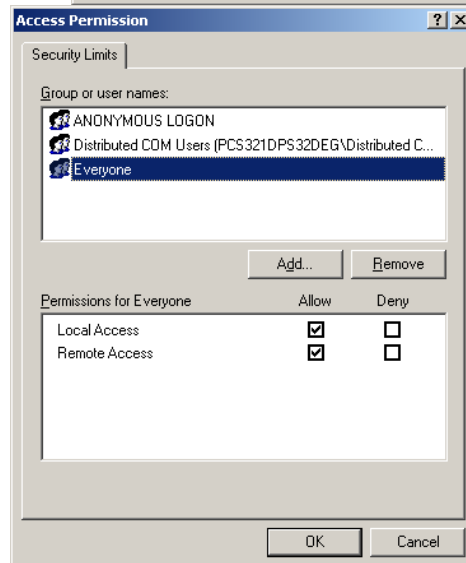
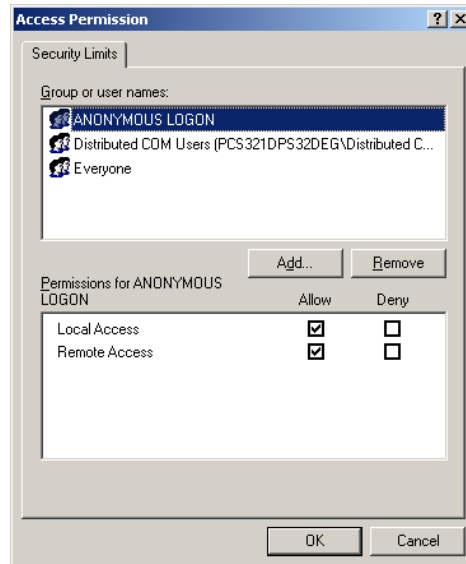
Set DCOM Launch and Access Permissions on machine level:

With the introduction of Windows Server 2003 SP1 also new DCOM policies have been introduced. These policies control on a machine level, which is allowed to launch and access DCOM processes. For setting up these in an adequate way please refer to the following steps.

- Start Local Security settings. For further informations have a look at the chapter [How to Launch Local Policies configuration](#).
- Navigate in the tree and select the Policy **DCOM: Machine Access Restrictions...**

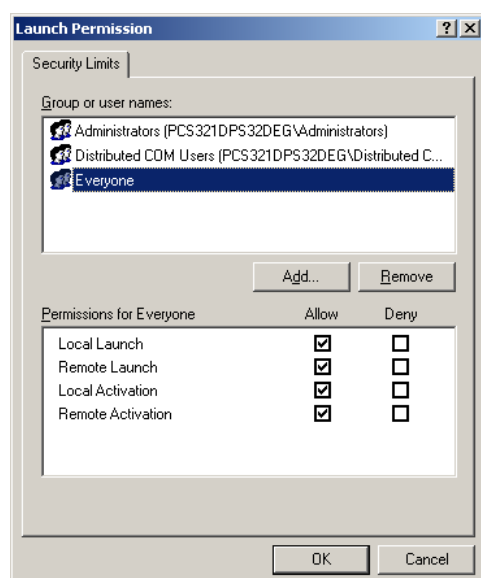


- Start **Access Permission** Dialog by selecting **Properties...** in the context menu on the selected item.
- Assign to **Everyone** and **Anonymous Logon** Group all Permissions and exit the dialog with **OK**.



- Navigate in the tree and select the Policy **DCOM: Machine Launch Restrictions...**
- Start **Launch Permission** Dialog by selecting **Properties...** in the context menu on the selected item.

- ➡ Assign to **Everyone** Group all Permissions and exit the dialog with **OK**.

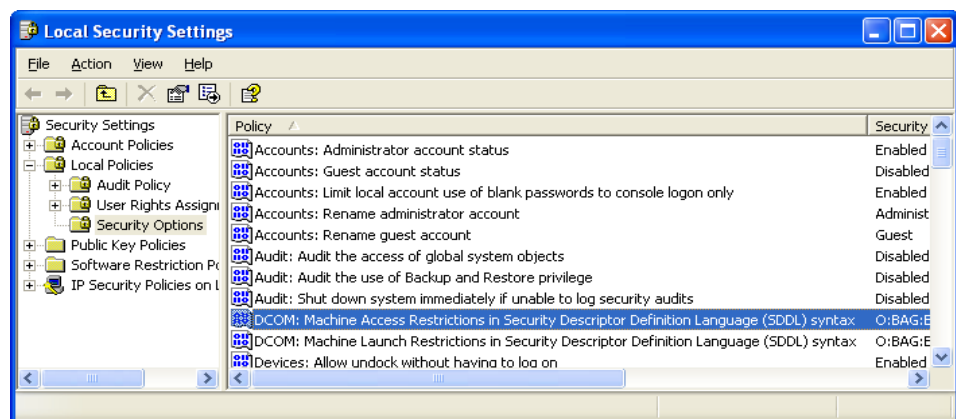


Client Settings

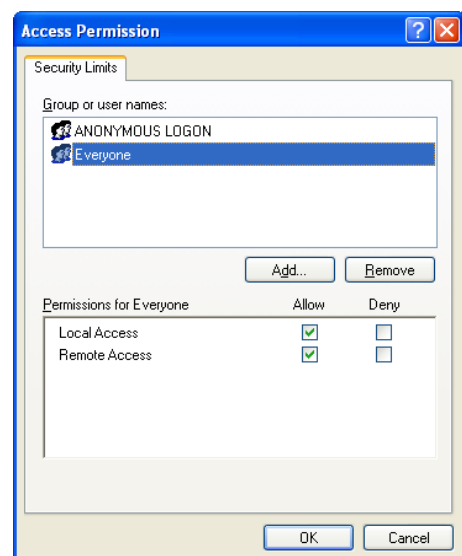
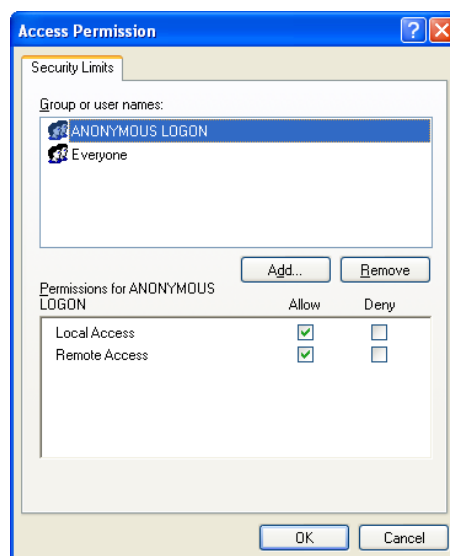
Windows XP SP2 Operation System add-ons

With the introduction of Windows XP SP2 also new DCOM policies have been introduced. These policies control on a machine level, which is allowed to launch and access DCOM processes. For setting up these in an adequate way please refer to the following steps

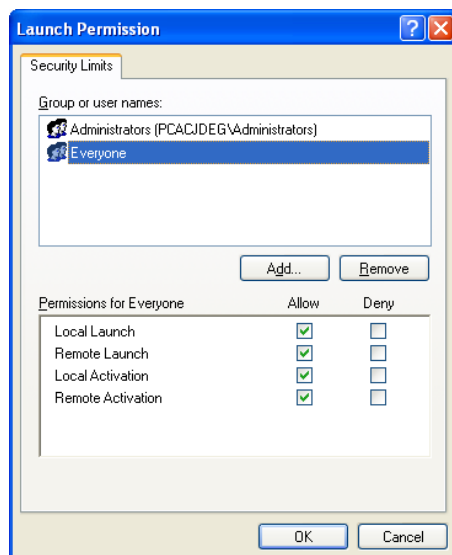
- ➔ Start Local Security settings. For further information have a look at the chapter [How to Launch Local Policies configuration](#).
- ➔ Navigate in the tree and select the Policy **DCOM: Machine Access Restrictions...**



- ➔ Start **Access Permission** Dialog by selecting **Properties...** in the context menu on the selected item.
- ➔ Assign to **Everyone** and **Anonymous Logon** Group all Permissions and leave the dialog with **OK**.



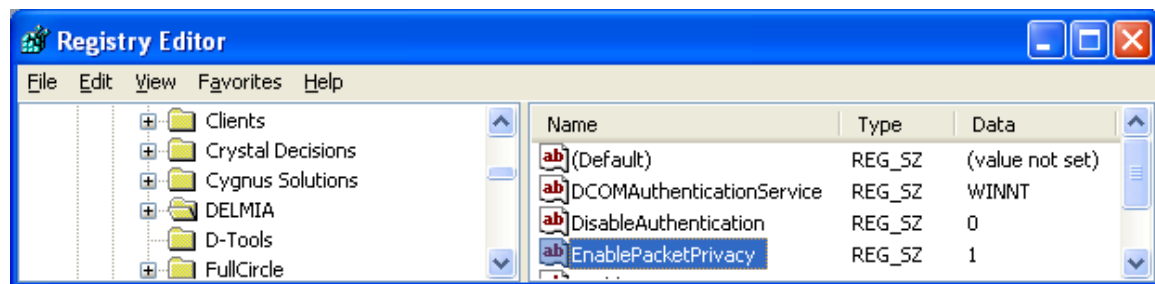
- Navigate in the tree and select the Policy DCOM: Machine Launch Restrictions...
- Start Launch Permission Dialog by selecting Properties... in the context menu on the selected item.
- Assign to Everyone Group all Permissions and leave the dialog with OK.



Working with DCOM encryption

By default, DCOM communication is not encrypted and sensitive data may be visible to non-privileged entities, i.e. when communication is routed through public or wireless networks. Therefore, DCOM allows configuring marshalling to be encrypted.

For running DCOM with encryption, an enabled DCOM Authentication is mandatory. In order to run DCOM with encryption all steps explained in the chapter [Working with enabled DCOM Authentication](#) have to be fulfilled. Furthermore the registry value “EnablePacketPrivacy” under “HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA” has to be set to “1” (Default is 0).



In order to encrypt security relevant data it is mandatory to modify the server side “EnablePacketPrivacy” value. The change of this value on the client side is not needed.

As encryption algorithm RC4 with a key length of 128bit (supported by OS till Windows 2000 service pack 2 and newer) is used. This encryption algorithm is provided by the DCOM Authentication Service “NTLMSSP”.

Which server processes to customize?

The decision which server processes to setup depends from different aspects:

➤ The following process have always to be customized:

1. DPE PPRServer
2. DPE Server Tools

➤ On a **Master** server additionally the following processes have to be customized:

1. DPE Pooling Server
2. DPE Lock Manager
3. DPE Update Manager

➤ When working with DPM clients the **DPE EPGenericServices** process has to be customized on **Slave** and **Master**.

➤ When working with **SSO** (Single Sign On) the **DPE EPSSOService** process has to be customized on **Slave** and **Master**.

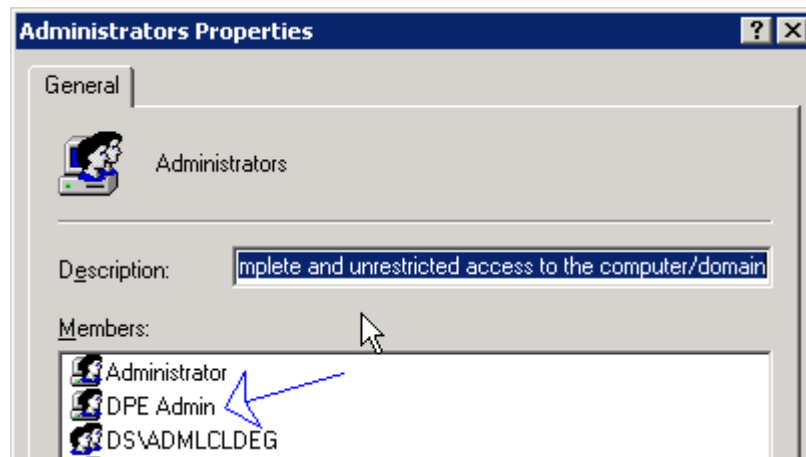
➤ When working with enabled **P&O Logging** or **Logging of Access to Export controlled data** additionally the **DPE EPLogger** process has to be customized (customer is free to decide where to run, on the Master only, on all server machines or ...).

Assign local administrator rights to a user

- On server machine select **Start / Control Panel** from the Windows taskbar.
- Start **Administrative Tools** in Control Panel view or menu.
- Start **Computer Management** in **Administrative Tools** view.
- Select **System Tools** for managing 'Local Users and Groups'.



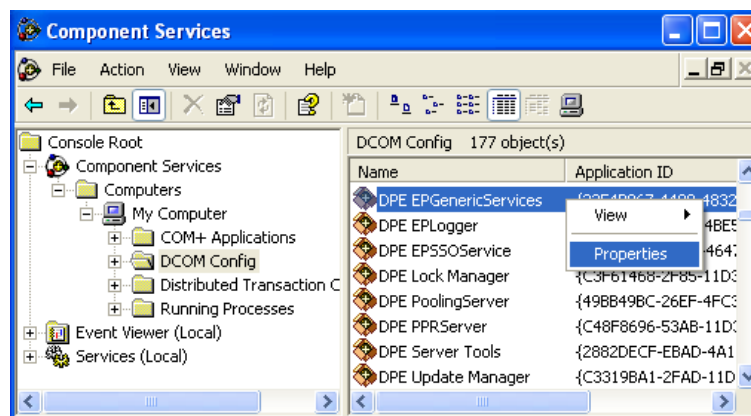
- Add Domain User here **DPE Admin** to group of 'Administrators' on server machine



How to Launch DCOM configuration Tool

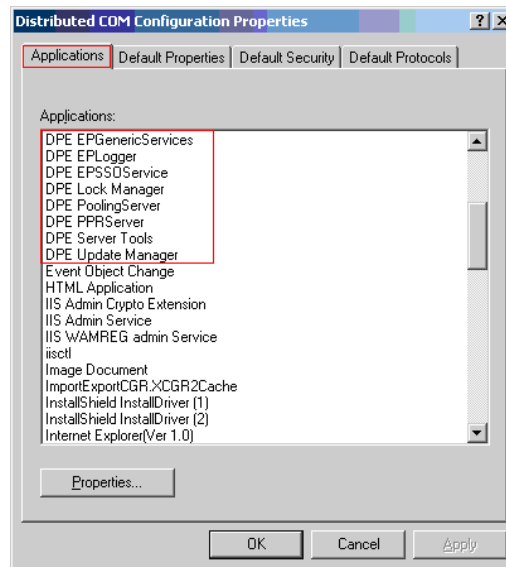
On Windows XP and Windows Server 2003

- Select **Start / Control Panel** from the Windows taskbar.
- Start **Administrative Tools** in **Control Panel** view or menu.
- Start **Component Services** in **Administrative Tools** view.
- The **Component Services** window appears. Navigate through the Component Services tree as shown by the following picture. (When selecting the **DCOM Config** item confirmation dialog boxes may be displayed, asking you to accept some key numbers. Accept the numbers by selecting Yes in these dialog boxes).



On Windows 2000

- Select **Start / Run** from the Windows taskbar.
- Enter the command "**DCOMCNFG**" in the Run dialog box, and click OK to launch the program (When DCOMCNFG is launched it may display a confirmation dialog box asking you to accept some key numbers. Accept the numbers by selecting **Yes** in these dialog boxes).
- The Distributed COM Configuration Properties dialog box appears:



How to Launch Local Policies configuration

- Select **Start / Control Panel** from the Windows taskbar.
- Start **Administrative Tools** in **Control Panel** view or menu.
- Start **Local Security Policies** in **Administrative Tools** view.
- The Local Security Policies window appears. Navigate through the Component Services tree as shown by the following picture.



How to enable DCOM?

By default, as shipped by Microsoft, DCOM is enabled on OS platforms. To enable DCOM on the OS the followings steps have to be performed:

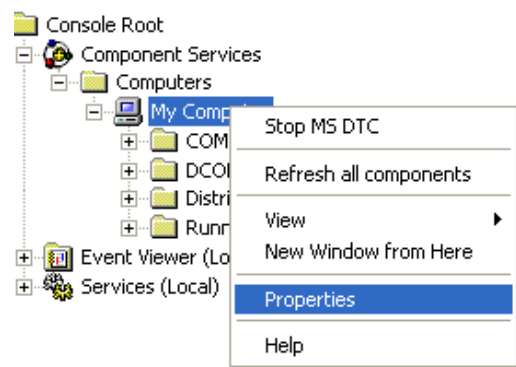
- ➔ Launch the DCOM configuration Tool. See [How to Launch DCOM configuration Tool](#) section.

Windows 2000

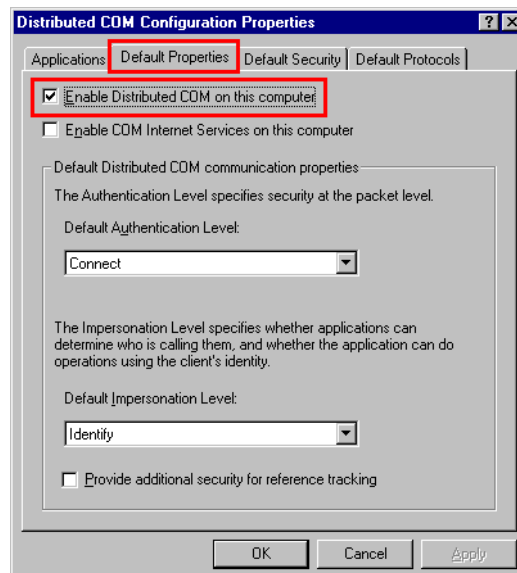
- ➔ On **Windows 2000** you are now in the right dialog.

Windows XP/ Windows Server 2003

- ➔ On **Windows XP/ Windows Server 2003** you have to navigate in the tree to **My Computer** item and select **Properties...**



- ➔ Select the **Default Properties** tab and check if DCOM is enabled.



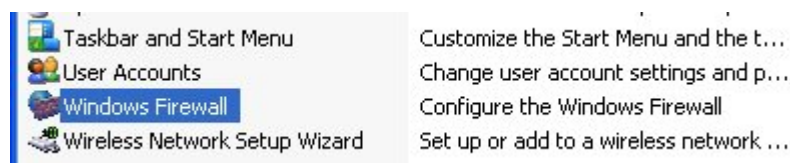
Customize Windows Firewall

An enabled Windows Firewall prevents by default the start of incoming connections. In the context of the DPE client an enabled Windows Firewall means customization effort is required. Otherwise the callbacks (First chapter) used by the solution will not work properly and Error messages rights after the Login appear.

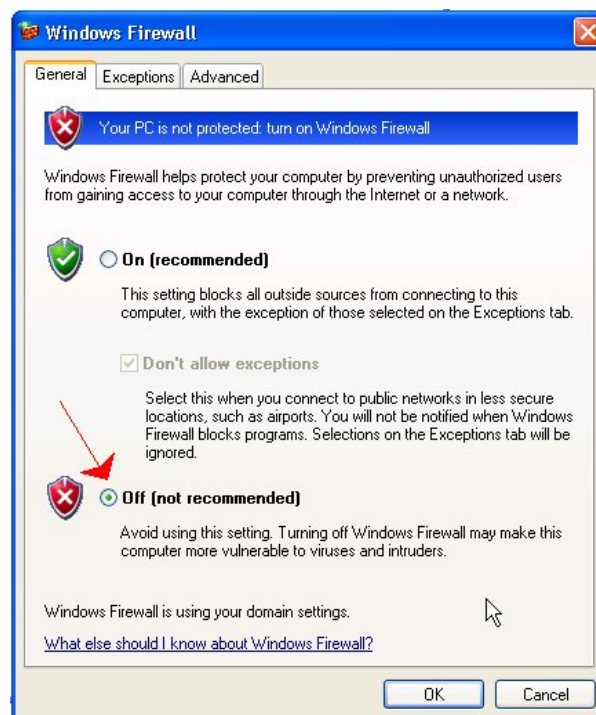
For setting up a DPM client this section can be skipped. The DPM client works without callbacks therefore no customization effort is required here.

How to disable Windows Firewall?

- ➔ Select **Start / Control Panel** from the Windows taskbar.
- ➔ Start **Windows Firewall** in **Control Panel** view or menu.

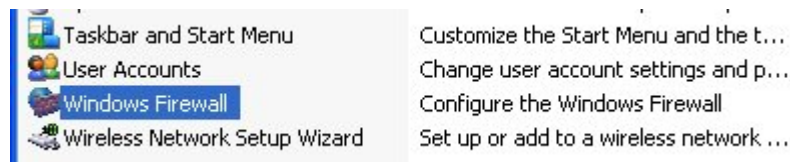


- ➔ Select **Off** in General Page in Windows Firewall Properties Dialog

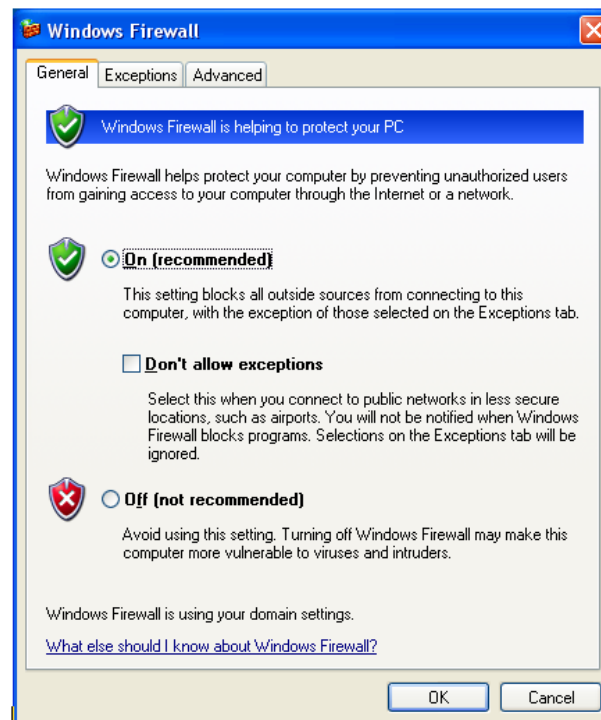


Enable DPE Client to run on Windows XP SP2 with an enabled Windows Firewall

- Select **Start / Control Panel** from the Windows taskbar.
- Start **Windows Firewall** in **Control Panel** view or menu.

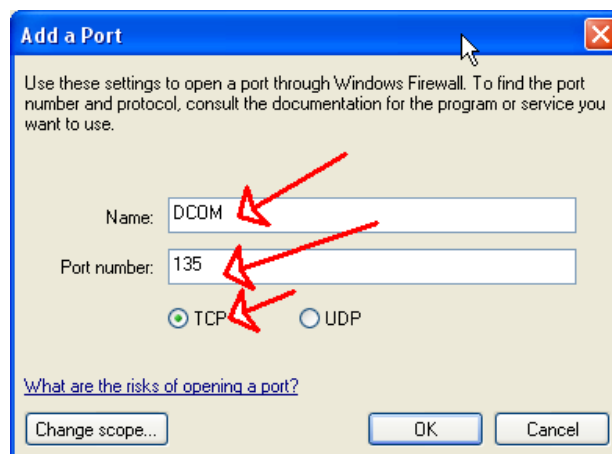
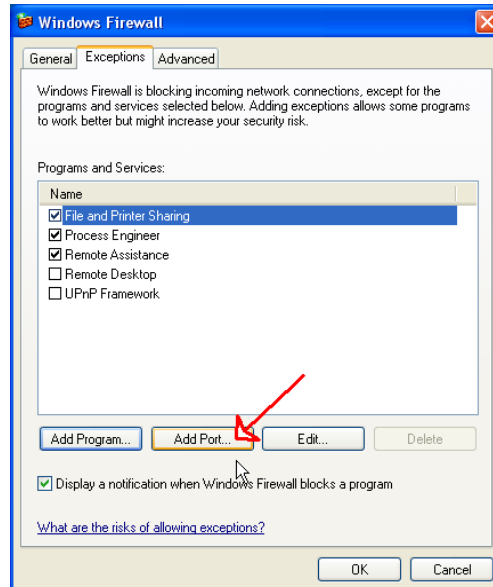


- Select **On (recommended)** in General Page in Windows Firewall Properties Dialog



Enable DPE Client to run on Windows XP SP2 with an enabled Windows Firewall

- Press button **Add Port...**
- Enter as port name: DCOM.
- Enter as port number: 135.
- Select TCP protocol and confirm with OK.

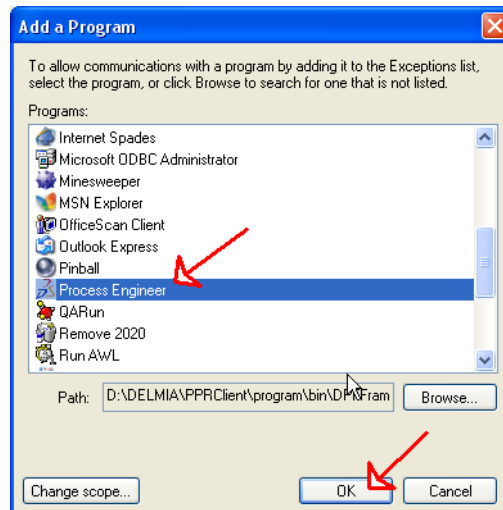
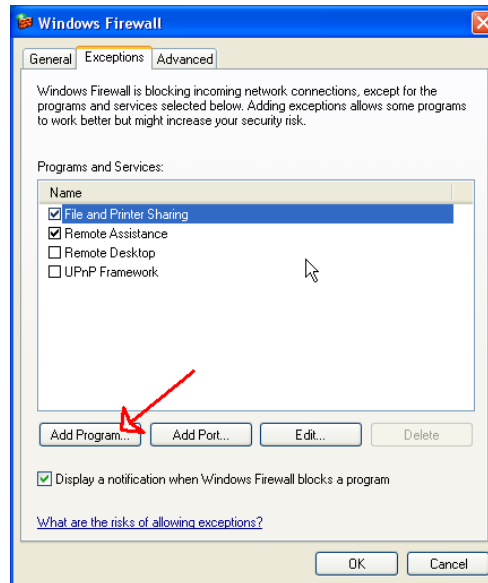


- Go to page Exceptions and press Button 'Add Program...' and select Process Engineer [DPFFrame.exe] in list of programs.
- Confirm with OK.

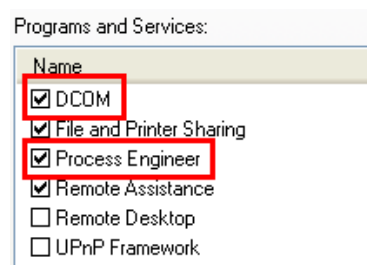


NOTE: If you want to use DPE Balancing application; add also EPBalancing.exe in exceptions list.

Enable DPE Client to run on Windows XP SP2 with an enabled Windows Firewall



- ➔ Please confirm that DCOM and Process Engineer is set to true in list of exceptions

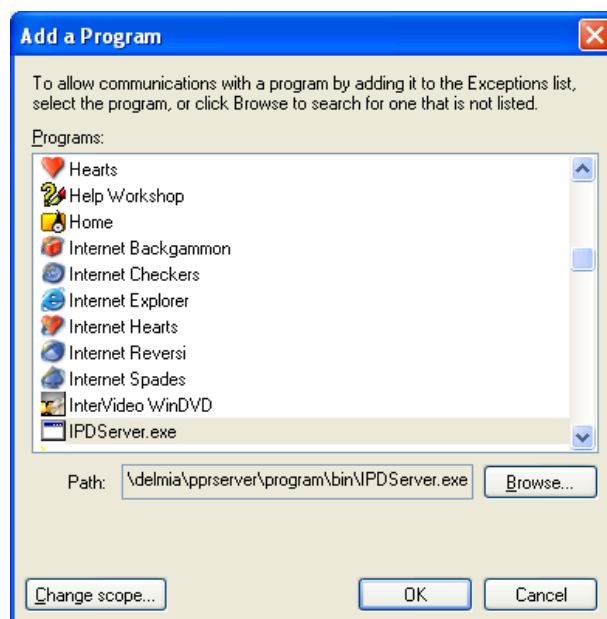


Enable Server Processes to run on Windows Server 2003 SP1 with an enabled Windows Firewall

For the setup of the Server Processes on Windows Server 2003 SP1 with enabled Windows Firewall the same steps are required, as for the DPE client on a Windows XP SP2 with enabled Windows Firewall. The only difference is, instead of customizing exceptions for Process Engineer and the Balancing module you have to customize exceptions for the server process.

For which server processes you need to configure exceptions, please refer to chapter [Which server processes to customize?](#)

Example DPE PPRServer (... \PPRServer\program\bin\IPDServer.exe)



Installation paths for the remaining server process:

- DPE Server Tools (... \PPRServer\program\bin\epservertools.exe)
- DPE Pooling Server (... \PPRServer\program\bin\leppoolingserver.exe)
- DPE Lock Manager (... \PPRServer\program\bin\LockMng.exe)
- DPE Update Manager (... \PPRServer\program\bin\UpdateMng.exe)
- DPE EPGenericServices (... \PPRServer\program\bin\epgenericservices.exe)
- DPE EPLogger (... \PPRServer\program\bin\EPLogger.exe)
- DPE EPSSOService (not used remote, no customization required)



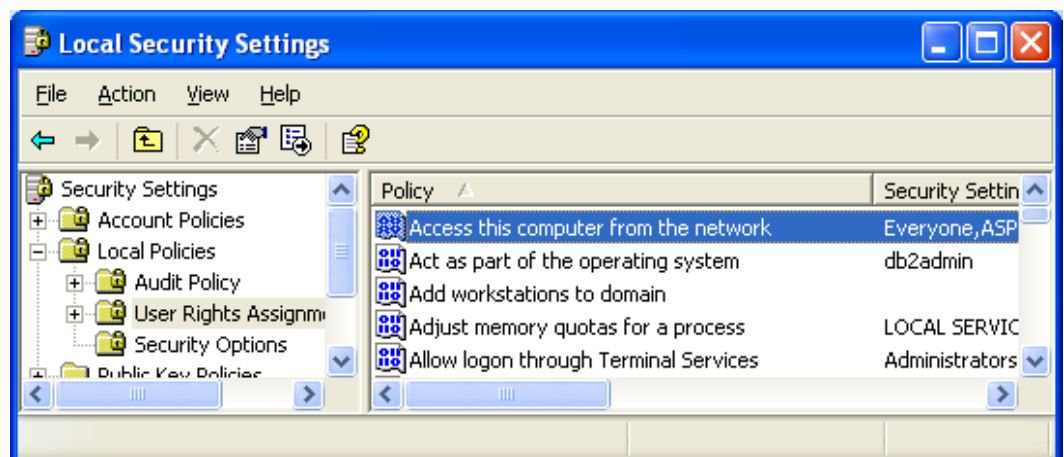
NOTE: Don't forget setup the exception for the port 135 ([see chapter before](#))

Checklist for connection problems

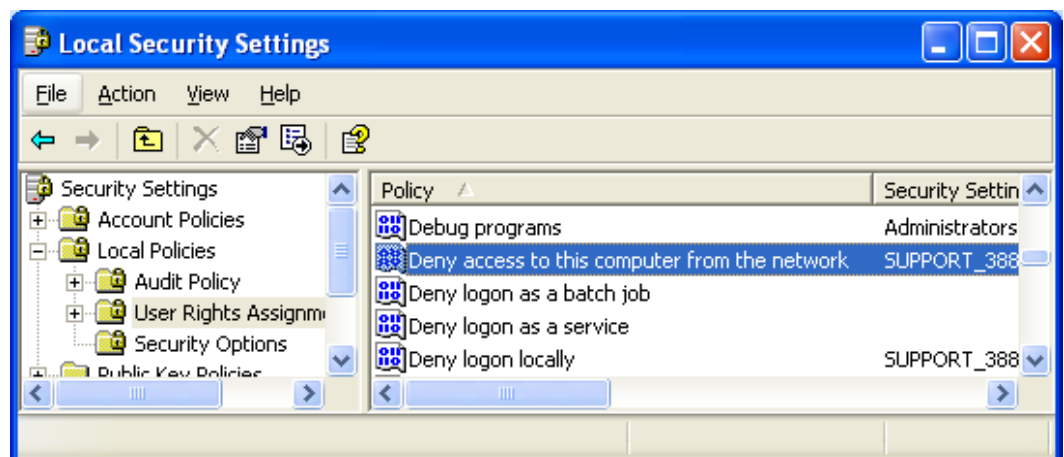
The settings explained here in this section are shipped by Microsoft by default in a way which allows our solution to work. This is the reason, why they haven't been mentioned in the previous chapters.

Prerequisites are:

- DCOM has to be [enabled](#) on Client and Server side (By default it is enabled).
- With Windows XP SP2, Windows Server 2003 SP1 and Windows Server 2003 64 Bit a **Windows Firewall** is shipped with the OS. The standard for Windows XP SP2, when it is installed, is an enabled Firewall. For the other mentioned OS it is disabled. Our solution is by default not able to run over Firewall. Therefore you have two options [disable the Firewall](#) or [Customize the Firewall](#) to run with the solution.
- The computer has to be accessible from network (The setting need to be fulfilled for server machines always, for client machines in case DPE client is running on)

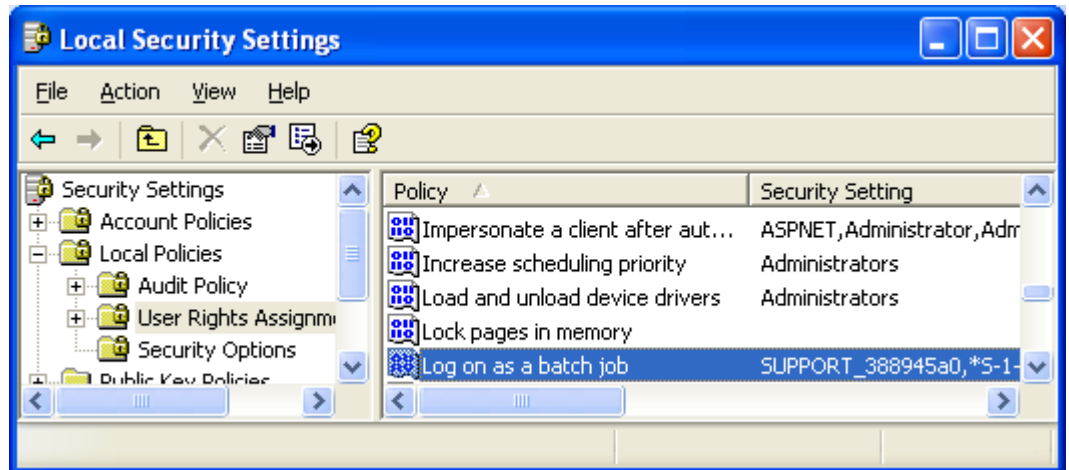


This setting has to be consistent with the **Deny access to this compute from network** setting:

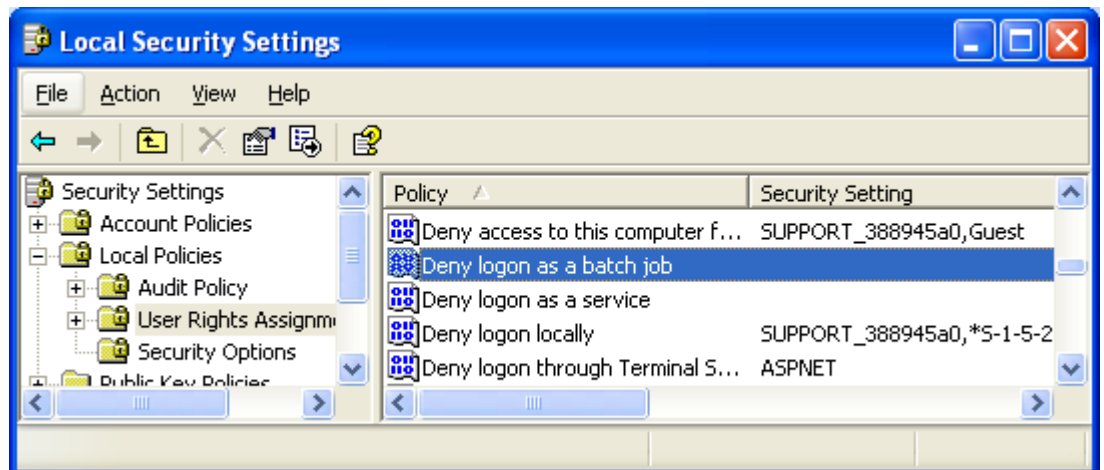


Enable Server Processes to run on Windows Server 2003 SP1 with an enabled Windows Firewall

- The user identity under which the server processes run has to have right to **Log on as a batch job** (By default, when the identity of the first process is set in the DCOM configuration, the user is added to list of users that have this right)



This setting has to be consistent with the **Deny log on as batch job** setting:



DCOM settings for multiple clients on Novell networks without a Windows Domain Controller

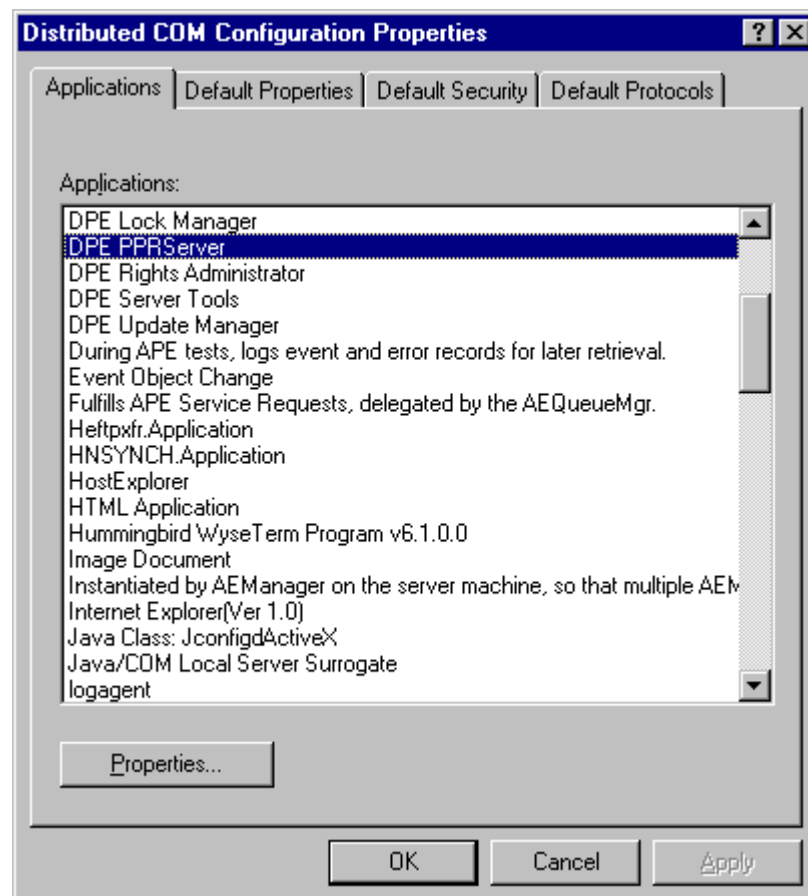
Since there is no domain when working in a Novell environment, there is no domain user available. In this case, a local user on the PPR server must be authorized to launch DCOM processes.

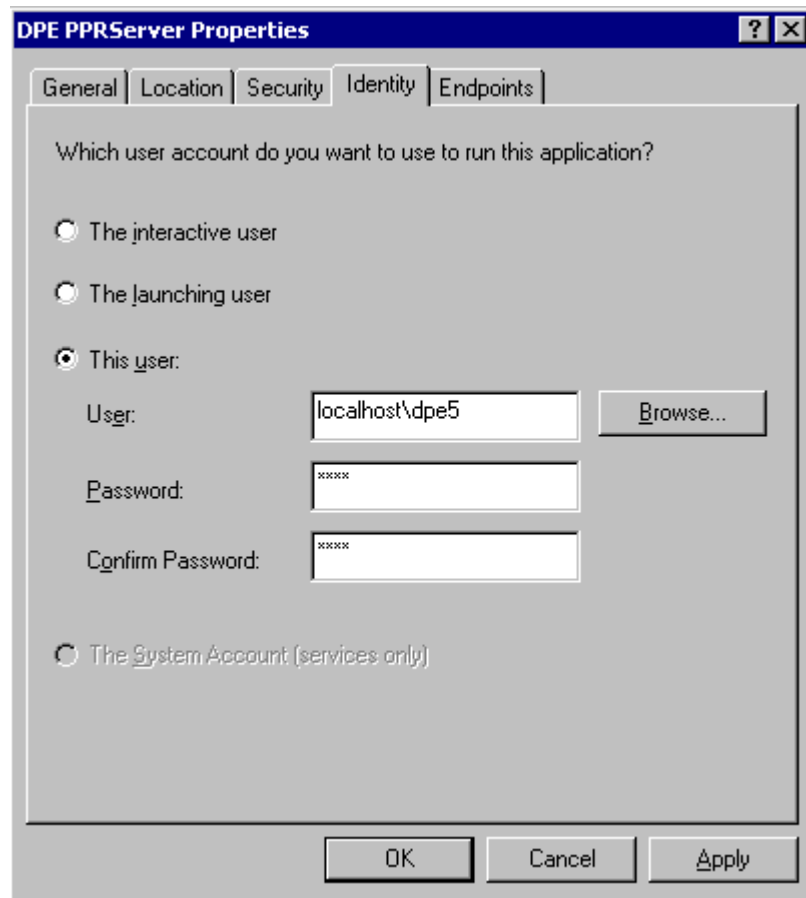
- In this case, the user for DCOM identity stays the same.
- On the clients, this user must be created. This means that if a local user "DELMIA_DCOM" (for example) is used for DCOM identity on the server, a user "DELMIA_DCOM" must be created with the same password on each PPRClient.

This solution is possible in Windows domains if there is no domain user available. For example:

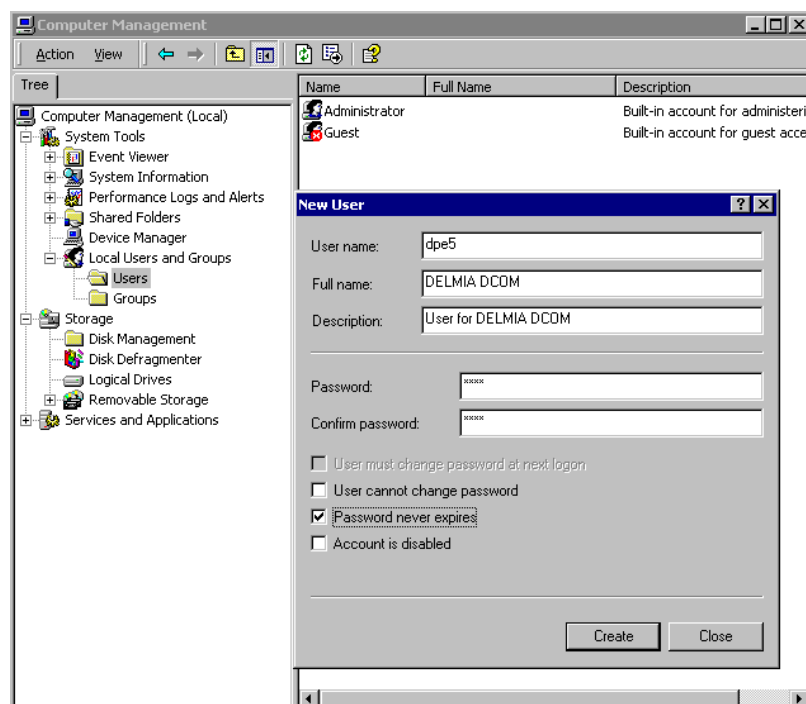
- username: **dpe5**
- password: **dpe5**

In a Novell network without a Windows Domain, the user dpe5 (in this example) must exist on the server machine as a local user with administrator privileges and on all clients as a local user with the same password as on the server.





- ➔ On the client machines, the user *dpe5* with password *dpe5* must also be created:

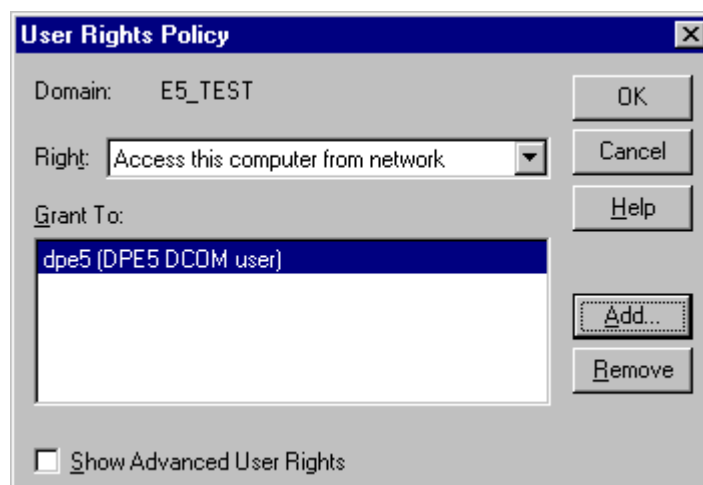
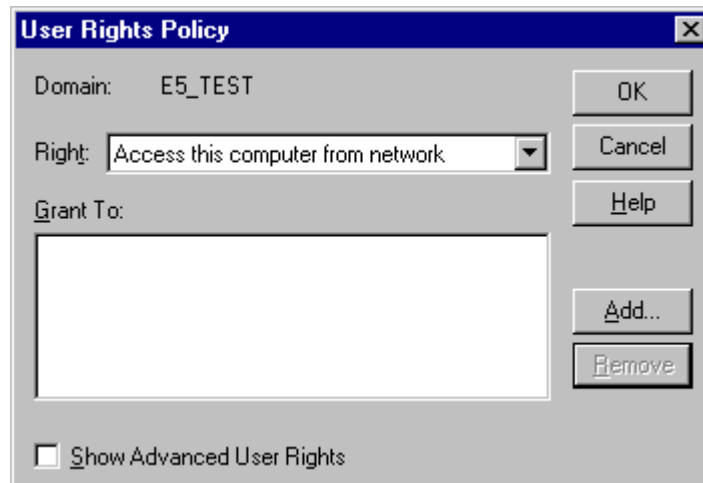


Authentication

For authentication from server to client, an additional policy for accessing the client must also be set.

On Windows, select

User management / Policy / User Rights / Access this computer from network / Add / user "dpe5" (the DCOM identity user):



Note that the dpe5 username and password are only used as examples in the scenario above. Any name may be used; however it is important to use the same username and password for the client, server and DCOM-identity on the server.

DCOM HTTP Tunneling

General

In order to use DCOM HTTP Tunneling several installation requirements to activate MS CIS (COM Internet Services) and RPC over HTTP must be followed (details see section setup and client and server machine configuration below).

The new connection modus has to work using the COM garbage collector. Connecting a client has to receive an interface to a connection object, as long as the connection object exists the connection is valid. As soon as the connection object is destroyed, either by release from the client or by the COM Garbage Collector, the connection object has to kick off the normal server deregister mechanism.

DCOM HTTP-Tunneling can only be used for client-server connections. Connections between server processes on several machines (DPE Master/Slave) cannot be tunneled. Considering this scope a firewall protection allowing only traffic on HTTP port 80 can only be installed between client and server. If a firewall has to be placed between several server machines it must allow all DCOM port traffic. Respecting security requirements the location of several server machines within the company network should be choose carefully.

Using HTTP-Tunneling has an impact on the DPE system performance depending on the kind of data transferred over the client server connection and the network latency time and bandwidth.

Functional changes Server:

For the new connection modus additional points have to be ensured:

- Disable the periodical check of the server to ensure if client is still alive.
- Disable the server termination notification mechanism. The server notifies client about the server termination or the termination done via the Server-Tools.
- Disable the HasCommitToBeDone mechanism. Server notifies client about to change status of the Save button (disabling and enabling Save button).

Functional changes Client:

- In the EPIPDClient module provide a possibility to connect to server using the new connection modus.
- Disable registration of server callback in the UpdateDispatcher module.
- Disable registration of update callback in the ConfigfactoryCache module. In generally this means that as long as clients are connected to the system no changes to customization data are allowed.

Clients that use COM HTTP Tunneling don't receive

- Update messages.
- Notifications about the termination of server.
- Notifications about the termination of their transactions via ServerTool.
- No com objects created on client side can't be passed to the server.
- The Auto relation asynchrony callback mechanism can't be used.
- If the client is remotely terminated from the ServerTools-Client, it does not receive a confirmation message after disconnecting from the PPR server. The PPR client will only recognize the disconnection status if the user tries to fetch data which are still not cached on clients side or if calling other server functions directly. In this case the client will display an error message indicating an invalid transaction.

Functional changes ServerTools-Client:

The ServerTools-Client can be switched to a callbackless connection mode by registry:

Key: HKEY_LOCAL_MACHINE\SOFTWARE\DeImia\EPSToolsUI

entry: connectionmode

value:

0 – using callbacks for a non-tunneled server connection

1 – callbackless server connection supporting HTTP-Tunneling

Restrictions for connectionmode = 1:

- ❑ The ServerTools-Client does not support the interactive client termination, instead the immediate termination of the server connection is proceeded. ServerPool and Machine nodes in the navigator tree are not updated automatically.
- ❑ A lower updating frequency for the server process view and the load index view is used:

low:	6 sec.
normal:	3 sec.
high:	1 sec.

The ServerTools-Client does not support the interactive client termination, instead the immediate termination of the server connection is proceeded

ServerPool and machine nodes in the navigator tree are not updated automatically

Lower updating frequency for the server process view and the load index view

The Process Monitor view and the PoolingServer LoadMonitor view are refreshed only in the active window – to actualize data in inactive views these windows must be set as foreground window first.

Setup

If DCOM-HTTP Tunneling should be used for a client server connection, the following system requirements must be fulfilled:

System requirements

Client machine:

- Windows2000 Workstation

Server machine:

- Windows2000 Server
- IIS 5.0 (Internet Information Server)
- RPC-Proxy installed

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndcom/html/cis.asp>

Consider installation and configuration of system internet components (MS Internet Explorer) may affect proceeding of HTTP tunneled DCOM RPC calls:

Known influences are:

- Proxy Server for Internet Explorer (security and server domain name resolution)
- Configuration of the Default Webpage in IIS
- PPC-Proxy configuration in IIS (access permissions)

Client and Server machine configuration:

The DCOM HTTP-Tunneling is supported by Windows 2000 and must be activated on each client and server machine. To activate the Tunneling communication protocol the DCOM configuration tool dcomcnfg.exe is used. After opening dcomcnfg.exe in the property tab 'standard protocols' the DCOM protocol 'Tunneling TCP/IP' must be added. The protocol order is different for a client and a server machine:

Protocol order client machine:

- Tunneling TCP/IP
- Connection oriented TCP/IP
- Other protocols

Protocol order server machine:

- Connection oriented TCP/IP
- Tunneling TCP/IP
- Other protocols

For a server machine the Tunneling protocol must be listed below the TCP/IP protocol because communication between server processes does not support Tunneling. The communication protocol will be selected by the first protocol match for the communication partners.

In the dcomcnfg property tab 'standard properties' the check box:

Activate COM Internet services on this computer:

- ☞ Must be enabled.
- ☞ After setting up the DCOM protocols the system must be rebooted.
More information about the DCOM Tunneling configuration:

Client machine:

<http://support.microsoft.com/default.aspx?scid=kb;en-us;265340>

Server: machine:

<http://support.microsoft.com/default.aspx?scid=kb;EN-US;282261>

The DCOM standard protocol setup is effective for all DCOM components of the machine. Setting up the machine standard protocol list is the only valid configuration for client and server machines.

Client specific system configuration:

- If using a Proxy Server (Internet configuration), it must be configured to allow HTTP connect on port 80 without authentication.
- To establish a http tunneled server connection, the client must be able to resolve the server domain. The server domain name resolution can be provided by a domain name server or can be registered in the Windows system32 HOST file. The name resolution can be checked by typing <server_machine>.<domain>
- In the MS Internet Explorer – the server default web page should be displayed. Consider the proxy server configuration for the availability of the server domain resolution.
- Copy registry key
"HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings" to "HKEY_USERS\S-1-5-18\Software\Microsoft\Windows\CurrentVersion\Internet Settings" and "HKEY_USERS\S-1-5-20\Software\Microsoft\Windows\CurrentVersion\Internet Settings".

Server specific system configuration:

- The connection timeout in the IIS Default Webpage must be specified high enough to cover the actual network availability (minimum 300 sec.).
- The RPC-Proxy must be listed as ISAPI-Filter for the server machine in the IIS (configured by the Windows2000 COM Internet Services Proxy installation of Networking Services).
- The DCOM security settings for the DPE server processes remain valid if DCOM Tunneling is used and must be set equally for all server processes as described in the DPE DCOM security configuration.
- Add the following registry entry
KEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Rpc\RpcProxy
Entry: "Allow Anonymous"
Type: DWORD Value
Value: 00000001
More Information in: <http://support.microsoft.com/kb/833003/en-us>.
- Check the registry entry
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Rpc\RpcProxy\ValidPorts" value should be configured as follows :
<Server-name>:593;<Servername>.<Domainname>:593;<ServerIPAddress>:593;
<Servername>:100-5000;<Servername>.<Domainname>:100-5000;<ServerIPAddress>:100-5000.
- Replace <...> names where <ServerIPAddress> is IP address of the server machine.

Example

A WAN with a firewall has to be simulated or installed to test the new connection scenario. Also a V5 client has to be provided that uses the new method.

To test the HTTP tunneled client server connection, the firewall placed between client and server has only to allow traffic on port 80. It is not possible to install a firewall between several server machines – the inter-server communication cannot be tunneled. If a firewall must be installed between two server machines the normal DCOM traffic must be allowed. The V5 client should load, save and do all operation that normally works.

To prepare the Test DCOM HTTP Tunneling must be activated on all client and server machines. Tunneling must be activated on the Windows system and additionally the E5 client installation must be switched for Tunneling usage.

To check if the RPC-Proxy is installed in the server machine, in the IIS console (Windows administration -> Internet Information Services) within the standard web page entry, a node RPC must be displayed with an entry *rpcproxy.dll*.

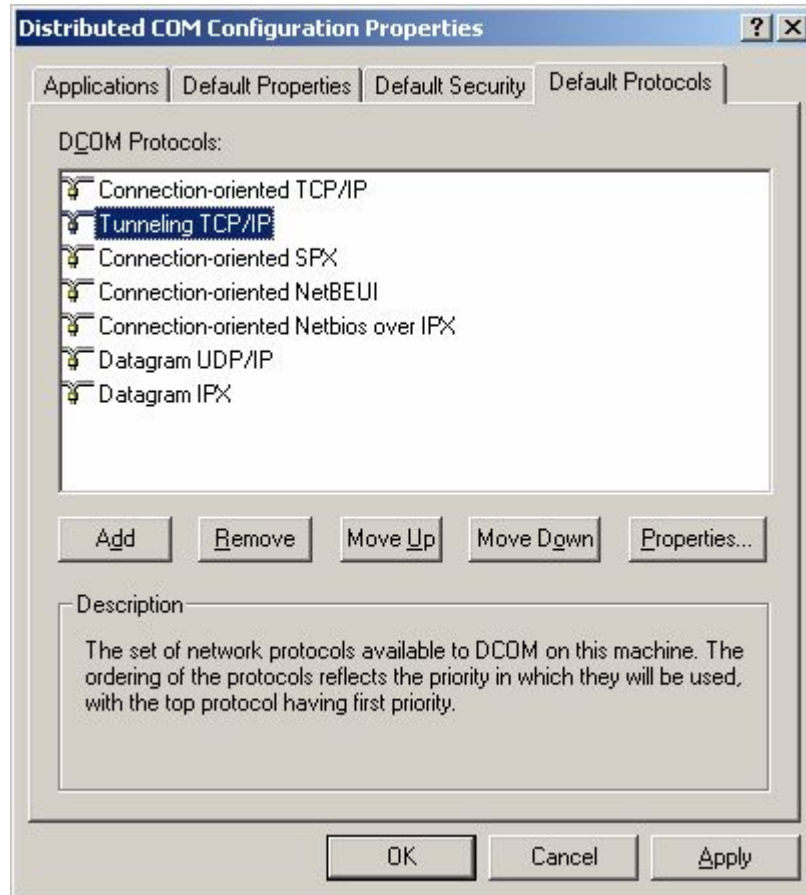
From the client machine, the default web page of the IIS on the server must be accessible - to check server access open the Internet Explorer on the client and enter:

- <server_name>.<server_domain>
- (replace <...> with the names)
- the default webpage (e.g. 'under construction') should be displayed.

1) Enable DCOM Tunneling mode in the PPR client installation by setting the registry entry 'connectionmode'

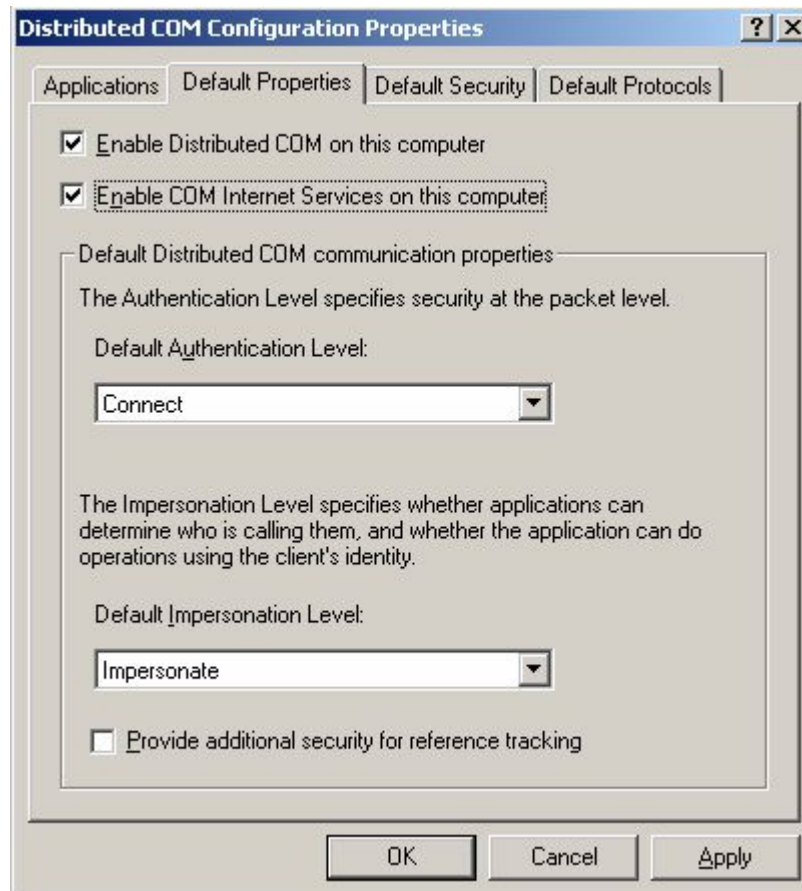
- HKEY_LOCAL_MACHINE\SOFTWARE\Delmia\ergoplan\
connectionmode = 1
- HKEY_LOCAL_MACHINE\SOFTWARE\Delmia\EPServerToolsUI\
connectionmode = 1

- 2) Enable Windows DCOM Tunneling protocol on the server machine:
 - open dcomcnfg.exe on the server machine
 - select standard protocol tab:



- Add the protocol 'Tunneling TCP/IP'
- Move the Tunneling protocol on second position after 'Connection oriented TCP/IP'

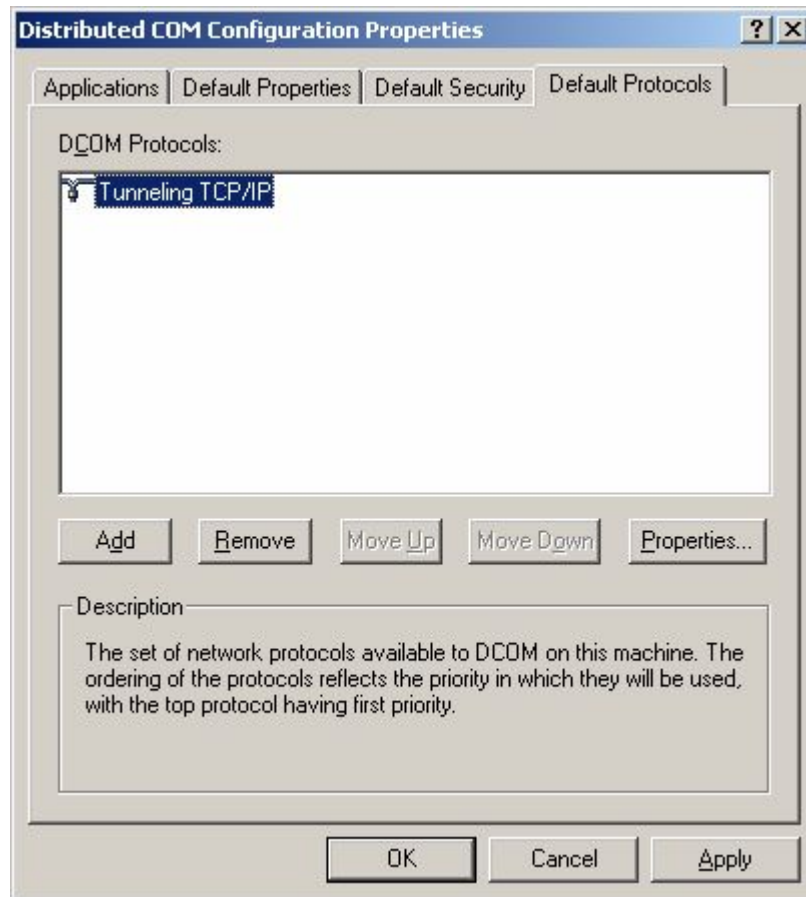
- ➡ Select Default Properties tab:



- ➡ Enable checkbox 'Activate COM Internet Services on this computer'
- ➡ Reboot the server machine

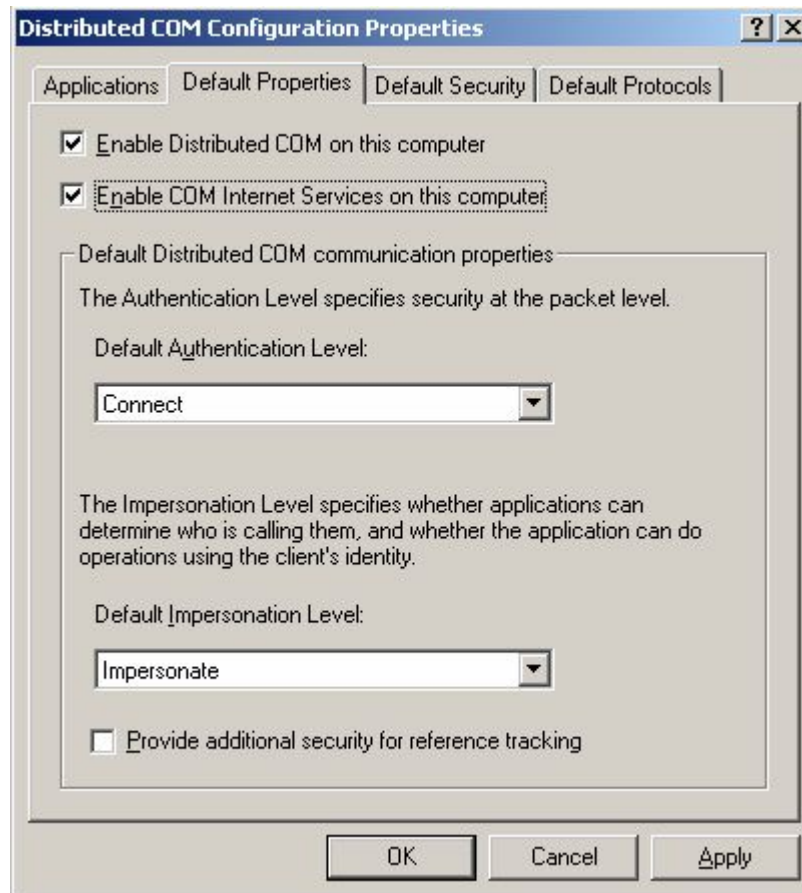
More information in: <http://support.microsoft.com/default.aspx?scid=kb;EN-US;282261>

- 3) Enable Windows DCOM Tunneling protocol on the client machine:
 - Open dcomcnfg.exe on the client machine
 - Select standard protocol tab:



- Add the protocol 'Tunneling TCP/IP'
- Move the Tunneling protocol on top position
- To be sure that no other protocol is used on the client for the test, remove all other protocols

- ➔ Select Default Properties tab:



- ➔ Enable checkbox 'Activate COM Internet Services on this computer'
- ➔ Reboot the client machine

More information in

<http://support.microsoft.com/default.aspx?scid=kb;en-us;265340>

Additional a robustness test has to be done. The following issues have to be evaluated:

How does the Garbage Collector of COM work?

How robust is the network connection? Is it very sensitive or does the IIS handle Communication problems?