Steps to Install MtcOpcAgent for Siemens 840D

Thisi document describes how to install the 64-bit implementation of MtcOpcAgent for reading status from a Siemens 840D Powerline CNC using remote Simnumerik OPC connection to read data from CNC.

1. Ping remote machine to make sure it is running and available on the network.
2. Now comes the bad part – DCOM. Use TestDCOM to make sure you can ping the CNC ip, and then CONNECT to create a connection to the OPC server. If not, don't bother going any farther, as you will need to "fix" you DCOM permissions.
3. Install OPC Core Components 3.00 Redistributable (x64)

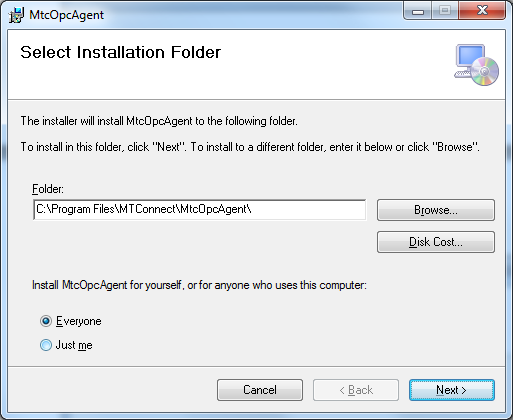
Rename "OPC Core Components Redistributable (x64).msx"

to "OPC Core Components Redistributable (x64).msi", double click and install.

1. Microsoft Visual C++ 2010 Redistributable Package (x64) installs runtime components of Visual C++ Libraries required to run applications developed with Visual C++ on a computer that does not have Visual C++ 2010 installed. You can find it here: <http://www.microsoft.com/en-us/download/details.aspx?id=14632>

Rename vcredist\_x64.exg to vcredist\_x64.exe and run.

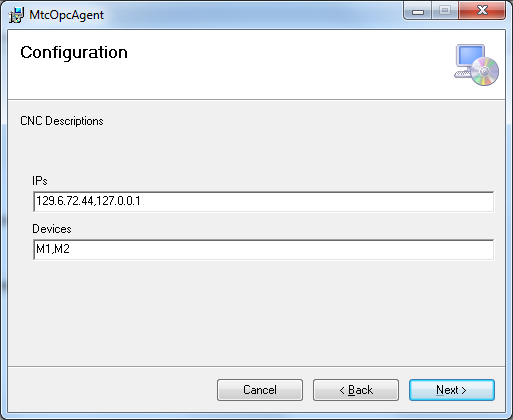
1. Install the Agent (which reads status data from the CNC using OPC and displays using http). Run the installation msi script.



1. Input the ip of the CNC and the name you want to use to describe it as a device, e.g., M2132 (no spaces or fancy characters please!), for example:

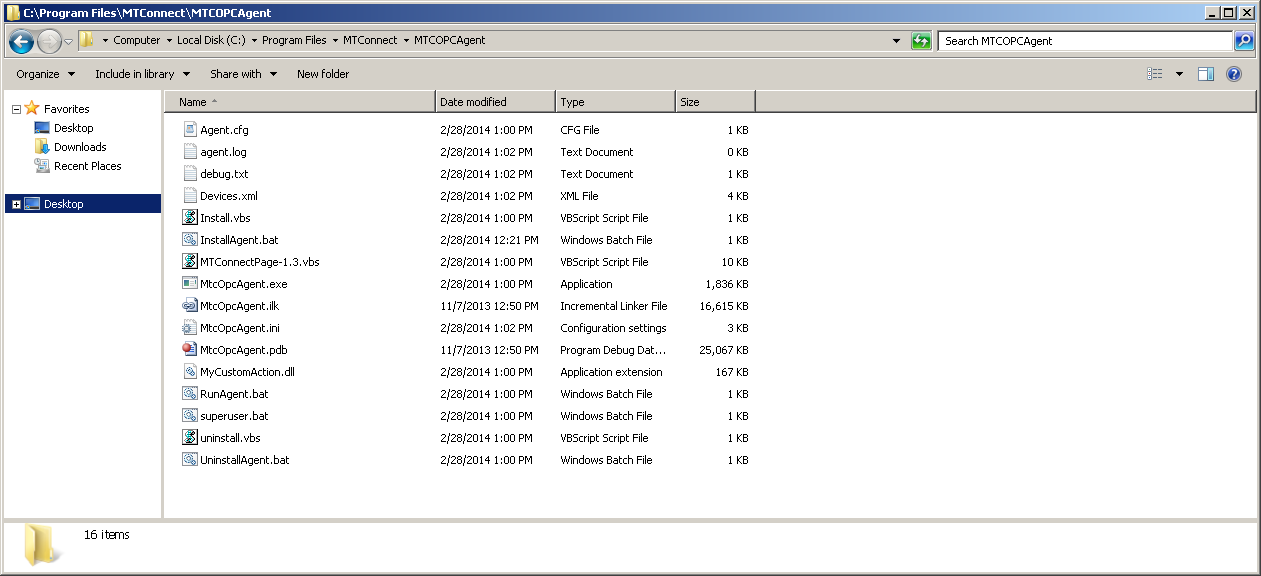
192.168.24.4,127.0.0.1

M1,M2

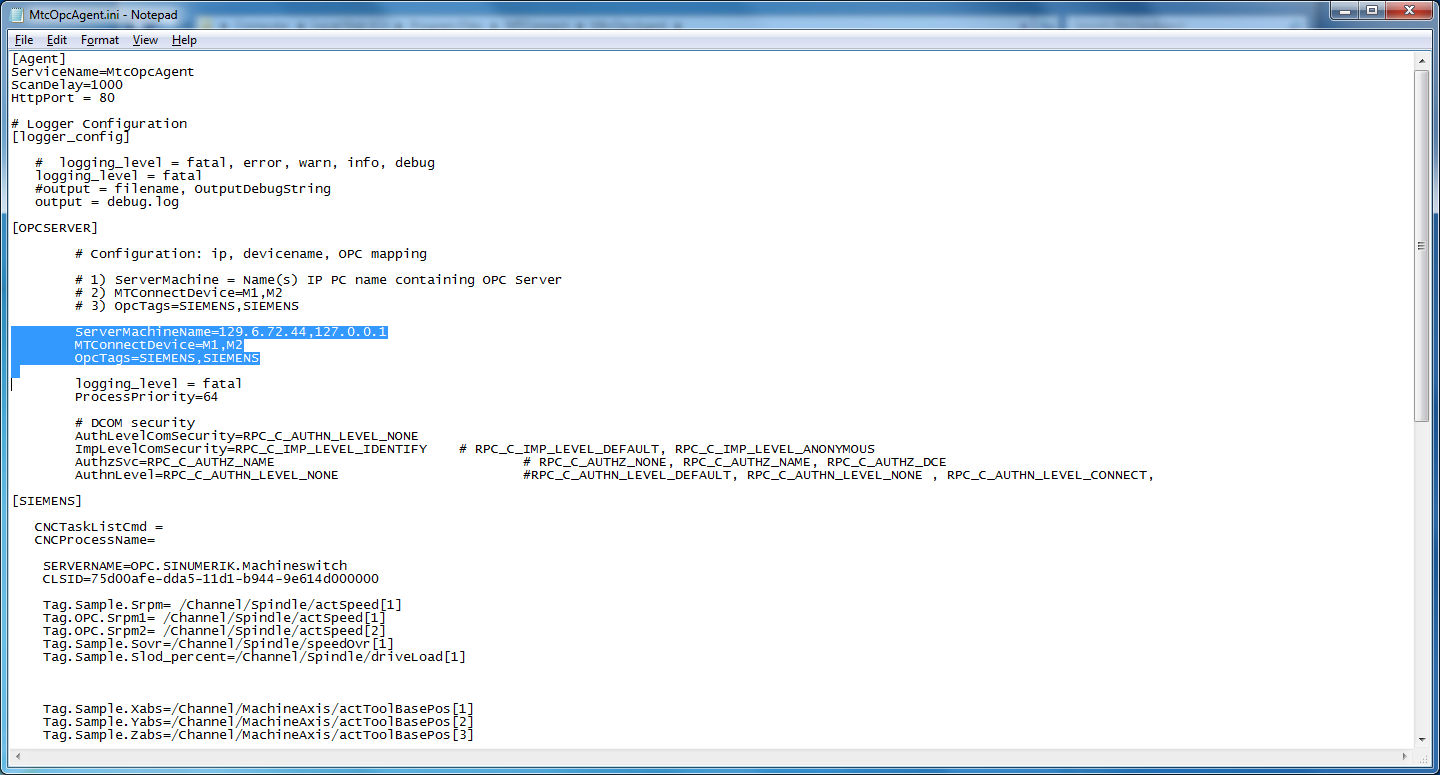


1. Next you will need to verify that the MtcOpcAgent has been installed as a service, and then start it (because you probably didn't have sufficient privileges to install services onto the Windows 7 box.

First, check if the service is not installed, navigate to the folder "C:/Program Files/MTConnect/MtcOpcAgent" and find the MtcOpcAgent.exe.

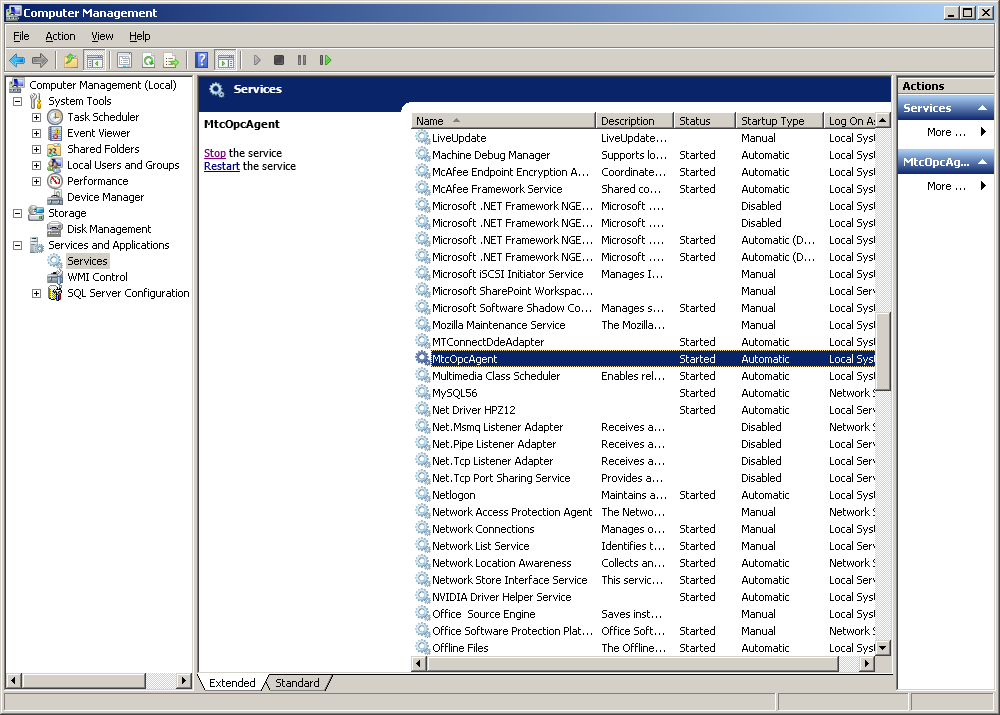


Verify that the configuration parameters are correct, open MtcOpcAgent.ini in notepad, and confirm highlighted text below matches what was entered during installation:



Next, check to see if MtcOpcAgent service is in Windows Service Control Manager (SCM):

Right-click My Computer -> Services and Applications -> Services



If MtcOpcAgent is not in SCM, install it:

Runas Install.bat as administrator (right click the bat file and click run as administrator).

🟏🟏🟏 PLEASE READ 🟏🟏🟏

You need to let the MtcOpcAgent.exe be allowed to pass through the firewall. DCOM needs port 135 opened. Let MtcOpcAgent.exe , but Runas administrator RunAgent.bat (and respond yes to the Firewall question to allow it through) or follow these directions from Microsoft:

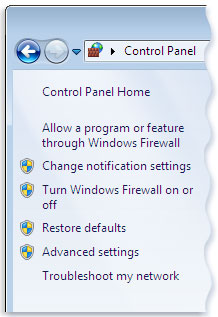
Note: You need to let the MtcOpcAgent.exe be allowed to pass through the firewall. DCOM needs port 135 opened. and MtcOpcAgent.exe opened in the firewall.

Runas administrator RunAgent.bat (and respond yes to the Firewall question to allow it through)

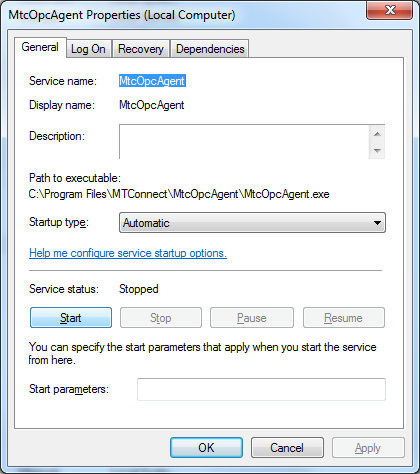
Or follow these directions from Microsoft:

### To allow a program to communicate through Windows Firewall

1. Open Windows Firewall by clicking the **Start** button Picture of the Start button, and then clicking **Control Panel**. In the search box, type **firewall**, and then click **Windows Firewall**.
2. In the left pane, click **Allow a program or feature through Windows Firewall**.

Left pane of Windows Firewall

1. Click **Change settings**. Administrator permission required If you're prompted for an administrator password or confirmation, type the password or provide confirmation.
2. Select the check box next to the program you want to allow, select the network locations you want to allow communication on, and then click **OK**.
3. Start the MtcOpcAgent service: double click the service entry and click start



Hopefully it has started and there are not problems.

## Multiple Spindles

Mutliple spindles flag and some OPC read value tags:

MultiSpindle= 1

. . .

Tag.Sample.Srpm= /Channel/Spindle/actSpeed[1]

Tag.OPC.Srpm1= /Channel/Spindle/actSpeed[1]

Tag.OPC.Srpm2= /Channel/Spindle/actSpeed[2]

In the above ini file, Tag.OPC.xxx means to read the value from the Siemens OPC controller into value xxx. The Tag.Sample.Srpm means the Srpm variable is reported to MTConnect. Bit of a kludge but handles the problem. When MultiSpindle ini flag is one it means to read both spindles’ values, and if one was non- zero use it. No one asked which spindle was active, etc.

The C++ code works but makes assumptions that only one spindle is active at a time.

if(\_nMultiSpindle)

{

int srpm = 0;

int srpm1 = opcitems.GetValue<int>("Srpm1", -1);

int srpm2 = opcitems.GetValue<int>("Srpm2", -1);

if (srpm1 > 0) srpm=srpm1;

if (srpm2 >0 && srpm2 > srpm1) srpm = srpm2;

opcitems.SetItemValue("Srpm", (long) srpm);

}

## OPC/DCOM Background

The Microsoft Component Object Model (COM) is a distributed, object-oriented system for creating binary software components that can interact. OPC DCOM is based on Microsoft COM/DCOM (Distributed Component Object Model). DCOM is in turn based on the DCE/RPC (Distributed Computing Environment/Remote Procedure Calls) “standard”. COM uses a proxy based communication strategy, that is, it offers an object-oriented client/server programming interface which allows programming objects to make indirect network connections to other network services, that is mostly transparent. These proxy connections can be to a DLL (called in-process), another process (called Local), or across a network (called Remote which is the role of DCOM).

In COM/DCOM, the proxy are defined using MIDL (Microsoft interface definition language). OPC develops a set of IDL interfaces (e.g., IOPCServer, IOPCGroup, IOPCItem where the leading “I” stands for Interface) that provides the client communication interface(s) to the OPC servers. There are two interface variations: Custom and Automation, which deal with early binding (at compile time, used in C++) or late binding (at run time, used in VB). This illustration below comes from the OPC DA specification:



OPC interfaces are defined in IDL, and a MIDL compiler processes the IDL files to generate a type library and output C++ headers and proxy files. With IDL, an object (which is an instance of the interface) can have one or many interfaces. Below is a snapshot of the OPC IOPCItem interface. Of note, the uuid is the universally unique identifier (UUID) attribute that is assigned to the interface and that distinguishes it from all other interfaces. The in, out and size IDL attributes are used in “Marshaling”, which is the process of packaging and unpackaging parameters so a remote procedure call can take place. In most COM interfaces, the proxies and stubs for standard marshaling (e.g., types such as integer, double, or VARIANT) are generated by the MIDL compiler.



The beauty of COM/DCOM is that you let Microsoft do all the heavy communication lifting, and users can define their own interfaces, which can then be integrated into COM. However, total world domination did not result from this programming standard. From Wikipedia: *DCOM was a major competitor to*[*CORBA*](http://en.wikipedia.org/wiki/CORBA)*. Proponents of both of these technologies saw them as one day becoming the model for code and service-reuse over the* [*Internet*](http://en.wikipedia.org/wiki/Internet)*. However, the difficulties involved in getting either of these technologies to work over Internet*[*firewalls*](http://en.wikipedia.org/wiki/Firewall_(networking))*, and on unknown and insecure machines, meant that normal* [*HTTP*](http://en.wikipedia.org/wiki/HTTP)*requests in combination with*[*web browsers*](http://en.wikipedia.org/wiki/Web_browser)*won out over both of them.* Another quote: *COM/DCOM security is by far the most misunderstood and therefore problematic issue for DCOM-enabled OPC component vendors*.

In general, DCOM/RPC uses TCP/IP (although you can use other communication schemes) to do socket communication. The DCOM port is TCP 135, which can be firewalled, thus blocking communication entirely. Of note, DCOM does not work over Network Address Translation-Based Firewall, please see <http://support.microsoft.com/kb/248809>.

The major headache with DCOM is the E\_ACCESSDENIED or “permission denied” error that occurs when you attempt to create an remote DCOM server, which indicates a DCOM configuration problem of some sort. To connect and communicate with an OPC Server, the DCOM configuration and authorization settings for authentication, impersonation, access, and launch permissions must be correct[[1]](#footnote-1).

* Authorization is the process whereby a client's access rights to a specific DCOM server are verified, is based on Access Control Lists (ACLs) associated with shared resources, e.g., EVERYONE, ANONYMOUS, etc.
* Authentication is the process by which the security system verifies that a client is actually who he claims to be. The DCOM authentication level is specified by both the client and server machines: the server specifies the minimum required authentication level for incoming calls (any call that comes in below this is automatically rejected via E\_ACCESSDENIED(the error code from Hell), and the client specifies it’s required authentication level for each interface call.
* Launch permissions control authorization to start a COM server during COM activation if the server is not already running.
* Access permissions control authorization of who is allowed to call a running COM server.
* Impersonation is for allowing the server to make connects back on the client machine, which is necessary for event callbacks – this is not even an option. No callbacks, ever.

The diagram below illustrates the sequencing of DCOM authorization.



TestDCOM

TestDCOM allows for testing of DCOM connection between machines.

### Configuring for easier Testing

Although you can change these things in the GUI, you can make your life easier by configuring the config.ini file.

[CONFIG]

ipaddr = 192.168.1.100

user = auduser

pw = SUNRISE

clsid = {75d00afe-dda5-11d1-b944-9e614d000000}

[USERS]

0=None

1=.,auduser,SUNRISE

2=.,auduser,SUNRISE840D

Enter the ip address of the remote machine.

The classid is current set to the Siemens 840D OPC Server COM class id.

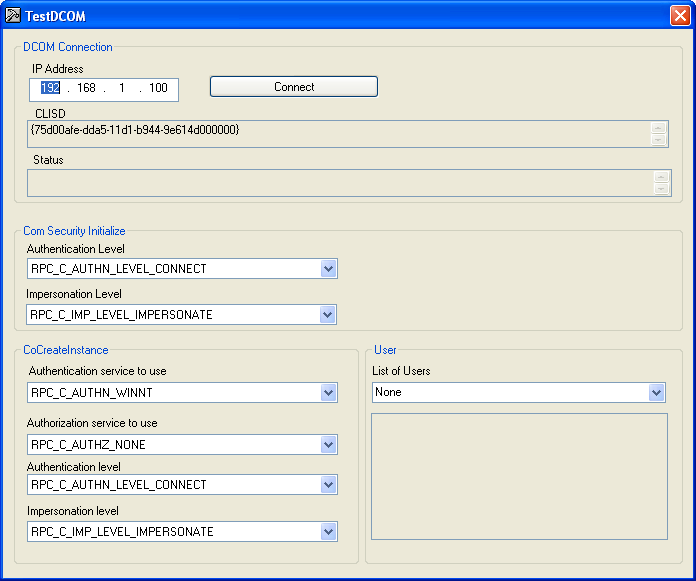
Under users you can add other user/password combinations, by

#=domain,accountname,password

Use domain “.” For workgroup domains (most likely on a CNC.)

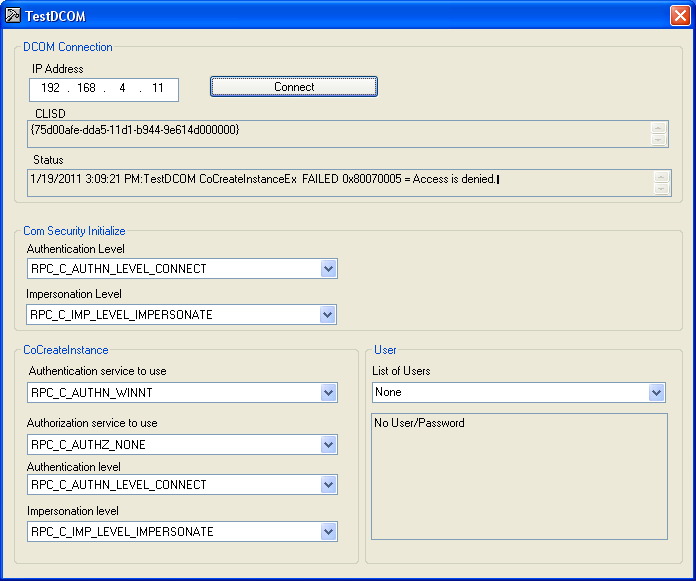
### Running TestDCOM

To run, double click the program, and you will see the following screen (with IP address different).



Hit the connect button and the Status field should change.

* If connected, a timestamp and the status message “Test DCOM Connection succeeded” will be displayed.
* If it fails you will see a time stamp and an error message like shown below, with CoCreateInstanceEx FAILED with error # and description – which as not very helpful.



**☹ Note:**

**DON’T CHANGE THE Authentication service to use from**

**RPC-C\_AUTHN\_WINNT**

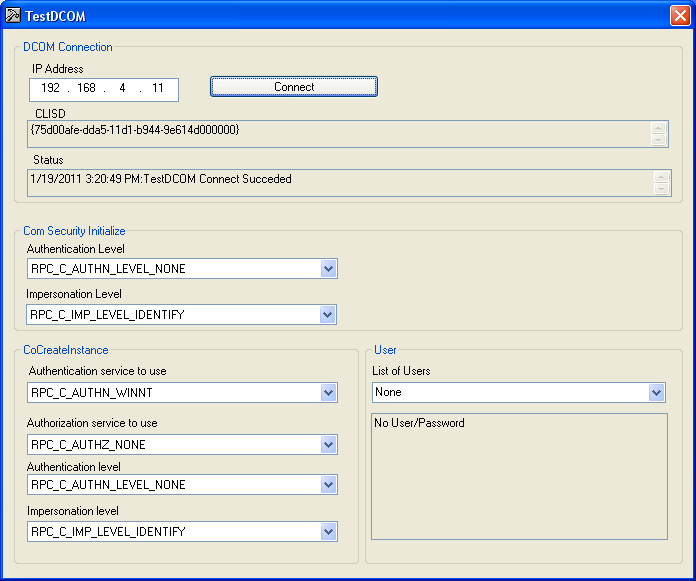
**!!!!!!!!!**

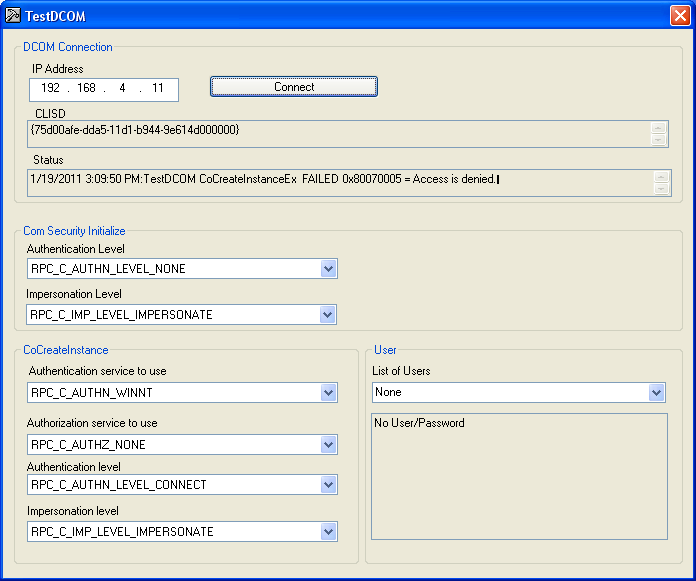
If you fail you can try various combinations of settings and user/password combination.

Change

Authorization service to use: to RPC\_AUTHZ\_NONE

Figure TRY THIS - IT IS THE MOST LENIENT PERMISSIONS!!





## Ini File Options

### Fake Spindle

In MtcOpcAgent.ini in the executable folder there is a flag to indicate faking a spindle:

: [OPCSERVER]

...

FakeSpindle= 0

If FakeSpindle is 1 then when the mode is AUTO and execution is EXECUTING Srpm will be 99. This is called a fake spindle for cases when you don’t have a spindle, or can’t identify the OPC tag name.

### MultiSpindle

In MtcOpcAgent.ini in the executable folder there is a flag to indicate two spindles:

: [OPCSERVER]

...

MultiSpindle= 0

MultiSpindle is programmed to handle only 1 or 2 spindles that enables the code to detect the larges value of the possible spindles (max 2 – hard coded).

If MultiSpindle is 1 then the tags Tag.OPC.Srpm{1,2} must be filled out: (Values read but no mtconnect mapping).

Tag.Sample.Srpm= /Channel/Spindle/actSpeed[1]

Tag.OPC.Srpm1= /Channel/Spindle/actSpeed[1]

Tag.OPC.Srpm2= /Channel/Spindle/actSpeed[2]

### New Devices XML File

In MtcOpcAgent.ini in the executable folder there is a flag to indicate whether a new Devices.xml file is to be written:

[OPCSERVER]

...

WriteNewDevicesFile=1

Where setting the flag equal to 1 indicates that a new file is to be written. You have to manually turn on and turn off this flag. There is no error checking.

### Reset at Midnight

In MtcOpcAgent.ini in the executable folder there is a flag to indicate whether to reset the agent (restart) at midnight every night. This is useful since there can be DCOM/COM leaks and socket errors can accumulate. It takes 5-110 seconds at midnight to restart the Agent. The flag is set in

: [OPCSERVER]

...

ResetAtMidnight=1

1. <http://www.opcfoundation.org/Archive/2a83366b-da1c-490a-99d2-f321e2d4ce51/White%20Papers/OPC,%20DCOM%20and%20Security.pdf> [↑](#footnote-ref-1)