TH Köln – University of Applied Sciences
Institute of Automation & IT

Case Study Report

OPC Interface Development and Performance Analysis of CODESYS, HIMA Servers and UNISIM Client

Submitted to:

Prof. Dr. Rainer Scheuring
January 2020

Submitted by:

Ayesha Noureen (11136999)
Elaheh Najafani (11136287)
Reeshika Sundaram (11134553)
Sunitha Radhakrishnan (11135196)

Technology
Arts Sciences
TH Köln

ABSTRACT

The goal of this case study is to develop an interface between CODESYS and UNISIM. CODESYS is the leading manufacturer-independent IEC 61131-3 automation software for control systems. CODESYS provides the OPC server "CODESYS OPC DA Server SL". UniSim will act as OPC client. Also, communication between the Interface was implemented and tested using MATRIKON Simulation Server in one machine and MATRIKON OPC Explorer as Client in other machine. For this client-server interaction on different computers DCOM technology (Distributed Component Object Model) was used. DCOM works when client program objects request services from server program objects on different computers in the same network. UNISIM Client was also connected to HIMA Server installed on another device using DCOM and the performance tests were carried out.

Keywords: Unisim, Codesys, HIMA, DCOM.

Table of Contents

SECTION 1

INTRODUCTION

In this case study OPC-DA environment is set up between client and server Machine. The Server Machine is set up on Microsoft Windows 10 Home and the client Machine is Microsoft Windows 7. The main Chapter deals with the Codesys automation software that is Installed on the Server machine and the PLC coding for the PID controller is done and uploaded on a virtual PLC supported by Codesys. The data is Fetched from the Codesys to the Unisim, installed on the Server Machine.

The UniSim design tool was provided by the University, the Codesys Development tool and a DA server were downloaded from the vendor's site with a trial limited license. Initially made the setup with Matrikon Simulation Server to check the communication and Read/Write from and to the server was done using ‘Tags’. Then the application was extended using Codesys DA server and finally also with HIMA server. It should be highlighted that Windows 10 Home edition of MS Windows is not allowed to use DCOM configuration which is the base for DA communication due to some built restrictions. There were certain files and configuration that was needed to be done in windows 10 Home to enable the configuration to get over with this problem that it explained in the further chapter. DCOM uses TCP/IP and Hypertext Transfer Protocol. DCOM comes as part of the Windows operating systems.

To enable the communication the client and server machine should be in the same Network and so connected our Client-Server Machine under the LAN (EDUROM) of the University.

CODESYS DA server, HIMA Server, Matrikon Simulation Server

Honeywell Unisim - client

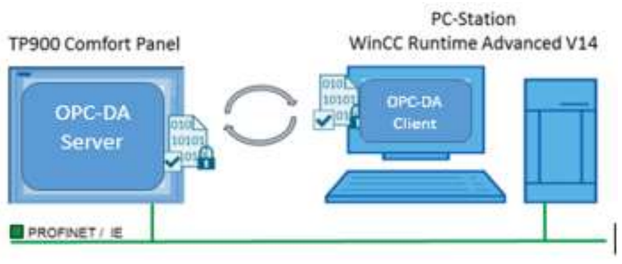


Figure1: Client-Server Architecture

SECTION 2

ABOUT OPC

2.1 NEED FOR OPC

A factory Automation system consist of various devices and controller generated by different vendors based on different protocols. It is necessary to communicate these devices with the Business and management system. OPC provides an environment to enable such communication and to extract real time data from the plant. OPC reduces the need of additional hardware driver for converting proprietary protocols to client protocol.

Automation Industries is based on the Microsoft operating system, Automation vendors uses Microsoft's COMs and DCOM setting to enable the communication with the Business and management system acting as Human Machine Interface (HMIs).

2.2 OPC FUNDAMENTALS

OPC (OLE for Process Control) is a client / Server based communication which means that you have one or more sever waiting for the request of several clients. Once server gets a request from the client machine it responds to that and then goes to the wait state Client Instruct the server to send updates when such comes into the server. In OPC it's the client that decide when and what data a server should fetch from the underlying system. Client also decide how often a server should quire a system.

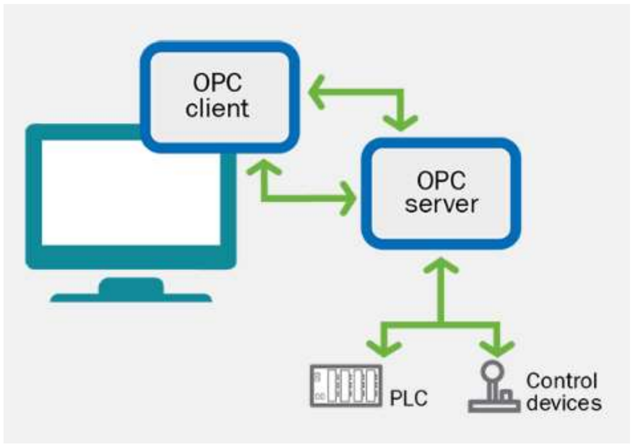


Figure2: OPC Client-Server devices with Controller (PLC)

2.3 OPC PROTOCOLS

The different classical OPC protocols are completely self-sustained and have nothing in common. That means that the quality field in DA have no connection to the same field in HAD. Currently in the classic OPC model you have the following protocols; **DA** (Data access), **AE** (Alarm & Events), **HDA** (Historical Data Access), **XML DA** (XML Data Access) and finally **DX** (Data Xchange). Each of these protocols have their own read, write, etc. commands that only affect one protocol at the time. That is true even if one OPC server supports several of the protocols. The most commonly used and oldest protocol is the data access (DA) and in the upcoming chapter regarding this protocol is explained in details

2.4 OPC-DA

The most basic protocol of the OPC stack is the Data Access protocol that gets data out of the control systems into other systems on the shop floor. Each information about a specific tag or data point contains some information about it. First you have the data itself and that is called Value and of course the Name of it. To that comes a number of other pieces of information that describes the information, the first is the Timestamp that gives you the exact time when the value was read. This timestamp can be taken either directly from the underlying system or assigned to it when the data is read in the OPC server. The last piece is called Quality which gives a basic understanding if the data is valid or not.

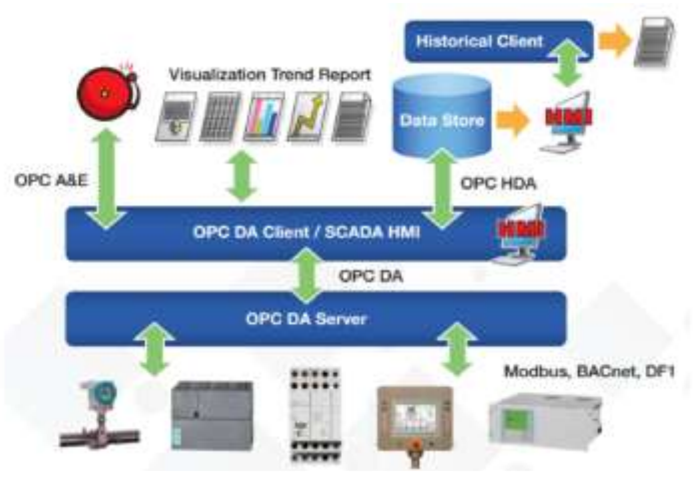


Figure3: OPC DA Interface

2.5 OPC UA vs OPC DA

S.NO	OPC DA	OPC UA
1	The transfer of the information that takes place between the server and client in OPC DA is only VQT that stands for Value Quantity and Time.	OPC UA which is the new version of OPC provides data and information modelling as well as many properties that can be shared between the client and server about a variable.
2	As OPC DA comes under OPC classic model, it supports DCOM communication for connecting the client and server, where DCOM is dependent on the operating system and supports only Window OS.	OPC UA do not rely on DCOM communication for connecting client and server, thus it is a platform or OS independent and it supports platform such as Linux (Java), Apple, or Windows.
3	OPC DA allows to access only the current data and is incapable to generate alarms, historical events	OPC UA supports features like historical events, multiple hierarchies and provides methods and programs (that are called commands)
4	One of the limitations of OPC DA is its inadequate security	This security issue is solved in a higher version that is OPC UA
5	It is an OPC DA specification that defines how real-time data can be exchanged between a data source and a data sink (for example a PLC and an HMI) without either of them having to know each other's native protocol.	It is an OPC UA specification that states how real-time data is communicated between machine to machine for industrial automation

For this Project , being laid our Emphasis on OPC DA and it can be explained in details in the upcoming chapters.

SECTION 3

DCOM CONFIGURATION

To establish a communication between two computer i.e Client and Server, DCOM configuration has to be done. There are many ways to setup this DCOM configuration. The system was made to have one lab PC Scheuring-701 PC with windows 7 OS acting as a client and one laptop having Windows 10 home acting as a server. There are different ways to add **user settings** and **local security setting** in Windows 10 Home. These are following steps to do in DCOM configuration.

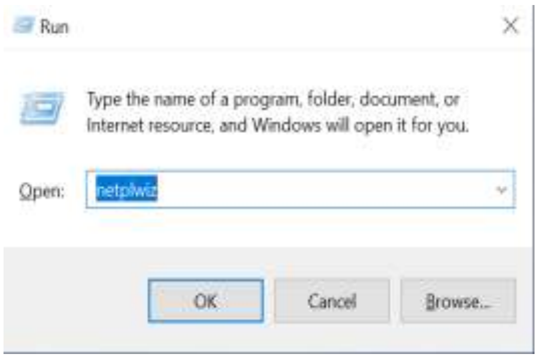
1. Configure your computer (Client and Server) to be a member of your network
2. Create Local User Account Settings in both Client and Server PC.
3. Configure DCOM users and Group
4. Configuration of Windows Firewall (Adding port and program rules)
5. Check DCOM properties in component Services (set default properties, default protocols, COM setting, Activating guest access i.e (local security policies)

3.1 Setting up User account:

In this case study, windows 10 Home was the server machine, so the steps to do DCOM configuration is bit different as compared to that of windows 10 Pro.

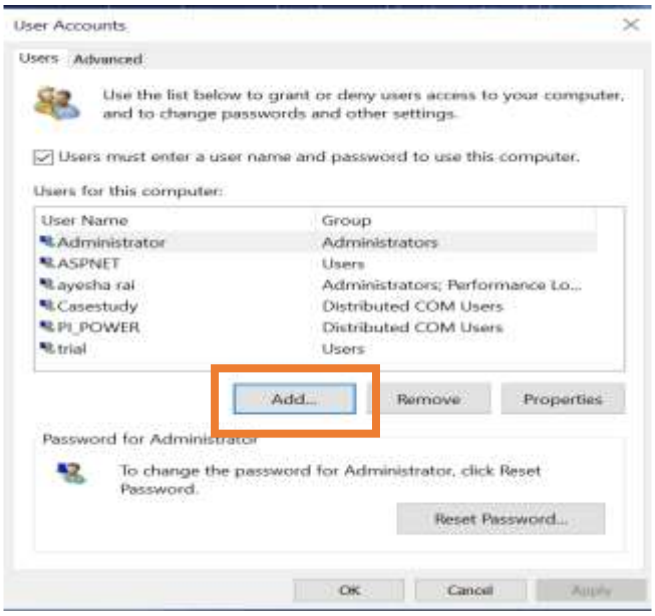
First step is to create a new user account by opening control panel and select User Accounts. A user "Case studies" has been created that has permissions to run and use DCOM application. To increase Security, you can create a user with restricted permissions. You need to have administrator permissions to add a user.

NOTE: In both client and Server, it is a must to have same username and password.



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: netplwiz



Use the list below to grant or deny users access to your computer, and to change passwords and other settings.

[✓] Users must enter a user name and password to use this computer.

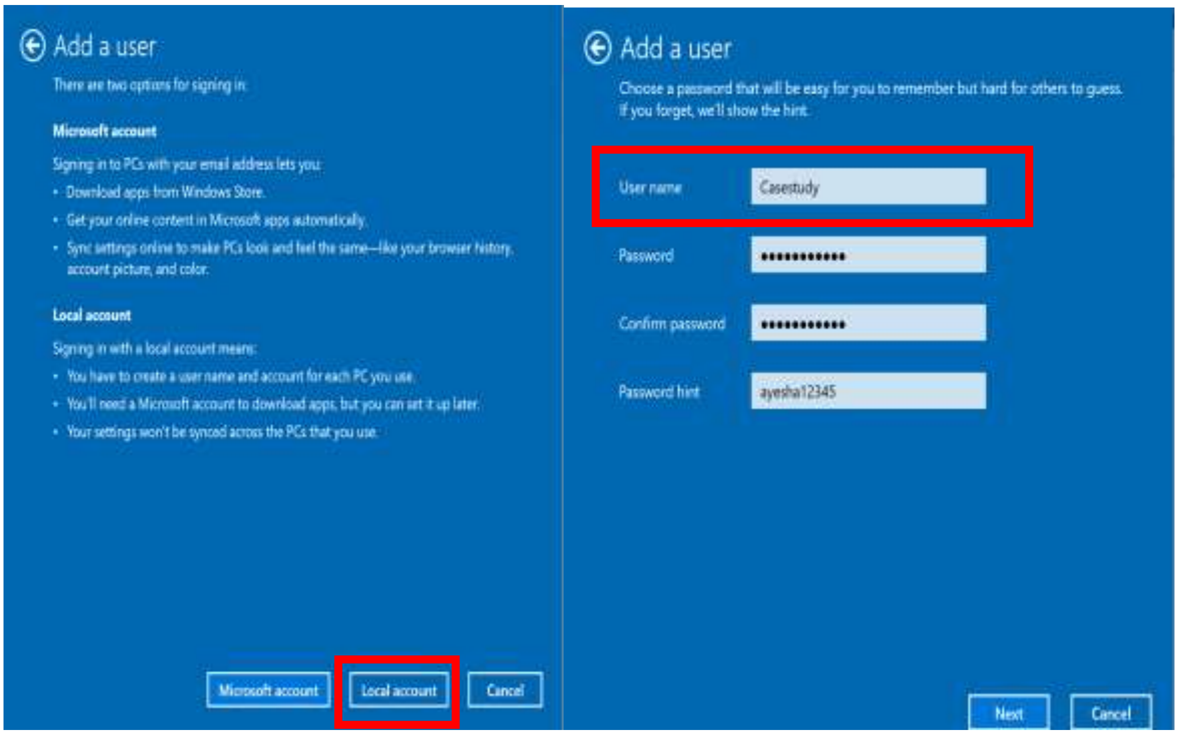
Users for this computer:

User Name	Group
Administrator	Administrators
ASPNET	Users
ayesha rai	Administrators; Performance Lo...
Casestudy	Distributed COM Users
PI_POWER	Distributed COM Users
trial	Users

Password for Administrator

To change the password for Administrator, click Reset Password.

Figure4:User account settings in Server machine



Add a user

There are two options for signing in:

Microsoft account

Signing in to PCs with your email address lets you:
- Download apps from Windows Store.
- Get your online content in Microsoft apps automatically.
- Sync settings online to make PCs look and feel the same—like your browser history, account picture, and color.

Local account

Signing in with a local account means:
- You have to create a user name and account for each PC you use.
- You'll need a Microsoft account to download apps, but you can set it up later.
- Your settings won't be synced across the PCs that you use.

Add a user

Choose a password that will be easy for you to remember but hard for others to guess. If you forget, we'll show the hint.

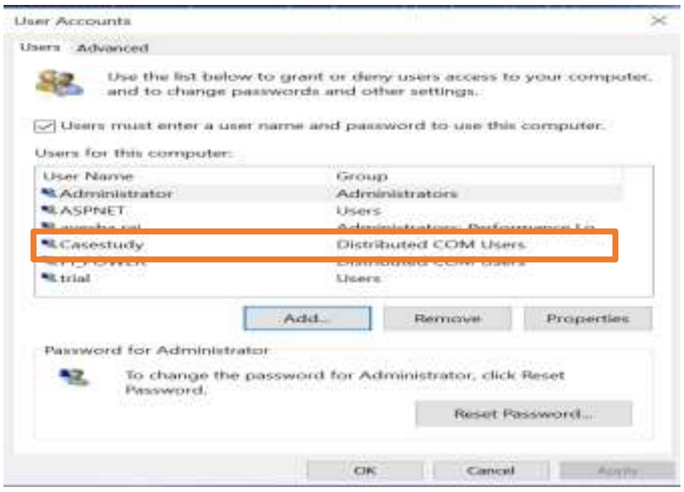
User name	Casestudy
Password	••••••••••••
Confirm password	••••••••••••
Password hint	ayesha12345

Figure5:Choosing Local account

Add a local account and mention the credentials. This has to be taken care when the OS is different. Now the User "Casestudy" is created.

3.2 Adding Users to DCOM Group and Giving Permissions:

Now add users and DCOM group for this particular user using the command "netplwiz" in Windows 10 Home. Note: In Windows 10 Pro, use "lusrmgr.msc". As there is already a User account as "Casestudy" and specified it under the group of distributed COM Users hence it is displayed in the list. To specify the account in the DCOM group steps are discussed in the further sections:



User Accounts

Users Advanced

Use the list below to grant or deny users access to your computer, and to change passwords and other settings.

☑ Users must enter a user name and password to use this computer.

Users for this computer:

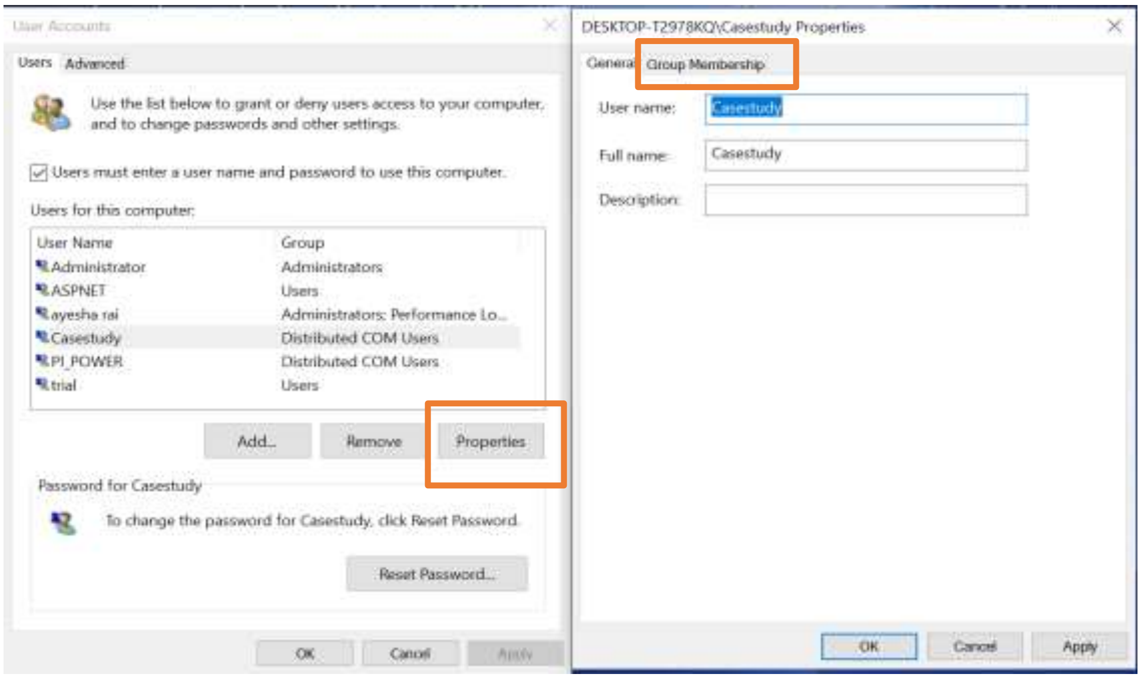
User Name	Group
Administrator	Administrators
ASPNET	Users
[illegible]	[illegible]
Casestudy	Distributed COM Users
[illegible]	Distributed COM Users
trial	Users

Password for Administrator

To change the password for Administrator, click Reset Password.

Figure6: User account 'Case study' created

Once a User account is created, select properties and go to Group Membership



User Accounts

Users Advanced

Use the list below to grant or deny users access to your computer, and to change passwords and other settings.

☑ Users must enter a user name and password to use this computer.

Users for this computer:

User Name	Group
Administrator	Administrators
ASPNET	Users
ayesha rai	Administrators; Performance Lo...
Casestudy	Distributed COM Users
PI_POWER	Distributed COM Users
trial	Users

Password for Casestudy

To change the password for Casestudy, click Reset Password.

DESKTOP-T2978KQ\Casestudy Properties

General Group Membership

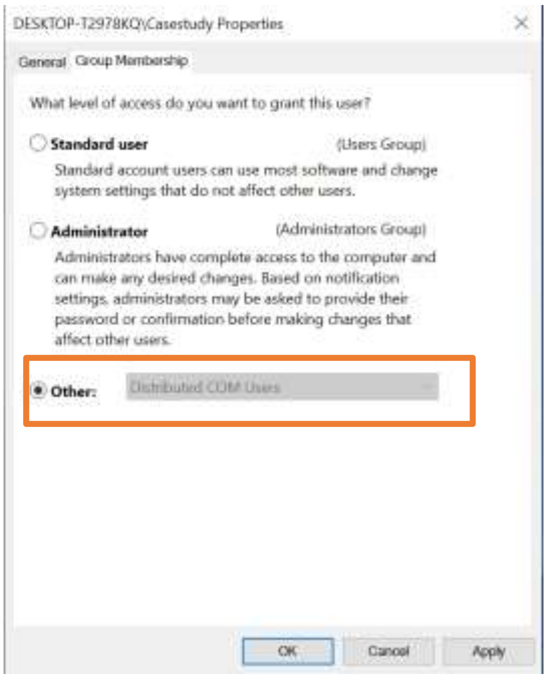
User name: Casestudy

Full name: Casestudy

Description:

Figure7: Steps to Create Group and adding users to group

Select others and specify the user account "Case study" under Distributed COM User group



DESKTOP-T2978KQ\Casestudy Properties

General Group Membership

What level of access do you want to grant this user?

○ **Standard user** (Users Group)

Standard account users can use most software and change system settings that do not affect other users.

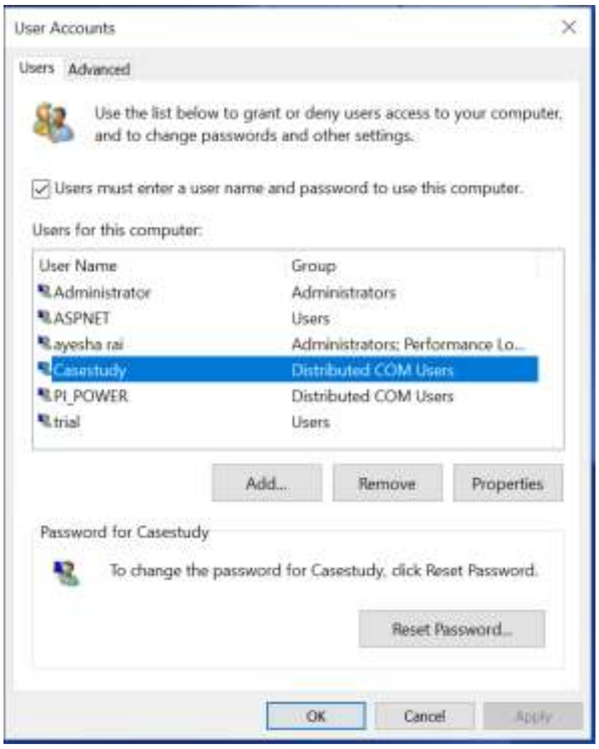
○ **Administrator** (Administrators Group)

Administrators have complete access to the computer and can make any desired changes. Based on notification settings, administrators may be asked to provide their password or confirmation before making changes that affect other users.

● **Other:** Distributed COM Users

Figure8: After creation of group

Once all the steps are followed a User account named 'Case study' is specified under the 'DCOM users' group is created.


User Accounts

Users | Advanced

Use the list below to grant or deny users access to your computer, and to change passwords and other settings.

☑ Users must enter a user name and password to use this computer.

Users for this computer:

User Name	Group
Administrator	Administrators
ASPNET	Users
ayesha rai	Administrators; Performance Lo...
Casestudy	Distributed COM Users
PI_POWER	Distributed COM Users
trial	Users

Password for Casestudy

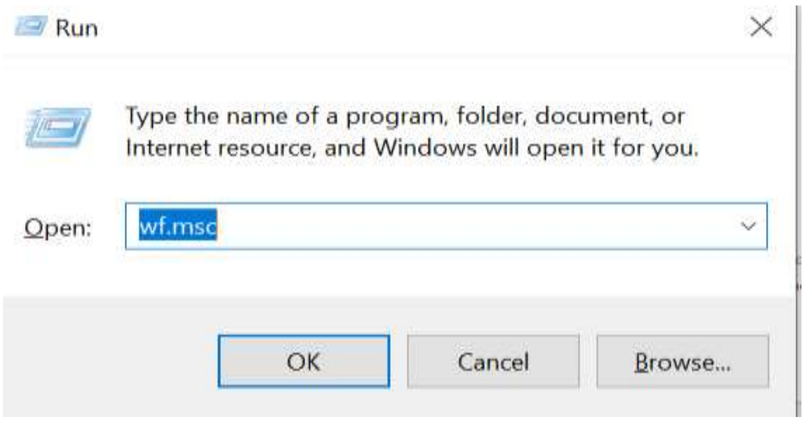
To change the password for Casestudy, click Reset Password.

Figure9: Final settings shown

NOTE: You can also create several users and add them to one DCOM users group or You can add existing users to this group.

3.3 Configuring the Windows Defender Firewall for DCOM and OPC in Client and Server:

Firewall configuration is mandatory when client and server are in different computers. The DCOM technology is reserved for port number 135. For client and Servers to run normally you should permit this connection in firewall. Below are the steps for this configuration:

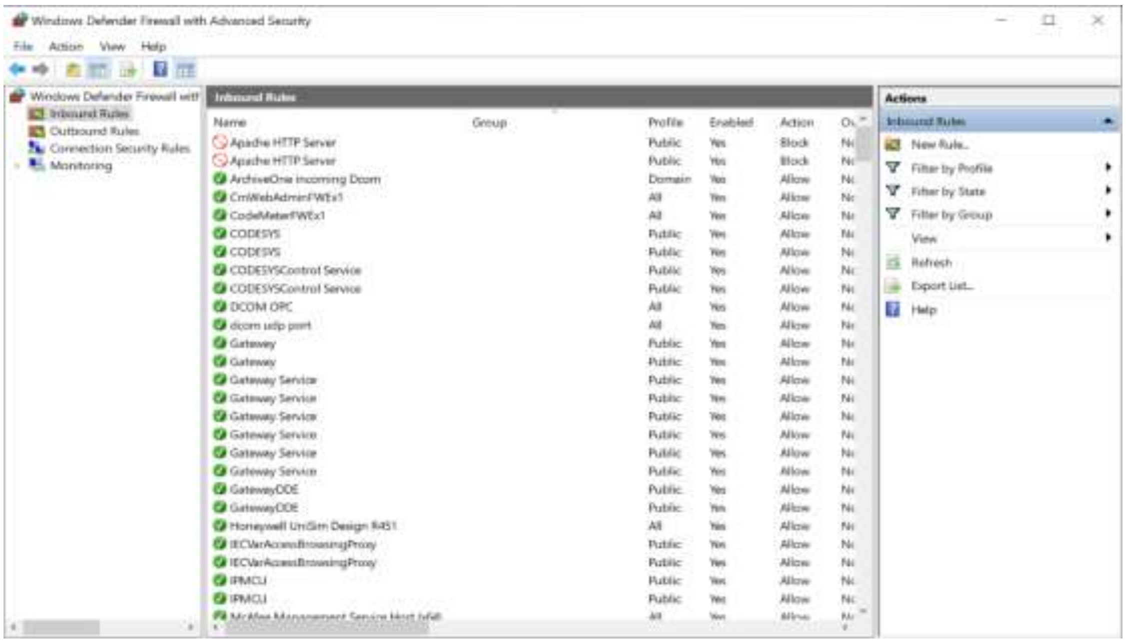

Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: wf.msc

Figure10: Shortcut for Windows Firewall settings

For Windows 10 and Windows 7: The shortcut to open Windows Defender Firewall is as shown above wf.msc



Inbound Rules

Name	Group	Profile	Enabled	Action	Ov
Apache HTTP Server		Public	Yes	Block	No
Apache HTTP Server		Public	Yes	Block	No
ArchiveOne incoming Dcom		Domain	Yes	Allow	No
CmWebAdminFWEx1		All	Yes	Allow	No
CodeMeterFWEx1		All	Yes	Allow	No
CODESYS		Public	Yes	Allow	No
CODESYS		Public	Yes	Allow	No
CODESYSControl Service		Public	Yes	Allow	No
CODESYSControl Service		Public	Yes	Allow	No
DCOM OPC		All	Yes	Allow	No
dcom udp port		All	Yes	Allow	No
Gateway		Public	Yes	Allow	No
Gateway		Public	Yes	Allow	No
Gateway Service		Public	Yes	Allow	No
Gateway Service		Public	Yes	Allow	No
Gateway Service		Public	Yes	Allow	No
Gateway Service		Public	Yes	Allow	No
Gateway Service		Public	Yes	Allow	No
Gateway Service		Public	Yes	Allow	No
GatewayDDE		Public	Yes	Allow	No
GatewayDDE		Public	Yes	Allow	No
Honeywell UniSim Design R451		All	Yes	Allow	No
IECVarAccessBrowsingProxy		Public	Yes	Allow	No
IECVarAccessBrowsingProxy		Public	Yes	Allow	No
IPMCU		Public	Yes	Allow	No
IPMCU		Public	Yes	Allow	No
McAfee Management Service Host (x64)		All	Yes	Allow	No

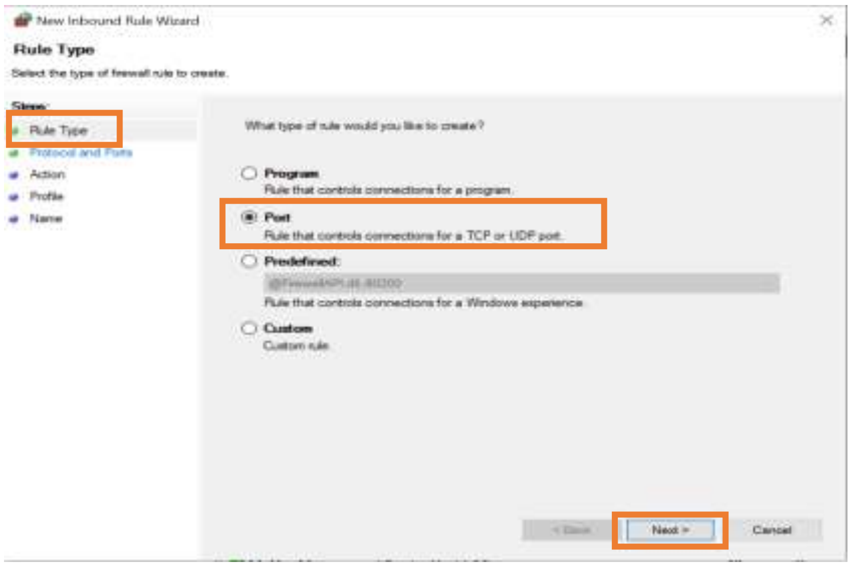
Figure11: Select 'New Rule' from Inbound rules tab

Here in the above pop-up window, specify the New Rules. For OPC clients to be able to connect the OPC server create two port rules and one Program rule for server setup.

1. Port rule for TCP and UDP protocol is 135
2. Program Rule

3.3.1 Port rule:

By default, Windows Defender Firewall allows all outbound network traffic unless it matches a rule that prohibits the traffic. To block outbound network traffic on a specified TCP or UDP port number, use the Windows Defender Firewall with Advanced Security node in the Group Policy Management console to create firewall rules. This type of rule blocks any outbound network traffic that matches the specified TCP or UDP port numbers.



Rule Type

Select the type of firewall rule to create.

What type of rule would you like to create?

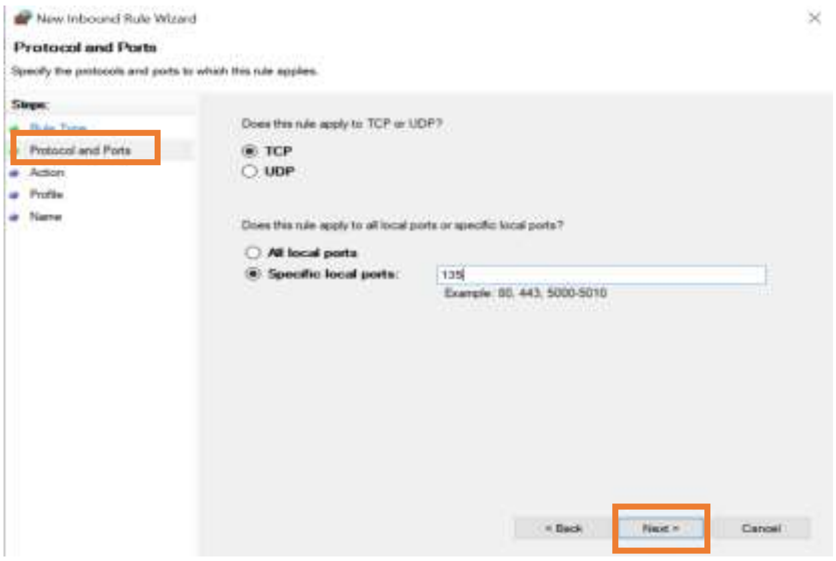
○ **Program**
Rule that controls connections for a program.

[✓ ● **Port**]
Rule that controls connections for a TCP or UDP port.

○ **Predefined:**
@FirewallAPI.dll,-80200
Rule that controls connections for a Windows experience

○ **Custom**
Custom rule.

Figure12: Select 'Port Rule' and Click Next


New Inbound Rule Wizard

Protocol and Ports

Specify the protocols and ports to which this rule applies.

Does this rule apply to TCP or UDP?

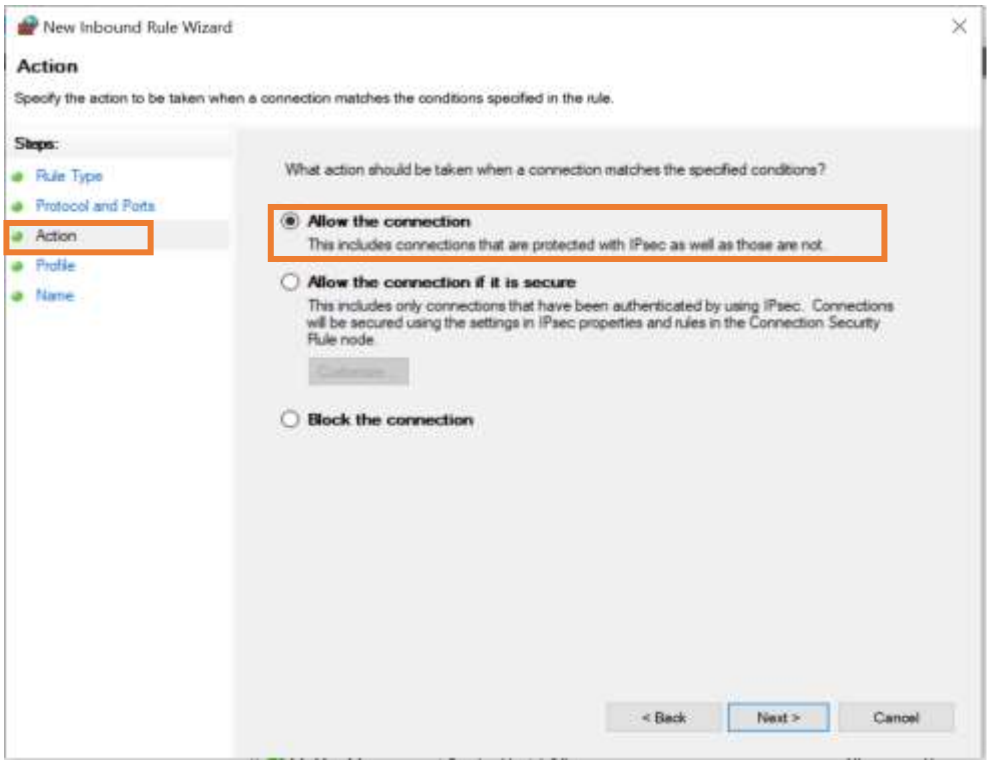
[✓ TCP] UDP

Does this rule apply to all local ports or specific local ports?

All local ports [✓ Specific local ports:] 135

Example: 80, 443, 5000-5010

Figure13: Specify TCP and local port number as 135 and repeat same for UDP


New Inbound Rule Wizard

Action

Specify the action to be taken when a connection matches the conditions specified in the rule.

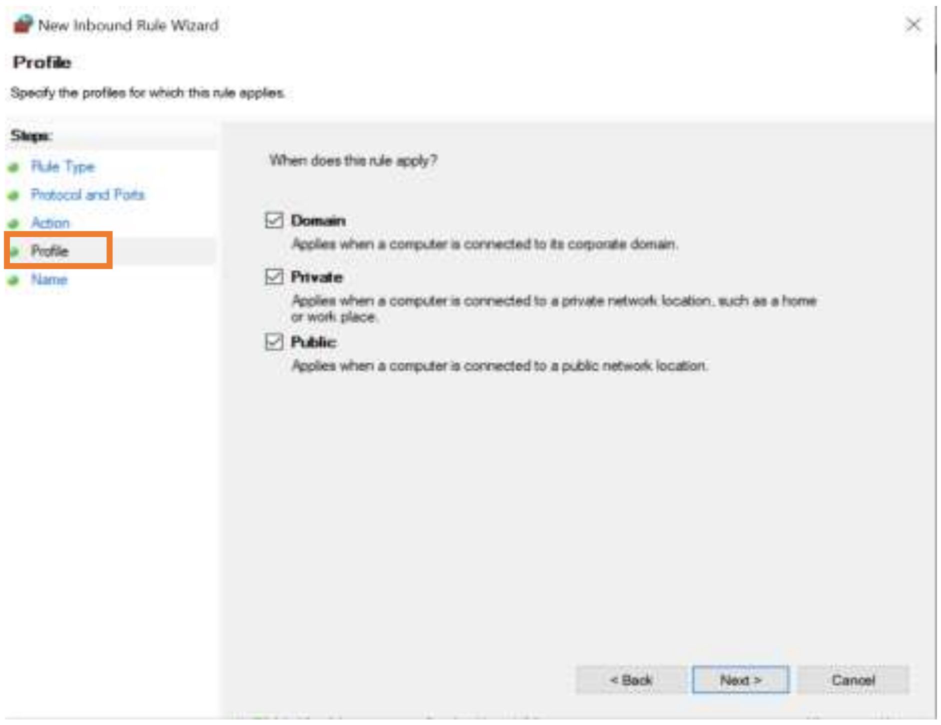
What action should be taken when a connection matches the specified conditions?

[✓ **Allow the connection**]
This includes connections that are protected with IPsec as well as those are not.

Allow the connection if it is secure
This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.

Block the connection

Figure14: In Action tab, Allow the Connection and Give Next


New Inbound Rule Wizard

Profile

Specify the profiles for which this rule applies.

Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

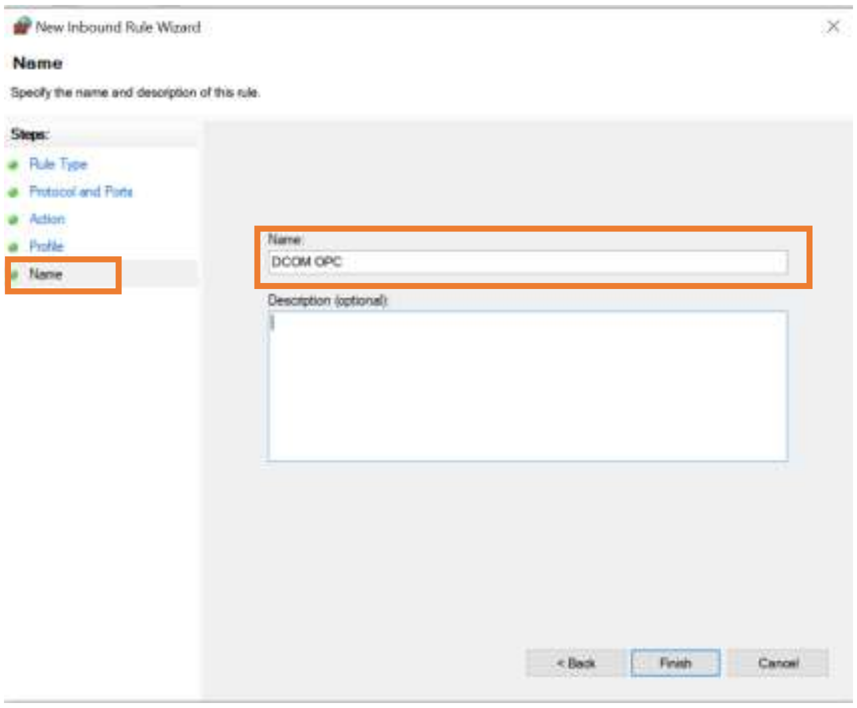
When does this rule apply?

☑ **Domain**
Applies when a computer is connected to its corporate domain.

☑ **Private**
Applies when a computer is connected to a private network location, such as a home or work place.

☑ **Public**
Applies when a computer is connected to a public network location.

Figure15: In Profile tab, Check all 3 boxes


New Inbound Rule Wizard

Name

Specify the name and description of this rule.

Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

Name:
DCOM OPC

Description (optional):

Figure16: Specify the name of the new inbound rule, in our case it is DCOM OPC.

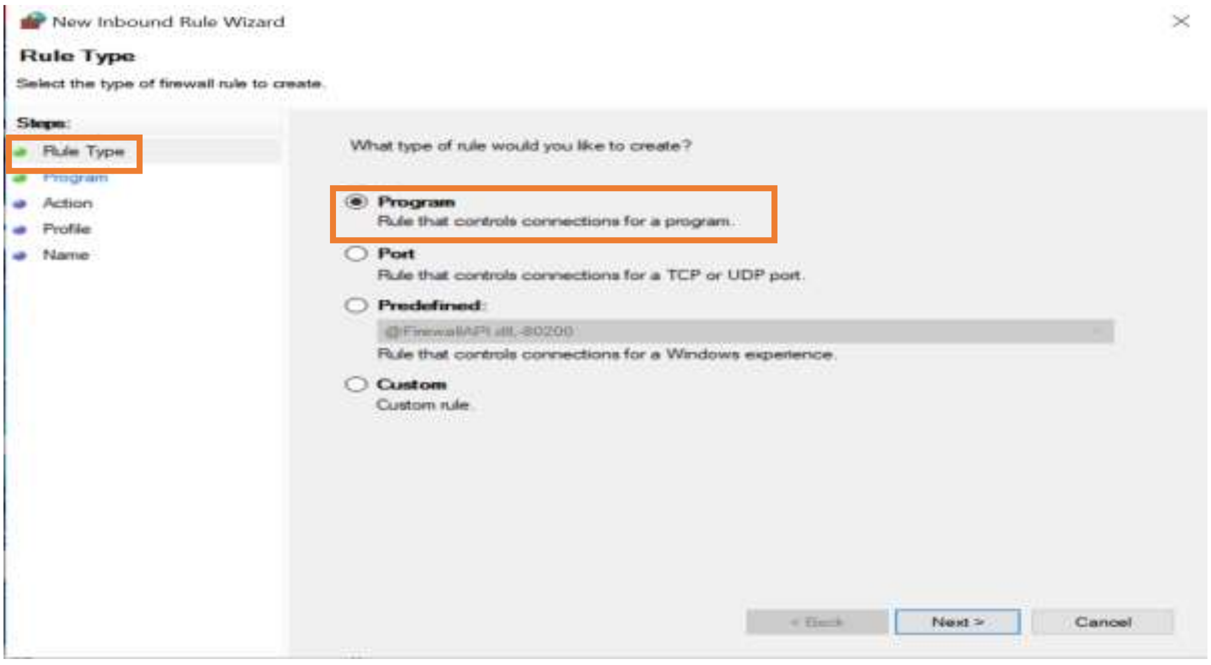
Now the Port rule for TCP is created and the same procedure is followed with the UDP

3.3.2 Configuring the Program rule for Server setup:

To allow inbound network traffic to a specified program or service, use the Windows Defender Firewall with Advanced Security node in the Group Policy Management MMC snap-in to create firewall rules. This type of rule allows the program to listen and receive inbound network traffic on any port.

You need to permit activity for every OPC server running on this computer and you should also permit network activity for OPC ENUM settings that allows remote client to receive the list of servers from this computer.

Below is the example of creating a program rule for OPC ENUM.



New Inbound Rule Wizard

Rule Type

Select the type of firewall rule to create.

Steps:
- Rule Type
- Program
- Action
- Profile
- Name

What type of rule would you like to create?

[✓ **Program**] Rule that controls connections for a program.

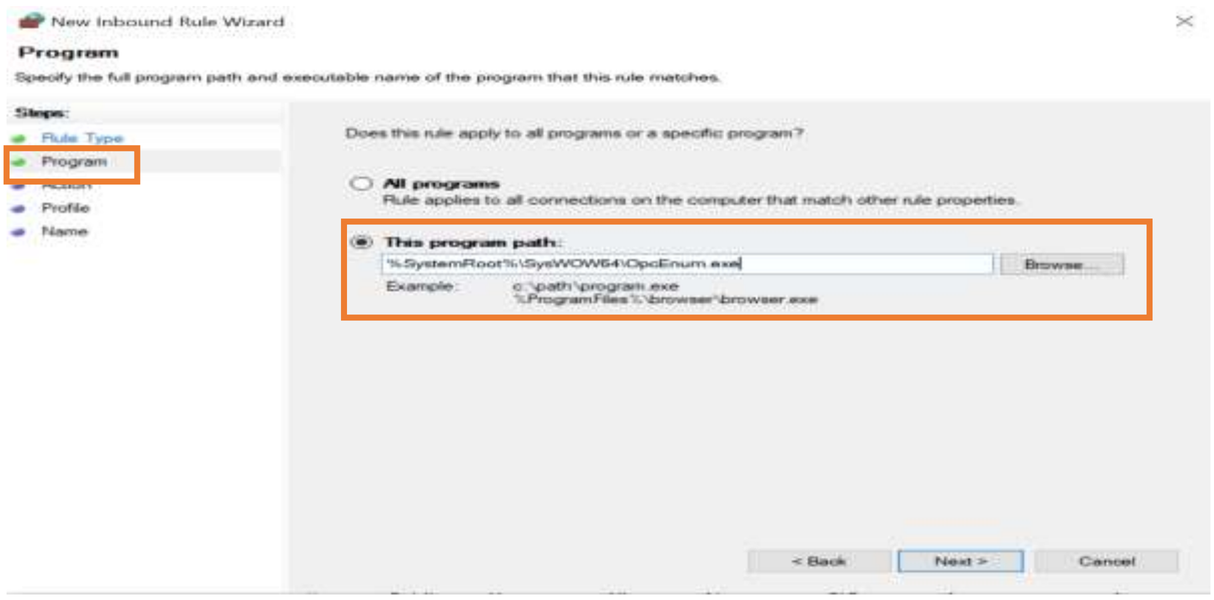
Port Rule that controls connections for a TCP or UDP port.

Predefined: @FirewallAPI.dll,-80200 Rule that controls connections for a Windows experience.

Custom Custom rule

Figure17: Select Program Rule

Click Next and in Action tab “Allow all Connection” and then in Profile tab, apply this rule for Domain, Private and Public and Click Next and Specify the Rule name as “OPC Enum” and Select OK.



New Inbound Rule Wizard

Program

Specify the full program path and executable name of the program that this rule matches.

Steps:
- Rule Type
- Program
- Action
- Profile
- Name

Does this rule apply to all programs or a specific program?

All programs Rule applies to all connections on the computer that match other rule properties.

[✓ **This program path:**] %SystemRoot%\SysWOW64\OpcEnum.exe

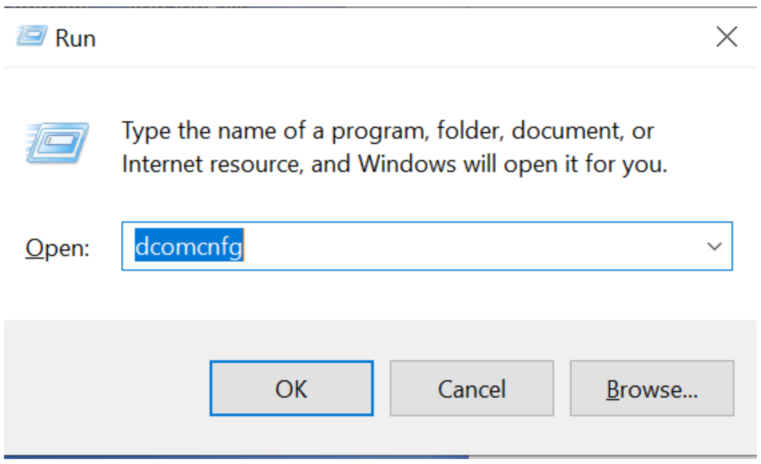
Example: c:\path\program.exe
%ProgramFiles%\browser\browser.exe

Figure18: Specify path of ‘OPCEnum.exe’ file and Name the program rule.

3.4 Specifying DCOM Properties:

For OPC servers to run correctly, you should specify the DCOM network and security properties. Basic Requirement is: OPC Core Components. This has to be installed in the computer according to the bit operating system.

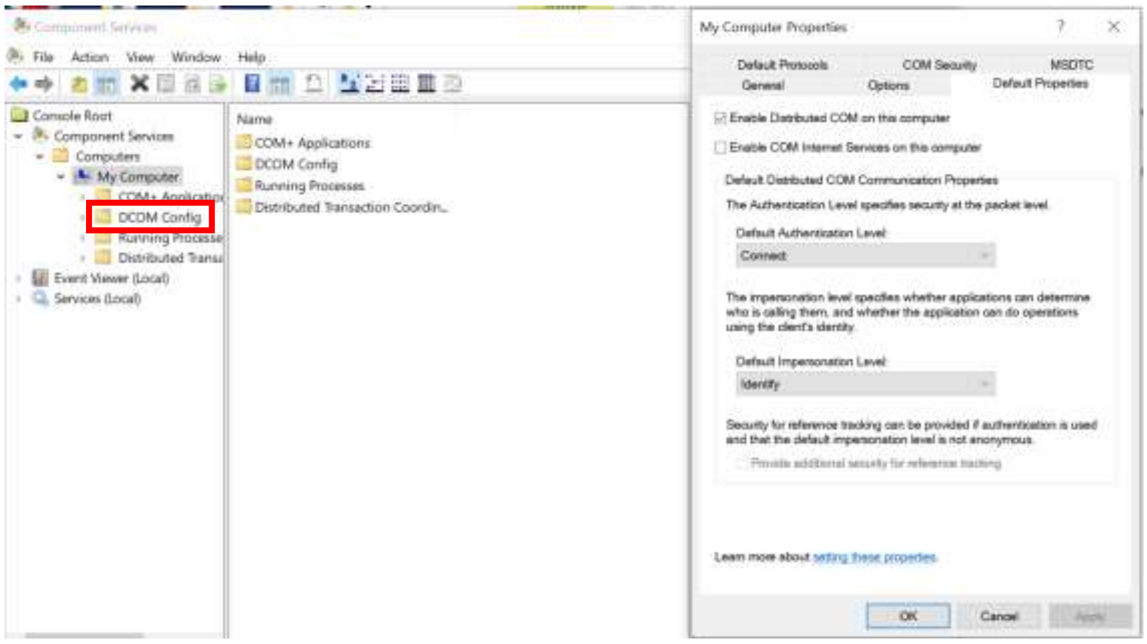
The shortcut key to go to Component Services which has the DCOM settings is "***dcomcnfg***".


Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: dcomcnfg

Figure19: Shortcut key to enter 'DCOM' Configuration settings.


Component Services

Console Root
- Component Services
  - Computers
    - My Computer
      - COM+ Applications
      - DCOM Config
      - Running Processes
      - Distributed Transaction Coordinator
- Event Viewer (Local)
- Services (Local)

Name
- COM+ Applications
- DCOM Config
- Running Processes
- Distributed Transaction Coordin...

My Computer Properties

[✓] Enable Distributed COM on this computer

[ ] Enable COM Internet Services on this computer

Default Distributed COM Communication Properties

The Authentication Level specifies security at the packet level.

Default Authentication Level: Connect

The impersonation level specifies whether applications can determine who is calling them, and whether the application can do operations using the client's identity.

Default Impersonation Level: Identify

Security for reference tracking can be provided if authentication is used and that the default impersonation level is not anonymous.

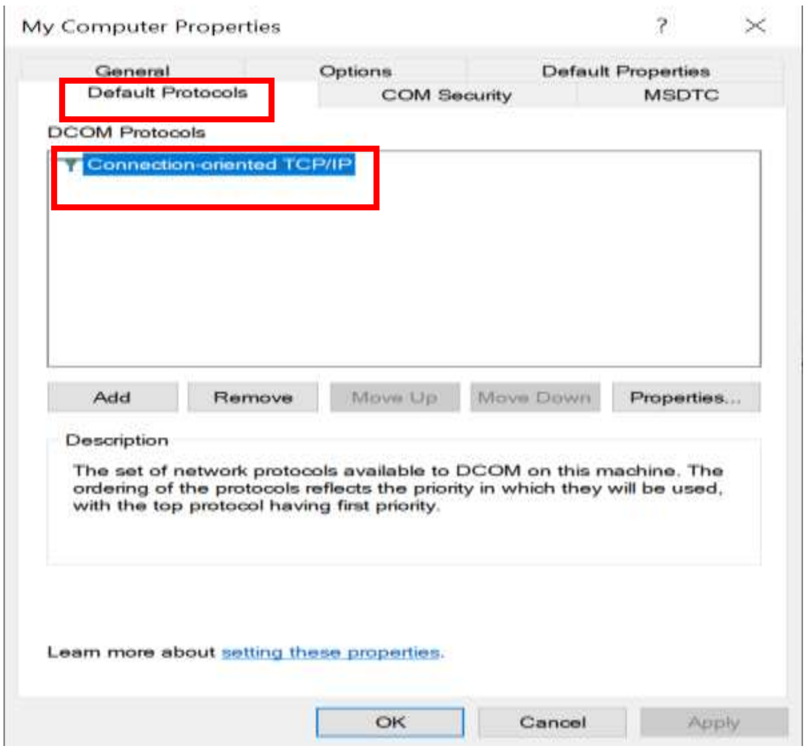
[ ] Provide additional security for reference tracking

Learn more about setting these properties.

Figure20: Default properties tab

Right click on "My Computer" and select properties. The Properties dialog box appears as shown above.

Choose "Default Protocols" tab. It can be noted that more the number of protocols more will be the traffic. It is recommended to have as less protocol to have efficient client-server communication. For our client-server machine the DCOM configuration runs on TCP/IP protocols. It should look like below snap-in


My Computer Properties

General | Options | Default Properties | Default Protocols | COM Security | MSDTC

DCOM Protocols

Connection-oriented TCP/IP

Add | Remove | Move Up | Move Down | Properties...

Description

The set of network protocols available to DCOM on this machine. The ordering of the protocols reflects the priority in which they will be used, with the top protocol having first priority.

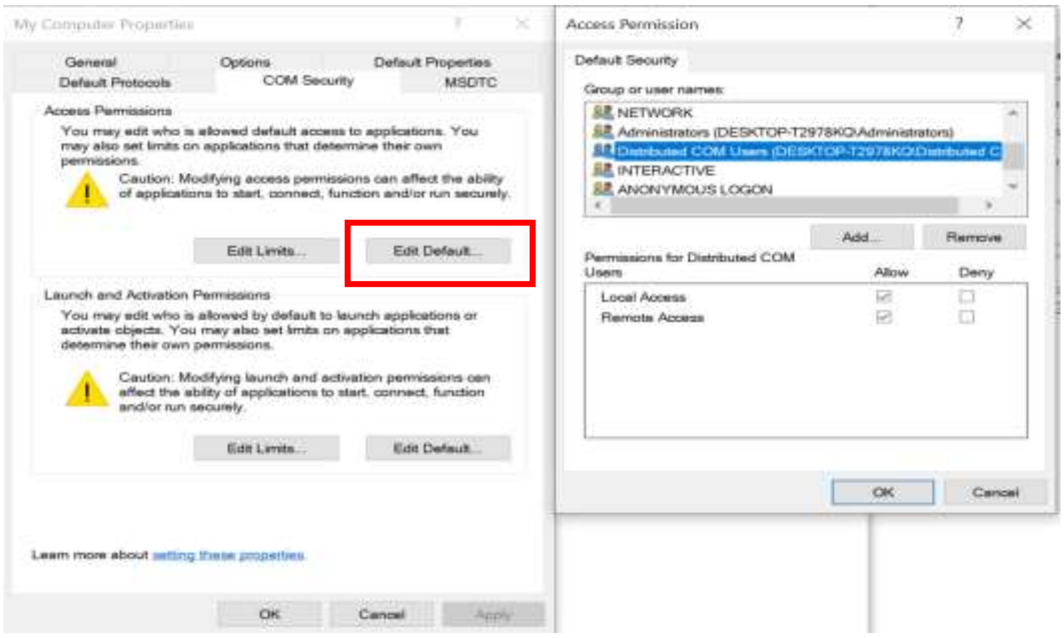
Learn more about setting these properties.

OK | Cancel | Apply

Figure21: Only TCP/IP Protocol should be there.

Now in COM security tab do the following:

In Access Permissions pane, Select "Edit Default" and give permissions for every user and group as shown below:


My Computer Properties

General | Options | Default Properties | Default Protocols | COM Security | MSDTC

Access Permissions

You may edit who is allowed default access to applications. You may also set limits on applications that determine their own permissions.

Caution: Modifying access permissions can affect the ability of applications to start, connect, function and/or run securely.

Edit Limits... | Edit Default...

Launch and Activation Permissions

You may edit who is allowed by default to launch applications or activate objects. You may also set limits on applications that determine their own permissions.

Caution: Modifying launch and activation permissions can affect the ability of applications to start, connect, function and/or run securely.

Edit Limits... | Edit Default...

Learn more about setting these properties.

OK | Cancel | Apply

Access Permission

Default Security

Group or user names:

- NETWORK
- Administrators (DESKTOP-T2978KQ\Administrators)
- Distributed COM Users (DESKTOP-T2978KQ\Distributed C
- INTERACTIVE
- ANONYMOUS LOGON

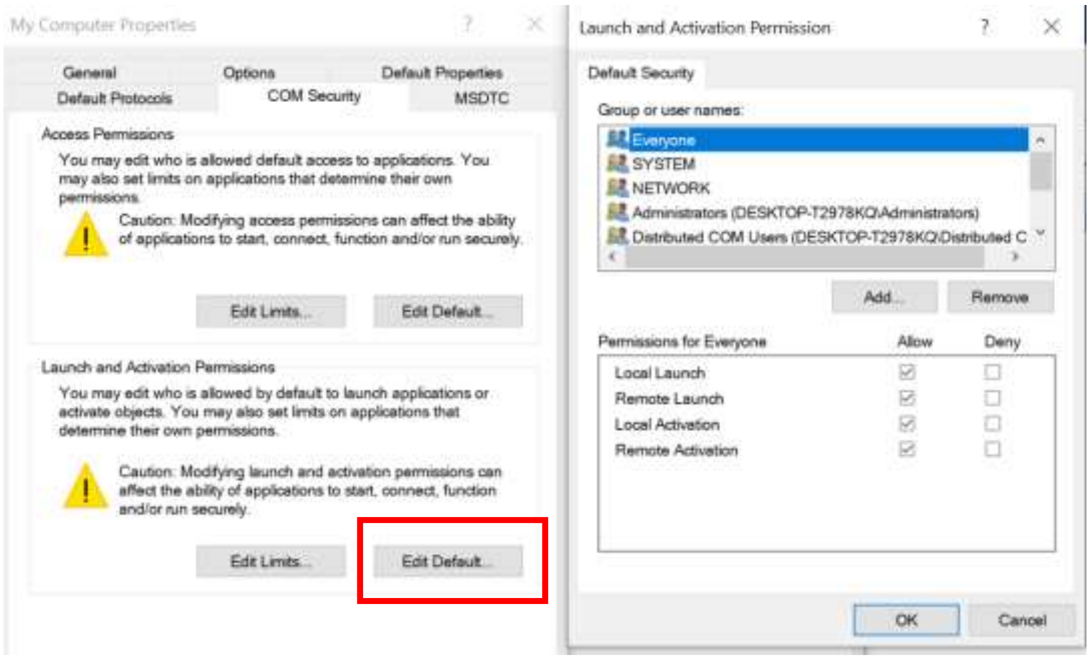
Add... | Remove

Permissions for Distributed COM Users	Allow	Deny
Local Access	✓	☐
Remote Access	✓	☐

OK | Cancel

Figure22: COM Security tab settings-Access Permissions tab

In Launch and Activation Permissions pane, Select “Edit Default” and give permissions for every user and group as shown below


My Computer Properties

General | Options | Default Properties
Default Protocols | COM Security | MSDTC

Access Permissions

You may edit who is allowed default access to applications. You may also set limits on applications that determine their own permissions.

Caution: Modifying access permissions can affect the ability of applications to start, connect, function and/or run securely.

Edit Limits... | Edit Default...

Launch and Activation Permissions

You may edit who is allowed by default to launch applications or activate objects. You may also set limits on applications that determine their own permissions.

Caution: Modifying launch and activation permissions can affect the ability of applications to start, connect, function and/or run securely.

Edit Limits... | Edit Default...

Launch and Activation Permission

Default Security

Group or user names:

- Everyone
- SYSTEM
- NETWORK
- Administrators (DESKTOP-T2978KQ\Administrators)
- Distributed COM Users (DESKTOP-T2978KQ\Distributed C

Add... | Remove

Permissions for Everyone	Allow	Deny
Local Launch	✓	
Remote Launch	✓	
Local Activation	✓	
Remote Activation	✓	

OK | Cancel

Figure23: Launch and Activation Permissions

3.5 Specifying OPC Server Properties

Initially MATRIKON OPC simulation server was used for testing and understanding purposes. So, specify the DCOM configuration for this server along with client computer and checked for communication by reading/writing data using “Tags” in Matrikon. Matrikon OPC Explorer was installed in Client system and this acts as a data sink. Matrikon OPC Server for Simulation and Testing simulates a Connection to a sample device or sample application. It transfers the simulate data that is read by Matrikon OPC Explorer.

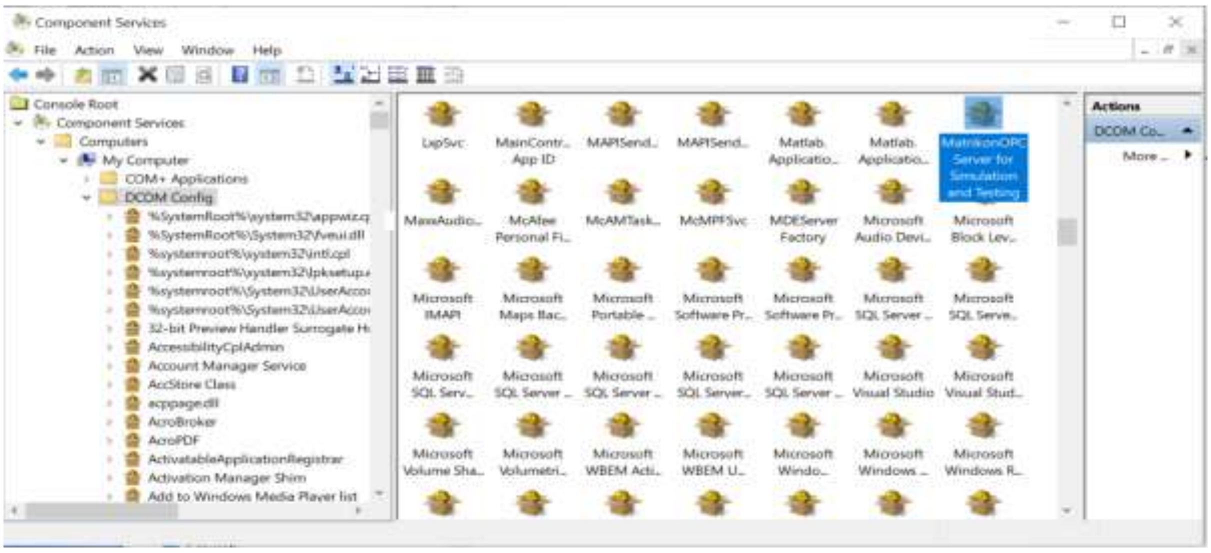
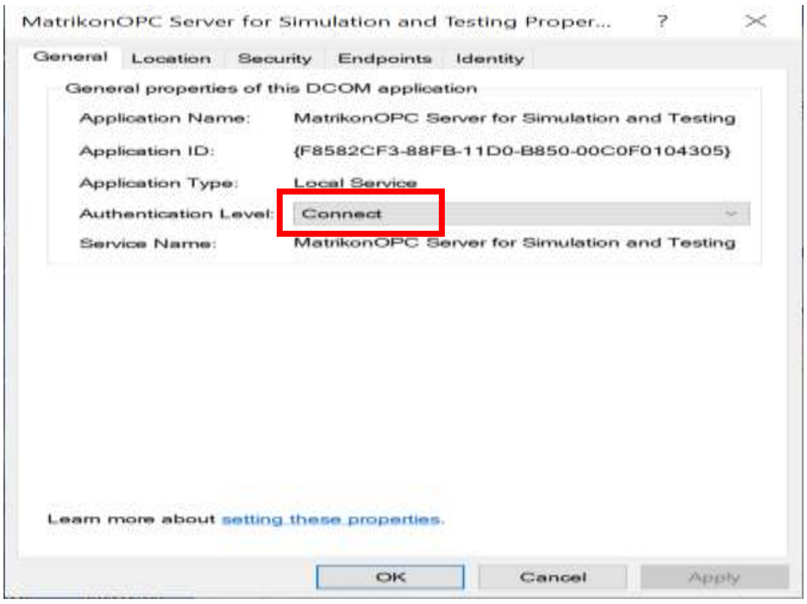


Figure24: Screenshot for DCOM configuration with Matrikon OPC server.

Right Click on the Matrikon OPC server and select Properties. The below dialog box appears.


MatrikonOPC Server for Simulation and Testing Proper...

General | Location | Security | Endpoints | Identity

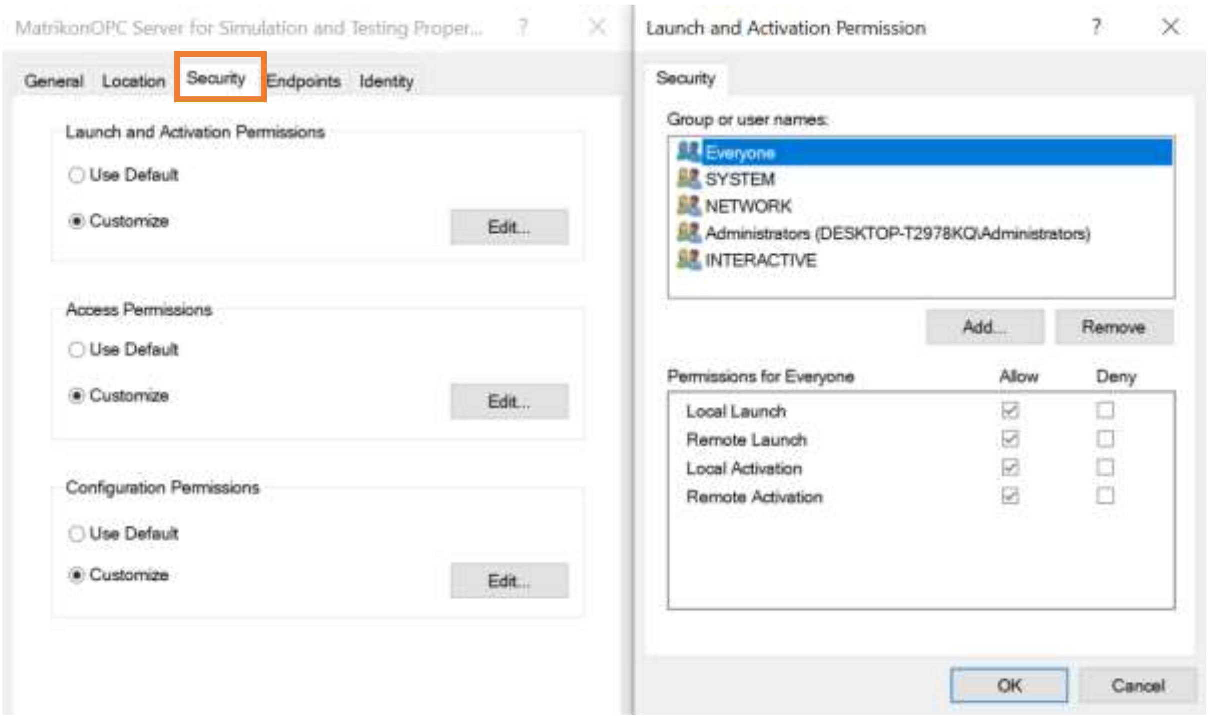
General properties of this DCOM application

Application Name: MatrikonOPC Server for Simulation and Testing
Application ID: {F8582CF3-88FB-11D0-B850-00C0F0104305}
Application Type: Local Service
Authentication Level: Connect
Service Name: MatrikonOPC Server for Simulation and Testing

Learn more about setting these properties.

OK | Cancel | Apply

Figure25: Verify the Authentication level


MatrikonOPC Server for Simulation and Testing Proper...

General | Location | Security | Endpoints | Identity

Launch and Activation Permissions
Use Default
[✓ Customize] Edit...

Access Permissions
Use Default
[✓ Customize] Edit...

Configuration Permissions
Use Default
[✓ Customize] Edit...

Launch and Activation Permission

Security

Group or user names:
Everyone
SYSTEM
NETWORK
Administrators (DESKTOP-T2978KQ\Administrators)
INTERACTIVE

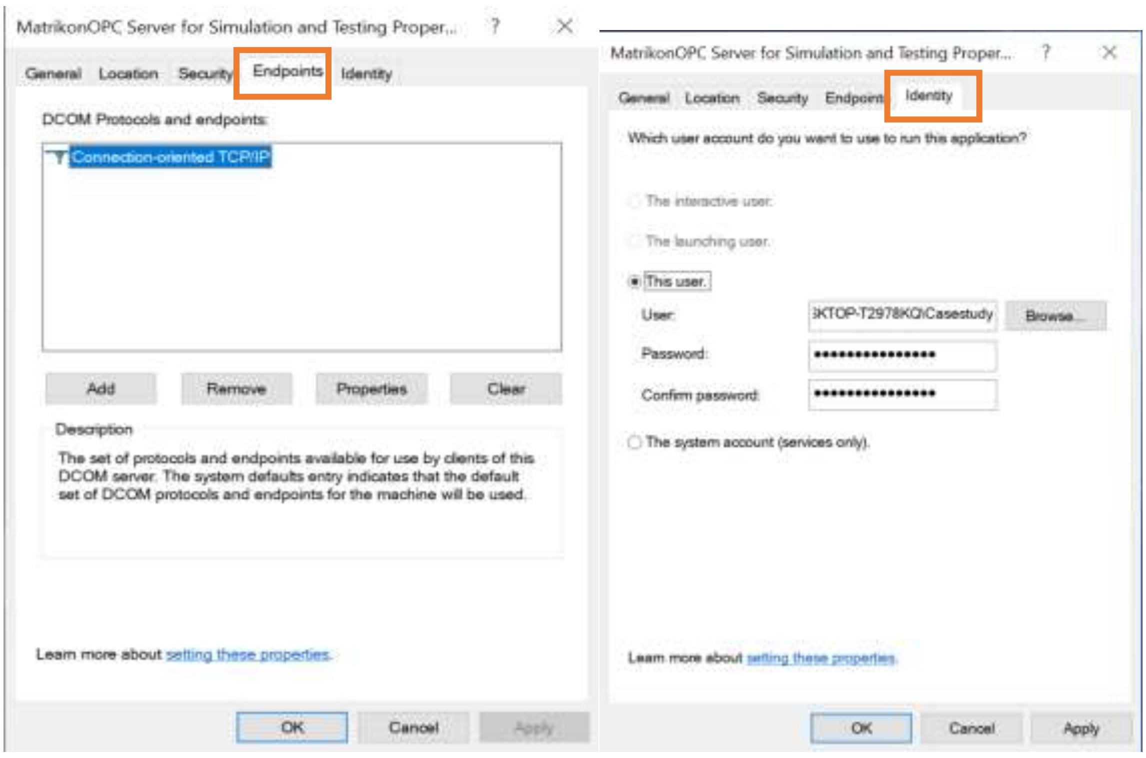
Add... | Remove

Permissions for Everyone	Allow	Deny
Local Launch	✓	☐
Remote Launch	✓	☐
Local Activation	✓	☐
Remote Activation	✓	☐

OK | Cancel

Figure26: Check all settings under 'Security' tab.

In security tab give customize option for Launch and activation , Access and configuration permissions and "Allow" permissions for everyone in group and user names. Also in Endpoints allow Connection oriented TCP/IP in DCOM Protocols.


MatrikonOPC Server for Simulation and Testing Proper...

General | Location | Security | Endpoints | Identity

DCOM Protocols and endpoints:

Connection-oriented TCP/IP

Add | Remove | Properties | Clear

Description

The set of protocols and endpoints available for use by clients of this DCOM server. The system defaults entry indicates that the default set of DCOM protocols and endpoints for the machine will be used.

Learn more about setting these properties.

OK | Cancel | Apply

MatrikonOPC Server for Simulation and Testing Proper...

General | Location | Security | Endpoints | Identity

Which user account do you want to use to run this application?

- ○ The interactive user.
- ○ The launching user.
- ● This user.
  - User: 3KTOP-T2978KQ\Casestudy | Browse...
  - Password: ••••••••••••••••
  - Confirm password: ••••••••••••••••
- ○ The system account (services only).

Learn more about setting these properties.

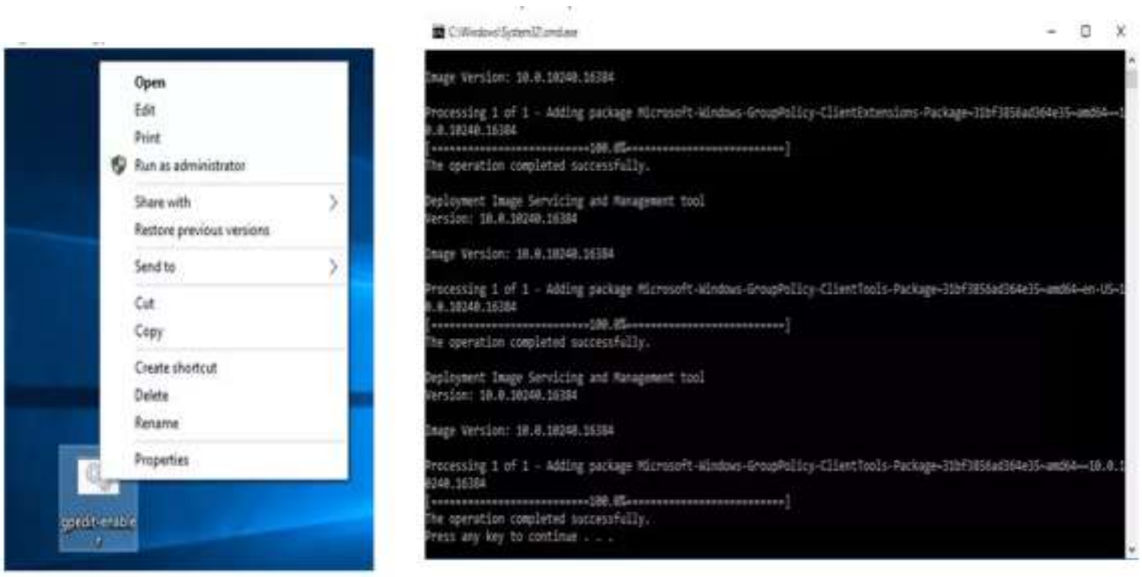
OK | Cancel | Apply

Figure27: Set username and password

As shown above you should specify the previously created user that will launch the OPC server on the "Identity" tab.

3.6 Local Security Policy:

One should configure the local security policy. This is different for Windows 10 Home and Pro edition. In our case its windows 10 Home. First, download "gpedit.enabler.bat file" from Internet. Once this file is downloaded, run this file.



C:\Windows\System32\cmd.exe

```
Image Version: 10.0.10240.16384

Processing 1 of 1 - Adding package Microsoft-Windows-GroupPolicy-ClientExtensions-Package~31bf3856ad364e35~amd64~~1
0.0.10240.16384
[==========================100.0%==========================]
The operation completed successfully.

Deployment Image Servicing and Management tool
Version: 10.0.10240.16384

Image Version: 10.0.10240.16384

Processing 1 of 1 - Adding package Microsoft-Windows-GroupPolicy-ClientTools-Package~31bf3856ad364e35~amd64~en-US~1
0.0.10240.16384
[==========================100.0%==========================]
The operation completed successfully.

Deployment Image Servicing and Management tool
Version: 10.0.10240.16384

Image Version: 10.0.10240.16384

Processing 1 of 1 - Adding package Microsoft-Windows-GroupPolicy-ClientTools-Package~31bf3856ad364e35~amd64~~10.0.1
0240.16384
[==========================100.0%==========================]
The operation completed successfully.
Press any key to continue . . .
```

Figure28: For windows 10 Home, 'gpedit' batch file is needed to configure local security policy

Once adding package group policy client tools, Run the command ***“gpedit.msc”*** Local group policy Editor will open.

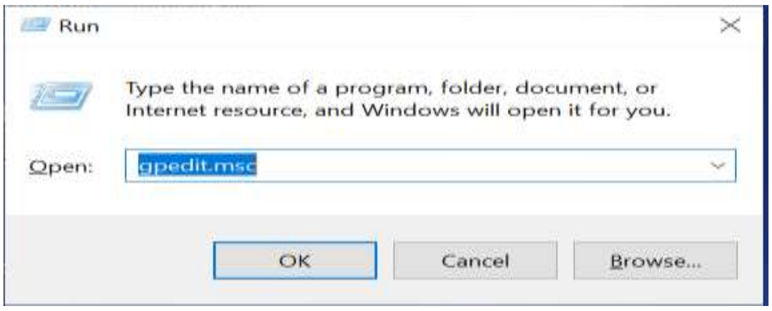
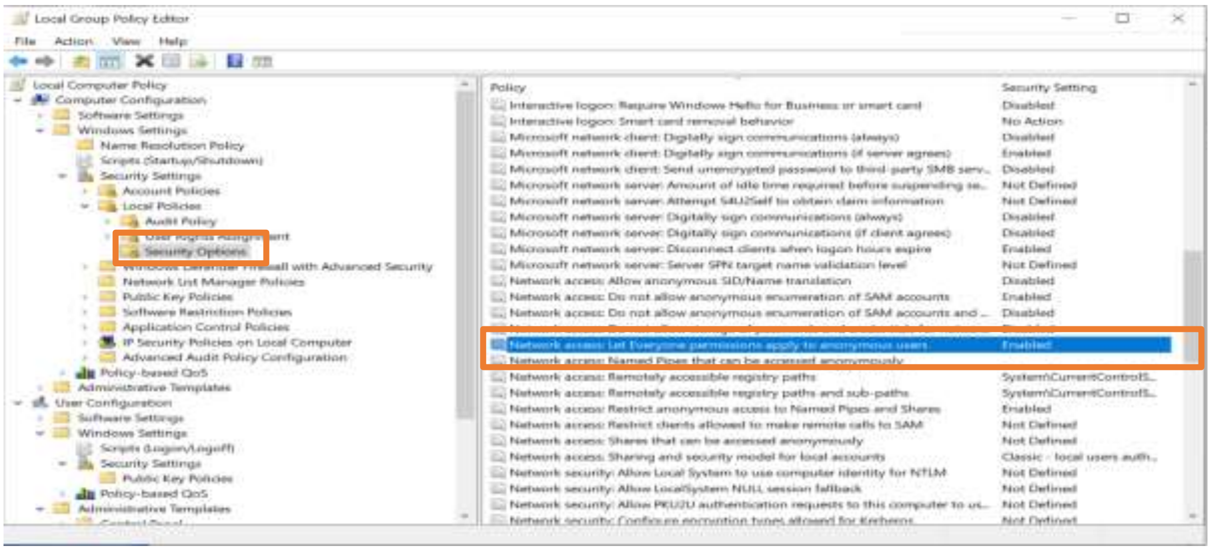

Run
Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.
Open: gpedit.msc
OK
Cancel
Browse...

Figure29: Shortcut key to local group policy editor.

Enable the following option in Local Group Policy Editor, click on computer configuration then choose windows setting and then security setting. In local policies you have multiple security options so need to Enabled policy “Network access: Let Everyone permissions apply to anonymous Users”.


Local Group Policy Editor

File Action View Help

- Local Computer Policy
  - Computer Configuration
    - Software Settings
    - Windows Settings
      - Name Resolution Policy
      - Scripts (Startup/Shutdown)
      - Security Settings
        - Account Policies
        - Local Policies
          - Audit Policy
          - User Rights Assignment
          - Security Options
        - Windows Defender Firewall with Advanced Security
        - Network List Manager Policies
        - Public Key Policies
        - Software Restriction Policies
        - Application Control Policies
        - IP Security Policies on Local Computer
        - Advanced Audit Policy Configuration
      - Policy-based QoS
    - Administrative Templates
  - User Configuration
    - Software Settings
    - Windows Settings
      - Scripts (Logon/Logoff)
      - Security Settings
        - Public Key Policies
      - Policy-based QoS
    - Administrative Templates
      - Control Panel

Policy	Security Setting
Interactive logon: Require Windows Hello for Business or smart card	Disabled
Interactive logon: Smart card removal behavior	No Action
Microsoft network client: Digitally sign communications (always)	Disabled
Microsoft network client: Digitally sign communications (if server agrees)	Enabled
Microsoft network client: Send unencrypted password to third-party SMB serv...	Disabled
Microsoft network server: Amount of idle time required before suspending se...	Not Defined
Microsoft network server: Attempt S4U2Self to obtain claim information	Not Defined
Microsoft network server: Digitally sign communications (always)	Disabled
Microsoft network server: Digitally sign communications (if client agrees)	Disabled
Microsoft network server: Disconnect clients when logon hours expire	Enabled
Microsoft network server: Server SPN target name validation level	Not Defined
Network access: Allow anonymous SID/Name translation	Disabled
Network access: Do not allow anonymous enumeration of SAM accounts	Enabled
Network access: Do not allow anonymous enumeration of SAM accounts and ...	Disabled
[illegible]	[illegible]
Network access: Let Everyone permissions apply to anonymous users	Enabled
Network access: Named Pipes that can be accessed anonymously	
Network access: Remotely accessible registry paths	System\CurrentControlS...
Network access: Remotely accessible registry paths and sub-paths	System\CurrentControlS...
Network access: Restrict anonymous access to Named Pipes and Shares	Enabled
Network access: Restrict clients allowed to make remote calls to SAM	Not Defined
Network access: Shares that can be accessed anonymously	Not Defined
Network access: Sharing and security model for local accounts	Classic - local users auth...
Network security: Allow Local System to use computer identity for NTLM	Not Defined
Network security: Allow LocalSystem NULL session fallback	Not Defined
Network security: Allow PKU2U authentication requests to this computer to us...	Not Defined
Network security: Configure encryption types allowed for Kerberos	Not Defined

Figure30: Check if the policy is enabled

3.7 Matrikon OPC Client and Server for Simulation and Testing:

Now in Client Computer Open “Matrikon OPC Explorer” and click on Network Neighbourhood. You will see available server list. Choose the Matrikon Simulation Server from the list and give “**CONNECT**”

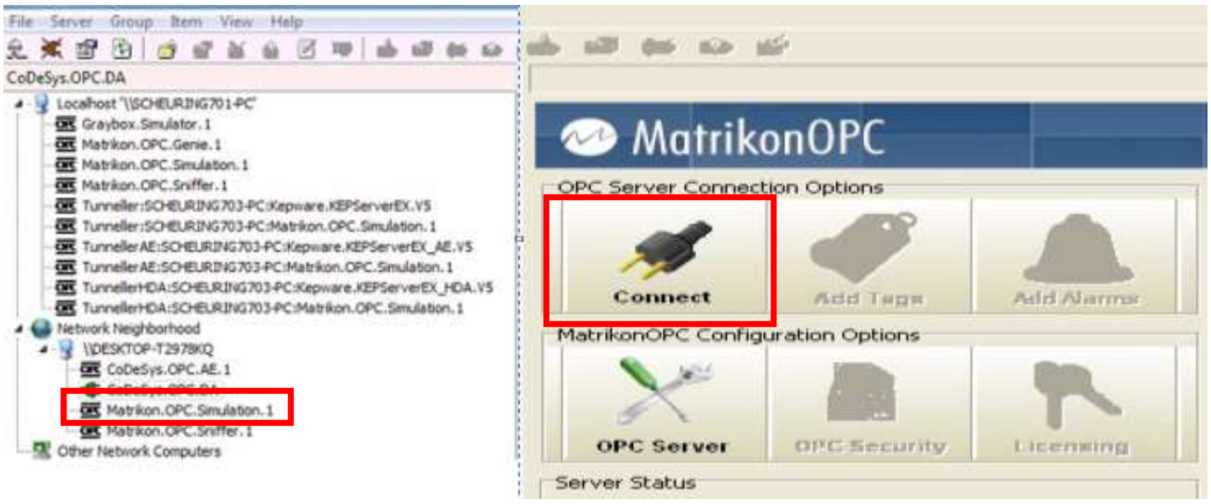

File Server Group Item View Help
CoDeSys.OPC.DA
Localhost '\\SCHEURING701-PC'
Graybox.Simulator.1
Matrikon.OPC.Genie.1
Matrikon.OPC.Simulation.1
Matrikon.OPC.Sniffer.1
Tunneller:SCHEURING703-PC:Kepware.KEPServerEX.V5
Tunneller:SCHEURING703-PC:Matrikon.OPC.Simulation.1
TunnellerAE:SCHEURING703-PC:Kepware.KEPServerEX_AE.V5
TunnellerAE:SCHEURING703-PC:Matrikon.OPC.Simulation.1
TunnellerHDA:SCHEURING703-PC:Kepware.KEPServerEX_HDA.V5
TunnellerHDA:SCHEURING703-PC:Matrikon.OPC.Simulation.1
Network Neighborhood
\\DESKTOP-T2978KQ
CoDeSys.OPC.AE.1
CoDeSys.OPC.DA
Matrikon.OPC.Simulation.1
Matrikon.OPC.Sniffer.1
Other Network Computers
MatrikonOPC
OPC Server Connection Options
Connect
Add Tags
Add Alarms
MatrikonOPC Configuration Options
OPC Server
OPC Security
Licensing
Server Status

Figure31: Matrikon OPC Explorer acts as a client

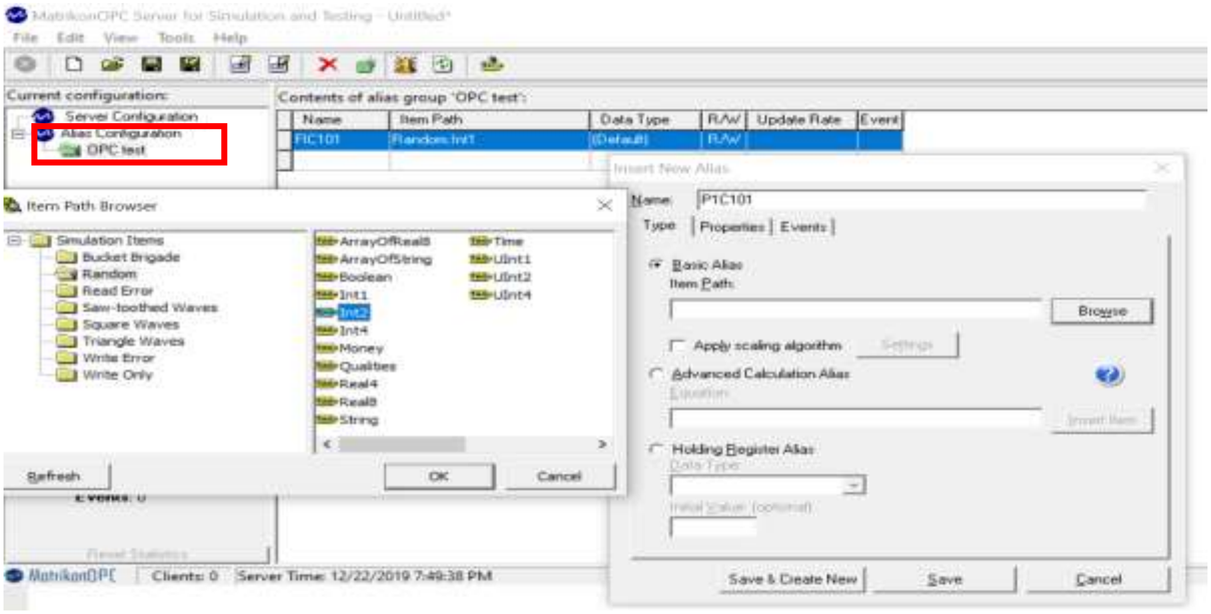


Figure32: This dialog box appears after connection and clicking on

'Add tags' and reading/writing variables in server

Create a new Alias and add the variables that you want to Read/Write to the client and from the client. Save them. After Connection you can "Add tags" and read/write variables from the server to the client.

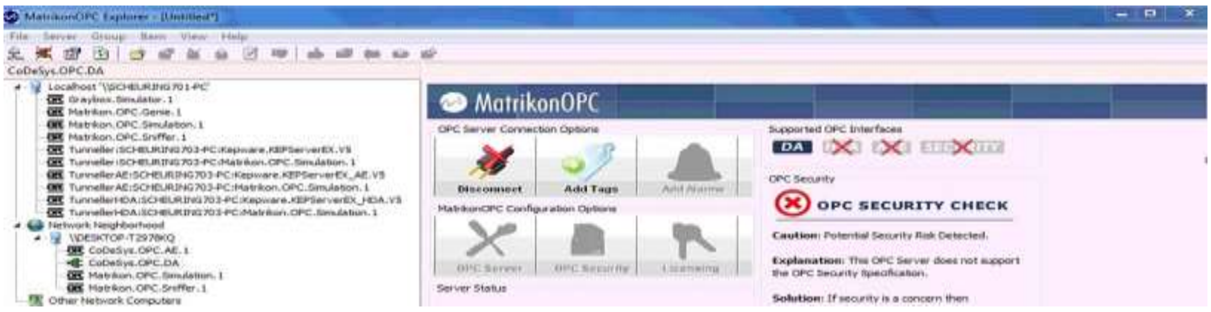
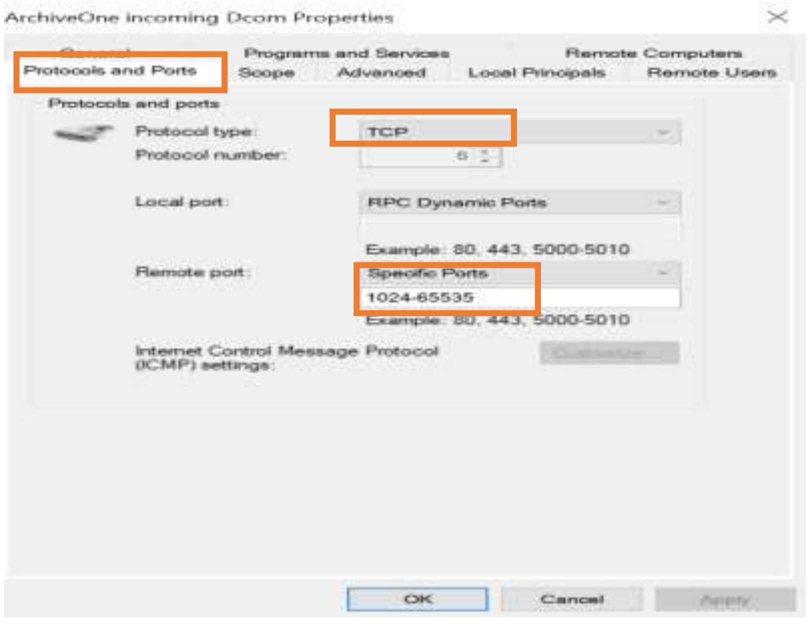

MatrikonOPC Explorer - [Untitled*]
File Server Group Item View Help
CoDeSys.OPC.DA
MatrikonOPC
OPC Server Connection Options
Disconnect
Add Tags
Add Alarms
MatrikonOPC Configuration Options
OPC Server
OPC Security
Licensing
Server Status
Supported OPC Interfaces
DA
OPC Security
OPC SECURITY CHECK
Caution: Potential Security Risk Detected.
Explanation: This OPC Server does not support the OPC Security Specification.
Solution: If security is a concern then

Figure33: Once adding tags and checking the communication

This is an example of connecting to the server along with DCOM settings and then reading/writing data. Now our DCOM setup has been established and the OPC server is able to communicate with client.

CHALLENGES FACED FOR CONNECTION OF OPC SERVER

- *RPC Server Unavailable – This error was occurring when trying to connect to OPC server from Client.*
- *Description: This error means it is impossible to establish a network connection to the RPC service.*
    - *If the error occurs during an attempt to get the list of OPC servers on the remote computer, you should check the firewall settings. The OpcEnum.exe file located in the Windows\System32 folder by default must be added to exceptions.*
    - *You should make sure that inbound DCOM connections to port 135 are permitted*
    - *If the error occurs during an attempt to connect to the OPC server, you should make sure that the executable file of the OPC server is added to the Windows firewall exception list.*
- *Troubleshooted this error as follows by Configuring RPC rule in new inbound rule of Windows Firewall console. Give the following settings:*



ArchiveOne incoming Dcom Properties

General | Programs and Services | Remote Computers
Protocols and Ports | Scope | Advanced | Local Principals | Remote Users

Protocols and ports

Protocol type: TCP
Protocol number: 6
Local port: RPC Dynamic Ports
Example: 80, 443, 5000-5010
Remote port: Specific Ports
1024-65535
Example: 80, 443, 5000-5010
Internet Control Message Protocol (ICMP) settings: Customize...

OK | Cancel | Apply

Figure33: Once adding tags and checking the communication

- *On the "Scope" Page, Select "Any IP Addresses to allow all remote connections, click Next.*
  Select "Allow the connection" in Action tab.
  Select only the "Domain" option in Profile tab.
  In Name page, Enter the rule name "ArchiveOne Incoming DCOM Properties" Click Finish.
  Verify the Rule is enabled. Now connection was established between client and server

SECTION 4

CODESYS V3.5 SOFTWARE

CODESYS is the leading manufacturer-independent IEC 61131-3 automation software for engineering control systems**.**

CODESYS is developed and marketed by the German software company 3S-Smart Software Solutions located in the Bavarian town of Kempten. Version 3.05 was released in 2019. The software tool covers different aspects of industrial automation technology with one surface.

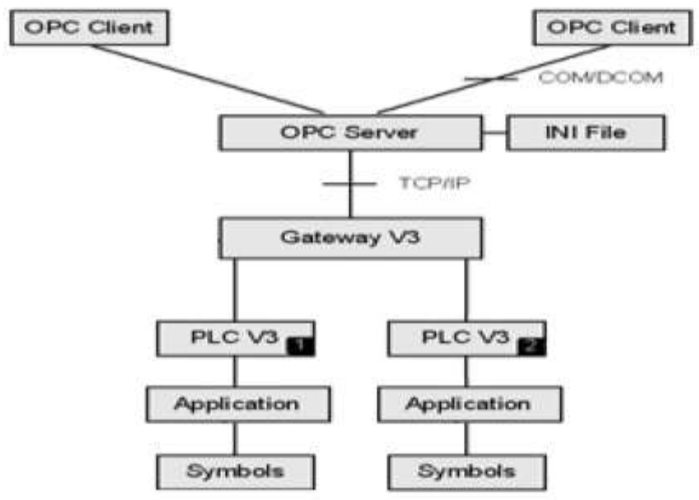


Figure34: CODESYS System Architecture

In our case studies, the following software were used in Codesys

1. Codesys Development system V3.5 SP15 – Used for Programming PLC
2. Codesys Gateway V3
3. Codesys Control Win x64 – Virtual PLC
4. Codesys OPC DA server.

4.1CODESYS Development System

CODESYS is a development system from 3S-Smart Software Solutions. It is used by numerous manufacturers in the automation sector to program and parameterize a wide variety of automation components. All five programming languages for application programming defined in the IEC 61131-3 are available in the CODESYS development environment.

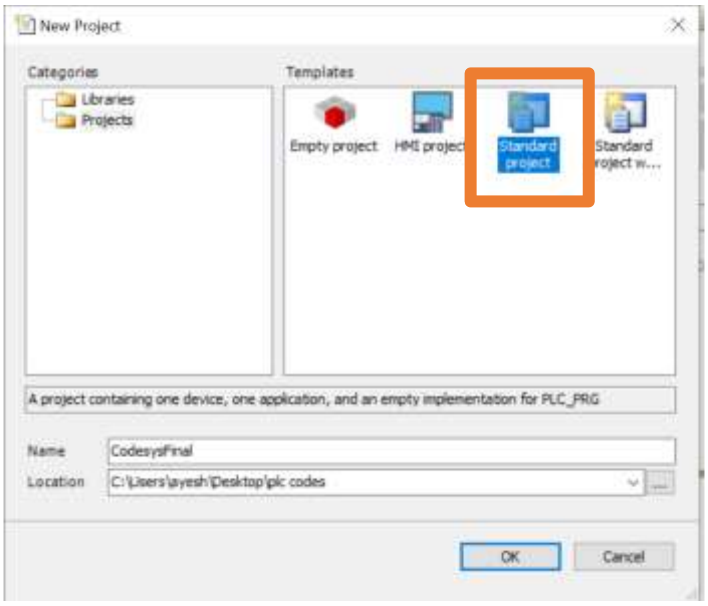
- IL (instruction list) is an assembler like programming language (Is now deprecated but available for backward compatibility)
- ST (structured text) is similar to programming in Pascal or C

- LD (ladder diagram) enables the programmer to virtually combine relay contacts and coils
- FBD (function block diagram) enables the user to rapidly program both Boolean and analogue expressions
- SFC (sequential function chart) is convenient for programming sequential processes and flows

NOTE: In this case studies, Continuous Function Charts (CFC) are used to program the PLC. Continuous Function Charts. It means all function are elaborate every scan of plc and output will be energized always.

4.1.1 Steps to create a new project using CFC.

When click Codesys Development system V3.5 SP-15, home page open and in basic operations you can create a New project, in Templates choose Standard project and give name and saved project. After that choose programmable Device "CODESYS Control Win V3 x 64" and specify language for PLC program i.e. Continuous Function Chart(CFC).

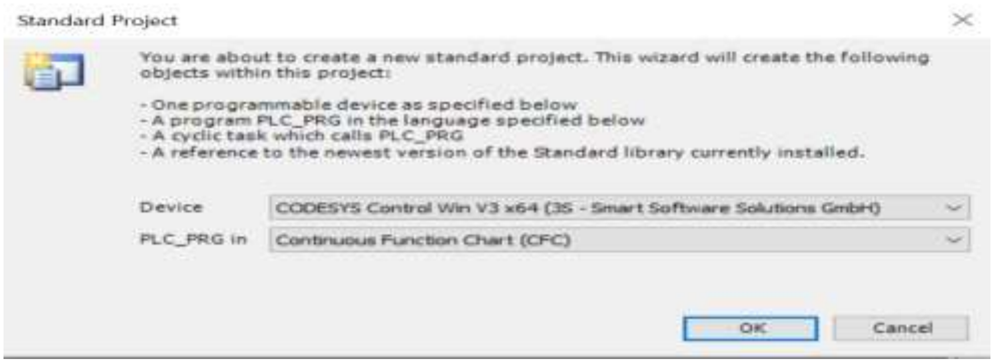


A project containing one device, one application, and an empty implementation for PLC_PRG

Name: CodesysFinal

Location: C:\Users\ayesh\Desktop\plc codes

Figure35: Project creation



You are about to create a new standard project. This wizard will create the following objects within this project:

- One programmable device as specified below
- A program PLC_PRG in the language specified below
- A cyclic task which calls PLC_PRG
- A reference to the newest version of the Standard library currently installed.

Device: CODESYS Control Win V3 x64 (3S - Smart Software Solutions GmbH)

PLC_PRG in: Continuous Function Chart (CFC)

Figure36: Our Project PLC Program is in CFC

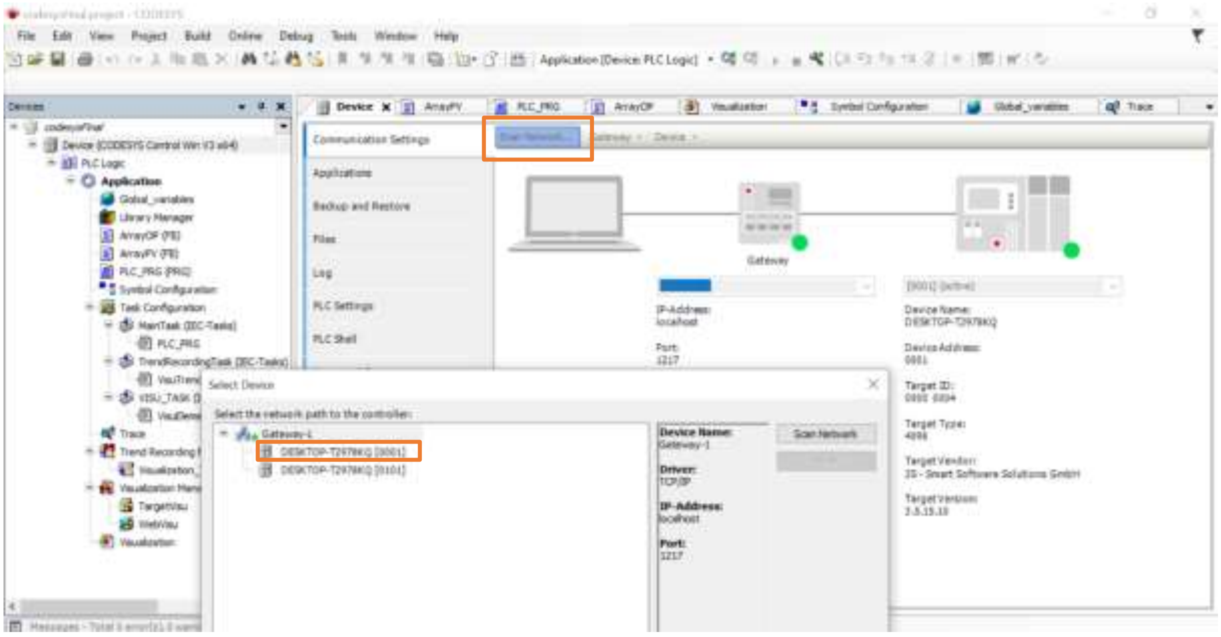
4.1.2 Connection of Devices

Since Our Virtual PLC is embedded in the software itself so its connected within the localhost through gateway. To connect it with local host:

-Click on the Gateway

-Scan Network

-Select the Device (the name of the Device, in our case the name of our Server machine is DESKTOP-T2978KQ)



IP-Address:
localhost

Port:
1217

[0001] (active)

Device Name:
DESKTOP-T2978KQ

Device Address:
0001

Target ID:
0000 0004

Target Type:
4096

Target Vendor:
3S - Smart Software Solutions GmbH

Target Version:
3.5.15.10

Device Name:
Gateway-1

Driver:
TCP/IP

IP-Address:
localhost

Port:
1217

Figure37: Configuration Settings of PLC and CODESYS server

4.2 Gateway

CODESYS Gateway acts as a bridge between the Development system to load the program and send it to the controller (PLC). The communication between the OPC server and a CODESYS V3 controller is always done via CODESYS Gateway V3.

Before Scanning the Network make sure that the PLC and Gateway are running, this can be verified from the Task Bar.

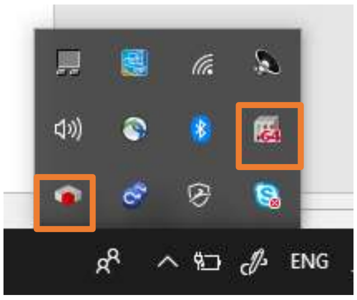
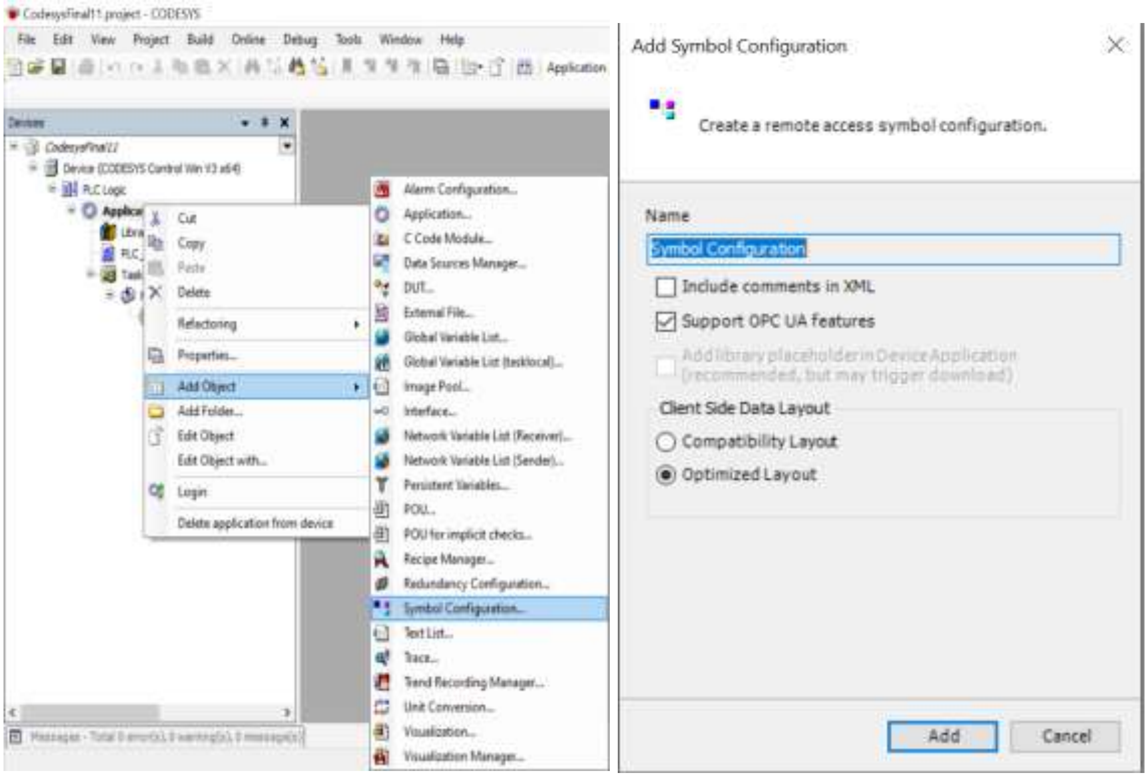


Figure38: Connection of PLC using Gateway

4.2.1 Adding Symbol Configuration:

To begin with the PLC programming, first add the symbol configuration in the PLC logic. Symbol Configuration is used for preparing symbols with specific access rights for project variables. Through these symbols you can access the variables from outside, for example from an OPC server. When generating code, CODESYS also generates a symbol file (*. xml) that includes the description of the symbols.


Add Symbol Configuration

Create a remote access symbol configuration.

Name: Symbol Configuration

☐ Include comments in XML

☑ Support OPC UA features

☐ Add library placeholder in Device Application (recommended, but may trigger download)

Client Side Data Layout

○ Compatibility Layout

● Optimized Layout

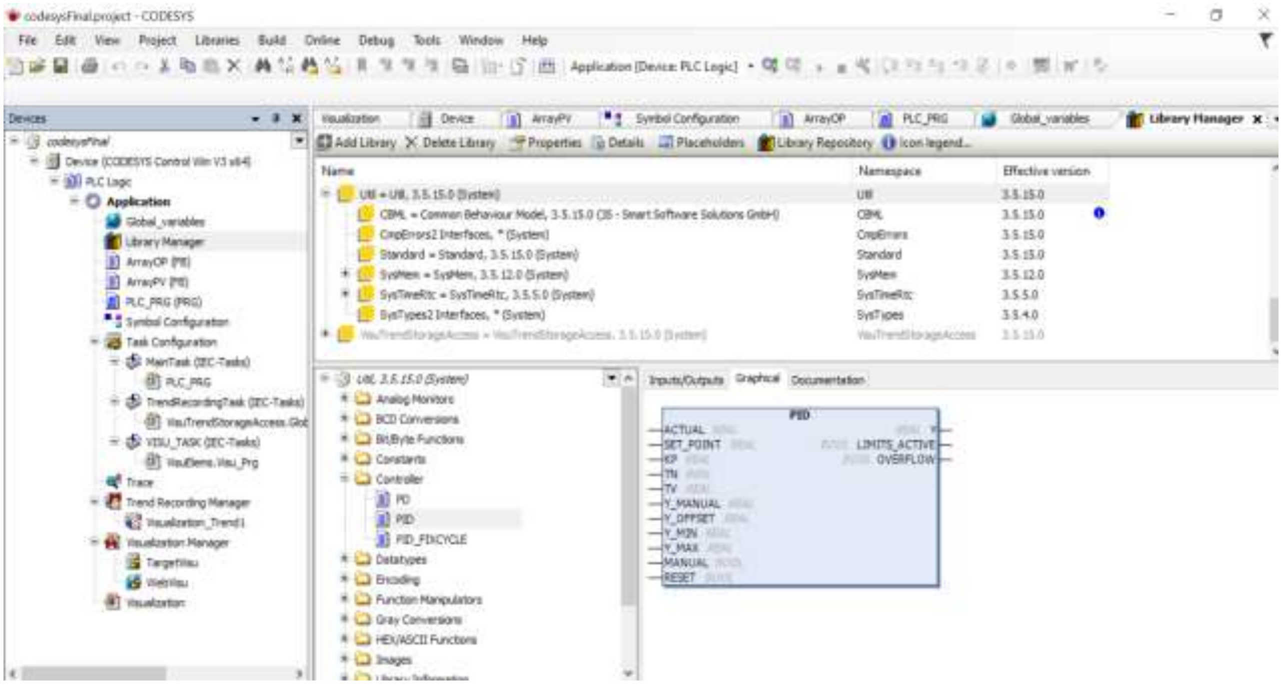
Figure39: Symbol Configuration

4.3 PLC Programming:

The Continuous Function Chart (CFC) implementation language is a graphical programming language which extends the standard languages of IEC 61131-3. This programming language graphically program a system by means of a POU in CFC. Elements can be inserted and positioned freely.

To add the PID controller, the “Util” library is added in the library manager. The following steps explains how to get util library in the library manager

-Select Library Manager
-Click on Edit object and the library manager open the Editor
-Click on Add library
-Browse the util library
-Select the util library and give Add.
-The util library is added in the library Manager
Once the “util” library is installed, Select controller and open PID function block.



Name	Namespace	Effective version
Util = Util, 3.5.15.0 (System)	Util	3.5.15.0
CBML = Common Behaviour Model, 3.5.15.0 (3S - Smart Software Solutions GmbH)	CBML	3.5.15.0
CmpErrors2 Interfaces, * (System)	CmpErrors	3.5.15.0
Standard = Standard, 3.5.15.0 (System)	Standard	3.5.15.0
SysMem = SysMem, 3.5.12.0 (System)	SysMem	3.5.12.0
SysTimeRtc = SysTimeRtc, 3.5.5.0 (System)	SysTimeRtc	3.5.5.0
SysTypes2 Interfaces, * (System)	SysTypes	3.5.4.0
VisuTrendStorageAccess = VisuTrendStorageAccess, 3.5.15.0 (System)	VisuTrendStorageAccess	3.5.15.0

Figure40: PID CONTROLLER FUNCTION BLOCK

4.3.1 SCALE_R Function Description:

In this PID programming, used the "SCALE_R" function to scale our input. For this purpose, added "OSCAT" basic 333 library from CODESYS store. The description about SCALE_R is given below:

Type Function: REAL
Input X: REAL (input)
I_LO: REAL (min input value)
I_HI: REAL (input value max)
O_LO: REAL (min output value)
O_HI: REAL (output value max)
Output REAL (output value)

SCALE_R scales an input value REAL and calculates an output value in REAL. The input value X is limited here to I_LO and I_HI. SCALE_R can also be negative output values and work with a negative slope, but the values I_LO and I_HI must always be I_ LO < I_HI.

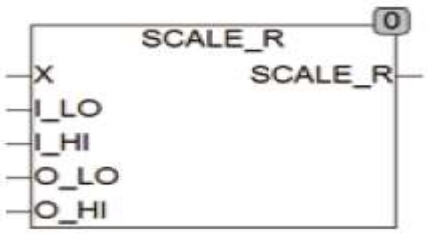


Figure41: Scale_R Function

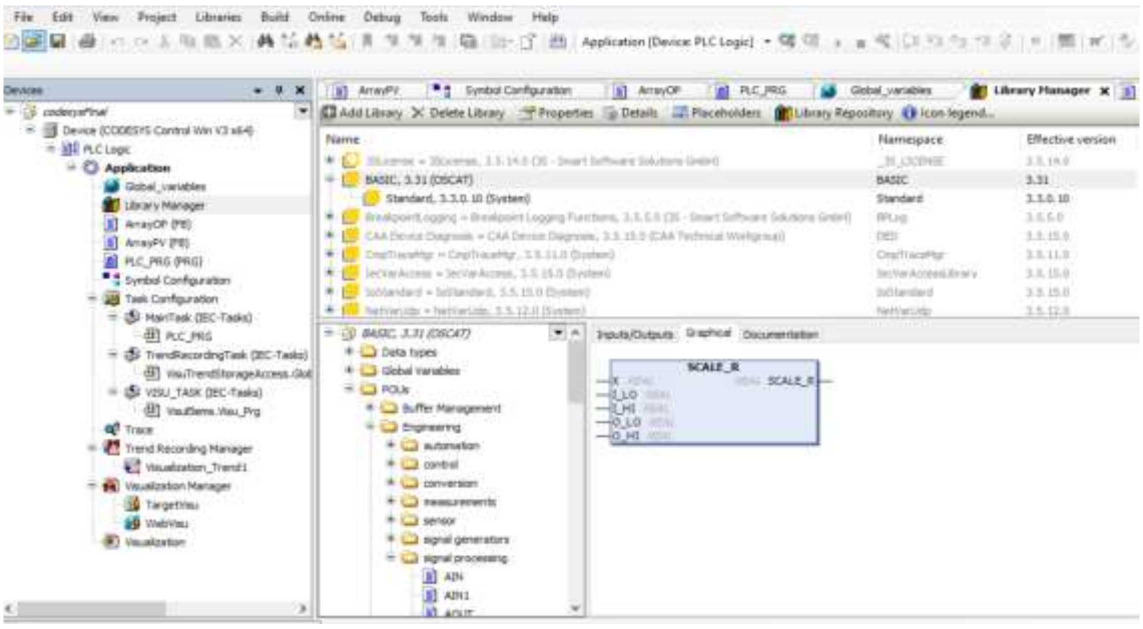
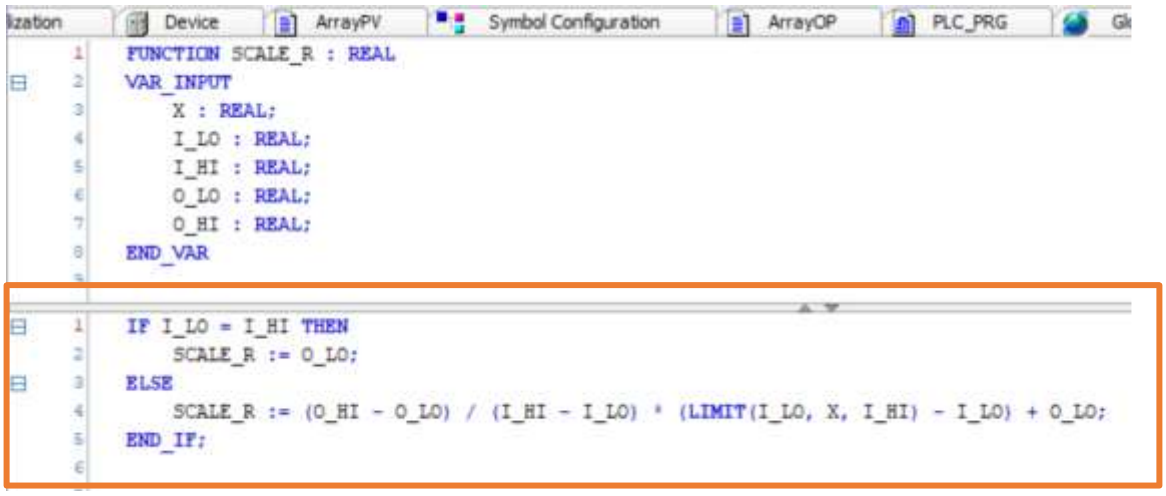


Figure42: OSCAT Library for Scale_R function has to be downloaded from Codesys store and added here in Library Manager.



```
FUNCTION SCALE_R : REAL
VAR_INPUT
    X : REAL;
    I_LO : REAL;
    I_HI : REAL;
    O_LO : REAL;
    O_HI : REAL;
END_VAR

IF I_LO = I_HI THEN
    SCALE_R := O_LO;
ELSE
    SCALE_R := (O_HI - O_LO) / (I_HI - I_LO) * (LIMIT(I_LO, X, I_HI) - I_LO) + O_LO;
END_IF;
```

Figure43: Program using SCALE_R function to scale the inputs and outputs of PID.

4.3.2 Significance of Scale_R function Block in our program:

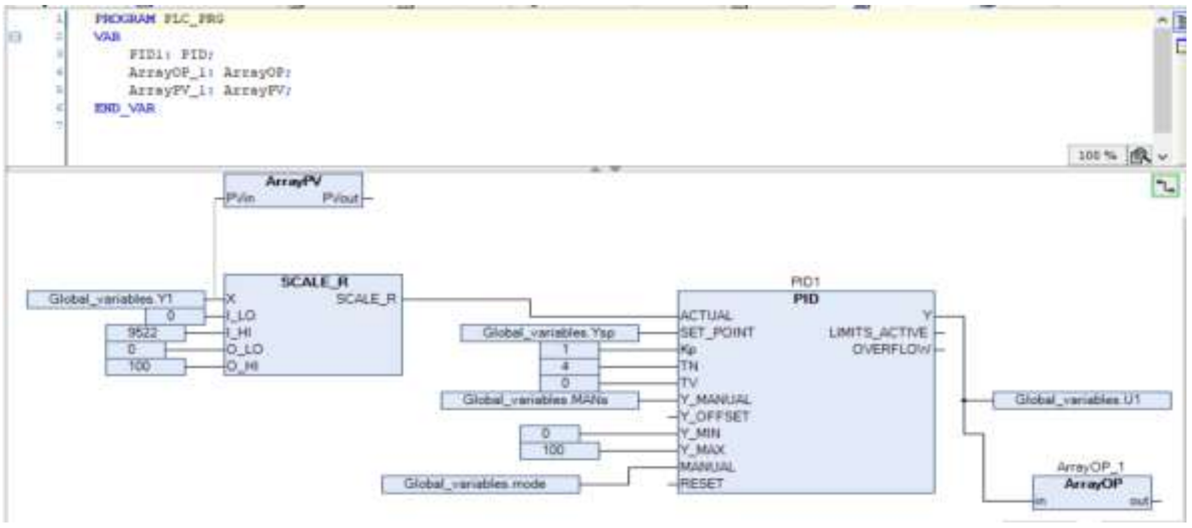
While performing Simulation, it may be necessary to adjust the controller's sample rate for optimum performance. Changing the sample rate will directly affect the closed loop response of the PID control algorithm. To maintain stability and performance, the PID parameters must be scaled to match the change in sample rate. According to this PLC program, the input has been scaled to the PID Controller in the range between 1-100.

4.3.3 PID Controller Programming:

In this PID program, CFC was used that consists of one function block PID Controller and SCALE_R function. PID Controller were made to operate in Auto and Manual Mode. The parameters of this controller were set to:

PARAMETERS	DESCRIPTION	DATA TYPE AND VALUES
ACTUAL	Process Variable(PV)	Real
SET_POINT	Desired Value	Real
KP	Proportionality constant	1
TN	Integral Time	4
TV	Derivative time	0
MANUAL	Mode(True – Man , False – Auto)	Boolean
Y	Manipulated variable, OP of the controller	Real
Y_MANUAL	Set the values – Man Mode	Real

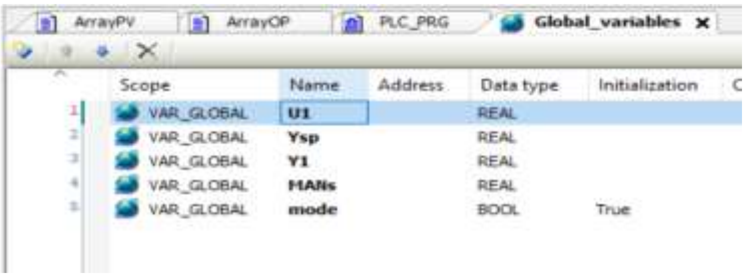
Table3: PID CONTROLLER Function Block Parameters



```
PROGRAM PLC_PRG
VAR
    PID1: PID;
    ArrayOP_1: ArrayOP;
    ArrayPV_1: ArrayPV;
END_VAR
```

Figure44: PID Program screenshot

U1 is the output of the controller, which is taken as OP and Y1 provides the input to the controller which is th PV. The significance of OP and PV can be explained in the next chapter with application of Unisim and setting up OPC DA server between CodeSys and Unisim.



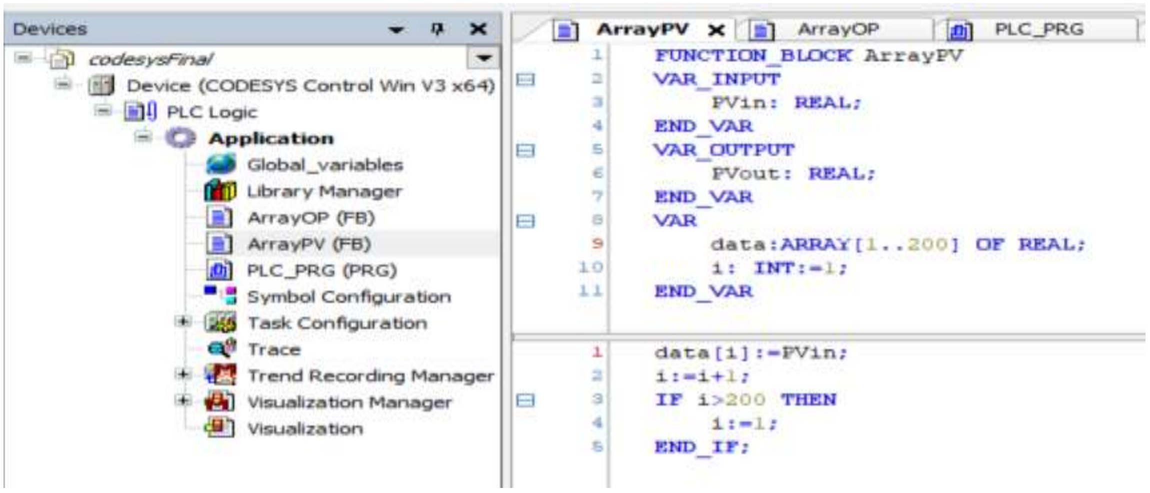
Scope	Name	Address	Data type	Initialization
VAR_GLOBAL	U1		REAL	
VAR_GLOBAL	Ysp		REAL	
VAR_GLOBAL	Y1		REAL	
VAR_GLOBAL	MANs		REAL	
VAR_GLOBAL	mode		BOOL	True

Figure45: Global Variables declaration of PID controller

4.3.4 Declaring Arrays:

Two array blocks were used for storing values of OP and PV for our PID controller. This was an alternate way to calculate the delays in updating the OP and PV to client (Unisim) from server (OPC DA – CODESYS). Here one can find the difference in Index values of the OP and PV and multiply with cyclic time to calculate the delay.

The Array Element of PV stores the First 200 values of the input to the PID controller that is generated as feedback from Unisim(Client).

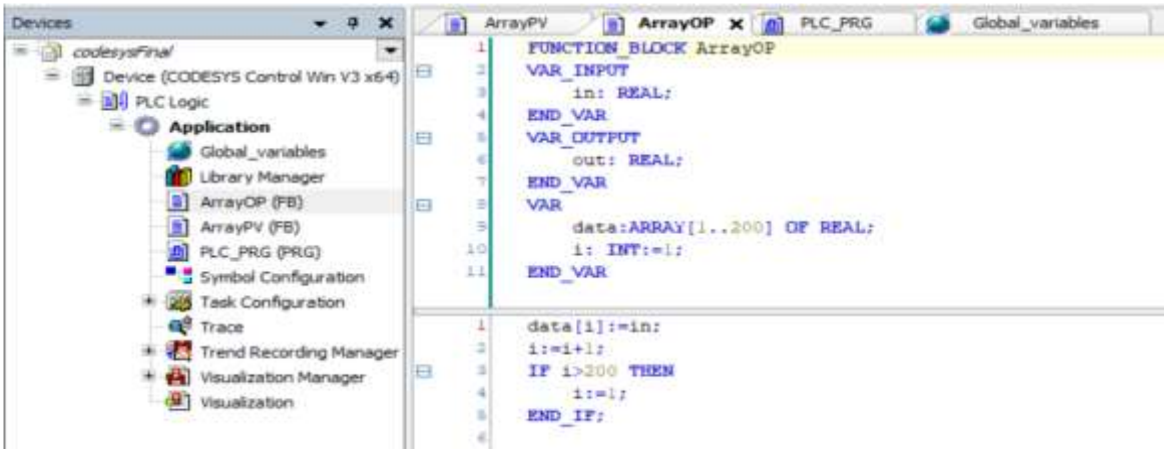


```
FUNCTION_BLOCK ArrayPV
VAR_INPUT
    PVin: REAL;
END_VAR
VAR_OUTPUT
    PVout: REAL;
END_VAR
VAR
    data:ARRAY[1..200] OF REAL;
    i: INT:=1;
END_VAR
```

```
data[i]:=PVin;
i:=i+1;
IF i>200 THEN
    i:=1;
END_IF;
```

Figure46: Array declaration for PV

The Array Element of OP stores the First 200 values of the input to the PID controller that is taken from PID controller and fed it to Unisim(Client).

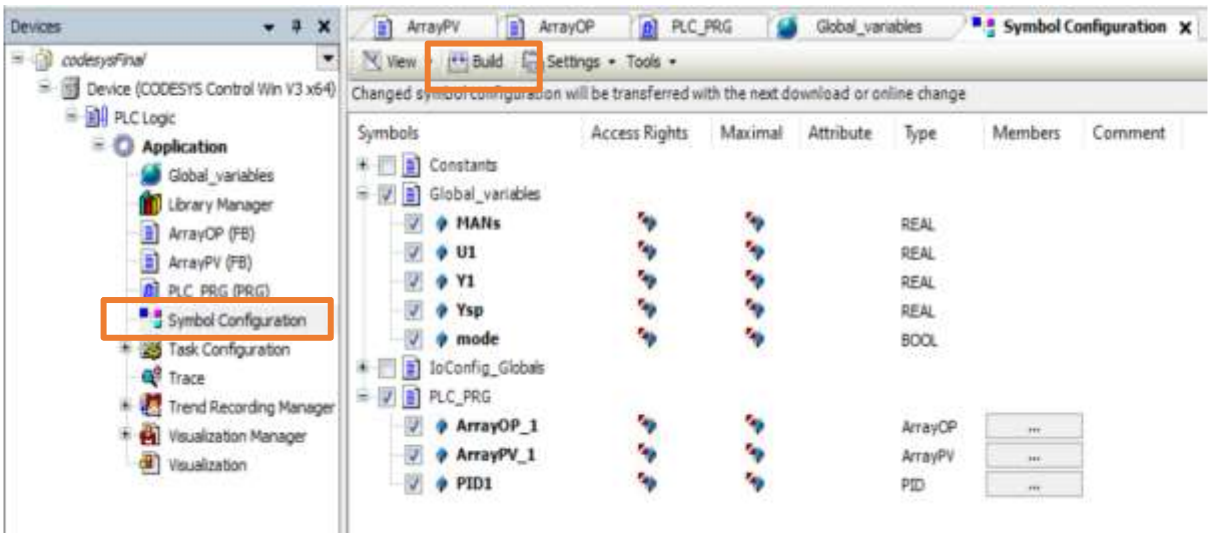


```
FUNCTION_BLOCK ArrayOP
VAR_INPUT
    in: REAL;
END_VAR
VAR_OUTPUT
    out: REAL;
END_VAR
VAR
    data:ARRAY[1..200] OF REAL;
    i: INT:=1;
END_VAR
```

```
data[i]:=in;
i:=i+1;
IF i>200 THEN
    i:=1;
END_IF;
```

Figure47: Array declaration for OP

Finally, all the variables added in the PLC program is displayed in the Symbol configurator automatically and have to build all these variable to be able to read and write data inside it from the external source after setting up OPC DA server.



Changed symbol configuration will be transferred with the next download or online change

Symbols	Access Rights	Maximal	Attribute	Type	Members	Comment
Constants						
Global_variables						
MANs				REAL		
U1				REAL		
Y1				REAL		
Ysp				REAL		
mode				BOOL		
IoConfig_Globals						
PLC_PRG						
ArrayOP_1				ArrayOP	...	
ArrayPV_1				ArrayPV	...	
PID1				PID	...	

Figure48: Build project after completion of program.

After Building the variables, Go to Login – Download the Project to PLC. And then the program will run and you can visulaize the results in "Visulaization Window" which is a inbuilt feature in Codesys.

4.4 CODESYS OPC DA Server

To receive a communication from the PLC via OPC server to the OPC client, the following steps are necessary:

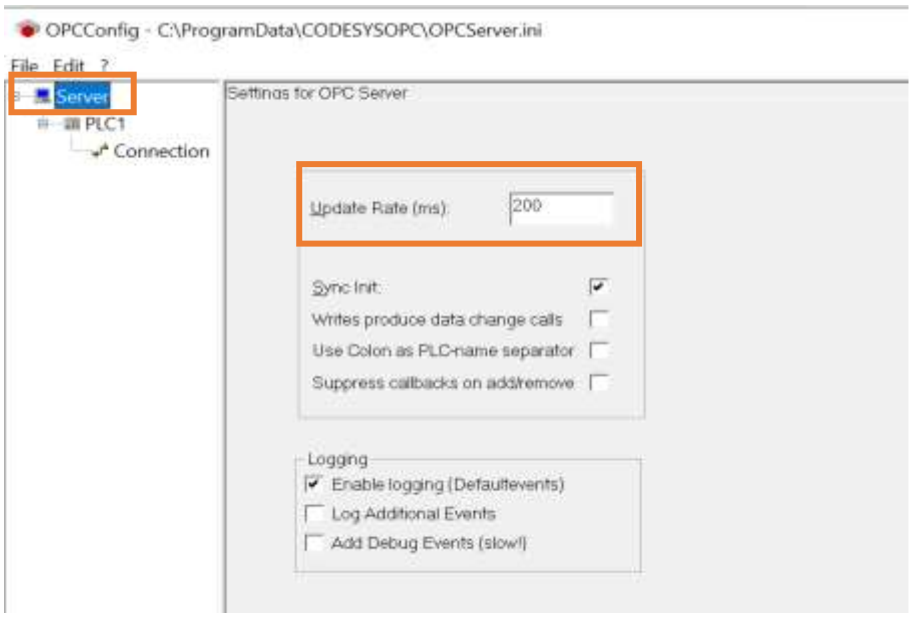
1. The application on the control contains symbols (Items) configured with symbol configuration

2. The OPC server receives all necessary settings to establish the connections in "**OPCServer.ini**". The name and path of the OPC server INI file is by default "OPCServer.ini" in the installation directory.

3. After selection of server CODESYS.OPC.DA the desired OPC-Client can browse all configured symbols.

The OPC Server will be started automatically by the operating system as soon as a client establishes a connection. The OPC Server then will terminate automatically as soon as all clients have closed their connections to the server.

Downloaded a demo Version of CODESYS OPC server to get the understanding of the software. This trial version lasts one month after that one has to purchase it from the CODESYS server site. Then, built the PID Program and checked with client-server configuration. The setup of OPC server is done using OPC Configurator and 200ms is the default update rate of the Server. This is the cycle time according to which all item data are read from the PLC. The data then gets written into the server with which the client communicates with a separately defined update rate.



OPCConfig - C:\ProgramData\CODESYSOPC\OPCServer.ini

File Edit ?

Server
PLC1
Connection

Settings for OPC Server

Update Rate (ms): 200

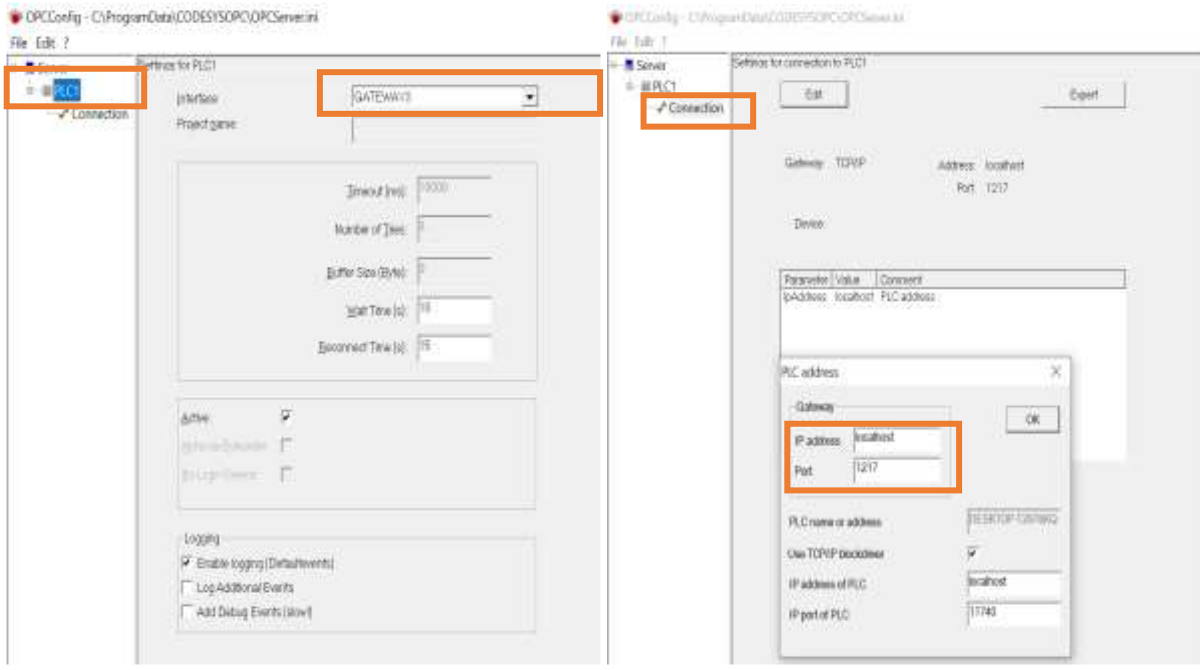
Sync Init: ✓
Writes produce data change calls: ☐
Use Colon as PLC-name separator: ☐
Suppress callbacks on add/remove: ☐

Logging
✓ Enable logging (Defaultevents)
☐ Log Additional Events
☐ Add Debug Events (slow!)

Figure49: Server update rate

Next Step is to right click in the Server tab and select option "Append PLC" to add a PLC to the DA Server. The below settings were made to setup the server and PLC. Gateway3 is used here as interface between the Server and the PLC.

Finally establish the connection by clicking on the Connection inside the PLC tab, select "Edit" and Insert the IP address of the Gateway(localhost) and Device name of the PLC (DESKTOP-T2978KQ)



OPCConfig - C:\ProgramData\CODESYSOPC\OPCServer.ini

File Edit ?

Server
PLC1
Connection

Settings for PLC1

Interface: GATEWAY3
Project name:

Timeout (ms): 10000
Number of Tries: 3
Buffer Size (Byte): 0
Wait Time (s): 10
Reconnect Time (s): 15

Active: ✓
☐
☐

Logging
✓ Enable logging (Defaultevents)
☐ Log Additional Events
☐ Add Debug Events (slow!)

Settings for connection to PLC1

Edit   Export

Gateway: TCP/IP   Address: localhost
Port: 1217

Device:

Parameter	Value	Comment
ipAddress	localhost	PLC address

PLC address

Gateway
IP address: localhost
Port: 1217

OK

PLC name or address: DESKTOP-T2978KQ
Use TCP/IP blockdriver: ✓
IP address of PLC: localhost
IP port of PLC: 11740

Figure50: Connection to PLC

SECTION 5

UNISIM

5.1 UNISIM Introduction

UNISIM acts as a client for our setup. The windows machine in which Unisim Simulation software is running is windows 7, there is no additional set up need to be done. Verify if client as well as server machine both in the same network. For our case study, both the machines are connected to the same LAN Adapter, pinged both the machine and checked if they are able to respond or not.

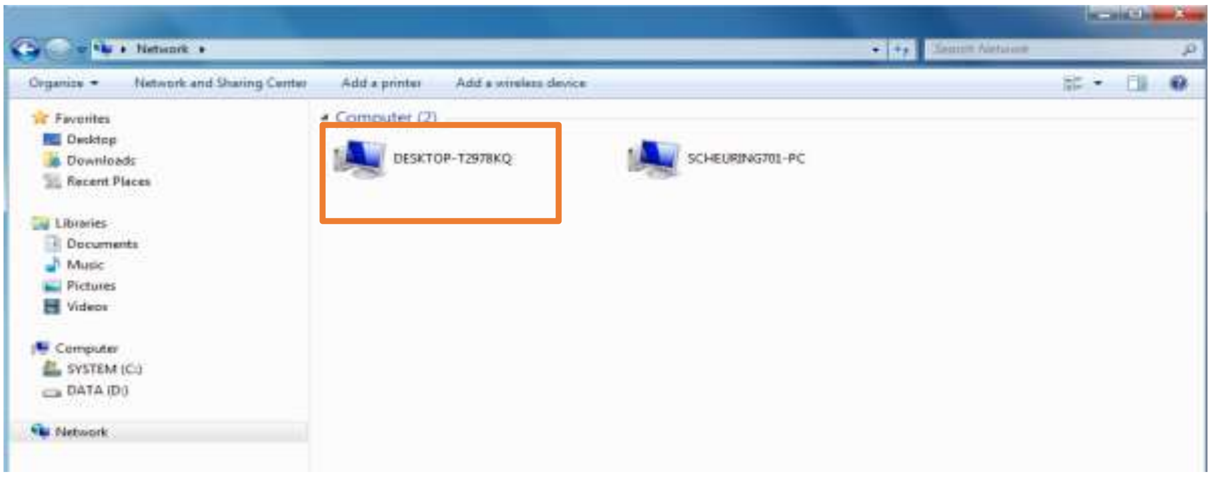


Figure51: Client and Server Machine in the same network

5.2 Unisim Configuration and Connection to the Server

In this project, implemented a simple project using Linear Valve and connecting with inlets and outlet. This acts as a plant just to establish the Interface to Automation software for making the DA Connection. Example of this is shown below:

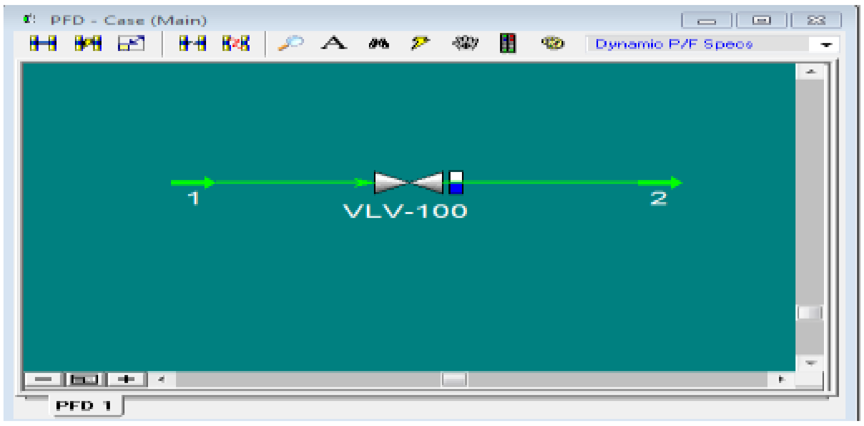
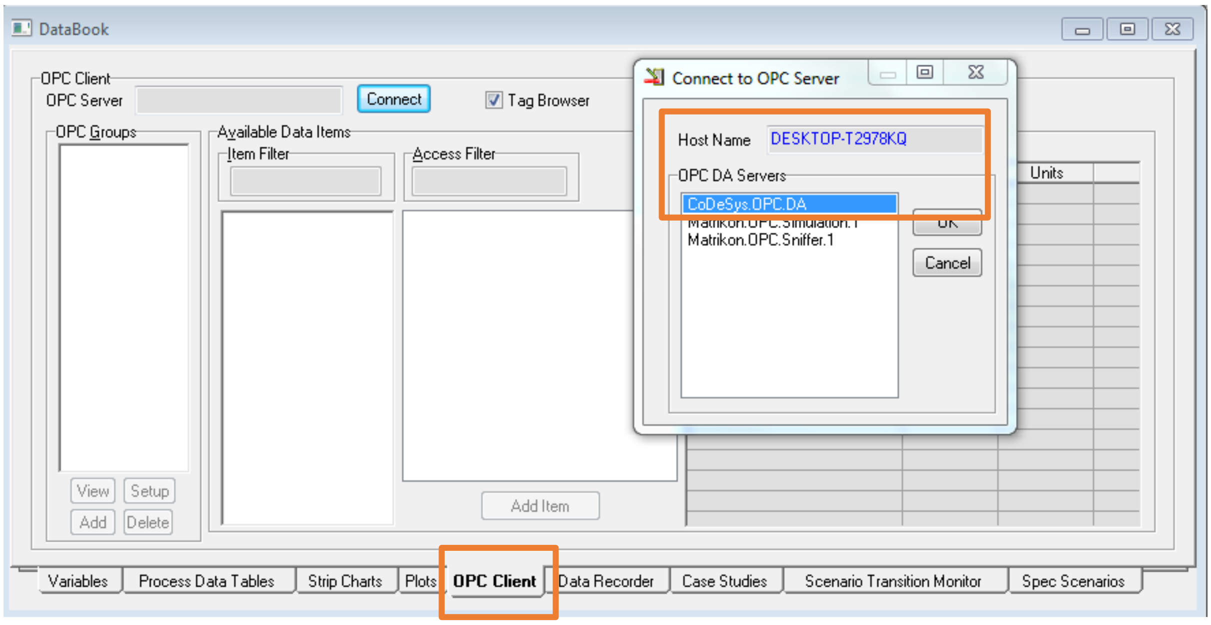


Figure52: Unisim Valve Project

For UNISIM select all the basic packages –NRTL and component like water to enter in the Basic Simulation Environment. Then, go to Tools in the upper tab and open “Databook”. Then click on the “OPC client tab” and connect with our CODESYS DA SERVER machine by specifying the device name and connect with the required server.



Connect to OPC Server

Host Name DESKTOP-T2978KQ

OPC DA Servers
CoDeSys.OPC.DA
Matrikon.OPC.Simulation.1
Matrikon.OPC.Sniffer.1

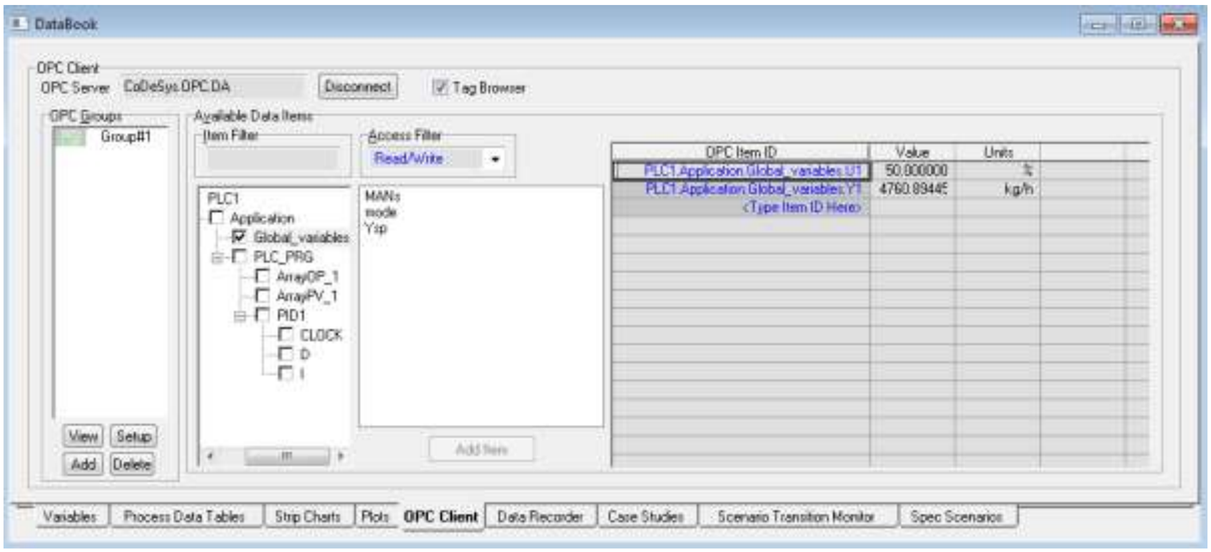
OK
Cancel

Figure53: Connection to the Codesys OPC server from Unisim Databook

After connection is established, Tick the “Tag Browser”. Select “ADD” in left pane and “Group1” appears, here insert the Global Variables “OP” AND “PV” from the PLC program of the DA server. In our Case,

Global_Variable U1 = OP = Valve Inlet 1 (Actuator Desired Position in %)

Global_Variable Y1 = PV = Mass flow in Kg/hr (Outlet 2 of the VLV 100)



OPC Server CoDeSys.OPC.DA

Group#1

Access Filter: Read/Write

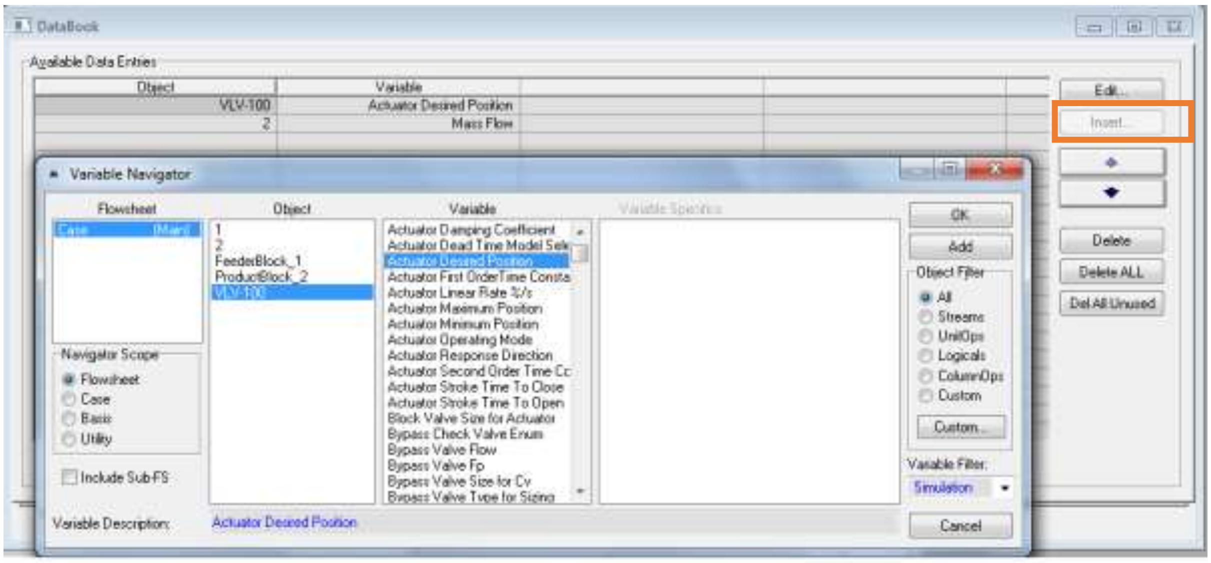
PLC1
Application
Global_variables
PLC_PRG
ArrayOP_1
ArrayPV_1
PID1
CLOCK
D
I

MANs
mode
Ysp

OPC Item ID	Value	Units
PLC1.Application.Global_variables.U1	50.000000	%
PLC1.Application.Global_variables.Y1	4760.89445	kg/h
<Type Item ID Here>		

Figure54: Databook with Group added

Now Choose "Variable" tab, and add the variables from UniSim client side which exchange data from the server side. This tab consists of all the Unisim's objects which is used in the controlled process. The required parameters can be inserted using "Insert" button. One example is shown below:



Object	Variable
VLV-100	Actuator Desired Position
2	Mass Flow

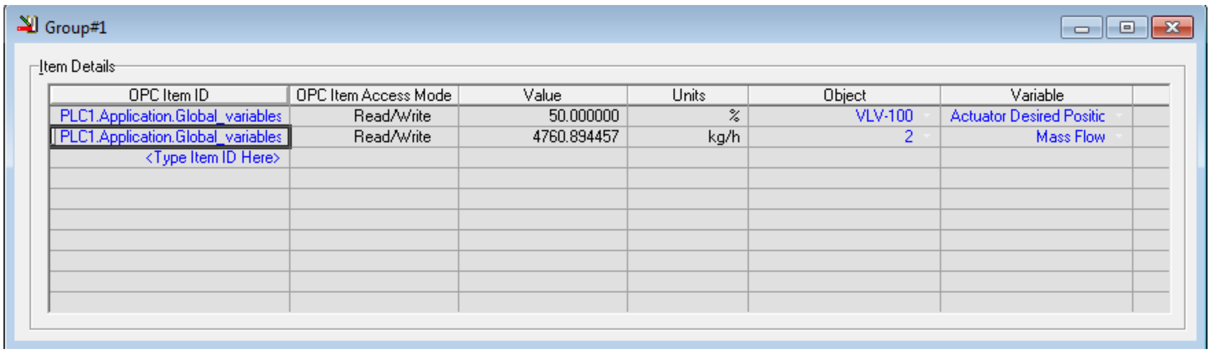
Object:
1
2
FeederBlock_1
ProductBlock_2
VLV-100

Variable:
Actuator Damping Coefficient
Actuator Dead Time Model Sele
Actuator Desired Position
Actuator First OrderTime Consta
Actuator Linear Rate %/s
Actuator Maximum Position
Actuator Minimum Position
Actuator Operating Mode
Actuator Response Direction
Actuator Second Order Time Cc
Actuator Stroke Time To Close
Actuator Stroke Time To Open
Block Valve Size for Actuator
Bypass Check Valve Enum
Bypass Valve Flow
Bypass Valve Fp
Bypass Valve Size for Cv
Bypass Valve Type for Sizing

Variable Description: Actuator Desired Position

Figure55: Adding variables from Unisim to Databook

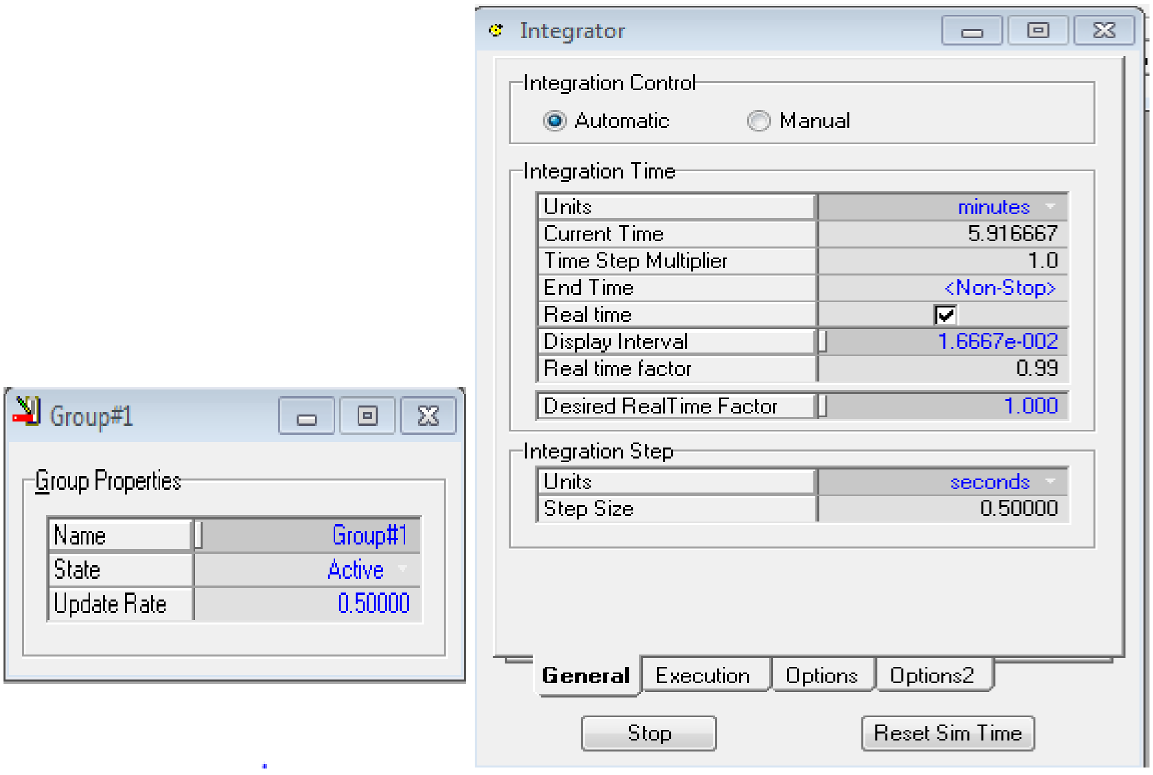
Finally, to check and edit the proper conjunction of variables from Server to Unisim Databook can be viewed using the "VIEW" button of the respective "Groups". The VIEW tab is shown below. Select the respective Object and Variable which denotes the UNISIM parameters.



OPC Item ID	OPC Item Access Mode	Value	Units	Object	Variable
PLC1.Application.Global_variables	Read/Write	50.000000	%	VLV-100	Actuator Desired Positic
PLC1.Application.Global_variables	Read/Write	4760.894457	kg/h	2	Mass Flow
<Type Item ID Here>					

Figure56: Conjunction of variables from Codesys PLC Program with Unisim

In Unisim (Client) "Integrator Step Size" and "Group Update Rate" plays a vital role while interfacing with OPC server to monitor the overall performance. Hence this should be appropriately chosen according to the application such that it gives better performance while running in simulation environment. This will affect the OPC DA communication and the lags between server and client if not chosen specifically.



Group#1

Group Properties

Name	Group#1
State	Active
Update Rate	0.50000

Integrator

Integration Control

[✓ Automatic] Manual

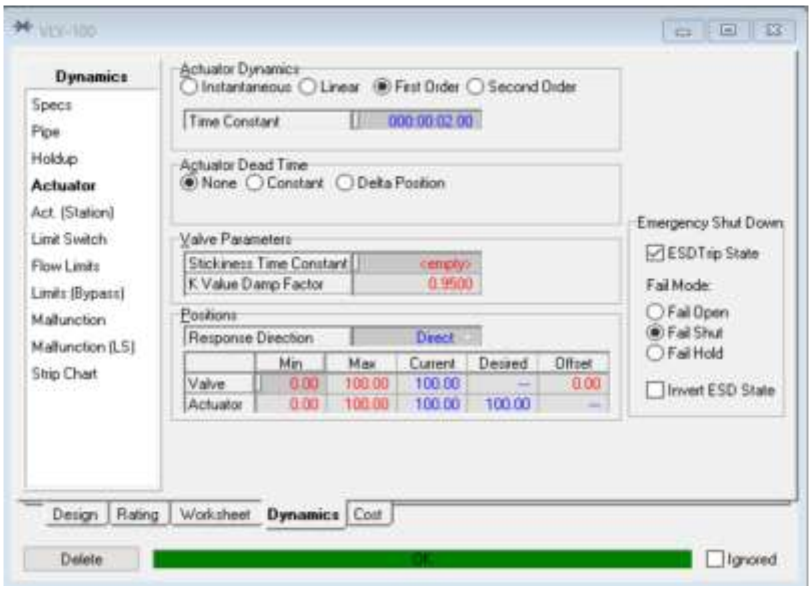
Integration Time

Units	minutes
Current Time	5.916667
Time Step Multiplier	1.0
End Time	<Non-Stop>
Real time	✓
Display Interval	1.6667e-002
Real time factor	0.99
Desired RealTime Factor	1.000

Integration Step

Units	seconds
Step Size	0.50000

Figure57: Integration Steps size and Group update rate



VLV-100

Dynamics

Actuator Dynamics

Instantaneous Linear [✓ First Order] Second Order

Time Constant	000:00:02.00

Actuator Dead Time

[✓ None] Constant Delta Position

Valve Parameters

Stickiness Time Constant	<empty>
K Value Damp Factor	0.9500

Positions

Response Direction	Direct

	Min	Max	Current	Desired	Offset
Valve	0.00	100.00	100.00	---	0.00
Actuator	0.00	100.00	100.00	100.00	---

Emergency Shut Down

[✓ ESDTrip State]

Fail Mode:

Fail Open [✓ Fail Shut] Fail Hold

Invert ESD State

Figure58: Valve Actuator Dynamics

Once all settings have been established and the interface is set up and running fine, the DA Communication can be analysed by reading/writing all data values and variables between the PID-PLC-SERVER-CLIENT. It is Obvious that the Performance-Simulation and Testing of this loop should be done which is a vital role in any OPC Interface between plant and automation system.

SECTION 6

SIMULATION EXPERIMENTS AND RESULTS

In our case-study, a simple Valve Project has been created in Unisim where OP, PV, SETPOINT and variables for different modes have been used for data transfer from server to client are executed. The main aim behind this is to monitor the delays between the sending and receiving of data from client and server. Also, checked the performance by varying the sample time, cycle time and other important parameters for different values and experiments.

The PID controller program in CODESYS are designed to operate in two modes. In manual mode, you adjust the output percentage and the controller ignores the set point. In auto mode, the controller adjusts the output to maintain the process variable at the desired set point.

6.1 PID in Auto Mode

The PLC program is now made to run in AUTO mode by setting in PID *"MANUAL=FALSE".* The setpoint for this mode was set to *"SET_POINT" = 44 kg/hr*. In Unisim, PV range was set to 9522 kg/hr as maximum flow. This was scaled from 0 to 100 kg/hr using SCALE_R function and then experiments were done. In this mode, the output signal (OP) is controlled by comparing PV with setpoint and continuously adjusts the OP using tuning constants KP=1 and TN=4 in our case. Also, here TV (Derivative time constant) is 0 and hence PI controller is used. As PI controller avoid large disturbances and noise presents during operation process.

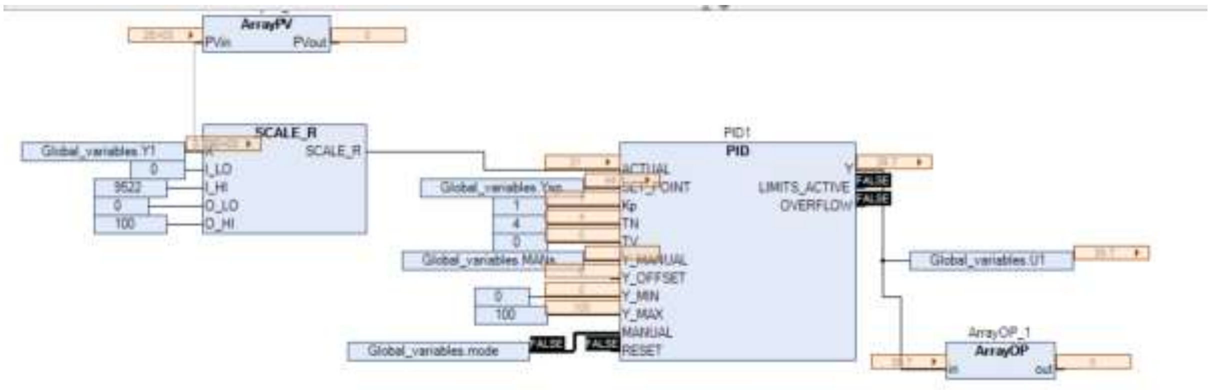
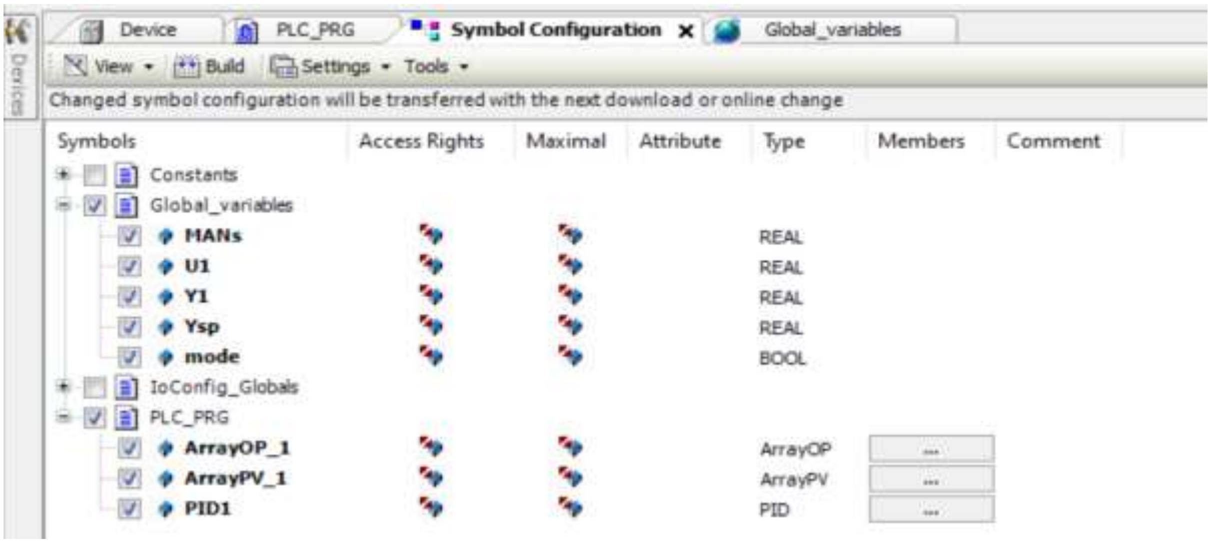


Figure59: PID Controller in AUTO mode

6.2 Symbol configuration of the PID_PLC program:

Inside the Symbol Configuration, a list of variables used for PID Controller is displayed. These variables are sent to OPC DA server and downloaded to the PLC device.


Device | PLC_PRG | Symbol Configuration | Global_variables

View | Build | Settings | Tools

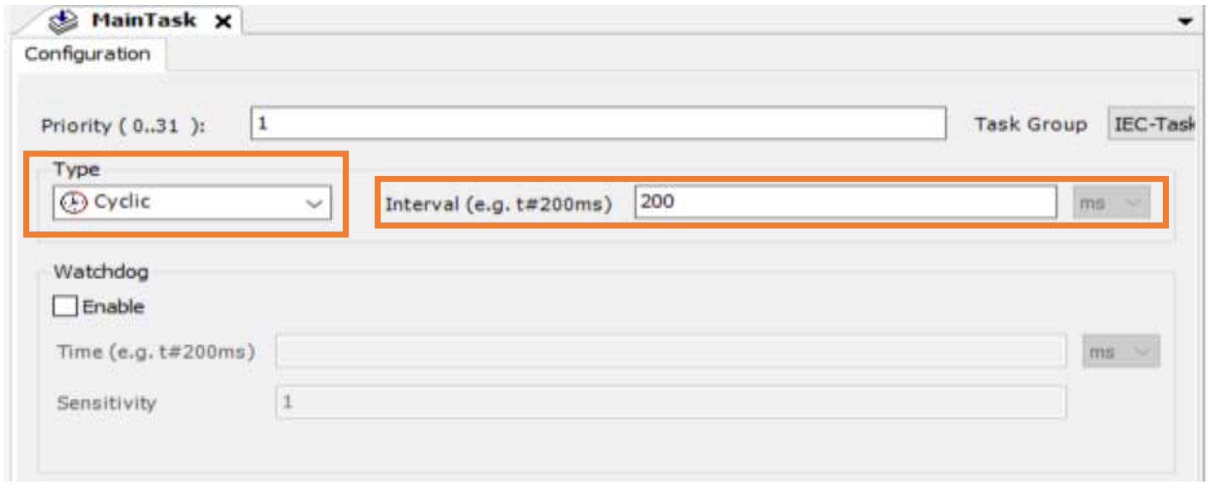
Changed symbol configuration will be transferred with the next download or online change

Symbols	Access Rights	Maximal	Attribute	Type	Members	Comment
☐ Constants						
☑ Global_variables						
☑ MANs				REAL		
☑ U1				REAL		
☑ Y1				REAL		
☑ Ysp				REAL		
☑ mode				BOOL		
☐ IoConfig_Globals						
☑ PLC_PRG						
☑ ArrayOP_1				ArrayOP	...	
☑ ArrayPV_1				ArrayPV	...	
☑ PID1				PID	...	

Figure60: Symbol Configuration of our PLC Program

6.3 Defining the PLC Cyclic time:

CODESYS processes the task of PLC in cycles. The cycle time of the task is defined in the input field Interval as 200ms which is a standard value. It's the Time interval after which the task should be restarted. In our case, modify this cyclic time for various runs to check the performance delays between server and client. For Example: 200ms,100ms and 50ms. The results of these are discussed in below section.


MainTask

Configuration

Priority (0..31): 1

Task Group: IEC-Task

Type: Cyclic

Interval (e.g. t#200ms): 200 ms

Watchdog

☐ Enable

Time (e.g. t#200ms) ms

Sensitivity 1

Figure61: Cyclic Time of PLC

6.4 Visualization in CODESYS:

In Visualization toolbox, "*TRACE*" was selected to see the plots in GUI way once the program is run. One can choose the different modes by pressing this red button which is programmed in our PLC. Here, OP and PV plots can be seen w.r.t time by adjusting setpoint on the left slide bar and modes in the button. Also, possible to choose the values in manual mode with this right slide bar.

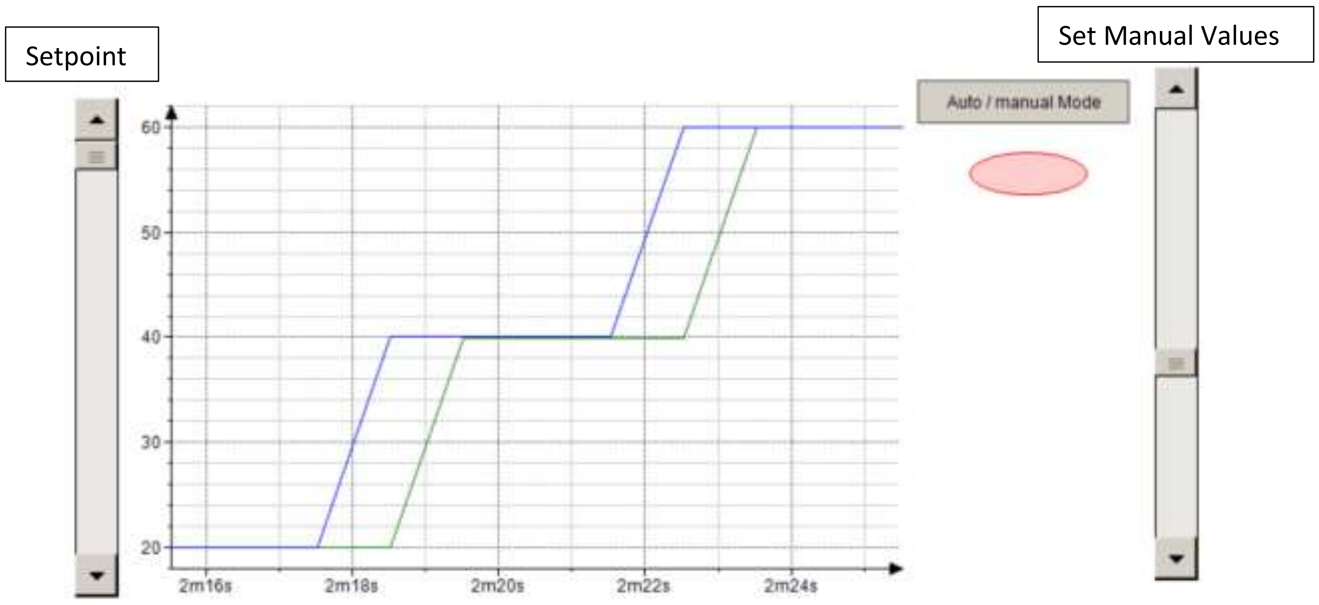


Figure62: Visualization of our PID prog- Auto/Manual Mode can be selected with red button

6.4.1 Trace in Auto mode:

This plot is obtained by using the Visualization window **where setpoint was changed from 0 to 33 and then a jump was made to 75**. Blue curve denotes OP, Green denotes PV and Red curve denotes the SETPOINT. The result graph obtained below are by selecting TRACE in *Application → Add Object → Trace.*

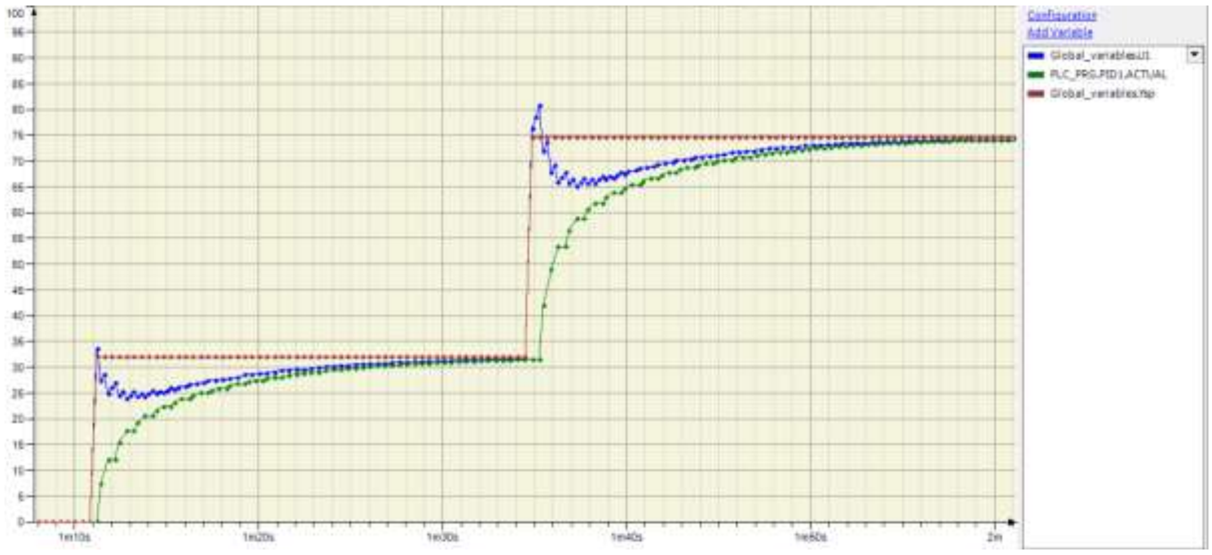


Figure64: Trace Visualization in Auto mode.

- Here, a slight delay of the green curve can be seen (PV) from the blue curve (OP).
- The graph shown below is the zoomed part when the PV reaches SP after some time in automatic mode

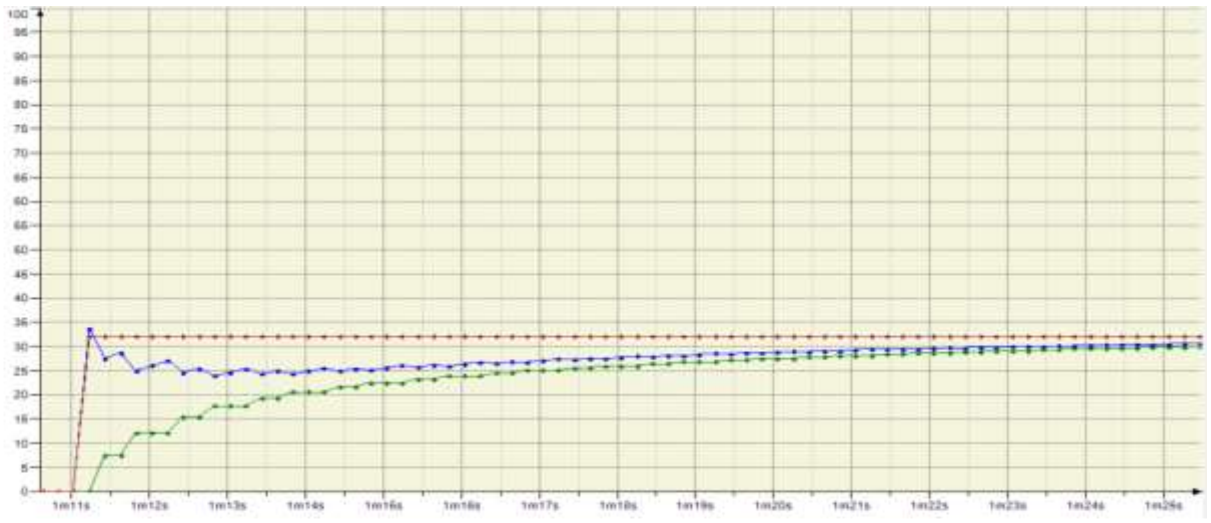
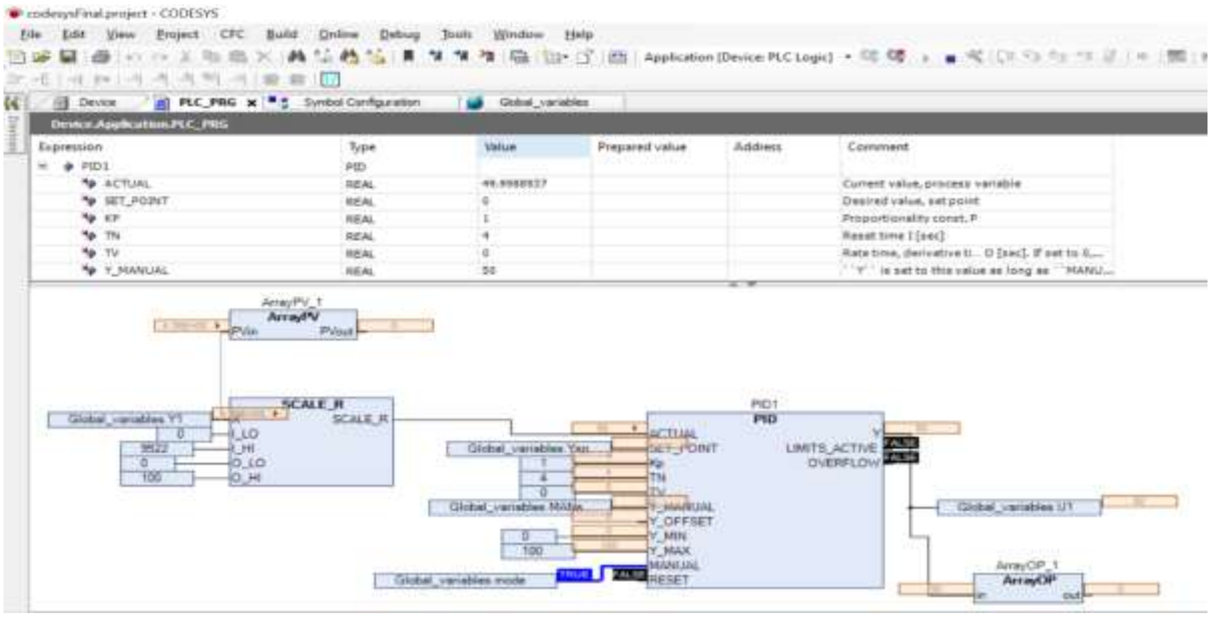


Figure65: Zoomed Graph when SP reaches PV (Auto Mode)

6.5 PID in Manual Mode

In this program, the PV value is fetched from Unisim and then insert into *"Global_variables.Y1"* in PLC which is scaled into range of 0-100 using SCALE_R function. These scaled PV values are fed to PID controller Block input variable *"ACTUAL"*. Setting the manual mode can be done in Visualization window. OP of the controller in manual mode can be controlled using the variables *"Y_MANUAL"* in the PID block.



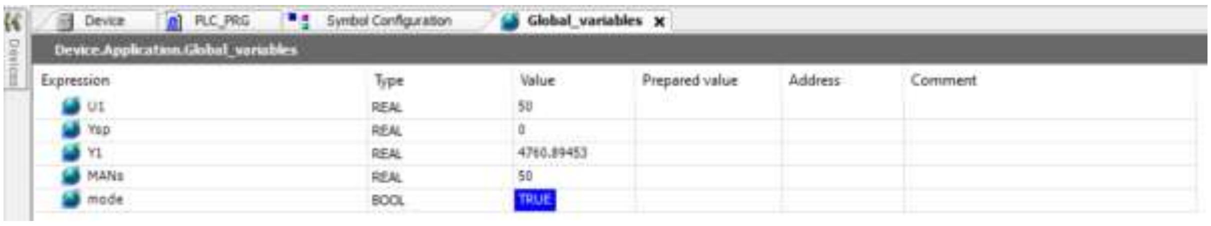
codesysFinal.project - CODESYS

Device.Application.PLC_PRG

Expression	Type	Value	Prepared value	Address	Comment
PID1	PID				
ACTUAL	REAL	49.9988837			Current value, process variable
SET_POINT	REAL	0			Desired value, set point
KP	REAL	1			Proportionality const. P
TN	REAL	4			Reset time I [sec]
TV	REAL	0			Rate time, derivative ti... D [sec]. If set to 0,...
Y_MANUAL	REAL	50			''Y'' is set to this value as long as ''MANU...

Figure58: PID working in Manual Mode

In global variables, we used "*Global_variables.mode"* as True and False to switch to manual and automatic mode and it also can be done in visualization. Another way to operate in manual mode is to prepare and force these values (say 20%,40%,60%....100%) in Global variables section called *"MANs*" and then write them into the PLC and see the plots. Below is the example of MANUAL MODE SET TO TRUE.



Expression	Type	Value	Prepared value	Address	Comment
U1	REAL	50			
Ysp	REAL	0			
Y1	REAL	4760.89453			
MANs	REAL	50			
mode	BOOL	TRUE			

Figure59: Global variables where Manual Mode is set to TRUE

6.5.1 Trace in Manual Mode: Valve Instantaneous

In Unisim (Client-plant), VLV-100 was selected as Instantaneous and integrator step size as 500ms and "Databook" Group Update rate as 500ms. In CODESYS (server) cyclic time was selected as 200ms and OPC configurator update rate as 200ms. After these settings, Global variable are selected in manual mode and OP values 20,40,60,80,100 % are forced in "*MAN*" global variable. Now the program is built and made to run.

Cyclic time (ms)	OPC configurator update rate (ms)	Group update rate (ms)	Step size (ms)	Experiments delays (ms)
200	200	500	500	400
				400
				200
				200
				400
			Mean	320
			Std	97.97958971
			Min	200
			Max	400
			Median	400
			Mode	400

Table4: Tabulation of Standard case

The above values were observed for the default parameters to monitor the delay between CODESYS server and UNISIM client interaction. Also, the statistics were calculated and the Average delay was 320ms with a standard deviation of 97.97ms.

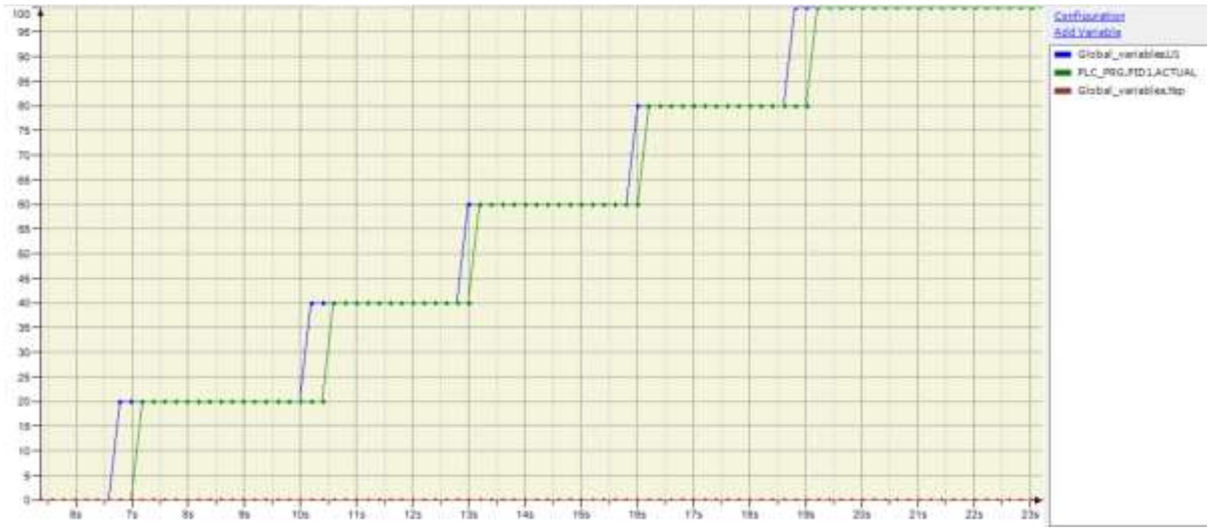


Figure66: Sample Trace with default values of PID-PLC Control loop to monitor the delays

6.5.2 TRACE (Manual Mode: Valve in First Order and Linear)

In Unisim, VLV-100 was set to First order and Linear and the below graph was obtained in TRACE with standard values.

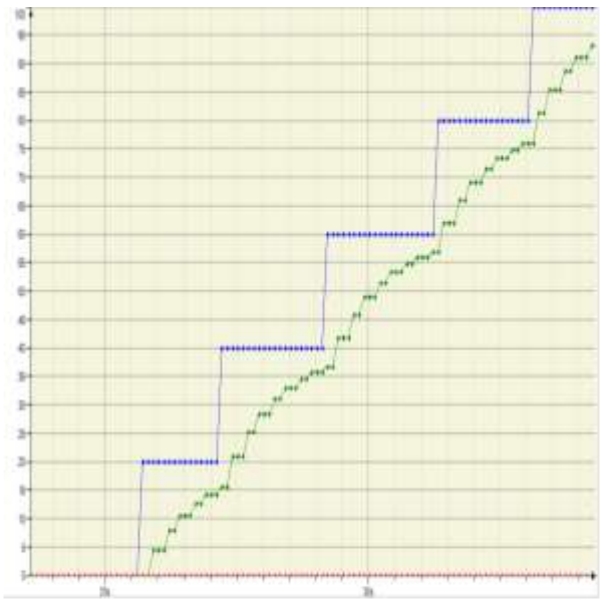


Figure67: First Order-VLV100

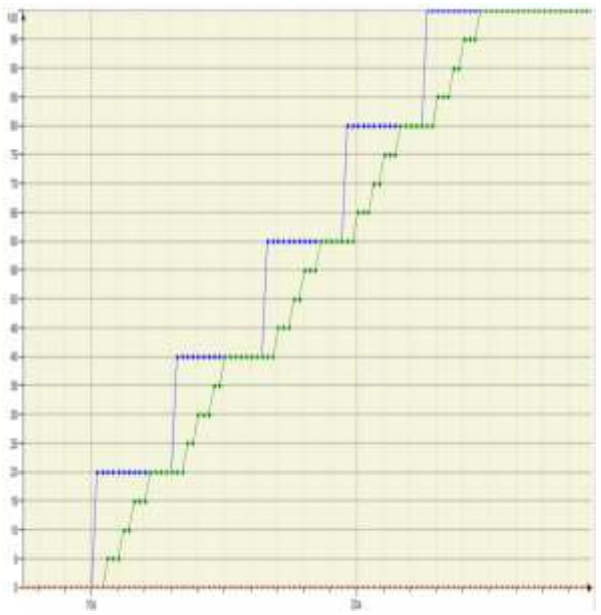


Figure68: Linear-VLV100

In above figures, valve actuator dynamics as First Order and Linear was selected to observe the difference in trace for both cases. Blue curve shows OP of valve while Green curve shows PV of PID controller. To obtain these plots, set integrator step size to 500ms, Group Update rate as 500ms and cyclic time 200ms. In case of Valve First order OP values are changed from 20,40,60,80,100% and corresponding Values of PV has been changed and delays has been observed between OP and PV values. Similarly, for linear valve OP and PV values have some delays when jumps are made from 20% to 100% OP values.

When valve is in instantaneous mode then it is good to calculate delays more accurately as compared to valve in first order and linear case as expected. For performance delays, set the valve in instantaneous mode and changed different parameters to analyse results.

6.6 RESULTS FOR PERFORMANCE DELAYS:

For measuring the delays in this control loop (PID-PLC-SERVER-CLIENT), CODESYS development environment i.e PLC-PID program was kept in running mode, the TRACE was downloaded and then OP values was prepared and forced from 20,40,60,80,100%. The experiments were repeated by changing the PLC cyclic time, OPC Configurator Update Rate (200 and 100ms), Group Update rate and Integrator Step size. The below results were obtained.

TABLE 6.9.1: Cyclic Time=200ms, OPC Configurator Update Rate=200ms, Step Size=500ms.

Group Update Rate (ms)	Average Delays (ms)	Standard deviation	Min	Max	Median	Mode
500(Default)	400	126.49	200	600	400	400
100	360	149.66	200	600	400	200
1000	600	219.08	200	800	600	800
250	400	126.49	200	600	400	400
200	400	126.49	200	600	400	400

Observation: When changing group update rate from 200,250 from 500 (default) there is no change in average delay as it is the same as 400ms. The longer the time of the group update rate, the longer the delay.

TABLE 6.9.2: Cyclic time=100ms, OPC Configurator Update Rate=200ms and 100ms, Step Size=500ms

Group Update Rate(ms)	Opc configurator update rate(ms)	Average Delays(ms)	Standard deviation	Min	Max	Median	Mode
500	200	340	205.91	100	600	400	100
100	200	280	146.96	100	400	400	400
500	100	260	48.98	200	300	300	300

Observation: When **varying the OPC configurator update rate** from 200 to 100ms, for a constant step size and group update rate the delay is reduced from 340ms to 260ms.

TABLE 6.9.3: Cyclic time=100ms, OPC Configurator Update Rate=100ms, Step Size=200ms

Group Update Rate (ms)	Average Delays (ms)	Standard deviation	Min	Max	Median	Mode
200	240	48.98	200	300	200	200
100	160	48.98	100	200	200	200

Observation: When **varying the group update rate** from 200 to 100ms, average delay reduced from 240ms to 160ms.

TABLE 6.9.4: Cyclic time=50ms, OPC Configurator Update Rate=200ms, Step Size=500ms

Group Update Rate(ms)	Average Delays(ms)	Standard deviation(ms)	Min	Max	Median	Mode
500	290	146.28	100	500	350	350
100	280	74.83	200	400	300	200
1000	520	140	350	750	550	550

Observation: When **changing group update rate 500,100,1000ms for cyclic time of 50ms** and keeping all other factors i.e OPC configurator and step size constant, observed that the longer the time of group update rate longer will be the delay

TABLE 6.9.5: Cyclic time=20ms, OPC Configurator Update Rate=200ms, Group update rate=100ms, Step Size=500ms

Group Update Rate(ms)	Cyclic time (ms)	Average Delays (ms)	Standard deviation	Min	Max	Median
100	20	316	79.39	220	420	280

Observation: Cyclic time 20ms is minimum cyclic time which is set in Codesys to check the delays with default values of OPC configurator update rate 200ms and step size 500ms, observed that for this particular case average delay of 316ms has been obtained. The Cyclic time of 20ms is not feasible for all applications.

6.7 CONCLUSION:

In this Case-study, OPC DA interface between client – server was established and analysed within the network. A control loop was developed between the CODESYS Automation software and UNISIM plant, and then data was transferred from and to the server. Also, the controllability and communication in this loop was studied and simulation were performed.

It was observed in Unisim that the parameters - Group Update Rate and Integrator Step size plays a vital role for any OPC interface and communication. And in any Automation Software i.e in CODESYS the parameters cyclic time, OPC Configurator rate plays a significant role. Hence for a better performance of the controlled loop, the plant time constant of 2seconds, Group Update rate of 500ms and Integrator Step Size of 500ms would be standard case. And in CODESYS the Update rate of the OPC server is by default 200ms and it should be lesser than the Integrator Step size in plant. The Cyclic time of 20ms is not feasible for all applications.

In the control loop, the OP and PV values, performance delay was studied by feeding back the PV from the UNISIM to the Codesys server. While changing the group update rate from the standard case, it was observed that the longer the update rate like 1000ms the longer was the delay of 600ms and it was as expected. For a feasible performance of the client and server interface using DA communication, below are the following conditions:

Plant Time Constant > Integrator Step Size > Group Update Rate

Integrator Step Size >> OPC Configurator Update rate

SECTION 7

UNISIM AS CLIENT WITH HIMA SERVER – EXPERIMENTS & RESULTS

7.1 Introduction

OPC DA (Data Access): transfer the real-time and ensure the global variables are read from a controller and written to the OPC Client or vice versa.

An HIMA OPC DA Server can communicate with 254 controller, supports up to 10 OPC Clients and a maximum of 100 000 DA tags. Maximum process data volume that a HIMA controller can exchange to one OPC Server is 128 kB.

7.1.1 Hardware and software requirements:

- PC with the following features:
  - Core Duo
  - 3 GB RAM
  - approx. 20 MB hard disk space available
  - Windows XP Professional, SP2 (32-bit) or higher, or
  - Windows Server 2003, SP1 (32-bit) or higher, or
  - Windows 7 Ultimate / Professional (32-bit or 64-bit), or
  - Windows Server 2008 R2 (64-bit)
- A SILworX full version 4.116 (or beyond) is necessary for operating and programming the X-OTS (OPC SERVER). SILworX is the programming and debugging tool (PADT) developed by HIMA. The tool also enables online access to systems for diagnostics, forcing, and other purposes.
- OPC client
- The Ethernet network should have a bandwidth of at least 100 Mbit/s (better 1GBit/s).
- X-OTS (OPC SERVER) is the simulation of a HIMax or HIMatrix controller.

In this case study, used X-OTS to test user programs. Also, used the OPC interface of X-OTS to connect control to process simulators (UniSim). The OPC interface of X-OTS is suitable for data access (DA). X-OTS allocates all the global variables of the SILworX projects as OPC tags.

7.1.2 Architecture:

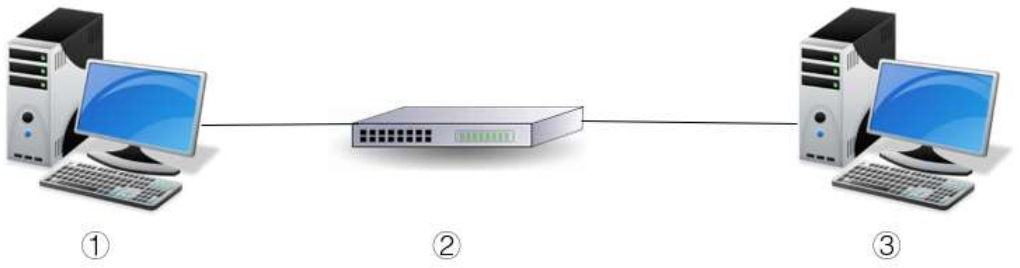


Figure69: Architecture.

① OPC Client (UniSim)

② Switch

③ SILworX & X-OTS (OPC SERVER)

7.2 Establish communication

In this case study in order to establish the OPC communication follow the below steps:

1. Installation of X-OTS (OPC SERVER) on a Windows PC

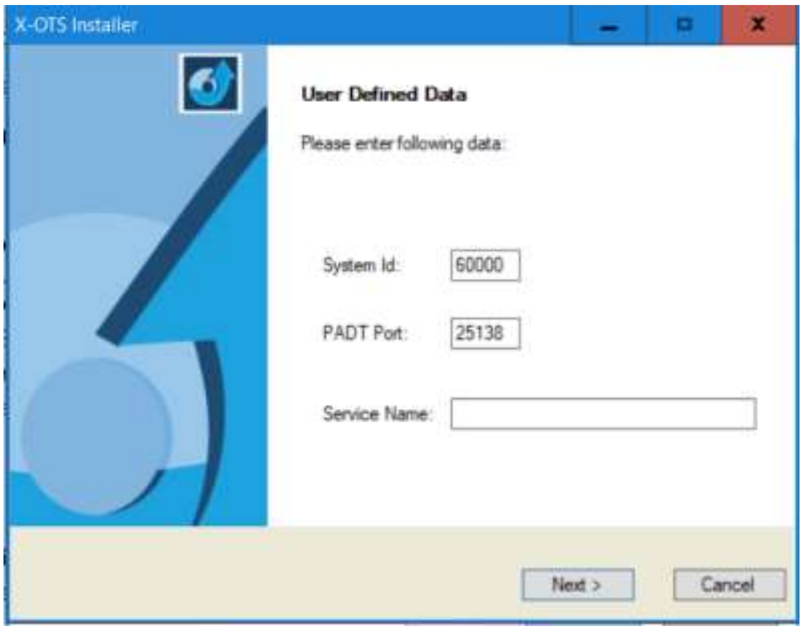
2. Configuration of an OTS resource using SILworX

3. Creation of the programs, code generation and load

4. installation and configuration of OPC client.

7.2.1 Installation

During installation, the following parameters must be specified:



User Defined Data

Please enter following data:

System Id: 60000

PADT Port: 25138

Service Name:

Figure70: X-OTS(Server) parameters.

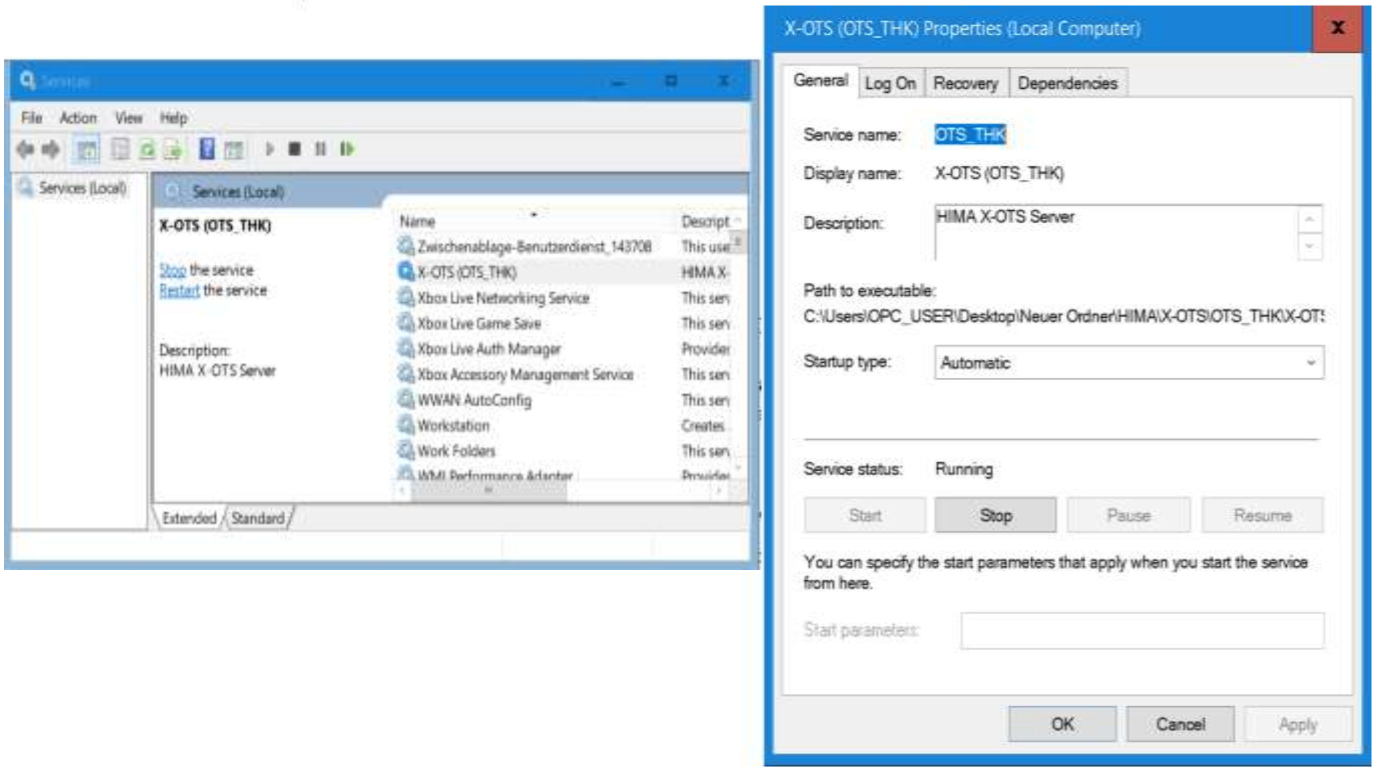
- System ID: System ID assigned to the OTS resource which is specified during the SILworX configuration.
- PADT Port: Number of the port that establishes the connection between X-OTS (OPC SERVER) and SILworX. This port number must be greater than 1024 to prevent problems with other programs.
- Service name

Up to 10 X-OTS (OPC SERVER) instances can be installed on a PC. X-OTS (OPC SERVER) operates as a service and runs under the Windows operating system. Automatic start of the X-OTS (OPC SERVER) or OPC Servers after restarting the PCs can be configured during the installation or:

1. In Windows, go to Start, Settings, Control Panel, Administration, Services and select X-OTS (OPC SERVER) from the list.

2. Select Properties from the context menu.

3. In the General tab, select the Automatic start type.



X-OTS (OTS_THK)

Stop the service
Restart the service

Description:
HIMA X-OTS Server

Name	Descript
Zwischenablage-Benutzerdienst_14370B	This use
X-OTS (OTS_THK)	HIMA X-
Xbox Live Networking Service	This ser
Xbox Live Game Save	This ser
Xbox Live Auth Manager	Provider
Xbox Accessory Management Service	This ser
WWAN AutoConfig	This ser
Workstation	Creates
Work Folders	This ser
WMI Performance Adapter	Provider

Service name: OTS_THK
Display name: X-OTS (OTS_THK)
Description: HIMA X-OTS Server
Path to executable:
C:\Users\OPC_USER\Desktop\Neuer Ordner\HIMA\X-OTS\OTS_THK\X-OT
Startup type: Automatic
Service status: Running
You can specify the start parameters that apply when you start the service from here.

Figure71: Automatic start of the X-OTS

In this project the SILWorX and X-OTS (OPC SERVER) are on same PC that firewall was installed on it thus the TCP/UDP PADT and HH ports of the X-OTS have configured as exception in the firewall configuration. Since RPC server uses ports in the range 1024 to 65535 this port range configured as exception as well.

7.2.2 Configuration

To configure an OTS resource use SILworX to add X-OTS as new resource in a configuration.

7.2.2.1 To create an OTS resource:

1. Choose the configuration and select New from the context menu or the action bar.

2. Click Operator Training System.

3. Enter the name of the new OTS resource in the Name field.

4. Click OK.

7.2.2.2

Also an OTS resource must be licensed to operate. The activation code must then be specified in the SILworX license management.

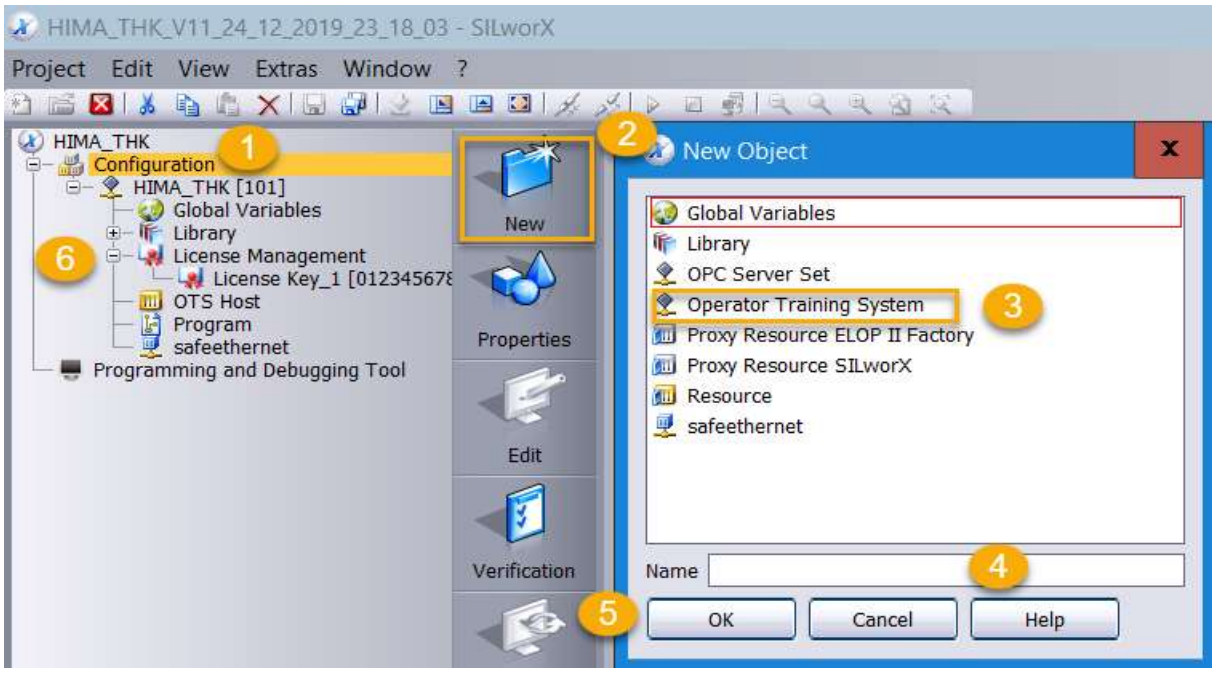
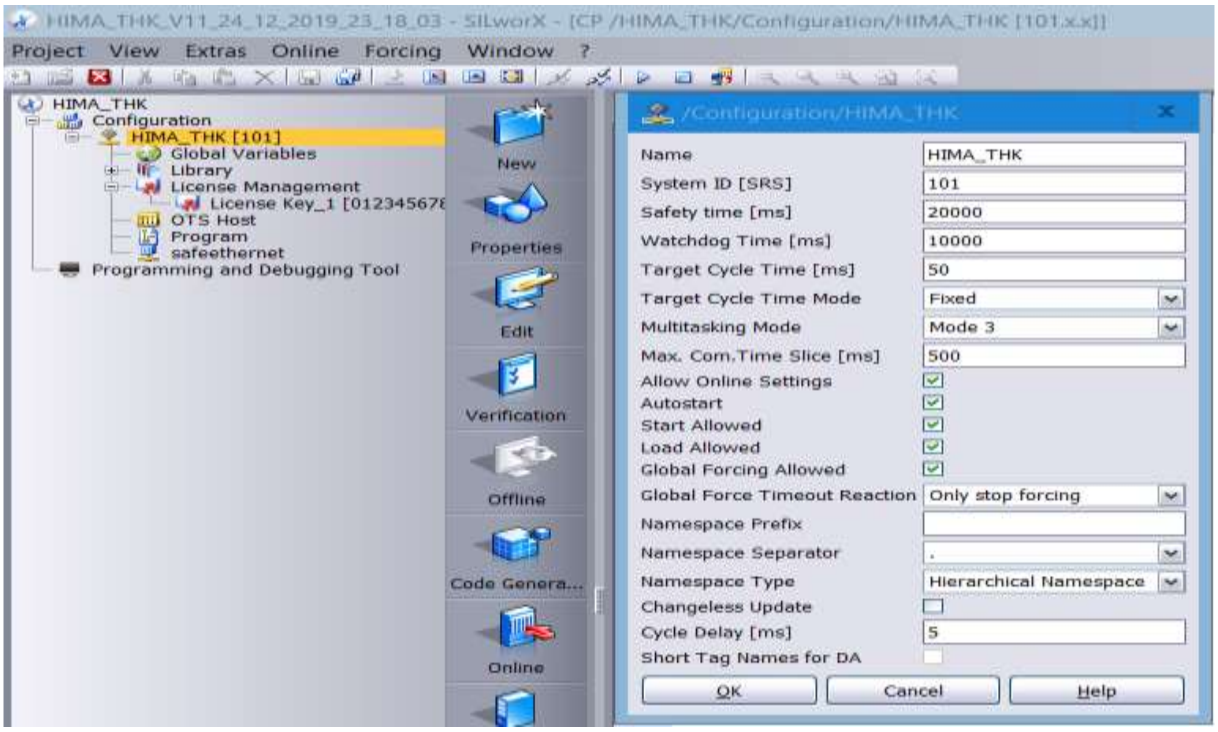


***Figure72: Steps to create* OTS resource in SILworX.**

Important system Parameters of the OTS Resource to be set:

- Name: Name of the OTS resource.
- System ID: The system ID value must be identical to the value set when installing the corresponding X-OTS (OPC SERVER) instance.
- Max.Com. Time Slice ASYNC [ms]: is the time period in milliseconds (ms) per CPU cycle assigned to the processor module for processing the communication tasks.
- Target Cycle Time [ms]: The target cycle time defines the desired duration of a CPU cycle. A user program is only completely processed once during each CPU cycle.
- Target Cycle Time Mode: In this case study, use Fixed mode that means, if a CPU cycle is shorter than the defined Target Cycle Time, the CPU cycle is extended to the target cycle time.

- Multitasking Mode: In this case study, Mode 3 was used that means the processor waits during the unneeded execution time of user programs to expire and thus increases the cycle.
- Changeless Update: If it set to ON, cyclically, X-OTS (OPC SERVER) always provides all the items to the OPC client. If it set to OFF, X-OTS (OPC SERVER) only provides changed values to the OPC client.



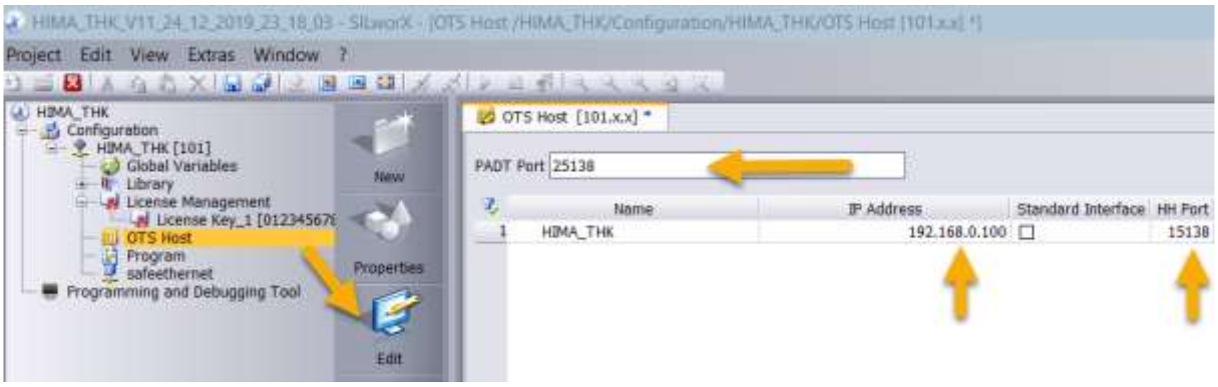
Name	HIMA_THK
System ID [SRS]	101
Safety time [ms]	20000
Watchdog Time [ms]	10000
Target Cycle Time [ms]	50
Target Cycle Time Mode	Fixed
Multitasking Mode	Mode 3
Max. Com.Time Slice [ms]	500
Allow Online Settings	✓
Autostart	✓
Start Allowed	✓
Load Allowed	✓
Global Forcing Allowed	✓
Global Force Timeout Reaction	Only stop forcing
Namespace Prefix	
Namespace Separator	.
Namespace Type	Hierarchical Namespace
Changeless Update	☐
Cycle Delay [ms]	5
Short Tag Names for DA	☐

Figure73: **Parameters of the OTS Resource.**

7.2.2.3 OTS Host Set-Up:

The OTS host is a sub-element of the OTS resource and contains parameters of the computer on which the X-OTS (OPC SERVER) is running. Right-click anywhere in the OTS host editor and select New IP Connection.

- Set the PADT port same as value specified during the X-OTS (OPC SERVER) installation.
- For Ethernet connection parameters, Set the IP address of the PC on which the X-OTS (OPC SERVER) is installed as well as HH Port that process data connections to the partners such as other X-OTS (OPC SERVER) instances, Controllers and OPC server.


HIMA_THK_V11_24_12_2019_23_18_03 - SILworX - [OTS Host /HIMA_THK/Configuration/HIMA_THK/OTS Host [101.x.x] *]
Project Edit View Extras Window ?
HIMA_THK
Configuration
HIMA_THK [101]
Global Variables
Library
License Management
License Key_1 [01234567
OTS Host
Program
safeethernet
Programming and Debugging Tool
New
Properties
Edit
OTS Host [101.x.x] *

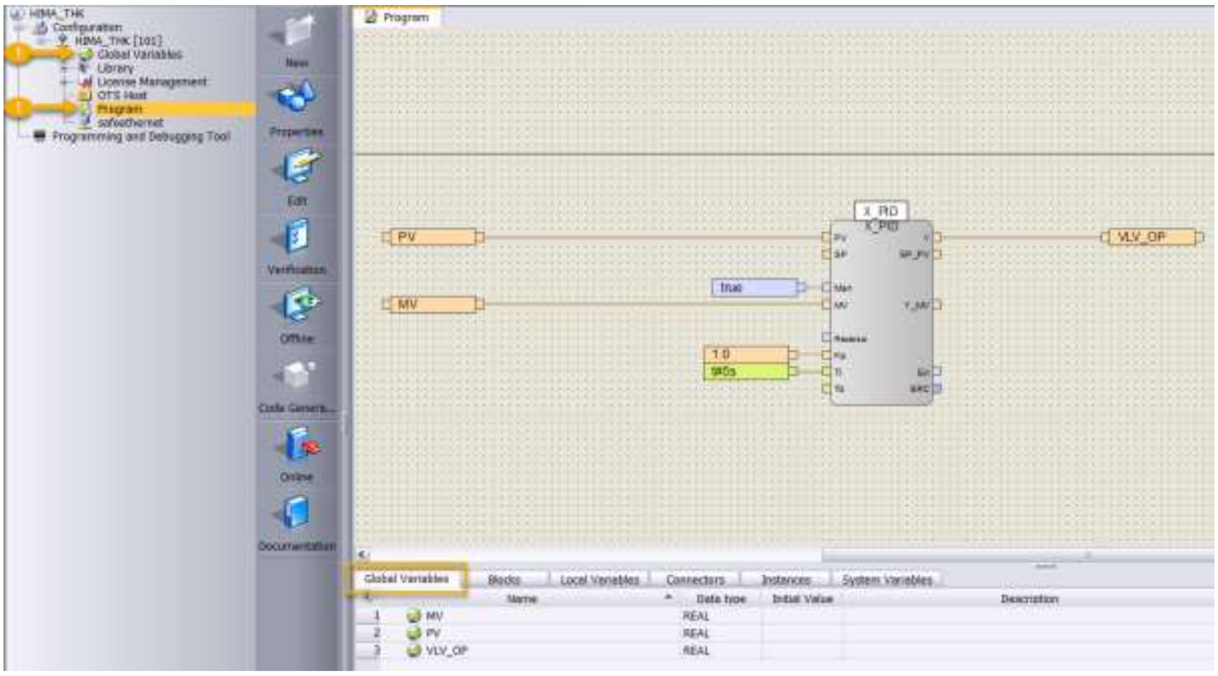
PADT Port 25138

	Name	IP Address	Standard Interface	HH Port
1	HIMA_THK	192.168.0.100	☐	15138

Figure74: OTS Host Set-Up.

7.2.3 Creating Programs, Generating and Loading the Code

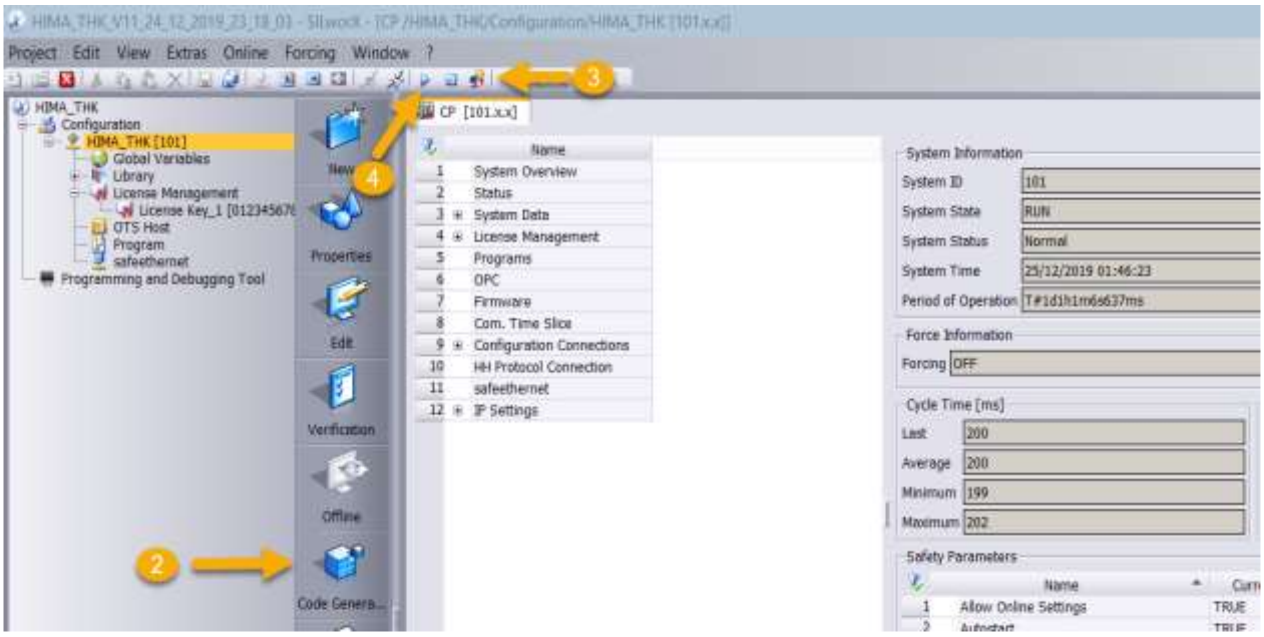
1. The user program and the corresponding global variables have been created.



	Name	Data type	Initial Value	Description
1	MV	REAL		
2	PV	REAL		
3	VLV_OP	REAL		

Figure75: Creating program and global variables.

2. The code has been generated to the resource.
3. Generated code has been loaded into the OTS resource.
4. OTS resource has been started with SILworX. The OPC tags of an OTS resource are only available if the resource is in RUN.



HIMA_THK_V11_24_12_2019_23_18_01 - SILworX - [CP /HIMA_THK/Configuration/HIMA_THK [101.x.x]]

	Name
1	System Overview
2	Status
3	System Data
4	License Management
5	Programs
6	OPC
7	Firmware
8	Com. Time Slice
9	Configuration Connections
10	HH Protocol Connection
11	safeethernet
12	IP Settings

System Information

System ID	101
System State	RUN
System Status	Normal
System Time	25/12/2019 01:46:23
Period of Operation	T#1d1h1m6s637ms

Force Information

Forcing	OFF

Cycle Time [ms]

Last	200
Average	200
Minimum	199
Maximum	202

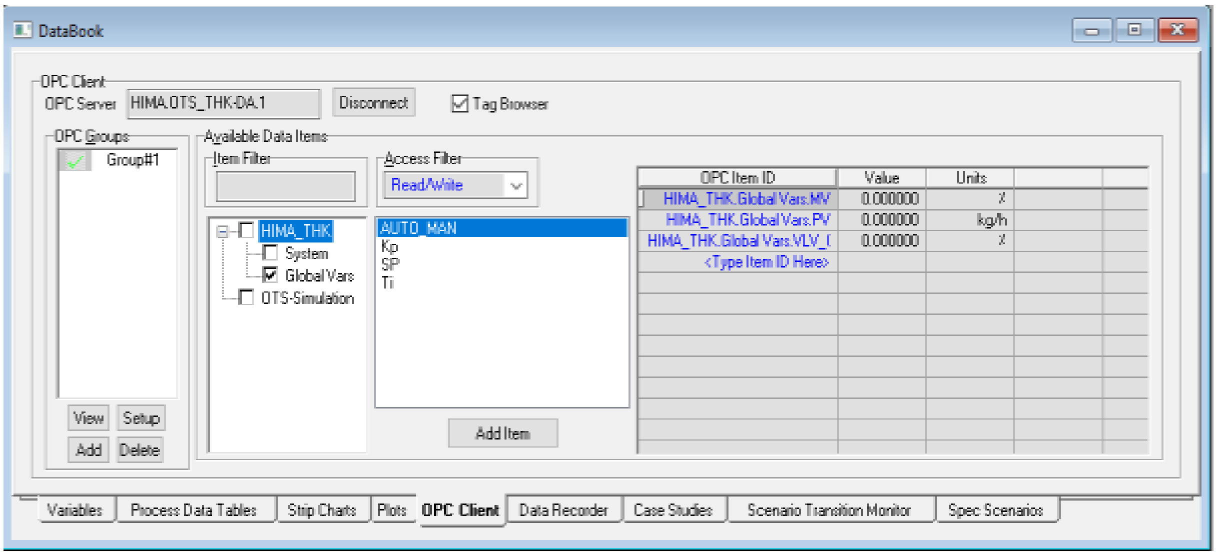
Safety Parameters

	Name	Curr
1	Allow Online Settings	TRUE
2	Autostart	TRUE

Figure76: Code generation and download.

The OPC tags are classified in the following groups:

- Simulation interface: The simulation interface is composed of OPC tags that can be used to control the simulation.
- Global variable: The global variables defined in SILworX for the resource are available as OPC tags.
- System variable: The resource's system variables are available as OPC tags.



OPC Client

OPC Server: HIMA.OTS_THK-DA.1

OPC Groups: Group#1

Available Data Items

Item Filter:

Access Filter: Read/Write

HIMA_THK
- System
- ✓ Global Vars
- OTS-Simulation

AUTO_MAN
Kp
SP
Ti

OPC Item ID	Value	Units
HIMA_THK.Global Vars.MV	0.000000	%
HIMA_THK.Global Vars.PV	0.000000	kg/h
HIMA_THK.Global Vars.VLV_I	0.000000	%
<Type Item ID Here>		

Figure77: Available data item.

Number of OPC tags per X-OTS resource, depend on the PC performance.

7.2.4 Configuration of OPC client

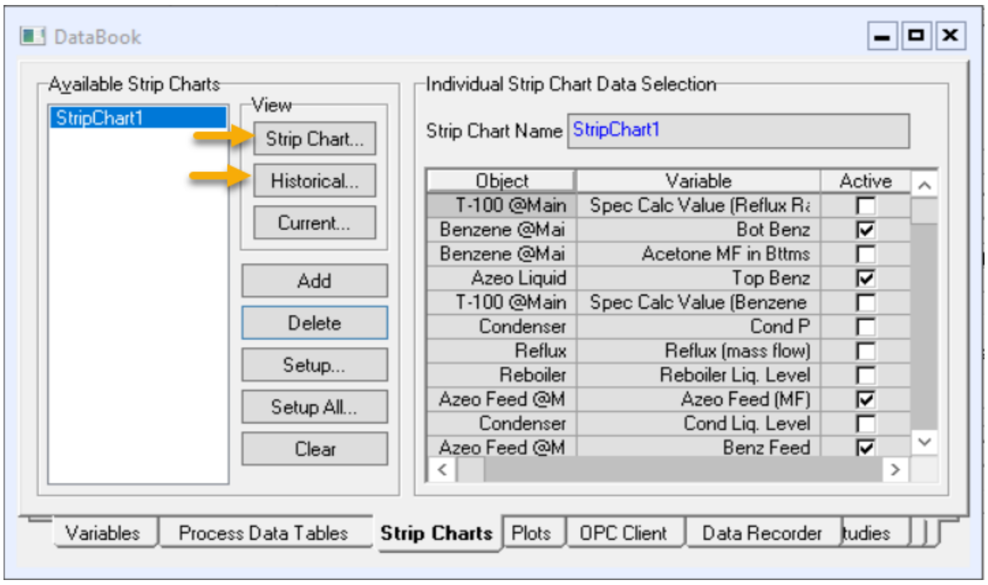
All the settings and configurations has been done as it described in section 5.

7.3 EXPERIMENTS & RESULTS

In this series of experiments, the plant includes a Linear Valve with First Order Actuator connected to the inlet and outlet pipe. OP (manual opening) and PV (mass flow) variables have been transferred between OPC server and client. The PID controller program in SILworX implemented and operated in manual mode.

The experiments were done by varying the integrator step size, cycle time and group update rate and the delay on transferring data between OPC client and server has been monitored by using OP and PV variables.

Strip chart and Historical Data view from strip charts tab of data book used to record the graph and CSV file as experiment's results.

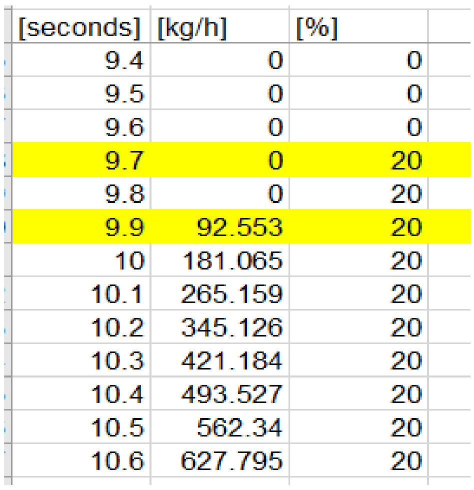


Strip Chart Name StripChart1

Object	Variable	Active
T-100 @Main	Spec Calc Value (Reflux Ra	☐
Benzene @Mai	Bot Benz	☑
Benzene @Mai	Acetone MF in Bttms	☐
Azeo Liquid	Top Benz	☑
T-100 @Main	Spec Calc Value (Benzene	☐
Condenser	Cond P	☐
Reflux	Reflux (mass flow)	☐
Reboiler	Reboiler Liq. Level	☐
Azeo Feed @M	Azeo Feed (MF)	☑
Condenser	Cond Liq. Level	☐
Azeo Feed @M	Benz Feed	☑

Figure78: Strip chart and Historical data.

Below you can see an exported CSV file from Historical data for integrator size of 100ms which is used for precise reading of delay.



[seconds]	[kg/h]	[%]
9.4	0	0
9.5	0	0
9.6	0	0
9.7	0	20
9.8	0	20
9.9	92.553	20
10	181.065	20
10.1	265.159	20
10.2	345.126	20
10.3	421.184	20
10.4	493.527	20
10.5	562.34	20
10.6	627.795	20

Figure79: Exported CSV file.

7.3.1 Experiment#1: varying integrator step size.

In this experiment, changed integrator step size from 100ms to 500ms and kept the other parameter constant to observe the effect of step size on OPC delay. To conduct the experiments, the OP value is changed and the delay between OP and PV values recorded. The yellow curve shows the OP of valve and the red curve shows PV. Results can be seen in detail in the below table, strip charts and CSV files exported from Historical data used to count the delay. Below, you can see strip chart for one of two experiment as an example.

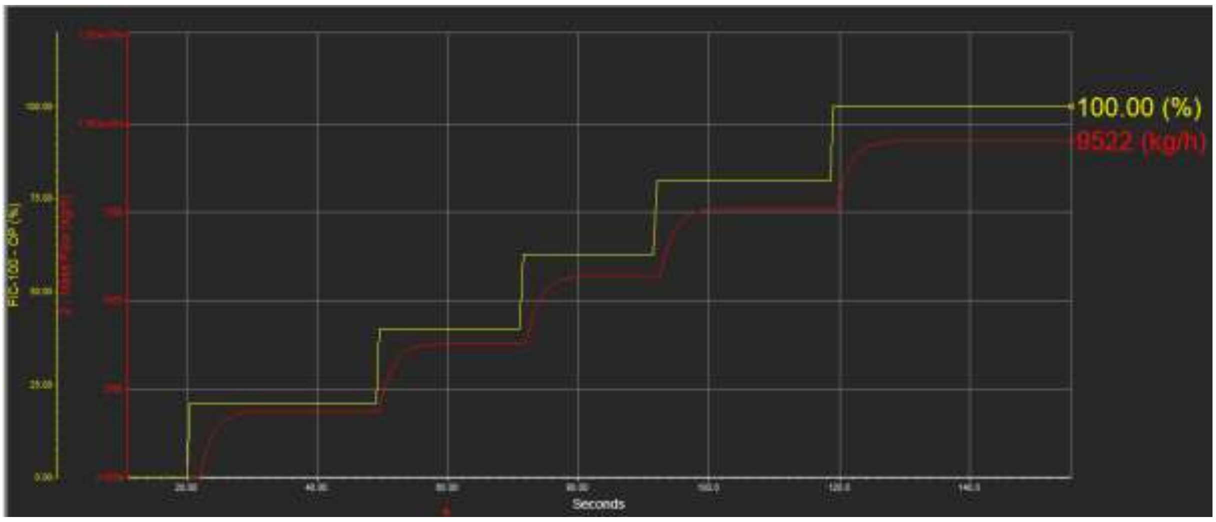


Figure80: Strip chart for Cycle Time=200ms, Step Size=500ms and Group Update Rate= 100ms.

OP	Cycle time	Group Update rate	Step size	Delay
20	200	100	100	200
40	200	100	100	200
60	200	100	100	100
80	200	100	100	300
100	200	100	100	300
			Mean	220
			standard deviation	83.67
20	200	100	500	2000
40	200	100	500	500
60	200	100	500	1000
80	200	100	500	1000
100	200	100	500	1000
			Mean	1100
			standard deviation	547.72

Observation: By increasing the integrator step size for constant cycle time and group update rate, the delay has increased.

7.3.2 Experiment#2: Varying cycle time.

In this experiment,changed cycle time from 50ms to 200ms and kept the other parameter constant to observe the effect of cycle time on OPC delay, other steps to conduct the experiment are similar to the previous one. Results can be seen in detail in the below table and for instance, you can see the strip chart for one of two experiments in below.

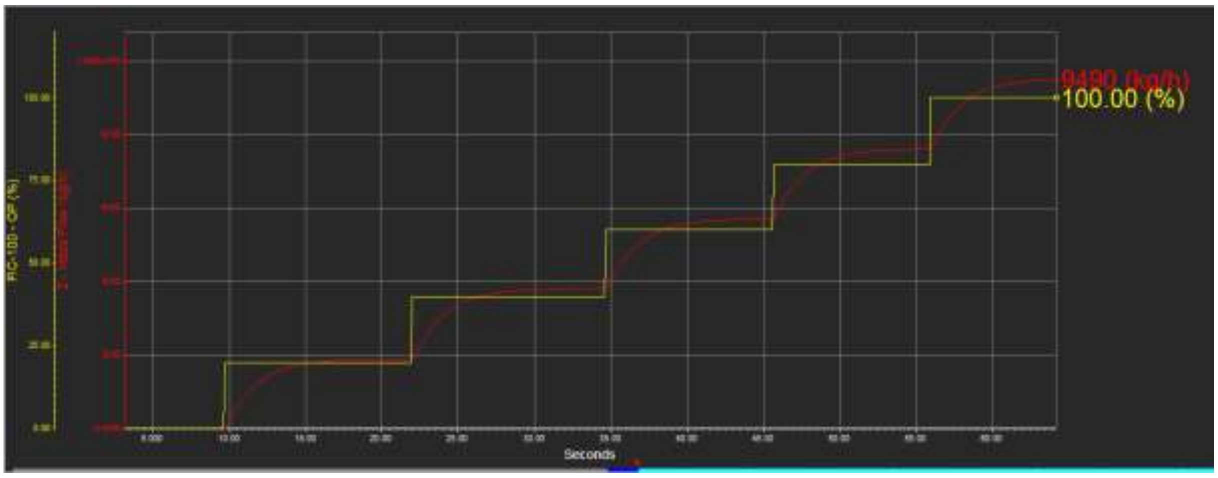


Figure81: Strip chart for Cycle Time=50ms, Step Size=100ms, Group Update Rate= 100ms.

OP	Cycle time	Group Update rate	Step size	Delay
20	200	100	100	200
40	200	100	100	200
60	200	100	100	100
80	200	100	100	300
100	200	100	100	300
			Mean	220
			standard deviation	83.67
20	50	100	100	200
40	50	100	100	200
60	50	100	100	200
80	50	100	100	200
100	50	100	100	100
			Mean	180
			standard deviation	44.72

Observation: Decrease in cycle time with constant step size and group update rate, have reduced the delay.

7.3.3 Experiment#3: varying group update rate.

In this experiment, changed group update rate to 100ms, 500ms and 1s and kept the other parameter constant to observe the effect of this parameter on communication delay, other steps to conduct the experiment are similar to the previous ones. Results can be seen in detail in the below table and you can see the strip chart for one the experiments as an example in below.

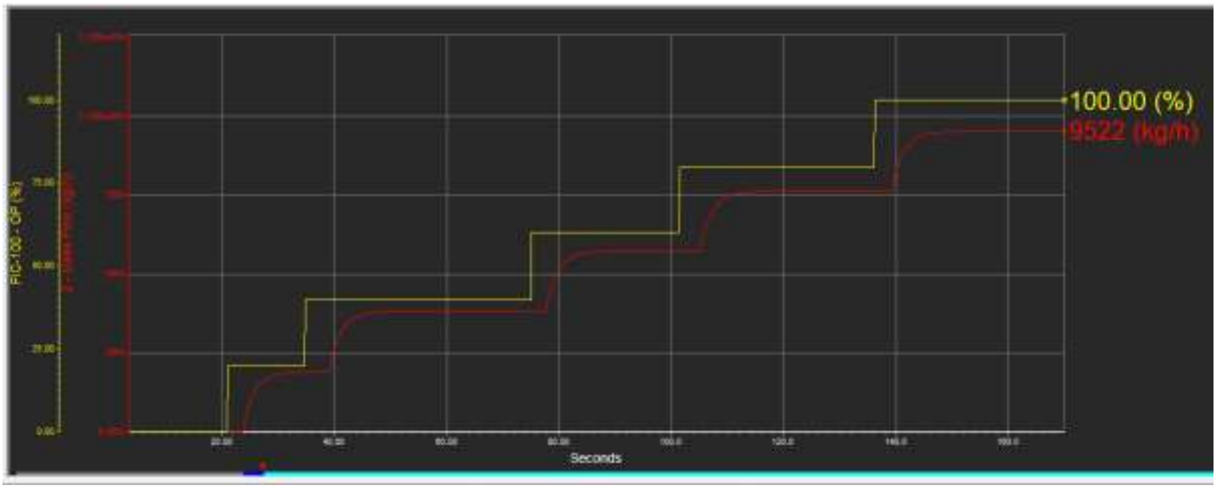


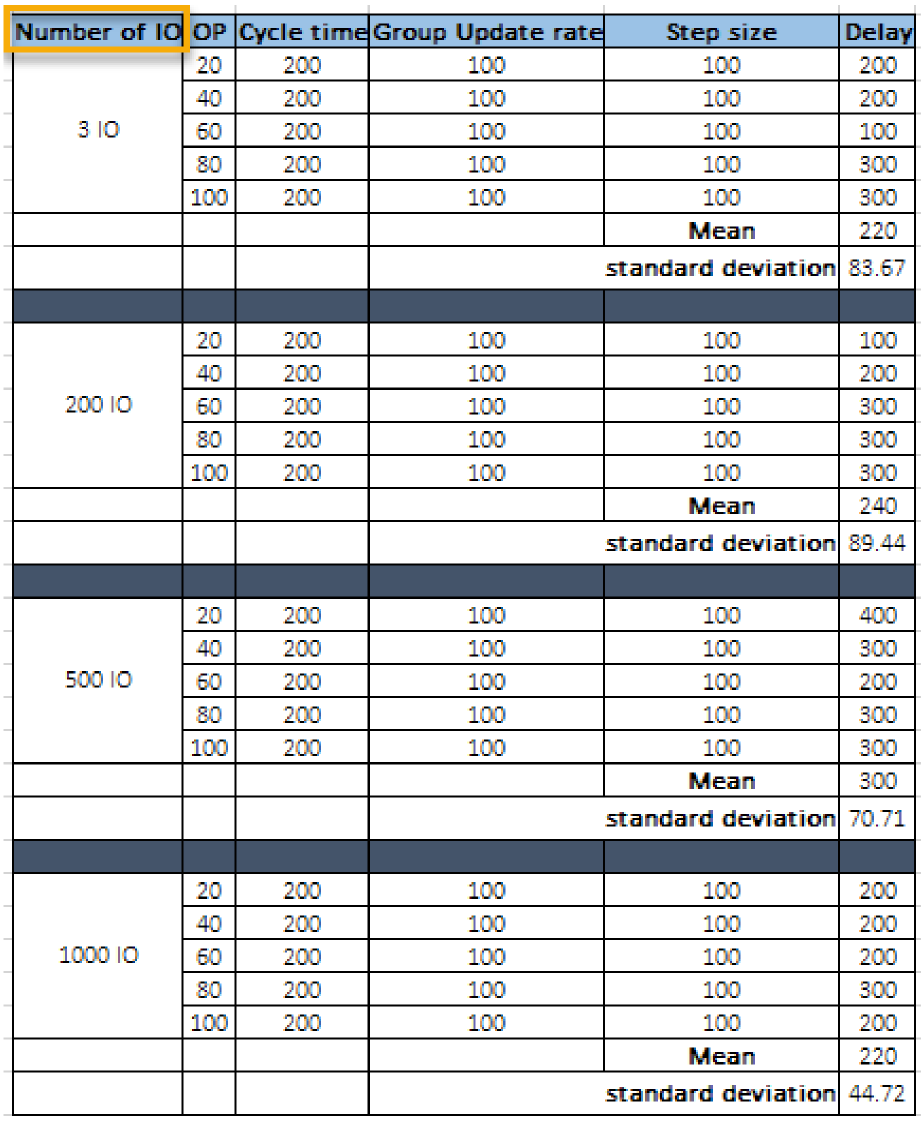
Figure82: Strip chart for Cycle Time=200ms, Step Size=100ms and Group Update Rate= 1000ms.

OP	Cycle time	Group Update rate	Step size	Total Delay
20	200	100	100	200
40	200	100	100	200
60	200	100	100	100
80	200	100	100	300
100	200	100	100	300
			Mean	220
			standard deviation	83.66600265
20	200	500	100	1800
40	200	500	100	1900
60	200	500	100	1400
80	200	500	100	1400
100	200	500	100	1400
			Mean	1580
			standard deviation	248.997992
20	200	1000	100	2700
40	200	1000	100	4300
60	200	1000	100	2600
80	200	1000	100	3700
100	200	1000	100	3200
			Mean	3300
			standard deviation	710.6335202

Observation: By increasing the group update rate for constant cycle time and integrator step size, the delay has increased significantly.

7.3.4 Experiment#4: varying number of transferred tags.

In this experiment, observed the effect of the increase in transferred tag, through OPC when the other parameters are the same, on OPC delay. Different value for OP has been forced manually and the delay between OP and PV change observed, results can be seen in detail in the below table.



Number of IO	OP	Cycle time	Group Update rate	Step size	Delay
3 IO	20	200	100	100	200
	40	200	100	100	200
	60	200	100	100	100
	80	200	100	100	300
	100	200	100	100	300
				Mean	220
				standard deviation	83.67
200 IO	20	200	100	100	100
	40	200	100	100	200
	60	200	100	100	300
	80	200	100	100	300
	100	200	100	100	300
				Mean	240
				standard deviation	89.44
500 IO	20	200	100	100	400
	40	200	100	100	300
	60	200	100	100	200
	80	200	100	100	300
	100	200	100	100	300
				Mean	300
				standard deviation	70.71
1000 IO	20	200	100	100	200
	40	200	100	100	200
	60	200	100	100	200
	80	200	100	100	300
	100	200	100	100	200
				Mean	220
				standard deviation	44.72

Observation: By adding more number of the tag into OPC client for a constant group update rate, cycle time and integrator step size, no considerable effect observed.

7.4 Conclusion:

From the results it can be seen clearly, increase in group update rate, cycle time and integrator step size will lead to a rise in the delay, however, an increase in cycle time had a slight effect on delay. On the other hand, change in group update rate made significant change on delay.

8 Summary of OPC Interface with Client and different Servers:

OPC communication let to transfer real-time data between data acquisition devices to display and interface devices. Using OPC DA there is no need to custom driver based communications which were associated with many problems such as high cost, proprietary technology that tied users to a particular vendor, hard to configure and hard to keep up-to-date. OPC DA made it possible to read and write data without the data-sink having to know the data-source's native protocol or internal data structure. It is understood that because of OPC server, client applications need not know anything about hardware protocol details. The main benefit of OPC is 'Interoperability' – Unix/Linux and Windows – both platforms are supported by OPC.

References:

1. OPC DA and UA Concepts: https://theautomization.com/opc-ua-vs-da/
2. How does OPC work for Industry 4.0? - https://www.einfochips.com/blog/implementing-opc-the-interoperability-protocol-for-embedded-automation-industry-4-0-and-the-internet-of-things/
3. CODESYS PLC Programming: SCALE_R Concept, OSCAT Library: https://store.codesys.com/oscat-basic.html?___store=en&___from_store=default
4. CODESYS OPC DA SERVER: https://store.codesys.com/codesys-opc-da-server-sl-bundle.html?___store=en
5. DCOM Configuration: https://www.aggsoft.com/asdl-dcom-opc-config-3.html
6. MATRIKON OPC Explorer and Server: https://www.matrikonopc.com/training/opc-demo-tutorial.pdf
7. HI_801_480_E_X-OPC_Server_Manual
8. HI_801_296_E_HIMax_OTS_Manual