VC++ OPC SHDR Implementation

# November 16, 2011 Contents

[Revision History 4](#_Toc455478136)

[Copyright 4](#_Toc455478137)

[Disclaimer 4](#_Toc455478138)

[Terminology 4](#_Toc455478139)

[Background 5](#_Toc455478140)

[SHDR Background 6](#_Toc455478141)

[OPC Background 7](#_Toc455478142)

[1. read Config file 7](#_Toc455478143)

[2. OPC Server Connect 9](#_Toc455478144)

[3. OPC Group and Item Subscription 10](#_Toc455478145)

[Basic Installation 11](#_Toc455478146)

[Testing opc-shdr 12](#_Toc455478147)

[Uninstall 13](#_Toc455478148)

[Installation Troubleshooting 13](#_Toc455478149)

[OPC DCOM Deployment Checklist 13](#_Toc455478150)

[Firewall Requirements 17](#_Toc455478151)

[To allow a program to communicate through Windows Firewall 24](#_Toc455478152)

[Creating Custom Action DLL Project 26](#_Toc455478153)

[Insert Custom Action DLL into MSI Setup Project 27](#_Toc455478154)

[Reinstalling Issues 30](#_Toc455478155)

[Configuring DCOM for MTConnect 32](#_Toc455478156)

[Run dcomcnfg 33](#_Toc455478157)

[Open DCOM config inside Component Services 35](#_Toc455478158)

[Right Click My Computer and Select Properties 37](#_Toc455478159)

[Set Siemens 840D OPC COM server Properties 38](#_Toc455478160)

[Set Siemens OPC Server Authentication Level to None 39](#_Toc455478161)

[Location 40](#_Toc455478162)

[Security 41](#_Toc455478163)

[Launch and Activation Permissions 42](#_Toc455478164)

[Access Permissions 48](#_Toc455478165)

# Revision History

**Date Description Author Version**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author** | **Version** | **Description** |
| 11/16/11 | NIST | 1.0 | Initial SHDR and OPC implementation |

# Copyright

*Software produced by the National Institute of Standards and Technology (NIST), an agency of the U.S. government and by statute is not subject to copyright in the United States. Recipients of this software assume all responsibility associated with its operation, modification, maintenance, and subsequent redistribution.*

# Disclaimer

*No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. Certain commercial equipment, instruments, or materials are identified in this report in order to facilitate understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.*

# Terminology

**Adapter** An optional software component that connects the Agent to the Device.

**Agent** A process that implements the MTConnect specification, acting as an interface to the device.

**Device** A piece of equipment capable of performing an operation. A device is composed of a set of components that provide data to the application. The device is a separate entity with at least one Controller managing its operation.

**Sample** A sample is a data point for continuous data items, that is, the value of a data item at a point in time.

# Background

This document describes a Visual C++ Windows “OPC-SHDR” implementation of a SHDR adapter that reads OPC data from one or more remote OPC Server(s) and streams SHDR data to MTConnect Agent(s). The OPC SHDR adapter installs as a Windows service, and collects OPC data and streams the data to listening Agents. OPC is an factory floor communication standard and readers are referred to [www.opcfoundation.org](http://www.opcfoundation.org) for more details on OPC. Readers are advised to visit the MTConnect Institute website ([www.mtconnect.org](http://www.mtconnect.org/)) to access the latest specifications and more information about MTConnect.

The OPC-SHDR program is configurable to change name and locations of OPC Server as well as properties of the MT Connect Adapter. The OPC-SHDR installation requirements were:

1. Windows 32bit for XP, Vista, 7. Win64 has not been tested.
2. The OPC common components and Proxy/Stub DLLs need to be installed.
3. All CNC Data Access exe or DLL files are located in the same directory as the exe file being used.

The OPC-SHDR implementation is a Visual C++ Windows implementation that runs as a windows service. Figure 1 shows the targeted architecture for the deployment, where SHDR-OPC Adapter is running on the separate PC from the CNC and communicated using DCOM to the OPC Server. . The OPC implemented in this document is known as OPC Data Access 2.0, or OPCDA 2.0. The SHDR-OPC Adapter could run on the CNC, but is tailored for remote access. The system architecture then communicates SHDR commands across the Ethernet to the MTConnect Agent. Again, the Agent could be located on the same PC as the Adapter. The architecture below shows two a private factory Ethernet and a more public Company intranet. This is shown as a basic security architecture, however, a firewall, or complete internet accessible architecture could be deployed.



Figure 1 OPC-SHDR DEPLOYMENT SYSTEM ARCHITECTURE

## SHDR Background

The SHDR protocol is not part of the MTConnect® specification, and readers are advised to read “MTConnect® Standard SHDR Protocol Companion Specification” for more details. Briefly, the SHDR protocol “was designed to be as simple as possible”. SHDR provides a TCP socket stream such that one or more Agents can connect to the SHDR socket and listen to the data stream, with behavior similarl to telnet. The communication format is a pipe ‘|’ delimited data stream that begins with a timestamp and then follows with a timestamp in ISO 8601 date time format with optional decimal places. The time is required to be in UTC and therefore must have a trailing Z to indicate that no time zone (+0) is being used. SHDR handles Samples, Events, Conditions, Assets, and Alarms (deprecated). Below is an EBNF representation of the SHDR including the new multiline asset implementation:

<SHDR> ::= UTCDATE "|" <StatementList>

<StatementList> ::= <Statement> | <Statement> EOL <StatementList>

<Statement> ::= <SimpleStatement> | < MultilineStatement >

<SimpleStatement> ::= <Tag> "|" <Value> { "|" <Value>}\*

<MultilineStatement> ::= "@" <Tag> "@" "|" ID "|" --multiline—{A-Z}+ .\* --multiline—{A-Z}+  
<Tag> ::= <Sample> | <Event> | <Condition>

<Sample> ::= ID

<Event> ::= ID

<Condition> ::= ID “|” <Level> “|” <NativeCode> “|” <NativeSeverity> “|” <Qualifier> “|” <Description>

<Level> ::= UNVAILABLE | NORMAL | WARNING | FAULT

<NativeCode> ::= NUMBER | TEXT

<NativeSeverity> ::= TEXT

<Qualifier> :== HIGH, LOW, …

<Description> ::= TEXT

Here is a sample SHDR string for transmitting s data, and the line must be terminated with a line feed (\n) which is ASCII character 10 or 0xA.

2010-09-29T23:59:33.460470Z|Xact|1.4198908806\n

# OPC Background

This section gives a more in depth discussion on the features and configuration of the OPC Client adapter.

The Figure below details the basic operation of the adapter program. The program reads the ***adapter.ini*** file which determines the OPC client parameterization. And then a sequence of operations, based on the config settings which will be described in greater detail.



Figure 2 OPC Client Adapter Process

## 

## read Config file

This table summarizes the configuration parameters:

|  |  |
| --- | --- |
| [ADAPTER] | Contains parameters to configure the SHDR aspect of MTConnect adapter. |
| ServiceName=MTConnectOpcAdapter | Gives the OPC-SHDR a name that will be entered into Service Control Manager (SCM) |
| ScanDelay=1000 | Configures how long the SHDR thread waits between cyclic updates. |
| Port = 7878 | Configures the TCP socket port for listeners to connect to. |
| [OPCSERVER] | Contains a list of OPC specific configuration parameters for communicating with one or more OPC servers. |
| OpcTags = MATRIKONTAGS | Describes the Ini section for each OPC device where the list of SHDR and OPC tags mapping is located. Each name separated by a comma, i.e., “,”. |
| MTConnectDevice=CNC1,CNC2 | Device name(s) as found in the Devices.xml configuration file. Please refer to the MTConnect documentation for information on how devices are named. Each device name separated by a comma, i.e., “,”. |
| ServerMachineName=192.168.1.102 | PC Name(s) or IP address where OPC Servers are located. Each ip address separated by comma, i.e., “,”. |
| ResetAtMidnight=true | Parameter that causes the Service to be reset at midnight. When true, reset. This is to clear an accumulated problems. |
| Language=English | Alarm management – not used. |
| #User=.,auduser,SUNRISE | If user defined, does user/password authentication to log onto remote OPC server machine. Specified as domain “,” user “,” password. Note, clear text passwords are required. In general for CNCs, these must be matching workgroup user/password (not domain) for CNC and FEPC. |
| ServerRate = 1000 | Cyclic Synchronous OPC read of data, in milliseconds. |
| QueryServerPeriod=5000 | Time allotted between remote OPC connection attempts. |
| ProcessPriority=64 | Process priority of OPC thread. |
| # DCOM security | There are parameters to enable using a particular user name and password for DCOM authentication. Further, if defined, parameter can be used to define CComSecurity |
| AuthLevelComSecurity | The amount of authentication provided to help protect the integrity of the data (none, connect, packet, etc.) for COM default security setup. RPC\_C\_AUTHN\_LEVEL\_NONE |
| ImpLevelComSecurity | Specifies an impersonation level, which indicates the amount of authority given to the server when it is impersonating the client RPC\_C\_IMP\_LEVEL\_IDENTIFY, RPC\_C\_IMP\_LEVEL\_ANONYMOUS |
| AuthzSvc | Defines what the server authorizes. RPC\_C\_AUTHZ\_NONE or RPC\_C\_AUTHZ\_NAME |
| AuthnLevel | Indicates the amount of authentication provided to help protect the integrity of the data, RPC\_C\_AUTHN\_LEVEL\_DEFAULT, RPC\_C\_AUTHN\_LEVEL\_NONE , RPC\_C\_AUTHN\_LEVEL\_CONNECT |
| [SERVERTAGS] | Section defines set of OPC server specific information and tag mapping. The section name is used by the OPCSERVER section as a information key. |
| SERVERNAME | Name to identify server. Can be prog id, but is not used. |
| CLSID | Hard coded universal unique id, CLSID which identifies the OPC Server, e.g., Siemens 840D CNC id is 75d00afe-dda5-11d1-b944-9e614d000000 |
| CNCProcessName | Name of program that OPC server is monitoring, not used. Was used to determine if CNC running before attempting connection. Very hard to detect runningn program remotely, e.g., Siemens CNCprogram is maschine.exe |
| Tag… | Series of tags defining the OPC variables to read and then map into MTConnect device variables. |

## OPC Server Connect

The OPC-SHDR program will attempt to establish a communication link with the OPC server. This step will repeat indefinitely and waits before connection attempt as specified by CONFIG file parameters QueryServerPeriod.

The QueryServerPeriod parameter determines how long to wait (in milliseconds) before attempting the next connection.

QueryServerPeriod = 10000

The CLSID tag describes Microsoft’s Component Object Model (COM) GUID (Globally Unique Identifier) of the OPC server. It is hard coded as a Class ID (CLSID) and not a Program ID, because if you are trying to activate a remote OPC server and the Program ID is not registered in the local registry, the remote COM creation may fail. THIS VALUE WILL NOT BE CHANGED AND WORKS FOR ALL Siemens 840D OPC Servers.

PROGID = OPC.SINUMERIK.Machineswitch

OpcServerClsid=75d00afe-dda5-11d1-b944-9e614d000000

## OPC Group and Item Subscription

OPCDA control device objects and interfaces are called OPC Servers. Applications, called OPC Clients, can connect to OPC Servers provided by one or more vendors. Once the OPC Server connection has been established, an OPC group is created and tags are added for data monitoring.



Figure 3 OPC System Architecture

Figure 3 shows the functionality of the OPC DA specification within a CNC Architecture, which includes the following concepts:

* **OPC Server** is a COM object to which the OPC client first connects. The OPC Server handles connectivity to automation hardware. Its responsibility is to manage OPC groups, translate errors, provide server status, and allow browsing of OPC items.
* **OPC Group** is a COM object for logically organizing data items. OPC clients can pick and choose among the known OPC items on the OPC server in order to create groups. OPC Groups are managed by the OPC client who can activate or deactivate the group, change the group name or update rate among and subscribe for data change event notification. Reading and writing of OPC data is done through the OPC Group.
* **OPC Item** is a single tag (or automation device data point) managed by the OPC server. OPC does not define any application item tag names, e.g., AxisLocation. Instead, OPC clients rely on the vendor to allow clients to browse for OPC data items, or provide a list of OPC data items.

🟏🟏🟏 Note, the SHDR OPC client only uses one OPC Group with a set of items.🟏🟏🟏

Figure 3 OPC System ArchitectureFigure 3 shows OPC data management for a simple CNC OPC Server example. The OPC Client creates 2 OPC Groups, Group 1 containing the MachineMode, CycleOn, Ready and Program OPC Items, while Group 2 contains the BlockLine and BlockNumber OPC Items. Group 1 and 2 could run at different update rates if the data timeliness is an issue. To improve performance, if the CNC is in Manual mode, the Group 2 items BlockLine and BlockNumber could be deactivated. Note, only one OPC group and a set of tags are possible with the current setup.

Under an OPC Server section in the INI file, is a mapping of MT Connect tags into an equivalent OPC Server tag. Because MT Connect has two forms of tags – Sample and Event, the general form of the tag definition is

Tag.{Enum|Const}.[Sample|Event|Condition]=OPCTag

{X|Y} means optionally X or Y and where [X |Y] means one of X or Y. For example, Tag.Sample.Srpm specifies that the Spindle tag is a Sample, whose MT Connect name is “Srpm” and whose OPC tag is /Channel/Spindle/actSpeed[1] as defined below.

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

There is no enum involved in the Srpm. In the case of machine mode, the tag uses an Enum to specify this and Enum tags are associated with various mode values.

Tag.Enum.Event.controllermode=/Bag/State/opMode

Shown below enumeration strings are assigned to tag mode in the INI file using the mode MT Connect terminology mapping OPC server values (0,1,2).

Enum.controllermode.0 = MANUAL

Enum.controllermode.1 = MANUAL\_DATA\_ENTRY

Enum.controllermode.2 = AUTOMATIC

# Basic Installation

🟏🟏🟏 PRELIMINARY NOTES 🟏🟏🟏

1) By default, MTConnect adapter uses port 7878 for socket communication to an Agent. If a hardware or software firewall is blocking this port, no SDHR data will be available. Likewise, if the DCOM port is blocked by the firewall, the OPC communication will not work.

2) Ping the IP address to make sure the CNC is online for OPC communication.

3) Test the MTConnect data stream with telnet or http

There are no install wizards, instead some manual and batch script configuration is necessary in order to get things to run properly.

First, configure the adapter.ini file. If you are connecting to a one Siemens 840D CNC OPC Server, only the IP Address of the OPC server will need to be modified.

Second, you will need make sure that the tags described in the Agent devices.xml configuration file matches the tags that you are using in the adapter.ini.

Third, you will have to make sure that the OPC/DCOM security permissions (firewall, dcomcnfg, etc.) are properly configured.

To install the MTConnect Agent service, run :

adapter.exe install

which installs the service into the Windows Service Control Manager (SCM). No other registry modifications are necessary.

### Testing opc-shdr

The MTConnect adapter uses port 7878 (configurable) for communication to the outside world. If a hardware or software firewall is blocking this port, no SHDR data will be available. You should ping the CNC IP address to make sure the CNC is online.

You can test the MTConnect SHDR data stream with the following

http://adapter-ipaddress:7878

where you use the actual ip address for adapter-ipaddress or use telnet:

Start->Run telnet

------------------------

Welcome to Microsoft Telnet Client

Escape Character is 'CTRL+]'

Microsoft Telnet> open ipaddress 7878

Telnet Client is not installed by default on Windows 7/Vista. If it is not installed, the following will correct this:

Start

Control Panel

Programs And Features

Turn Windows features on or off

Check Telnet Client

Hit OK

After that you can start Telnet via Command Prompt.

### Uninstall

To uninstall the MTConnect OPC-SHDR service, run

adapter.exe uninstall

Note, you may get a error if the service is running. You should stop the service via the Service Control Manager first.

After removing the service from the SCM, you can delete the installation folder.

# Installation Troubleshooting

Possible reasons the Adapter fails to run:

1. Files were mangled. Visually inspect xml and ini files to make sure CR/LF weren’t converted to LF only. Solution: Redownload.
2. Works locally, but cannot see Adapter SHDR outside of local PC. Check firewall port 7878 blocking.
3. No data – check OPC connection. This is also known as the DCOM nightmare.
4. If you are using Windows XP Service Pack 2 with Firewall enabled, check to make sure the OPC-SHDR service is allowed through the firewall of blocked socket numbers. This will not prevent OPC-SHDR **service** from appearing to work, but http Port 7878 must be available to communicate to the MT Connect Agent.

To see if http port 7878 is available on the FEPC, do the following (on WinXP, similar on Win7):

**Run->Control Panel -> Firewall ->Exceptions.**

You should see the list of programs and services that can use sockets.

## OPC DCOM Deployment Checklist

This section discusses how to connect an OPC Client and and OPC Server by configuring DCOM and other Windows security measures. The discussion the Siemens 840D OPC Server will be used as an example, with a Program ID of OPC.SINUMERIK.MachineSwitch. Below, FEPC stands for FEPC or the PC on which the Adapter runs. Please note, Adapter running on the CNC should not have as much security issues.

There are couple of rules of thumbs which can help, not eliminate DCOM headaches:

1. Reboot every time after configuring DCOM,
2. Make sure you have administrator rights, there are now flavors of administrator rights, that can make configuring windows difficult,



The following table lists a set of operations with a MINIMAL AMOUNT OF SECURITY in order to allow a connection between the OPC-SHDR adapter and the OPC server. The table below shows how to connect to the Siemens 840D OPC Server, with a Program ID of OPC.SINUMERIK.MachineSwitch. Below, FEPC stands for FEPC or the PC on which the Adapter runs. Please note, Adapter running on the CNC should not have any security issues.

|  |  |  |
| --- | --- | --- |
| FEPC | Ping server | 🗸 |
| CNC | No firewall blocking | 🗸 Check for Windows firewall blocking and especially 3rd party firewalls in task manager administered by IT |
| CNC | No NAT box | 🗸 When you try to make Distributed Component Object Model (DCOM) calls over a Network Address Translation (NAT)-based Firewall, you receive the error 0x800706BA (RPC\_S\_SERVER\_UNAVAILABLE) See http://support.microsoft.com/kb/248809 |
| CNC | DCOM enabled | 🗸  Run->dcomcnfg  ConsoleRoot->ComponentService->Computers->My Computer (Right click select Properties)  [Default Properties] -> Enabled Distributed COM on this machine CHECK  [Default Properties] -> Default Authentication Level -> NONE  [Default Properties] -> Default Impersonation Level –> Identity |
| CNC | DCOM Limits | Run->dcomcnfg  ConsoleRoot->ComponentService->Computers->My Computer (Right click select Properties)  [COM Security] Access Permissions : Edit Default  : System, Everyone  [COM Security] Launch and Activation Permissions : Edit Default  : Admin, System, Everyone, Interactive |
| CNC | OPC Server Configure | Run->dcomcnfg  ConsoleRoot->ComponentService->Computers->My Computer -> DCOM Config (Double Click)  Select OPC.SINUMERIK.MachineSwitch – Right Click Properties  [GENERAL] Authentication Level -> NONE  [LOCATION] Run application on this computer – CHECKED  [SECURITY] Launch and Activation Permissions : Customize  : Admin, System, Everyone, Interactive  [SECURITY] Access Permissions: Customize  : System, Everyone, Interactive  [ENPOINTS]  [IDENTITY] The interactive user ?????????????????  REBOOT |
| FEPC | Authentication Level=None,  If you have non-domain users, check if you have included "Everyone” in the  1. access and  2. launch permissions | Run->dcomcnfg  ConsoleRoot->ComponentService->Computers->My Computer (Right click select Properties)  [Default Properties] -> Enabled Distributed COM on this machine CHECK  [Default Properties] -> Default Authentication Level -> **NONE must match NONE on CNC**  [COM Security] Access Permissions : Edit Default  : System, Everyone  [COM Security] Launch and Activation Permissions : Edit Default  : Admin, System, Everyone, Interactive |
| CNC | Interactive User | Admin group? Or User? What is the difference?  Interactive Includes all users who log on to a Windows NT or  Windows 2000 system locally (at the console). It  does not include users who connect to Windows NT  or Windows 2000 resources across a network or are  started as a server.  Network Includes all users who connect to Windows NT or  Windows 2000 resources across a network. It does  not include those who connect through an  interactive logon.  Creator/Owner The Creator/Owner group is created for each  sharable resource in the Windows NT system. Its  membership is the set of users who either create  resource s(such as a file) and those who take  ownership of them.  Everyone All users accessing the system, whether locally,  remotely, or across the network.  System The local operating system. |
| FEPC | Service Test | Domain user?  No user?  Local user? .\auduser SUNRISE – don’t forget .\ |
| CNC |  | Use Event Viewer to find out additional auditing information on why the DCOM connection fail – MUST BE ENABLE using Local Policy  Start -> Administrative Tools->Local Security Policy.  Local Policies-> Audit Policy  1. Enable auditing for success and failure for the following options: Audit logon events,  2. Audit object access,  3. Audit privilege use. |
|  |  |  |
| FEPC | Policy Management | GROUP POLICY  Start->Run-> gpedit.msc  1.        Go to Computer Configuration | Windows Settings | Security Settings | Local Policies | Security Options.  2.        In the right pane, double-click "DCOM: Machine access restrictions ..." . Check "Enable policy", click "Edit security" and give full permissions to the same users as above.  3.        Do the same for "DCOM: Machine launch restrictions ...".  4.        Close the console. |
|  |  | LOCALSECURITYPOLICY????????  Since we logon to a domain (NW)    1.        If you log on to a domain  Repeat the same steps for the "Default Domain Policy" (see Start | Programs | Administrative tools|LocalSecurityPolicy). |

**A brief background on default Launch and Access permissions in DCOM**: Launch permissions define who can launch a COM based application (such as an OPC server) both over the network or locally. Access permissions define who can access that application once it has been launched. Applications can get their Launch and Access permissions from one of three places: they can use explicitly defined setting for their application, they can use the default permissions or they can set their own permissions programmatically. Because an application could set its own permissions programmatically, the explicitly defined or default settings, although set properly, may not be used and therefore the user is not able to explicitly have control over these settings. To overcome this security flaw, Microsoft has added .limits. to the DCOM security settings from Launch and Access to limit the permissions that an application can use. This limit prevents the application from using permissions beyond what is specified in the DCOM configuration settings. By default the limits set by Service Pack 2 will not allow for OPC communications over the network.

**Edit the Limits for Access and Launch**

a. Access Permissions – Edit Limits... You need to check the Remote Access box for the user labeled ANONYMOUS LOGIN in this dialog. Note: This setting is necessary for OPCEnum.exe to function and for some OPC Servers and Clients that set their DCOM 'Authentication Level' to 'None' in order to allow anonymous connections. If you do not use OPCEnum you may not need to enable remote access for anonymous users.

b. Launch and Activation Permissions – Edit Limits...

You need to check the remote boxes for the user labeled Everyone in this dialog.

If remote OPC Clients will access the server, ensure that ‘SYSTEM’ is listed in the ‘Group or user names’ list box with the ‘Allow’ check box checked for ‘Local Access’ and ‘Remote Access’. If not, click the [Add] button, then add ‘SYSTEM’.

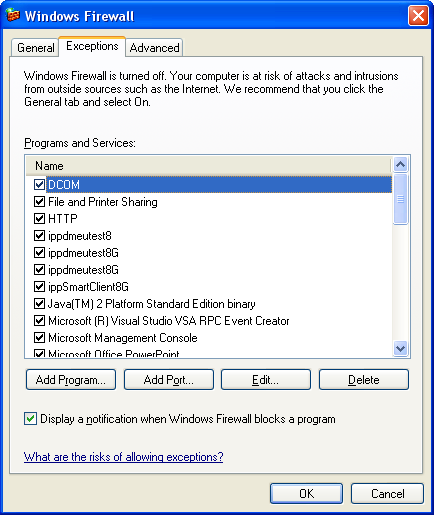
## Firewall Requirements

1. Local Users Authenticate as Themselves.

Simple File Sharing forces every remote user to Authenticate as the Guest User Account, which causes problems. By default, the Simple File Sharing user interface is turned on in Windows XP Professional-based computers that are joined to a workgroup. Windows XP Professional-based computers that are joined to a domain use only the classic file sharing and security interface.

1. DCOM Firewall Exception & DCOM Port Exception

MTConnect/OPC uses DCOM to communicate. DCOM and port 135 which provides RPC-based services for DCOM, must be added to exceptions to allow communication traffic to be able to go through all firewalls, including Windows as shown below:



Above, if DCOM is ON, this means we should be able to directly access an 840D OPC Server.

Generally, if there is a firewall blocking DCOM socket port access, you will get this error message.

0x800706ba - The RPC server is unavailable

1. USER HAS CREDENTIALS ON REMOTE SYSTEM

DCOM is used to remotely create the OPC Server component for the OPC Client. To do this, DCOM obtains the Client's current username associated with the current. Windows guarantees that this user credential is authentic. DCOM then passes the username to the machine or process where the component is running. DCOM on the component's machine then validates the username again using whatever authentication mechanism is configured (can be NONE) and checks the access control list for the component. If the OPC Client's username is not included in this list (either directly or indirectly as a member of a group of users), DCOM rejects the call before the component is ever involved. The figure “OPC Server DCOM Component Creation Timeline” below describes the sequence events in creating the remote OPC Server component that involves permissions to remote log on, permissions to launch(create) the OPC Server component, permission to access the OPC Server component.

If \*ANY\* of these fails, you get same error message, the dreaded:

0x80070005 - General access denied error

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)

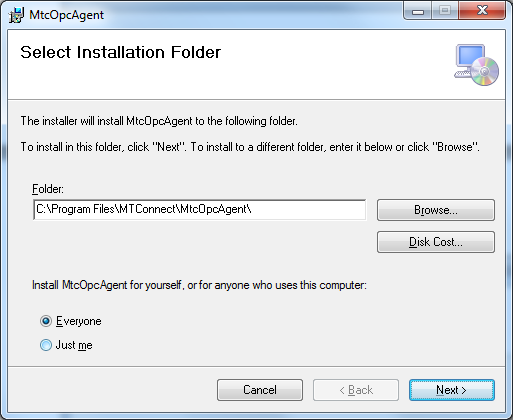
Rename "OPC Core Components Redistributable (x64).msx"

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.

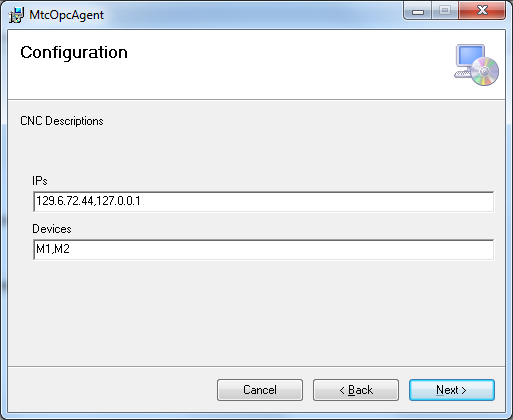
1. Install the Agent (which reads status data from the CNC using OPC and displays using http). Run the installation msi script.
2. Select Installation Folder



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:

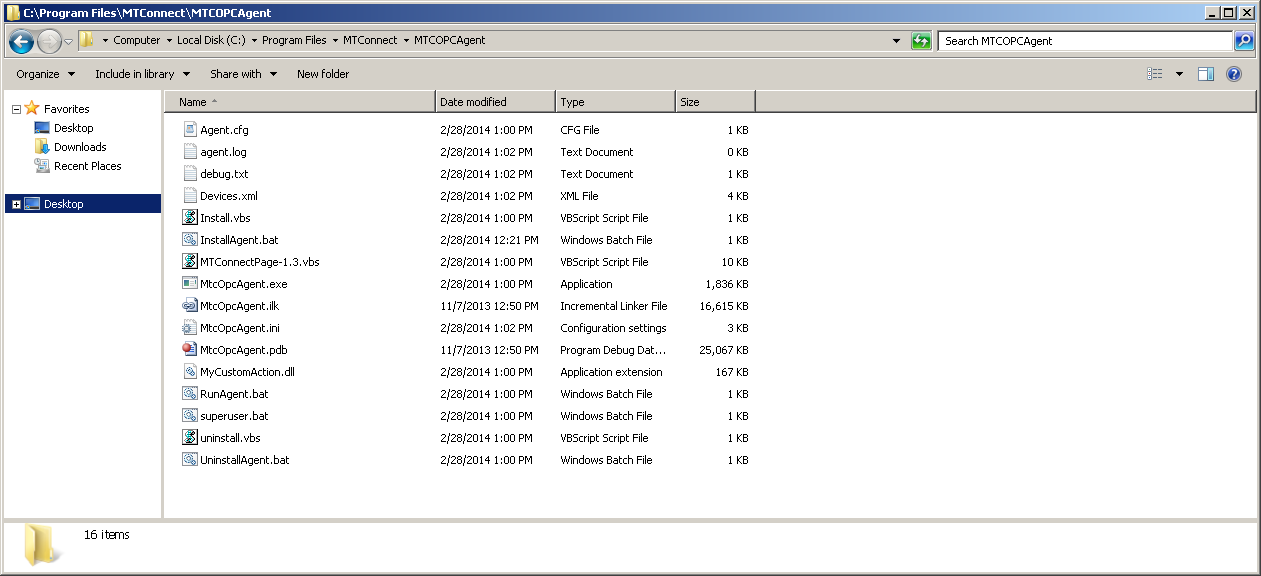
192.168.24.4,127.0.0.1

M1,M2

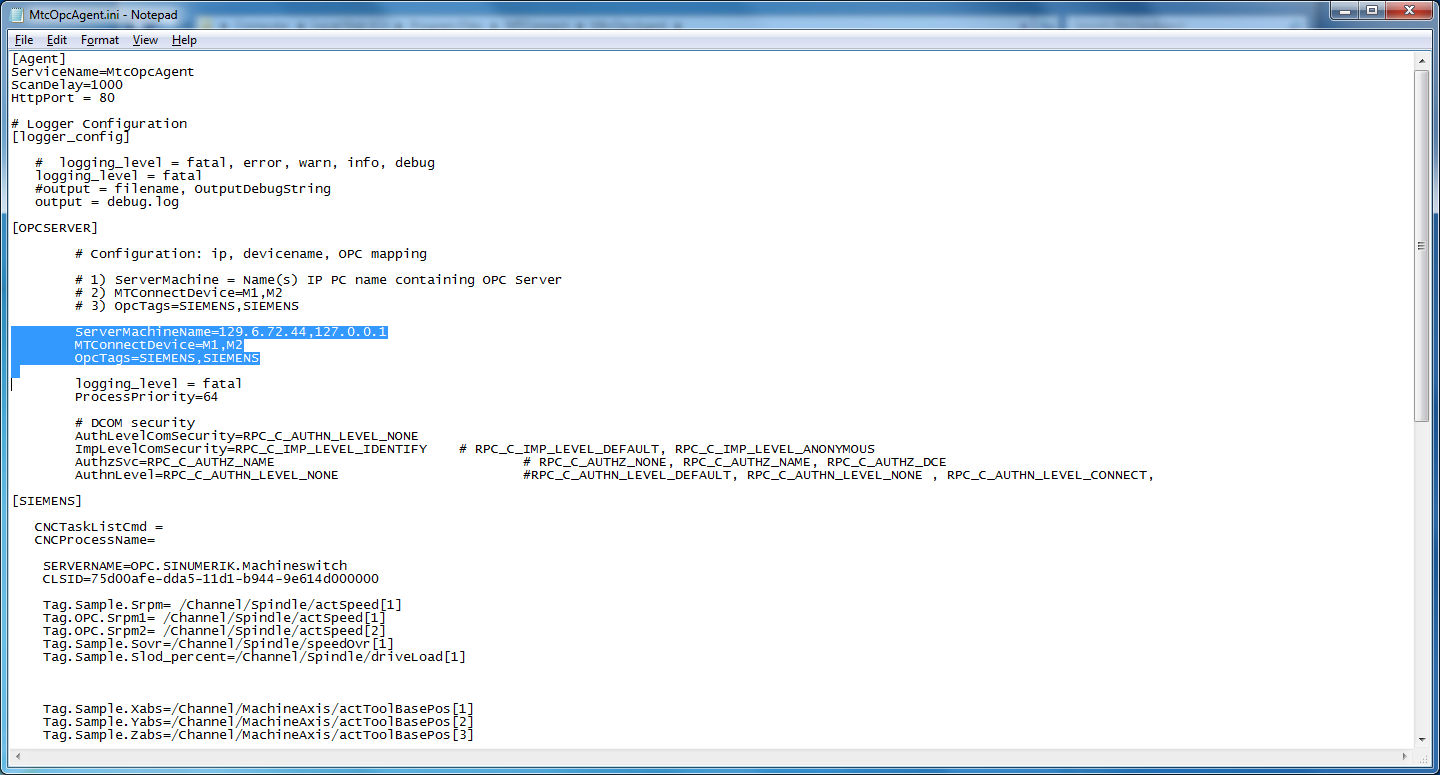


1. Next you will need to verify that the MtcOpcAgent has been installed as a service, and then start it (because you probably didn't have sufficient privileges to install services onto the Windows 7 box.

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

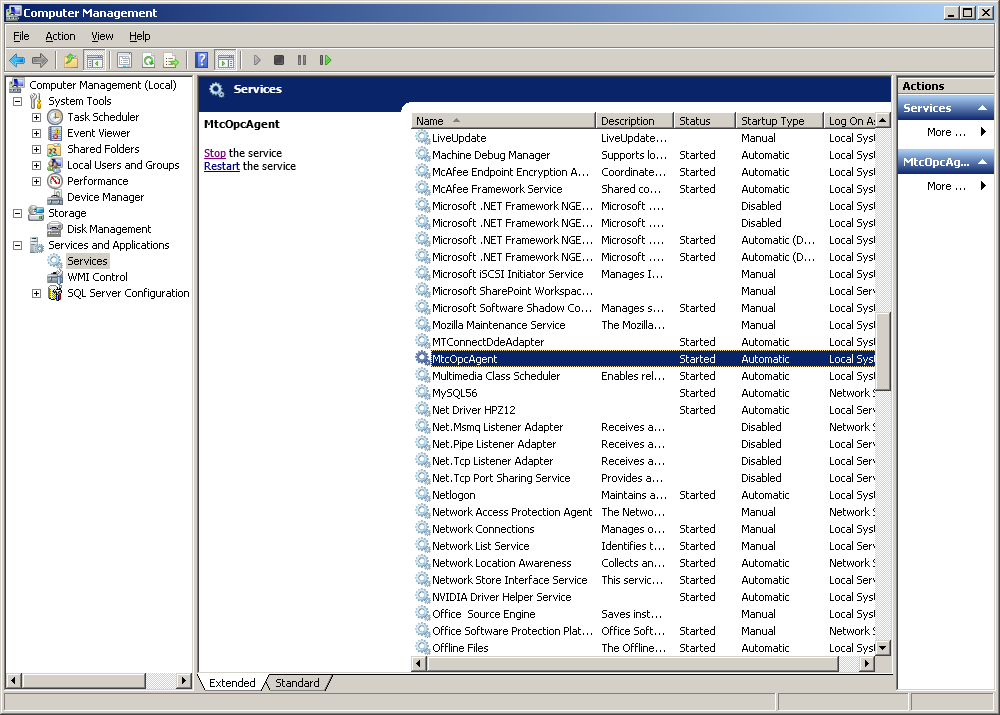


Verify that the configuration parameters are correct, open MtcOpcAgent.ini in notepad, and confirm highlighted text below matches what was entered during installation:



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:

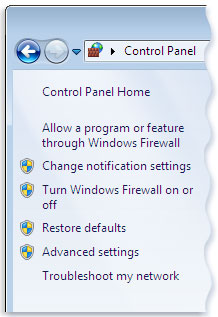
Note: You need to let the MtcOpcAgent.exe be allowed to pass through the firewall. DCOM needs port 135 opened. and MtcOpcAgent.exe opened in the firewall.

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

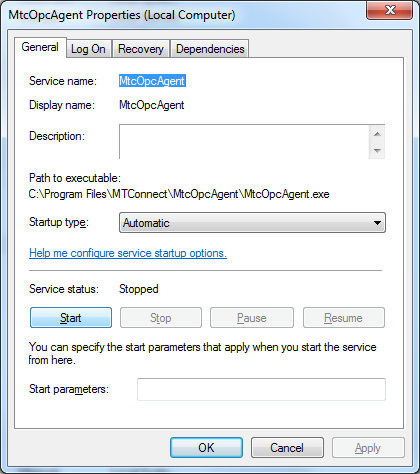
Or follow these directions from Microsoft:

### 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.

An MTConnect CUSTOM ACTION for Windows Installer Scripts (msi)

This document briefly describes the process of creating a Visual Studio Setup and Deployment project (now obsolete). Readers can look at <http://bonemanblog.blogspot.com/2005/11/custom-action-tutorial-part-ii.html>

for a more detailed explanation.

## Creating Custom Action DLL Project

Step 1 is to create a C++ win32 Dll MyCustomAction project to handle Install/Uninstall options. Eventually you will want to ATL/WTL capabilities so we can display windows (static linking to ATL lib) but these are best done manually.

Step 2 is to modify the MyCustomAction.def DLL definitions file, which should be written as:

; MyCustomAction.def

;

; defines the exported functions which will be available to the MSI engine

LIBRARY "MyCustomAction"

EXPORTS

Install

Commit

Rollback

Uninstall

And the basic MyCustomAction.cpp file should contain:

BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul\_reason\_for\_call, LPVOID lpReserved )

{

return TRUE;

}

extern "C" UINT \_\_stdcall Install(MSIHANDLE hInstall){return ERROR\_SUCCESS;}

extern "C" UINT \_\_stdcall Commit(MSIHANDLE hInstall){return ERROR\_SUCCESS;}

extern "C" UINT \_\_stdcall Rollback(MSIHANDLE hInstall){return ERROR\_SUCCESS;}

extern "C" UINT \_\_stdcall Uninstall (MSIHANDLE hInstall){return ERROR\_SUCCESS;}

You will need to add the MSI functionality to the source file to compile and link:

#include <msi.h>  
#include <msiquery.h>

#pragma comment(lib, "msi.lib")

with the following explanation of each step.

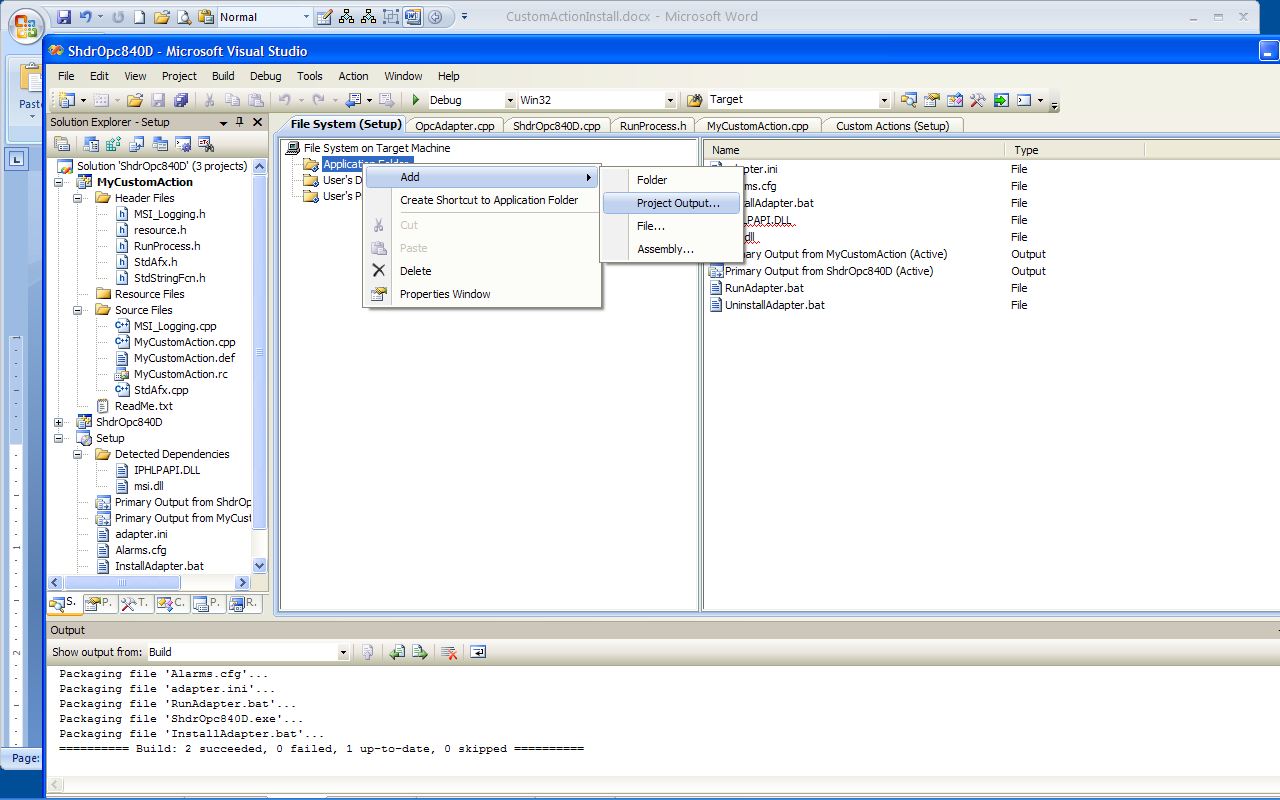
**Install**

**Commit** **-** Custom actions are executed upon successful completion of the installation script. If the [InstallFinalize action](http://msdn.microsoft.com/en-us/library/windows/desktop/aa369505(v=vs.85).aspx) is successful, the installer will then run any existing Commit Custom actions. The only mode parameter the installer sets in this case is MSIRUNMODE\_COMMIT.

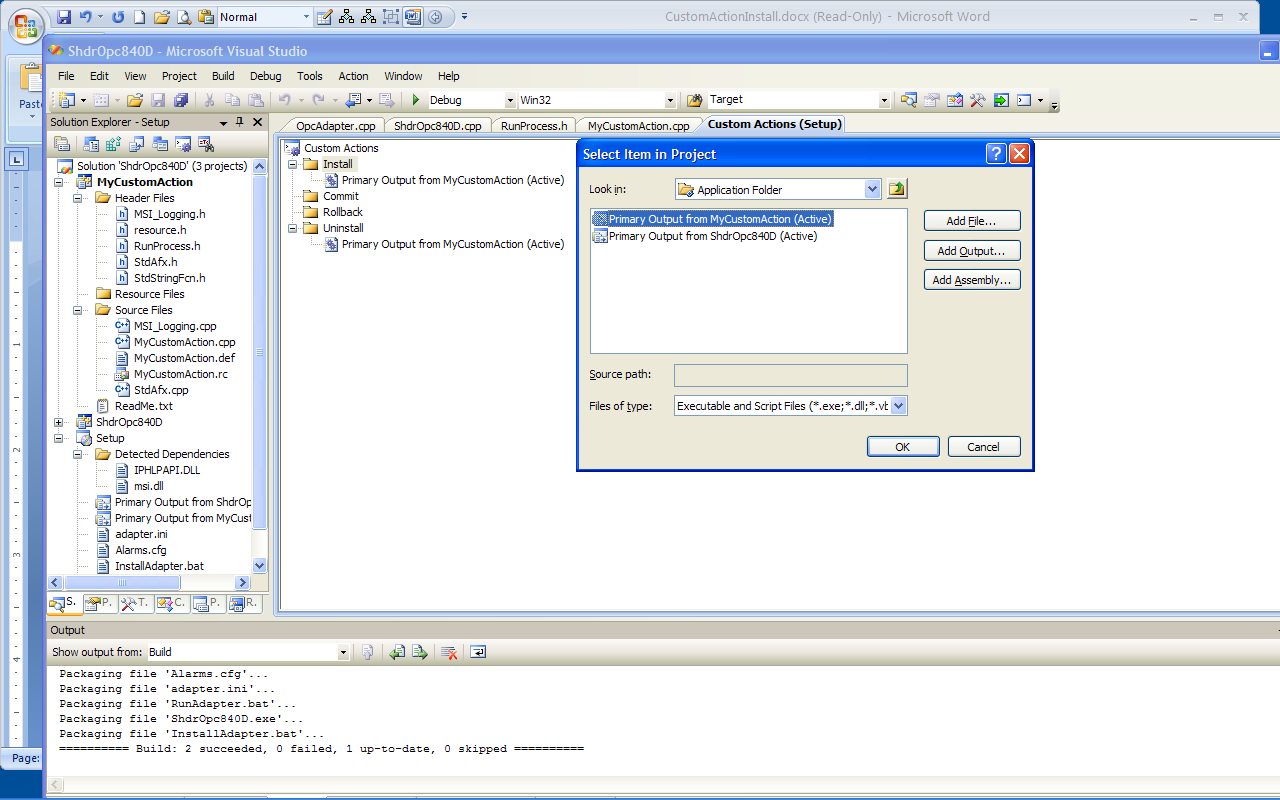
**Rollback** - If an installation is unsuccessful, the installer attempts to rollback the changes made during the installation and restore the original state of the computer. A rollback custom action is a type of [deferred execution custom action](http://msdn.microsoft.com/en-us/library/windows/desktop/aa368268(v=vs.85).aspx), because its execution is deferred when it is invoked during the installation sequence. It differs from a regular deferred custom action in that it is only executed during a rollback. A rollback custom action must always precede the deferred custom action it rolls back in the action sequence. A rollback custom action should also handle the case where the deferred custom action is interrupted in the middle of execution. For example, if the user were to press the Cancel button while the custom action was executing.

## Insert Custom Action DLL into MSI Setup Project

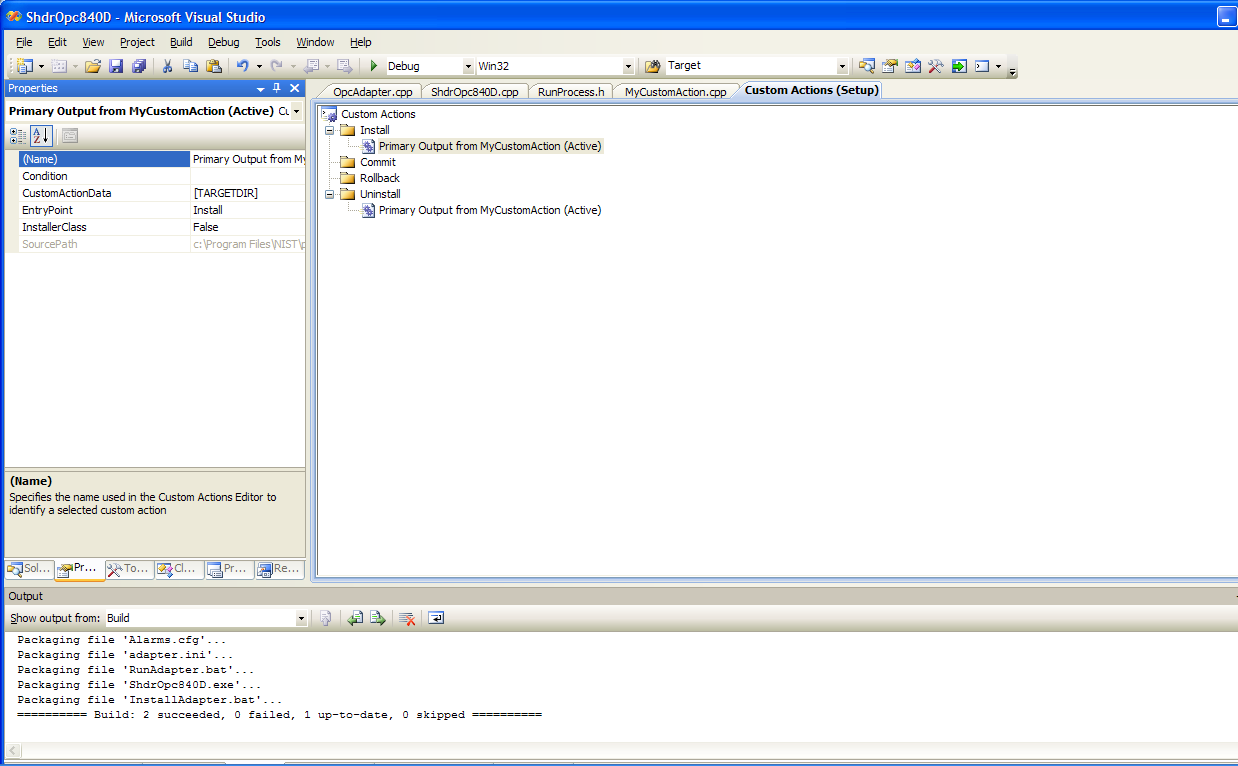
Compile and then insert the DLL into setup project file area as a Project Output:



Then add the DLL to the Custom Action to the Install folder by right clicking (ADD) and then selecting it from the browse folder dialog.



Note the entry point point matches the Install subroutine. If you will want to pass parameters, you will need to add them to the CustomActionData field, in the screen shot case below, we will send in [TARGETDIR] - note capitalization and brackets are requires. You can only read the entire CustomActionData field in the DLL Install method.



Here is the code to read the CustomActionData (note capitalization) in the DLL project

extern "C" UINT \_\_stdcall Install(MSIHANDLE hInstall)

{

TCHAR szBuffer1[MAX\_PATH] = {'0'};

DWORD dwLen = MAX\_PATH;

// Returns ERROR\_SUCCESS, ERROR\_MORE\_DATA, ERROR\_INVALID\_HANDLE, ERROR\_BAD\_ARGUMENTS

UINT hr = MsiGetProperty(hInstall, \_T("CustomActionData"), szBuffer1, &dwLen);

std::string path(szBuffer1);

}

Note, each time you MsiGetProperty reset dwLen to the MAX\_PATH. You can search the internet for ways to read large portions of CustomActionData.

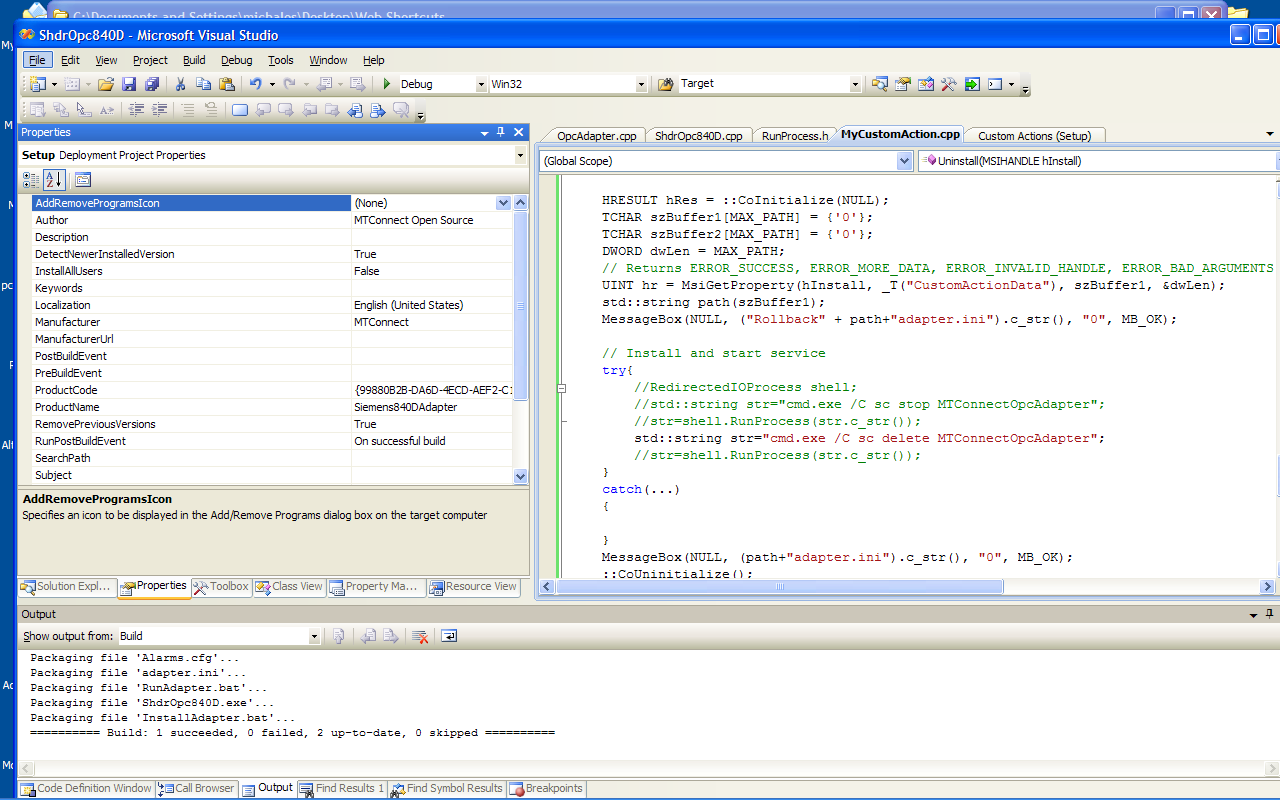
Using this path, you can then configure an Installation file for example.

## Reinstalling Issues

Click the Project Icon, (below Setup), then make sure no Install item is open, then click Properties tab next to solution

DetecNewerInstallation - True

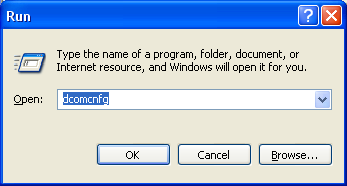
To make sure custom action is invoked, Set RemovePreviousVersions - True



# Configuring DCOM for MTConnect

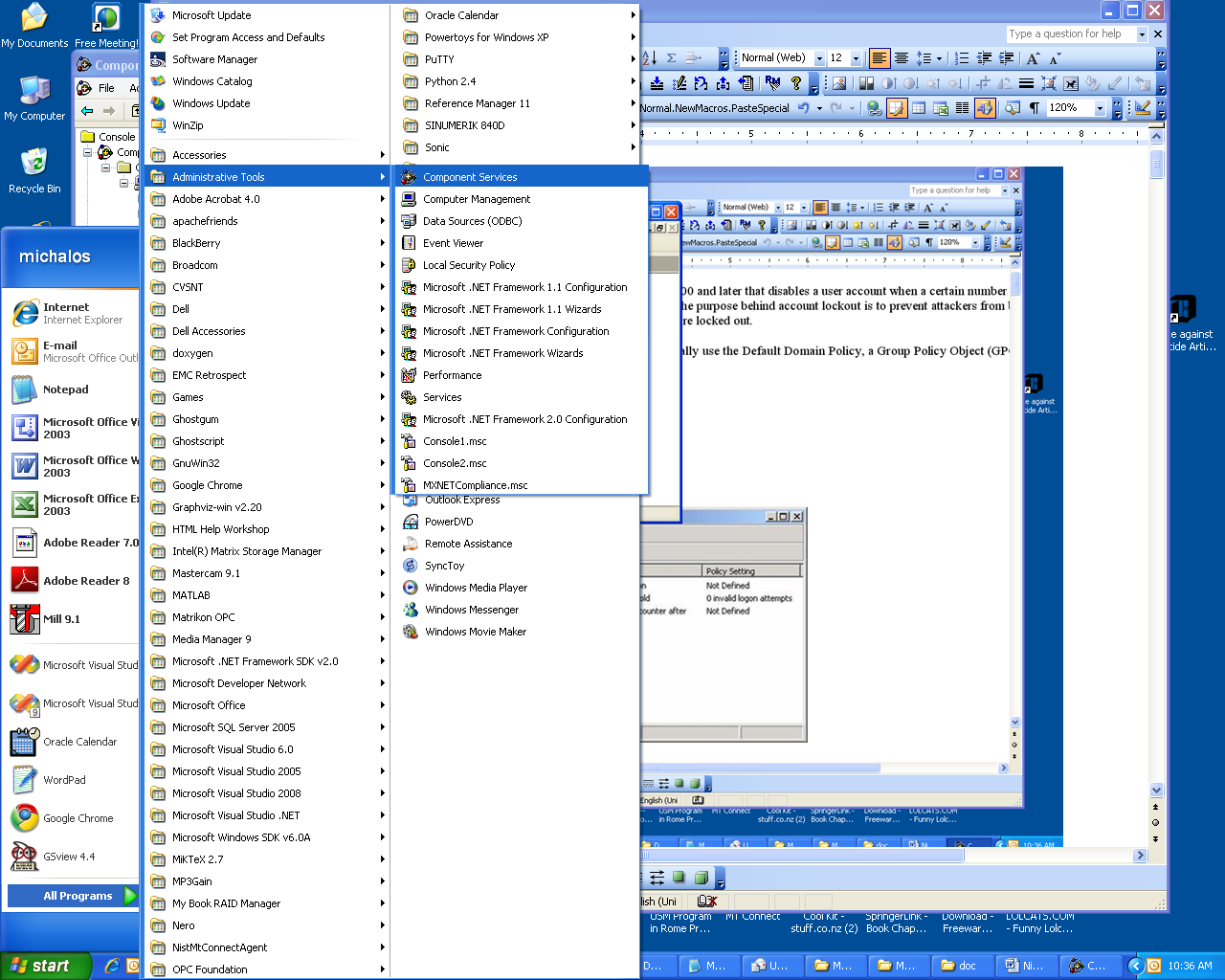
## Run dcomcnfg

Either Start->Run dcomcnfg from the command line



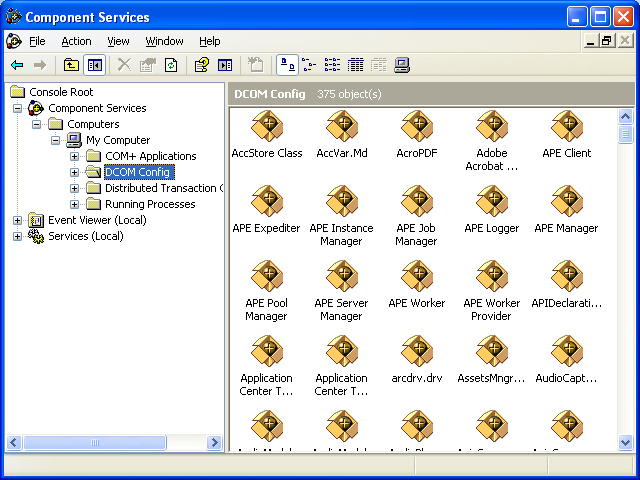
Or

Start Administration Tools->Cmponent Services



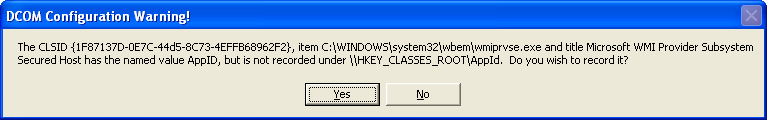
## Open DCOM config inside Component Services

Click Component Services -> Computers -> My Computer



Note you may get a bunch of DCOM configuration warnings: IGNORE THEM.

Example:



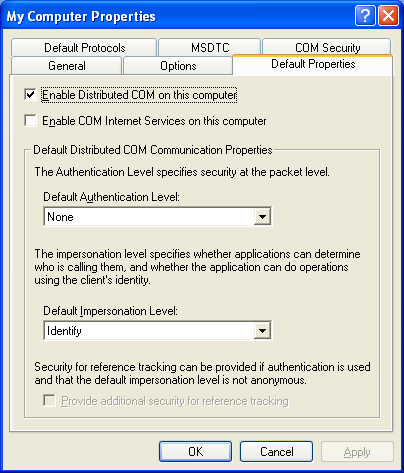
## Right Click My Computer and Select Properties

Make sure Enable Distributed COM on this computer is checked.

## SID: see that these two settings have been configured for DST 444.

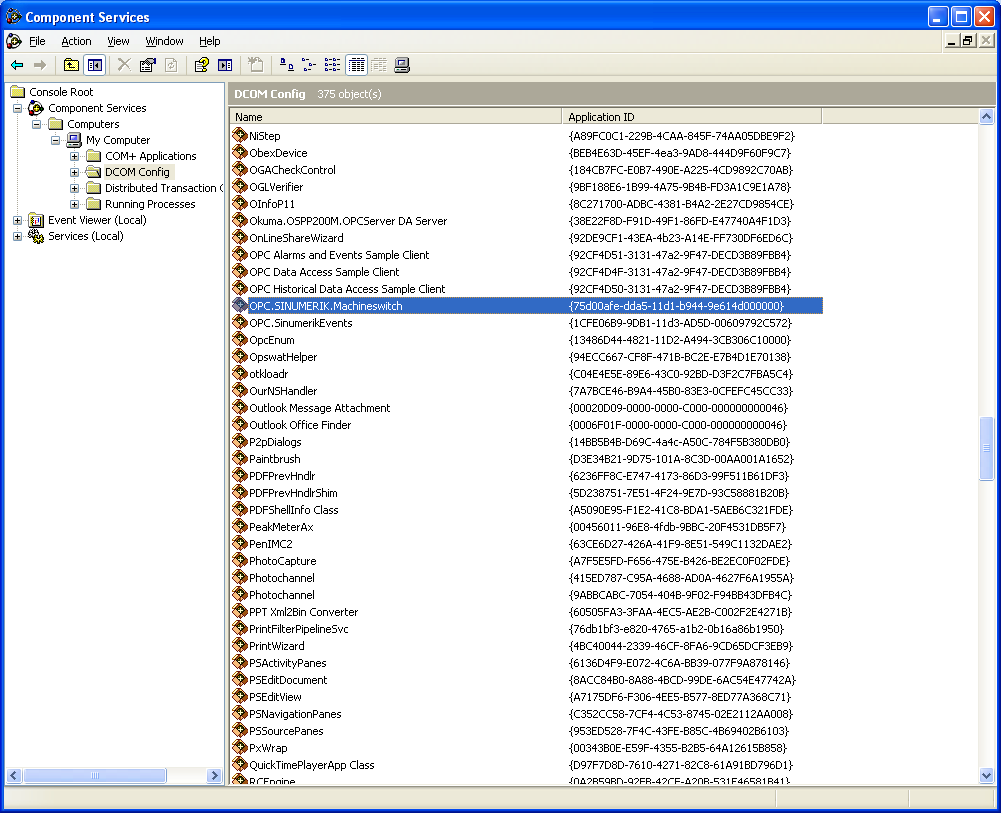
Set Authentication to None

Set Impersonation to Impersonate.

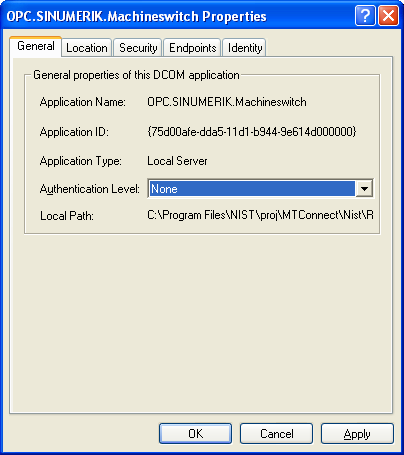


## Set Siemens 840D OPC COM server Properties

Select DCOM Config, (use List view option) and then right click on OPC.SINUMERIK.MachineSwitch and select Properties.

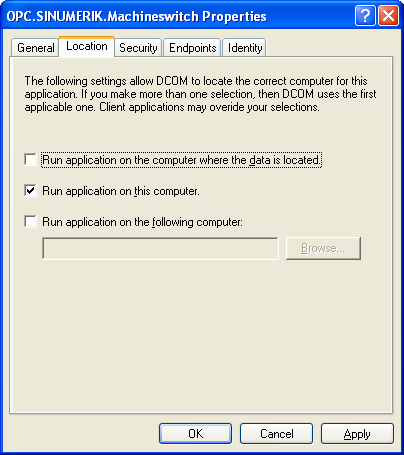


## Set Siemens OPC Server Authentication Level to None



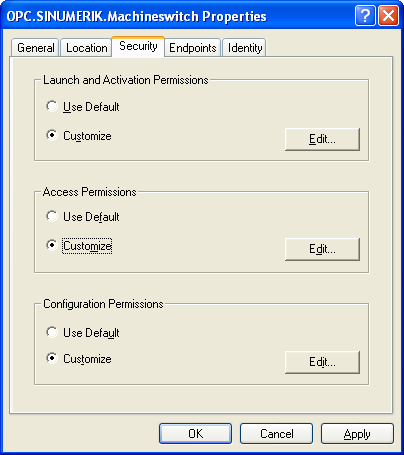
## Location

Run the OPC server on this machine



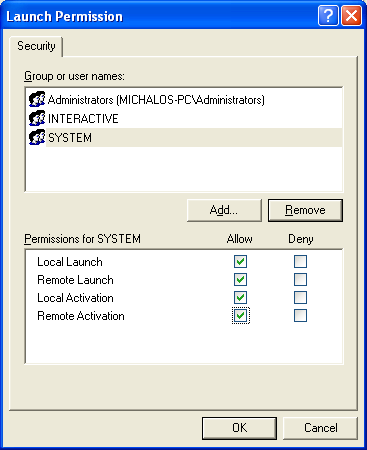
## Security

Check customize all Launch, Activation, and Configuration Permissions

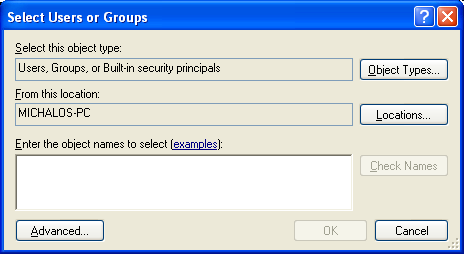


## Launch and Activation Permissions

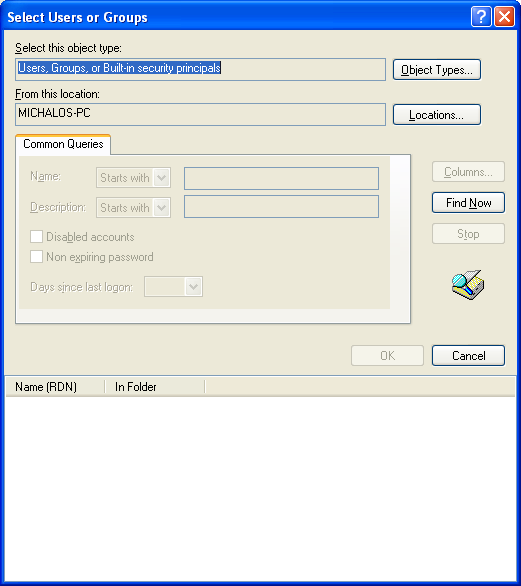
Make sure System has permissions for Remote Launch and Activation



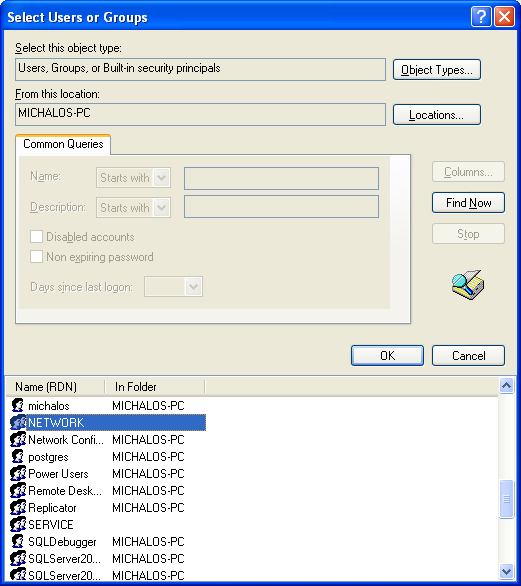
Add Network (not sure this is necessary). Select Add, Select users or Groups Dialog pops up



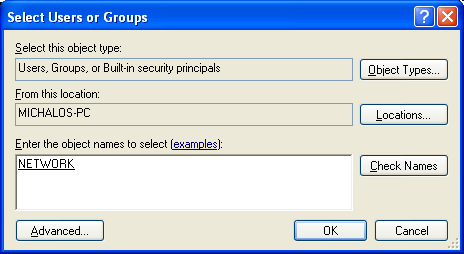
Select Advanced



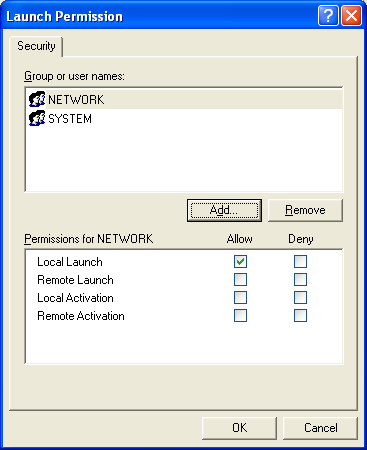
Click Find Now



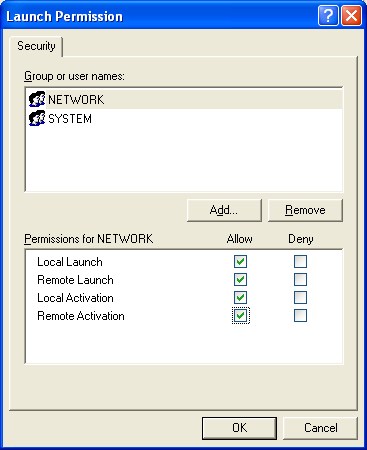
Select network and click OK



Select OK.



Allow Remote Lauch and Activation and Click OK



## Access Permissions

Check customize all Launch, Activation, Access and Configuration Permissions

Make sure System has permissions Remote Launch and Activation