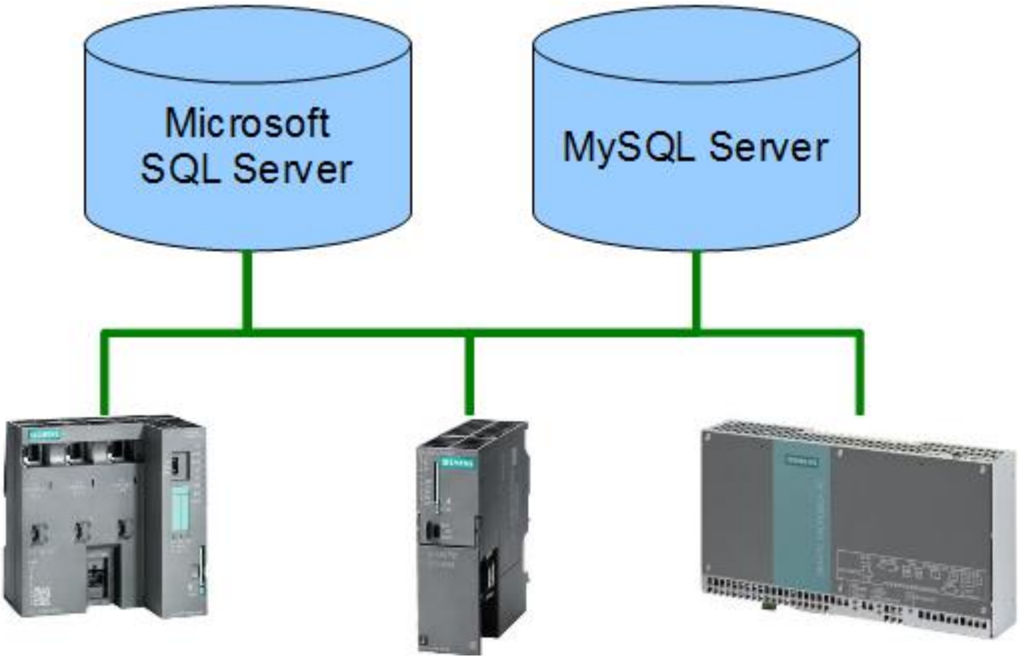
Installation Manual

MS SQL for PLCSQL link

SQL Client in a
Siemens S7 PLC



Revision date: 2018-09-13
Revised by: Anders Jorsal, Alsmatik A/S
Version: 2014-5w
Revision date: 2019-03-12
Revised by: FBH, Alsmatik A/S
Version: 2014-6w

Contents

Introduction

In order to get it to work, it is important to setup the database, to fit with the design for PLCSQL link.

This manual is for Microsoft SQL Server 2014, but can also be used as guidance for MS SQL 2012 and MS SQL 2016, SQL 2016 is only running on Windows 10.

For guidance in setting up PLCSQL link PLC program, we refer to the document “PLCSQL PLC Installation Manual”.

If you still have questions after reading this manual, please send them to info@plcsql.com

How does the system work?

We have tried to make a system where it is possible to communicate with a SQL data base without being an “SQL expert”, nor being an “PLC expert” regarding communication etc. the system contains a “standard” PLC program, and a “standard” SQL data base that “fits” together.

In the PLC we are using the following basic tag types:

Bool.	Is stored in an “Bool” table in the database.
Int.	Is stored in an “Int” table in the database.
Dint.	Is stored in an “Dint” table in the database.
Real	Is stored in an “Real” table in the database.
String	Is stored in an “String” table in the database.

To distinguish between the different tags, every tag has a specific number.
In the PLC there is an “Array” that contains all the tags, and in the database the different tables contain the corresponding data types and numbers as in the PLC, so you have complete control with the tags.

Now, somebody will ask, “but we have to connect an existing data base”, yes that is no problem, you can easily interconnect between different data bases, so we strongly recommend to “make” a data base exclusively for PLCSQL, so you easily can check where the problem could be when something is not working.

On the following pages there is a schematic view of the layout and the possibilities you have with the PLCSQL system.

⚠

Please note the following.

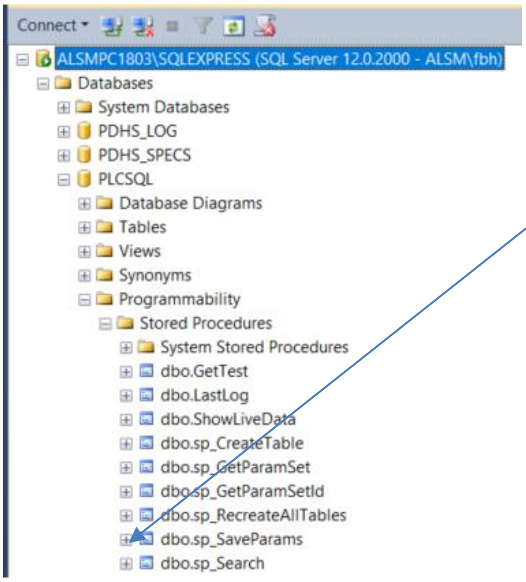
Parameter 10001, 15001, and 30001 is used internally in the “Log” parameters and in the “Recipe” parameters. DON’T write to these parameters.

10001:	SetCount
15001:	SetID
30001:	DateTimeStamp.

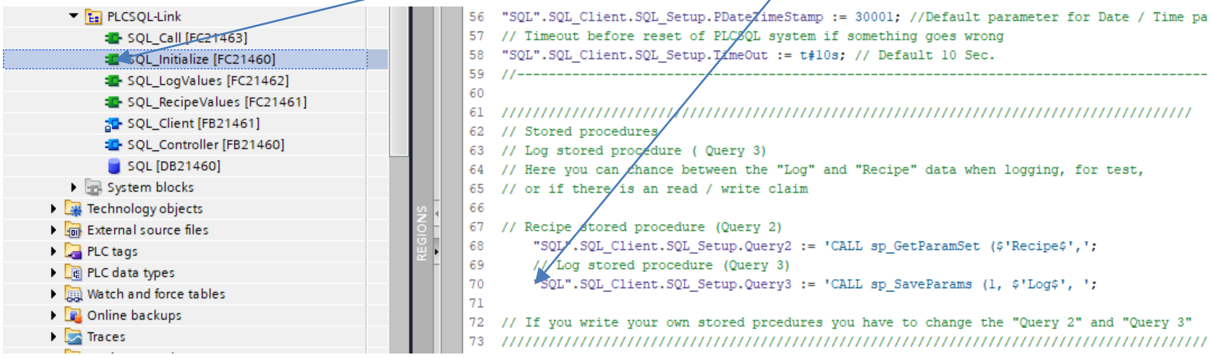
How does PLCSQL work, Log, write to SQL server

In Mssql we want to store the Value 2.3009 in the ParamID[1].

First we need to setup the Call in the PLC for the stored procedure in Mssql.

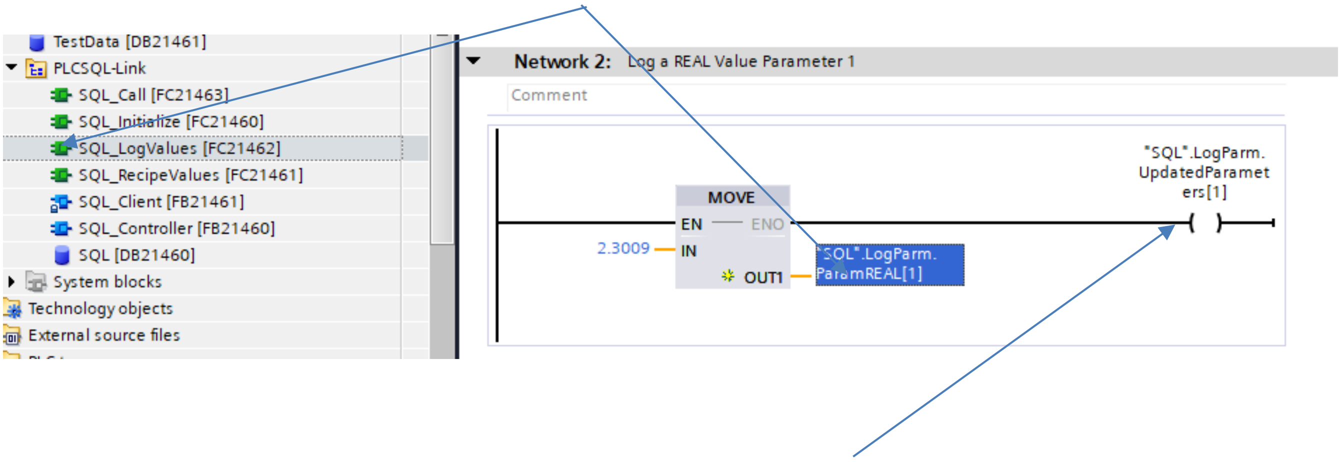


In SQL_Initialize DB you setup the SQL.SQL_Client.SQL_Setup.Query3 to match the procedure in Mssql: "SQL".SQL_Client.SQL_Setup.Query3 := 'CALL sp_SaveParams (1, $'Log$', ';



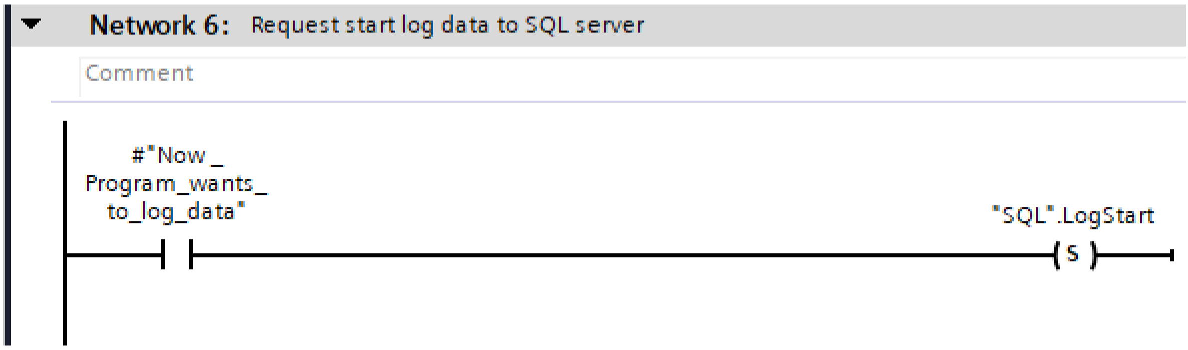
```
"SQL".SQL_Client.SQL_Setup.PDateTimeStamp := 30001; //Default parameter for Date / Time pa
// Timeout before reset of PLCSQL system if something goes wrong
"SQL".SQL_Client.SQL_Setup.TimeOut := t#10s; // Default 10 Sec.
//------------------------------------------------------------------------------------------

////////////////////////////////////////////////////////////////////////////////////////
// Stored procedures
// Log stored procedure ( Query 3)
// Here you can chance between the "Log" and "Recipe" data when logging, for test,
// or if there is an read / write claim

// Recipe stored procedure (Query 2)
    "SQL".SQL_Client.SQL_Setup.Query2 := 'CALL sp_GetParamSet ($'Recipe$',';
    // Log stored procedure (Query 3)
    "SQL".SQL_Client.SQL_Setup.Query3 := 'CALL sp_SaveParams (1, $'Log$', ';

// If you write your own stored prcedures you have to change the "Query 2" and "Query 3"
/////////////////////////////////////////////////////////////////////////////////////////
```

In the PLC we move the value 2.3009 to the SQL DB

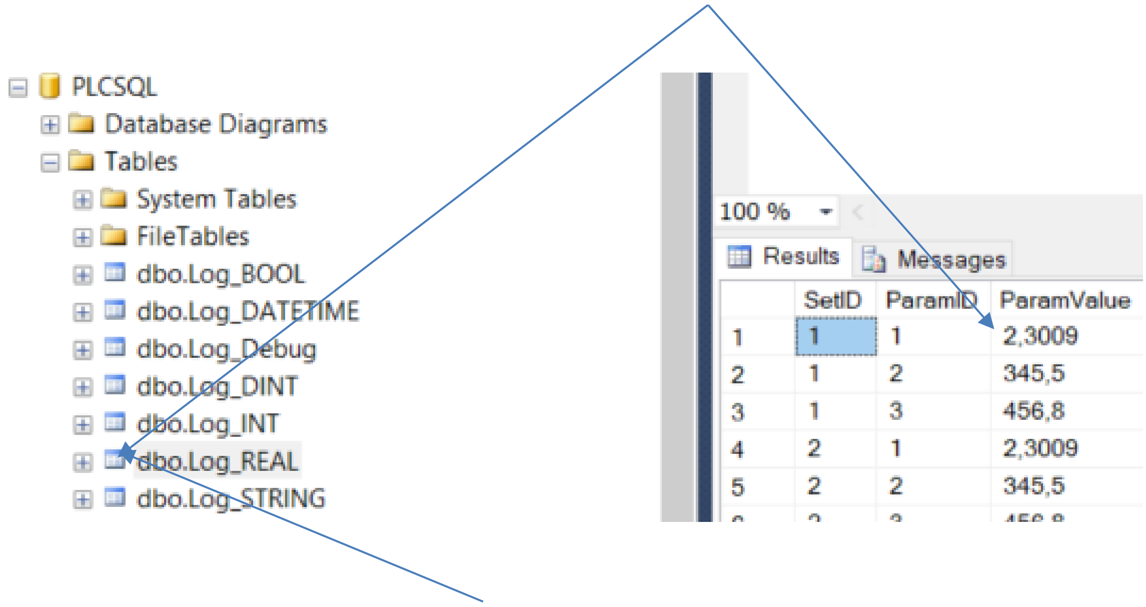


And we tell the system that there is an updated value on ParamID[1].

Last we set the bit SQL.LogStart and the value is stored in mssql.


Network 6: Request start log data to SQL server
Comment
#"Now_Program_wants_to_log_data"
"SQL".LogStart
S

In Mssql we can see that the Value 2.3009 is stored in ParamID[1].

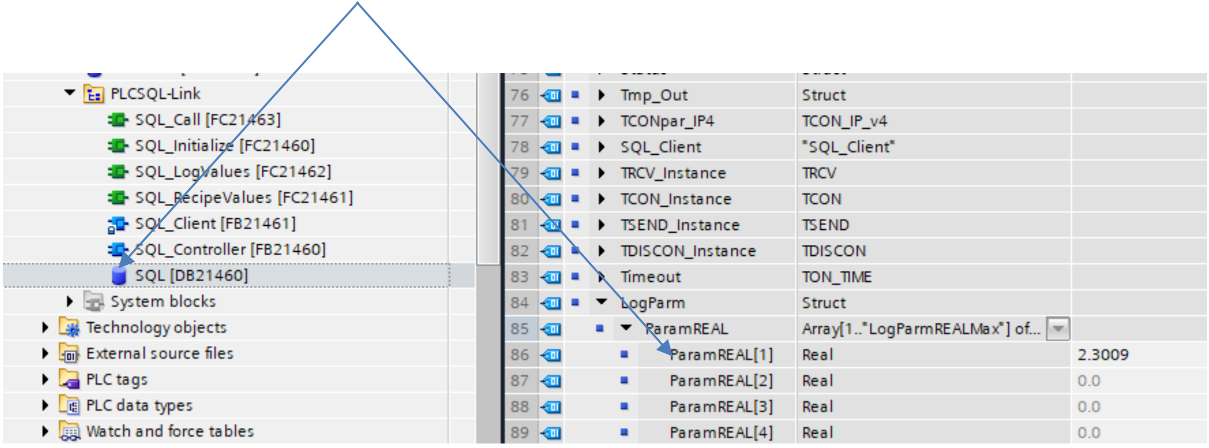


- PLCSQL
  - Database Diagrams
  - Tables
    - System Tables
    - FileTables
    - dbo.Log_BOOL
    - dbo.Log_DATETIME
    - dbo.Log_Debug
    - dbo.Log_DINT
    - dbo.Log_INT
    - dbo.Log_REAL
    - dbo.Log_STRING

	SetID	ParamID	ParamValue
1	1	1	2,3009
2	1	2	345,5
3	1	3	456,8
4	2	1	2,3009
5	2	2	345,5
6	2	3	456,8

Right click on dbo.Log_REAL and select top 1000 rows to see the view.

ParamId[1] is defined as a REAL type in both the PLC and Mssql.



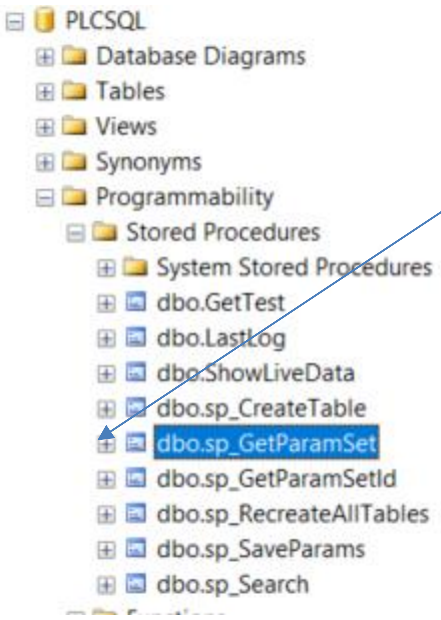
- PLCSQL-Link
  - SQL_Call [FC21463]
  - SQL_Initialize [FC21460]
  - SQL_LogValues [FC21462]
  - SQL_RecipeValues [FC21461]
  - SQL_Client [FB21461]
  - SQL_Controller [FB21460]
  - SQL [DB21460]
- System blocks
- Technology objects
- External source files
- PLC tags
- PLC data types
- Watch and force tables

76	Tmp_Out	Struct	
77	TCONpar_IP4	TCON_IP_v4	
78	SQL_Client	"SQL_Client"	
79	TRCV_Instance	TRCV	
80	TCON_Instance	TCON	
81	TSEND_Instance	TSEND	
82	TDISCON_Instance	TDISCON	
83	Timeout	TON_TIME	
84	LogParm	Struct	
85	ParamREAL	Array[1.."LogParmREALMax"] of...	
86	ParamREAL[1]	Real	2.3009
87	ParamREAL[2]	Real	0.0
88	ParamREAL[3]	Real	0.0
89	ParamREAL[4]	Real	0.0

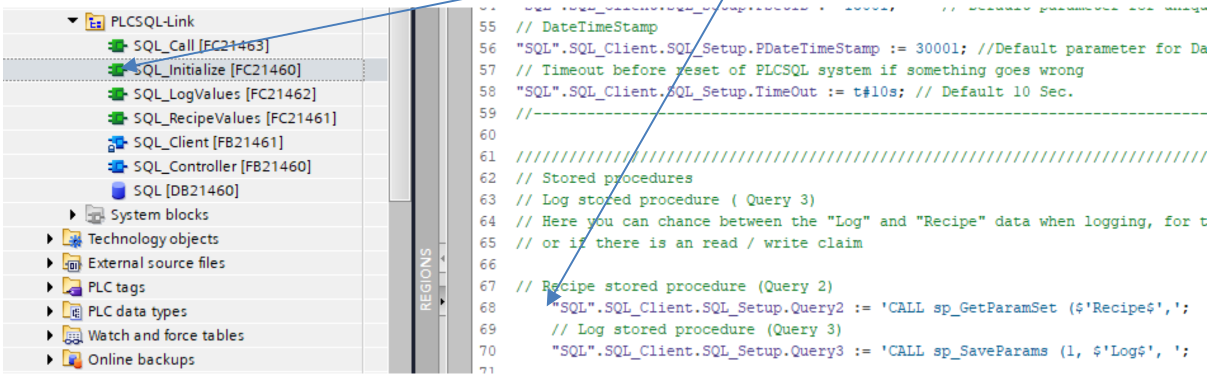
How does PLCSQL work, Recipe, read from SQL server

In Mssql we want to read the stored value in ParamID[3].

First we need to setup the Call in the PLC for the stored procedure in Mssql.

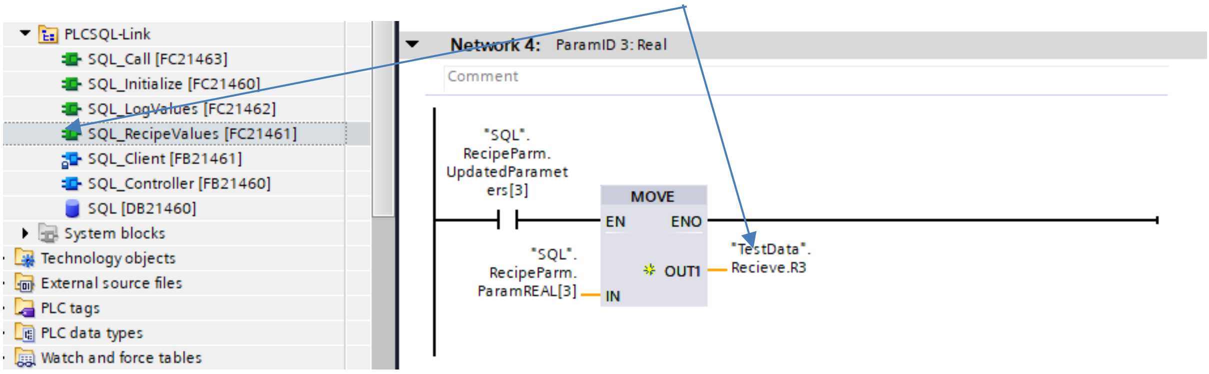


SQL_Initialize DB you setup the SQL.SQL_Client.SQL_Setup.Query2 to match the procedure in Mssql: "SQL".SQL_Client.SQL_Setup.Query2 := 'CALL sp_GetParamSet ($'Recipe$',';

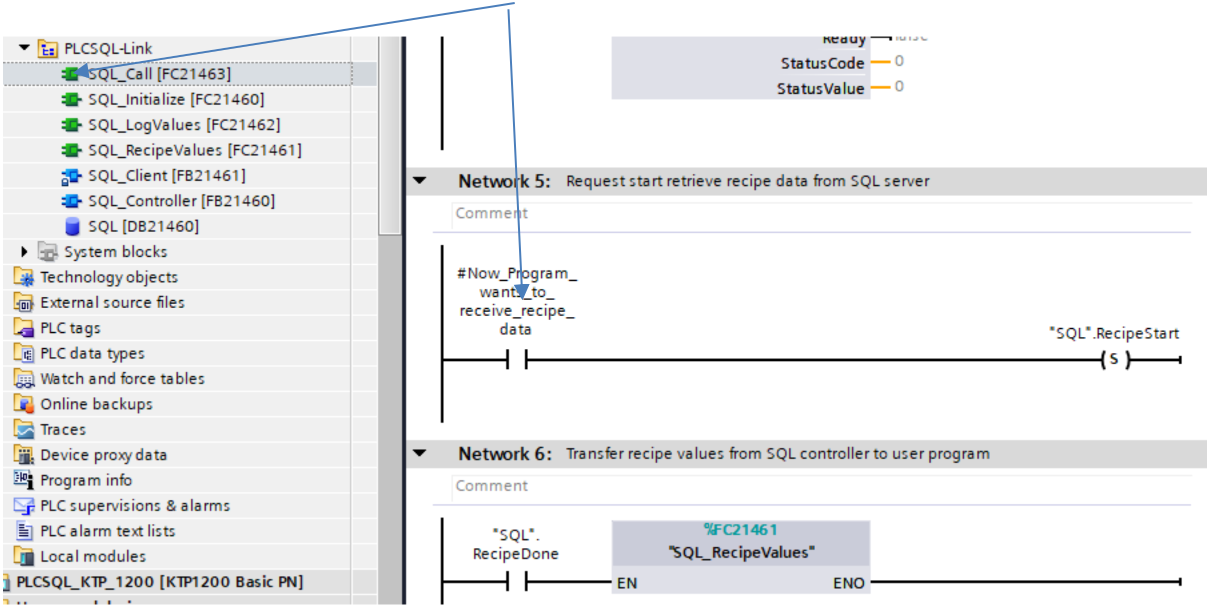


```
// DateTimeStamp
"SQL".SQL_Client.SQL_Setup.PDateTimeStamp := 30001; //Default parameter for Da
// Timeout before reset of PLCSQL system if something goes wrong
"SQL".SQL_Client.SQL_Setup.TimeOut := t#10s; // Default 10 Sec.
//---------------------------------------------------------------------------

////////////////////////////////////////////////////////////////////////////
// Stored procedures
// Log stored procedure ( Query 3)
// Here you can chance between the "Log" and "Recipe" data when logging, for t
// or if there is an read / write claim

// Recipe stored procedure (Query 2)
    "SQL".SQL_Client.SQL_Setup.Query2 := 'CALL sp_GetParamSet ($'Recipe$',';
    // Log stored procedure (Query 3)
    "SQL".SQL_Client.SQL_Setup.Query3 := 'CALL sp_SaveParams (1, $'Log$', ';
```

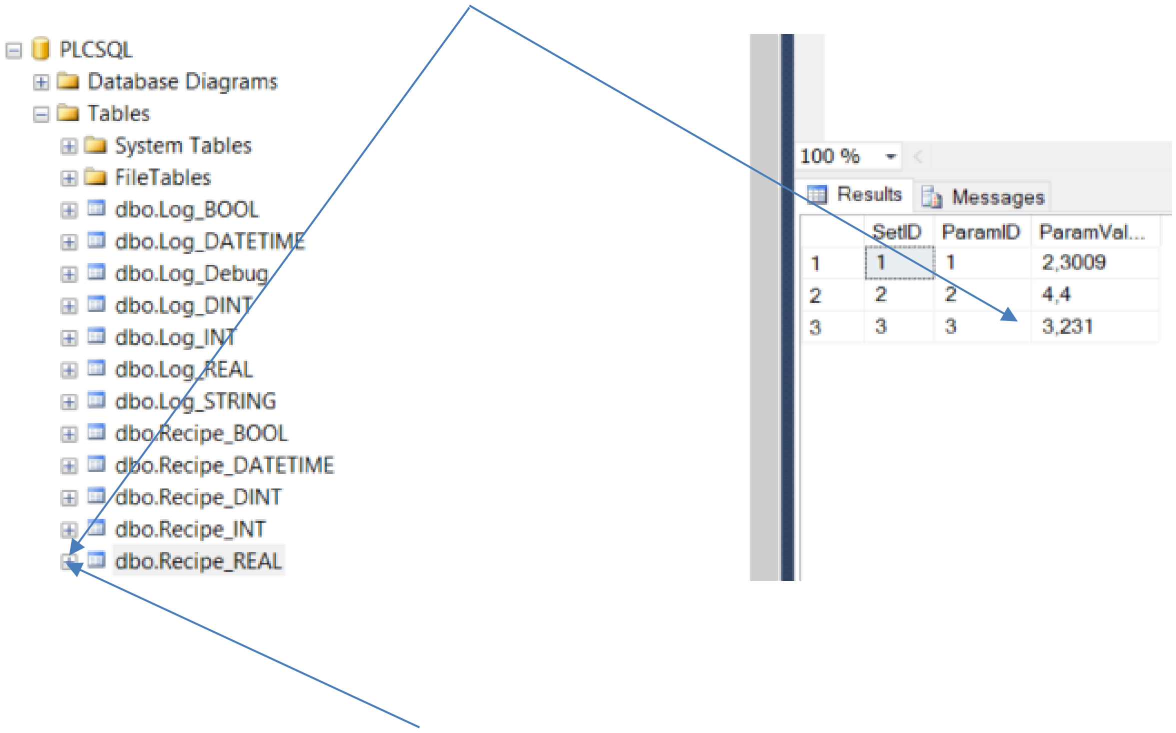
Move the received data from Mssql to your variable in the PLC.



In the Plc we need to trigger the start log bit.



In Mssql we want to read a Real value with ParamID 3 from Recipe

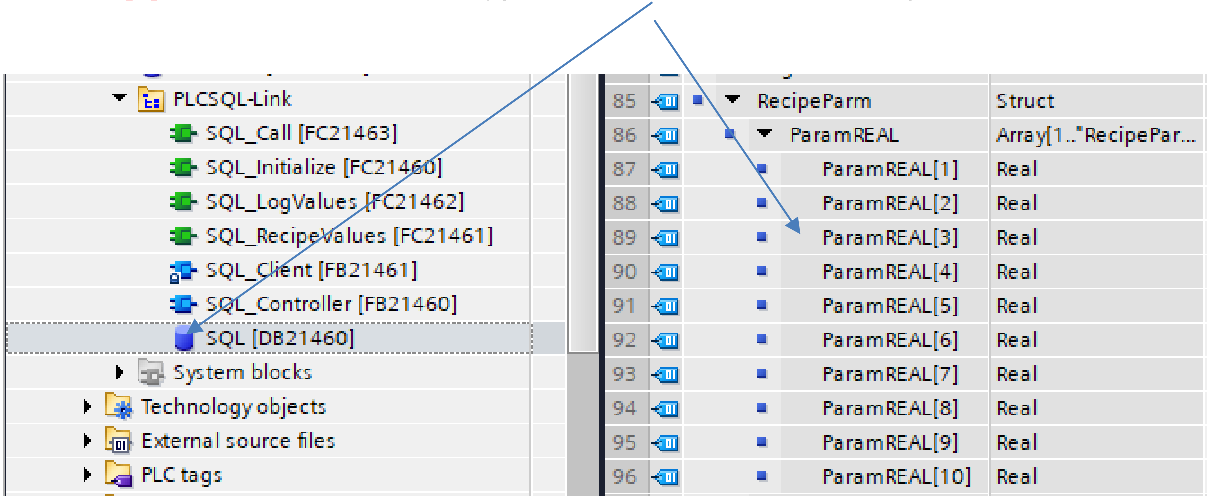


- PLCSQL
  - Database Diagrams
  - Tables
    - System Tables
    - FileTables
    - dbo.Log_BOOL
    - dbo.Log_DATETIME
    - dbo.Log_Debug
    - dbo.Log_DINT
    - dbo.Log_INT
    - dbo.Log_REAL
    - dbo.Log_STRING
    - dbo.Recipe_BOOL
    - dbo.Recipe_DATETIME
    - dbo.Recipe_DINT
    - dbo.Recipe_INT
    - dbo.Recipe_REAL

	SetID	ParamID	ParamVal...
1	1	1	2,3009
2	2	2	4,4
3	3	3	3,231

Right click on dbo.Log_REAL and select Top 1000 rows to see the view.

ParamId[3] is defined as a REAL type in both the PLC and Mssql.



- PLCSQL-Link
  - SQL_Call [FC21463]
  - SQL_Initialize [FC21460]
  - SQL_LogValues [FC21462]
  - SQL_RecipeValues [FC21461]
  - SQL_Client [FB21461]
  - SQL_Controller [FB21460]
  - SQL [DB21460]
- System blocks
- Technology objects
- External source files
- PLC tags

85	RecipeParm	Struct
86	ParamREAL	Array[1.."RecipePar...
87	ParamREAL[1]	Real
88	ParamREAL[2]	Real
89	ParamREAL[3]	Real
90	ParamREAL[4]	Real
91	ParamREAL[5]	Real
92	ParamREAL[6]	Real
93	ParamREAL[7]	Real
94	ParamREAL[8]	Real
95	ParamREAL[9]	Real
96	ParamREAL[10]	Real

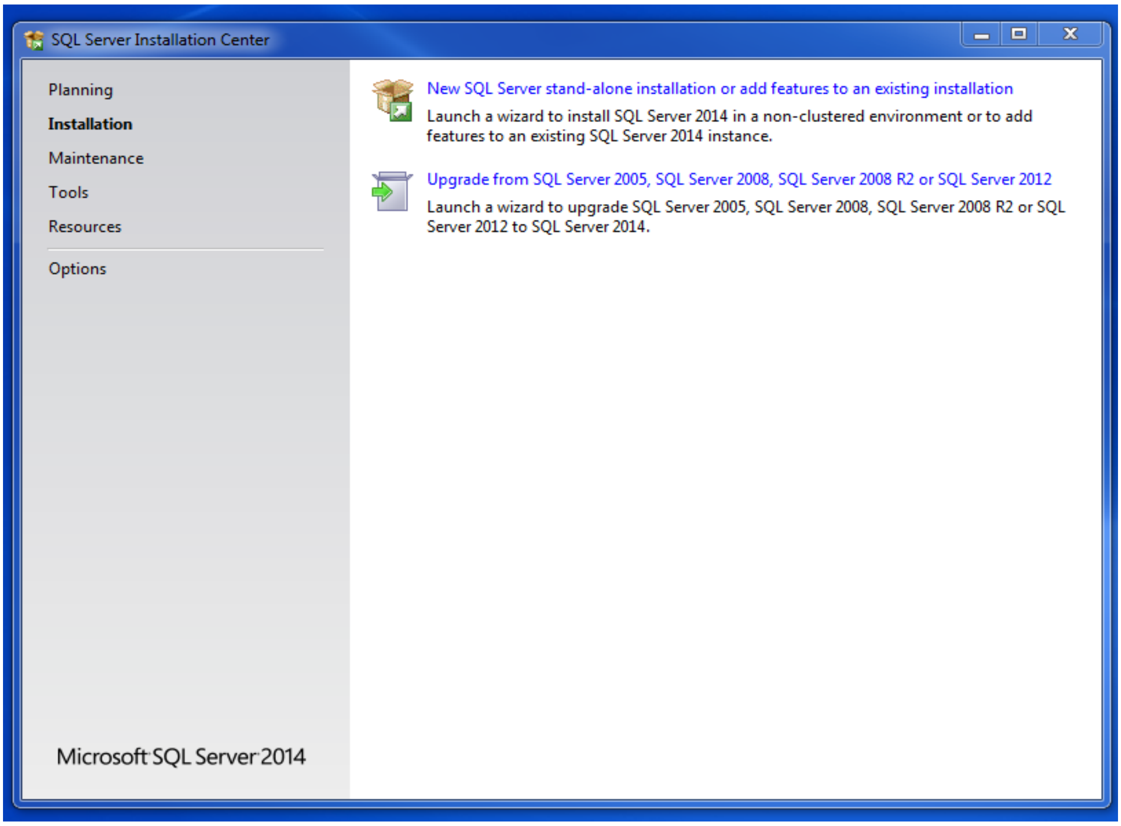
Example software

In this example we are using the program “Microsoft SQL Server Express” Ver. 2014. Installed on “Windows 7 Professional 64 bit”.

With this installation manual you are able to setup the database server, for getting and putting data by the PLCSQL link.

Install MS SQL Server

Install the software MS SQL server Express.
With the file: SQLEXPRADV_x64_ENU.exe, the software includes all the necessary tools you need.
Follow the installation guide of the software.



SQL Server Installation Center

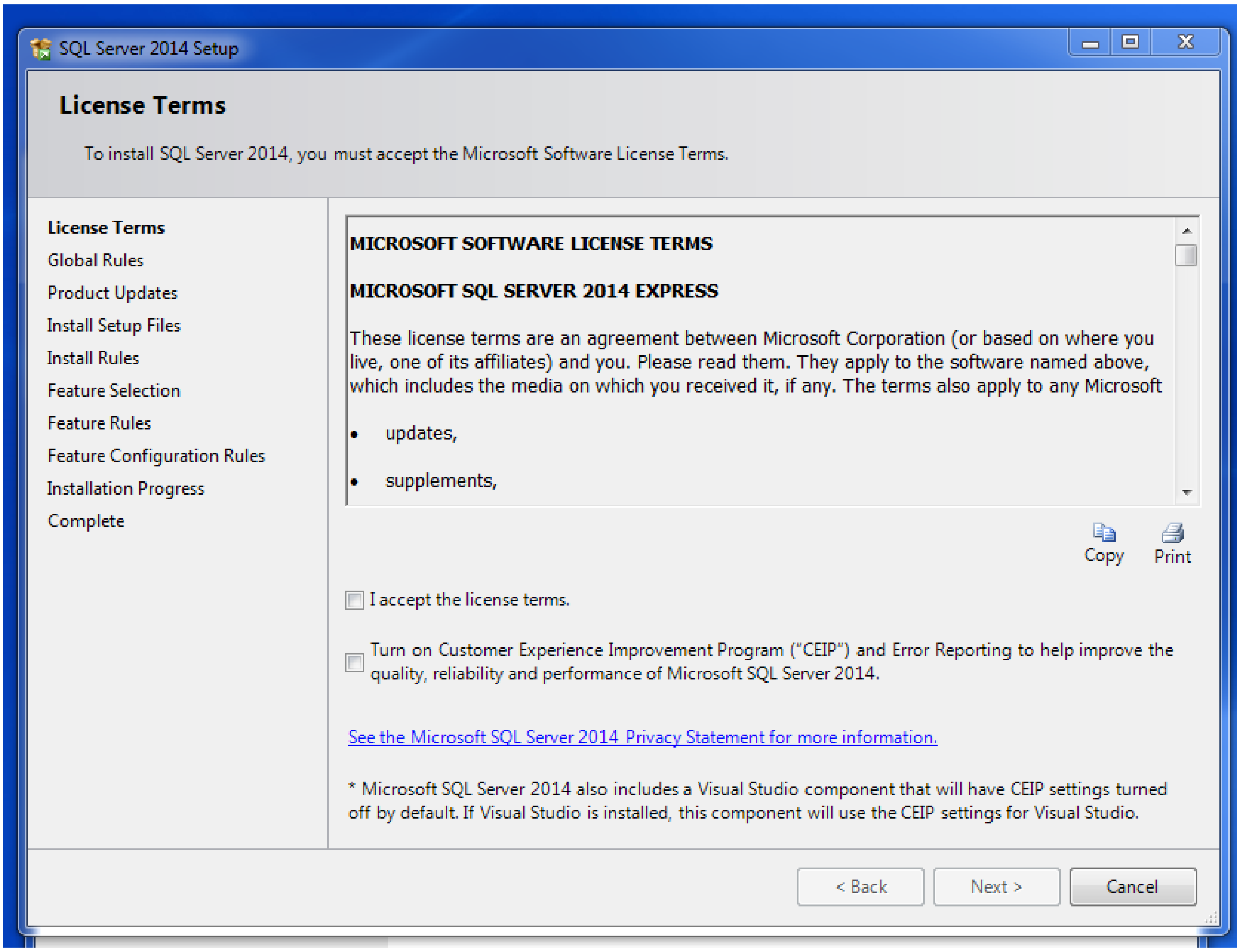
New SQL Server stand-alone installation or add features to an existing installation

Launch a wizard to install SQL Server 2014 in a non-clustered environment or to add features to an existing SQL Server 2014 instance.

Upgrade from SQL Server 2005, SQL Server 2008, SQL Server 2008 R2 or SQL Server 2012

Launch a wizard to upgrade SQL Server 2005, SQL Server 2008, SQL Server 2008 R2 or SQL Server 2012 to SQL Server 2014.

Select “New SQL Server ……….”



To install SQL Server 2014, you must accept the Microsoft Software License Terms.

MICROSOFT SOFTWARE LICENSE TERMS

MICROSOFT SQL SERVER 2014 EXPRESS

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to the software named above, which includes the media on which you received it, if any. The terms also apply to any Microsoft

- updates,
- supplements,

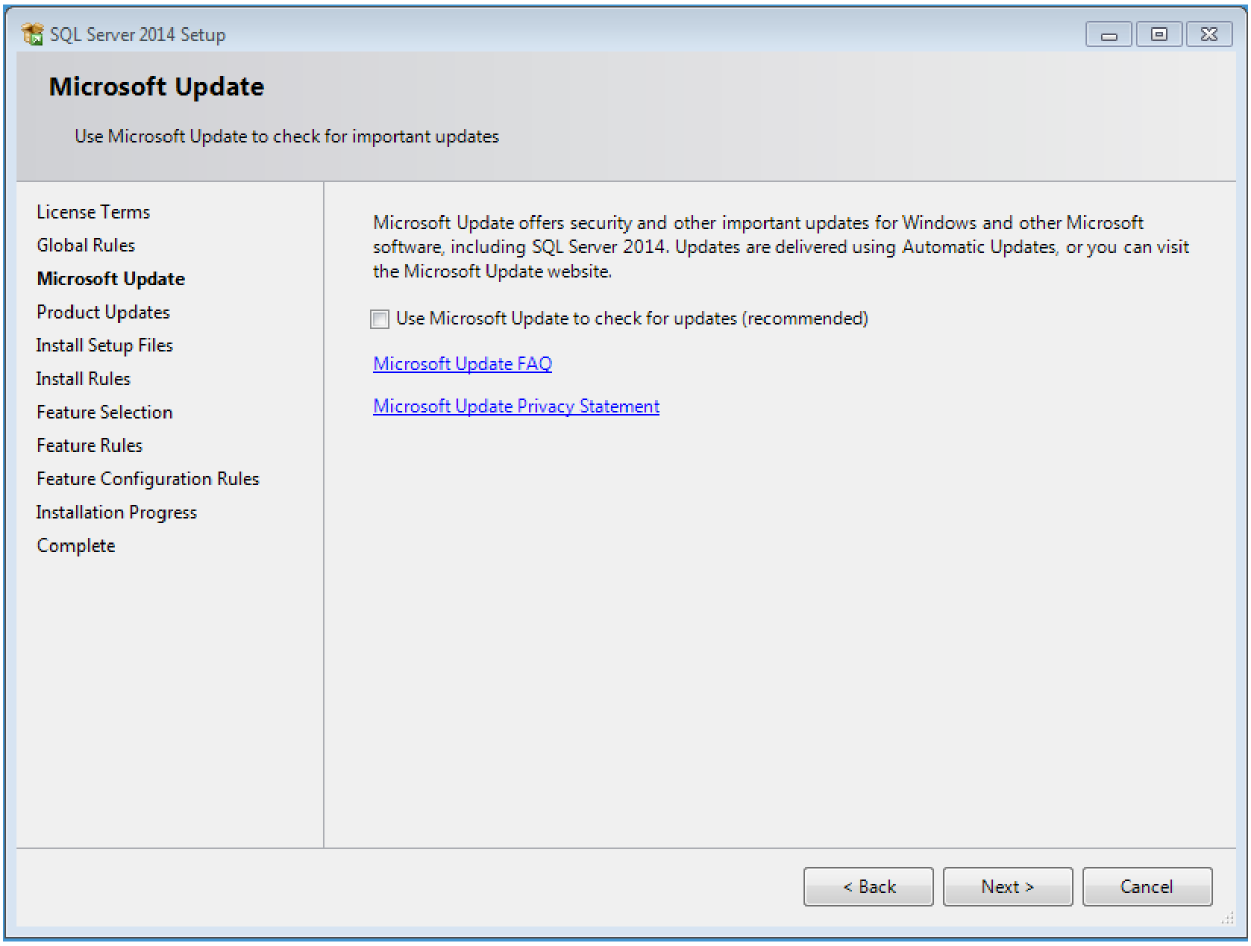
□ I accept the license terms.

□ Turn on Customer Experience Improvement Program ("CEIP") and Error Reporting to help improve the quality, reliability and performance of Microsoft SQL Server 2014.

See the Microsoft SQL Server 2014 Privacy Statement for more information.

* Microsoft SQL Server 2014 also includes a Visual Studio component that will have CEIP settings turned off by default. If Visual Studio is installed, this component will use the CEIP settings for Visual Studio.

Accept license terms.



Microsoft Update

Use Microsoft Update to check for important updates

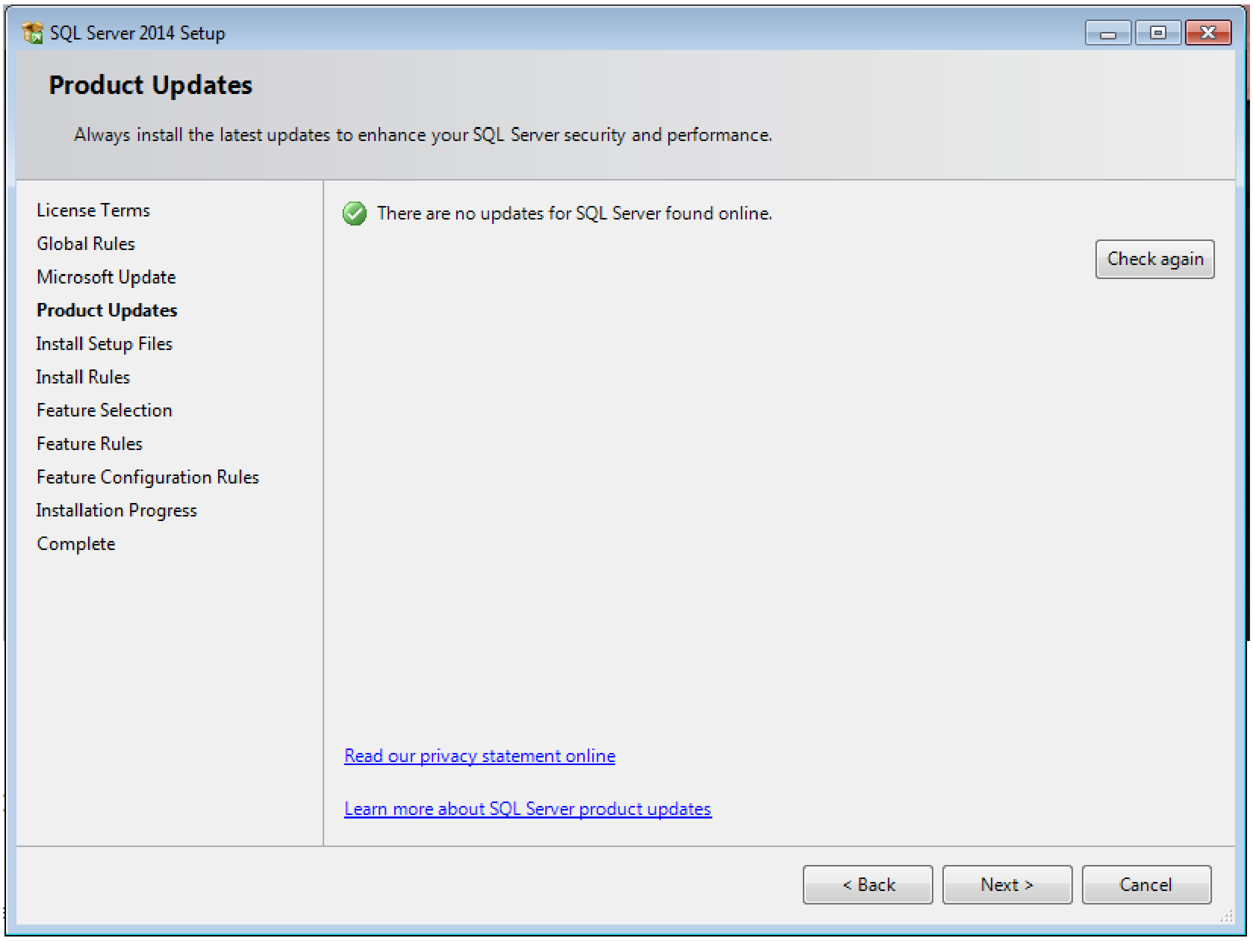
Microsoft Update offers security and other important updates for Windows and other Microsoft software, including SQL Server 2014. Updates are delivered using Automatic Updates, or you can visit the Microsoft Update website.

□ Use Microsoft Update to check for updates (recommended)

Microsoft Update FAQ

Microsoft Update Privacy Statement

Click “Next” (If this picture is displayed)



SQL Server 2014 Setup

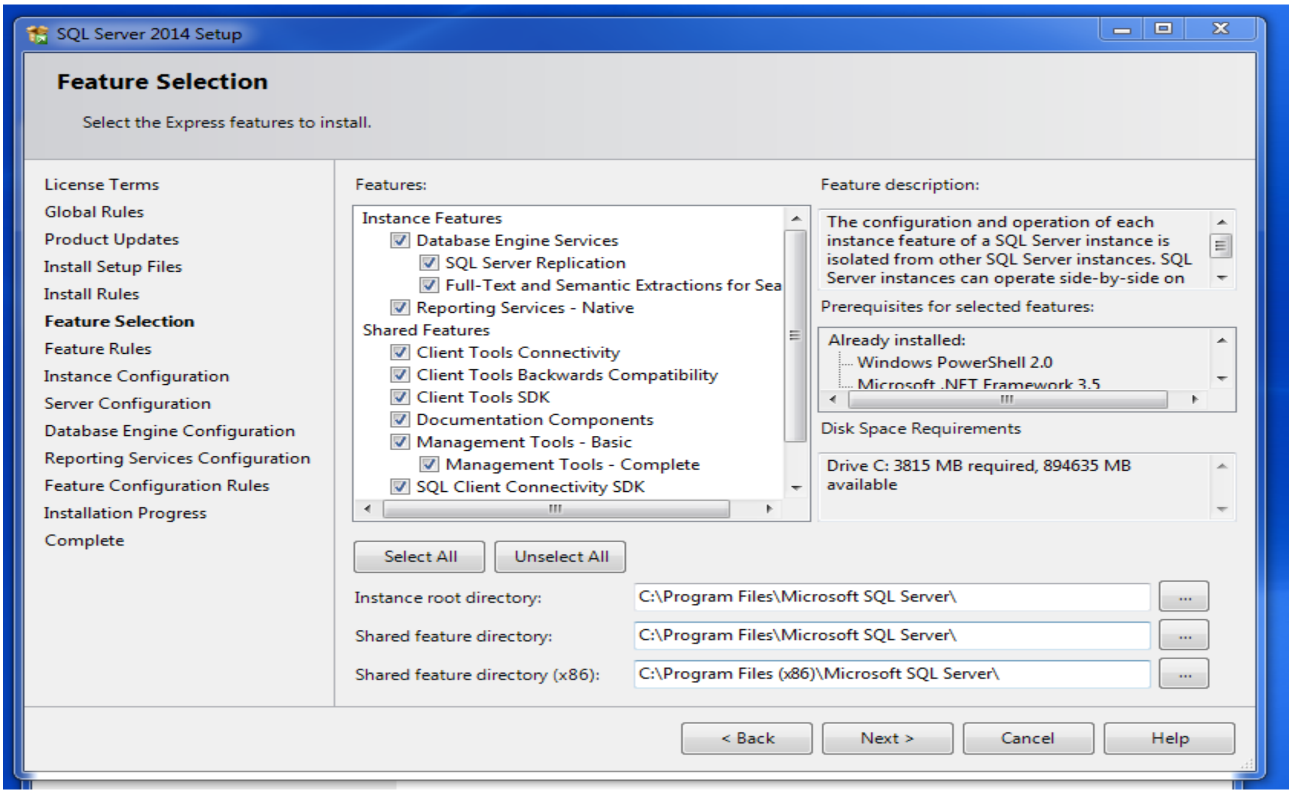
Product Updates

Always install the latest updates to enhance your SQL Server security and performance.

There are no updates for SQL Server found online.

Check again
Read our privacy statement online
Learn more about SQL Server product updates
< Back
Next >
Cancel

Click “Next” (If this picture is displayed)



SQL Server 2014 Setup

Feature Selection

Select the Express features to install.

Features:

- Instance Features
  - ☑ Database Engine Services
    - ☑ SQL Server Replication
    - ☑ Full-Text and Semantic Extractions for Sea
  - ☑ Reporting Services - Native
- Shared Features
  - ☑ Client Tools Connectivity
  - ☑ Client Tools Backwards Compatibility
  - ☑ Client Tools SDK
  - ☑ Documentation Components
  - ☑ Management Tools - Basic
    - ☑ Management Tools - Complete
  - ☑ SQL Client Connectivity SDK

Feature description:

The configuration and operation of each instance feature of a SQL Server instance is isolated from other SQL Server instances. SQL Server instances can operate side-by-side on

Prerequisites for selected features:

Already installed:
Windows PowerShell 2.0
Microsoft .NET Framework 3.5

Disk Space Requirements

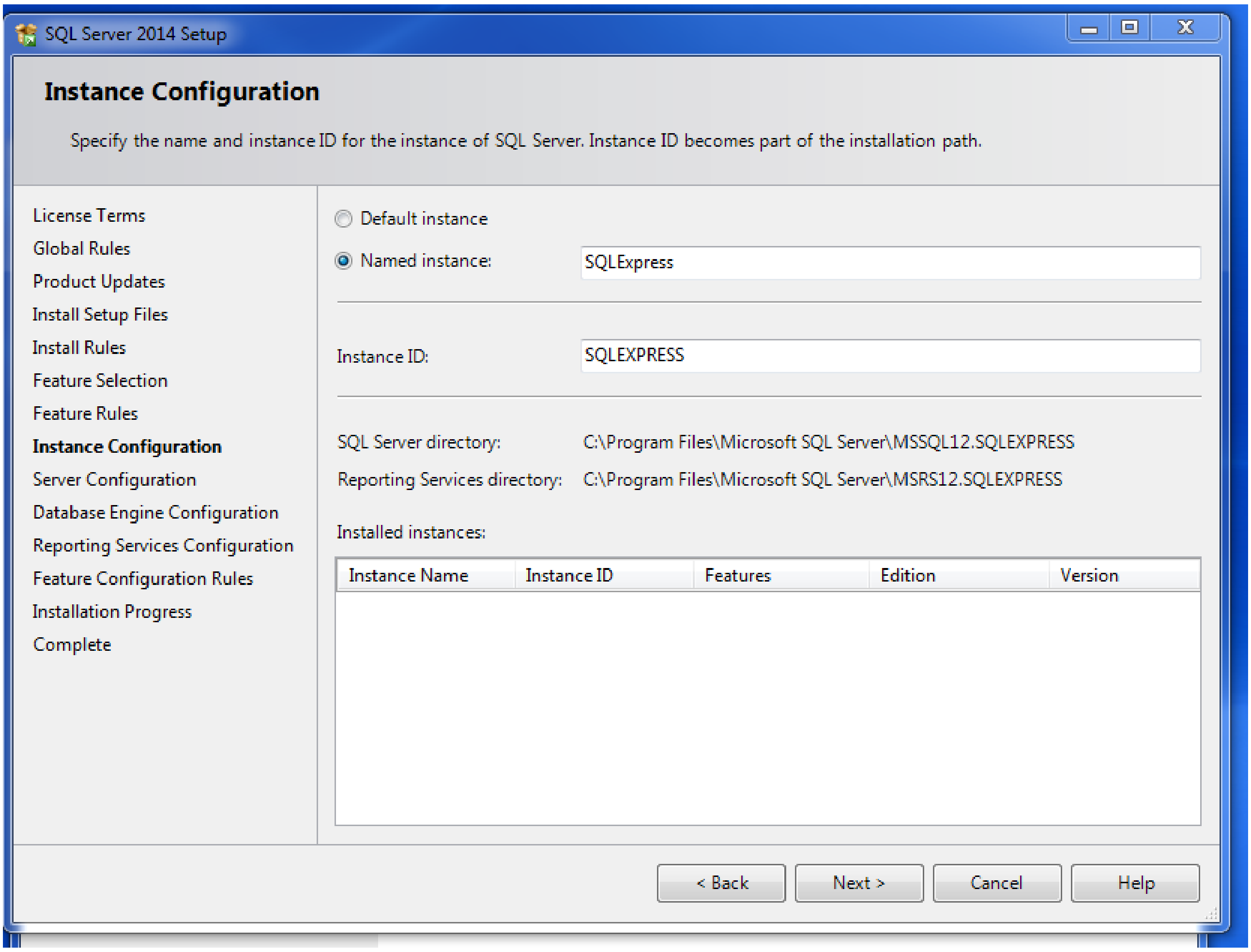
Drive C: 3815 MB required, 894635 MB available

Instance root directory: C:\Program Files\Microsoft SQL Server\

Shared feature directory: C:\Program Files\Microsoft SQL Server\

Shared feature directory (x86): C:\Program Files (x86)\Microsoft SQL Server\

Default selection, just click “Next”



Instance Configuration

Specify the name and instance ID for the instance of SQL Server. Instance ID becomes part of the installation path.

○ Default instance

● Named instance: SQLExpress

Instance ID: SQLEXPRESS

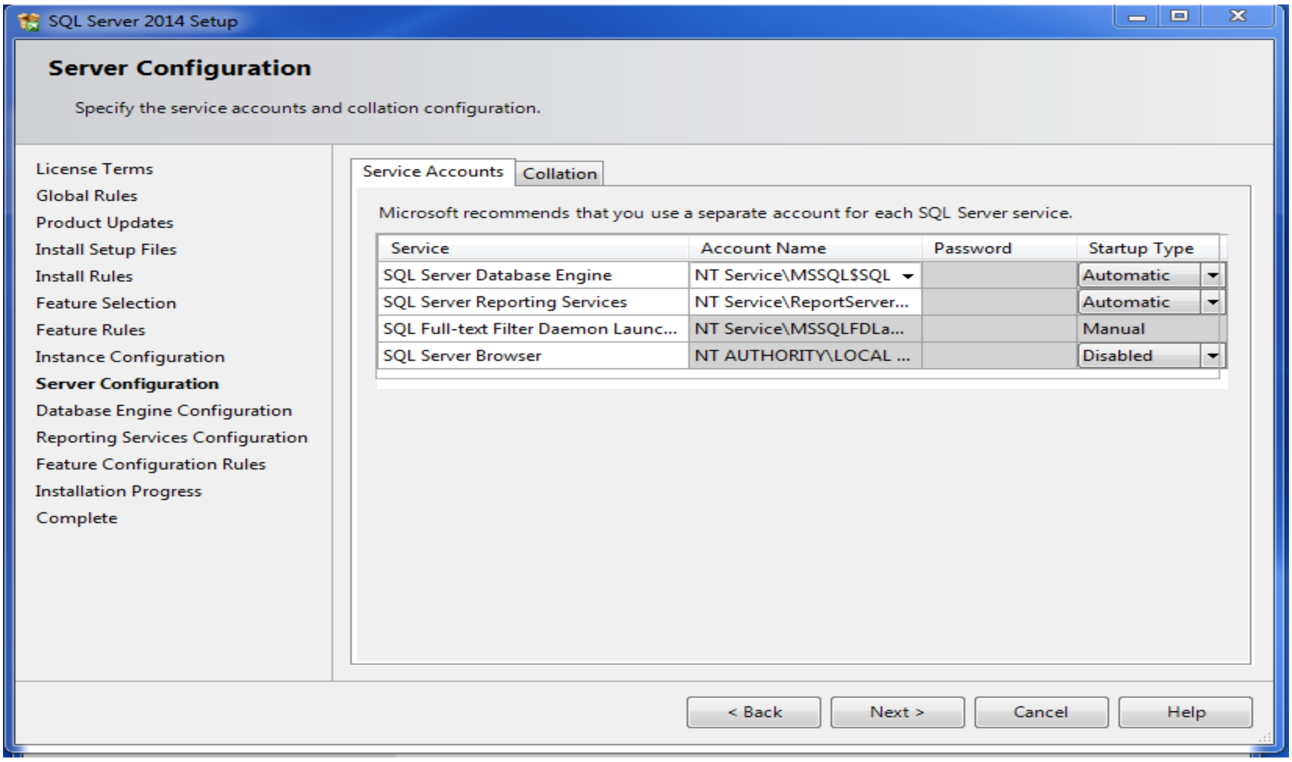
SQL Server directory: C:\Program Files\Microsoft SQL Server\MSSQL12.SQLEXPRESS

Reporting Services directory: C:\Program Files\Microsoft SQL Server\MSRS12.SQLEXPRESS

Installed instances:

Instance Name	Instance ID	Features	Edition	Version

Keep the name, or make your own name, or just select “Default instance”



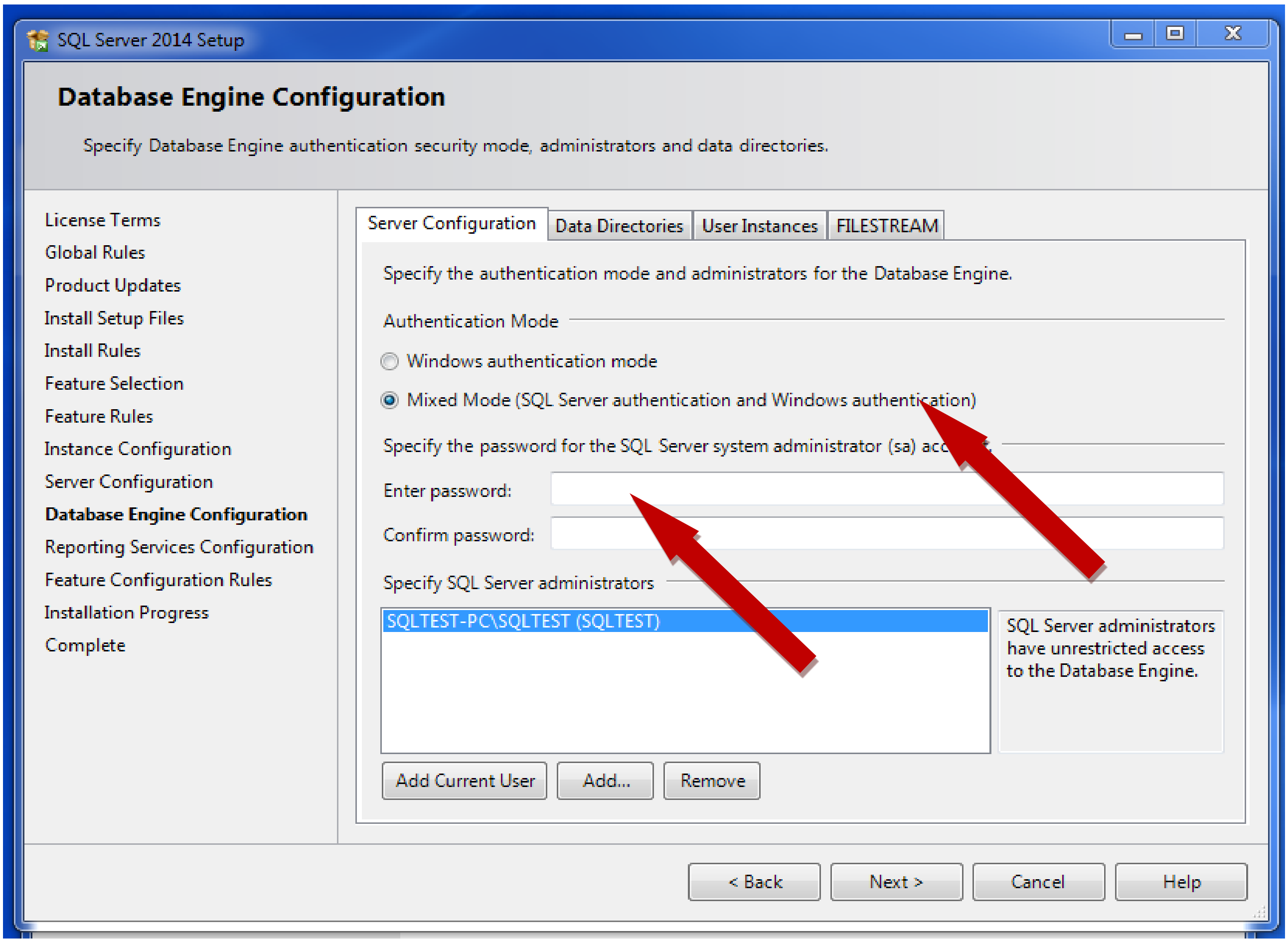
Server Configuration

Specify the service accounts and collation configuration.

Microsoft recommends that you use a separate account for each SQL Server service.

Service	Account Name	Password	Startup Type
SQL Server Database Engine	NT Service\MSSQL$SQL		Automatic
SQL Server Reporting Services	NT Service\ReportServer...		Automatic
SQL Full-text Filter Daemon Launc...	NT Service\MSSQLFDLa...		Manual
SQL Server Browser	NT AUTHORITY\LOCAL ...		Disabled

Default settings, click “Next”.



SQL Server 2014 Setup

Database Engine Configuration

Specify Database Engine authentication security mode, administrators and data directories.

Specify the authentication mode and administrators for the Database Engine.

Authentication Mode

○ Windows authentication mode

● Mixed Mode (SQL Server authentication and Windows authentication)

Specify the password for the SQL Server system administrator (sa) account.

Enter password:

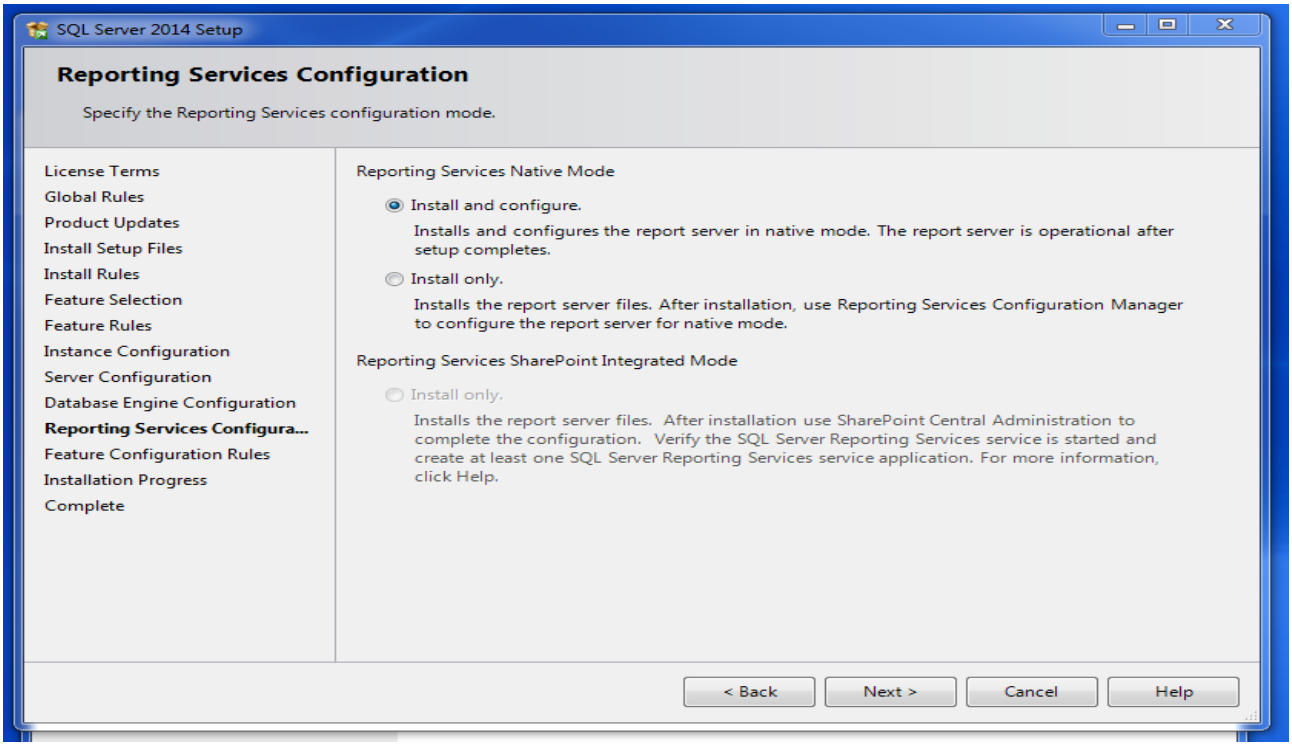
Confirm password:

Specify SQL Server administrators

SQLTEST-PC\SQLTEST (SQLTEST)

SQL Server administrators have unrestricted access to the Database Engine.

Select “Mixed Mode”, and type a password at your own choice for the user “sa”. The user “sa” is the “System Administrator”, and is a mandatory requirement.



Reporting Services Configuration

Specify the Reporting Services configuration mode.

Reporting Services Native Mode

● Install and configure.

Installs and configures the report server in native mode. The report server is operational after setup completes.

○ Install only.

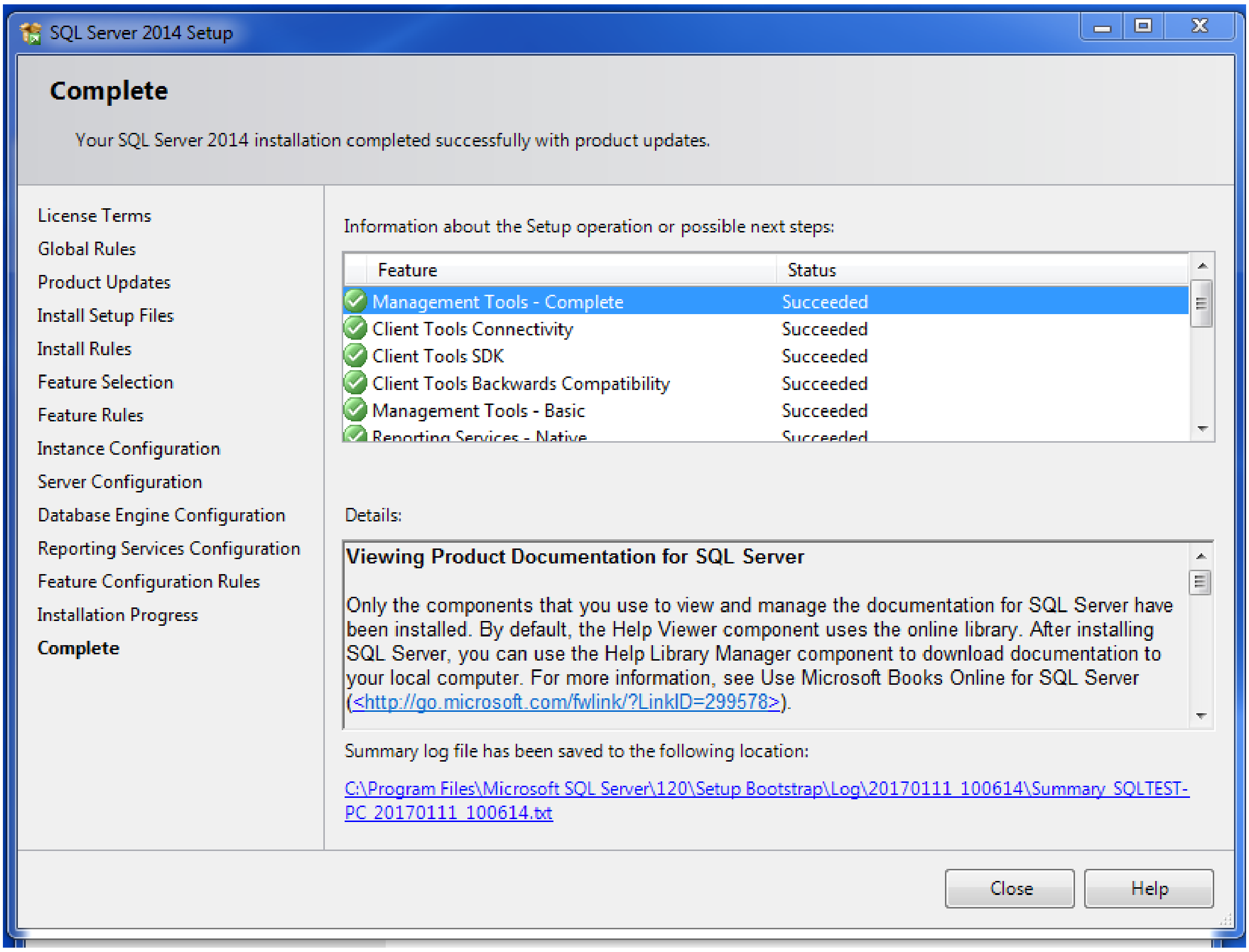
Installs the report server files. After installation, use Reporting Services Configuration Manager to configure the report server for native mode.

Reporting Services SharePoint Integrated Mode

○ Install only.

Installs the report server files. After installation use SharePoint Central Administration to complete the configuration. Verify the SQL Server Reporting Services service is started and create at least one SQL Server Reporting Services service application. For more information, click Help.

Default setting, just click “Next”.



Complete

Your SQL Server 2014 installation completed successfully with product updates.

Information about the Setup operation or possible next steps:

Feature	Status
Management Tools - Complete	Succeeded
Client Tools Connectivity	Succeeded
Client Tools SDK	Succeeded
Client Tools Backwards Compatibility	Succeeded
Management Tools - Basic	Succeeded
Reporting Services - Native	Succeeded

Details:

Viewing Product Documentation for SQL Server

Only the components that you use to view and manage the documentation for SQL Server have been installed. By default, the Help Viewer component uses the online library. After installing SQL Server, you can use the Help Library Manager component to download documentation to your local computer. For more information, see Use Microsoft Books Online for SQL Server (<http://go.microsoft.com/fwlink/?LinkID=299578>).

Summary log file has been saved to the following location:

C:\Program Files\Microsoft SQL Server\120\Setup Bootstrap\Log\20170111_100614\Summary_SQLTEST-PC_20170111_100614.txt

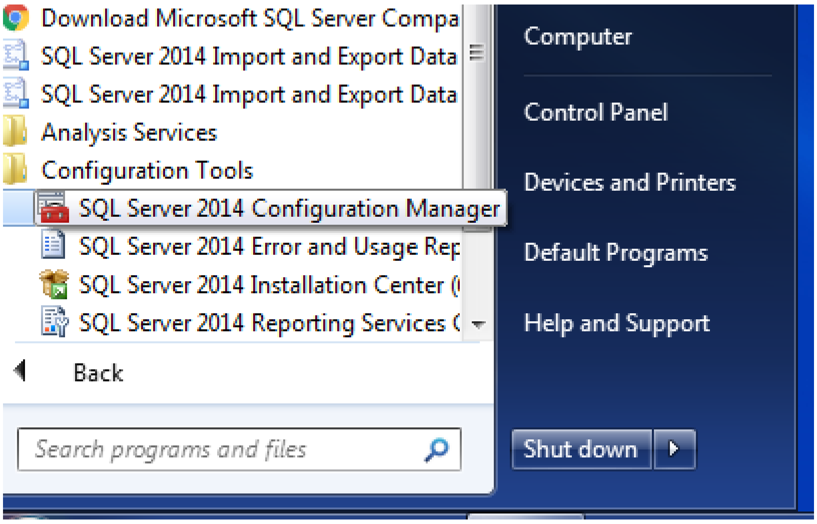
After some time, you should get this window, telling that the server is installed correctly.

Close the window.

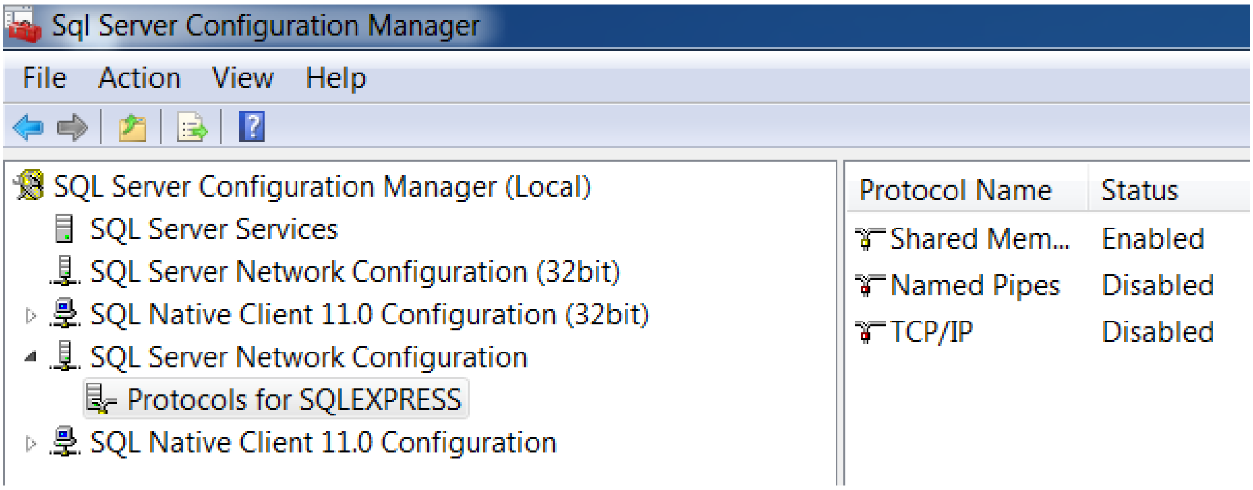
This ends the installation of the SQL server.

SQL Server Configurator Manager, TCP/IP and Port settings

Next we have to activate and open a port, so that the PLC can communicate with the SQL server.


Download Microsoft SQL Server Compa
SQL Server 2014 Import and Export Data
SQL Server 2014 Import and Export Data
Analysis Services
Configuration Tools
SQL Server 2014 Configuration Manager
SQL Server 2014 Error and Usage Rep
SQL Server 2014 Installation Center (
SQL Server 2014 Reporting Services (
Back
Search programs and files
Computer
Control Panel
Devices and Printers
Default Programs
Help and Support
Shut down

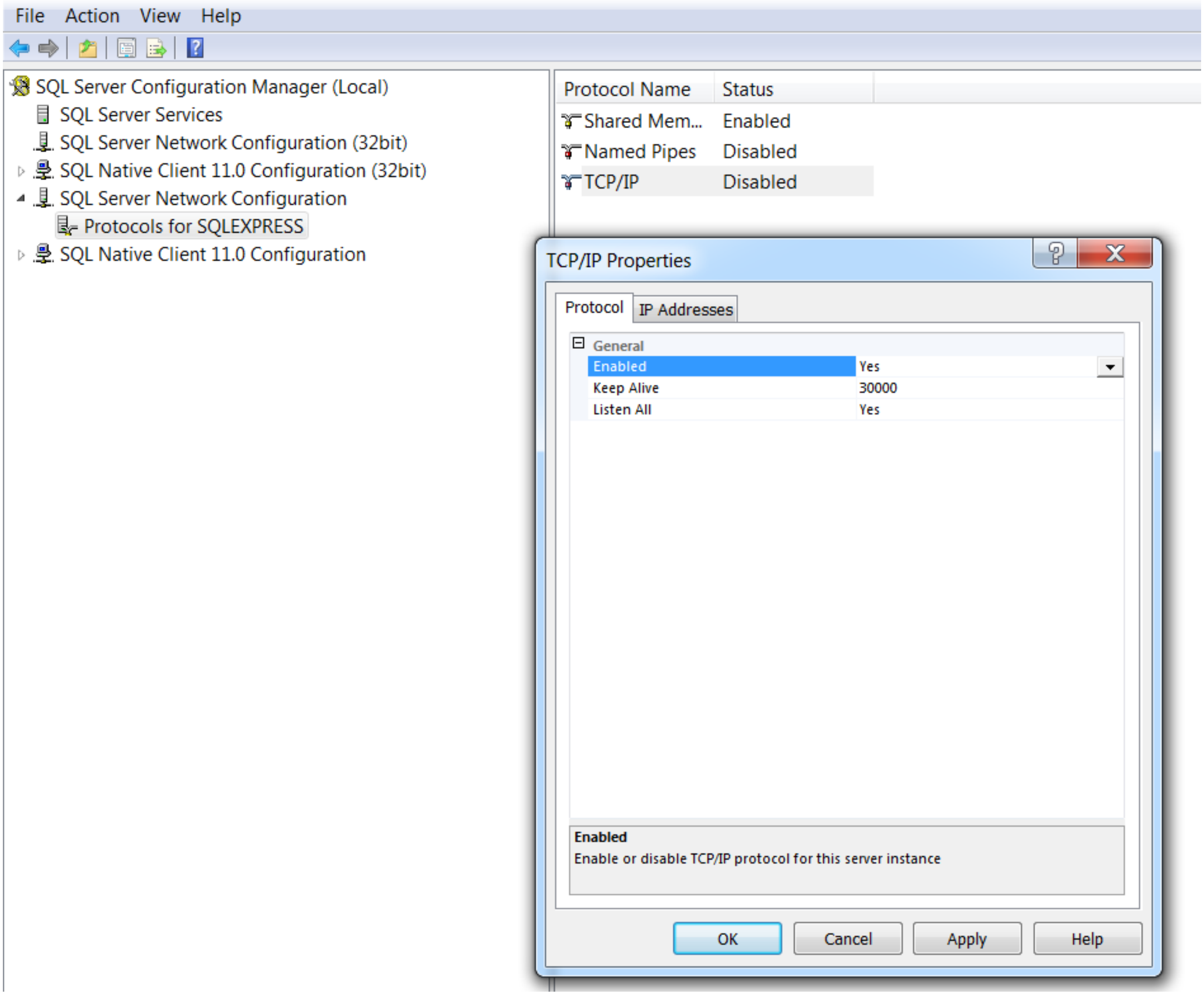
Open the “SQL Server Configuration Manager”.


Sql Server Configuration Manager
File
Action
View
Help
SQL Server Configuration Manager (Local)
SQL Server Services
SQL Server Network Configuration (32bit)
SQL Native Client 11.0 Configuration (32bit)
SQL Server Network Configuration
Protocols for SQLEXPRESS
SQL Native Client 11.0 Configuration

Protocol Name	Status
Shared Mem...	Enabled
Named Pipes	Disabled
TCP/IP	Disabled

Select “SQL Server Network Configuration / Protocols for SQLEXPRESS”, the name “SQLEXPRESS” is the name you chose when you installed the SQL server (Instance Configuration).

Here we have to enable the “TCP/IP” communication, and select the port that is used to communicate with the SQL server.



- SQL Server Configuration Manager (Local)
  - SQL Server Services
  - SQL Server Network Configuration (32bit)
  - SQL Native Client 11.0 Configuration (32bit)
  - SQL Server Network Configuration
    - Protocols for SQLEXPRESS
  - SQL Native Client 11.0 Configuration

Protocol Name	Status
Shared Mem...	Enabled
Named Pipes	Disabled
TCP/IP	Disabled

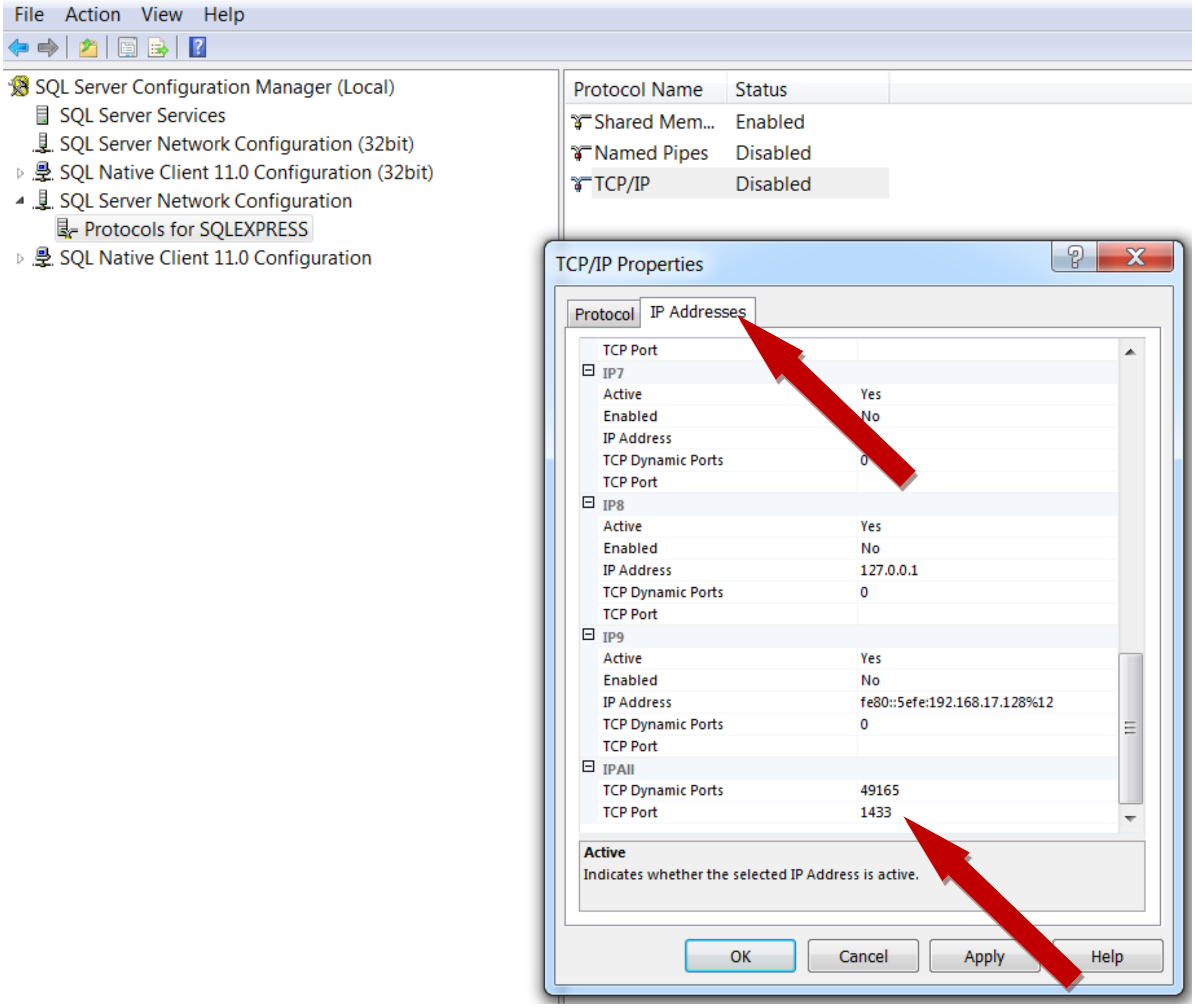
TCP/IP Properties

General

Enabled	Yes
Keep Alive	30000
Listen All	Yes

Enabled
Enable or disable TCP/IP protocol for this server instance

First, Enable the protocol TCP/IP



SQL Server Configuration Manager (Local)
- SQL Server Services
- SQL Server Network Configuration (32bit)
- SQL Native Client 11.0 Configuration (32bit)
- SQL Server Network Configuration
  - Protocols for SQLEXPRESS
- SQL Native Client 11.0 Configuration

Protocol Name	Status
Shared Mem...	Enabled
Named Pipes	Disabled
TCP/IP	Disabled

TCP/IP Properties

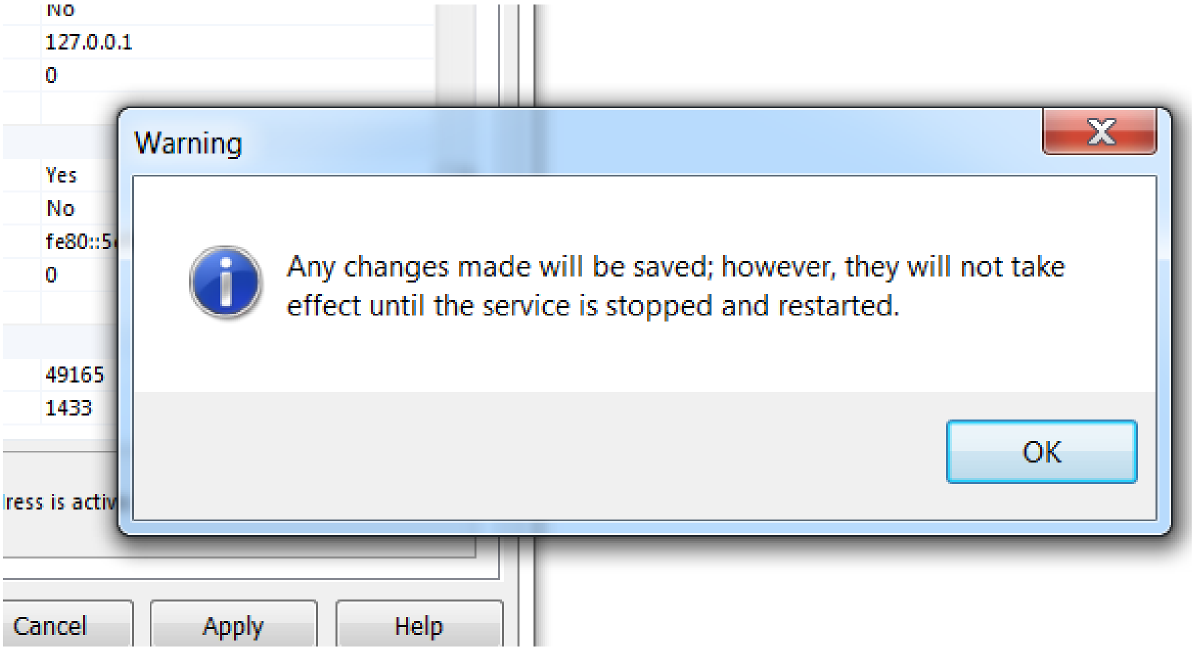
Protocol | IP Addresses

TCP Port	
IP7	
Active	Yes
Enabled	No
IP Address	
TCP Dynamic Ports	0
TCP Port	
IP8	
Active	Yes
Enabled	No
IP Address	127.0.0.1
TCP Dynamic Ports	0
TCP Port	
IP9	
Active	Yes
Enabled	No
IP Address	fe80::5efe:192.168.17.128%12
TCP Dynamic Ports	0
TCP Port	
IPAll	
TCP Dynamic Ports	49165
TCP Port	1433

Active
Indicates whether the selected IP Address is active.

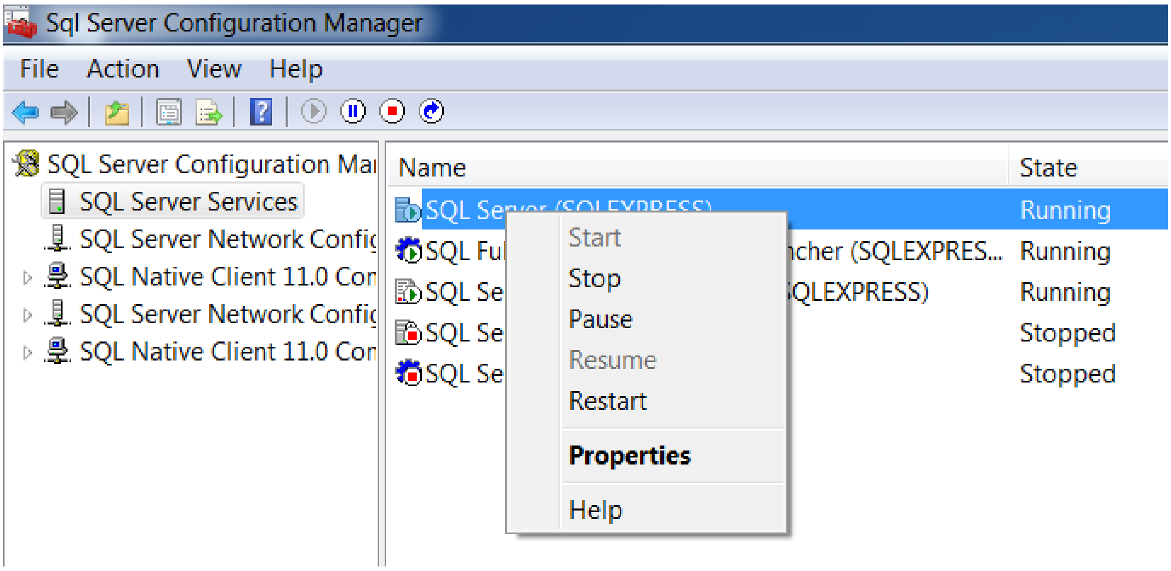
Then select “IP Addresses”, scroll down to the bottom, and type “1433” in “TCP Port”

Close the window by clicking “OK”.



Any changes made will be saved; however, they will not take effect until the service is stopped and restarted.

You will get this message; you have to restart the server.

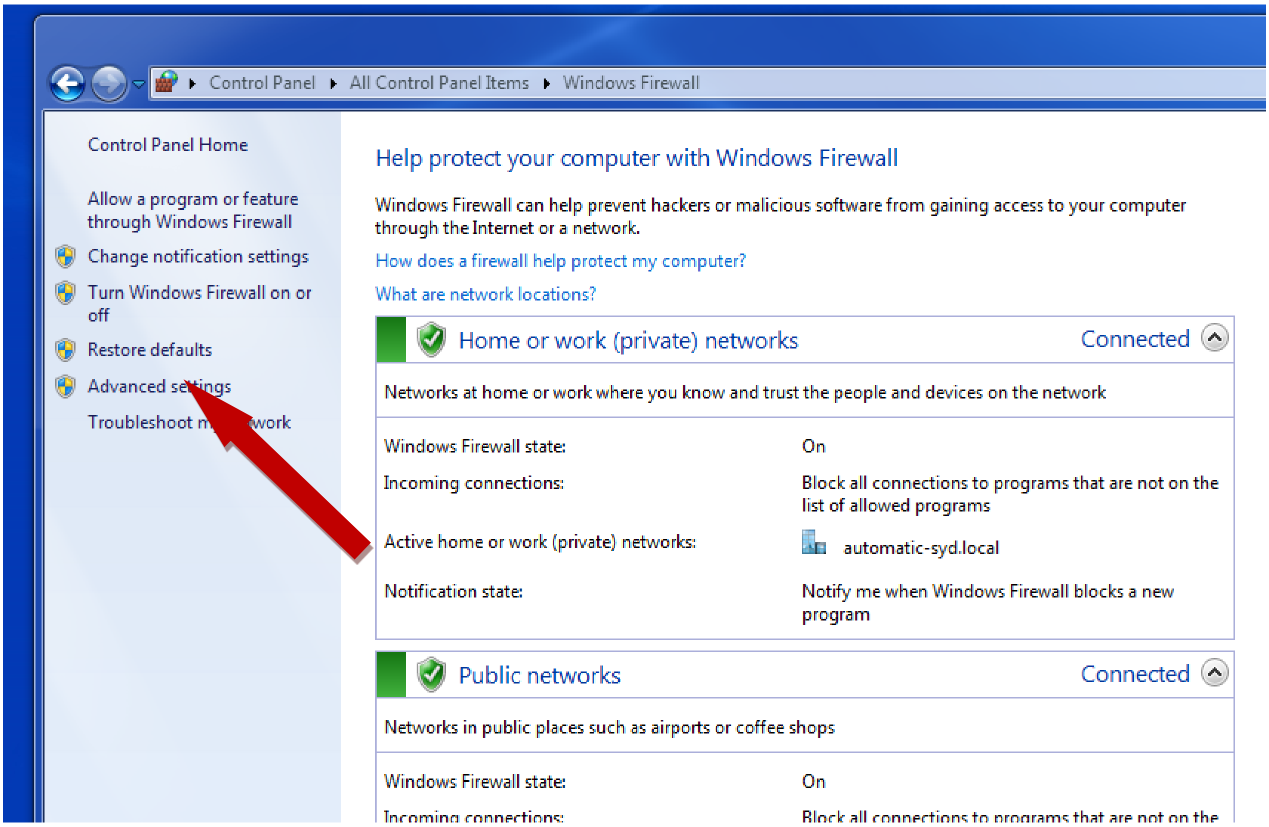


You can restart by right click on “SQL Server” and select “Restart”.

This ends the communication setup of the SQL server.

Setting up Firewall

Next we need to make some changes in the “Firewall”, you can disable the Firewall, or you can make the changes needed.
It is necessary to make these changes, so the PLC can communicate with the SQL server.


Control Panel ▸ All Control Panel Items ▸ Windows Firewall
Control Panel Home
Allow a program or feature through Windows Firewall
Change notification settings
Turn Windows Firewall on or off
Restore defaults
Advanced settings
Troubleshoot my network

Help protect your computer with Windows Firewall

Windows Firewall can help prevent hackers or malicious software from gaining access to your computer through the Internet or a network.

How does a firewall help protect my computer?

What are network locations?

Home or work (private) networks — Connected

Networks at home or work where you know and trust the people and devices on the network

Windows Firewall state:	On
Incoming connections:	Block all connections to programs that are not on the list of allowed programs
Active home or work (private) networks:	automatic-syd.local
Notification state:	Notify me when Windows Firewall blocks a new program

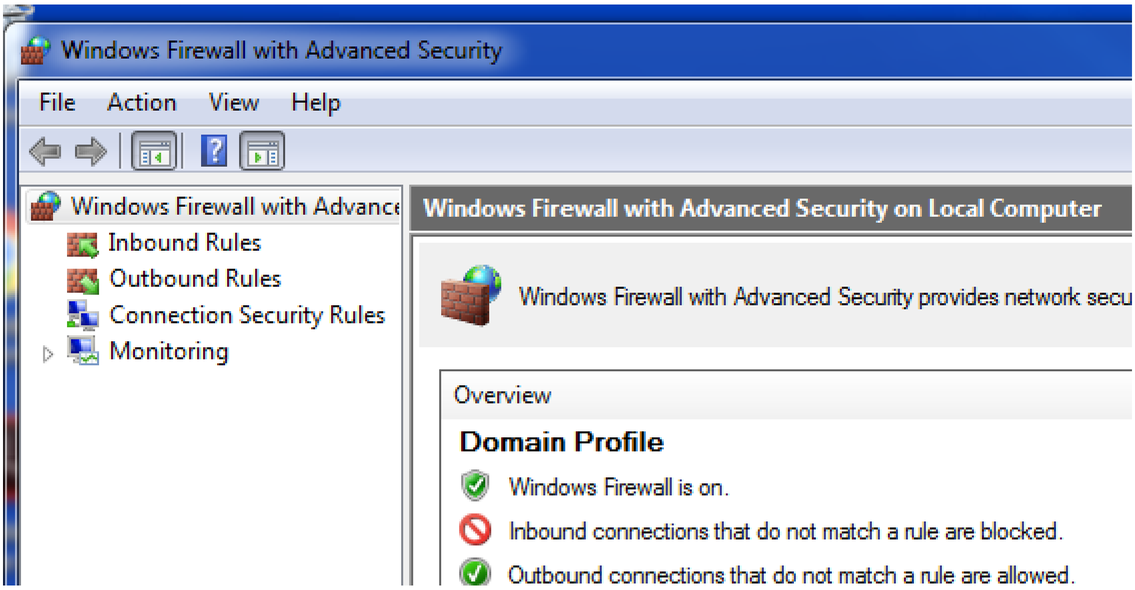
Public networks — Connected

Networks in public places such as airports or coffee shops

Windows Firewall state:	On
Incoming connections:	Block all connections to programs that are not on the

Open the “Firewall”, (placed in “Control Panel”).

Select “Advanced settings”


Windows Firewall with Advanced Security
File
Action
View
Help
Windows Firewall with Advance
Inbound Rules
Outbound Rules
Connection Security Rules
Monitoring
Windows Firewall with Advanced Security on Local Computer
Overview

Windows Firewall with Advanced Security provides network secu

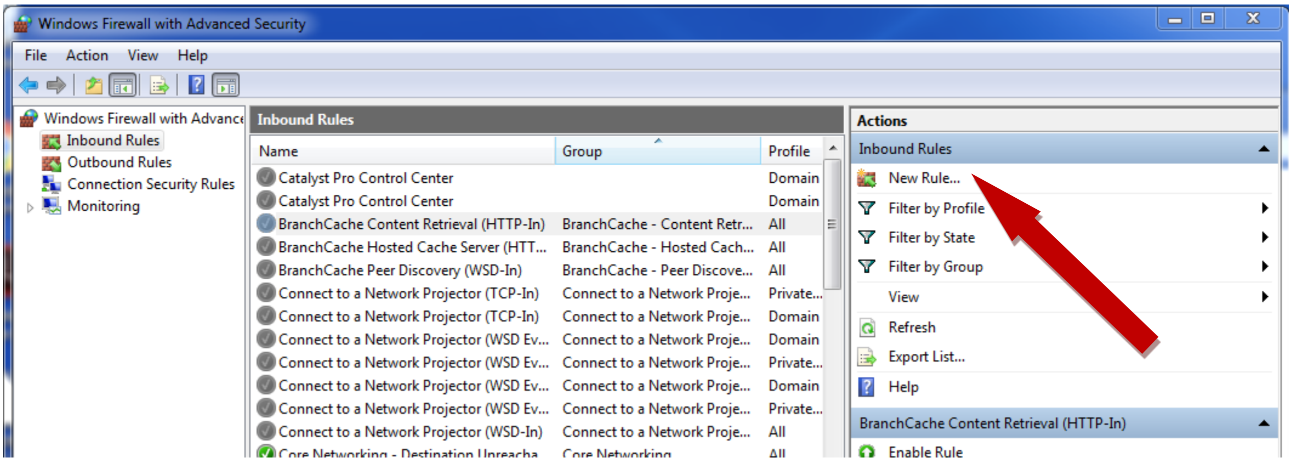
Domain Profile

Windows Firewall is on.

Inbound connections that do not match a rule are blocked.

Outbound connections that do not match a rule are allowed.

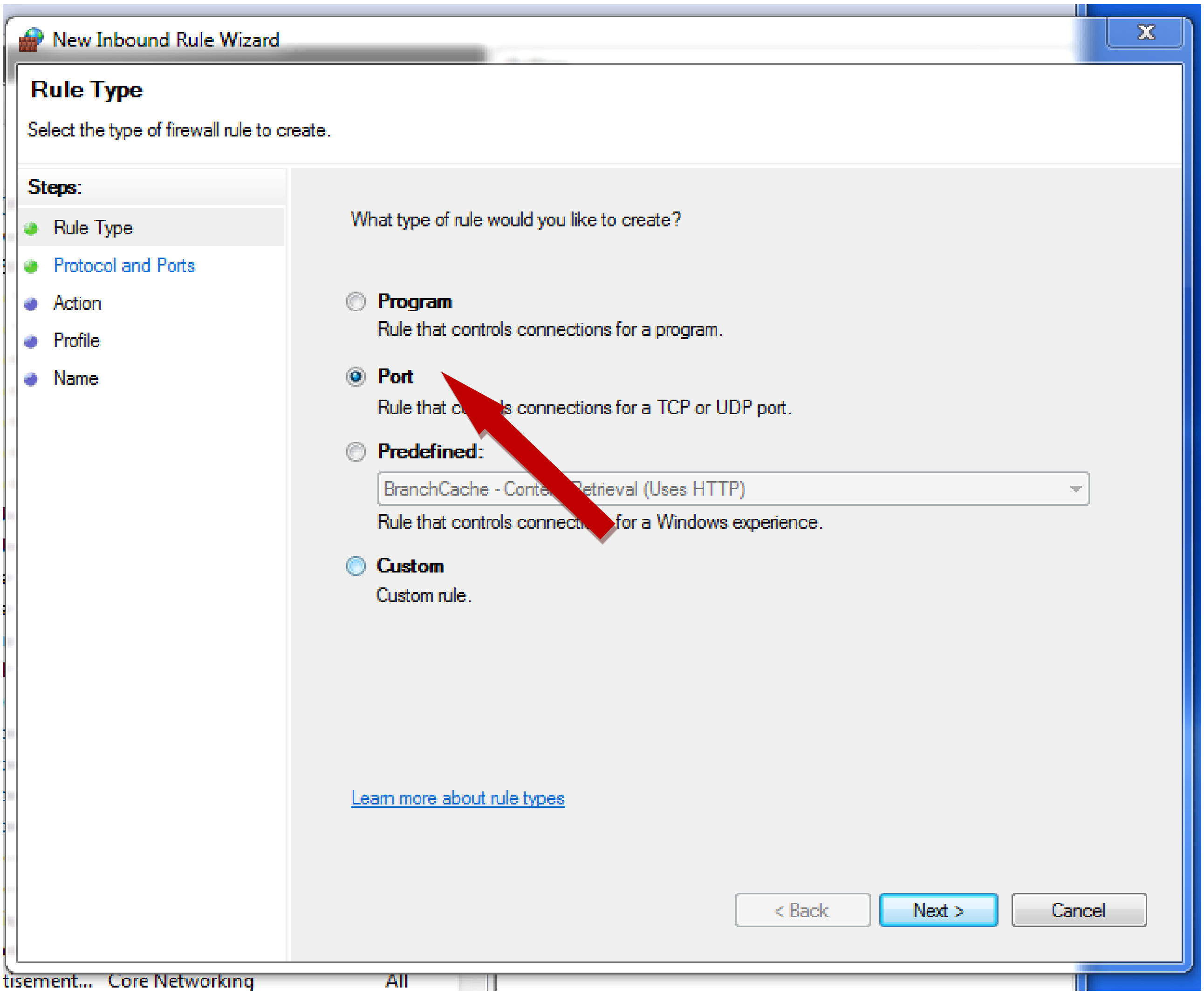
Select “Inbound Rules”



Inbound Rules

Name	Group	Profile
Catalyst Pro Control Center		Domain
Catalyst Pro Control Center		Domain
BranchCache Content Retrieval (HTTP-In)	BranchCache - Content Retr...	All
BranchCache Hosted Cache Server (HTT...	BranchCache - Hosted Cach...	All
BranchCache Peer Discovery (WSD-In)	BranchCache - Peer Discove...	All
Connect to a Network Projector (TCP-In)	Connect to a Network Proje...	Private...
Connect to a Network Projector (TCP-In)	Connect to a Network Proje...	Domain
Connect to a Network Projector (WSD Ev...	Connect to a Network Proje...	Domain
Connect to a Network Projector (WSD Ev...	Connect to a Network Proje...	Private...
Connect to a Network Projector (WSD Ev...	Connect to a Network Proje...	Domain
Connect to a Network Projector (WSD Ev...	Connect to a Network Proje...	Private...
Connect to a Network Projector (WSD-In)	Connect to a Network Proje...	All
Core Networking - Destination Unreacha...	Core Networking	All

Select “New rule”



Rule Type

Select the type of firewall rule to create.

What type of rule would you like to create?

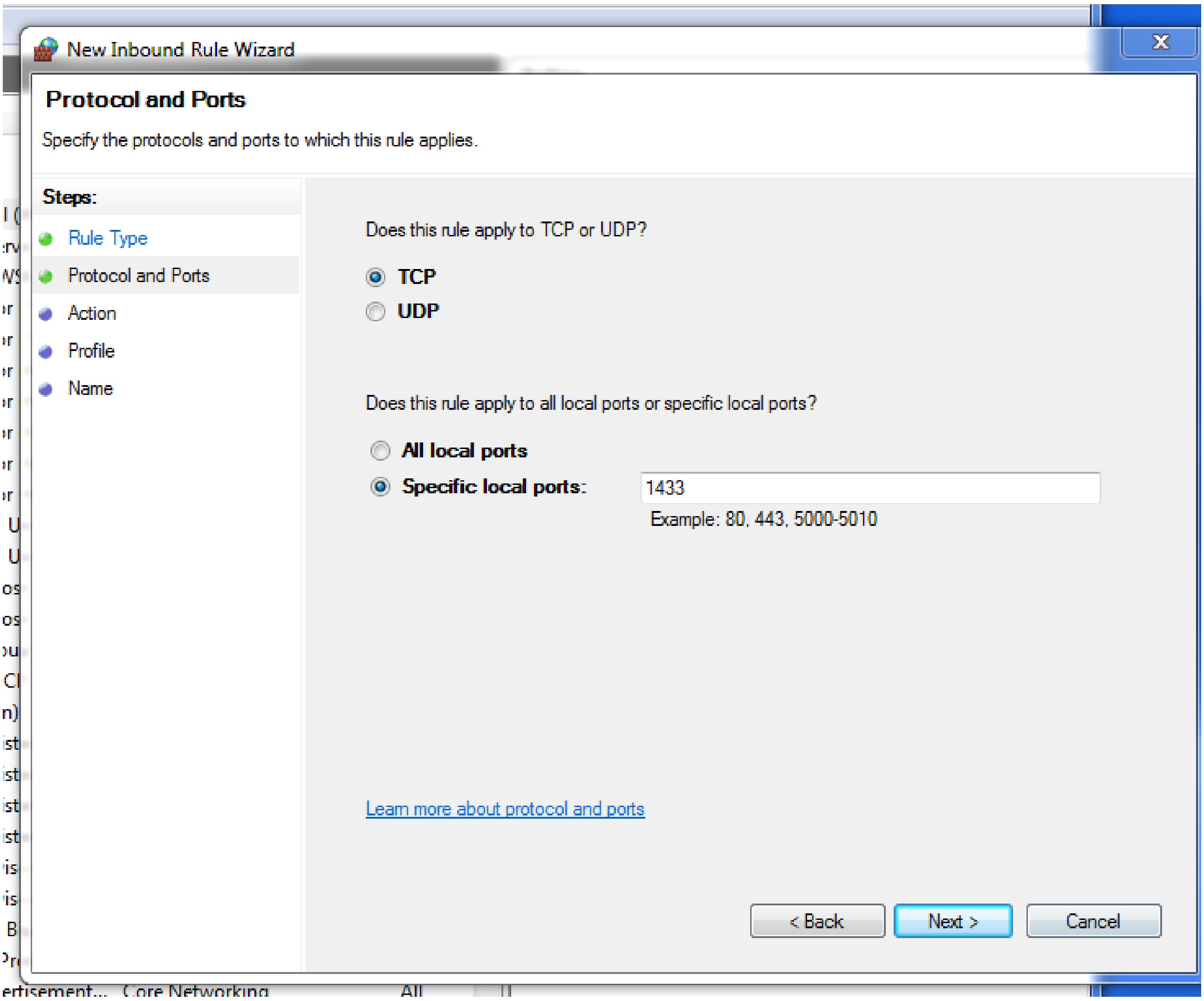
○ **Program**
Rule that controls connections for a program.

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

○ **Predefined:**
BranchCache - Content Retrieval (Uses HTTP)
Rule that controls connections for a Windows experience.

○ **Custom**
Custom rule.

Select “Port”, and “Next”



New Inbound Rule Wizard

Protocol and Ports

Specify the protocols and ports to which this rule applies.

Steps:

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

Does this rule apply to TCP or UDP?

● TCP

○ UDP

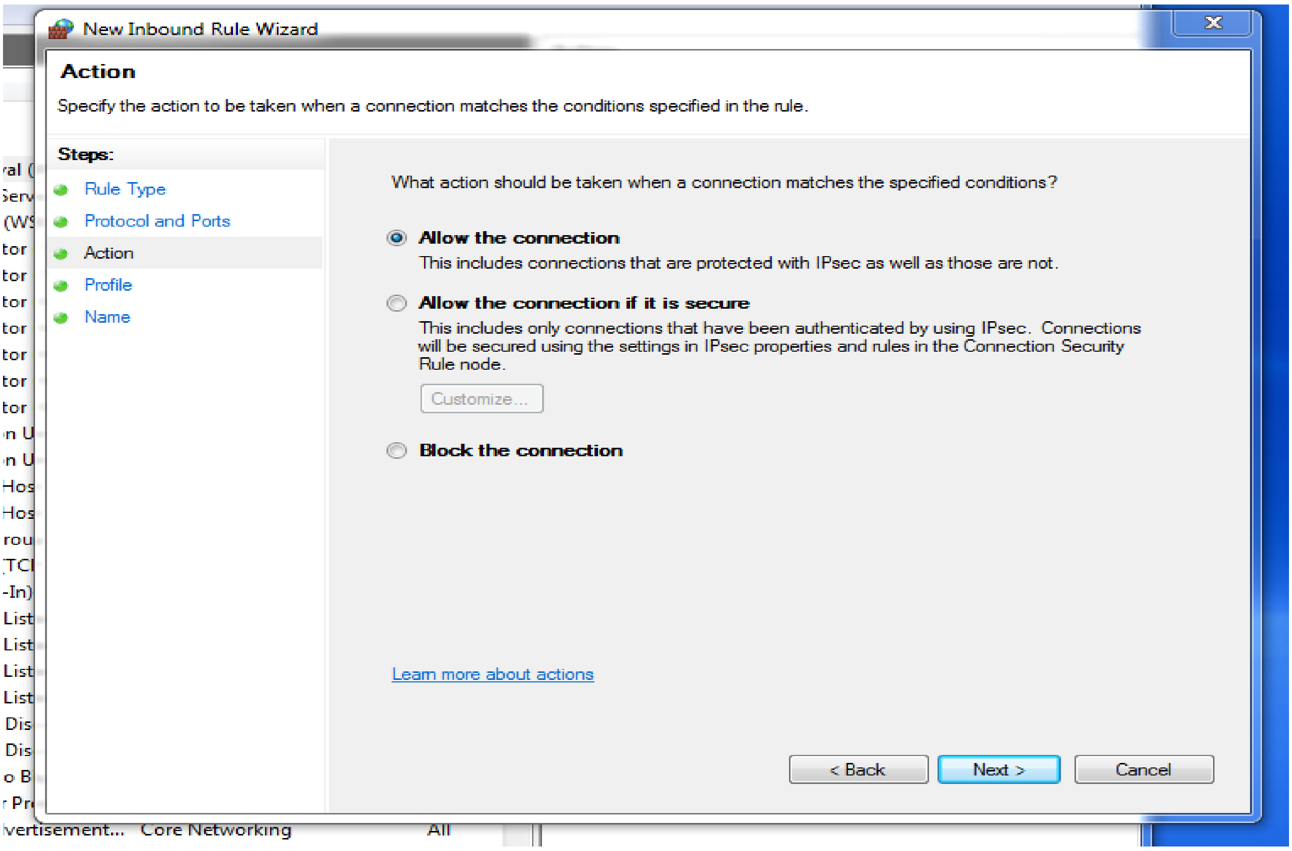
Does this rule apply to all local ports or specific local ports?

○ All local ports

● Specific local ports: 1433

Example: 80, 443, 5000-5010

Select “TCP” and write “1433” in “Specific local ports”, and “Next”.



New Inbound Rule Wizard

Action

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

Steps:

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

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

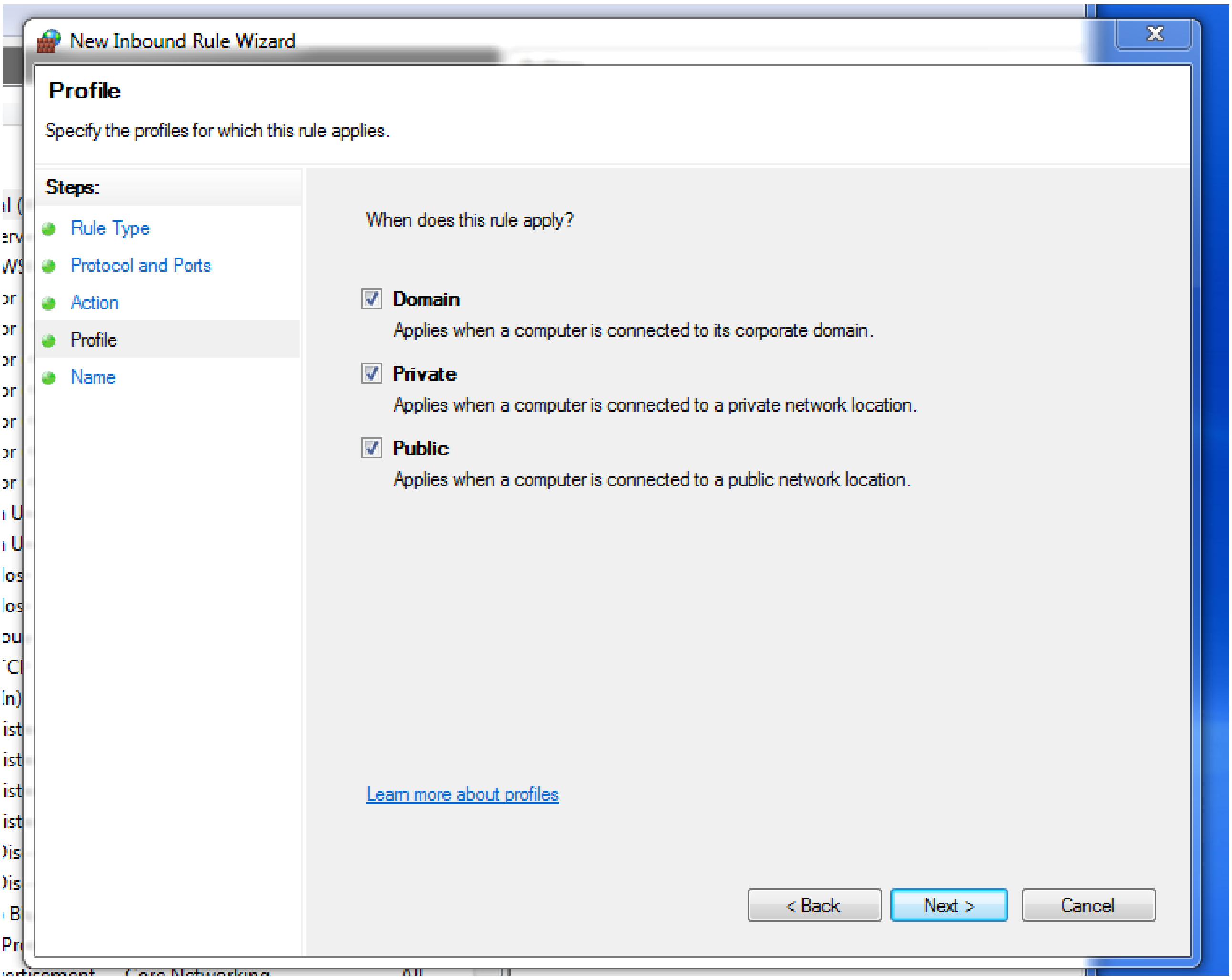
● **Allow the connection**
This includes connections that are protected with IPsec as well as those are not.

○ **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**

Learn more about actions

“Allow the connection”, and “Next”.



Profile

Specify the profiles for which this rule applies.

When does this rule apply?

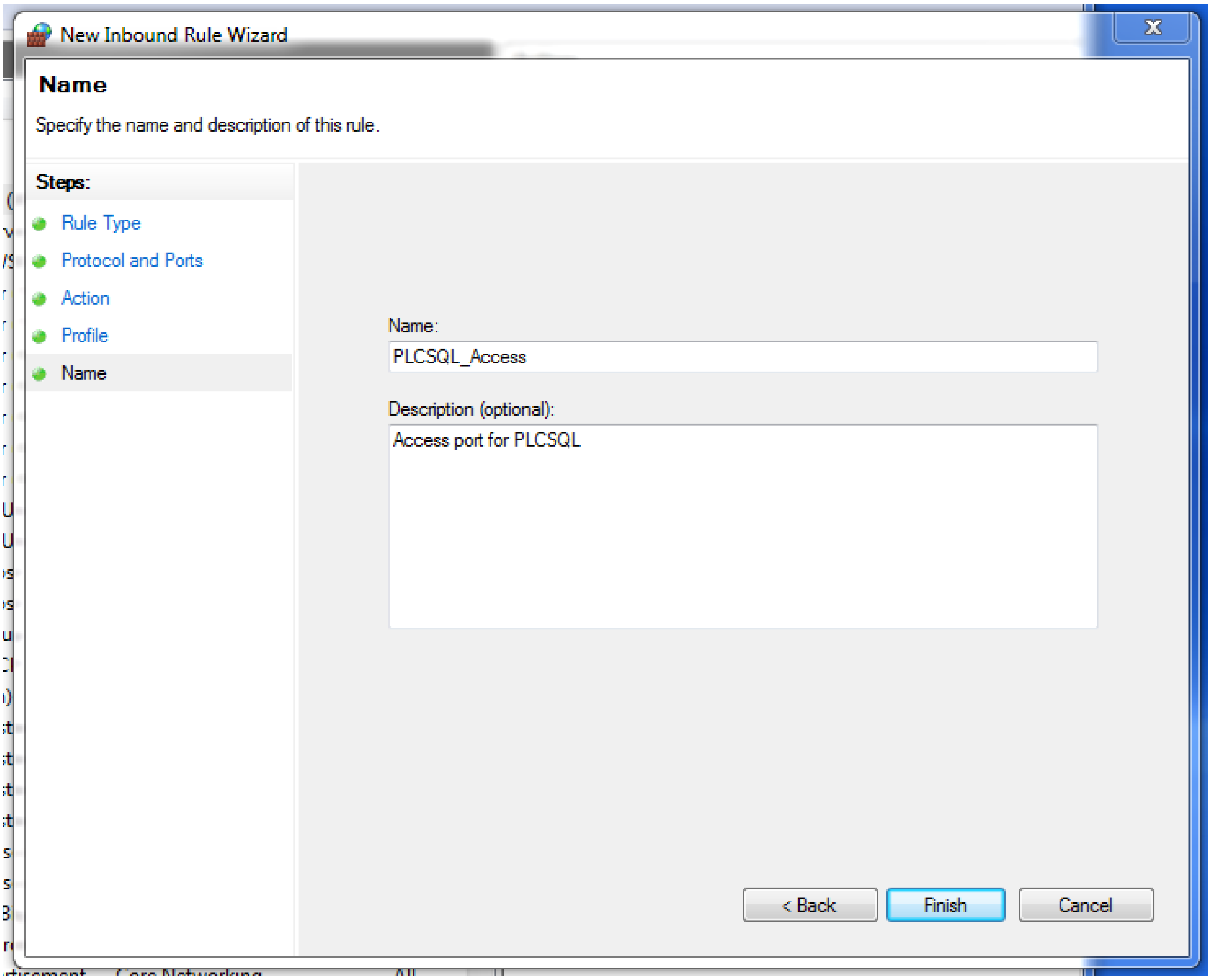
☑ **Domain**
Applies when a computer is connected to its corporate domain.

☑ **Private**
Applies when a computer is connected to a private network location.

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

Learn more about profiles

Default setting, be sure that your net card is connected to the correct “Profile” if you change this setting, and “Next”.



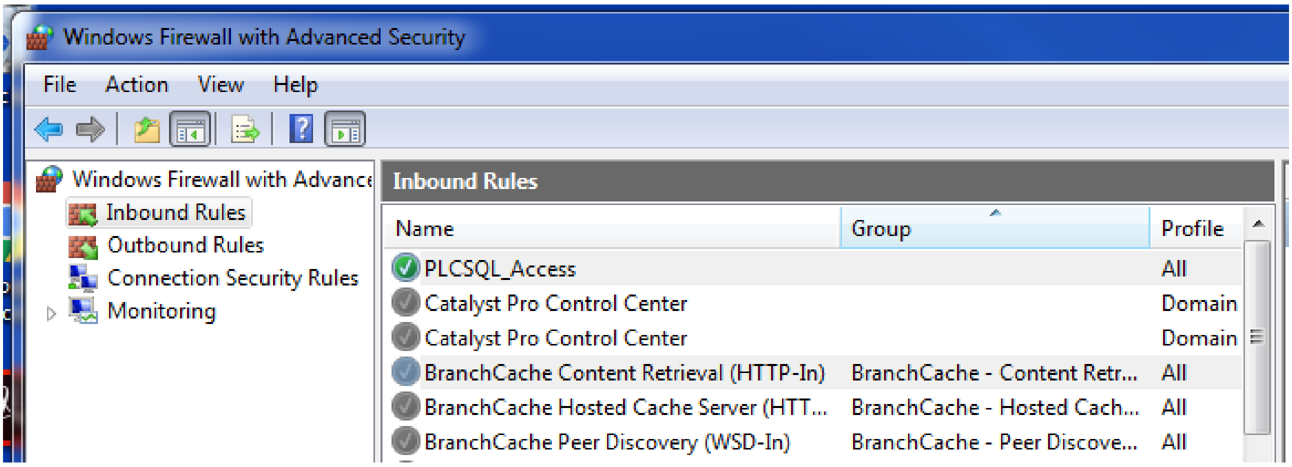
Name

Specify the name and description of this rule.

Name: PLCSQL_Access

Description (optional): Access port for PLCSQL

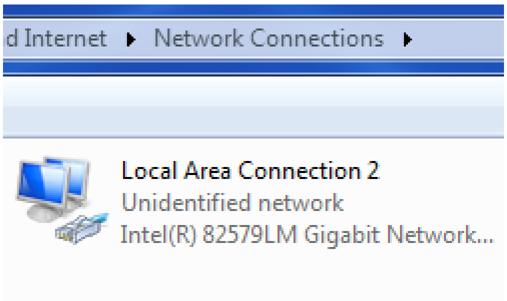
At last, you have to give the Inbound rule a name at your own choice, and “Finish”.


Windows Firewall with Advanced Security
File
Action
View
Help
Windows Firewall with Advance
Inbound Rules
Outbound Rules
Connection Security Rules
Monitoring
Inbound Rules

Name	Group	Profile
✓ PLCSQL_Access		All
✓ Catalyst Pro Control Center		Domain
✓ Catalyst Pro Control Center		Domain
✓ BranchCache Content Retrieval (HTTP-In)	BranchCache - Content Retr...	All
✓ BranchCache Hosted Cache Server (HTT...	BranchCache - Hosted Cach...	All
✓ BranchCache Peer Discovery (WSD-In)	BranchCache - Peer Discove...	All

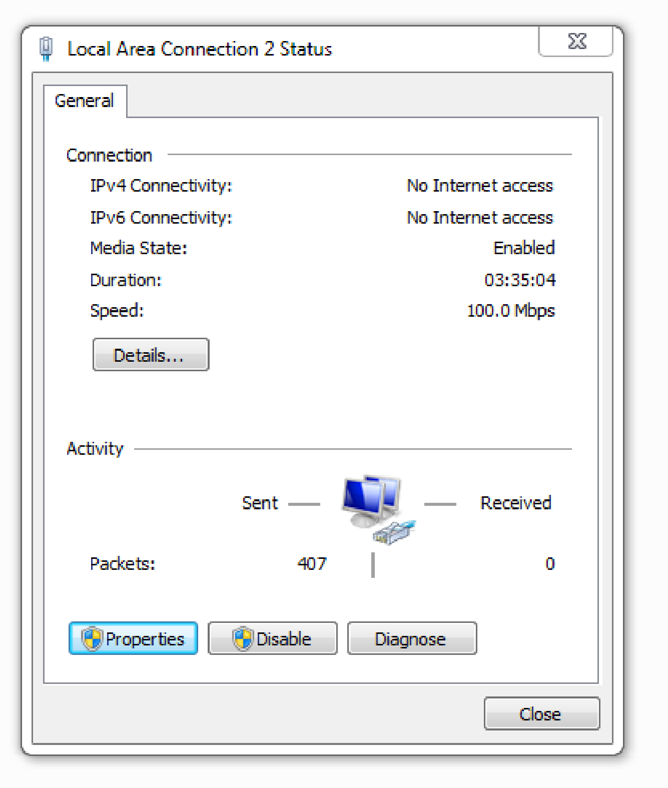
The new rule for the Firewall.
It is not necessary to make a rule for “Outbound”.

Setting up SQL Server IP address.



d Internet ▸ Network Connections ▸
Local Area Connection 2
Unidentified network
Intel(R) 82579LM Gigabit Network...

Select the net card you want to use to communicate with the PLC.



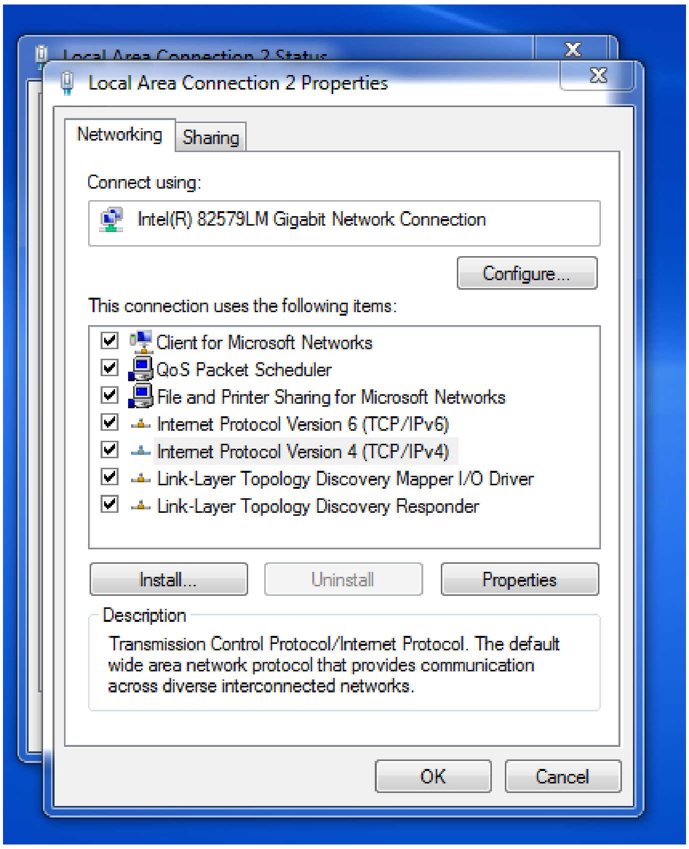
Connection

IPv4 Connectivity:	No Internet access
IPv6 Connectivity:	No Internet access
Media State:	Enabled
Duration:	03:35:04
Speed:	100.0 Mbps

	Sent	Received
Packets:	407	0

Double click on the net card to open the setup utility.

Select “Properties”



Local Area Connection 2 Status
Local Area Connection 2 Properties
Networking
Sharing

Connect using:

Intel(R) 82579LM Gigabit Network Connection

This connection uses the following items:

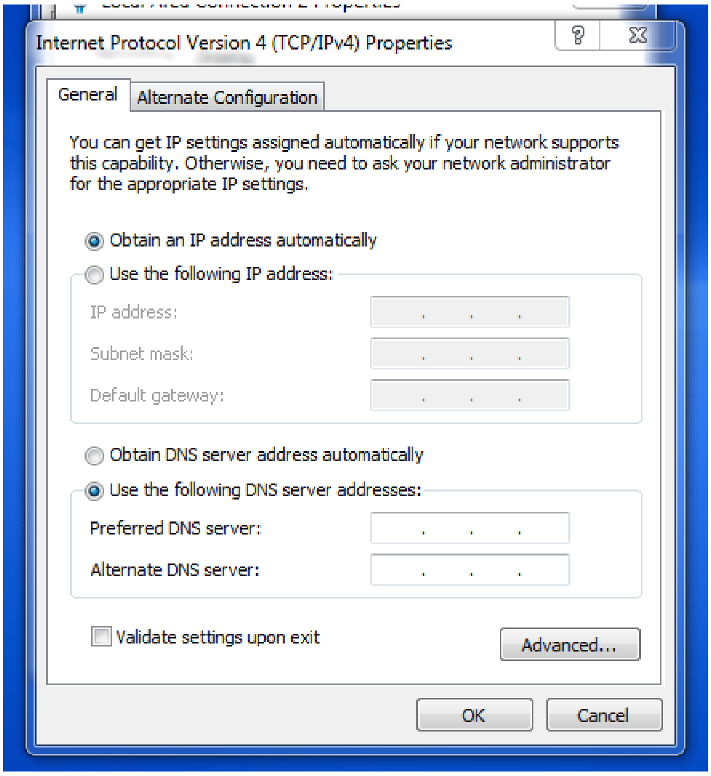
- ☑ Client for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ Internet Protocol Version 6 (TCP/IPv6)
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- ☑ Link-Layer Topology Discovery Mapper I/O Driver
- ☑ Link-Layer Topology Discovery Responder

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

Select “Internet Protocol Version 4 (TCP/IPv4)”

Select “Properties”



Internet Protocol Version 4 (TCP/IPv4) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically

○ Use the following IP address:

IP address: . . .

Subnet mask: . . .

Default gateway: . . .

○ Obtain DNS server address automatically

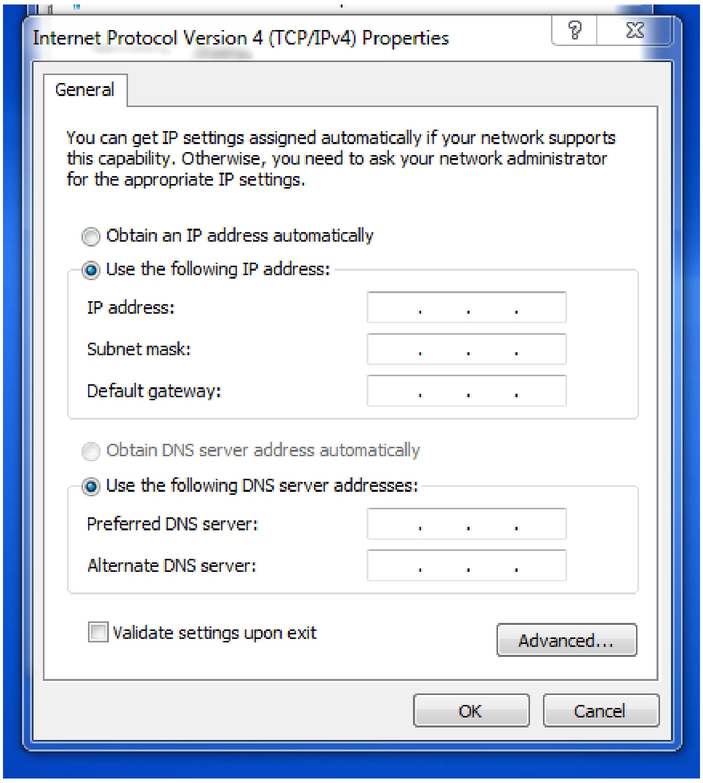
● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit

Select “Use the following IP address”



Internet Protocol Version 4 (TCP/IPv4) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: . . .

Subnet mask: . . .

Default gateway: . . .

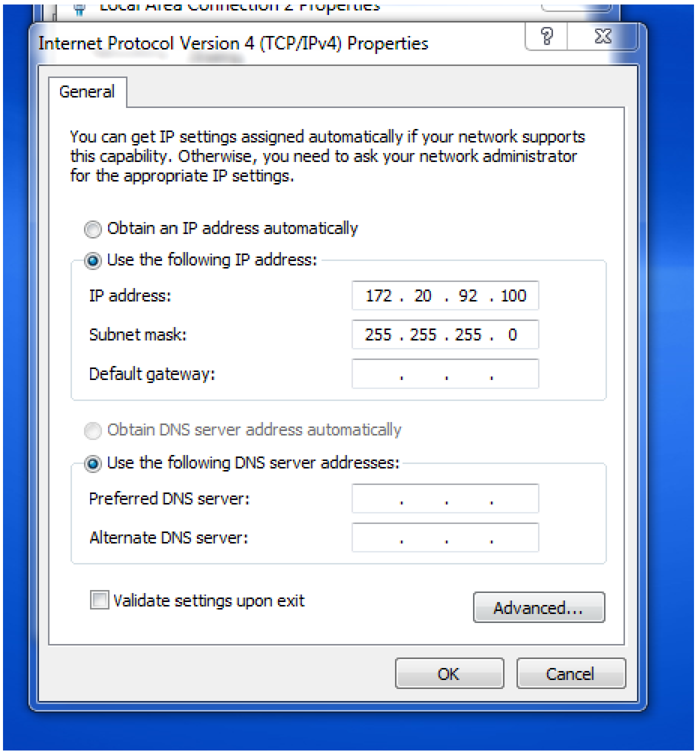
○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit


Local Area Connection 2 Properties
Internet Protocol Version 4 (TCP/IPv4) Properties
General
Advanced...
OK
Cancel

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 172 . 20 . 92 . 100

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

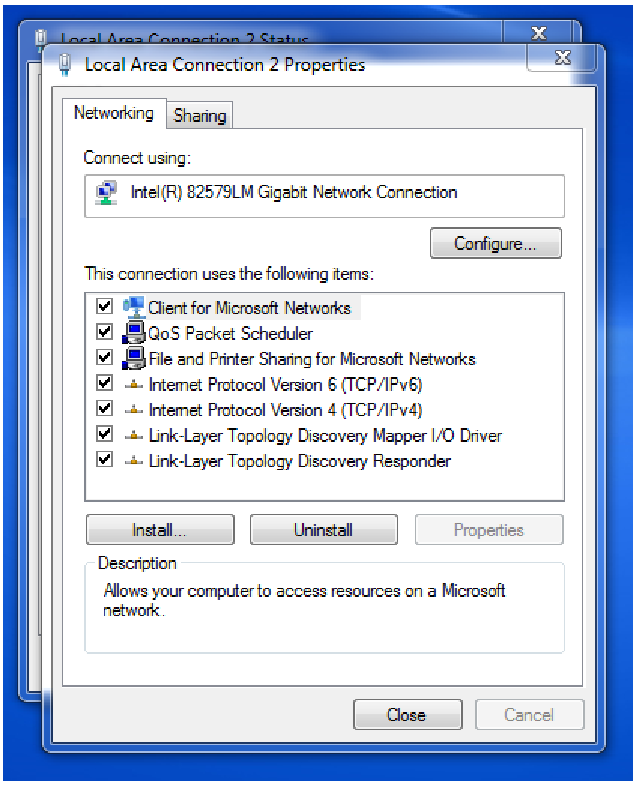
● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit

Type the IP address you want the server to have, Click “OK”.
It is necessary that the SQL server has a fixed IP address.



Connect using:

Intel(R) 82579LM Gigabit Network Connection

This connection uses the following items:

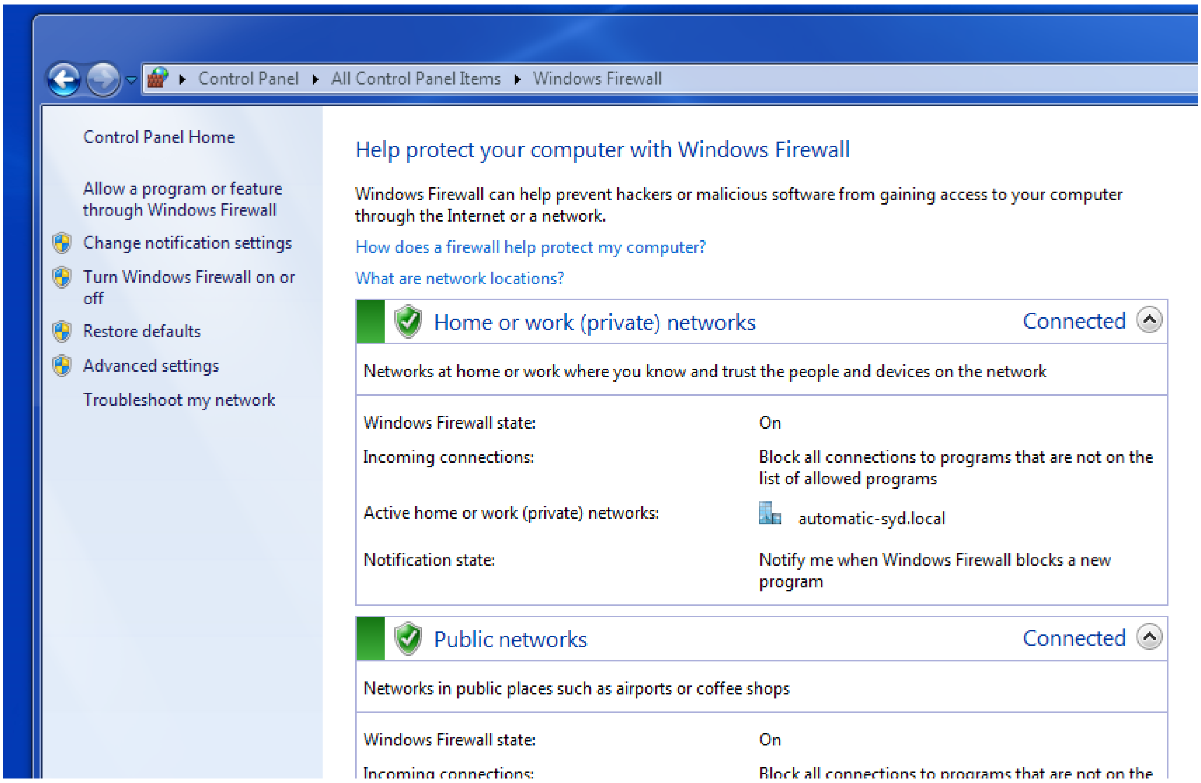
- ☑ Client for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ Internet Protocol Version 6 (TCP/IPv6)
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- ☑ Link-Layer Topology Discovery Mapper I/O Driver
- ☑ Link-Layer Topology Discovery Responder

Description

Allows your computer to access resources on a Microsoft network.

Close the windows again.

If you want to test the connection to the SQL server from another PC, you may use the build in command “Ping”.
From Windows 7, the “Ping” answer is blocked by the firewall, so you have to enable the server to answer the “Ping” request.


Control Panel ▸ All Control Panel Items ▸ Windows Firewall
Control Panel Home
Allow a program or feature through Windows Firewall
Change notification settings
Turn Windows Firewall on or off
Restore defaults
Advanced settings
Troubleshoot my network
How does a firewall help protect my computer?
What are network locations?
Connected
Connected

Help protect your computer with Windows Firewall

Windows Firewall can help prevent hackers or malicious software from gaining access to your computer through the Internet or a network.

Home or work (private) networks — Connected

Networks at home or work where you know and trust the people and devices on the network

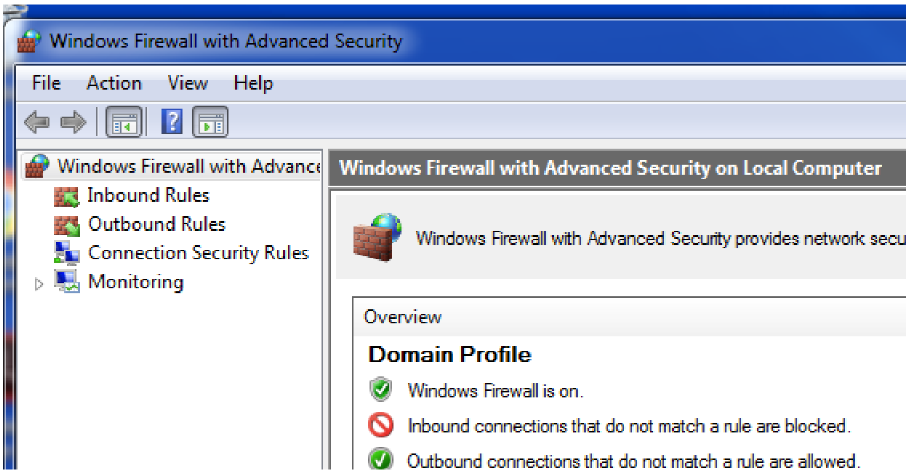
Windows Firewall state:	On
Incoming connections:	Block all connections to programs that are not on the list of allowed programs
Active home or work (private) networks:	automatic-syd.local
Notification state:	Notify me when Windows Firewall blocks a new program

Public networks — Connected

Networks in public places such as airports or coffee shops

Windows Firewall state:	On
Incoming connections:	Block all connections to programs that are not on the

Open firewall, and select “Advanced settings”


Windows Firewall with Advanced Security
File
Action
View
Help
Windows Firewall with Advance
Inbound Rules
Outbound Rules
Connection Security Rules
Monitoring

Windows Firewall with Advanced Security on Local Computer

Windows Firewall with Advanced Security provides network secu

Overview

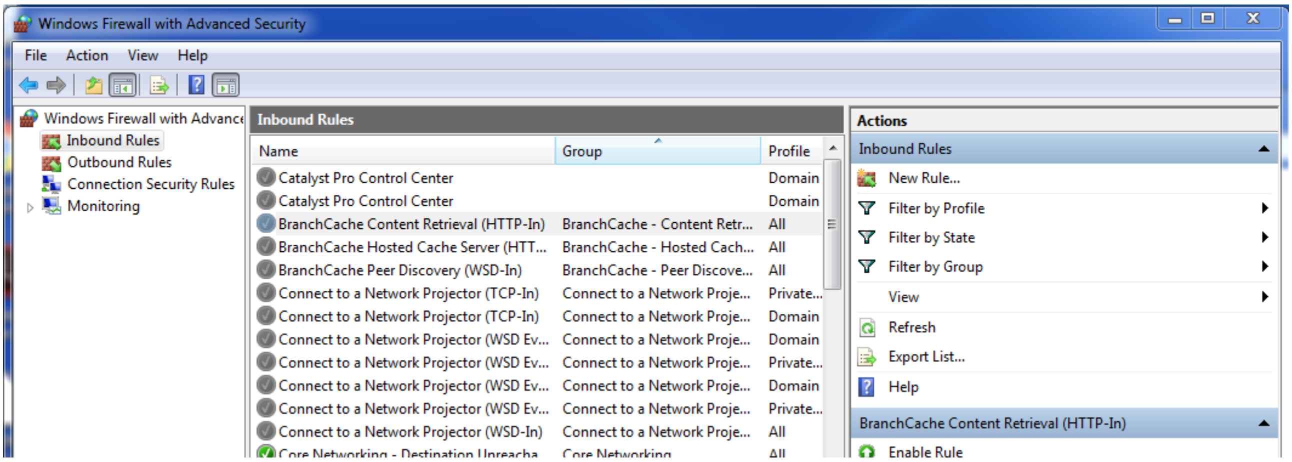
Domain Profile

Windows Firewall is on.

Inbound connections that do not match a rule are blocked.

Outbound connections that do not match a rule are allowed.

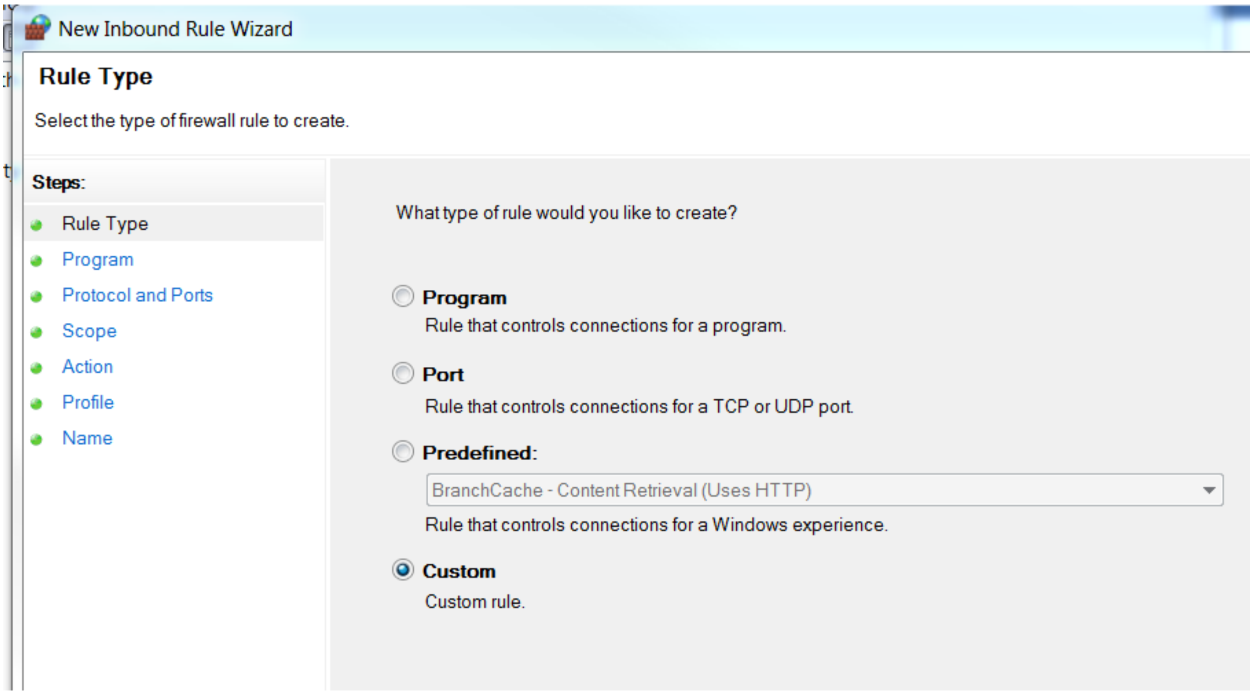
Select “Inbound Rules”



Inbound Rules

Name	Group	Profile
Catalyst Pro Control Center		Domain
Catalyst Pro Control Center		Domain
BranchCache Content Retrieval (HTTP-In)	BranchCache - Content Retr...	All
BranchCache Hosted Cache Server (HTT...	BranchCache - Hosted Cach...	All
BranchCache Peer Discovery (WSD-In)	BranchCache - Peer Discove...	All
Connect to a Network Projector (TCP-In)	Connect to a Network Proje...	Private...
Connect to a Network Projector (TCP-In)	Connect to a Network Proje...	Domain
Connect to a Network Projector (WSD Ev...	Connect to a Network Proje...	Domain
Connect to a Network Projector (WSD Ev...	Connect to a Network Proje...	Private...
Connect to a Network Projector (WSD Ev...	Connect to a Network Proje...	Domain
Connect to a Network Projector (WSD Ev...	Connect to a Network Proje...	Private...
Connect to a Network Projector (WSD-In)	Connect to a Network Proje...	All
Core Networking - Destination Unreacha...	Core Networking	All

Select “New Rule…..



New Inbound Rule Wizard

Rule Type

Select the type of firewall rule to create.

What type of rule would you like to create?

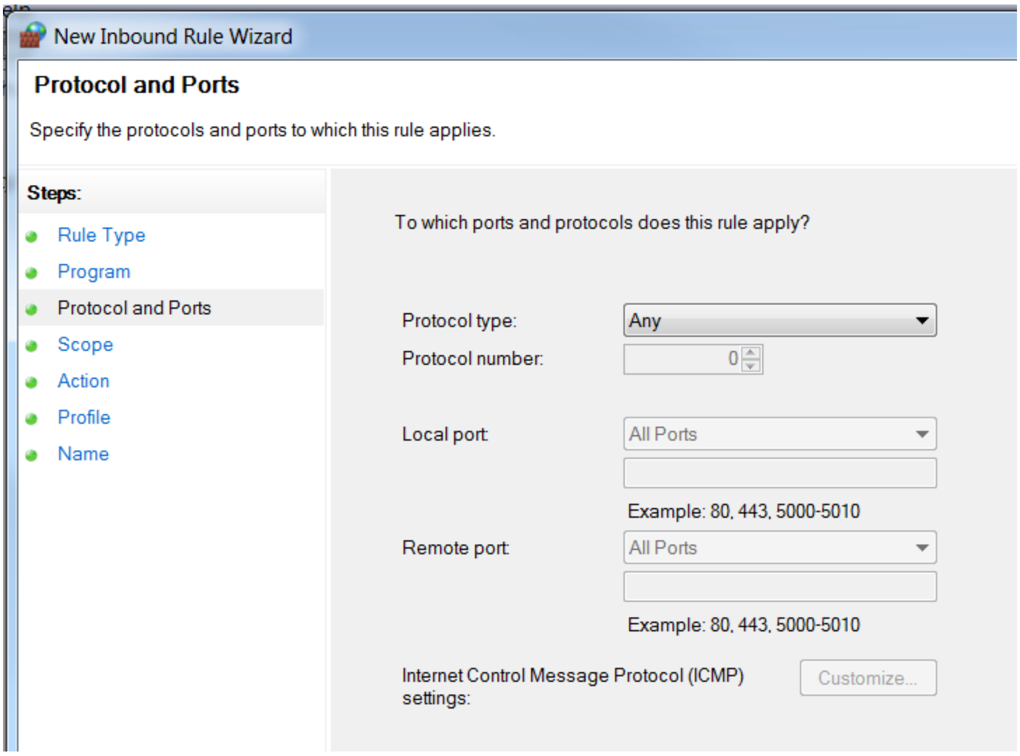
○ **Program**
Rule that controls connections for a program.

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

○ **Predefined:**
BranchCache - Content Retrieval (Uses HTTP)
Rule that controls connections for a Windows experience.

● **Custom**
Custom rule.

As “Rule Type”, select “Custom”.



New Inbound Rule Wizard

Protocol and Ports

Specify the protocols and ports to which this rule applies.

To which ports and protocols does this rule apply?

Protocol type: Any

Protocol number: 0

Local port: All Ports

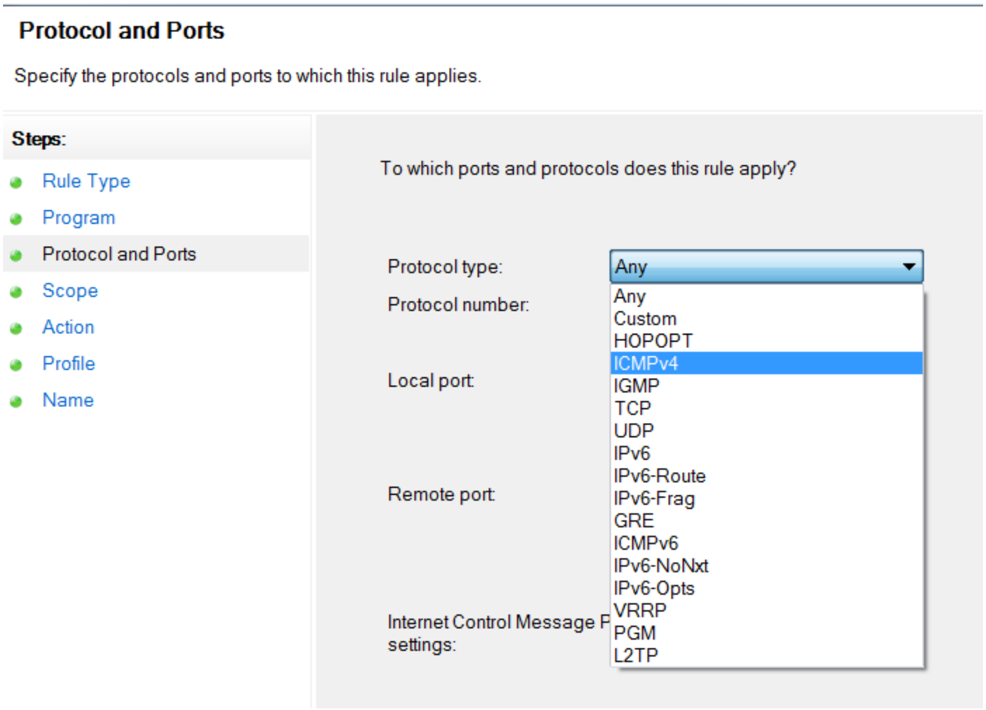
Example: 80, 443, 5000-5010

Remote port: All Ports

Example: 80, 443, 5000-5010

Internet Control Message Protocol (ICMP) settings:

Select “Protocols and Ports”.



Protocol and Ports

Specify the protocols and ports to which this rule applies.

To which ports and protocols does this rule apply?

Protocol type: Any

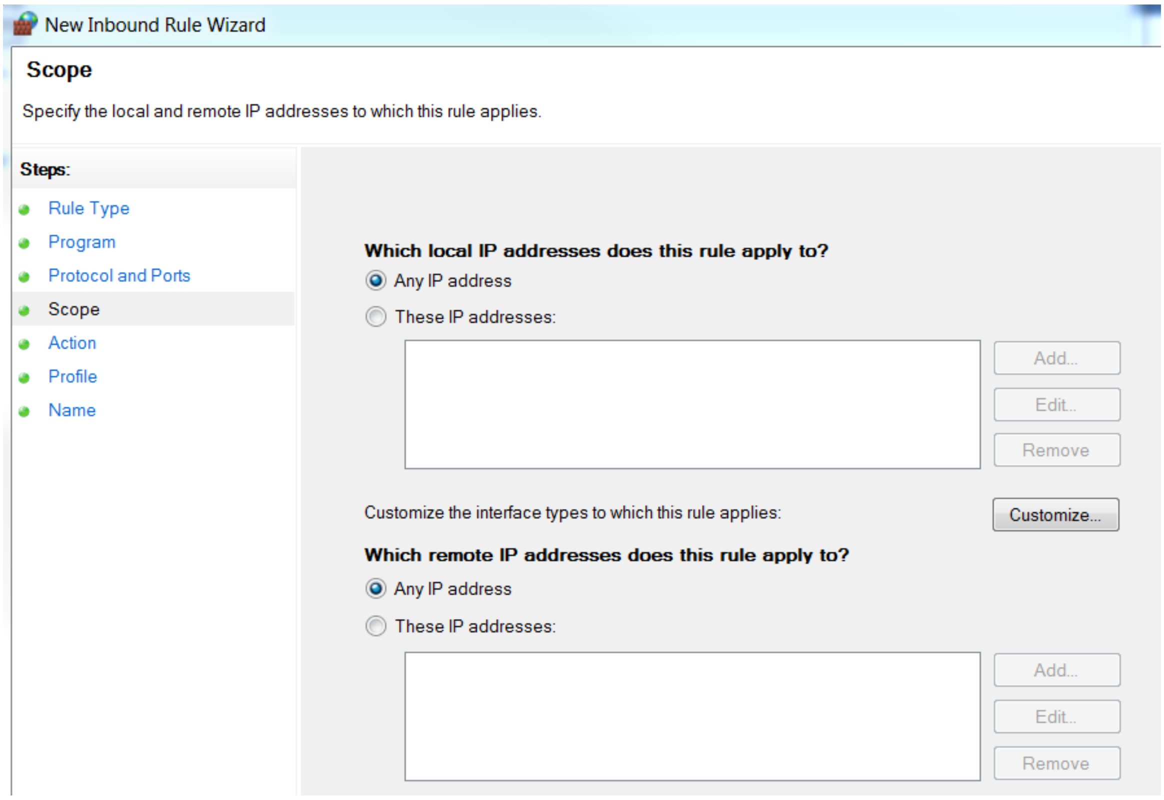
Protocol number:

Local port:

Remote port:

Internet Control Message P
settings:

Select “ICPMv4”, next.



New Inbound Rule Wizard

Scope

Specify the local and remote IP addresses to which this rule applies.

Which local IP addresses does this rule apply to?

● Any IP address

○ These IP addresses:

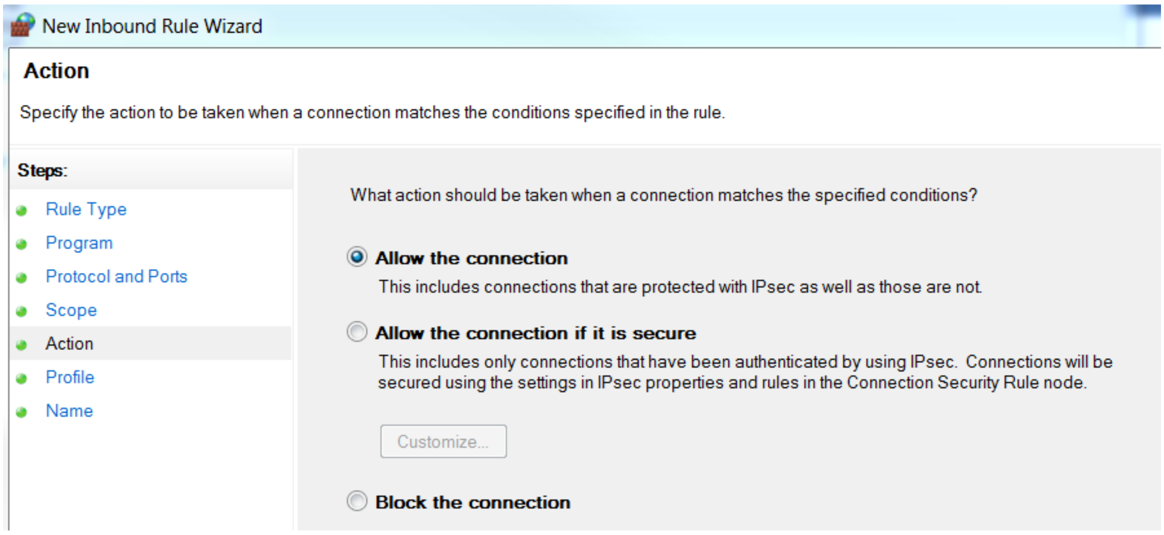
Customize the interface types to which this rule applies:

Which remote IP addresses does this rule apply to?

● Any IP address

○ These IP addresses:

Here you can specify witch IP the rule apply to.



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**

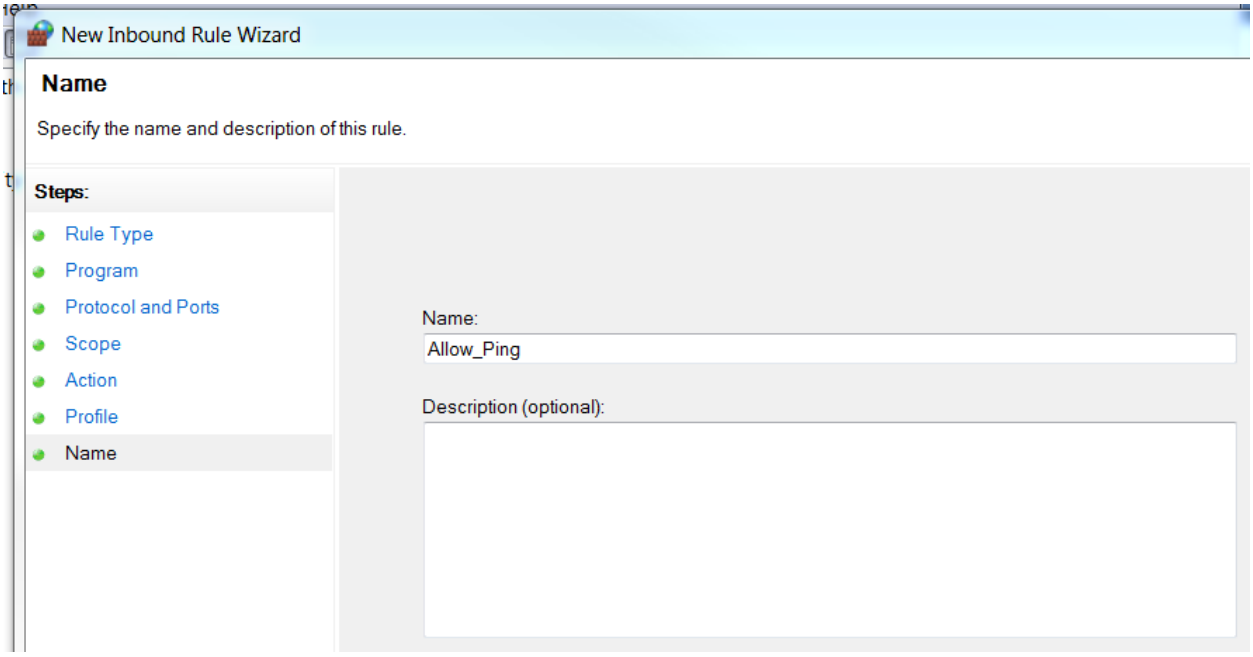
This includes connections that are protected with IPsec as well as those are not.

○ **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**

If the connection not is “allowed” it makes no sense in this case.



Name

Specify the name and description of this rule.

Name: Allow_Ping

Description (optional):

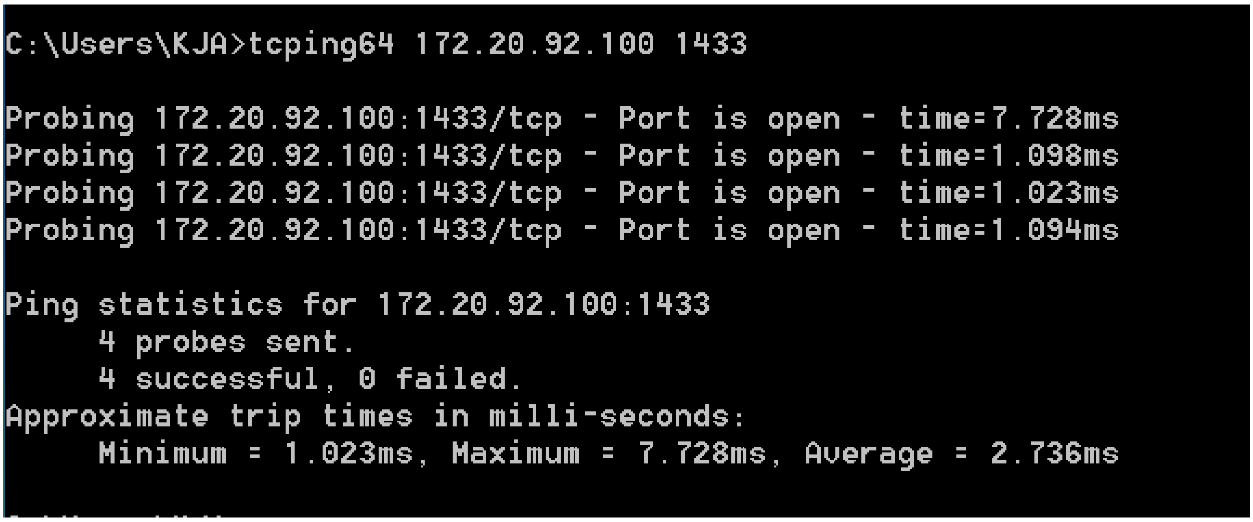
Give the rule a name.

Now you should have the possibility to “Ping” the server IP with the build in “Ping” command.

I want also to check if the port 1433 is available, and I found a free tool on the internet that can check the IP and the Port with the same tool.

The tool is called “Tcping”, just search on the internet.

The tool is running in the “CMD” prompt, just like “Ping”.



```
C:\Users\KJA>tcping64 172.20.92.100 1433

Probing 172.20.92.100:1433/tcp - Port is open - time=7.728ms
Probing 172.20.92.100:1433/tcp - Port is open - time=1.098ms
Probing 172.20.92.100:1433/tcp - Port is open - time=1.023ms
Probing 172.20.92.100:1433/tcp - Port is open - time=1.094ms

Ping statistics for 172.20.92.100:1433
     4 probes sent.
     4 successful, 0 failed.
Approximate trip times in milli-seconds:
     Minimum = 1.023ms, Maximum = 7.728ms, Average = 2.736ms
```

The connection is OK and the port is open.

Create PLCSQL database.

Included in the delivery, there is a script that creates the following in the SQL database.
The default name of the database is “PLCSQL”

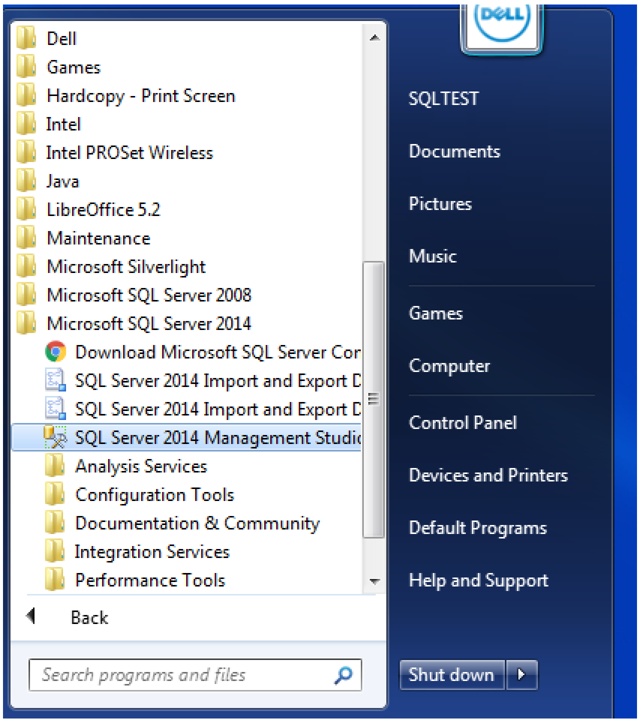
1: Tables for “Log” data from the PLC to SQL.
2: Tables for “Recipe” data from SQL to PLC.
3: Default user “plcsql” (you may change the user before you run the script).
4: Password “link” for user “plcsql” (you may change the Password before you run the script).
5: Permissions to the user “plcsql” to access, read, and write in PLCSQL database. (you may change the name of the database before you run the script)
6: Stored procedures that is used to handle the data flow between SQL and PLC.

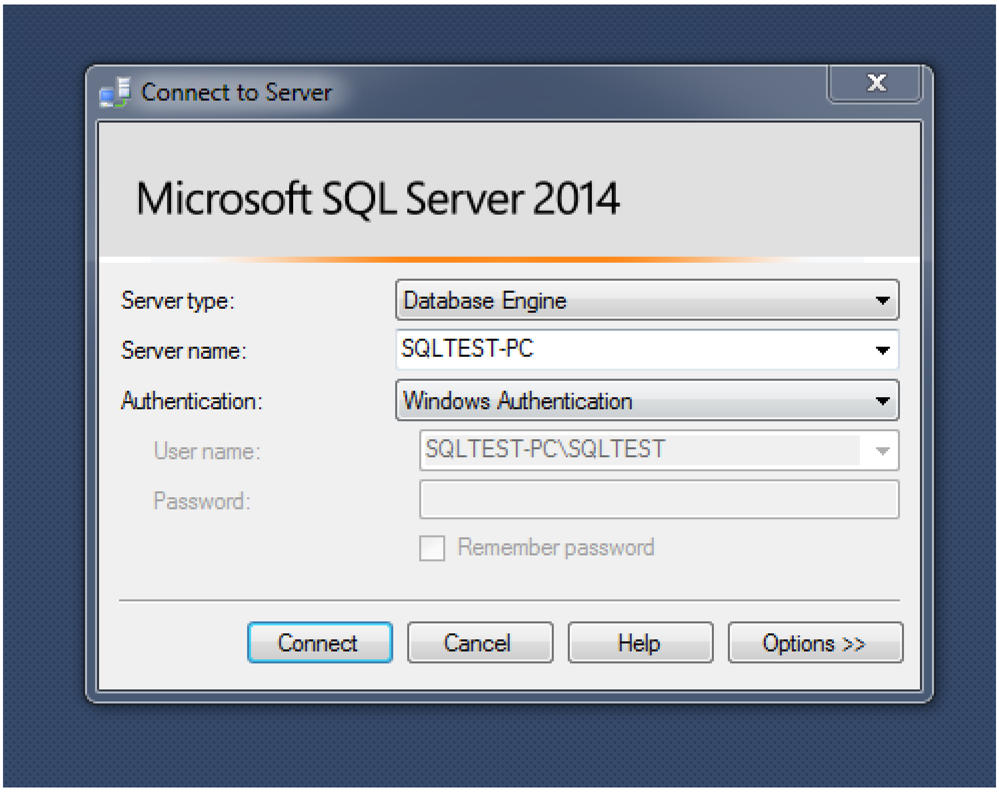
⚠

To Create the database needed for PLCSQL, you will need the program “SQL Server XXXX Management Studio”. If you have chosen the “correct” server installation, then the file is already installed, if not, then you have to download and install “SQL Server Management Studio”. You don’t have to choose the same version of “SQL Server Management Studio”, as your SQL Server installation, a newer version will also run without problems.

The name of the script, is “Create-database-plcsql.sql”, and is included in the delivery.

You can run it by double clicking on the file, or by opening the “SQL Server Management Studio”.





Server type: Database Engine

Server name: SQLTEST-PC

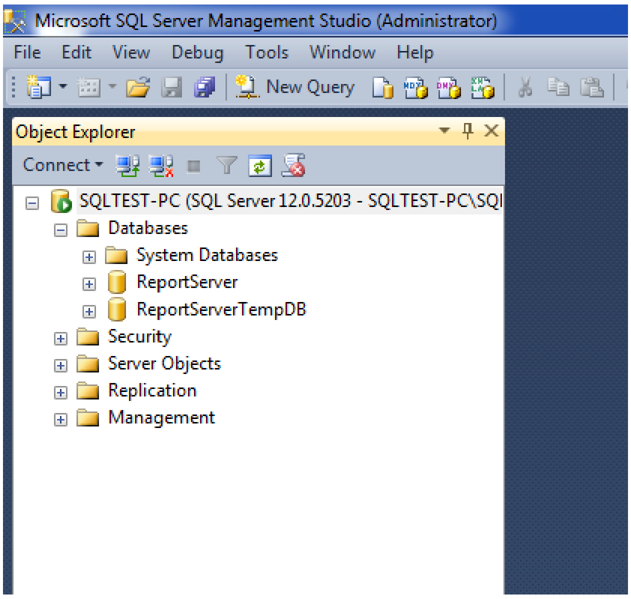
Authentication: Windows Authentication

User name: SQLTEST-PC\SQLTEST

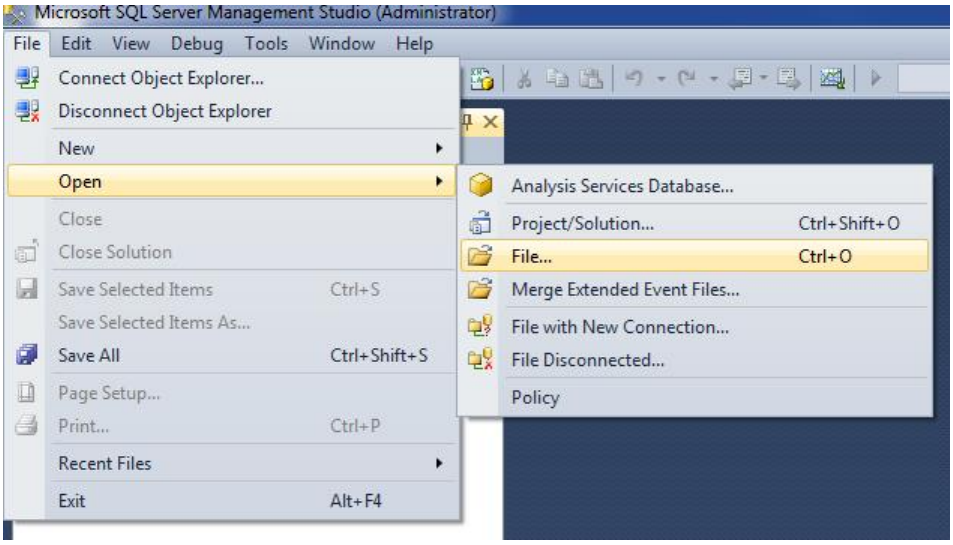
Password:

□ Remember password

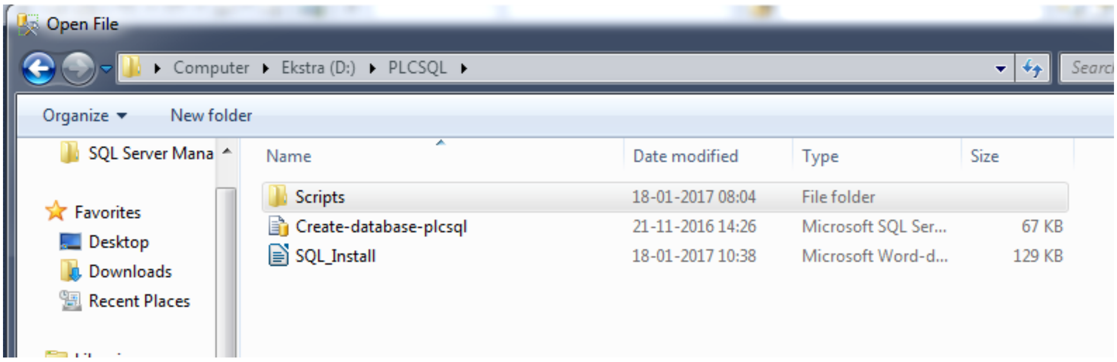
If you open the file “Create-database-plcsql.sql”, or open “SQL Server Management Studio”, then you have to “Connect” to the SQL Server, use “Windows Authentication”, or log in as the user “sa” by selecting “SQL Server Authentication”.


Microsoft SQL Server Management Studio (Administrator)
File Edit View Debug Tools Window Help
New Query
Object Explorer
Connect
SQLTEST-PC (SQL Server 12.0.5203 - SQLTEST-PC\SQL
Databases
System Databases
ReportServer
ReportServerTempDB
Security
Server Objects
Replication
Management

When the Management Studio is opened, you will have a picture like that above.

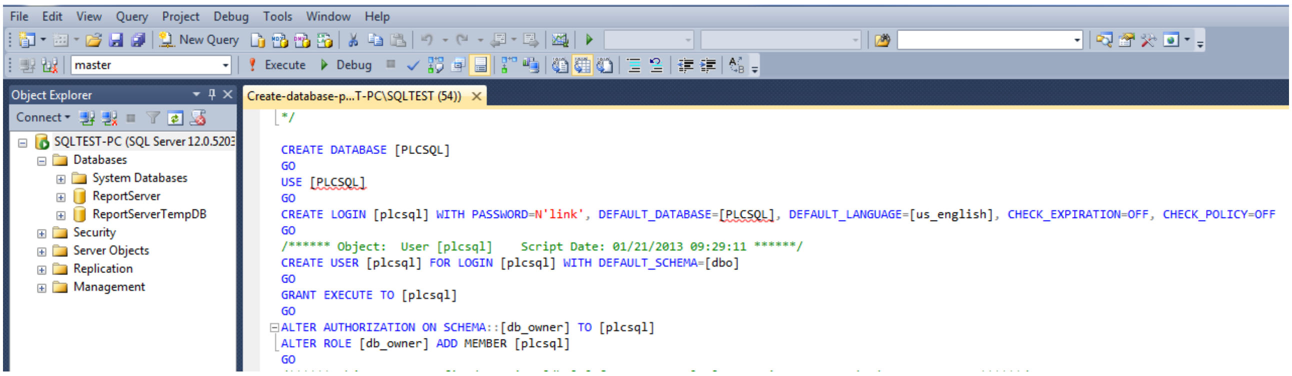

Microsoft SQL Server Management Studio (Administrator)
File Edit View Debug Tools Window Help
Connect Object Explorer...
Disconnect Object Explorer
New
Open
Close
Close Solution
Save Selected Items Ctrl+S
Save Selected Items As...
Save All Ctrl+Shift+S
Page Setup...
Print... Ctrl+P
Recent Files
Exit Alt+F4
Analysis Services Database...
Project/Solution... Ctrl+Shift+O
File... Ctrl+O
Merge Extended Event Files...
File with New Connection...
File Disconnected...
Policy

If you just opened Management Studio, then select “File/Open/File”, to start the script that is necessary to set up the PLCSQL system.



Name	Date modified	Type	Size
Scripts	18-01-2017 08:04	File folder	
Create-database-plcsql	21-11-2016 14:26	Microsoft SQL Ser...	67 KB
SQL_Install	18-01-2017 10:38	Microsoft Word-d...	129 KB

Browse to the location of the file, and select the file and click “Open”



```
*/

CREATE DATABASE [PLCSQL]
GO
USE [PLCSQL]
GO
CREATE LOGIN [plcsql] WITH PASSWORD=N'link', DEFAULT_DATABASE=[PLCSQL], DEFAULT_LANGUAGE=[us_english], CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
GO
/****** Object:  User [plcsql]    Script Date: 01/21/2013 09:29:11 ******/
CREATE USER [plcsql] FOR LOGIN [plcsql] WITH DEFAULT_SCHEMA=[dbo]
GO
GRANT EXECUTE TO [plcsql]
GO
ALTER AUTHORIZATION ON SCHEMA::[db_owner] TO [plcsql]
ALTER ROLE [db_owner] ADD MEMBER [plcsql]
GO
```

Here is the beginning of the script where the “Standard” user is “handled”

In the following line, the user name [plcsql] and the password ‘link’ is set.
CREATE LOGIN [plcsql] WITH PASSWORD=N'link', DEFAULT_DATABASE=[PLCSQL],

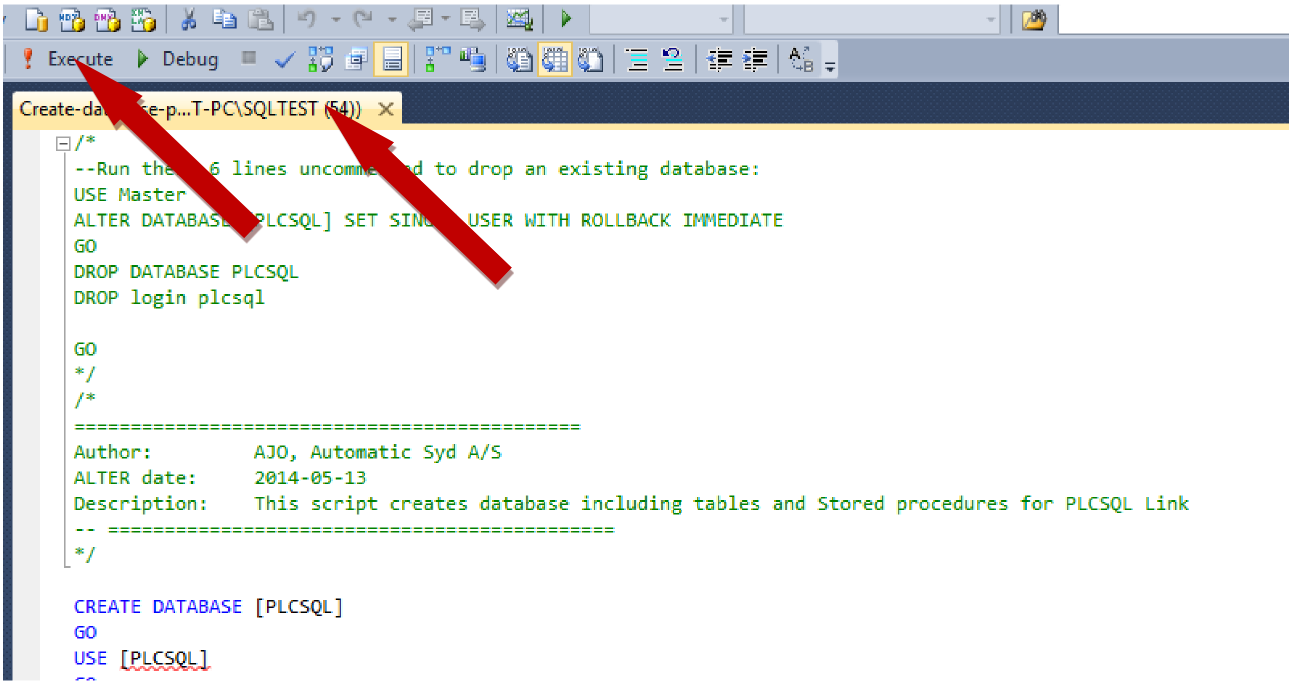
CAUTION! It is the USER that decides which data base there is connected to..

You can of course use your own name(s) and password(s).

Permission to user “plcsql” to “run” procedures.
GRANT EXECUTE TO [plcsql]

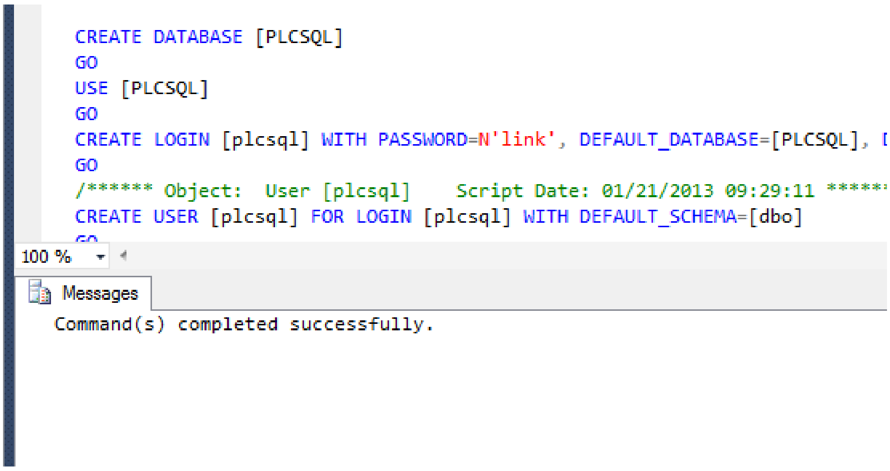
Give the user “plcsql” the “right” to the data base (read and write), and set user “plcsql” as a user of this data base “PLCSQL”
ALTER AUTHORIZATION ON SCHEMA::[db_owner] TO [plcsql]
ALTER ROLE [db_owner] ADD MEMBER [plcsql]

If you don’t want the script to generate the “default” user, you have to “Comment” the lines that are generating the user. “/*” start a comment “*/” end a comment, you can use this over several lines.



```
/*
--Run the  6 lines uncomm   d to drop an existing database:
USE Master
ALTER DATABAS  PLCSQL] SET SIN   USER WITH ROLLBACK IMMEDIATE
GO
DROP DATABASE PLCSQL
DROP login plcsql

GO
*/
/*
=============================================
Author:         AJO, Automatic Syd A/S
ALTER date:     2014-05-13
Description:    This script creates database including tables and Stored procedures for PLCSQL Link
-- =============================================
*/

CREATE DATABASE [PLCSQL]
GO
USE [PLCSQL]
```

To run the opened script, press “Execute”.
If the “Execute” is not selectable, then click on the yellow line in the top of the script where the name of the script is shown.

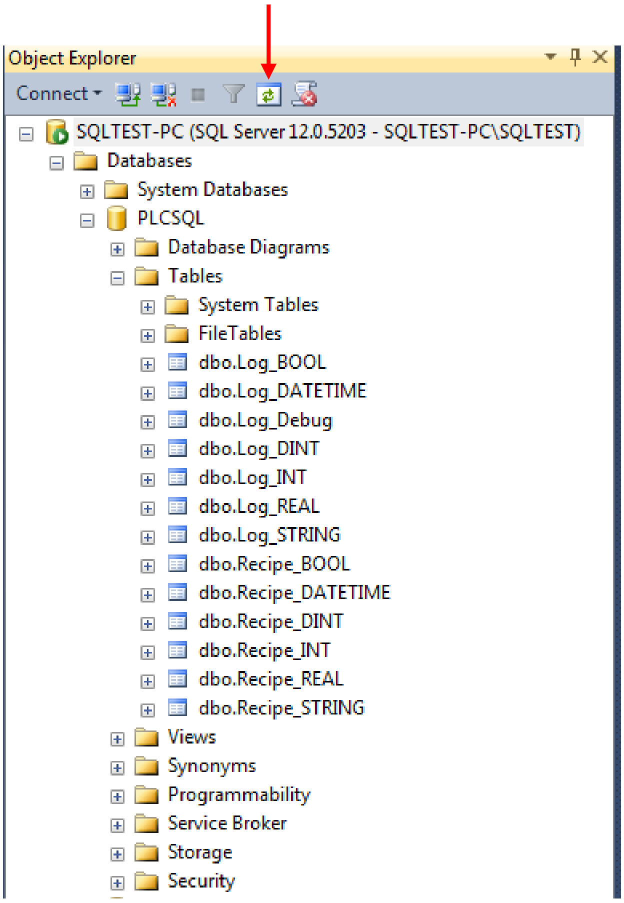


```
CREATE DATABASE [PLCSQL]
GO
USE [PLCSQL]
GO
CREATE LOGIN [plcsql] WITH PASSWORD=N'link', DEFAULT_DATABASE=[PLCSQL], I
GO
/****** Object:  User [plcsql]    Script Date: 01/21/2013 09:29:11 ******
CREATE USER [plcsql] FOR LOGIN [plcsql] WITH DEFAULT_SCHEMA=[dbo]
```

```
Command(s) completed successfully.
```

When the creation of the data base is finish, you should get this message.

Then you have to press “F5” or press the “Update” icon to get the following picture.

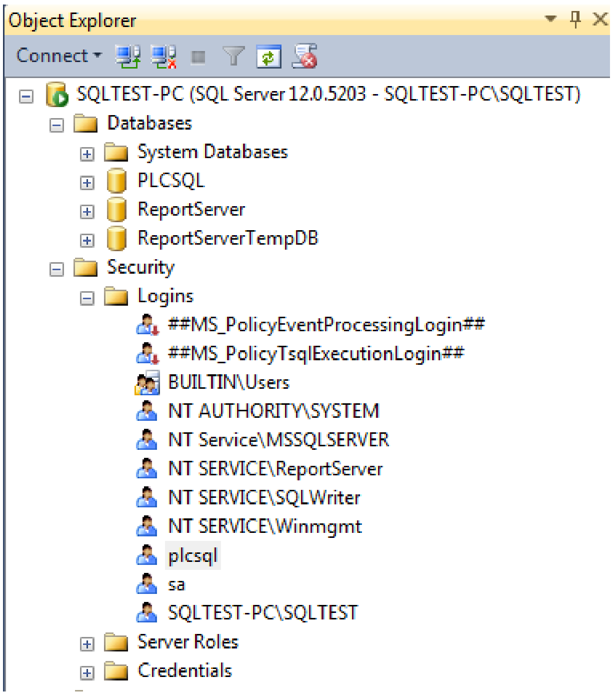


- SQLTEST-PC (SQL Server 12.0.5203 - SQLTEST-PC\SQLTEST)
  - Databases
    - System Databases
    - PLCSQL
      - Database Diagrams
      - Tables
        - System Tables
        - FileTables
        - dbo.Log_BOOL
        - dbo.Log_DATETIME
        - dbo.Log_Debug
        - dbo.Log_DINT
        - dbo.Log_INT
        - dbo.Log_REAL
        - dbo.Log_STRING
        - dbo.Recipe_BOOL
        - dbo.Recipe_DATETIME
        - dbo.Recipe_DINT
        - dbo.Recipe_INT
        - dbo.Recipe_REAL
        - dbo.Recipe_STRING
      - Views
      - Synonyms
      - Programmability
      - Service Broker
      - Storage
      - Security

This an overview of the “tags” that we are using in PLCSQL.

The “dbo.Log.BOOL; DINT;INT;REAL;STRING, is “WRITE” only from the PLC.
The “dbo.Log.DATETIME” is updated automatically every time you WRITE.
The “dbo.Log.Debug” is only for testing new functions, and is normally not used.

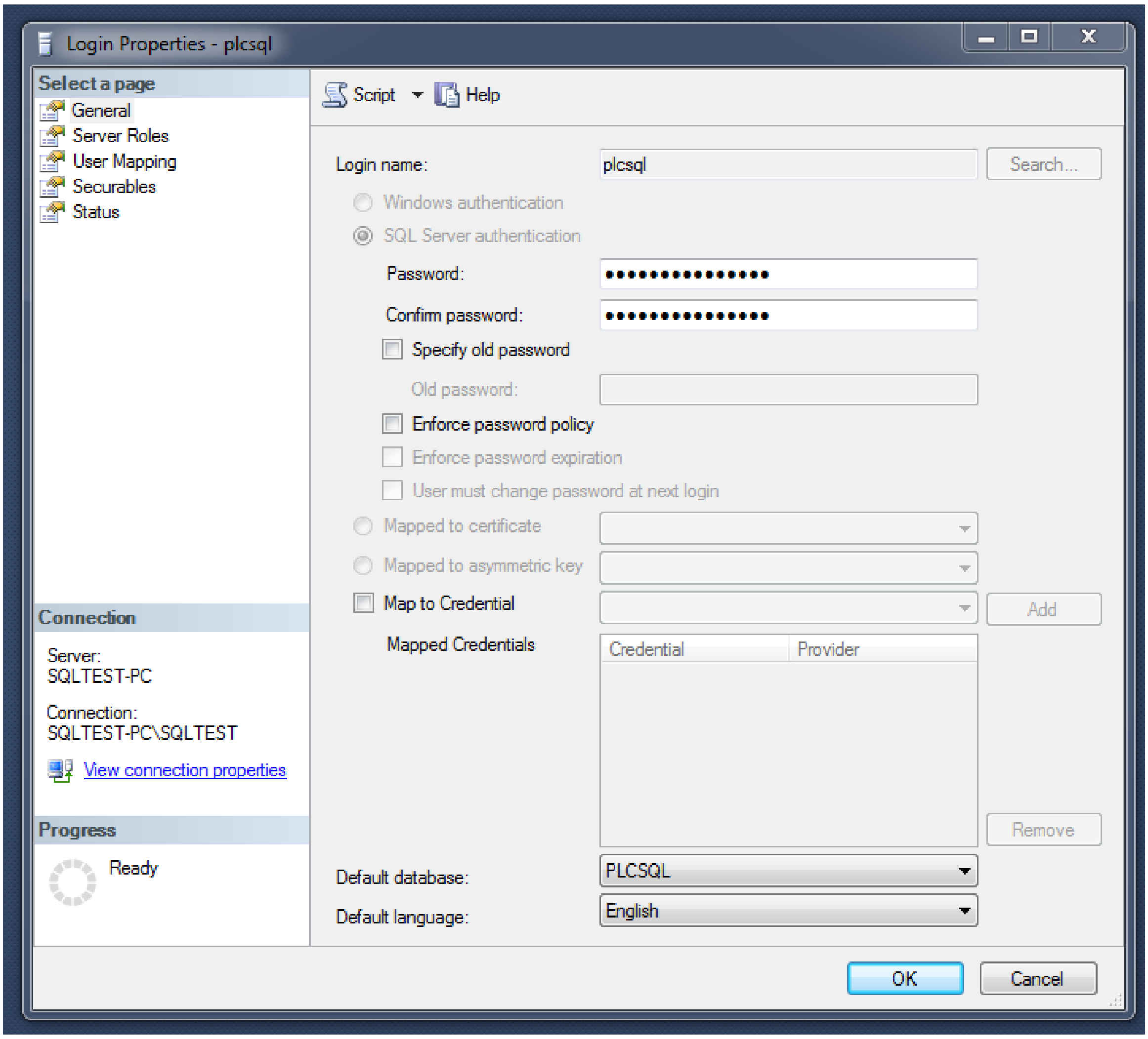
The “dbo.Recipe.BOOL; DINT; INT; REAL; STRING, is READ and WRITE from the PLC.
The “dbo.Recipe.DATETIME” is updated automatically every time there is a change in the data.
????.



- SQLTEST-PC (SQL Server 12.0.5203 - SQLTEST-PC\SQLTEST)
  - Databases
    - System Databases
    - PLCSQL
    - ReportServer
    - ReportServerTempDB
  - Security
    - Logins
      - ##MS_PolicyEventProcessingLogin##
      - ##MS_PolicyTsqlExecutionLogin##
      - BUILTIN\Users
      - NT AUTHORITY\SYSTEM
      - NT Service\MSSQLSERVER
      - NT SERVICE\ReportServer
      - NT SERVICE\SQLWriter
      - NT SERVICE\Winmgmt
      - plcsql
      - sa
      - SQLTEST-PC\SQLTEST
    - Server Roles
    - Credentials

You can check the user by selecting “Security/Logins”

You can see that the “Standard” user “plcsql” is created.



Login name: plcsql

○ Windows authentication

● SQL Server authentication

Password: •••••••••••••••

Confirm password: •••••••••••••••

□ Specify old password

Old password:

□ Enforce password policy

□ Enforce password expiration

□ User must change password at next login

○ Mapped to certificate

○ Mapped to asymmetric key

□ Map to Credential

Mapped Credentials

Default database: PLCSQL

Default language: English

Connection

Server:
SQLTEST-PC

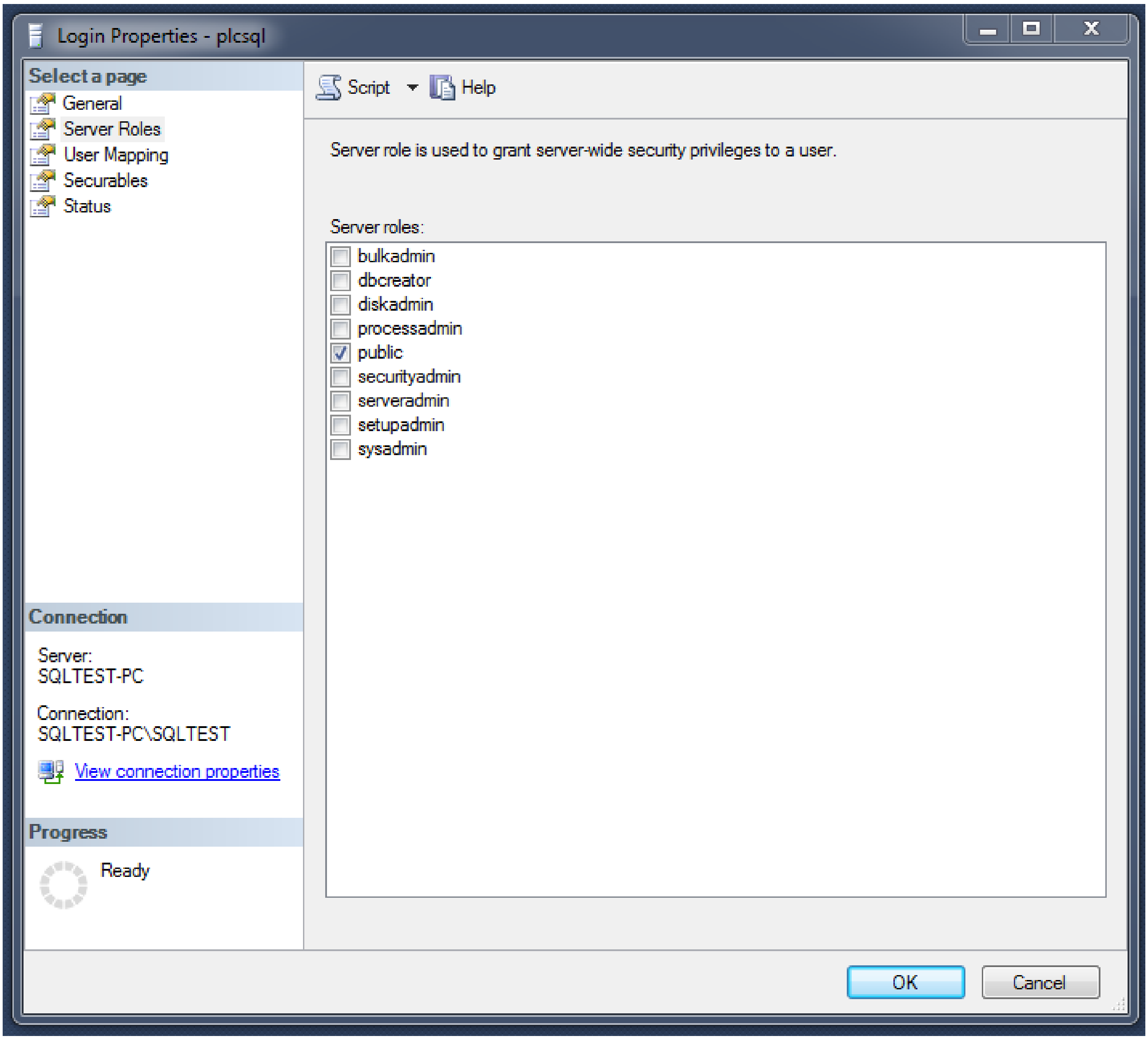
Connection:
SQLTEST-PC\SQLTEST

Progress

Ready

You can check the properties of the user by double click on the “user”

In the “General” properties, you can change the password.
DON’T use “Enforce password policy”, that will only give you problems.



Server role is used to grant server-wide security privileges to a user.

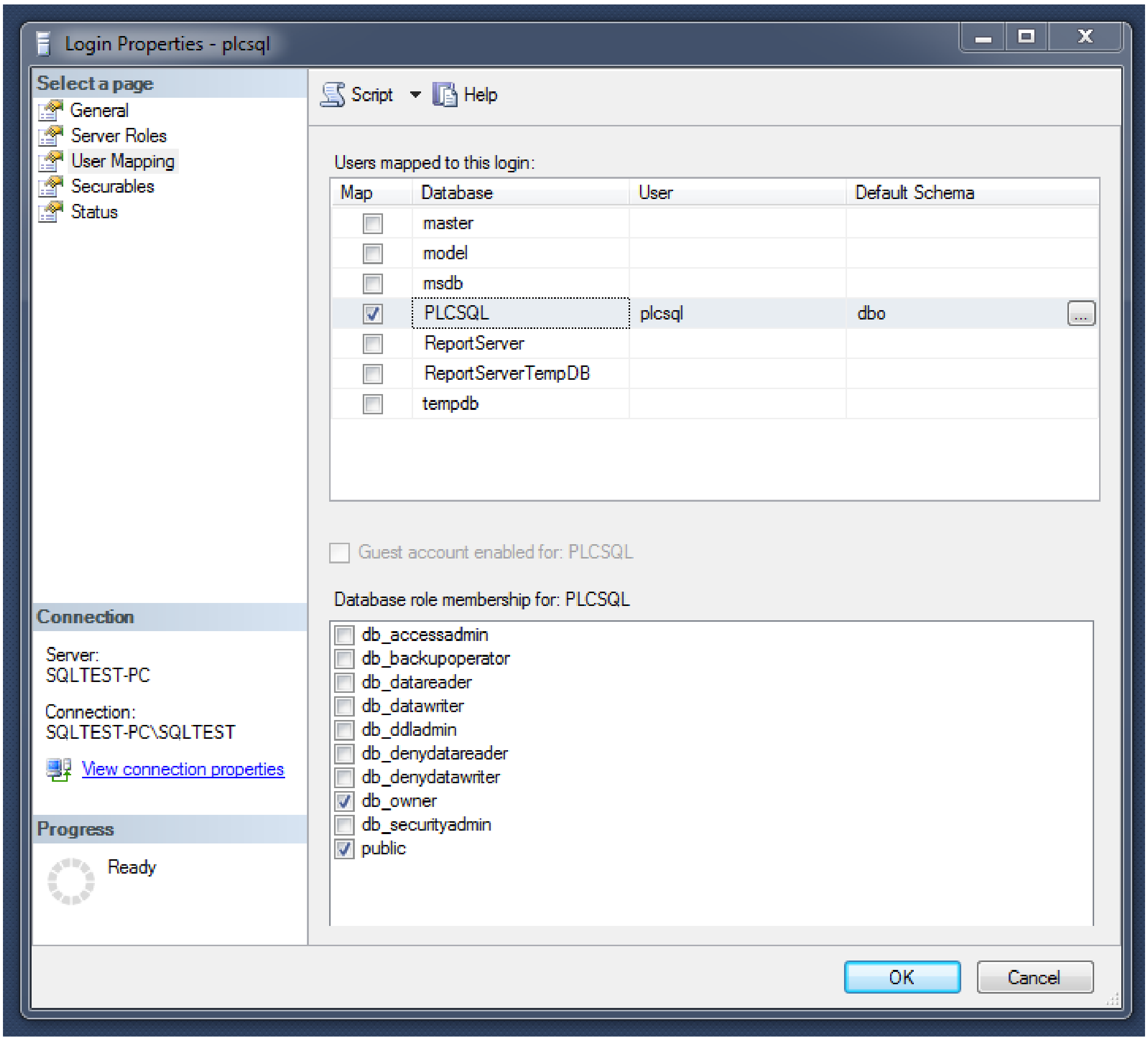
Server roles:

- □ bulkadmin
- □ dbcreator
- □ diskadmin
- □ processadmin
- ☑ public
- □ securityadmin
- □ serveradmin
- □ setupadmin
- □ sysadmin

Server:
SQLTEST-PC

Connection:
SQLTEST-PC\SQLTEST

“Server Roles”, don’t change if you don’t know what you are doing.



Server:
SQLTEST-PC

Connection:
SQLTEST-PC\SQLTEST

Users mapped to this login:

Map	Database	User	Default Schema
□	master		
□	model		
□	msdb		
☑	PLCSQL	plcsql	dbo
□	ReportServer		
□	ReportServerTempDB		
□	tempdb		

□ Guest account enabled for: PLCSQL

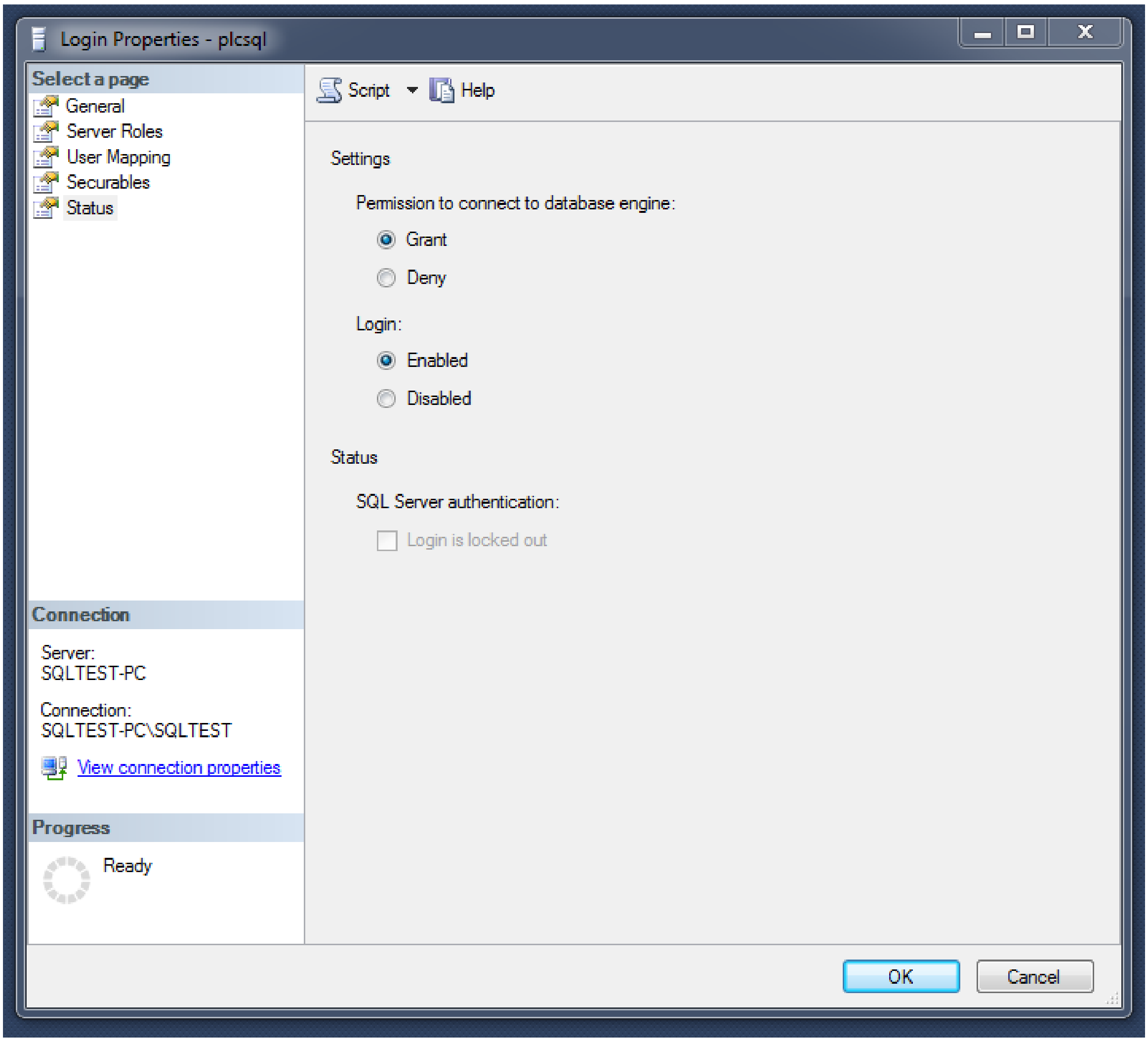
Database role membership for: PLCSQL

- □ db_accessadmin
- □ db_backupoperator
- □ db_datareader
- □ db_datawriter
- □ db_ddladmin
- □ db_denydatareader
- □ db_denydatawriter
- ☑ db_owner
- □ db_securityadmin
- ☑ public

User Mapping”, in this case is selected “db_owner”, that covers all. ????????
As minimum the following has to be selected.
“public”
“db_datareader”
“db_datawriter”

!!!!!! CAUTION it is the USER that decides which data base there is connected to. !!!!!!

When the user is logged in to the data base, the user is connected the selected data base.



Settings

Permission to connect to database engine:

● Grant

○ Deny

Login:

● Enabled

○ Disabled

Status

SQL Server authentication:

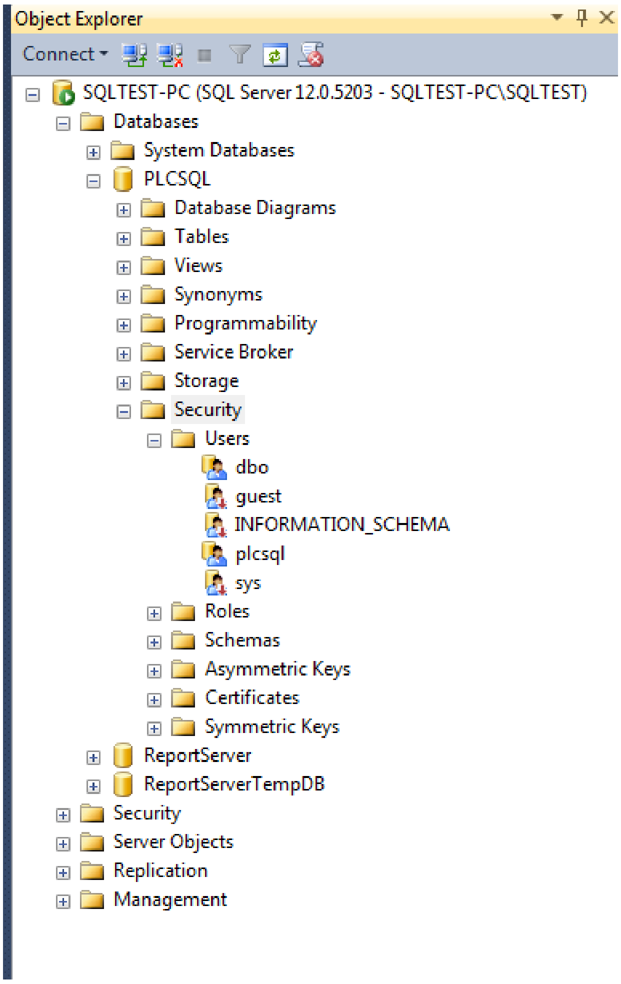
□ Login is locked out

Server:
SQLTEST-PC

Connection:
SQLTEST-PC\SQLTEST

Ready

If the user cannot connect to the data base or login to the data base, the system will not run.



Object Explorer

Connect ▾

```
⊟ SQLTEST-PC (SQL Server 12.0.5203 - SQLTEST-PC\SQLTEST)
    ⊟ Databases
        ⊞ System Databases
        ⊟ PLCSQL
            ⊞ Database Diagrams
            ⊞ Tables
            ⊞ Views
            ⊞ Synonyms
            ⊞ Programmability
            ⊞ Service Broker
            ⊞ Storage
            ⊟ Security
                ⊟ Users
                      dbo
                      guest
                      INFORMATION_SCHEMA
                      plcsql
                      sys
                ⊞ Roles
                ⊞ Schemas
                ⊞ Asymmetric Keys
                ⊞ Certificates
                ⊞ Symmetric Keys
        ⊞ ReportServer
        ⊞ ReportServerTempDB
    ⊞ Security
    ⊞ Server Objects
    ⊞ Replication
    ⊞ Management
```

The user must also be present in the “PLCSQL/Security/Users” folder, is done automatically.

This ends the installation and setup of the SQL server.

System parameters.

There are 3 “system parameters” placed in “dbo.Log” and “dbo.Recipe”
In both tables it is the following 3 parameters.

10001:	SetCount	Data type “Int”
15001:	SetID	Data type “Dint”
30001:	DateTimeStamp.	Data type “String”

The “SetCount” parameter 10001 contains the number of parameters excluding parameter 10001,15001, and 30001 in “this” actual “dataset”.

The “SetID” parameter 15001, contains the unique number that every “dataset” get when something is stored in the SQL database. The “SetID” number changes only when something is stored in the SQL database, e.g. you trigger the “Log data” function from the PLC or you generate a new “Recipe” in the SQL database. Every time you save a “dataset” in the SQL database, all parameters in this specific “dataset” will get the same “SetID” number, it is the “SetID” number that “connects” all the parameters in this specific “dataset” together.

The “DateTimeStamp” contains the date and time when this “dataset” was stored in the SQL database.

Appendix, Code Snippets

Attention! Some the following code examples are not intended to use on a “Company” data base, they are intended to use on a local “test” data base, so you can find out how the system is working.

Generate a “Recipe” in the SQL data base.

Normally the “Recipe” data will be generated elsewhere in the “system”, but for testing purposes you can generate your own recipes.

```
use PLCSQL

EXEC sp_SaveParams 1, N'Recipe'N'1;1.111;10002;1;15002;11111111;30003;HelloWorld 1'
```

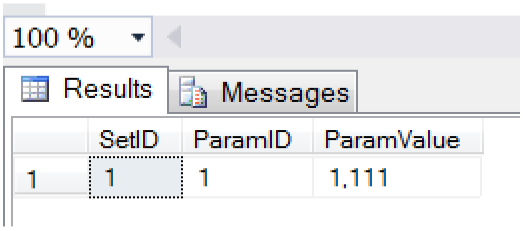
Explanation
`sp_SaveParams,` is the name of the “Procedure” that saves the data
“1”, is a “user” number, the number is mandatory, but you can change the value.
“N” sets the character set to Unicode.
‘Recipe’ is the “Table” we are using.
“1;1.111” the first “1” is the parameter number, the “1.111” is the value to place in the parameter
“10002;1” is again parameter number “100002” and value “1”
“15002;11111111” is again parameter number “15002”, and value “11111111”
“30003;HelloWorld 1” is again parameter number “30003”, and value “HelloWorld 1”
All parameter and values are typed as “strings”, the “” at start and end defines a string.

Type definitions

1..9999	REAL type.
10001..14999	INT type, signed.
15001..19999	DINT type, signed
20001..29999	BOOL type
30001..30999	STRING types, length 1..254 characters.

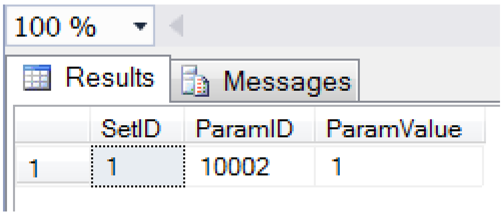
CAUTION
Parameter number 10001, 15001, and 30001 are used internally and may not be written to by the user.

Output from the PLCSQL data base



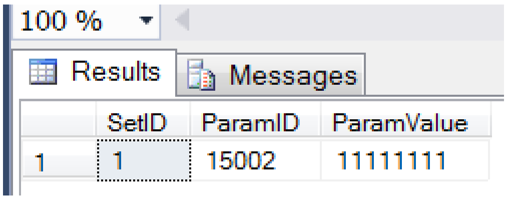
	SetID	ParamID	ParamValue
1	1	1	1,111

Table “dbo.Recipe_REAL”
The “SetID” is “1”



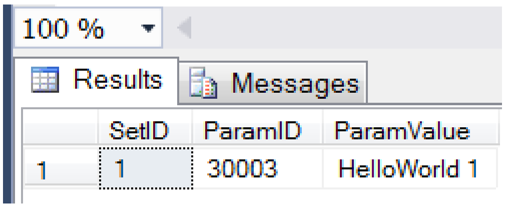
	SetID	ParamID	ParamValue
1	1	10002	1

Table “dbo.Recipe_INT”
The “SetID” is “1”



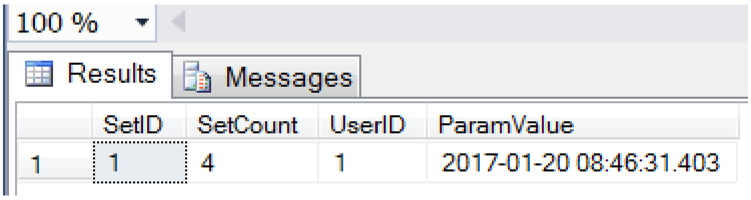
	SetID	ParamID	ParamValue
1	1	15002	11111111

Table “dbo.Recipe_DINT”
The “SetID” is “1”



	SetID	ParamID	ParamValue
1	1	30003	HelloWorld 1

Table “dbo.Recipe_STRING”
The “SetID” is “1”



	SetID	SetCount	UserID	ParamValue
1	1	4	1	2017-01-20 08:46:31.403

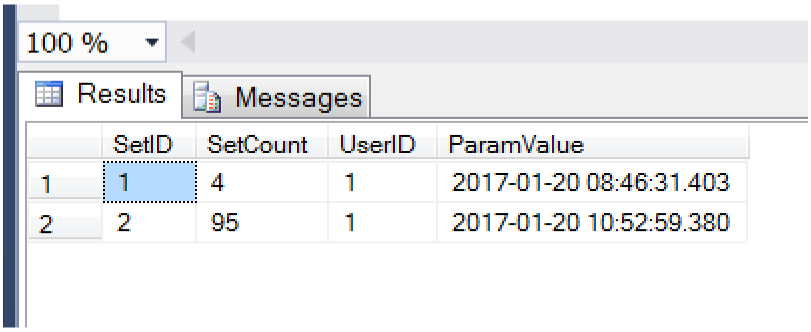
Table “dbo.Recipe_DATETIME”
The “SetID” is “1”
“SetCount” is 4, meaning we have 4 entries in this recipe.
“UserID” is 1
“ParamValue” the date and time where this recipe was stored in the data base.

If you now from the PLC run the following command “GetParamSet;30003;HelloWorld 1”,(syntax not correct, only shown for better understanding).
Then you will get the 4 parameter’s in this recipe. You can choose any of the 4 parameter in the “GetParamSet” command, because all other parameters with the same “SetID” will be send to the PLC.
Next time you store the same recipe or another recipe, the “SetID” will change (+1)

If you want to generate “Big” recipes, the following is an example to do that.
All parameters will get the same “SetID” and the same “Timestamp”

```
use PLCSQL
/* Declare variable*/
declare @params nvarchar(max)
/* Following generates 1 "BIG" recipe*/

/* Remember that Parameter 10001, 15001, and 30001 is used by the system, NO USER access
*/

/* First 9 */
set @params = N'1;1.1; 10002;10002; 15002;15002; 20001;1; 30002;This is string 30002;'
set @params += N'2;2.2; 10003;10003; 15003;15003; 20002;1; 30003;This is string 30003;'
set @params += N'3;3.3; 10004;10004; 15004;15004; 20003;1; 30004;This is string 30004;'
set @params += N'4;4.4; 10005;10005; 15005;15005; 20004;1; 30005;This is string 30005;'
set @params += N'5;5.5; 10006;10006; 15006;15006; 20005;1; 30006;This is string 30006;'
set @params += N'6;6.6; 10007;10007; 15007;15007; 20006;1; 30007;This is string 30007;'
set @params += N'7;7.7; 10008;10008; 15008;15008; 20007;1; 30008;This is string 30008;'
set @params += N'8;8.8; 10009;10009; 15009;15009; 20008;1; 30009;This is string 30009;'
set @params += N'9;9.9; 10010;10010; 15010;15010; 20009;1; 30010;This is string 30010;'

/* 10 - 19*/
set @params += N'10;10.10; 10011;10011; 15011;15011; 20011;1; 30011;This is string 30011;'
set @params += N'11;11.11; 10012;10012; 15012;15012; 20012;1; 30012;This is string 30012;'
set @params += N'12;12.12; 10013;10013; 15013;15013; 20013;1; 30013;This is string 30013;'
set @params += N'13;13.13; 10014;10014; 15014;15014; 20014;1; 30014;This is string 30014;'
set @params += N'14;14.14; 10015;10015; 15015;15015; 20015;1; 30015;This is string 30015;'
set @params += N'15;15.15; 10016;10016; 15016;15016; 20016;1; 30016;This is string 30016;'
set @params += N'16;16.16; 10017;10017; 15017;15017; 20017;1; 30017;This is string 30017;'
set @params += N'17;17.17; 10018;10018; 15018;15018; 20018;1; 30018;This is string 30018;'
set @params += N'18;18.18; 10019;10019; 15019;15019; 20019;1; 30019;This is string 30019;'
set @params += N'19;19.19; 10020;10020; 15020;15020; 20020;1; 30020;This is string 30020;'


EXEC sp_SaveParams 1, N'Recipe', @params
```

Again here, if you now from the PLC run the following command “GetParamSet;10020;10020”, (syntax not correct, only shown for better understanding).
Then you will get all 95 (19 lines X 5 parameters each line) parameters’ in this recipe. You can choose any of the 95 parameter in the “GetParamSet” command, because all other parameters with the same “SetID” will be send to the PLC.



	SetID	SetCount	UserID	ParamValue
1	1	4	1	2017-01-20 08:46:31.403
2	2	95	1	2017-01-20 10:52:59.380

After running the script, I here show you the “DATETIME” parameter, as you see, SetID = 2 SetCount = 95 parameters with SetID = 2.

!!!!!!!!!!!!!!!!!!!! Attention, the following 3 example's will result in DATALOSS !!!!!!!!!!!!!!

Delete the CONTENT of the "Log" tables.

```
use PLCSQL
truncate table Log_string
truncate table Log_bool
truncate table Log_dint
truncate table Log_int
truncate table Log_real
truncate table Log_datetime
```

Delete the CONTENT of the "Recipe" tables.

```
use PLCSQL
truncate table Recipe_string
truncate table Recipe_bool
truncate table Recipe_dint
truncate table Recipe_int
truncate table Recipe_real
truncate table Recipe_datetime
```

Delete the PLCSQL data base and the user "plcsql"

```
USE Master
ALTER DATABASE [PLCSQL] SET SINGLE_USER WITH ROLLBACK IMMEDIATE
GO
DROP DATABASE PLCSQL
DROP login plcsql

GO
```

The following “Function” can be used to test the performance of the PLC and the data base.

```
USE [PLCSQL]
GO
/****** Object:  StoredProcedure [dbo].[sp_SaveParamsN]    Script Date: 01/13/2017 12:41:52
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
--Dummy for performance test  AJO / KJA
-- =============================================
CREATE PROCEDURE [dbo].[sp_SaveParamsN]
@UserID INT, --Current user
@TablePrefix NVARCHAR(100), --String argument containing table prefix name (Log, Recipe,
Setup, User)
@tstring NVARCHAR(MAX) --String argument containing Array of ParamID,ParamValue
AS
SET NOCOUNT ON;
DECLARE @ParamID NVARCHAR(MAX)
DECLARE @ParamValue NVARCHAR(MAX)
DECLARE @SetID INT
DECLARE @TableSuffix NVARCHAR(MAX)
DECLARE @QueryString NVARCHAR(MAX)
DECLARE @TotalRows INT
DECLARE @delpos INT
DECLARE @MinREAL INT
DECLARE @MaxREAL INT
DECLARE @MinINT INT
DECLARE @MaxINT INT
DECLARE @MinBOOL INT
DECLARE @MaxBOOL INT
DECLARE @MinSTRING INT
DECLARE @MaxSTRING INT
DECLARE @MinDINT INT
DECLARE @MaxDINT INT
DECLARE @SetCount INT

--Output SetID,SetCount and TimeStamp to Client
SELECT CAST(dbo.ufn_getsetupvalue('SetID') AS INT) AS ParamID, @SetID AS ParamValue
SELECT CAST(dbo.ufn_getsetupvalue('TimeStamp') AS INT) AS ParamID,
CONVERT(VARCHAR(254), GETDATE(), 120) AS ParamValue --Formats DateTime as 2011-
10-15 12:00:00
SELECT CAST(dbo.ufn_getsetupvalue('SetCount') AS INT)  AS ParamID, CAST(@SetCount
AS SMALLINT) AS ParamValue
```

Function “SaveParamsN”, can be used to make some “performance” test. The function DON’T save anything in the data base, but just answers the PLC, so the PLC think that everything is OK, in this way there is a possibility to see what time the PLC uses to generate the data, and what time the data base uses to handle the request.