

AADvance Controller

Catalog Numbers T9110, T9300, T9310, T9401/2, T9431/2, T9451, T9481/2

Allen-Bradley
by ROCKWELL AUTOMATION

Configuration Guide Workbench R2.x Original Instructions

Important User Information

Read this document and the documents listed in the additional resources section about installation, configuration, and operation of this equipment before you install, configure, operate, or maintain this product. Users are required to familiarize themselves with installation and wiring instructions in addition to requirements of all applicable codes, laws, and standards.

Activities including installation, adjustments, putting into service, use, assembly, disassembly, and maintenance are required to be carried out by suitably trained personnel in accordance with applicable code of practice.

If this equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.

WARNING: Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.

ATTENTION: Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence.

IMPORTANT Identifies information that is critical for successful application and understanding of the product.

Labels may also be on or inside the equipment to provide specific precautions.

SHOCK HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.

BURN HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.

ARC FLASH HAZARD: Labels may be on or inside the equipment, for example, a motor control center, to alert people to potential Arc Flash. Arc Flash will cause severe injury or death. Wear proper Personal Protective Equipment (PPE). Follow ALL Regulatory requirements for safe work practices and for Personal Protective Equipment (PPE).

About This Publication

This software technical manual defines how to configure an AADvance® controller

using the AADvance® Workbench software version 2.x to satisfy your system operation and application requirements.

In no event will Rockwell Automation be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment. The examples given in this manual are included solely for illustrative purposes. Because of the many variables and requirements related to any particular installation, Rockwell Automation does not assume responsibility or reliability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, with respect to use of information, circuits, equipment, or software described in this manual.

All trademarks are acknowledged.

Disclaimer

It is not intended that the information in this publication covers every possible detail about the construction, operation, or maintenance of a control system installation. You should also refer to your own local (or supplied) system safety manual, installation and operator/maintenance manuals.

Revision And Updating Policy

This document is based on information available at the time of its publication. The document contents are subject to change from time to time. The latest versions of the manuals are available at the Rockwell Automation Literature Library under "Product Information" , “Critical Process Control & Safety Systems".

Downloads

The product compatibility and download center is rok.auto/pcdc.

Select the Find Downloads option under Download.

In the Product Search field enter "AADvance" and the AADvance option is displayed.

Double-click on the AADvance option and the latest version is shown.

Select the latest version and download the latest version.

AADvance Release

This technical manual applies to AADvance system release 1.40

Latest Product Information

For the latest information about this product review the Product Notifications and Technical Notes issued by technical support. Product Notifications and product support are available at the Rockwell Automation Support Center at rok.auto/knowledgebase.

At the Search Knowledgebase tab select the option By Product then scroll down and select the ICS Triplex® product AADvance.

Some of the Answer IDs in the Knowledge Base require a TechConnect℠ Support Contract. For more information about TechConnect Support Contract Access Level and Features, click this link: Knowledgebase Document ID: IP622 - TechConnect Support Contract - Access Level & Features.

This will get you to the login page where you must enter your login details.

IMPORTANT A login is required to access the link. If you do not have an account then you can create one using the "Sign Up" link at the top right of the web page.

Purpose Of This Manual

Starting with a new panel and a software distribution DVD, this manual explains how to install the AADvance Workbench. This manual defines the process you must follow to configure your system. It includes background information and step-by-step instructions for the following:

- Installing the software and setting up the software licensing
- Setting the controller IP Address and achieving communications with the controller
- Defining the processor configuration
- Defining the application variables
- Configuring the I/O modules and channels
- Setting up MODBUS, CIP and SNCP

The instructions are based on using Release 2.1 of the AADvance Workbench under the Windows XP operating system; other Windows operating systems are supported - refer to the topic Chapter 2 "Planning the AADvance Workbench Software Installation" for a full list of supported windows operating systems.

This manual includes reference information about module status parameters and I/O variables, to help you make a decision about which variable types to use and full descriptions of the data values supplied by the I/O modules.

- If the information in this document does not agree with the applicable project codes and standards, the system integrator must find a solution for the mismatch.
- If AADvance is used for a safety function, the system integrator must apply

Who Should Use Manual

The data in this manual is written for system integrators who know about building and setting up new systems. The system integrator must make sure that the system complies with the local, national and international standards for the application that AADvance is being used for.

Environmental Compliance

Rockwell Automation maintains current product environmental information on its website at rok.auto/pec.

Download Firmware, AOP, EDS, and Other Files

Download firmware, associated files (such as AOP, EDS, and DTM), and access product release notes from the Product Compatibility and Download Center at rok.auto/pcdc.

Summary of Changes

This publication contains the following new or updated information. This list includes substantive updates only and is not intended to reflect all changes.

Global changes

This table identifies changes that apply to all information about a subject in the manual and the reason for the change. For example, the addition of new supported hardware, a software design change, or additional reference material would result in changes to all of the topics that deal with that subject.

Subject	Reason
Applied latest publication template	Marketing product change

New or enhanced features

This table contains a list of topics changed in this version, the reason for the change, and a link to the topic that contains the changed information.

Subject	Reason
AADvance Release on page 3	Updated AADvance system release information.
Additional Resources on page 5	Added reference to AADvance®-Trusted® SIS Workstation software User Guide, publication ICSTT-UM002.
N on page 235	Removed NFPA 87.
proof test on page 236	Updated definition.

Additional Resources

These documents contain additional information concerning related products from Rockwell Automation.

Resource	Description
AADvance Controller Safety Manual, ICSTT-RM446	This technical manual defines how to safely apply AADvance controllers for a Safety Instrumented Function. It sets out standards (which are mandatory) and makes recommendations to make sure that installations satisfy and maintain their required safety integrity level.
AADvance Controller System Build Manual, ICSTT-RM448	This technical manual describes how to assemble a system, switch on and validate the operation of your system.
AADvance Controller Configuration Guide Workbench 1.x, ICSTT-RM405	This software technical manual defines how to configure an AADvance controller using the AADvance Workbench software version 1.x to satisfy your system operation and application requirements.
AADvance-Trusted SIS Workstation Software User Guide, ICSTT-UM002	This publication provides how-to instructions for AADvance®-Trusted® SIS Workstation software configuration and use.
AADvance Controller OPC Portal Server User Manual, ICSTT-RM407	This manual describes how to install, configure and use the OPC Server for an AADvance Controller.
AADvance Controller PFH and PFDavg Data, ICSTT-RM449	This document contains the PFH and PFD_{avg} Data for the AADvance Controller. It includes examples on how to calculate the final figures for different controller configurations.
AADvance Controller Solutions Handbook, ICSTT-RM447	This technical manual describes the features, performance and functionality of the AADvance controller and systems. It gives guidance on how to design a system to satisfy your application requirements.
AADvance Controller Troubleshooting and Maintenance Manual, ICSTT-RM406	This technical manual describes how to maintain, troubleshoot and repair an AADvance Controller.
Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1	Provides general guidelines for installing a Rockwell Automation industrial system.
Product Certifications website, rok.auto/certifications.	Provides declarations of conformity, certificates, and other certification details.

You can view or download publications at rok.auto/literature.

Chapter 7

Configuring Timing and Communications

Chapter 8

Configuring the Controller I/O

Chapter 9

Sharing Data between ControlLogix and AADvance

Chapter 10

Configuring MODBUS

Chapter 11

SNCP

Chapter 12

Peer-to-Peer Network

Appendix A

History of Changes

Quick Start Guide

Creating Applications

The AADvance® Workbench is the software development environment for the AADvance controller. It lets you make one comprehensive control solution and then target parts of the solution to individual controllers. Interaction between resources is automatic, eliminating the need to have different synchronous schemes.

The AADvance Workbench supports the Function Block (FB), Structured Text (ST) and Ladder Diagrams (LD) languages of IEC 61131-3. The Instruction List (IL) language is not supported by Workbench 2.1 and the Sequential Function Chart (SFC) language is not supported by Workbench 2.1 in a safety related application. You can use one language or a combination of languages depending on what is applicable to the application and your programming style.

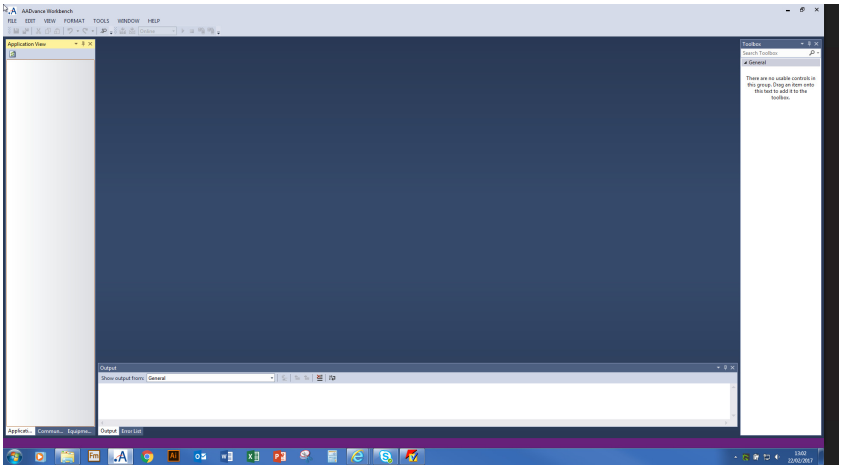
The AADvance Workbench runs on a Windows computer. You simulate and test programs on the computer before downloading them to the controller. The AADvance Workbench then works online to give you real-time monitoring, online updates and other features. Connectivity between the AADvance Workbench, the controller and between individual controllers, is through a standard Ethernet network.

Make a First Application

It only takes minutes to create the first application. If you have not already installed and licensed the AADvance Workbench, do this now. Installation instructions are provided in Chapter 2 Software Installation and Licensing.

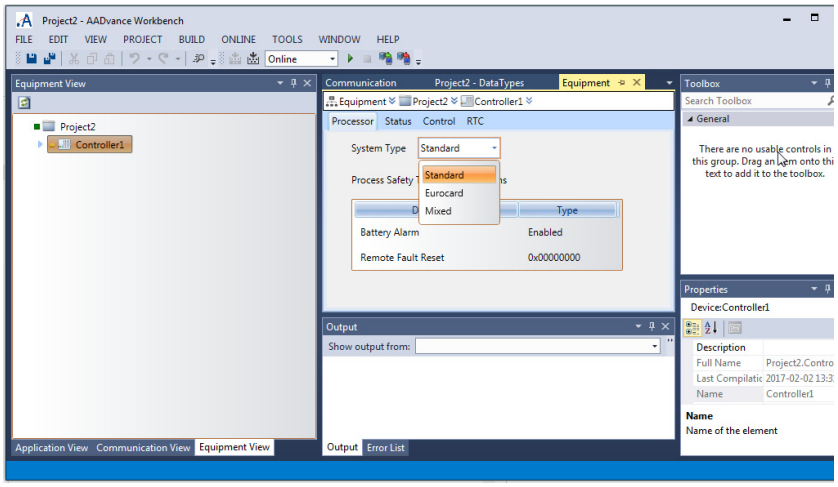
Begin by creating a project and defining the type of hardware for the controller:

1. Launch the software by clicking **Start** → All Programs → Rockwell Automation → AADvance 2.1 → AADvance 2.1.
   The application will open and display the following screen.
   Default Start-Up Screen



For subsequent use of the Workbench application, the layout for the screen presented on start-up is the same as the view displayed on the previous closure of the application. For further detail refer to Reset Window Layout on page 45.

2. Create a new project (File → New Project) and move to the Application View, Communication View or Equipment View.
   - the AADvance Workbench places a controller into the project
   - the controller is named Controller1.
3. Select Equipment View → Controller1 → identify the system type of the controller. The three options for system type are:
   - Standard
   - Eurocard
   - Mixed.



4. If the project needs more controllers, use the contextual menu on the project node (the first node in the tree) to add the controllers to the project. Then identify the class of each new controller.

After having identified the controller class you can configure the controller and place I/O modules with slots on the processor bus. Place input modules in simplex, duplex and triplex arrangements, output modules in simplex and duplex arrangements. The duplex and triplex arrangements use adjacent slots in groups of two or three.

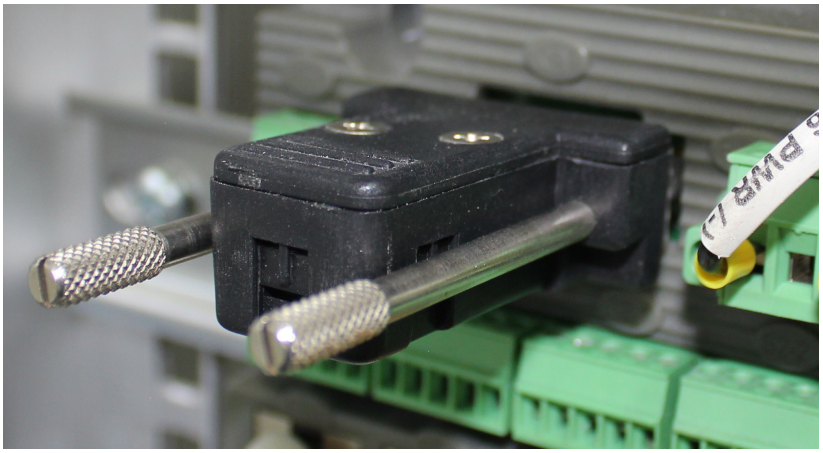
Proceed to the Application View, Communication View or Equipment View.

IMPORTANT The controller configuration must be exactly the same as the hardware arrangement. If you change the physical arrangement of the hardware, remember to change the Workbench configuration to match the new arrangement.

1. Use a spreadsheet or equivalent tool to assign tag names to the variables you want to use. If you chose structured variables for I/O channels, the AADvance Workbench generates a set of other variable elements with the same tag name for each element type.
2. Declare variables for I/O channels and module status parameters. An extensive range of variable types is available including a set of structured variables. You can subsequently add new variables if applicable.
3. Connect (wire) each I/O channel to variables to get input channel states and set output data values.
4. Set up the serial ports, process safety time, and SNTP and MODBUS services.
5. Allocate the IP addresses for communication with the AADvance controllers and configure the network communications parameters.

Program Enable Key

The standard AADvance controller has a program enable key which plugs into the KEY connector on the 9100 processor base unit. When the key is removed, the controller application is protected from unwanted changes. You must fit the program enable key before performing a download to the application, adjusting the application and setting or changing the controller IP address.



If the program enable key is not installed before performing a download to the application, adjusting the application and setting or changing the controller IP address will produce an error message. The error message will vary depending on the application status, running or stopped.

Figure 1 - With the application running the following error message occurs.

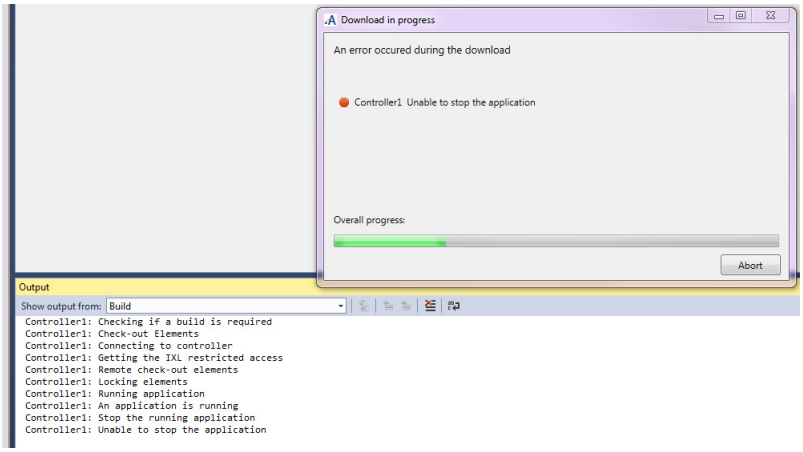
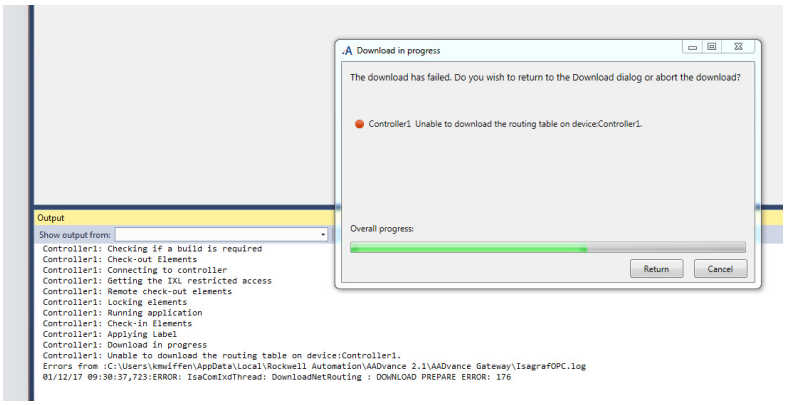


Figure 2 - With the application stopped the following error message occurs.



Hardware Redundancy

You create hardware redundancy in the AADvance Workbench when you identify the hardware configuration from the Equipment View. During the allocation of I/O modules to empty slots, you have the option of adding two or three modules. When you use the two or three option the AADvance Workbench automatically allocates the modules to a group of adjacent slots. The AADvance Workbench then only allows you to configure one set of I/O channels to the group.

You do not need to define redundancy for AADvance processors. The AADvance Workbench automatically connects to each processor when the network connection is set up.

Integrating the AADvance Controller with Other Systems

The AADvance controller connects to existing control systems and plant monitoring equipment. This connection enables a third-party control system to read the state of sensors connected to the controller.

The connection interfaces are through the controller network ports and serial ports, and use the following protocols:

- CIP™ over Ethernet/IP™
- Modbus RTU
- Open Modbus/TCP

- OLE for Process Control (OPC).

IMPORTANT	In order to support the integrity of the AADvance system, activation of network firewalls and Windows operating system firewalls is strongly advised.

Notes:

Software Installation and Licensing

The AADvance® Workbench software is a 32-bit application based on the Microsoft Visual Studio Shell that runs on all common Windows platforms.

This chapter provides the instructions to install and license the AADvance Workbench software.

Planning the AADvance Workbench Software Installation

IMPORTANT If you have not installed the AADvance Workbench software before, read all of this section before starting the installation.

The AADvance Workbench software uses the Microsoft Visual Studio 2010 with Service Pack 1. The installer automatically installs Microsoft Visual Studio 2010 Shell (isolated mode) and Microsoft .NET 4.0, but only if these are not already installed on the computer. This reduces the time necessary for updates of the AADvance Workbench, where the Microsoft Visual Studio and .NET are already on the computer.

The specification of the computer where you install the software must have the following specification, or better:

Operating system (32- bit or 64-bit):

- Microsoft Windows XP
- Windows Vista
- Windows 7
- Microsoft Windows Server 2003
- Microsoft Windows Server 2008.

Hardware:

- 1.6 GHz CPU
- 1 GB RAM (32-bit) or 2 GB RAM (64-bit) (add 512 MB if running in a virtual machine)
- DirectX 9 capable video card running at 1024 x 768 resolution display
- 5,400 RPM hard disk
- 3 GB available hard disk space
- DVD drive or network connection, to read software distribution files
- Network port (10/100 Base T Ethernet), for communications with the controller.

It is recommended that the computer has a 2.2 GHz or higher CPU; 1,024 MB or more RAM, a 1,280 x 1,024 display and a 7,200 RPM or higher hard disk.

It is also recommended that the hard disk has at least 10 GB free space. This provides sufficient space to hold the distribution zip file, the unzipped source files and the installed program files, and also sufficient space for Windows to operate reasonably quickly. You can retrieve a lot of this space by deleting the source files after finishing the installation.

Upgrading from Release 1.x

If you are upgrading to AADvance 2.x from AADvance 1.x, and you mean to migrate one or more existing projects to AADvance 2.x, do the following:

IMPORTANT You must do these steps before installing AADvance 2.x.

1. Make sure that the repository for the AADvance 1.x AADvance Workbench is on a local disk drive. The AADvance 2.x Workbench does not permit opening AADvance 1.x projects from a repository when no local copy exists.
2. If the project is stored as only an AADvance 1.x archive, put back the archive as a project in the AADvance 1.x Workbench.
3. Make sure that the AADvance 1.x project compiles fully.
4. Make sure that all users record all on-going modifications.
   - When you open a project from an AADvance 1.x repository in AADvance 2.x, the repository is upgraded to the current version and is no longer available in the previous version.
   - If you do not record all on-going modifications, the modifications become local and cannot be integrated into the repository.
5. Write down the ID of each MODBUS Slave set up for Ethernet communications. AADvance 2.x automatically assigns a value of 255 to a MODBUS Slave ID for which Ethernet was specified.
6. Write down which Ethernet ports are assigned to CIP binding groups. AADvance 2.x does not assign Ethernet ports to CIP binding groups, and you have to specify Ethernet ports for individual groups after the upgrade.

IMPORTANT The Instruction List (IL) language is not supported by Workbench 2.0 and the Sequential Function Chart (SFC) is not supported by Workbench 2.0 in a safety related application.

Install the AADvance Workbench Software - Web Distribution

If you need to update an existing installation, use the web distribution version of the AADvance Workbench software. The installation is smaller and quicker than the full product because the Microsoft .NET Framework and Visual Studio products which the software needs are already installed on the computer.

IMPORTANT If the AADvance Workbench has not been installed on the computer, you must install the full product first. Following the initial installation you can use the web distribution version to update the installation.

The installation of the web distribution version takes about 10 minutes on a computer with the recommended specification and will take longer if the machine has less than 1,024 MB RAM.

To install the AADvance Workbench software (web distribution) do the following:

1. Log onto the computer using an account with administrator permissions.
2. Close all programs and do not run them until the installation is completed.
3. The software is supplied as a compressed archive. Make a local copy of the archive on the hard disk. Then put the archive into a temporary folder on the hard disk. Find the file named setup.exe and run it.
4. If there is an older version of the software on the machine, the software will find it. Click **Yes** to update the existing version.
5. When the first window of the AADvance installer to opens, click **Next**.
6. Before proceeding, read the Release Notes.html file. This file is located in the same folder as setup.exe.
7. Return to the installer window. The installer window shows the license agreement. Accept the license agreement, click Next. The installation will start.
8. If the installer offers you the opportunity to make a backup, click **Yes**. Accept the default location for the backup files, and then wait for the installer to make the backup. The installer displays a message showing the location of the backup. Write down the location, then click **OK**.
9. If the computer has a repository of projects from AADvance release 1.x, use a repository path which is different from the one used for AADvance 1.x.
10. Wait for the installer to finish, then click **Finish**.
11. The installation of the AADvance Workbench is now complete.
12. Select No, just reset settings, overwrite my current settings, click **Next**.
13. Select CurrentSettings.vssettings, click **Finish**.

Install the AADvance Workbench Software - Full Product

If you need to install the software on a computer where the software has not been installed before, use the AADvance Workbench software full product. Make sure that you have at least 10 GB free space on the local hard disk before you install the software.

The full installation takes about 40 minutes on a computer having the recommended specifications and will take a longer time (typically 3 or 4 hours) if the machine has less than 1,024 MB RAM.

If the installation seems to have stopped (especially during the stages for Microsoft Visual Studio), let the installer run for at least ten minutes before investigating a possible problem.

IMPORTANT You must install the Microsoft .NET Framework to use the Workbench.

To install the AADvance Workbench software (full product) do the following:

1. Log onto the computer using an account with administrator permissions.
2. If you have Microsoft Visual Studio 2010 already installed on the computer (this is likely only if you are a software developer), make sure that it has Service Pack 1.
3. Close all programs and do not run them until the installation is completed.
4. The software is supplied as a compressed archive and on a DVD:

- If the software has been supplied as a compressed archive (such as a zip file), make a local copy of the archive on the hard disk. Then put the archive into a temporary folder on the hard disk. Locate the file named setup.exe and run it.
- If the software has been supplied on a DVD, place the disk into the DVD-ROM drive and wait for the disk to Autorun. If the installer does not start (typically because the Autorun feature has been disabled on the computer), find the file named setup.exe on the DVD and run it.

5. Wait for the first window of the AADvance installer to open.
6. Before proceeding, read the Release Notes.html. This file is located in the same folder as setup.exe.
7. Return to the installer window. The installer window shows the license agreement. Accept the license agreement, click **Next**.
8. Accept the Complete setup type (this is the default), click **Next**.
9. If you are upgrading from AADvance release 1.x, use a repository path which is different to the one used for AADvance 1.x.
10. If the Microsoft .NET Framework is not already installed on the computer, the installer will ask for your permission to install it. Click **Install**.
11. Wait for the Microsoft Visual Studio 2010 Isolated Shell English installation to run.
12. Wait for the Microsoft Visual Studio 2010 Service Pack 1 Setup to run.
13. Wait for Help Viewer Verification process to run.
14. Wait for the installer to configure the software and do a module file check.
15. Wait for the InstallShield Wizard Complete window to open, click **Finish**.
16. The installer places a shortcut on the Windows desktop.
17. The installation of the AADvance Workbench is now complete. If the software was supplied on a DVD, remove the DVD.

AADvance Workbench Licensing

The AADvance Workbench is licensed software. There are three types of license: full, single controller and demo.

- The single controller license is applicable for applications which use only one controller. The software features which add a second or subsequent controller to the project are disabled, and you cannot open an existing project which uses more than one controller.
- The full license supplies all of the features of the AADvance Workbench. It is applicable for applications with one or more controllers.
- The demo license is a like a full license, but with a time limit. You can use all of the features of the AADvance Workbench for up to 30 days after first running the AADvance Workbench is first run.

A demo license is supplied free of charge for a first installation on a computer. You change the demo license to a single controller license or a full license by purchasing an unlock code from Rockwell Automation, and entering the code into the software. When you use the demo license, the AADvance Workbench displays a Demo License window each time you try to open a project. The window includes the contact details at Rockwell Automation required for purchasing a license.

If you try to use the demo license for more than 30 days, the license expires. You cannot open a project or create a new one until you purchase a license.

Set Up a Single User License or a Full License

You have to set up a single user license or a full license if you want to use the AADvance Workbench after the demo license expires.

To set up a single user license or a full license, do the following:

1. Click **Help** → Licensing, a Licensing window is displayed.
2. Copy the three user codes into an email stating the type of license necessary (this is a single user license or a full license) and send the email to keymaster@ra.rockwell.com or keymaster@icstriplex.com.
3. Rockwell Automation will reply to the email and will send you two registration keys.
   - These registration keys are only valid for the provider user codes. These codes might have changed since the last time they were displayed, if they did you will need to resend the new user codes.
4. Enter the registration keys into the Licensing window, click **Validate**.
5. The software license can now be used.
6. Repeat this procedure on each other computer where you have installed and want to use the AADvance Workbench.

Managing a Paid-for License

After purchasing a paid-for license, the license persists through all Workbench updates and uninstall/re-install operations on the same computer.

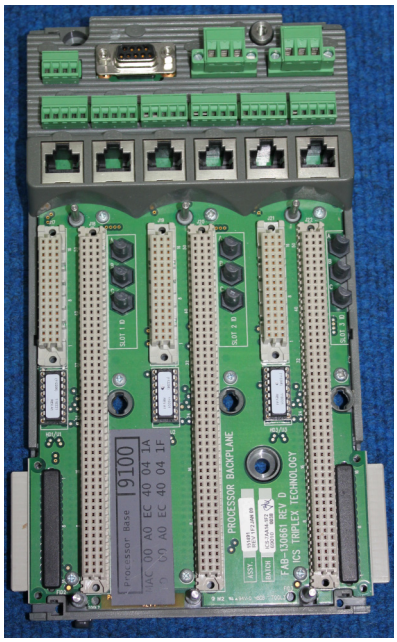
A paid-for license is specific to the computer for which it is purchased. If you need to move the license to a different computer, or you change the hard disk drive, contact Rockwell Automation for advice.

Notes:

Connecting the AADvance Workbench to the Controller

The AADvance® system uses Internet Protocol (IP) for communication/data transfer between the controller and the AADvance Workbench. This chapter describes how to connect the AADvance Workbench to the controller.

AADvance Processor Modules are mounted on a Processor Base Unit. Up to three Processor Modules can be mounted on the Processor Base Unit as shown below:



The Processor Base Unit connects to Input/Output (I/O) Base Units to form an interlocking backplane. The combined Processor and I/O Module assemblies comprise the Controller. The interlocked backplane is latched securely against a metal DIN rail. See also, Configuring the Controller I/O on page 109 .

Processor Modules can be mounted in any order onto any of the slots on the Processor Base Unit. However, for the controller to communicate with Workbench, the correct Internet Protocol (IP) address of the occupied position on the Processor Base Unit must be selected. For explanation, see below.

Controller IP Address

The AADvance controller stores its IP address data in non-volatile memory in the 9100 processor base unit. The data is independent of the 9110 processor modules in the controller, and so the controller keeps the address information when you remove a processor module.

You must set up the IP address data when you create a new system, or if you fit a new processor base unit.

After having set up the IP address data in the controller, you can configure the AADvance Workbench to find the controller on the network.

Allocating IP Addresses for Network Communications

For some systems, the administrator of the local area network will allocate the IP address for the controller. If this has not occurred, use an address from the ranges allocated to private networks:

- 10.0.0.0 to 10.255.255.255 (10/8 prefix)
- 172.16.0.0 to 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 to 192.168.255.255 (192.168/16 prefix).

Each controller on a local area network must have a unique IP address.

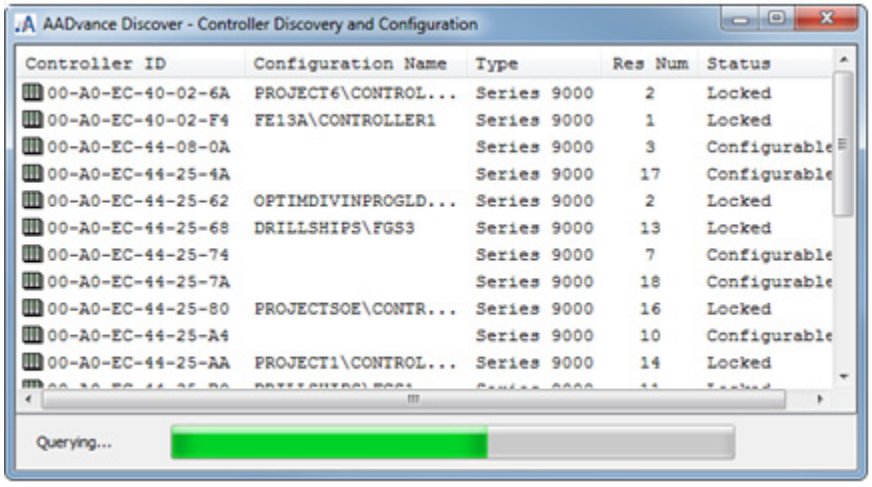
IMPORTANT Make sure that the two Ethernet ports on each 9110 processor module are on different subnets.

AADvance Discover Utility

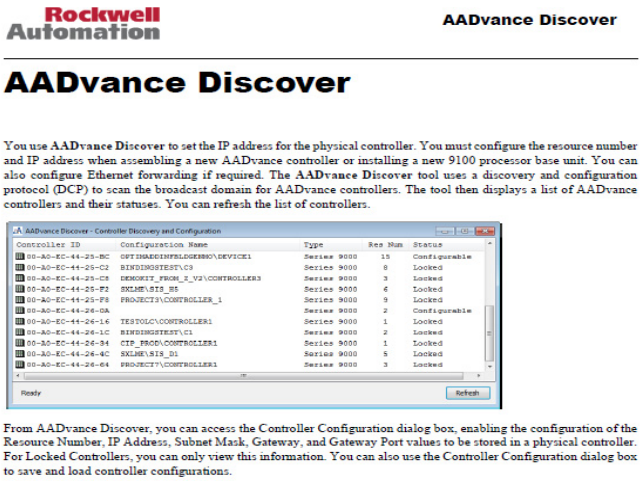
You use the AADvance Discover utility to set up the controller IP address. The utility uses a discovery and configuration protocol (DCP) to perform a scan of the broadcast domain for AADvance controllers and enables you to configure the resource number and IP Address to be stored in a controller. You can use the utility to save configurations and load them again in the future.

The AADvance Discover utility is part of the AADvance Workbench installation. It appears on the drop-down list in the View menu, which is a tab on the main menu bar.

There is also a feature on the Windows Start menu:
Click “Windows Start” → “All Programs”. → “Rockwell Automation” → “AADvance 2.1” → “AADvDiscover” to start the utility.
The utility displays a list of the AADvance controllers on the broadcast network, and reports a status for each one.



If the The AADvance discover utility is selected and the [F1] key is pressed on the keyboard a .pdf help-file is opened.



Rockwell Automation

AADvance Discover

AADvance Discover

You use **AADvance Discover** to set the IP address for the physical controller. You must configure the resource number and IP address when assembling a new AADvance controller or installing a new 9100 processor base unit. You can also configure Ethernet forwarding if required. The **AADvance Discover** tool uses a discovery and configuration protocol (DCP) to scan the broadcast domain for AADvance controllers. The tool then displays a list of AADvance controllers and their statuses. You can refresh the list of controllers.

From AADvance Discover, you can access the Controller Configuration dialog box, enabling the configuration of the Resource Number, IP Address, Subnet Mask, Gateway, and Gateway Port values to be stored in a physical controller. For Locked Controllers, you can only view this information. You can also use the Controller Configuration dialog box to save and load controller configurations.

Double-clicking on an entry in the list enables examining the resource and IP address settings for a controller. A Refresh button enables a scan of the network and creates a new list.

If a controller known to be on the network does not show in the list, look for communication blocking or miss-routing by Windows or by other network devices. DCP communications will not go through network bridges and routers.

The DCP is proprietary to Rockwell Automation. It uses the first MAC address of the 9100 processor base unit to identify each individual controller - the MAC address is the 'Controller ID' in the list.

The controller status will be 'No Response', 'Locked' or 'Configurable':

- 'No response' means that the controller is turned off, or the communications between the computer running the utility and the controller have failed.
- 'Locked' means that the utility has established communications with the controller, but one or more of the criteria for 'Configurable' status is not present.
- 'Configurable' means that the controller can be configured with its IP address. The utility has established communications with the controller, the program enable key is present (this plugs into the KEY connector on the 9100 processor base unit) and either no application is loaded or an application is loaded but not running.

The status must be shown as 'Configurable' before you can set the controller configuration. If you change the configuration of a controller, click the **Refresh** button to make a new list.

A status bar is displayed at the bottom of the window, below the list. This will show the message 'Initializing' as the tool starts followed by 'Searching', locating all the controllers connected to the network, then 'Querying' and finally 'Ready':

- 'Querying' scans the network for controllers and creates a list
- 'Ready' means the controller status and IP addresses are displayed ready for new settings.

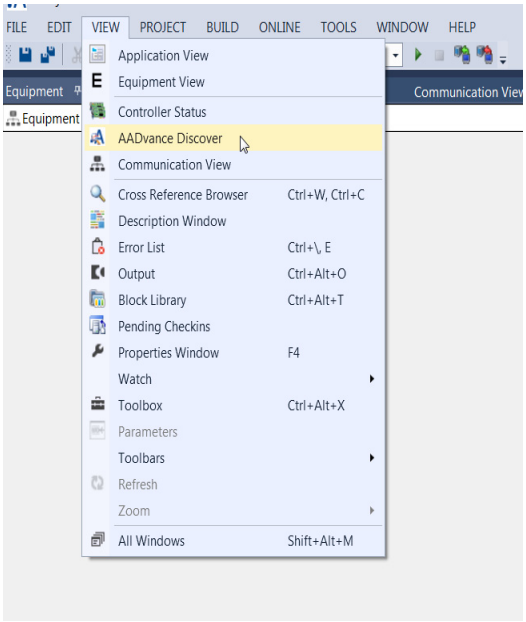
A Refresh button allows you to repeat the Querying process.

Configure the IP Address in the Controller

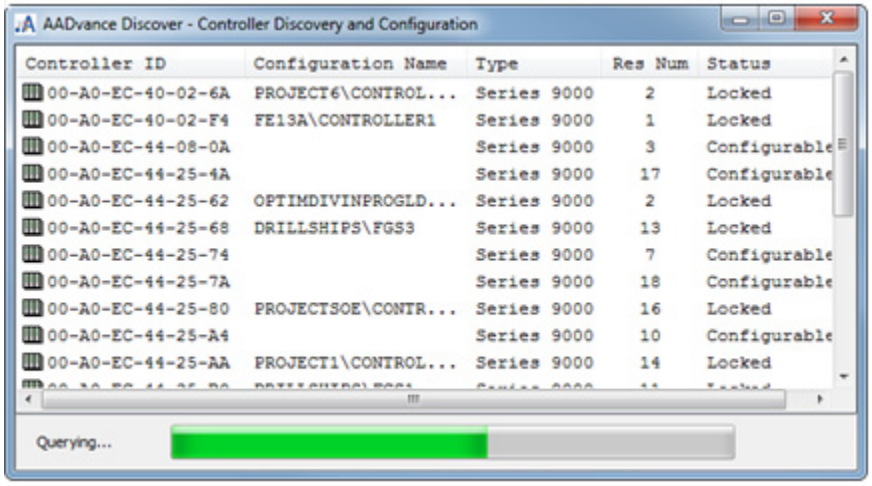
When you assemble a new AADvance controller, or install a new 9100 processor base unit, you have to configure the IP Address stored in the controller.

The procedure to configure the IP Address uses the AADvance Discover utility. Changes occur immediately and you do not have to start the controller again. To set the IP Address do the following:

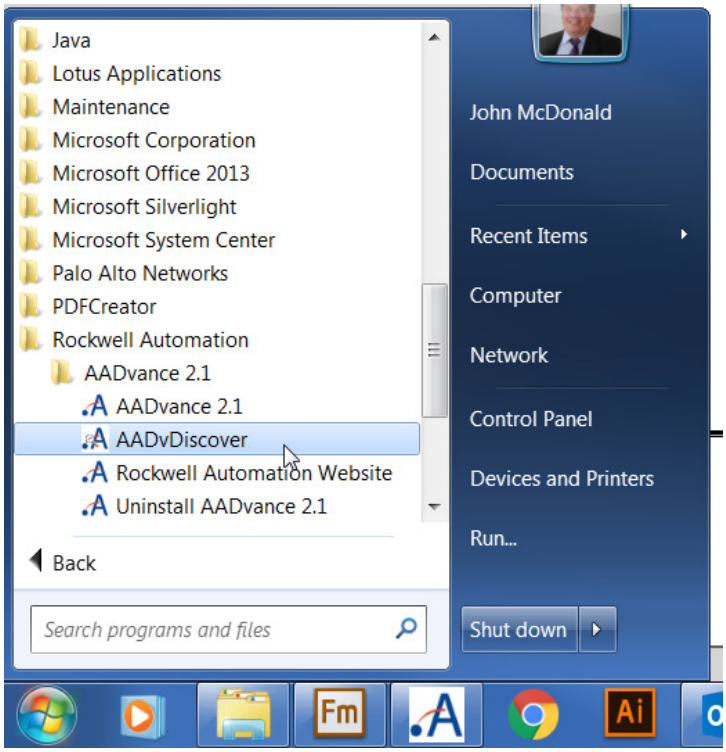
1. Write down the controller's first MAC address (the Controller ID) displayed on a label on the processor base unit. Install at least one 9110 processor module into the processor base unit.
2. Make sure the program enable key is inserted in the KEY connector on the processor base unit.
3. The AADvance Discover utility may be opened, either from the view tab on the menu bar, or through the Windows Start menu.
   - To start the AADvance Discover tool from the view tab on the menu bar within Workbench 2.1:
     Click **View** → AADvance Discover as shown below.



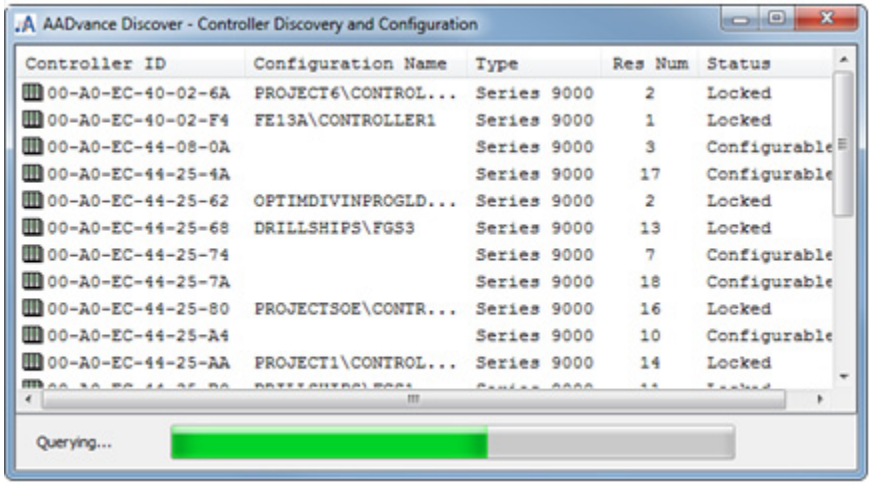
The AADvance Discover utility scans the network for controllers, and creates a list.



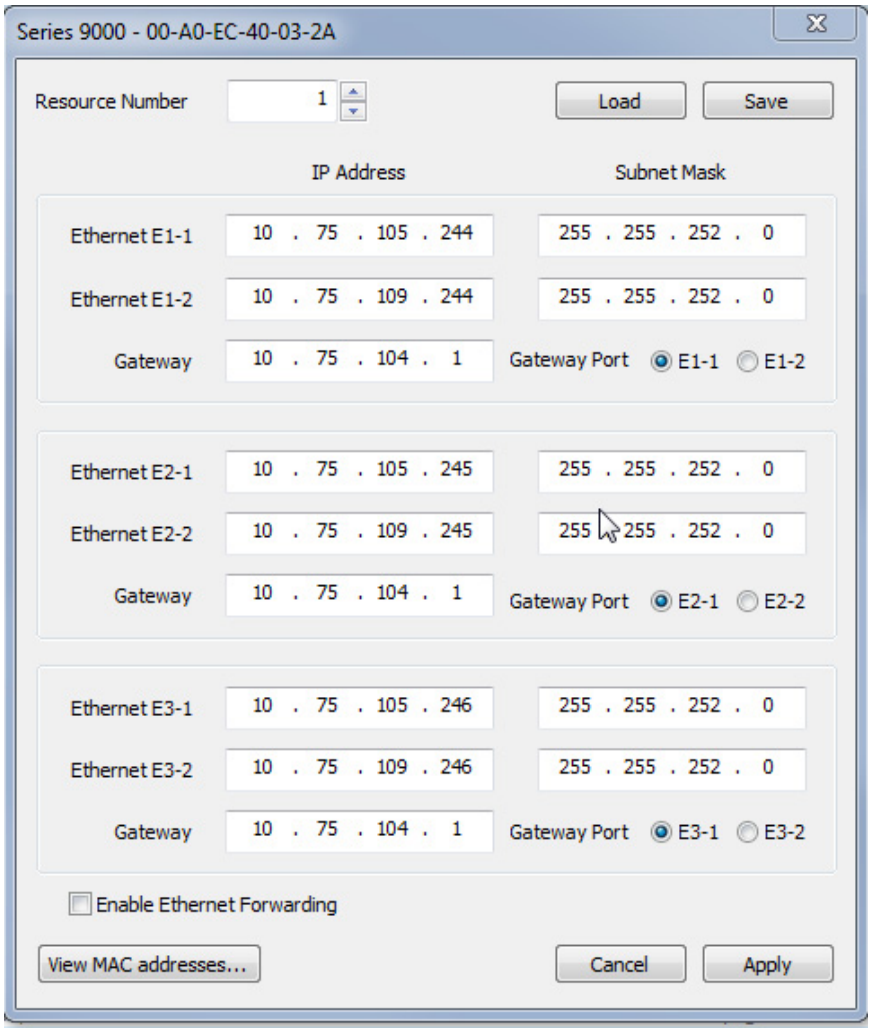
- Alternatively start the AADvance Discover tool from the Windows Start menu:
  Start → All Programs → Rockwell Automation → AADvance 2.1 → AADvDiscover as shown below.



The AADvance Discover tool above scans the network for controllers, and creates a list.



4. Locate the controller in the list and make sure that the status of the controller is Configurable.
5. Double-click on the MAC address in the Controller ID field.
   - The Resource Number, IP Address and Subnet Mask dialog boxes are displayed.



6. Enter the IP Address and Subnet Mask into the fields for each Ethernet port.
7. Enter the Gateway values for each processor module, then click **Apply**.

- Returning to the main window of the utility, the controller status will show In Progress and then Configurable.
- The controller uses the new settings.

Configure the Controller Resource Number in the Controller

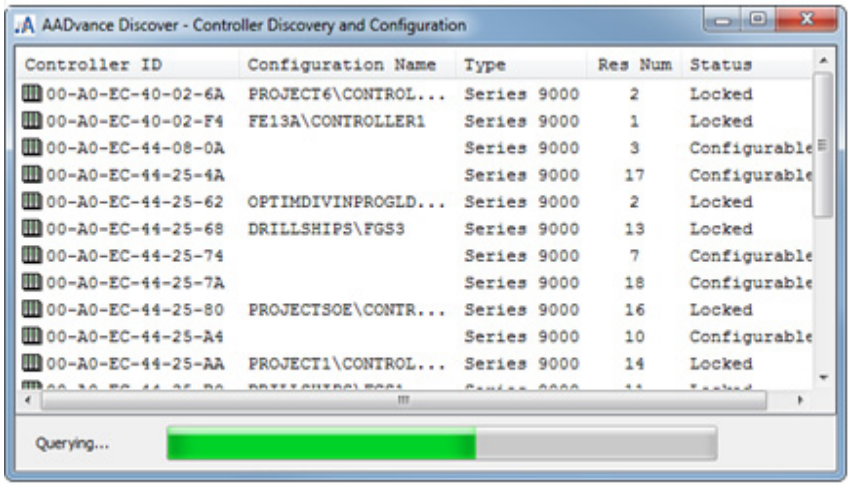
When assembling a new AADvance controller or installing a new 9100 processor base unit, the resource number stored in the controller must be configured. The resource number is a type of device address, and it must also be configured in the application.

The procedure to configure the resource number uses the AADvance Discover utility. To set the resource number do the following:

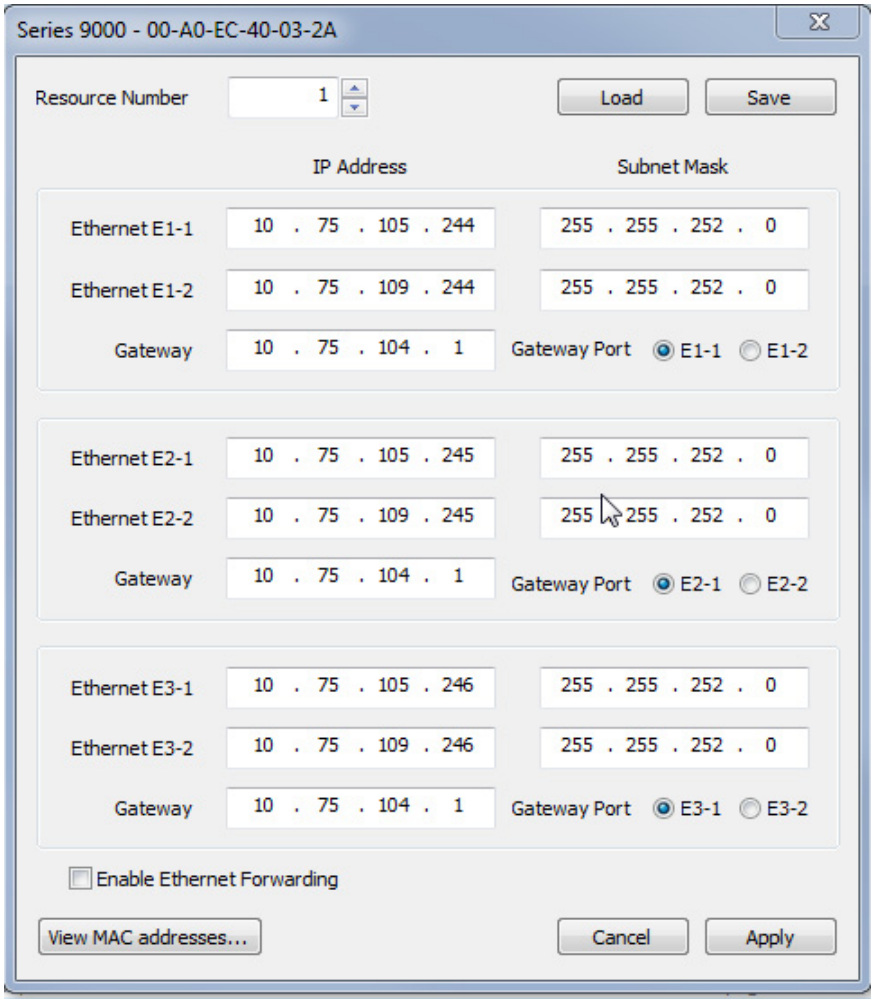
1. Write down the controller's first MAC address (the Controller ID) displayed on a label on the processor base unit. Install at least one 9110 processor module into the processor base unit.
2. Make sure the program enable key is inserted in the KEY connector on the processor base unit.
3. Start the AADvance Discover tool:
   a. Either from the view tab on the main menu bar within Workbench 2.1:
   - Click **View** → AADvance Discover

   b. Or from the Start menu:
   - Start → All Programs→ Rockwell Automation → AADvance2.1 → AADvDiscover.

   The AADvance Discover utility scans the network for controllers and creates a list.

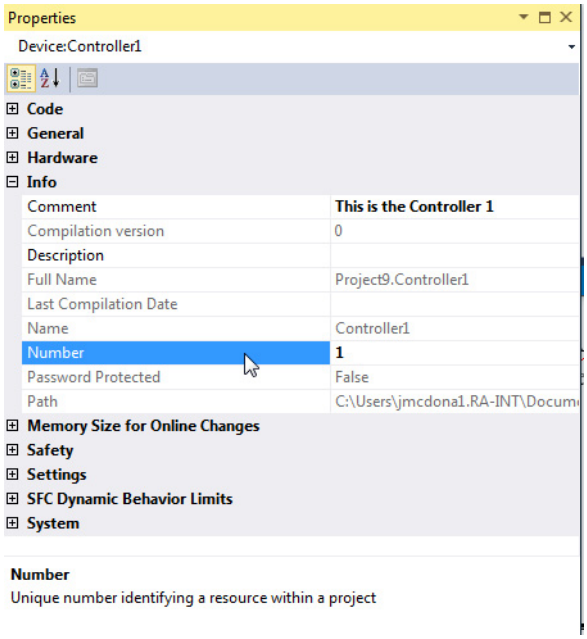


4. Locate the controller in the list and make sure that the status of the controller is **Configurable**.
5. Double-click the MAC address in the Controller ID field.
   - The Resource Number, IP Address and Subnet Mask dialog boxes are displayed.



6. Enter the resource value into the Resource Number field, then click **Apply**.
   - Returning to the main window of the utility, the controller status will show Pending Restart.
7. To finish the update, turn off the power to the controller.
8. Start the controller. Refresh the screen to make sure that the new resource number is shown in the resource field and that the controller status is configurable.

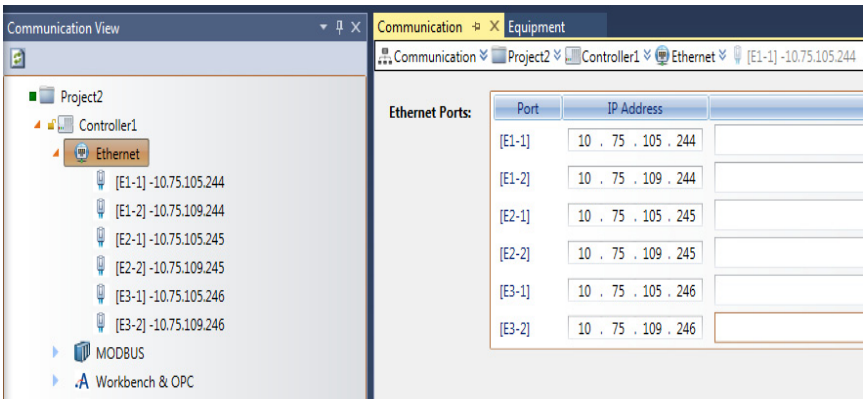
IMPORTANT The unique number identifying a resource within a project must also be configured in the Properties/Properties Window for the specified controller. See Configure Controller Properties on page 58



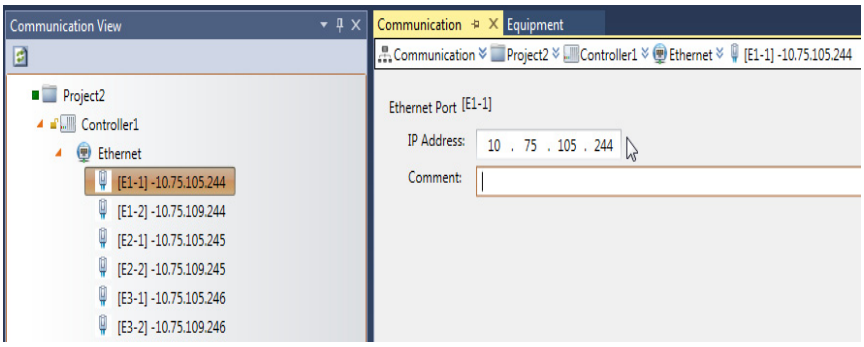
Configure the IP Address of the Target Controller

To connect the AADvance Workbench project to the target controller you have to indicate the IP addresses allocated for the controller to the project.

1. Select the Communication View.



2. Double-click the Ethernet node. The Ethernet Ports table screen is displayed.
3. Enter the IP Address for each port
   - Where a port has been left "Not Configured" the port can be configured individually.
4. Select the "Not Configured" port.
5. Enter the IP Address for the port.



System Security

An AADvance system, with its workstations and DCS interfaces, whether using Ethernet networks or Serial links is likely part of a larger corporate network which may expose the system to accidental or malicious infection, attack or less obvious security vulnerabilities. If appropriate (or defined in the SRS), a security risk assessment should be carried out and the appropriate level of risk mitigation applied.

The following general security steps should be used to verify that the system is secure:

WARNING: Network and workstation security must be set up when installing and setting up the system. As a minimum use the following security measures:

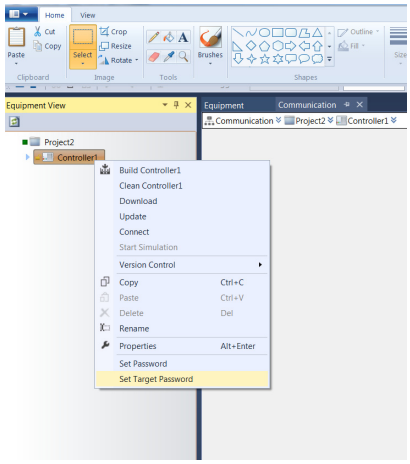
- The AADvance system must not be connected to a network with open unsecured access to the Internet.
- A router firewall must be active on the Workstation, helping prevent access to the unused Ethernet ports on each communication interface.
- Anti-virus software must be installed and be kept updated.

IMPORTANT Firewalls have been known to affect the operation of the AADvance Discover utility so it may be necessary to temporarily disable the Firewall when using this tool.

- If the workstation is a laptop, it must be kept locked when not in use.
- The Workbench software must be password protected. This can be done when the Workbench is installed.
- The application must be password protected if a program enable key is not used on the system.

Workbench Access

Passwords are used to protect Project access and target AADvance Controller access.



Setting Project Access Control

For project security, access control may be set using a password for projects, controllers, programs, libraries, and library functions and function blocks. Password definitions are limited to eight characters and can consist of letters, digits, and symbols. When projects are password-protected they cannot be opened for editing. Project sub-elements, can have their own level of access control. For example, a program having its own password remains locked and cannot be modified without entering its password.

NOTE Because programs are encrypted, password definitions must be retained.

In Application View, the following indicates the security state for elements:

The closed padlock indicates that a lock is applied to the element.

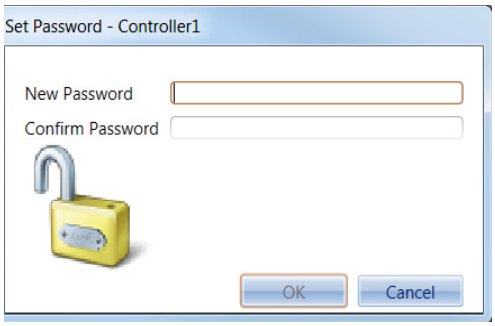
When opening a project having password-protected elements, the prompt to enter the password is only offered once for each element. Password-protected elements have the following modification restrictions:

Password-Protected Element	Modification Restrictions
Project	Opening the project
Controller	Adding, editing and deleting a program, modifying the communication protocols, modifying the system type and controller properties, wiring variables and adding, editing and deleting input / output modules
Program	Viewing the program
Library	Adding, editing and deleting a library function or function block
Library Function	Viewing the function
Library Function Block	Viewing the function block

Existing passwords may be edited for projects and project sub-elements. Existing passwords may also be removed. When copying, pasting, importing and exporting elements which have access control, password definitions are retained.

To set a password

1. In the Application View, right-click the controller instance, and then click **Set Password**.
2. In the Set Password dialog box, enter the required information, then click **OK**.



   - In the New Password field, type the required password.
   - In the Confirm Password field, re-type the required password.

To edit a password

1. In the Application View, right-click the required element, and then click **Set Password**.
2. In the Change Password dialog box, enter the required information, then click **OK**.

   - In the Old Password field, type the current password.
   - In the New Password field, type the required password.
   - In the Confirm Password field, re-type the required password.

To remove a password

1. In the Application View, right-click the required element, and then click **Set Password**.
2. In the Change Password dialog box, enter the required information, then click **OK**.
   - In the Old Password field, type the current password.
   - The New Password and Confirm Password fields must remain blank.

The element is no longer password-protected.

Setting Target Access Control

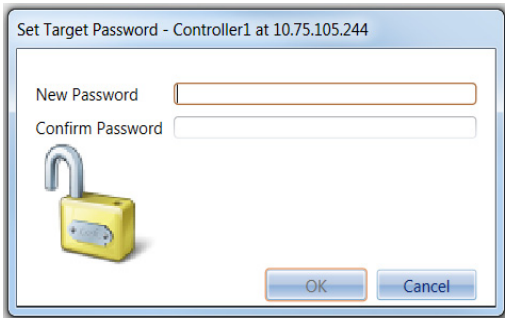
For controller security, access control may be set by defining a password for the target AADvance controller. Password definitions are limited to eight characters and can consist of letters, digits, and symbols. The target password protects the following operations:

- stopping the application
- downloading the application
- updating the application
- locking variables
- modifying the value of a variable.

Existing passwords may be edited for targets. Existing passwords can also be removed for target controllers. When setting, editing, and deleting the password for a controller target, the attached target must be running.

To set a password for a target

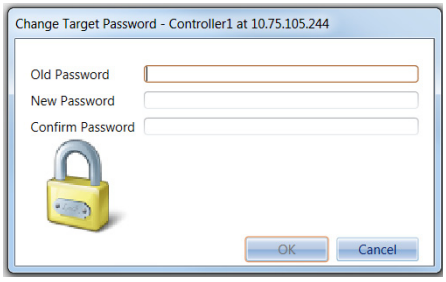
1. In the Application View, right-click the controller instance, and then click **Set Target Password**.
2. In the Set Password dialog box, enter the required information, then click **OK**.



   - In the New Password field, type the required password.
   - In the Confirm Password field, re-type the required password.

To edit a password for a target

1. In the Application View, right-click the required controller, and then click **Set Target Password**.
2. In the Set Password dialog box, enter the required information, then click **OK**.



   - In the Old Password field, type the current password.
   - In the New Password field, type the required password.
   - In the Confirm Password field, re-type the required password.

To remove a password for a target

1. In the Application View, right-click the required controller, and then click **Set Target Password**.
2. In the Set Password dialog box, enter the required information, then click **OK**.
   - In the Old Password field, type the current password.
   - The New Password and Confirm Password fields must remain blank.

The target is no longer password-protected.

Threat Analysis

A system threat analysis must be performed and all high risk threats mitigated before the system is commissioned.

Integrating the AADvance Controller with Other Systems

The AADvance controller connects to existing control systems and plant monitoring equipment. This connection enables a third-party control system read the state of sensors connected to the controller.

The connection interfaces are through the controller network ports and serial ports, and use the following protocols:

- CIP over EtherNet/IP
- Modbus RTU
- Open Modbus/TCP
- OLE for Process Control (OPC).

IMPORTANT It is strongly advised that network firewalls and Windows operating system firewalls are activated in order to support the integrity of the AADvance system.

Network Firewall

If the network used by the AADvance system is connected to another network, the connections should pass through a firewall.

The following transport layer ports (services) are supported by AADvance; some ports are always open, others are only open when configured.

AADvance Communication Ports

Protocol	Port Number	Port Open	Purpose	Port Open When
TCP	502	When configured	Modbus TCP slave	Open if Controller is configured as a Modbus TCP slave.
TCP	1132	Always	ISaGRAF®, application downloads, debug, SoE etc.	N/A
TCP	2000	When configured	Modbus RTU slave	Open if controller is configured as a Modbus RTU slave. The default port, 2000, is given in this table.

Protocol	Port Number	Port Open	Purpose	Port Open When
TCP	10001- 10006	When configured	Transparent Comms Interface (Serial Tunneling)	Serial tunneling must be enabled AND the resource is not currently loaded. The ports will be closed when the resource is restarted.
TCP	44818	Always	CIP Produce & Consume	N/A
TCP	55555	Always	Telnet (diagnostic interface) (1)	N/A
UDP	123	When configured	(S)NTP	The controller is configured as either a SNTP client or server. The ports are otherwise closed.
UDP	1123,1124	When configured	IXL bindings	Application uses IXL bindings • 1123 open on a producer • 1124 open on a consumer
UDP	2010	Always	Discovery and configuration protocol (DCP, Rockwell Automation)	N/A
UDP	2222	When configured	CIP Produce & Consume I/O	Open if CIP Produce & Consume I/O traffic is active
UDP	5000	When configured	Trusted© peer-to-peer (P2P)	At least one P2P network / subnet has been configured and is enabled.
UDP	44818	Always	CIP Producer & Consume	N/A

(1) The Telnet service provides various commands to gather diagnostic information about the controller and interact with it. It is intended to be used by Rockwell Automation Support Engineers, or users acting under the guidance of Rockwell Automation Support Engineers. It is important that this service should only be available to authorised users whose access is limited to the control network. The service must not be made available over the corporate network or the internet. When not required, the service should be blocked, even within the control network. Additionally, the programme enable key should be removed from the controller in normal use.

Those ports that are always open, even when not configured or unused, are open to unauthorized access. Use the following guidelines to protect all open transport layer ports.

1. If the network used by the AADvance system is connected to another network, the connection should pass through a firewall, to protect the AADvance system from potential threats from the other networks. Techniques to protect a control network from the rest of the plant's network are described in the RA/Cisco CPwE Design and Implementation Guide (ENET-TD001-EN-P). See below for some advice from the CPwE.
2. The firewall should be configured to block all communication ports. If necessary a specific port may be enabled to a device that needs to communicate with other devices on other networks.
3. Ingress rate limiting should be used to protect the AADvance from network storms. The limit chosen should not impede the expected peak ingress rate for that controller and should be determined by calculation or observation of the system's network traffic when running.
4. The SNCP port must only be allowed to pass through the firewall if the Windows PCs running the AADvance Workbench are on a separate network.
5. The variable bindings ports must only be allowed to pass through the firewall if the AADvance controller is communicating with another AADvance controller on a separate network.
6. The other communication ports (e.g. MODBUS, SNTP) must only be allowed to pass through the firewall if the AADvance controller or Windows PC communicates with other devices on other networks.

From the CPwE advice, techniques for securing a control network include (but are not limited to):

- Physical access to the Cell/Area zone should be restricted to authorized users. Especially, physical access within the Cell/Area zone to AADvance systems should be tightly controlled.
- Switches within the Cell/Area zone should be configured to permit only authorized devices (eg by MAC address).
- A firewall should be used to help prevent access to ports at the boundary to the Cell/Area zone.
- When it is known that services are not required to a device within the Cell/Area zone, then for enhanced security, firewall devices can be employed in "transparent mode" to protect the AADvance controllers, blocking access to the ports requiring protection.

Windows PC Firewall

The firewall on the Windows PC should be active. It should be configured to allow connections through the ports used to communicate with the AADvance Controller. Other ports may be configured for other applications (e.g. HMI, time synchronization, access to file shares on servers, Windows Update, virus detection software).

It may be necessary to allow the Windows PCs to access Windows Updates, time synchronization and virus updates.

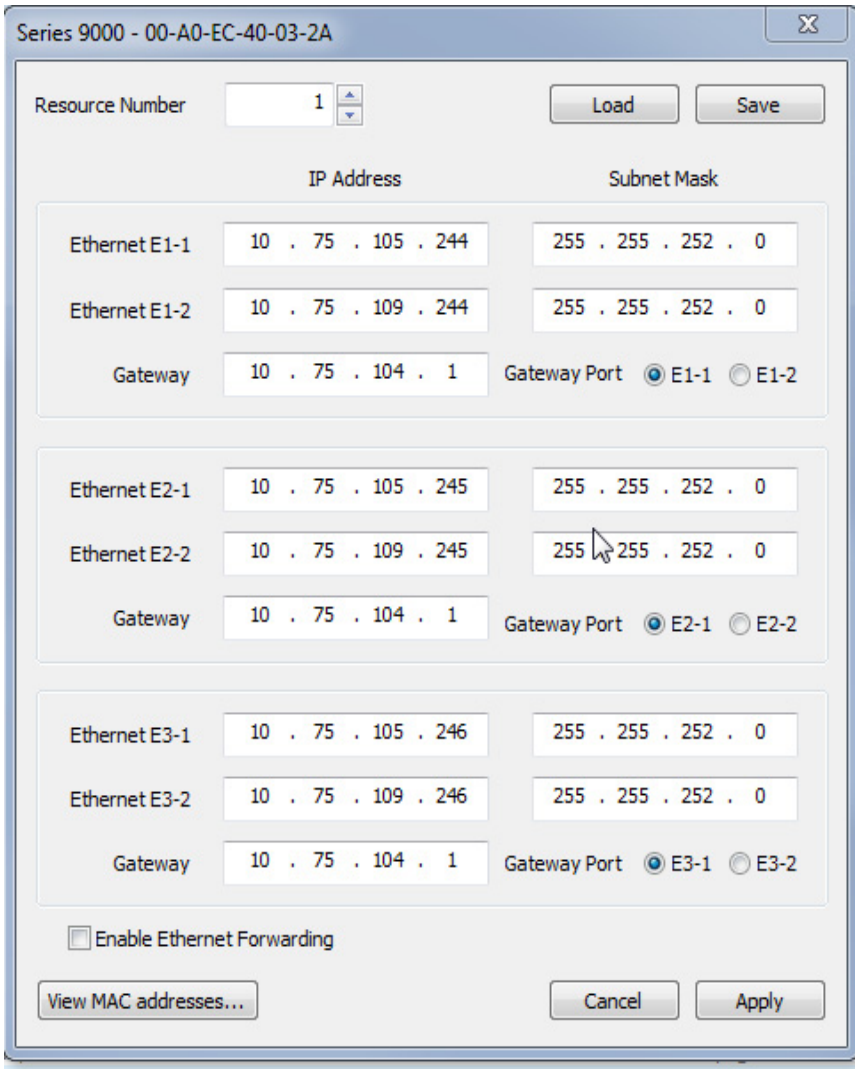
Windows PCs may also run other software (e.g. HMI, data logging), for which it may be necessary to allow other ports to pass through the firewall. Refer to the manufacturer's manual for details.

NOTE It is beyond the scope of this document to detail how to configure a firewall. Refer to the manufacturer's manual for instructions about firewall settings. It is recommended that the firewall settings are backed up in case of a firewall hardware failure.

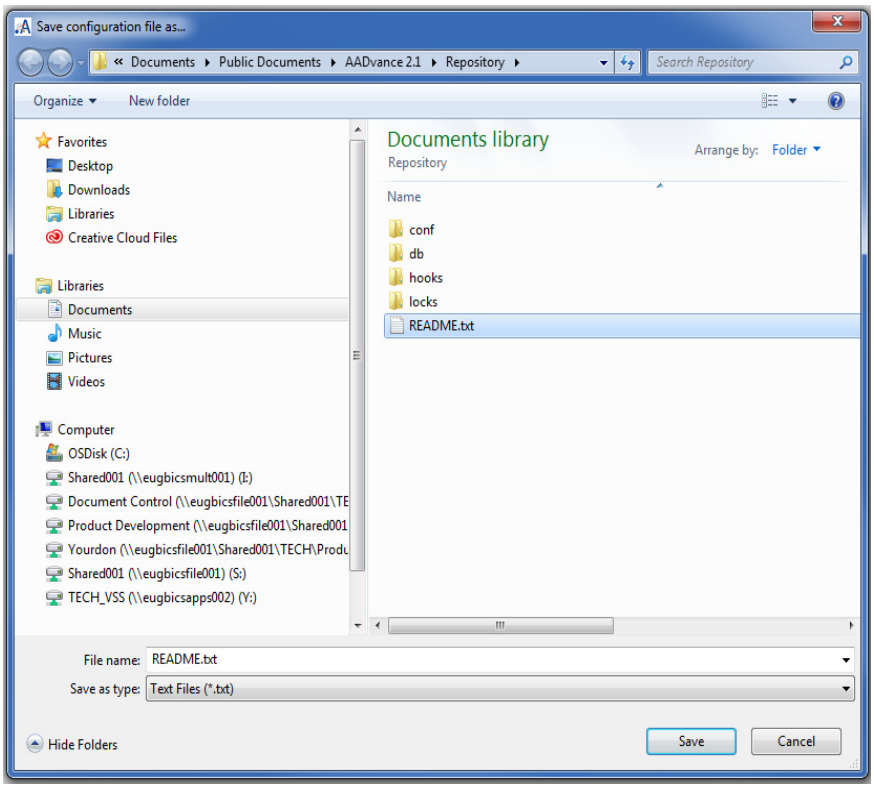
Save and Load a Configuration

Save a Configuration File

After having entered the IP Address details you can now save the configuration:



1. Click **Save** after having entered your required configuration.
2. Assign a name to the configuration and save it to an applicable location.



Reload a saved Configuration File.

To reload a saved configuration file:

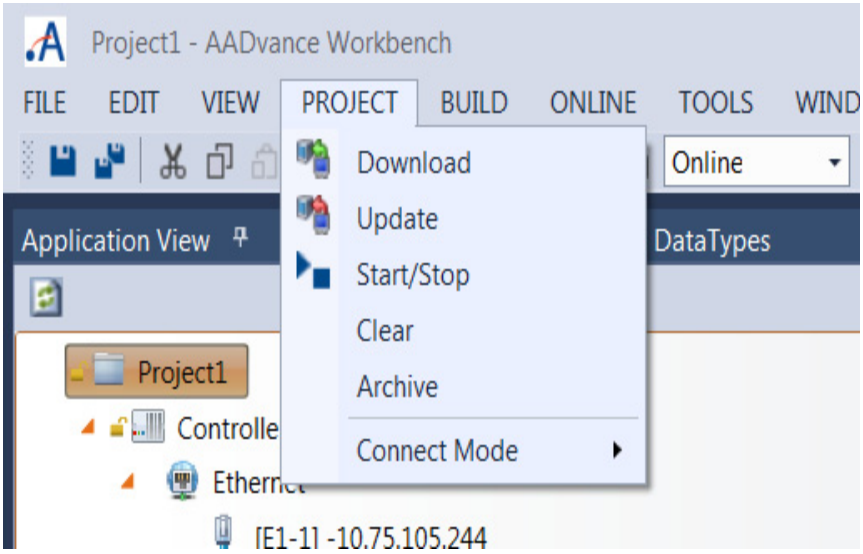
1. Open the AADvance Discover utility.
2. Double click a **MAC address** to open the Configuration dialog box.
3. Click Load to load the saved configuration.
4. Select the Configuration file where it was saved.
5. Confirm that the configuration loaded is the same as the saved configuration.

Replace an Application

To replace an application you have a choice of two methods. Method one uses the Clear and Download commands of the Project menu. Method two is a manual procedure that requires you to remove the processor modules.

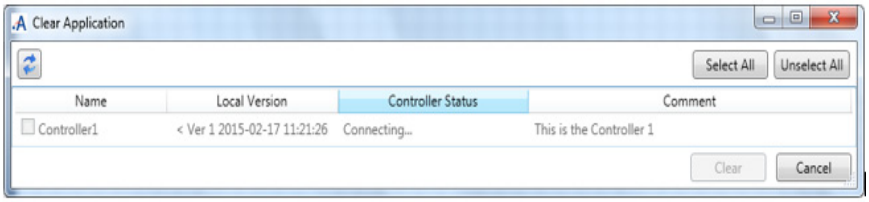
Project Clear Command Procedure

1. Build a new project using the same IP address as the current project.
2. Connect to the system that has the project you want to replace.
3. Select **Clear** command from the Project drop-down menu.



- A Clear Application window appears.

4. Choose the controller (s).



- The application will be cleared from that controller.

5. Use the Download option to download the new project to the controller.

Manual Procedure

WARNING: Ensure you take alternative measures to protect your business application before stopping an application in a Safety System.

1. Power down the system.
2. Remove all the processor modules.
3. Insert a processor into the previously empty slot on the T9100 base unit.
4. Power up the system and let the processor start for one minute until the Ready LED is green but the Run LED is red.
   - The processor will detect that its application does not match the signature on the base unit and will delete all its application files.
5. Power down the system.
6. Remove the cleared processor module.
7. Insert the cleared processor module in the slot where you connect the Ethernet port, usually the left slot (Slot A).
8. Power up the system and let the processor start for one minute until the Ready LED is green but the Run LED is red
9. Use AADvance Discover to configure the IP address (See Configure the IP Address in the Controller on page 28) and save the configuration (Refer to Save and Load a Configuration on page 40).
10. Press the **Apply** Tab, located in the bottom right corner of the Configuration Dialog Box, to apply the configuration file.
11. Download the application from Workbench.

12. Wait until the application has downloaded and the Ready and Run LEDs go green.
13. Insert a 2nd processor of a dual configuration and let it educate from the other processor.
14. Once the 2nd processor's Run LED is amber, press its Reset button to set it active.
15. Repeat if a third processor is required.

Starting a Project and Defining the Controller

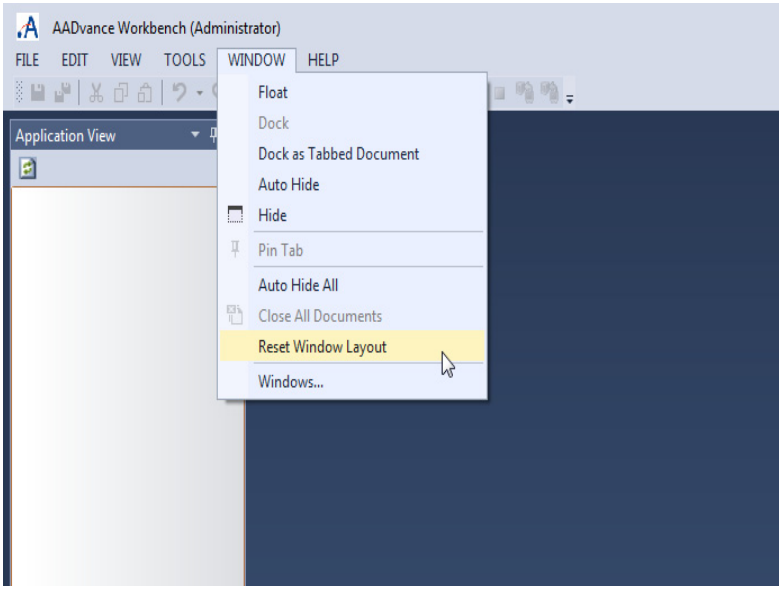
The configuration process starts by making a new AADvance® project or importing an existing project. After you have a project, you can use the Equipment View and the Communication View to move around the project, configure the controller type and, if applicable, add more controllers. You can then configure the timing, communications and hardware for the controllers.

IMPORTANT	When first adding a project the Controller nodes are called Controller 1, 2, etc. For the purpose of the screen captures for this manual, the names of the Controller nodes have been changed to Config1, 2, etc. You can change the controller node names as required.

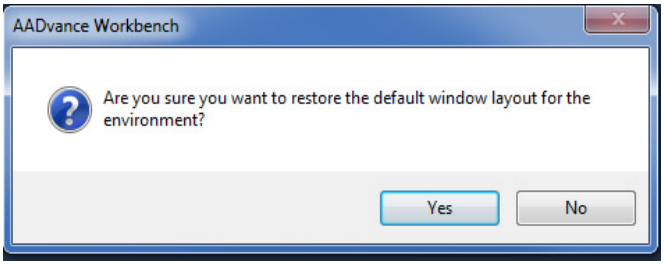
Reset Window Layout

The Workbench desktop screen is highly customizable and can be changed to suit user preference. The changes are automatically saved when the Workbench is shut-down and the new display settings are used the next time the Workbench is opened. However, if at any time the user wishes to return to the default screen environment, this can readily be achieved by using either the Window tab or Tools tab from the main menu.

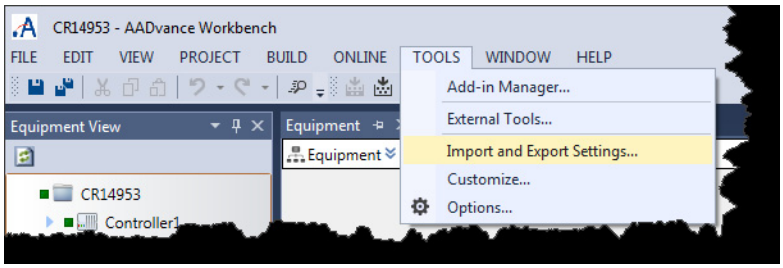
If the Window tab is selected, a drop-down box opens to reveal Reset Window Layout as one of the options.



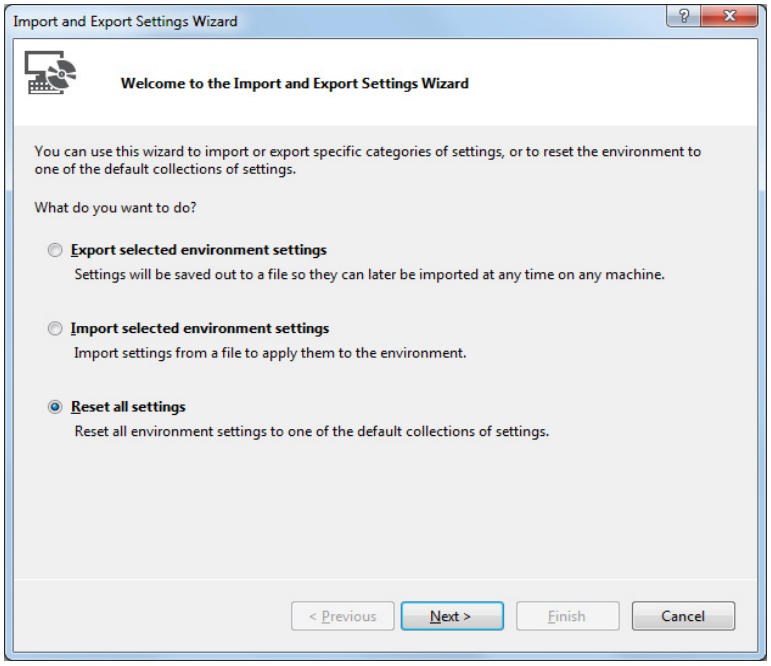
Selecting the Reset Window Layout option opens a dialog box to confirm the user choice.



Selecting the **Yes** button restores the original default setting as illustrated in the Quick Start Guide on page 13 .

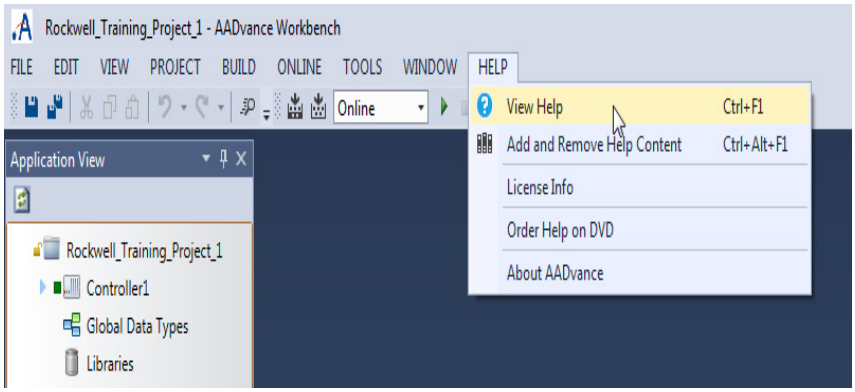


Selection of Import and Export Settings, opens the Import and Export Settings Wizard, which also has an option to restore the screen environment to one of the default collections of settings.

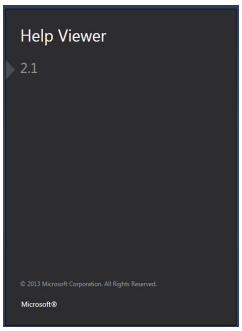


Help

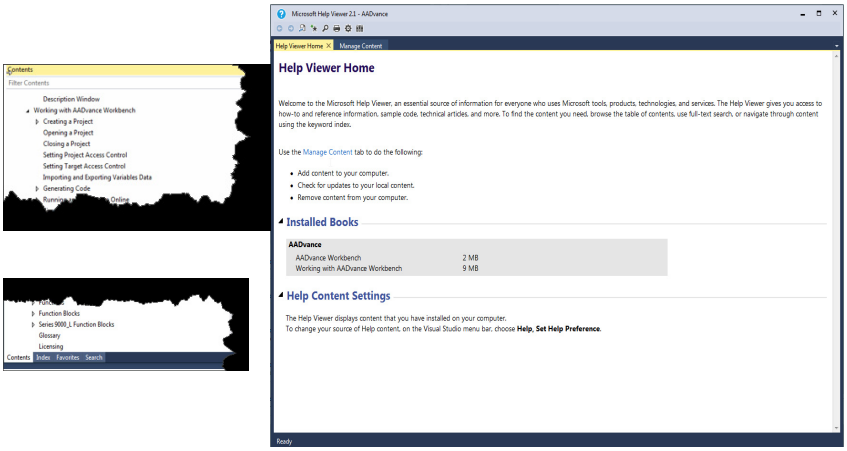
The AADvance Workbench includes a comprehensive Help section, which is accessed by selecting the **Help** tab on the main menu.



Selecting the **View Help** command opens the help viewer,

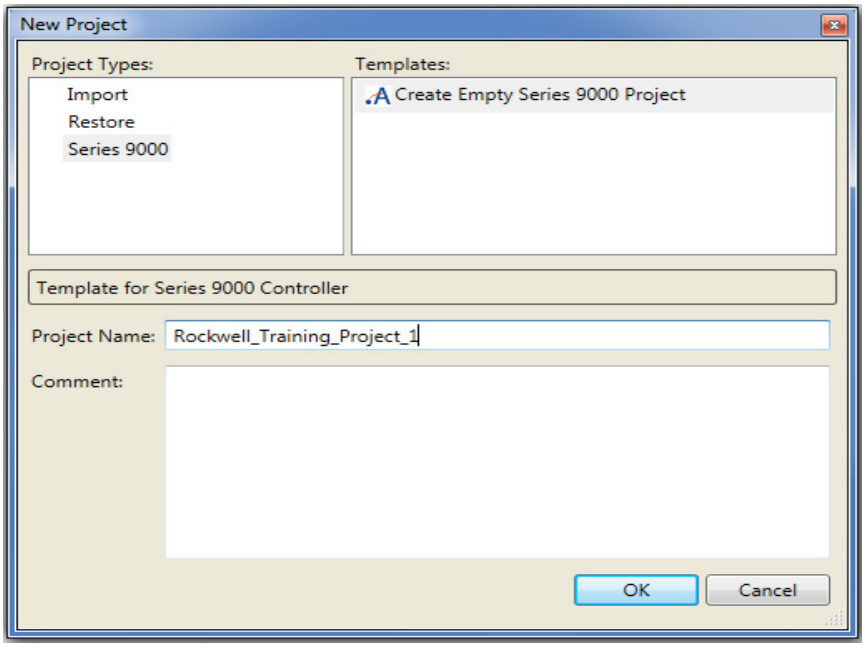


which displays the Help Contents and the Help Viewer Home page.

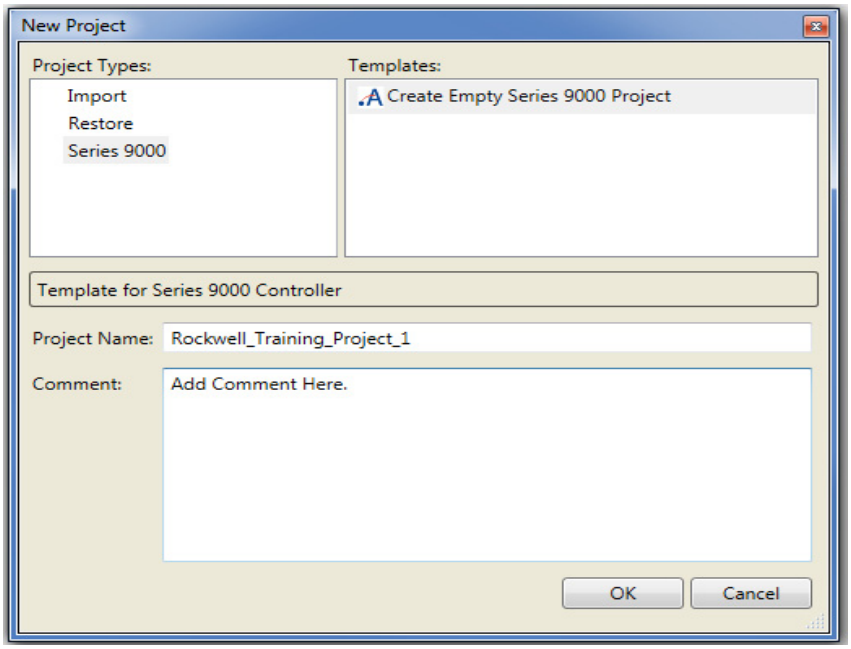


Make a New Project

1. Start the software if it is not already running by clicking **Start** ➔ All Programs ➔ Rockwell Automation ➔ AADvance 2.1➔ AADvance 2.1 or click on the **AADvance 2.1** icon located on desktop.
2. Select **File** then New Project (or press <ctrl> + <shift> + N). The New Project dialog box is displayed.

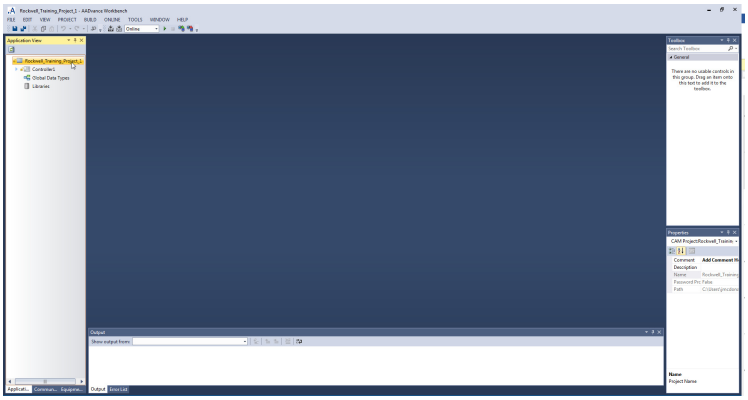


3. In the Project Types field select Series 9000.
4. Enter a project name, in this case Rockwell_ Training_Project_1 and add a comment in the comment field.

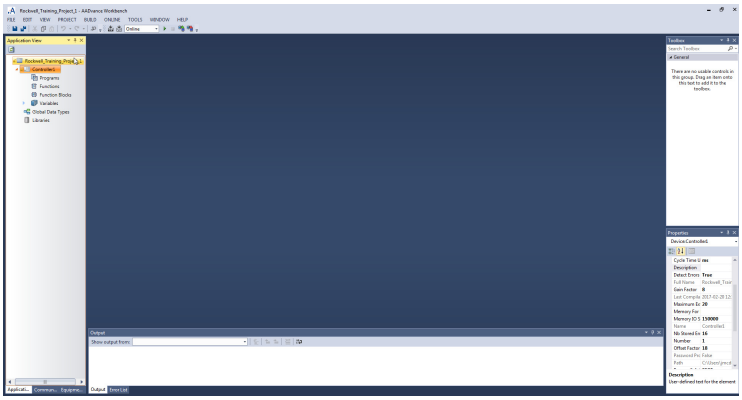


Make sure that the project name begins with a letter or an underscore (_), followed by letters, digits, and single underscores and also follows the rules for a Windows file name. Do not use space characters in the project name.

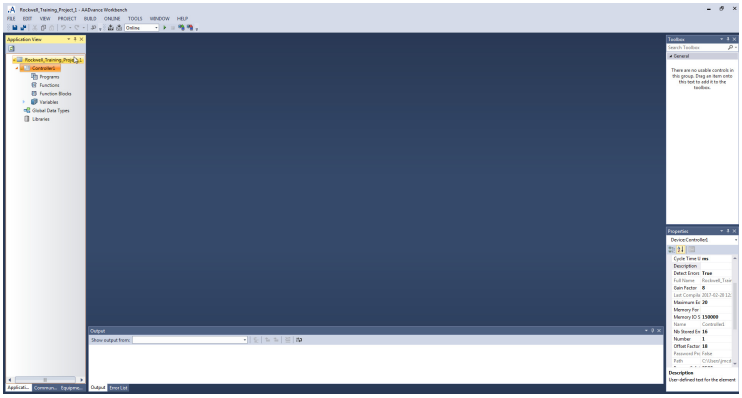
5. Click **OK**. The AADvance Workbench creates a project. The project is now created.
6. Steps 7 through to 12 are included to show how the screen appearance may vary, depending on the selected view. For step 13 to work it doesn't matter which view is selected.
7. Select project node (Application view).



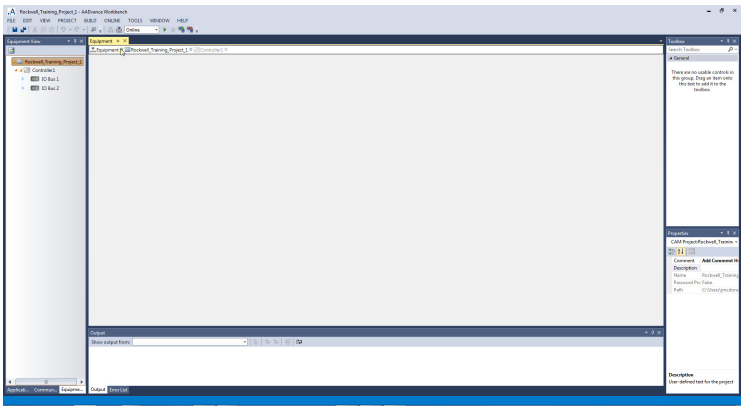
8. Select controller node (Application view).



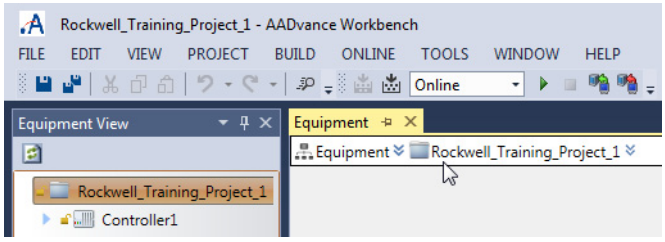
9. Select Equipment view.



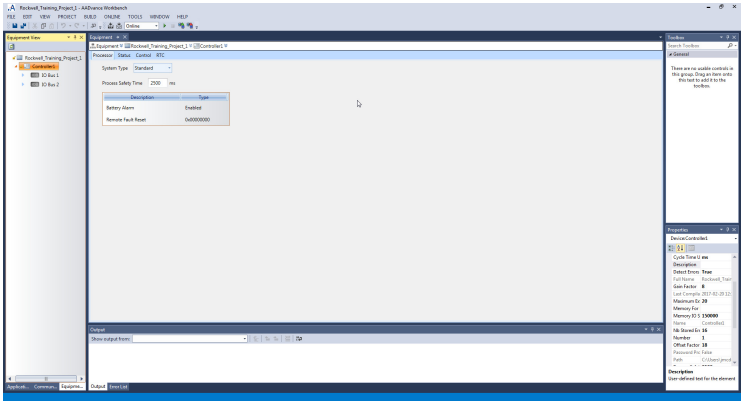
10. Select project node (Equipment view).



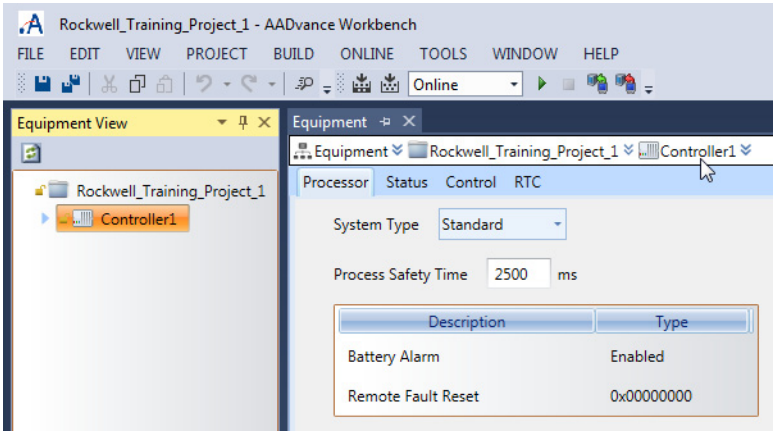
It can also be seen that the large window has an Equipment tab on the left side along the top. Also the equipment node, followed by the project node are seen below the Equipment tab.



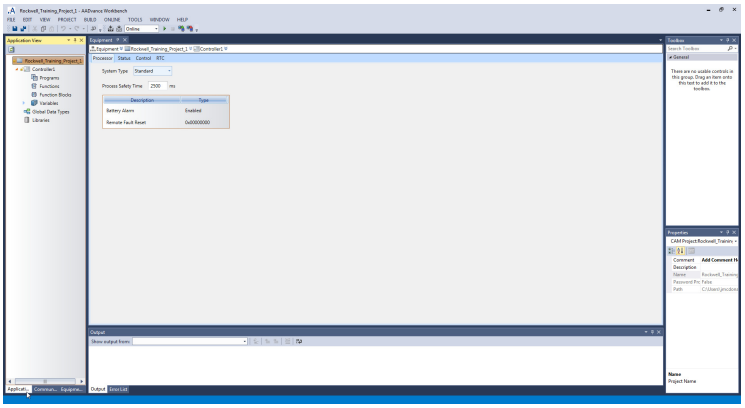
11. Select controller node (Equipment view).



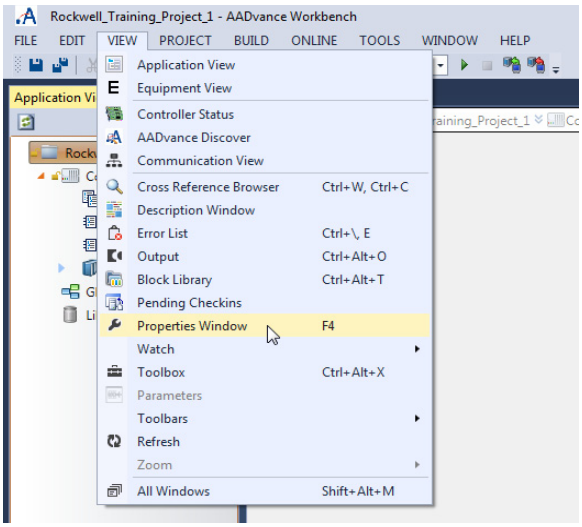
The Controller node is now also seen after the project node.



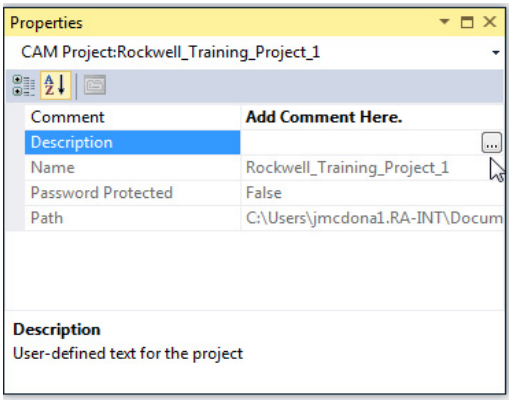
12. Re-selection of Application view does not change the appearance of the large window.



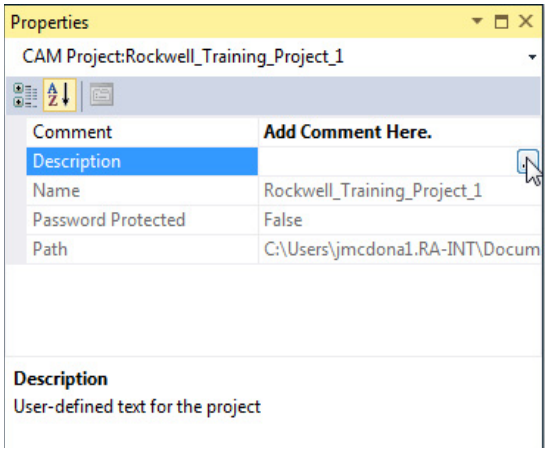
13. Select the required system tree node (project or controller). Initially project node is chosen.
14. Select Properties Window from the **VIEW** tab.



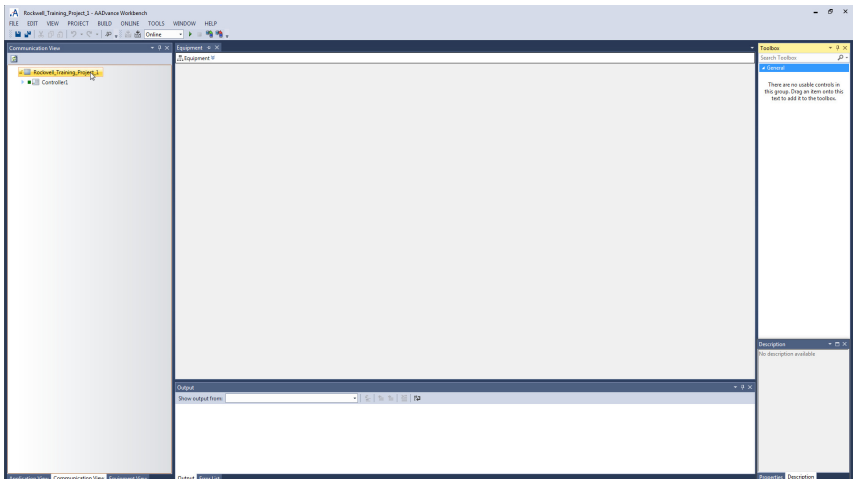
15. Select Description in the properties window.



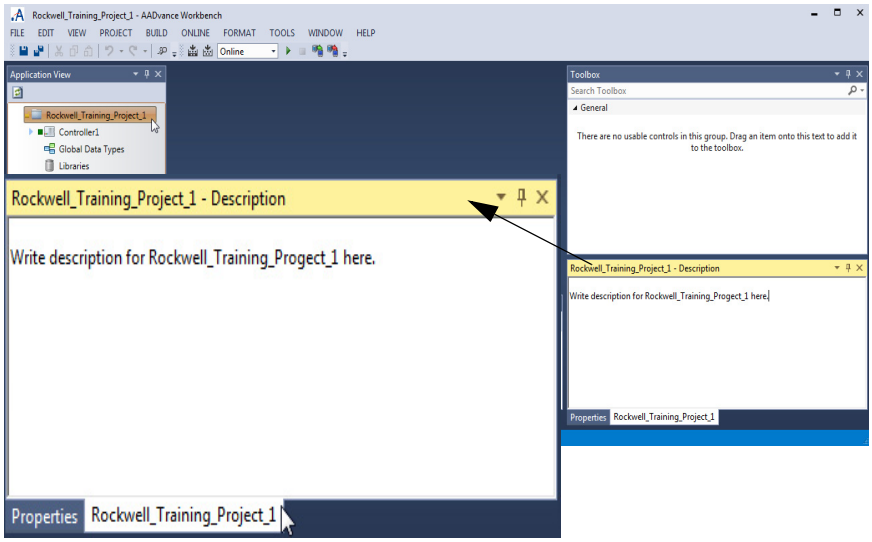
16. Left-click on the button on the right hand side of the next column of the description row.



17. The project name appears before "- Description" on the description window.



18. Add a description to the description window.



Import Existing Project

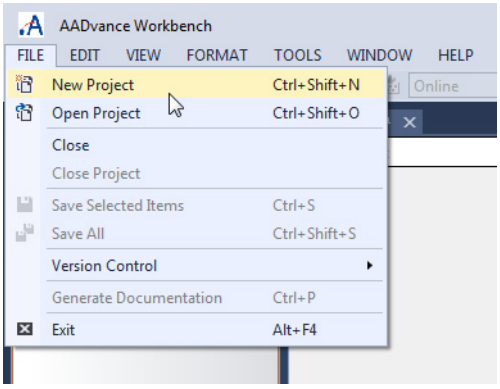
Procedure to import an existing project

To import an existing project, do the following:

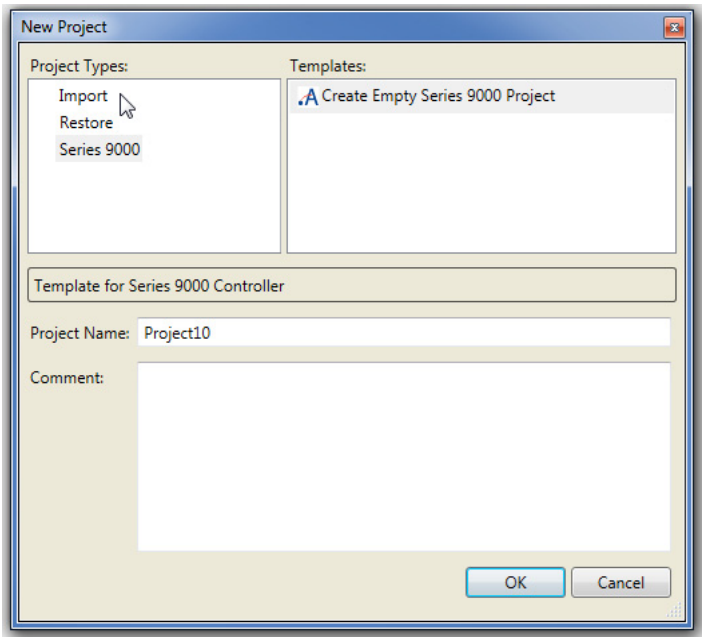
1. Launch the software if it is not already running by clicking:

Start ➔ All Programs ➔ Rockwell Automation ➔ AADvance 2.1 ➔ AADvance 2.1 or Click on the AADvance 2.1 icon located on desktop.

2. Select File, then New Project (or press <ctrl> + <shift> + N).



3. The New Project dialog box is displayed.



4. In the Project Types field, select **Import**.
5. From the Templates list, select **Import AADvance Project**.
6. Enter a project name (add a comment, if required).
   - The project name must begin with a letter or an underscore (_) followed by letters, digits, and single underscores, and must also follows the rules for a Windows file name. Do not use "spaces" in the name
7. Left-click **Browse** to select the database path.
8. In the Select Database File dialog box, locate and select the project database file (*.mdb), and then click **Open**

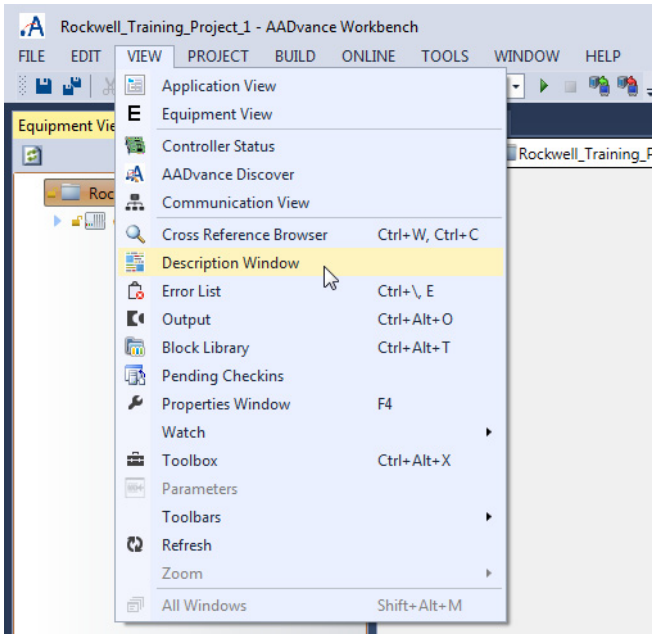
NOTE You may encounter a message asking if you want to update the database to the current version.

After updating the database, the project can no longer be opened in a previous version of the Workbench.

To continue the importation process, click **OK**.

9. In the New Project dialog box, click the **OK** button

If required, select the project node, then Description Window, from the drop-down box for the **VIEW** tab.



Then add a description in the in the description window.

If CIP is configured in the project, select the Ethernet ports for the CIP binding groups.

Compile the project and download it to each controller.

Procedure for Opening An Existing Project

To open a project from the Workbench.

1. From the File menu, click Open Project (or press Ctrl+Shift+O).
2. In the Open Project dialog box, select the required project, and then click **OK**.

The project structure is displayed in the Application View.

To open a project from the Projects directory

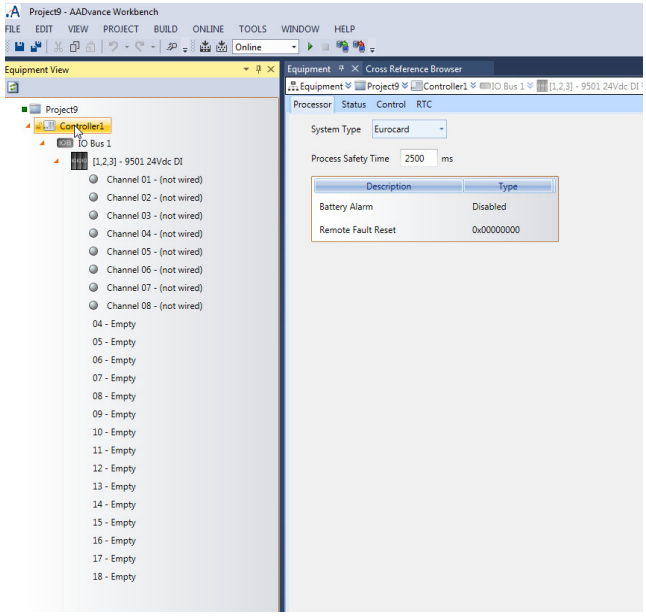
From the Windows Explorer, access the Projects directory and perform one of the following:

- Double-click the required *.aadsln file.
- Drag the *.aadproj or *.aadsln file onto the desktop Workbench icon.

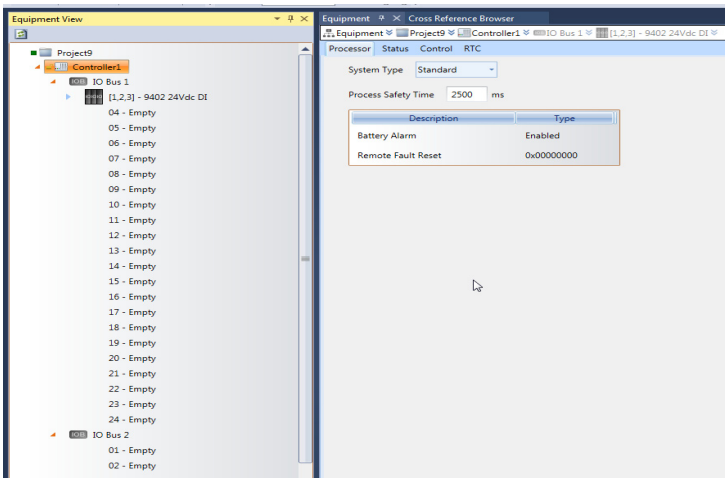
The Workbench opens displaying the required project.

Equipment View

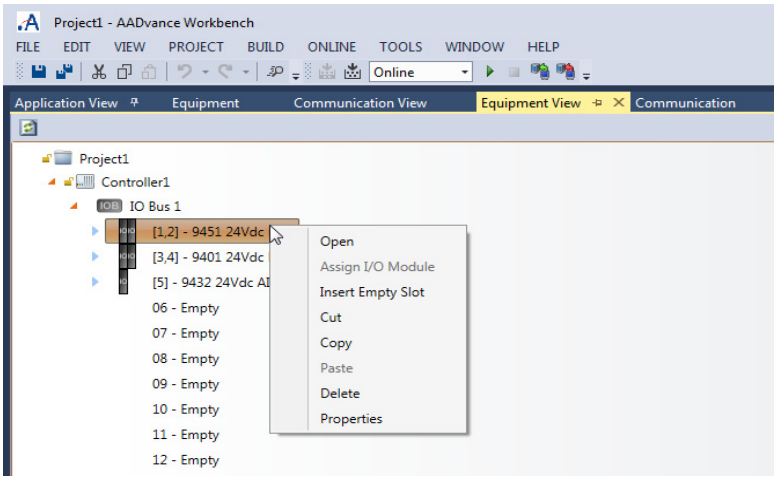
The Equipment view displays a graphical view of the project hardware in a logically organized tree. The tree has a configuration for each AADvance controller and each Eurocard controller defined in a project has a single IO Bus (IO Bus 1) with 18 slots available.



Each standard or mixed controller defined in a project has two IO Buses (IO Bus 1 and IO Bus 2) with 24 slots available for each IO Bus.



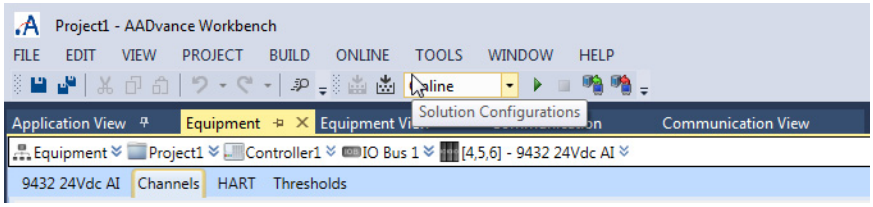
The bus has a series of slots for I/O modules and the tree shows the modules in their correct slots.



The Equipment view enables performing multiple actions, such as the following:

- configure the system type for a controller
- add or edit a controller
- add or edit I/O modules
- move or copy modules in a controller configuration
- copy modules from one controller to another
- configure the process safety time
- wire application variables to I/O channels
- define input thresholds (analogue and digital inputs)
- configure the behavior of output channels
- configure HART communications
- wire application variables to processor variables
- configure and use the real-time clock.

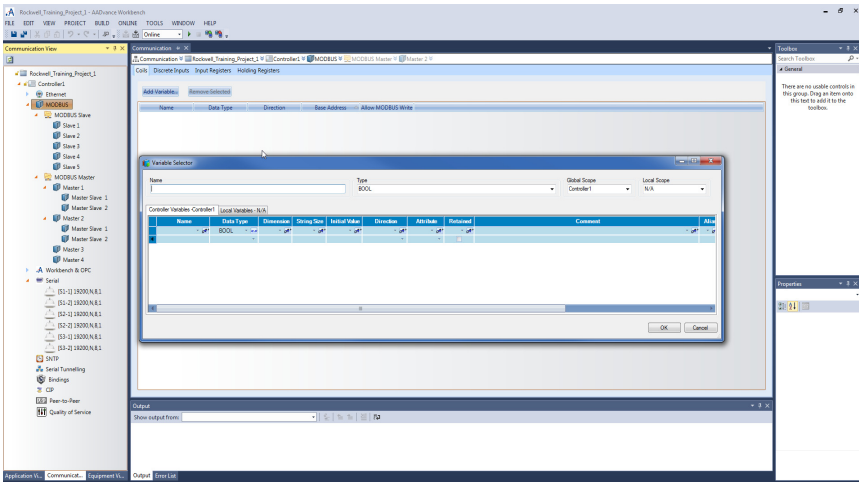
The context menus for the different nodes in the tree contain the actions relevant to the node.



Communication View

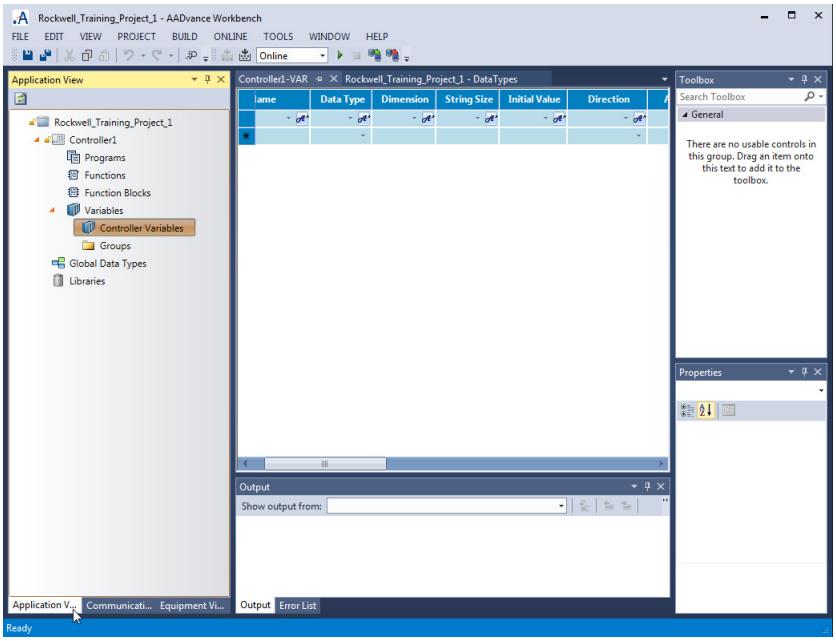
The Communication view is a graphical environment that displays information about communication set up and events. The Communication View displays a tree-like structure showing the available external communication functions and configuration.

In the Communication view you select the system tree nodes to access communication options for the system. In this view you can select specific communication functionality such as Serial Ports or Modbus and configure them for your system.



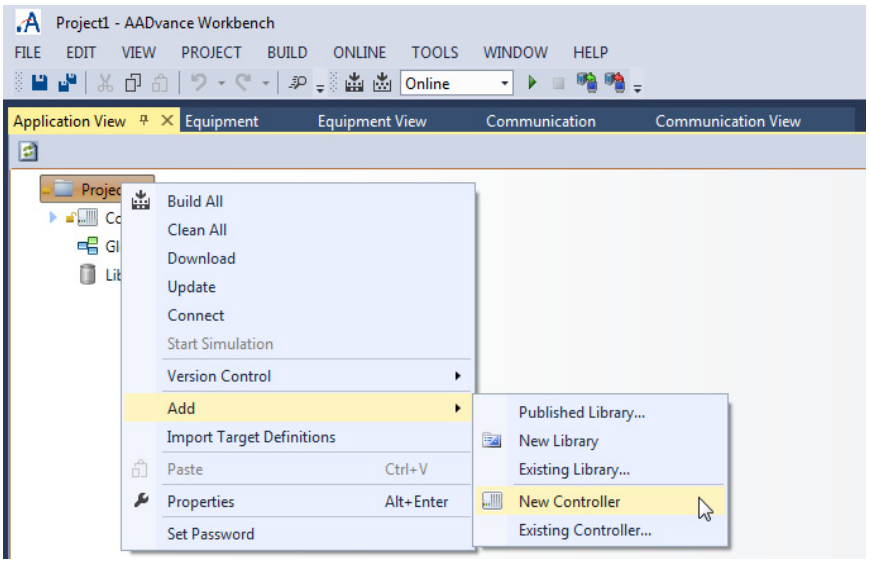
Application View

The Application view is an interface that displays a graphical view of projects in a logically organized tree structure. Projects include controllers, programs, functions, function blocks, controller variables, data types and libraries. A project can include multiple controllers. The physical locations of a project file can differ from the representation in the tree view structure. You can perform the project set up and configuration tasks using the contextual menus.

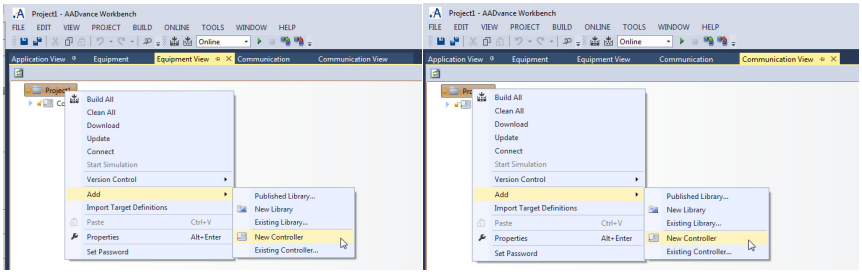


Add a New Controller

To add a new controller to a project:



1. Select the project in the Application view, Communication view or Equipment view.

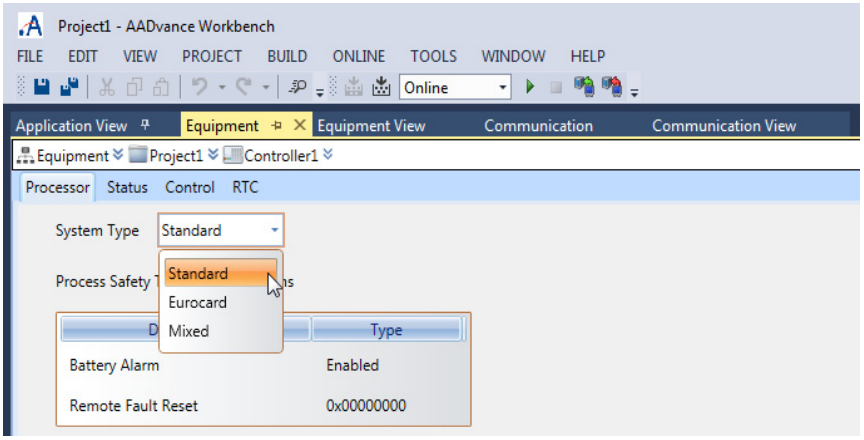


2. Open the contextual menu, select **Add** > New Controller.
   - the AADvance Workbench creates a controller node in the tree.
   - the new controller node has a standard name.
3. Give a meaningful name to the controller. Select the controller in the tree view and open the contextual menu, then select **Rename**.
4. Type in the name for the controller, press **Enter**.

Configure the Controller Type

The AADvance Workbench supports three types of AADvance controller: the standard product, the Eurocard controller and a mixed controller. The 'mixed' controller type lets you configure a standard controller with standard I/O and Eurocard I/O in the two 24-slot I/O buses.

You must set the correct controller type before adding I/O modules. This verifies that you add only the applicable I/O modules. To set the controller type:



1. Select **View** on the main tool-bar / menu bar.
2. Select the Equipment View from the drop-down box.
3. Select and double-click the Controller that the user requires (the initial default is Controller1).
4. Open the System Type drop-down box by clicking in the letterbox to the right of 'System Type' (see above) and select from the following options
   - Standard
   - Eurocard
   - Mixed (mixed type is not currently supported by hardware).

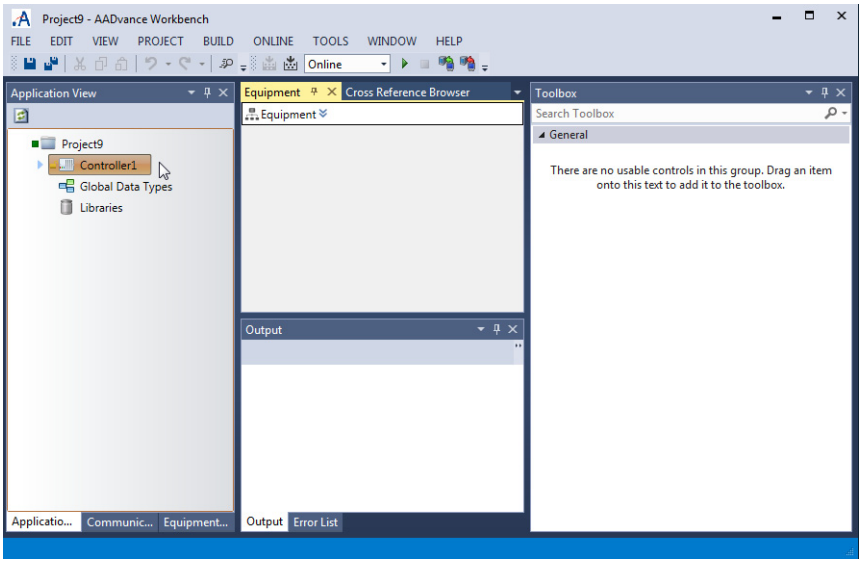
Configure Controller Properties

The Controllers have thirty-eight different elements, which must be configured. These are listed with the available descriptions in the following table.

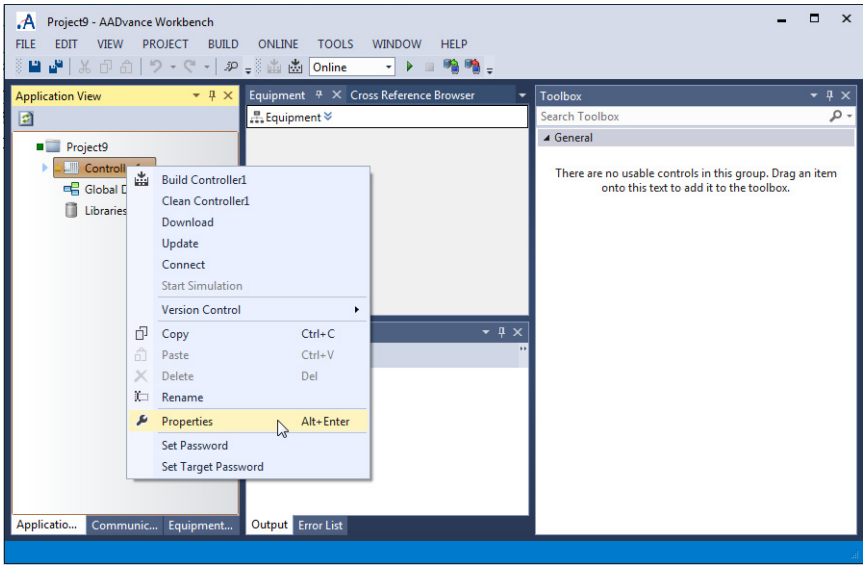
Element	Description
Code	
Code For Simulation	Indication of whether to produce code for simulation for an application.
Compiler Options	Parameters used by the code generator to compile and optimize target code.
• Check Array Index	Indication of whether to verify the validity of array indexes.
• Dump Configuration Files	Indication of whether to generate of resource level files containing debugging information and place them at the root of the resource folder.
• Dump Network	Indication of whether to generate network and configuration level files containing debugging information. The files are placed at the root of the network folder and at the root of the device folder.
• Dump POU Files	Indication of whether to generate resource level files containing debugging information and place them at the root of the resource folder.
• Enable Compiler Verification	
• Function Internal State Enable	Indication of whether to produce internal state information for functions. Functions containing no internal state information denote that the invocation of a function with the same arguments always yields the same values.
• Generate Map File	Indication of whether to generate resource level files containing debugging information. The files are placed at the root of the resource folder.
TIC Code	Indication of whether Target Independent Code is produced by the compiler. TIC code can be executed on virtual machines.
General	
Battery Alarm	
Remote Fault Reset	
Hardware	
Memory IO Structures	The memory space allocated for I/O structures. The default value is 150000 bytes.
Target	Target type to which is attached a device. Changing targets for a device affects all resources attached to the device.
Update IO Device Online	Indication of whether to enable modifying I/O devices and structures while running an application online, for targets supporting online changes.
Info	
Comment	Comment for the element.
Compilation Version	Last application image compiled on the local instance of the workbench.
Description	User defined text for the element.
Full Name	Full name of an element.
Last Compilation Date	Date of the last compilation.
Name	Name of the element.
Number	Unique number identifying a resource within a project.
Password Protected	Indication of whether the element is protected by a password.
Path	Path of the element.
Memory Size For Online Changes	
Code Size	For online changes, the amount of memory reserved for code sequence changes.
Maximum Extra POUs	The maximum number of POUs that can be added during online changes.

Element	Description
SFC States Mem Size	The memory space allocated for step and transition structures. A step requires 40 bytes and a transition requires 20 bytes.
User Variable Size	For online changes, the amount of memory reserved for adding variables data. When generating symbol monitoring information for a POU, the same amount of memory is also reserved for the POU.
Safety	
Process Safety Time	The process safety time configured for the module (milliseconds).
Settings	
Cycle Time	Amount of time given to each cycle. If a cycle is completed within the cycle timing period, the system waits until this period has elapsed before starting a new cycle.
Cycle Time Units	Unit of measure for the cycle time. Possible values are ms (milliseconds) or µs (microseconds). To use µs, the target must support this unit of measure.
Detect Errors	Indication of whether to store errors. You need to define Nb stored errors.
Memory for Retain	Location where retained values are stored (the required syntax depends on the implementation).
Nb Stored Errors	Number of entries, i.e., the size of the queue (FIFO) in which detected errors are stored.
Start Simulation Mode	Indication of whether a resource executes in real-time or cycle to cycle. Real-time mode is the run time normal execution mode where target cycles are triggered by the cycle timing.
Trigger Cycles	Indication of whether a resource cycle executes according to the defined cycle time. If a cycle is completed within the cycle time, the system waits until the cycle time has elapsed before starting a new cycle.
SFC Dynamic Behavior Limits	
Gain Factor	Factor of dynamic behaviour limits determining the amount of memory, allocated by a target at initialization time, designated to manage token moving. The amount of allocated memory is calculated as a linear relation with the number of SFC POUs: Alloc Mem (bytes) = N * NbElmt * Sizeof(typVa) NbElmt = GainFactor * NbOfSFC + OffsetFactor
Offset Factor	Factor of dynamic behaviour limits determining the amount of memory, allocated by a targetat initialization time, designated to manage token moving. The amount of allocated memory is calculated as a linear relation with the number of SFC POUs: Alloc Mem (bytes) = N * NbElmt * sizeof(type Va) NbElmt = GainFactor * NbOfSFC + OffsetFactor
System	
System Type	The Type of the System.

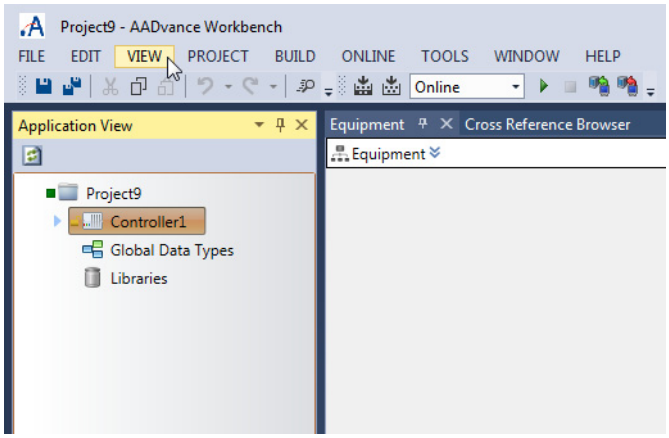
Select the controller, in Application view, Communication view or Equipment view, by covering the legend with the cursor, then left-click the mouse.



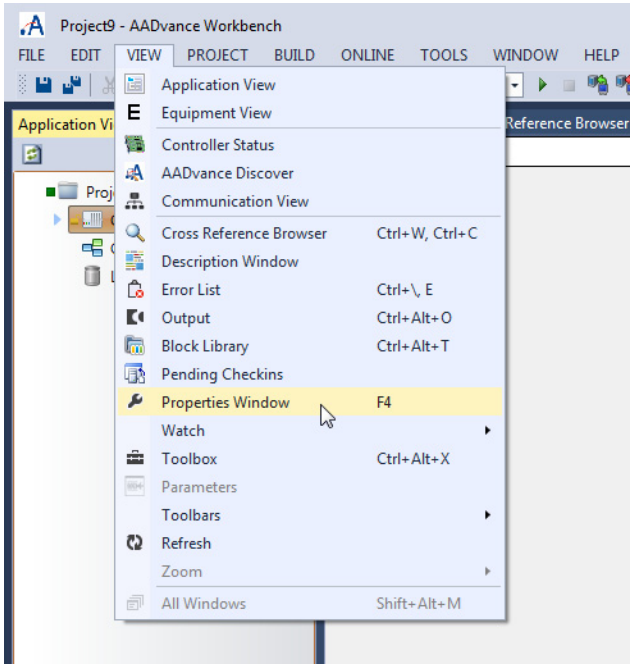
Right-click the mouse over the Controller legend to open a drop-down box, which has Properties as one of the list options. Keyboard shortcut [Alt]+[Enter] can also be used



Alternatively Place the Cursor over the **VIEW** tab and left-click the mouse.

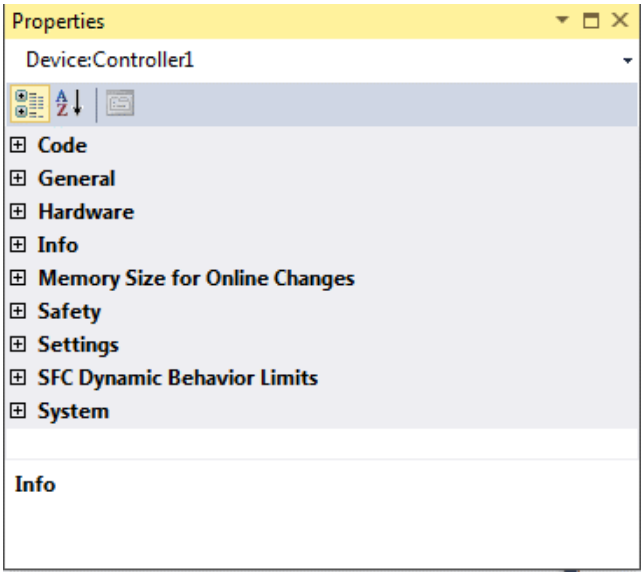


This will reveal the VIEW drop-down options menu, Which has Properties Window as one of the list options.

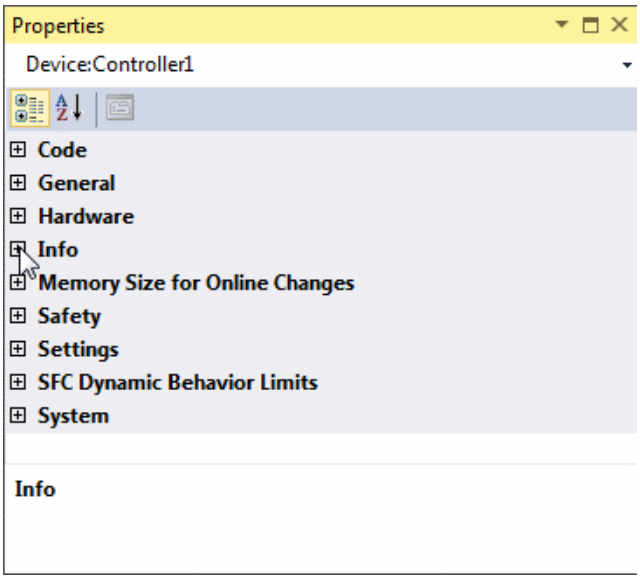


Key board shortcut [F4] can also be used.

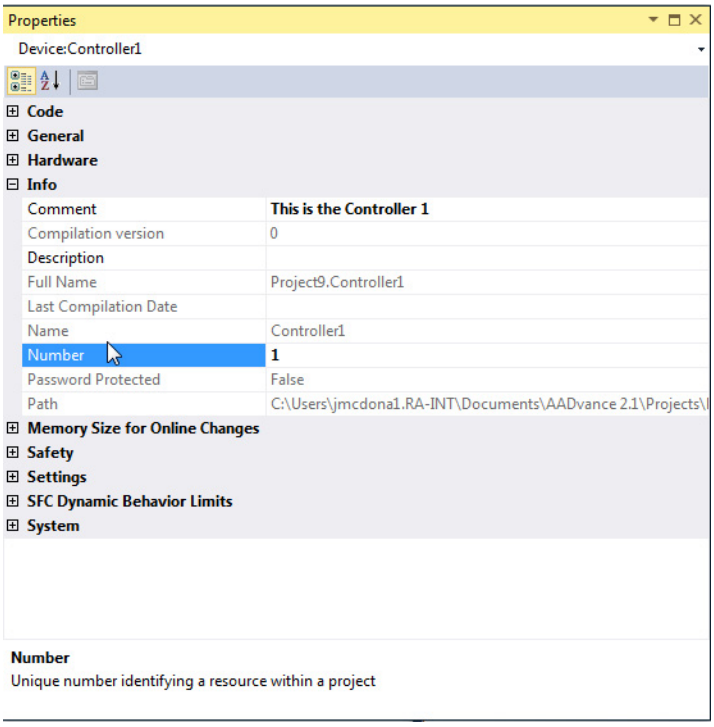
Left clicking the mouse with the cursor over the tab with the Properties legend reveals the Controller Properties configuration options.



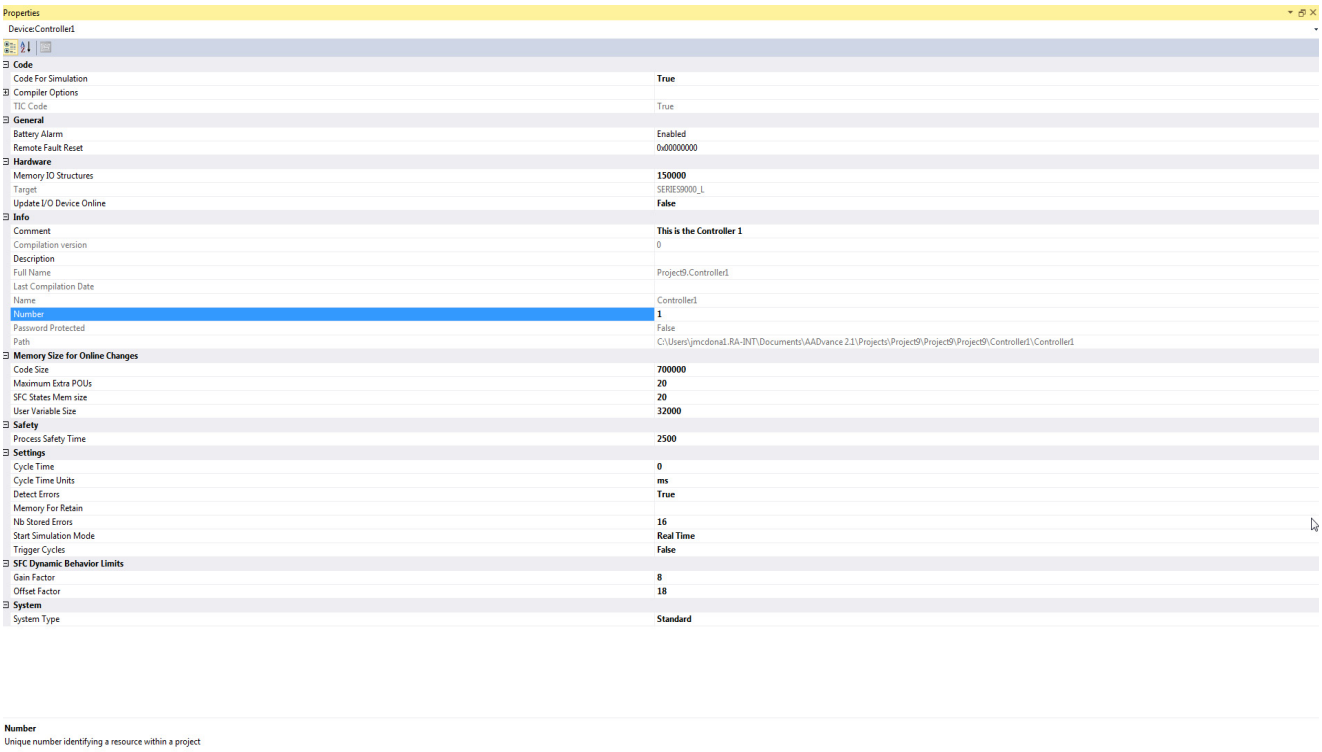
The options can be expanded by left clicking the mouse with the cursor over the buttons with the ⊞ legend.



The window will expand to reveal further options.



Repeating the expansion for all available buttons reveals the full detail of the Properties Window, which incorporates all the elements listed in the table above.



Controller Process Safety Time

The process safety time (PST) defines the maximum time that the controller lets its outputs stay in the ON state after the controller detects important

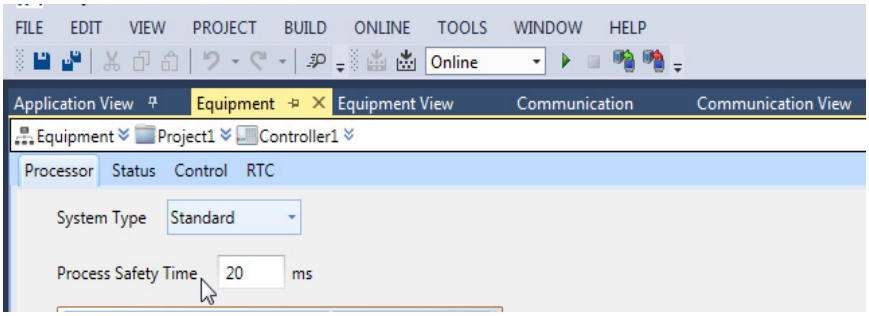
diagnostic or application faults. If the PST expires, the system sets its outputs to their specified safe states.

You have to set the top-level PST for the controller. Each group of I/O modules can inherit this setting or they can use their own setting as an alternative.

Configure the Top-level Process Safety Time

Choose a top-level process safety time which is larger than the application execution time. If you set a process safety time less than the application execution time, the application will not run.

Set the top-level process safety time:



1. Select **View** on the main tool-bar / menu bar.
2. Select **Equipment View** from the drop-down box.
3. Select and double-click the Controller that the user requires (the initial default is Controller1).
4. A default value of 2,500 ms is shown. Enter the time value into the Process Safety Time field. Choose from the following range of values:
   - Minimum: 20 ms
   - Maximum: 60,000 ms (which is 60 seconds or 1 minute)

IMPORTANT	A PST of 20 ms is not sufficient to let the controller run, short PSTs must be tested, including testing the addition of processors. For a large installation, set the top-level PST to 1,500 ms (1.5 s) or larger. This verifies that there is sufficient time for a second or third processor module to educate.

5. Press Ctrl+S to save your changes.

Online Update

You can modify a controller while it is operational and the application is running by using an online update. This process enables you to change the I/O configuration and the data and functions listed in the table. Changes to the I/O configuration must follow the changes to the hardware Configuration.

The default setting for the Online update feature is enabled, however, the online updates for I/O modifications is disabled by default. To perform an I/O online update you must first enable the feature.

ATTENTION: Performing an online update can affect the Safety Function of the system if the change is performed while the application is running. A change should only be attempted if absolutely necessary. Also, if the change is not performed correctly, the change can stop the application. Therefore, before performing an online update alternative safety measures should be implemented and be present for the duration of the update.

Changes Available Using through an Online Update

Controller Data/Function	Action
Bindings	Adding, deleting and editing: • Creating and deleting bindings between variables. • Changing the consume Status Variable and consumption behavior of a binding. Changing the producing variable, consuming variable, or network for a binding creates a new one. • Adjusting the update timeout period in the network parameters. The update timeout period is the maximum time during which the consumer can remain in the update state.
Internal Variables	Adding, deleting and editing internal variables: • When renaming or changing the data type of internal variables, the Workbench creates new variables. Therefore, variables are initialized. • Changing the alias, initial value, group, scope, direction, retain setting address, and comment of variables. When changing the initial values of a read-only internal value, the Workbench reinitializes the variable. When changing the scope of a variable, the Workbench reinitializes the variable. • Modifying the length of string variables. When decreasing the length, the contents of the string is truncated to the new length. • Switching a variable attribute between the input and output attribute. You cannot switch variables between the internal and input/output attribute. • Adding and removing elements in arrays for internal variables. For multi-dimensional arrays, you can only add elements to the first dimension. The Workbench initializes these new elements. Adding elements to other dimensions causes the Workbench to initialize a new array. • Renaming, adding variables, removing variables from the group to which a variable belongs or moving the variable to another group.
Programs	Adding, deleting, renaming and reordering (for execution within the programs section) programs. • When renaming programs, the Workbench detects a CRC mismatch and updates the code on the target for the program and reinitializes all local variables. When renaming SFC programs instances data and local variables are not preserved, i.e. elements are reset to their initial state. • When planning to add programs (other than SFC) using online changes, you need to allocate a sufficient number of maximum extra POUs. • Adding, deleting, renaming steps and transitions as well as modifying the initial step or the flow between elements. When modifying SFC programs, instance data and local variables is preserved, i.e. elements are not reset to their initial state. • Adding, deleting and moving action blocks within steps of SFC programs. Action blocks within steps are executed in order of appearance. You can also change the qualifier of an action block.
Functions and Function Blocks	Adding, deleting and moving function blocks. Adding and deleting function block instances. • Renaming and modifying user-defined functions and function blocks. • Adding, removing and modifying the parameters of user-defined functions and function blocks. When modifying functions and function blocks called by other POUs as well as the calling POUs.
I/O Channels	Adding, deleting and modifying I/O channels. • Changing the wired variables.

Controller Data/Function	Action
I/O Modules	Adding, deleting and modifying I/O modules. • Changing the wired variable. The gain and offset are configured by setting the Raw/Engineering Min and Max vales. • These online changes are only possible if the Update I/O Module Online controller property is set to True. **Note:** This is only possible on the analogue channels.
I/O Variables	Adding, deleting and modifying I/O variables. • Wiring, un-wiring and swapping I/O variables whose data type length (string variables), dimensions (arrays) and address remains unchanged. For these I/O variables you can modify the direction (input or output only), scope, attribute (read, write or free), retain flag alias and comment. • When modifying the direction, I/O variables cannot change to or from the internal type. Note that modifying the Equipment View causes the values of new and removed output I/O variables to be reinitialized.
I/O Configuration	• Change the I/O configuration: remove a module, move a module to a different slot, change a module or add a new module.

Archiving and Restoring Repository Projects

You can archive projects contained in a repository and you can restore projects that were previously archived. Archiving a repository project means copying a project and its modification history and placing it in a compressed file (.VSC). You can restore a compressed file in another repository while retaining the original modification history.

Projects archived from **AADvance2.X** workbench versions cannot be restored in **AADvance1.X** workbench versions.

From the Application view, you can perform version control operations for all elements. From the Communication and Equipment views, you can perform version control operations for projects and controllers.

To archive a repository project

1. From the required view, select the project to archive.
2. From the File menu, choose **Version Control**, then **Archive**.
3. In the Archive dialog, indicate whether to archive the complete project history or the latest project version, then click **Archive**.
4. In the Save As dialog, specify a name and browse for the storage location for the archive file.

The default name and storage location for archive files is the project name and folder.

To restore a repository project

1. From the File menu, choose **New Project**.

   The New Project dialog is displayed.
2. Select *Restore* from the Project Types list, then *Restore Project from an Archive* in the Templates list.
3. Click Browse.
4. In the Select Archive File dialog box, locate the project archive, then click **OK**.

After clicking OK in the Select Archive File dialog box, you are returned to the New Project dialog box.

5. In the New Project dialog box, click **OK**. The project is restored.

Processor Module Battery and Alarm

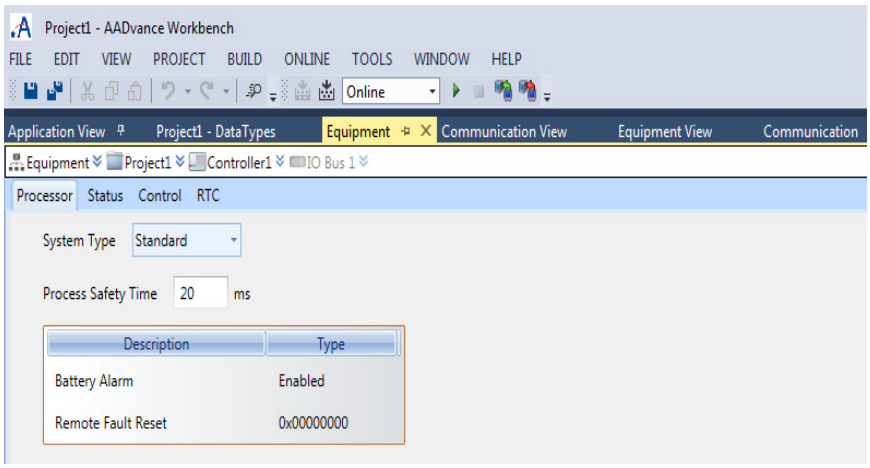
The 9110 processor module has a battery that supports retained variables and the real-time clock when the system power is switched off. The system has a processor battery alarm to inform you when the battery voltage is too low and you need to fit a new battery.

NOTE The Eurocard processor does not have a battery, and the battery alarm is disabled by default.

Configure the Processor Battery Alarm

You can disable the battery alarm, for example for undersea and other inaccessible systems not having a battery.

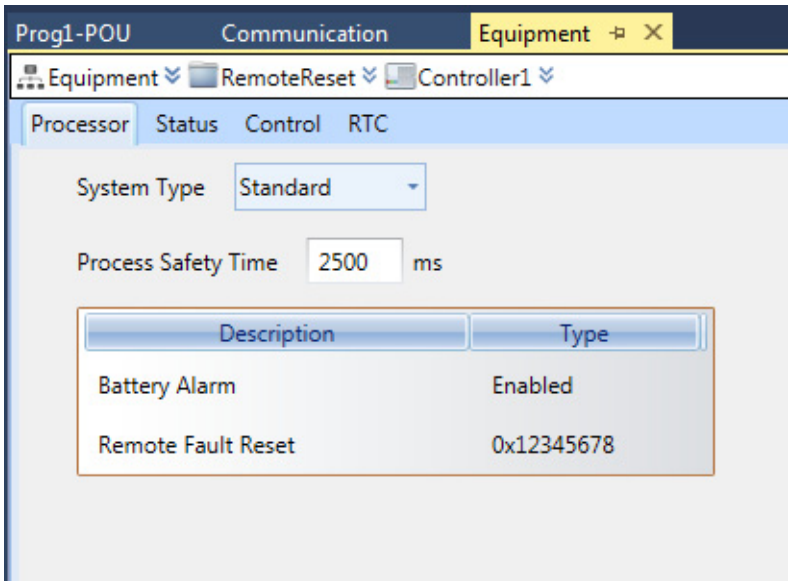
To set the battery alarm:



1. Select **VIEW** on the main tool-bar / menu bar.
2. Select **Equipment View** from the drop-down box.
3. Select and double-click the Controller that the user requires (the initial default is Controller1).
4. Set the Type value for the Battery Alarm to Enabled or Disabled.
   - The default value is Enabled for standard systems.

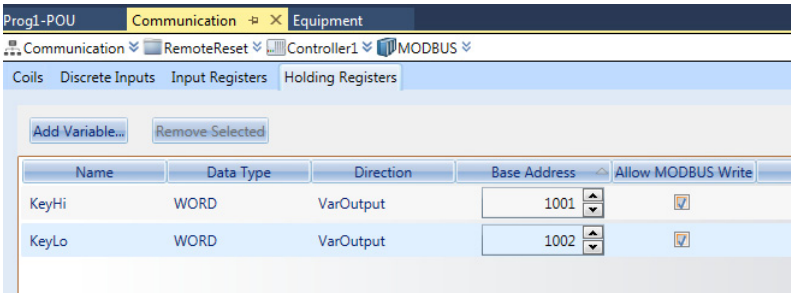
Remote Fault Reset

The remote fault reset enables faults to be cleared on processor modules without the need to physically press the **Fault Reset** button located on the T9110 front panel. As well, you can use the remote fault reset to join processors to a running system. The remote fault reset is required for sub-sea systems since it is impossible to physically manipulate the processors. When enabling Remote Fault Reset, a warning is displayed confirming that you want the controller to be remotely reset.

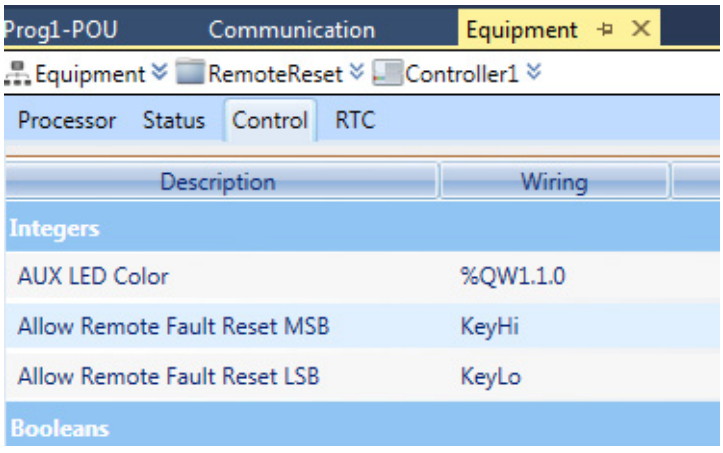


The default setting is disabled, which is 0x00000000. To enable, the authentication key must be a non-zero hexadecimal value, ranging 0x00000001 to 0xFFFFFFFF.

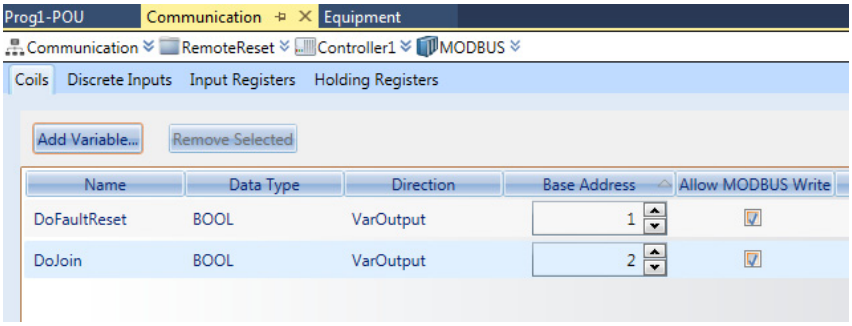
Four output variables must be configured to trigger the reset/join. A communication protocol must be able to write to the variables. For example, when using Modbus, the variables must be configured with Modbus addresses and "The Allow Modbus Write" must be enabled. Two of the variables are of type WORD. When combined these two words must match the configured authentication key before a remote reset/join will be permitted.



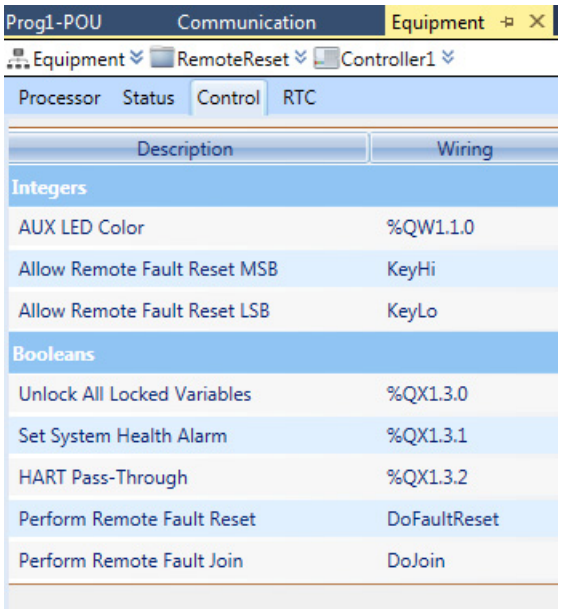
The two WORD outputs are wired to the Processor's Control integers Allow Remote Fault Reset MSB and Allow Remote Fault Reset LSB:



Two Boolean outputs are also required. One to trigger a remote reset, the other a join. These should be configured as Modbus coils:



The outputs should be wired to the Processor's Control Booleans, Perform Remote Fault Reset and Perform Remote Fault Join:



To perform a remote reset/join, the remote client must write the correct values to the authentication key output variables, then cause a rising edge on either the remote reset, or remote join Boolean.

Declaring Project Variables

Project variables can use the standard IEC 61131-3 data types, or you can define your own, based on the IEC 61131-3 types. You can also define structures using standard IEC 61131-3 types, arrays, and other user types. This chapter explains how to create and adjust variables for your project.

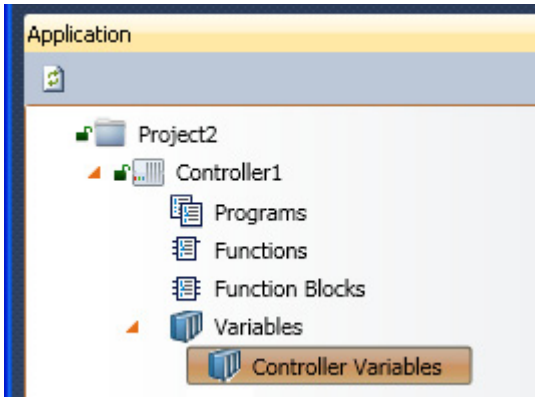
About Variables

The AADvance® Workbench maintains details of all your application variables. You can define your variables at the start of a new project, or while creating a program. Include processor and I/O module variables and the I/O channel variables.

The AADvance Workbench automatically assigns a set of variable elements when you select the data type.

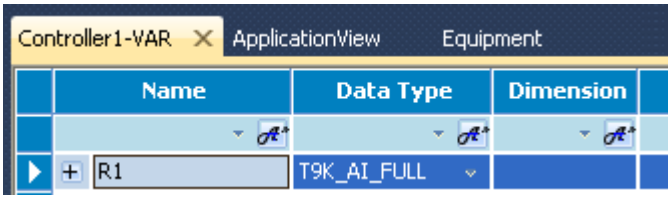
Declare or Change Variables

Use the Variables window (dictionary) to declare new variables and to change the properties of existing variables. Do the following:

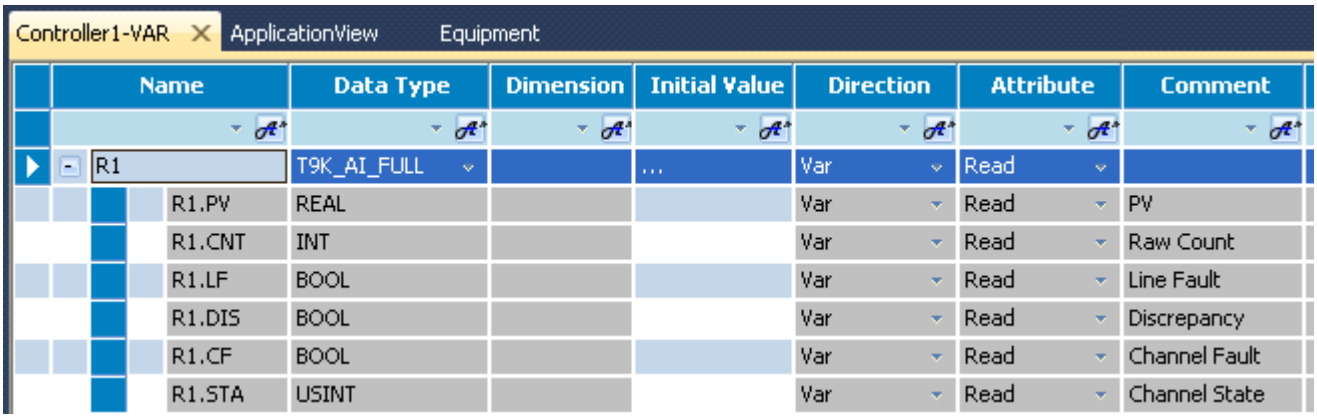


1. Use the Application view to select the applicable variables. For example, go the controller and then Variables to navigate to the Controller Variables shown in the illustration.
2. Either, double-click Controller Variables or right-click Controller Variables and then click **Open**.
   - The Variables window opens in the content window.

   You can also use the variable selector window to declare and change variables. You can open the variable selector window from more than one location, such as the MODBUS, OPC and SOE editors.
3. To declare a new variable, go to the bottom row of the grid (identified by an '*') and enter the name of the variable, then set the Data Type from the drop-down list.



4. If you set the Data Type to a structure, click the **+** symbol to show the elements of the variable.
   - The AADvance Workbench assigns a set of elements to match the structure.



5. Define the general properties for the variable and for each element to use in your application.
   - You can define an array. When declaring the variable, use the Dimension property to define the array.
   - For Boolean variables, you can specify user text messages for the FALSE and TRUE conditions.
   - For string variables, you specify the string length in the String Size property (Defaults to 80).
   - For complex variables, wired elements set to read or write must have the Attribute property set to Read/Write and the Direction set to the applicable value.
   - If you set the Direction of a variable to VarInput, you must set the Attribute property to Read.
   - If you set the Direction of a variable to VarOutput, you must set the Attribute property to Write or Read/Write.
6. If you use the variable for MODBUS, OPC or SOE, use the Communication view or the Equipment view to access the applicable MODBUS, OPC, or SOE editor to define additional parameters.
7. To adjust an existing variable, change the variable properties.

Properties for AADvance Project Variables

Name: A unique name for the variable. The name must start with a letter or an underscore character followed by letters, digits and single underscore characters. The name cannot use two underscore characters one after the other and must be less than 128 characters long.

- Example: di_full

Data Type: A standard IEC 61131 data type, AADvance (9K) structure, or a user-defined structure.

Dimension: The size (number of elements) of an array.

- Example: [1..4, 1..7]

String Size: The maximum number of characters the variable can contain, if the Data Type is STRING. The maximum length of a string is 252 characters, the last 3 bytes are reserved for maximum size (252), current length (1 to 252) and the null/termination character.

Initial Value: Initial variable value, numeric or textual.

Direction: One of the following values:

- VarInput: A variable connected to an input device (refreshed by the system)
- VarOutput: A variable connected to an output device
- Var: An internal variable updated by the programs or communication

Attribute: One of the following values:

- Read: A read-only variable with an initial value. The programs cannot change the variable.
- Write: A write-only variable with an initial value. The programs can only write to the variable.
- Read/Write: A variable that can be used for reading or writing.

Retained: TRUE or FALSE:

- TRUE: The controller stores the value in non-volatile memory and restores the value if the controller stops and restarts.

If the processor battery voltage is too low or the battery is not fitted, retained data will be lost, setting Retained to TRUE has no effect.

Comment: User comments. Not used by the application.

Alias: A name used in the ladder diagram and function block. The first part of the text in the comment field, truncated at 16 characters or before the first ':' character, whichever comes first.

Wiring: The I/O point to which the variable is wired to.

- The wiring property uses the syntax of a directly represented variable to show a channel that is not linked to a declared variable.
- The identifier of a directly represented variable is always %.
- Example: %IXs.c; where 'I' is an input, 'X' is a Boolean, 's' is the address of the module and 'c' is the channel number.

Allow OPC Write: Permits OPC clients to write to the variable (configured in OPC view).

Groups: Group name, or none.

Message TRUE: A user text message for the TRUE value of a Boolean variable, which can be used for Sequence of Events or OPC.

- Example: 'Pump on'.

Message FALSE: A user text message for the FALSE value of a Boolean variable, which can be used for Sequence of Events or OPC.

- Example: 'Pump off'.

Standard IEC 61131-3 Data Types

You can declare variables using the following standard IEC 61131-3 data types.

Name	Description
BOOL	Boolean logic (0 or 1), other value s: Off or On; False or True; Min or Max
BYTE	Byte value (8 bit); a group of binary digits or bits (usually eight) operated on as a unit.A byte considered as a unit of memory size.
DATE	Date value (32 bit)
DINT	Signed (±) double integer continuous value (32 bit): Values; -2147483648 to +2147483647
DWORD	Double word value (32 bit)
INT	Signed (±) single integer continuous value (16 bit) Values: -32768 to +32767
LINT	Long integer continuous value (64 bit) Values: -9223372036854775808 to +9223372036854775807
LREAL	Long real (floating) continuous value (64 bit) is quoted to 12 significant figures. This has three ranges as follows: • -1.79769E+308 to -5.0E-324 (negative values) • 0 • +5.0 E-324 to +1.79769E+308 (positive values)
LWORD	Long word value (64 bit)
REAL	Real (floating) continuous value (32 bit) is quoted to six significant figures. This has three ranges as follows: • -3.402823E+38 to -1.4E-45 (negative values) • 0 • +1.14E-45 to +3.402823E+38 (positive values)
SINT	Signed (±) short integer continuous value (8 bit) Values: -128 to +127
STRING	Character string having a defined size
TIME (1)	Time interval from 0 to 2^{32} ms (32 bit)
UDINT	Unsigned double integer continuous value (32 bit) Values: 0 to 4294967295
UINT	Unsigned single integer continuous value (16 bit) Values: 0 to 65535
ULINT	Unsigned long integer continuous value (64 bit) Values: 0 to 18446744073709551615
USINT	Unsigned short integer continuous value (8 bit) Values: 0 to 255
WORD	Word value (16 bit)

(1) The TIME data type cannot store dates.

Built-in Data Types

You can declare variables using the following built-in data types, and also the standard IEC 61131-3 data types.

FB types are used to declare function block instances. They are named sets of data used by the function block.

Name	Purpose or Application(s)
AVERAGE (FB Type)	Running average over N samples
BLINK (FB Type)	Blinking Boolean signal
CMP (FB Type)	Full comparison function block

Name	Purpose or Application(s)
CTD (FB Type)	Down counter
CTU (FB Type)	Up counter
CTUD (FB Type)	Up-down counter
DERIVATE (FB Type)	Differentiation according to time
F_TRIG (FB Type)	Falling edge detection
HYSTER (FB Type)	Boolean hysteresis on difference of reals
INTEGRAL (FB Type)	Integration over time
KvbConsNetStatus (FB Type)	Function Block (FB) to get status of a network link to a producer address
KvbProdNetStatus (FB Type)	Function Block (FB) to get status of a network link to a consumer address
LIM_ALRM (FB Type)	High/low limit alarm with hysteresis
ModbusServiceStatus (FB Type)	Modbus service status
R_TRIG (FB Type)	Rising edge detection
RS (FB Type)	Reset dominant bistable
SIG_GEN (FB Type)	Signal generator
SoeServiceStatus (FB Type)	Sequence of events service status
SR (FB Type)	Set dominant bistable
STACKINT (FB Type)	Stack of integer analogs
TOF (FB Type)	Off-delay timing
TON (FB Type)	On-delay timing
TP (FB Type)	Pulse timing
CONNECTION_STATUS	ControlLogix® CIP over Ethernet/IP consumed variable
T9K_AI_COMPACT	9000 series analogue input channel
T9K_AI_FULL	9000 series analogue input channel
T9K_AI_HART	9000 series HART data for analogue input channel
T9K_AI_HART_FULL	9000 series HART data for analogue input channel
T9K_AO_COMPACT	9000 series analogue output channel
T9K_AO_FULL	9000 series analogue output channel
T9K_DI_COMPACT	9000 series digital input channel
T9K_DI_FULL	9000 series digital input channel
T9K_DO_COMPACT	9000 series digital output channel
T9K_DO_FULL	9000 series digital output channel
T9K_FI_COMPACT	9000 series frequency input channel
T9K_FI_FULL	9000 series frequency input channel
T9K_TA_GROUP_STATUS	9000 series module status data

Notes:

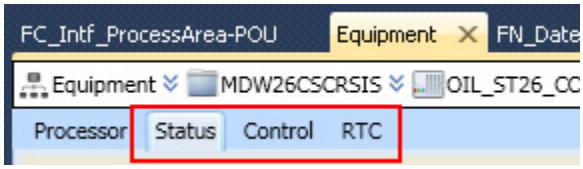
Setting Processor Variables

This chapter describes the processor variables and explains how to set up and use them in your application. The procedures use the Equipment View in the AADvance® Workbench.

Processor Variables

The 9110 processor module has status and control variables for use by the application. Status variables get data from the processor module, and control variables set data in the processor module.

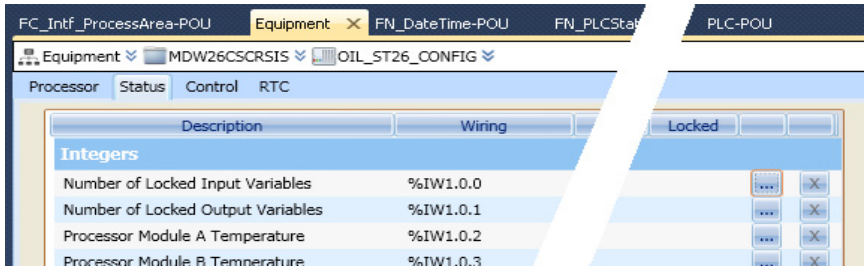
The AADvance Workbench presents the processor variables in three tabs on the **Equipment** tab of the content window:



- The **Status** tab holds status integers and status Booleans, which supply data about the controller for the application;
- The **Control** tab holds control integers and control Booleans enabling the application to send data to the controller;
- The **RTC** tab holds variables for the real-time clock: status variables, program variables and control variables. Status variable supply date/time data to the application. Program variables hold date/time data to be written to the real-time clock. Control variables control updates to the clock and set the clock.

To Wire a Processor Variable

To wire a 9110 processor variable:



1. From the Equipment view, double-click the required controller. The 'Equipment Property' page is displayed.
2. From the Equipment Property page, click one of the following tabs:
   - Status
   - Control

- RTC

  Each tab displays different processor variables performing specific functions.

3. From the **Processor Variable** tab, locate the required variable and in the wire column click .
   The 'Variable Selector' is displayed.
4. From the 'Variable Selector' select the required variable and click **OK**.
   The 'Processor Variable' is now wired.

To Unwire a Processor Variable

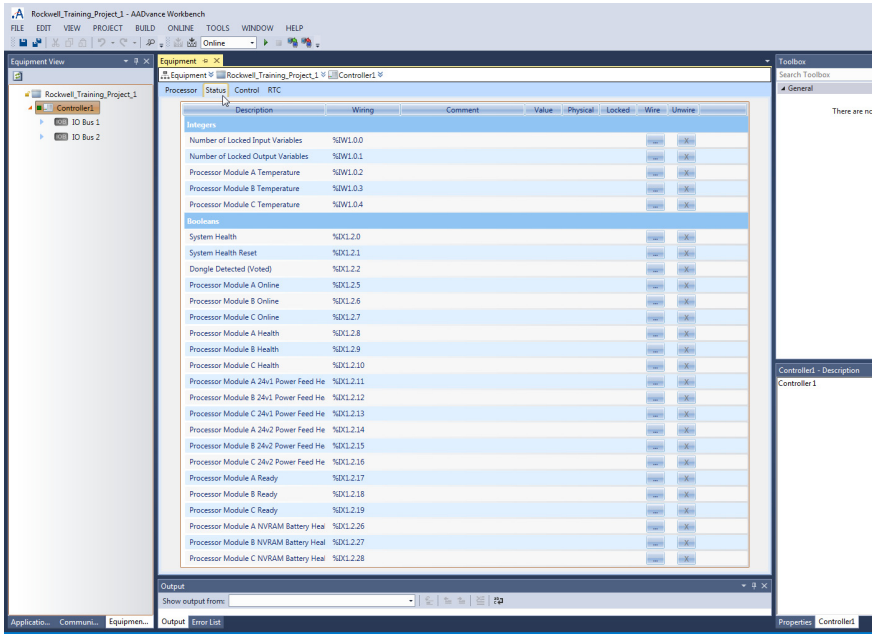
1. From the Equipment view, double-click the required controller.
   The 'Equipment Property' page is displayed.
2. From the 'Equipment Property' page, click one of the following tabs:
   - Status
   - Control
   - RTC
3. From the **Processor Variable** tab, locate the wired variable and in the '**Unwire**' column click .
   The Processor Variable is unwired.

Status Variables

The Status Variables include:

1. Status Integers, which comprise
   - Number of Locked Input Variables
   - Number of Locked Output Variables
   - Processor Module A Temperature
   - Processor Module B Temperature
   - Processor Module C Temperature.
2. Status Booleans, which comprise
   - System Health
   - System Health Reset, Voted 1 out of 3 (1003)
   - Dongle detected (Voted)
   - Processor Module A online
   - Processor Module B online
   - Processor Module C online
   - Processor Module A Health
   - Processor Module B Health
   - Processor Module C Health
   - Processor Module A 24 v1 Power Feed Health
   - Processor Module B 24 v1 Power Feed Health
   - Processor Module C 24 v1 Power Feed Health
   - Processor Module A 24 v2 Power Feed Health
   - Processor Module B 24 v2 Power Feed Health
   - Processor Module C 24 v2 Power Feed Health
   - Processor Module A Ready
   - Processor Module B Ready
   - Processor Module C Ready

- Processor Module A NVRAM Battery Health
- Processor Module B NVRAM Battery Health
- Processor Module C NVRAM Battery Health.



Status Integers

The Status Integers supply data about the controller to the application

Number of Locked Input Variables

Direction: input to application from controller

Type: word

Values:

- 0 to 65,535

Description:
Reports the quantity of input variables having been locked by the user. The top limit of 65,535 represents the capacity of the variable; the real limit is the number of variables in the application.

Number of Locked Output Variables

Direction: input to application from controller

Type: word

Values:

- 0 to 65,535

Description:
Reports the quantity of output variables having been locked by the user. The top limit of 65,535 represents the capacity of the variable; the real limit is the number of variables in the application.

Processor Module A Temperature

Direction: input to application from controller

Type: word

Values:

- 0 to 65,535

Description:
Reports the temperature of the 9110 processor module in the given slot in degrees centigrade. Set to 0 (zero) if no processor module is present.

Processor Module B Temperature

Direction: input to application from controller

Type: word

Values:

- 0 to 65,535

Description:
Reports the temperature of the 9110 processor module in the given slot in degrees centigrade. Set to 0 (zero) if no processor module is present.

Processor Module C Temperature

Direction: input to application from controller

Type: word

Values:

- 0 to 65,535

Description:
Reports the temperature of the 9110 processor module in the given slot in degrees centigrade. Set to 0 (zero) if no processor module is present.

Status Booleans

The Status Booleans supply data about the controller to the application.

System Health

Direction: Input to application from controller

Type: Boolean

Values:

- TRUE = All installed processor and I/O modules are healthy and System Health variable is FALSE (See Control Booleans). The processor module System Healthy LED is green.
- FALSE = One or more of the installed processors and/or I/O modules are reporting a module health problem or the System Health variable is TRUE (See Control Booleans). The processor module System Healthy LED is red.

NOTE The System Health alarm can be reset after a fault in a system having at least one healthy processor or I/O module in a module group.

System Health Reset (Voted 1oo3)

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = The **Fault Reset** button on a 9110 processor module has been pressed in the cycle before.
- FALSE = No **Fault Reset** button is active.

The default value is FALSE.

Description:
Reports that the **Fault Reset** button on a processor module has been pressed. The system health reset is triggered by pressing the button but the value does not change to TRUE until the start of the next application cycle. The value remains TRUE for the cycle and then reverts to FALSE even if the button has been pressed throughout.

Dongle Detected (Voted)

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = One or more 9110 processor modules can see a program enable key at the KEY connector on the 9100 processor base unit.
- FALSE = No processor module can see a program enable key.

Description:
Reports the presence or absence of a program enable key.

Processor Module A Online

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = The 9110 processor module in the given slot is online
- FALSE = The processor module is offline

The default value is TRUE

Description:
Reports that a processor module in a dual or triple modular redundant configuration is present and is communicating through the inter-processor link to one or both of its peers. Reports that a simplex processor module is present.

Processor Module B Online

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = The 9110 processor module in the given slot is online
- FALSE = The processor module is offline

The default value is TRUE

Description:
Reports that a processor module in a dual or triple modular redundant configuration is present and is communicating through the inter-processor link to one or both of its peers. Reports that a simplex processor module is present.

Processor Module C Online

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = The 9110 processor module in the given slot is online
- FALSE = The processor module is offline

The default value is TRUE

Description:
Reports that a processor module in a dual or triple modular redundant configuration is present and is communicating through the inter-processor link to one or both of its peers. Reports that a simplex processor module is present.

Processor Module A Health

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = The 9110 processor module in the given slot is healthy and its Healthy LED indicator is green.
- FALSE = The processor module is faulty and its Healthy LED indicator is red.

Description:
Reports the health status of a processor module.

Processor Module B Health

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = The 9110 processor module in the given slot is healthy and its Healthy LED indicator is green.
- FALSE = The processor module is faulty and its Healthy LED indicator is red.

Description:
Reports the health status of a processor module.

Processor Module C Health

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = The 9110 processor module in the given slot is healthy and its Healthy LED indicator is green.
- FALSE = The processor module is faulty and its Healthy LED indicator is red.

Description:
Reports the health status of a processor module.

Processor Module A 24 V1 Power Feed Health

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = power feed voltage is within specifications (18 to 32 Vdc).
- FALSE = power feed is outside specifications.

Description:
Reports the health of power feed 1 (nominal 24 Vdc) to the 9110 processor module in the given slot.

Processor Module B 24 V1 Power Feed Health

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = power feed voltage is within specifications (18 to 32 Vdc).
- FALSE = power feed is outside specifications.

Description:
Reports the health of power feed 1 (nominal 24 Vdc) to the 9110 processor module in the given slot.

Processor Module C 24 V1 Power Feed Health

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = power feed voltage is within specifications (18 to 32 Vdc).
- FALSE = power feed is outside specifications.

Description:
Reports the health of power feed 1 (nominal 24 Vdc) to the 9110 processor module in the given slot.

Processor Module A 24 V2 Power Feed Health

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = power feed voltage is within specifications (18 to 32 Vdc).
- FALSE = power feed is outside specifications.

Description:
Reports the health of power feed 2 (nominal 24 Vdc) to the 9110 processor module in the given slot.

Processor Module B 24 V2 Power Feed Health

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = power feed voltage is within specifications (18 to 32 Vdc).
- FALSE = power feed is outside specifications.

Description:
Reports the health of power feed 2 (nominal 24 Vdc) to the 9110 processor module in the given slot.

Processor Module C 24 V2 Power Feed Health

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = power feed voltage is within specifications (18 to 32 Vdc).
- FALSE = power feed is outside specifications.

Description:
Reports the health of power feed 2 (nominal 24 Vdc) to the 9110 processor module in the given slot.

Processor Module A Ready

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = The 9110 processor module in the given slot is synchronized (see description).
- FALSE = The processor module is out of synchronization or missing.

Description:
Reports that a processor module in a dual or triple modular redundant configuration is present and is synchronized with one or both of its peers. Reports that a simplex processor module is present.

Processor Module B Ready

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = The 9110 processor module in the given slot is synchronized (see description).
- FALSE = The processor module is out of synchronization or missing.

Description:
Reports that a processor module in a dual or triple modular redundant configuration is present and is synchronized with one or both of its peers. Reports that a simplex processor module is present.

Processor Module C Ready

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = The 9110 processor module in the given slot is synchronized (see description).
- FALSE = The processor module is out of synchronization or missing.

Description:
Reports that a processor module in a dual or triple modular redundant configuration is present and is synchronized with one or both of its peers. Reports that a simplex processor module is present.

Processor Module A NVRAM Battery Health

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = The backup battery in the 9110 processor module in the given slot is present and its voltage is within satisfactory limits.
- FALSE = The voltage of the backup battery is low or the battery is not there.

Description:
Reports the health status of the backup battery in a processor module. The battery voltage is checked at start up, then tested again every 24 hours (elapsed time).

Processor Module B NVRAM Battery Health

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = The backup battery in the 9110 processor module in the given slot is present and its voltage is within satisfactory limits.
- FALSE = The voltage of the backup battery is low or the battery is not there.

Description:
Reports the health status of the backup battery in a processor module. The battery voltage is checked at start up, then tested again every 24 hours (elapsed time).

Processor Module C NVRAM Battery Health

Direction: input to application from controller

Type: Boolean

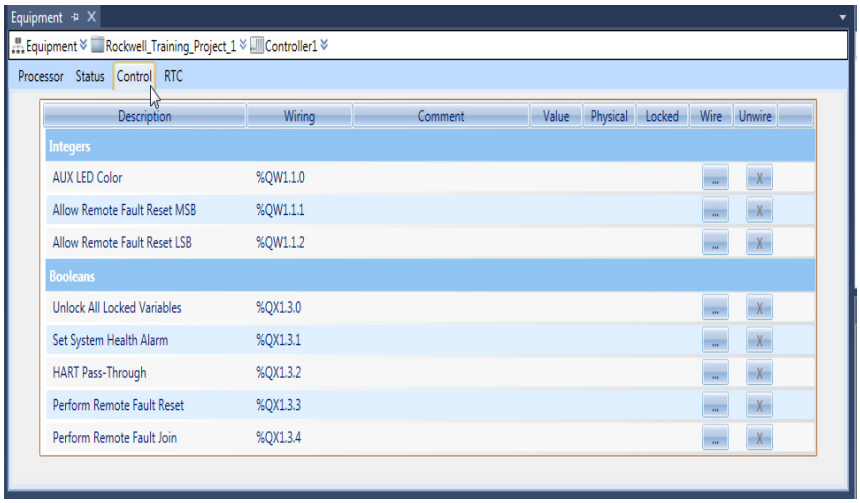
Values:

- TRUE = The backup battery in the 9110 processor module in the given slot is present and its voltage is within satisfactory limits.
- FALSE = The voltage of the backup battery is low or the battery is not there.

Description:
Reports the health status of the backup battery in a processor module. The battery voltage is checked at start up, then tested again every 24 hours (elapsed time).

Control Variables

Control variables include Control Integers and Control Booleans.



Control Integers

The Control Integers let the application send instructions to the controller.

AUX LED Colour

Direction: output from application to controller

Type: word

Values:

- 0..3 (0 = off, 1 = red, 2 = green, 3 = amber)
- Default 0

Description:
Sets the state of the LED indicator labeled 'Aux' on all 9110 processor modules.

Allow Remote Fault Reset Most Significant Bits (MSB)

Direction: output from the application to the controller

Type: word

Values: Possible values range from 0 to 65535

Description:
Used in combination with Allow Remote Fault Reset LSB, enables remotely resetting a processor or enables joining a processor to a running system. For a successful remote fault reset or join, the value entered for the Remote Fault Reset must match the combined values set for the control integers Allow Remote Fault Reset MSB and Allow Remote Fault Reset LSB. Also, the control booleans Perform Remote Fault Reset and Perform Remote Fault Join must be set to TRUE.

Allow Remote Fault Reset Least Significant Bits (LSB)

Direction: output from the application to the controller

Type: word

Values: Possible values range from 0 to 65535

Description:
Used in combination with Allow Remote Fault Reset MSB, enables remotely resetting a processor or enables joining a processor to a running system. For a successful remote fault reset or join, the value entered for the Remote Fault Reset must match the combined values set for the Control Integers Allow Remote Fault Reset MSB and Allow Remote Fault Reset LSB. Also, the Control Booleans Perform Remote Fault Reset and Perform Remote Fault Join must be set to TRUE.

Control Booleans

The Control Booleans let the application send instructions to the controller.

Unlock All Locked Variables

Direction: output from application to controller

Type: Boolean

Values:

- TRUE = Remove all locks.
- FALSE = No effect.

The default value is FALSE

Description:
Removes all user locks on input and output variables.

Set System Health Alarm

Direction: output from application to controller.

Type: Boolean

Values:

- TRUE = When the variable transitions from FALSE to TRUE, the system responds as if it had found a system level fault. The processor module System Healthy LED is set to RED and the System Health Boolean is set TRUE. When the variable is TRUE, the fault is immediately re-annunciated after pressing the **Fault Reset** button.
- FALSE = Does not send an alarm signal to the controller. After the variable transitions from TRUE to FALSE, the processor module System Healthy LED passes to GREEN after pressing the **Fault Reset** button.

The default value is FALSE

Description:
Sends a System Health alarm signal from the application to the controller.

HART Pass-Through

Direction: output from application to controller

Type: Boolean

Values:

- TRUE = HART Pass-Through is enabled and available for an Analog Module.
- FALSE = HART Pass-Through is disabled and not available.

The default value is FALSE

Description:
Starts the HART Pass-Through feature and allows HART messages on analog input and output modules. The system allows messages on each channel independently and together.

Perform Remote Fault Reset

Direction: output from the application to the controller

Type: Boolean

Values:

- TRUE - Resets the faulty processors
- FALSE - Remotely resetting processors is disabled

The default value is FALSE

Description:
When a rising edge is detected, resets the faulty processors

Perform Remote Fault Join

Direction: output from the application to the controller

Type: Boolean

Values:

- TRUE - The processors join the running system
- FALSE - Remotely joining processors to a running system is disabled.

The default value is FALSE

Description:
When a rising edge is detected, the processors join the running system

Real-Time Clock

The AADvance controller contains a real-time clock (RTC) used to make a record of the time and date of system events, SOE services, and log files. You can manually set the real-time clock of the processor by using RTC variables or use SNTP from the Communication View.

IMPORTANT You are strongly recommended to use SNTP to synchronize controllers so that all time-related events, services, and logs use the same time of day.

RTC Processor Variables

RTC processor variables manage information regarding the real-time clock. The **AADvance** controller has the following RTC processor variables:

- **RTC Status Variables:** Year, Month, Day of Month, Hours, Minutes, Seconds, Milliseconds
- **RTC Program Variables:** Year, Month, Day of Month, Hours, Minutes, Seconds, Milliseconds
- **RTC Control Variables:** RTC Write, RTC Read, Year, Month, Day of Month, Hours, Minutes, Seconds, Milliseconds

RTC Status Variables

The RTC Status Variables supply data about the controller real-time clock to the application.

RTC Status: Year

Direction: input to application from controller

Type: word

Values:

- 2,000 to 2,399, or 0 (see description)

Description:
Reports the oldest value of real-time clock (RTC) Year as voted by every 9110 processor module which is present and synchronized. Only updated if the real-time clock control Boolean RTC Read is set to TRUE. If RTC Read is FALSE, the value will be 0 (zero).

RTC Status: Month

Direction: input to application from controller

Type: word

Values:

- 1 to 12, or 0 (see description)

Description:
Reports the oldest value of real-time clock (RTC) month as voted by every 9110 processor module which is present and synchronized. Only updated if the real-time clock control Boolean RTC Read is set to TRUE. If RTC Read is FALSE, the value will be 0 (zero).

RTC Status: Day of Month

Direction: input to application from controller

Type: word

Values:

- 1 to 31, or 0 (see description)

Description:
Reports the oldest value of real-time clock (RTC) day of month as voted by every 9110 processor module which is present and synchronized. Only updated if the real-time clock control Boolean RTC Read is set to TRUE. If RTC Read is FALSE, the value will be 0 (zero).

RTC Status: Hours

Direction: input to application from controller

Type: word

Values:

- 0 to 23

Description:
Reports the oldest value of real-time clock (RTC) hours as voted by every 9110 processor module which is present and synchronized. Only updated if the real-time clock control Boolean RTC Read is set to TRUE. If RTC Read is FALSE, the value will be 0 (zero).

RTC Status: Minutes

Direction: input to application from controller

Type: word

Values:

- 0 to 59

Description:
Reports the oldest value of real-time clock (RTC) minutes as voted by every 9110

processor module which is present and synchronized. Only updated if the real-time clock control Boolean RTC Read is set to TRUE. If RTC Read is FALSE, the value will be 0 (zero).

RTC Status: Seconds

Direction: input to application from controller

Type: word

Values:

- 0 to 59

Description:
Reports the oldest value of real-time clock (RTC) seconds as voted by every 9110 processor module which is present and synchronized. Only updated if the real-time clock control Boolean RTC Read is set to TRUE. If RTC Read is FALSE, the value will be 0 (zero).

RTC Status: Milliseconds

Direction: input to application from controller

Type: word

Values:

- 0 to 999

Description:
Reports the oldest value of real-time clock (RTC) milliseconds as voted by every 9110 processor module which is present and synchronized. Only updated if the real-time clock control Boolean RTC Read is set to TRUE. If RTC Read is FALSE, the value will be 0 (zero).

RTC Program Variables

The RTC program variables contain the parts of the date to write to the real-time clock when the RTC control variable RTC Write is asserted TRUE. The values are written only if the RTC control variable Year is TRUE.

RTC Program: Year

Direction: output from application to controller

Type: word

Values:

- 2,000 to 2,399
- Default 0 (zero)

Description:
Specifies the Year part of the date to write to the real-time clock the next time the RTC control variable RTC Write is asserted TRUE. The value is written only if the RTC control variable Year is TRUE.

RTC Program: Month

Direction: output from application to controller

Type: word

Values:

- 1 to 12
- Default 0 (zero)

Description:
Specifies the number of the month part of the date to write to the real-time clock the next time the RTC control variable RTC Write is asserted TRUE. The value is written only if the RTC control variable Month is TRUE.

RTC Program: Day of Month

Direction: output from application to controller

Type: word

Values:

- 1 to 31
- Default 0 (zero)

Description:
Specifies the day of month part of the date to write to the real-time clock the next time the RTC control variable RTC Write is asserted TRUE. The value is written only if the RTC control variable Day of Month is TRUE.

RTC Program: Hours

Direction: output from application to controller

Type: word

Values:

- 0 to 23
- Default 0 (zero)

Description:
Specifies the time of day (in hours) part of the date to write to the real-time

clock the next time the RTC control variable RTC Write is asserted TRUE. The value is written only if the RTC control variable Hours is TRUE.

RTC Program: Minutes

Direction: output from application to controller

Type: word

Values:

- 0 to 59
- Default 0 (zero)

Description:
Specifies the minute in the hour part of the date to write to the real-time clock the next time the RTC control variable RTC Write is asserted TRUE. The value is written only if the RTC control variable Minutes is TRUE.

RTC Program: Seconds

Direction: output from application to controller

Type: word

Values:

- 0 to 59
- Default 0 (zero)

Description:
Specifies the second in the minute part of the date to write to the real-time clock the next time the RTC control variable RTC Write is asserted TRUE. The value is written only if the RTC control variable Seconds is TRUE.

RTC Program: Milliseconds

Direction: output from application to controller

Type: word

Values:

- 0 to 999
- Default 0 (zero)

Description:
Specifies the milliseconds in the second part of the date to write to the real-time clock the next time the RTC control variable RTC Write is asserted TRUE. The value is written only if the RTC control variable Milliseconds is TRUE.

RTC Control Variables

The RTC control variables permit updates to the parts of the date in real-time clock.

RTC Control: RTC Write

Direction: output from application to controller

Type: Boolean

Values:

- TRUE = Applies new values to real-time clock (see description).
- FALSE = No effect.

The default value is FALSE.

Description:

Sets new values for the real-time clock. There are seven values, all specified by the RTC program control words Year, Month, Day of Month, Hours, Minutes, Seconds and Milliseconds. Each value is set only if its related RTC control variable (which is a Boolean, and similarly named Year, Month, Day of Month, Hours, Minutes, Seconds or Milliseconds) is TRUE.

The change is started by the transition of the variable from FALSE to TRUE. If the application holds the TRUE state until the end of the application cycle (or longer), the application makes the change at the end of the cycle. There is no time limit on returning the value from TRUE to FALSE after the clock is updated. If the application returns the variable to FALSE before the end of the application cycle, the clock is not updated.

Example

Consider this scenario:

- The date is 28th October 2008, 8 hours, 12 minutes and 35 seconds
- RTC Control RTC Read is TRUE
- RTC Control Year, Month and Day of Month are TRUE
- RTC Control Hours, Minutes and Seconds are TRUE.

The RTC Status Variables is returned, and the real-time clock is set as follows:

- Year = 2008
- Month = 10
- Day of Month = 28
- Hours = 8
- Minutes = 12
- Seconds = 35.

RTC Control: RTC Read

Direction: output from application to controller

Type: Boolean

Values:

- TRUE = The controller updates RTC status values on each application cycle.
- FALSE = RTC status values are static (do not update).

The default value is FALSE

Description:

Checks if the RTC Status Variables (RTC Status: Year, RTC Status: Month, RTC Status: Day of Month, RTC Status: Hours, RTC Status: Minutes and RTC Status: Seconds) updates in real-time.

All the RTC Status Variables must be set to TRUE when the RTC Read variable is set to TRUE, otherwise the RTC value is not updated and reported.

RTC Control: Year

Direction: output from application to controller

Type: Boolean

Values:

- TRUE = RTC program Year is applied by RTC Write
- FALSE = RTC program Year is ignored

The default is FALSE until an initial value is specified in the application.

Description:

Checks if the value of the RTC program variable named Year must be applied to the real-time clock the next time the RTC control variable named RTC Write is set to TRUE.

The RTC program variable is only updated if the RTC control variable RTC Read is set to TRUE and all other RTC Control variables are set to TRUE.

RTC Control: Month

Direction: output from application to controller

Type: Boolean

Values:

- TRUE = RTC program month is applied by RTC Write.
- FALSE = RTC program month is ignored.

The default value is FALSE until an initial value is specified in the application.

Description:

Checks if the value of the RTC program variable named Month must be applied to the real-time clock the next time the RTC control variable named RTC Write is set to TRUE.

The RTC program variable is only updated if the RTC control variable RTC Read is set to TRUE and all other RTC Control variables are set to TRUE.

RTC Control: Day of Month

Direction: output from application to controller

Type: Boolean

Values:

- TRUE = RTC program day of month is applied by RTC Write.
- FALSE = RTC program day of month is ignored.

The default value is FALSE until an initial value is specified in the application.

Description:

Checks if the value of the RTC program variable named Day of Month must be applied to the real-time clock the next time the RTC control variable named RTC Write is set to TRUE.

The RTC program variable is only updated if the RTC control variable RTC Read is set to TRUE and all other RTC Control variables are set to TRUE.

RTC Control: Hours

Direction: output from application to controller

Type: Boolean

Values:

- TRUE = RTC program hours is applied by RTC Write
- FALSE = RTC program hours is ignored

The default value is FALSE until an initial value is specified in the application.

Description:

Checks if the value of the RTC program variable named Hours must be applied to the real-time clock the next time the RTC control variable named RTC Write is set to TRUE.

The RTC program variable is only updated if the RTC control variable RTC Read is set to TRUE and all other RTC Control variables are set to TRUE.

RTC Control: Minutes

Direction: output from application to controller

Type: Boolean

Values:

- TRUE = RTC program minutes is applied by RTC Write.
- FALSE = RTC program minutes is ignored.

The default value is FALSE until an initial value is specified in the application.

Description:

Checks if the value of the RTC program variable named Minutes must be applied to the real-time clock the next time the RTC control variable named RTC Write is set to TRUE.

The RTC program variable is only updated if the RTC control variable RTC Read is set to TRUE and all other RTC Control variables are set to TRUE.

RTC Control: Seconds

Direction: output from application to controller

Type: Boolean

Values:

- TRUE = RTC program seconds is applied by RTC Write
- FALSE = RTC program seconds is ignored

The default value is FALSE until an initial value is specified in the application.

Description:

Checks if the value of the RTC program variable named Seconds must be applied to the real-time clock the next time the RTC control variable named RTC Write is set to TRUE.

The RTC program variable is only updated if the RTC control variable RTC Read is set to TRUE and all other RTC Control variables are set to TRUE.

RTC Control: Milliseconds

Direction: output from application to controller

Type: Boolean

Values:

- TRUE = RTC program milliseconds is applied by RTC Write

- FALSE = RTC program milliseconds is ignored

The default value is FALSE until an initial value is specified in the application.

Description:

Checks if the value of the RTC program variable named Milliseconds must be applied to the real-time clock the next time the RTC control variable named RTC Write is set to TRUE.

The RTC program variable is only updated if the RTC control variable RTC Read is set to TRUE and all other RTC Control variables are set to TRUE.

To manually set the real-time clock

You manually set the real-time clock of the processor by using RTC variables.

1. From the Build menu, click **Build All**.
2. From the Project menu, either click **Download** or **Update**.
3. From the Online menu, click **Connect**.
4. Force the following **RTC Control Variables** to TRUE:
   RTC Read, Year, Month, Day of Month, Hours, Minutes, Seconds, and Milliseconds.

IMPORTANT Do not force RTC Write at this point, the value should be FALSE.

   The RTC Status Variables now display the current date and time in the processor.
5. Double-click each **RTC Program Variable** (Year, Month, Day of Month, Hours, Minutes, Seconds, and Milliseconds), specify a value, and then click **Write**.
6. To enable writing the new date and time to the processor, set RTC Write to TRUE and then FALSE.

The RTC Status Variables now display the set date and time in the processor.

Configuring Timing and Communications

This chapter describes the procedure to configure the AADvance® controller time settings.

Synchronizing the Controller Time

It is important that all of the controllers (and computers) producing log files in a system use the same time of day. This makes it much easier to use the Sequence of Events (SOE) service for Boolean variables also log-based troubleshooting and other investigations are also more dependable too.

The AADvance controller helps you to use the same time of day setting for all the equipment in a system by supporting the Simple Network Time Protocol (SNTP). The NTP protocol allows time to be communicated around networks. The simplified protocol uses Network Time Protocol (NTP) and is better suited to embedded devices. Usually, the controller is configured as an SNTP client. If there is no dedicated time server available the controller may be used as a (S)NTP server. You configure the controllers as SNTP clients to get their processor time data from a NTP server.

When the controller is configured as an SNTP client, the controller automatically gets time data and updates its time settings from a server using the NTP. There are many NTP time servers available on the Internet. However, the use of the Internet is not recommended for industrial applications because the network transmission delays are so unpredictable. For security reasons, industrial networks should not normally have access to the internet. Also public servers will only expect infrequent demands for time. The AADvance controller will poll at a relatively high rate. Further, a local time server makes broadcasting available, this being the most efficient way to disseminate time to many time clients. A local server provides a more predictable output and is easier to manage. You can use a local, hardware time server or a software-based server running on a computer.

Choosing a Network Time Server

There are three ways to provide a network time server:

- a hardware time server
- a time server program running on a computer
- a processor module in a suitably configured AADvance controller.

A hardware time server retrieves its time of day from multiple sources such as global positioning system (GPS) satellites, radio-based national and

international time standards such as DCF 77 (Germany) and MSF60 (England), and its own internal clock. This is the most adaptable server.

Alternatively you can use a software implementation to make a computer based NTP server. The program enables controlling and configuring the NTP service for Windows. Other features depend on the design of the software and can include switching between configurations, changing service settings, extracting related log entries, and displaying the current status of local and external NTP services.

If you use an AADvance controller as the SNTP time server, the controller will use the real-time clock in one of its 9110 processor modules as its source of time data. You will use the AADvance Workbench to configure the processor module. The default value for a new processor module includes the date, '1 January 1900'.

NOTE In a controller you may configure any of the processor modules as time servers. It is important not to configure processor modules to be time clients of any of their peer modules (or itself). The modules of a controller automatically agree the time amongst themselves.

Using the Controller as an SNTP Client

As an SNTP client the AADvance controller accepts the current time from an external NTP or SNTP network time server. The configured settings provide the following information to the controller:

- the IP address of the server
- the version of SNTP offered by the server and
- the mode by which the processor will acquire the time.

The controller supports the broadcast and the unicast modes of SNTP.

Using the Controller as an SNTP Server

As an SNTP server, the AADvance controller supplies a time signal for other equipment connected to the network. The configured settings inform the server of its connected networks to use.

You can configure a 9110 processor module as a broadcast server or a unicast server:

- If you set unicast mode for a processor module, the server waits until it is polled by a client and then sends a time signal.
- If you set broadcast mode the controller regularly transmits a time signal and also responds to unicast requests from clients. This is the most efficient mode if several devices will rely on the server for their source of time.

Using the Controller as an SNTP Client and Server

In a system having many AADvance controllers, setting up one of the controllers as a client and a server can be helpful. This controller becomes a client of the network time server (a hardware or software-based server) and distributes the time data to each of the other controllers, which are configured as clients. This arrangement can reduce the load on the server if clients are unicast clients (there is no incremental load on the server if all the clients are broadcast clients).

Time Synchronization SNTP

The AADvance controller can be configured to operate as an SNTP client or server or both.

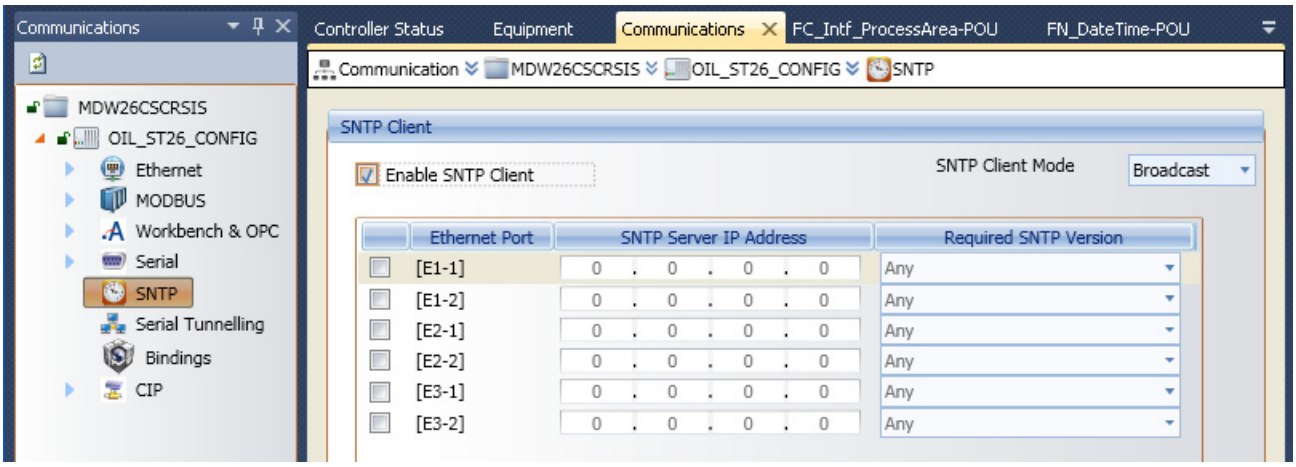
- The SNTP client settings inform the controller of the following information: the IP address of the SNTP server the version of SNTP offered by the server and the operating mode for the time synchronization signal that the processors will use for their real-time clock. The processor module can be configured as a unicast or broadcast client.
- The AADvance controller can also fulfill the role of an SNTP server. To enable serving of time on an interface, you need to enable the interface and then you need to specify the direct broadcast address for that interface. This works for broadcast or unicast modes. When the controller is configured as a broadcast server, the controller can still respond to unicast requests from clients.
- Configure the controller as both a client and a server if using an external time server and you want to use the controller to supply the time data to other controllers and devices.

IMPORTANT Changes to the SNTP settings are not active until after the power is cycled.

Configure the Controller as an SNTP Client

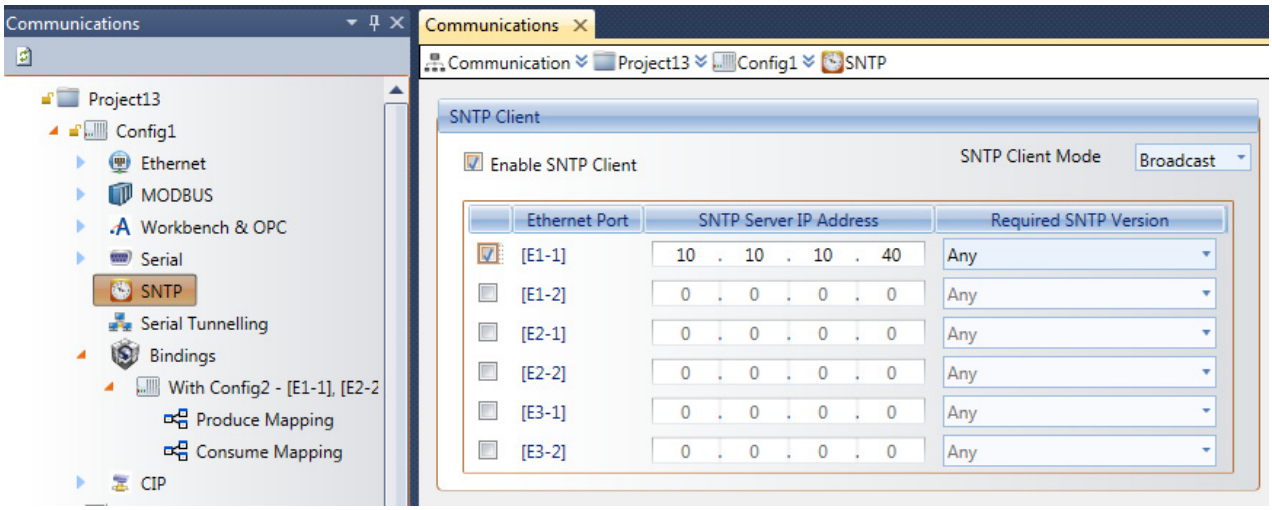
The AADvance controller can get its real-time clock time data from a primary network time server and, if applicable, a client of a secondary network time server. A controller will prefer any primary time server over any secondary server. Secondary servers will only be used once no primary is visible. In the case where more than one primary (or secondary if these have been resorted to) server is available, the lowest lettered module will be preferred (e.g., A will be preferred to B).

To configure the controller as an SNTP client, do the following:



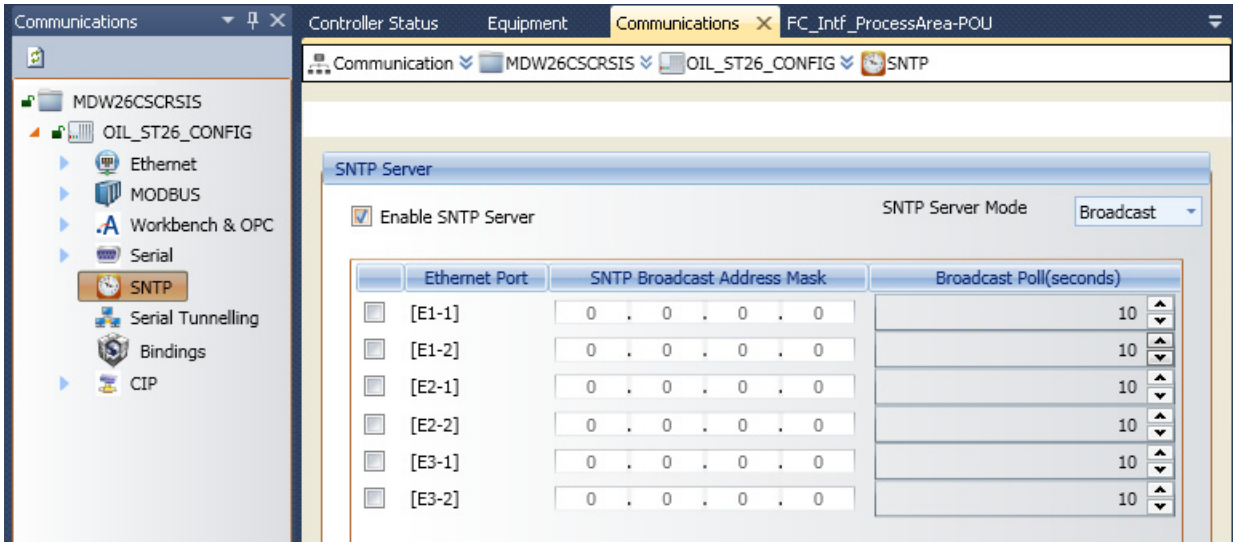
1. Go to the Communications view.
2. Select and expand the configuration, then select the **SNTP** node.
3. Select the **Communications** tab in the content window.
4. Select **Enable SNTP Client**.
5. Set the SNTP Client Mode to Unicast or Broadcast option as applicable.
   - In Broadcast mode the SNTP client waits for regular broadcasts from the server. This mode reduces network traffic and hence the load on the servers.
   - In Unicast mode the SNTP client actively polls all of the servers which are configured and uses their responses. The polling rate is fixed at 19 seconds.
6. The window lists six items, labelled E1-1 to E3-2.
   - If only one processor module is fitted in the left-most position, use item E1-1.
   - Otherwise, select an item related to a processor module and set the SNTP Server IP Address field to the IP addresses of the primary network time server.
7. Use the Required SNTP Version field to select the version of the time protocol offered by the server.
   - Select SNTPv1, SNTPv2, SNTPv3, SNTPv4 or Any.
   - If you do not know the version of NTP/SNTP that the server offers is unknown, use Any. This option disables some validation of the incoming signal.
   - If fault-tolerant operation is not necessary for the controller, the configuration of SNTP is now completed.
8. If the controller requires fault-tolerant operation, select a second item.
   - If the controller has two or three processor modules, use an item related to one of these processor modules.
   - Configure the details for the secondary network time server.

The following is an example of a module that has been configured as a broadcast client of the server 10.10.10.40. Validation of the server has been disabled.



Configure the Controller as an SNTP Server

The AADvance controller can supply a time signal for other equipment connected to the network. To configure the controller as an SNTP server do the following:

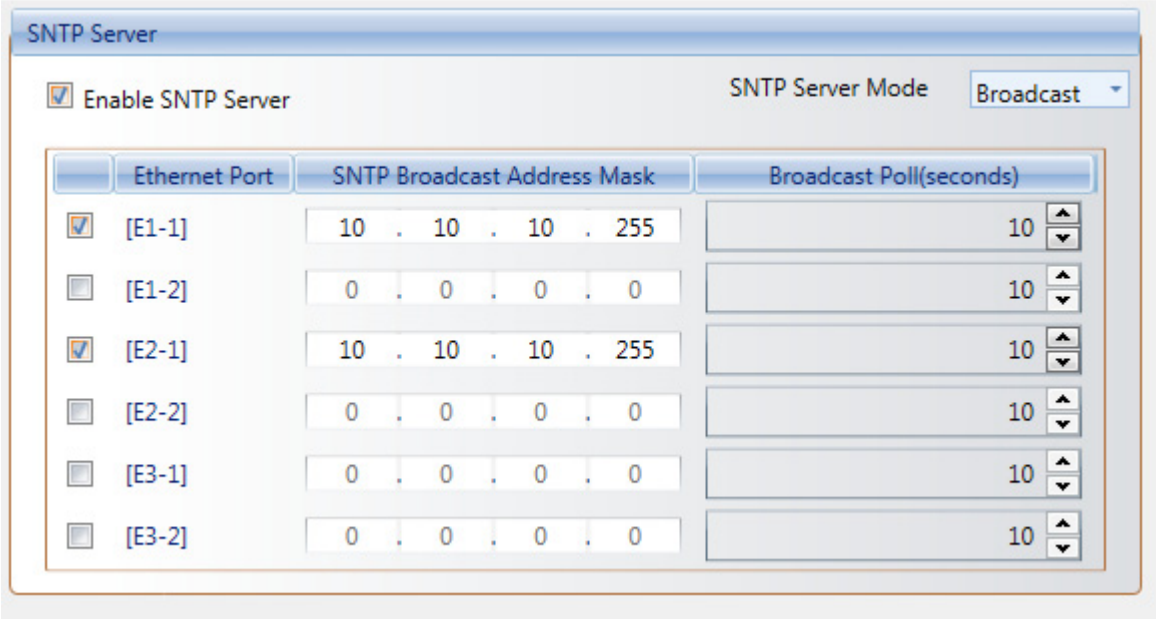


1. Go to the communications view.
2. Select and expand the configuration, then select the **SNTP** node.
3. Select the **Communications** tab in the content window.
4. Select **Enable SNTP Server**.
5. Set the SNTP Server Mode to Unicast or Broadcast as applicable.
   - In Broadcast mode the SNTP server makes regular broadcasts to other equipment on the network. The server also responds to unicast polling requests.
   - In Unicast mode the SNTP server waits to be polled by a client and then responds with a time signal. The server will not broadcast time signals.

6. The window lists six items, labelled E1-1 to E3-2. Select an item related to one processor module and set the SNTP Broadcast Address Mask field to the broadcast IP addresses for the network. If a processor module is fitted in the left-most position, use item E1-1.
   - Set the address to zero (0.0.0.0) to disable the server on that interface.
7. If the system requires fault-tolerant operation you can configure a second server on the other interface.
   - If the controller has two or three processor modules, use an item related to one of these processor modules.
8. Configure the details for the second network time server.

Example Server Configuration

The following is an example of a Module A that has been configured to broadcast time on its first interface while module B has also been configured to broadcast time on its first interface.



Using Serial Communications

Serial Port Protocols

The serial ports support the protocols listed in the table.

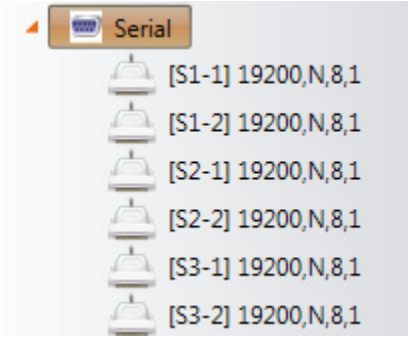
Table 1 - Serial Port Protocols

Type	Description
RS485fd	Full-duplex, 4-wire connection with a different bus for transmit and receive
RS485fdmux	Full-duplex, 4-wire connection with a different bus for transmit and receive and tri-state outputs on the transmit connections
RS485hdmux	Half duplex, 2-wire connection

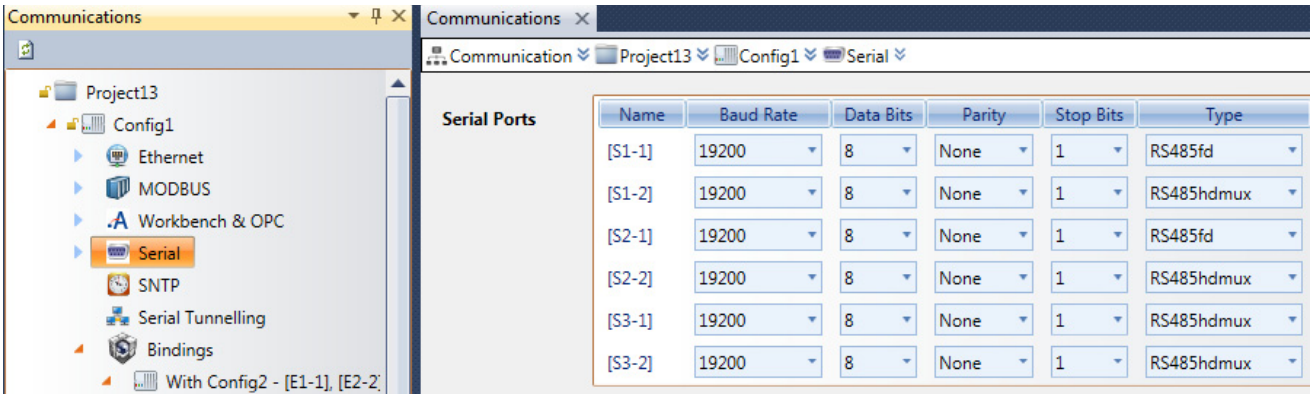
Configure the Serial Ports

The AADvance controller has a maximum of six serial communication ports, two for each 9110 processor module.

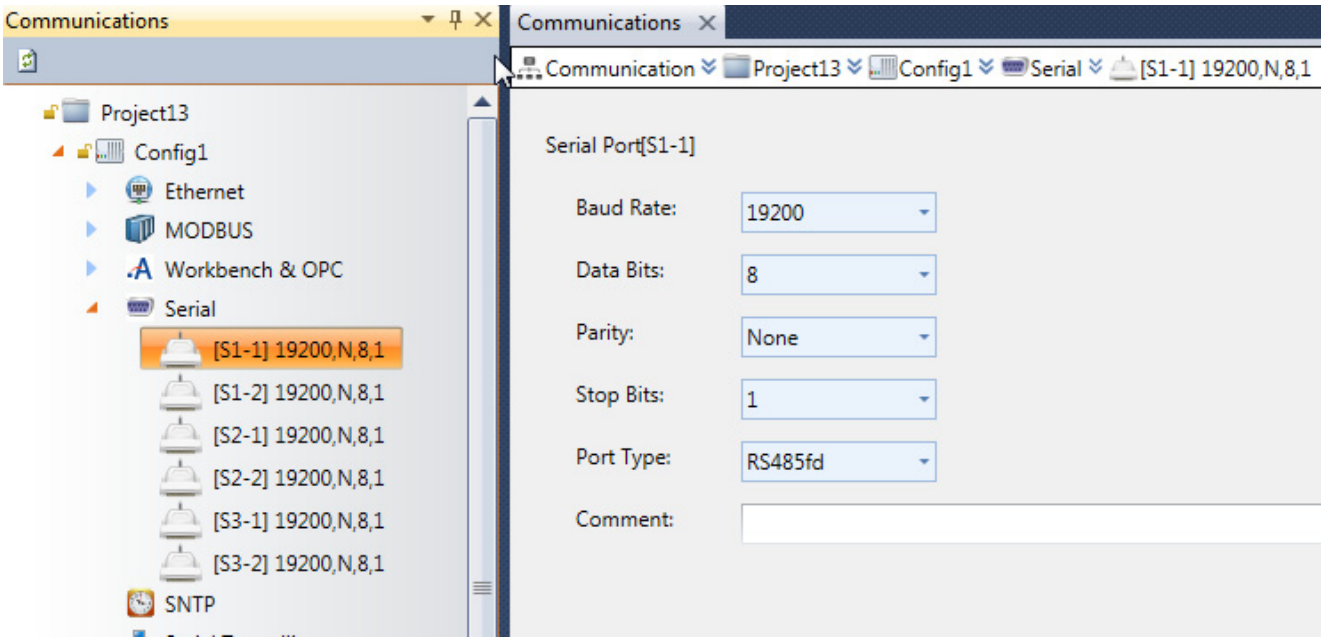
The serial port settings indicate the type of protocol and the data characteristics for each of the serial ports. To configure the serial ports do the following:



1. Select the **Serial** node.
   - The **Serial Ports** editor dialog box is displayed.



2. You can set the parameters for all the serial ports in one go or enter the parameters for each port individually.



Serial Port Parameters

Each serial port on the AADvance controller supports the set of control parameters as shown in the following table.

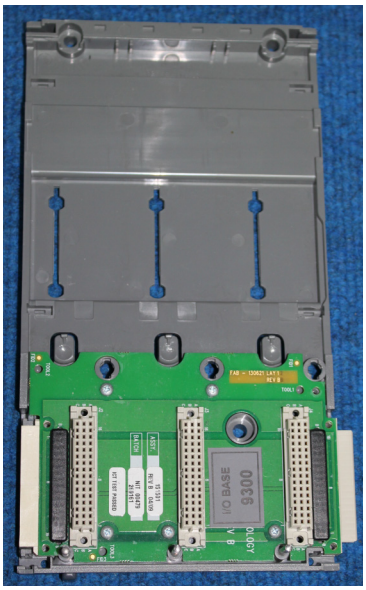
Table 2 - Controller Serial Port Parameters

Description	Value(s)	Default	Remarks
Baud	1,200; 2,400; 4,800; 9,600; 19,200; 38,400; 57,600; 76,800 or 115,200	19,200	
Data Bits	5 to 8	8	
Parity	None, Odd or Even	None	
Stop Bits	1 or 2	1	
Type	RS485fd RS485fdmux RS485hdmux	RS485hdmux	'fd' means 'full duplex' 'hd' means 'half duplex'

Most systems use two bits after each data byte. The two bits are either a parity bit (odd or even) and one stop bit, or no parity and two stop bits.

Configuring the Controller I/O

This chapter explains how to configure the controller I/O to match the hardware configuration and to meet your I/O requirements. The modules are mounted on I/O Base Units with Termination Assemblies. An example of the standard I/O Base Unit which accommodates three bus Termination Assemblies is shown below.



The input modules may be configured as simplex, duplex for dual redundancy or triplex for triple redundancy. The output modules may be configured as simplex or as duplex for dual redundancy. For the Input / Output (I/O) systems the type used must match the Termination Assembly. The certified revision list for the types of modules and Termination Assemblies is included in the following table:

	Module		Termination Assemblies		
	Module Code	**Number of Chanels**	**Simplex Code**	**Duplex Code**	**Triplex Code**
Digital Input	9401 9402	8 16	9801	9802	9803
Analogue Input	9431 9432	8 16	9831	9832	9833
Digital Output	9451	-	9851	9852	-
Analogue Output	9481 9482	3 8	9881	9882	-

For the code numbers in the body of the table the following applies:

1. All numbers are four digits long.
2. First digit (9) identifies series type as 9000 Series, which is AADvance®.

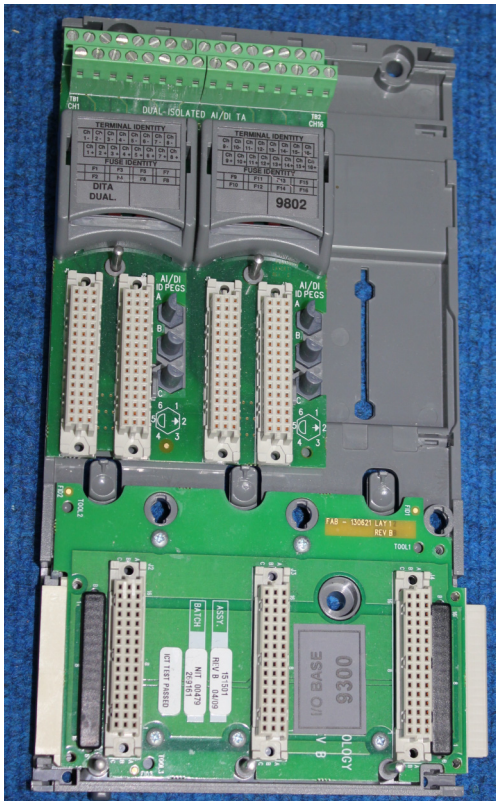
3. Second digit identifies part type, where 4 = Module and 8 = Termination Assembly.
4. Third digit is 0, 3, 5 or 8; where:
   - 0 = Digital Input
   - 3 = Analogue Input
   - 5 = Digital Output
   - 8 = Analogue Output.
5. Fourth digit varies for Modules and Termination Assemblies.
   a. For Modules, the numbers 1 and 2 specify the number of channels as follows:
   - Digital Input; 1 = 8 and 2 = 16
   - Analogue Input; 1 = 8 and 2 = 16
   - Digital Output; no distinction (only 1 is used)
   - Analogue output; 1 = 3and 2 = 8.

   b. For Termination Assemblies:
   - 1 = Simplex
   - 2 = Duplex
   - 3 = Triplex.

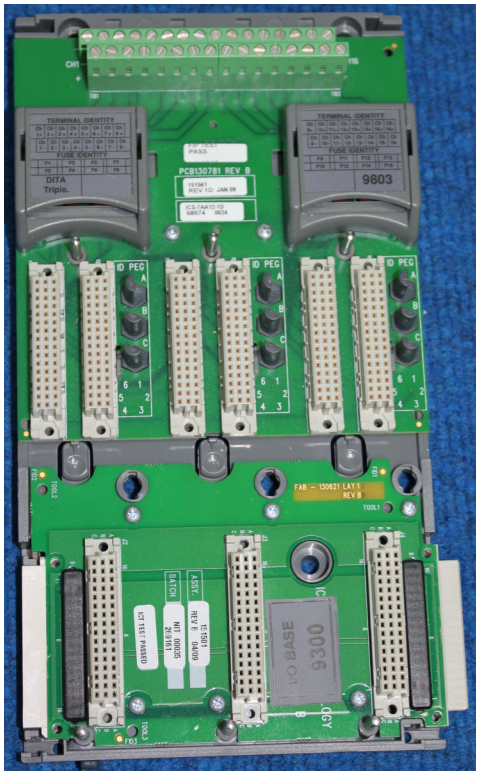
Examples of the Termination Assemblies are as follows:

Simplex

Duplex



Triplex



About Adding and Configuring I/O Modules

You use the Equipment view to add I/O modules to a configuration to copy the hardware arrangement of the modules. You can configure the modules in a

simplex arrangement or in a redundant group of two or three. It is recommended that you configure your system when building the system.

In most cases, you configure the modules to meet your system requirements. However, you can anticipate future additions to the I/O hardware by configuring groups for simplex modules or triple groups for dual arrangements. You can also create a "Hot Swap" capability if required.

To add and configure I/O modules:

1. Assign I/O modules to the IO Bus 1 or IO Bus 2 slots for the specific controller.
2. Choose the appropriate termination assembly for the arrangement.
3. Set the process safety time for the I/O modules.
4. Configure the I/O module Status Variables.
5. Configure the I/O module Channel Variables.

NOTE The procedure uses variables you have already declared using the application view. If you create a new variable during the procedure, the Workbench prompts you to store it with the other variables.

The system enables you to perform the following tasks:

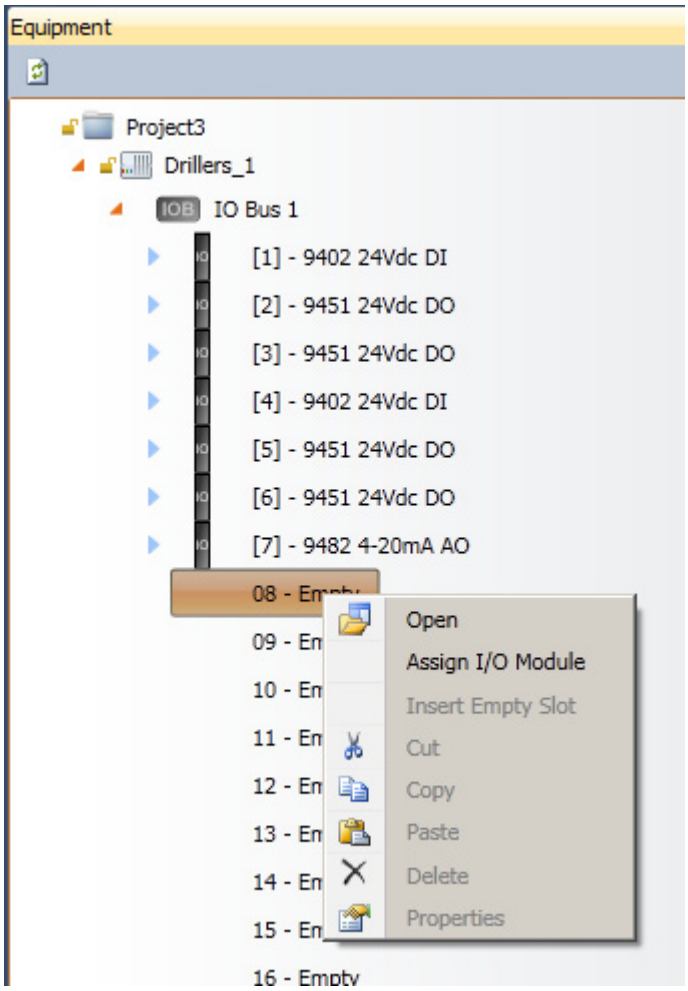
- assign an I/O module to an empty slot
- create a group arrangement
- move a module to a different slot
- remove a module
- reduce a group arrangement from a triple to a dual or a dual to a single
- insert an empty slot.

Assign I/O Modules to a Configuration

To assign an I/O module:

1. Select the configuration in the Equipment View.
2. Expand the IO Bus 1 of the controller you are configuring.
3. At an empty slot right-click and select **Assign I/O Module**.
4. On the Assign I/O Module window, select the required module and then select the appropriate type of termination assembly to suit the arrangement and the module. Click **OK**.

NOTE If you are assigning a simplex module you can put the module in any empty slot. If you are creating a redundant group you must locate two or three empty slots adjacent to each other.

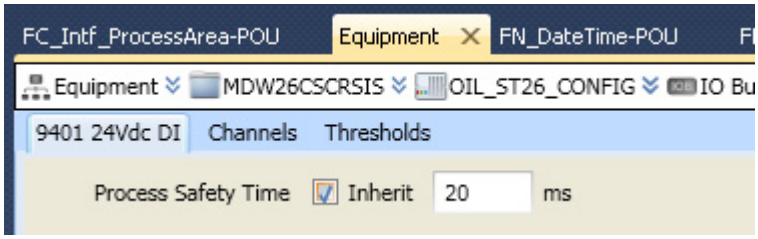


NOTE When you choose a dual or triple termination assembly the system automatically assigns two or three modules to the adjacent slots.

Configure the I/O Module Process Safety Time

The process safety time for an I/O module defines the maximum time (in milliseconds) that the controller allows its outputs to remain in the on state after the controller detects important diagnostic faults or application faults related to the I/O module. If the PST expires, the system will go to its safe condition.

You can use the top-level process safety time set for the controller or specify a value for the I/O module. The process safety time setting can be the same as the top-level setting or less, but it must not be more. To set the process safety time for an I/O module, do the following:

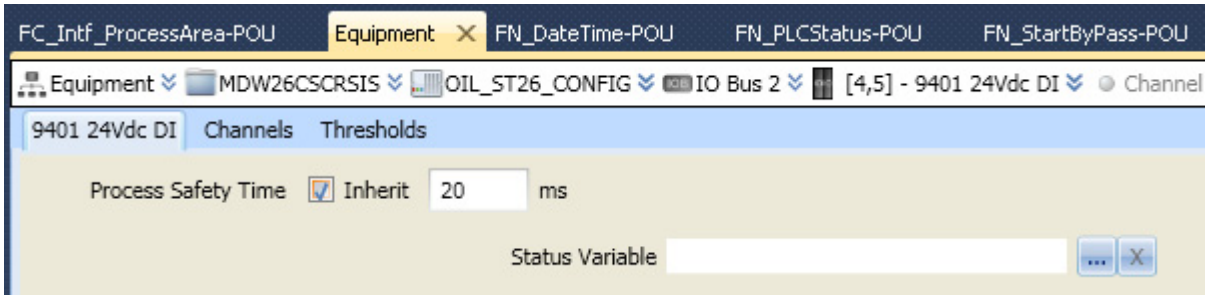


1. Select the I/O module in the Equipment View.
2. Go to the **Equipment** tab on the content window and select the tab with the name of the module ('9401 24Vdc DI' in the illustration).
3. Clear the Inherit field and indicate the time value into the Process Safety Time field. Choose from the following range of values:
   - Minimum: 20 ms
   - Maximum: 60,000 ms (60 seconds)
   - Default: 2,500 ms (2.5 seconds).

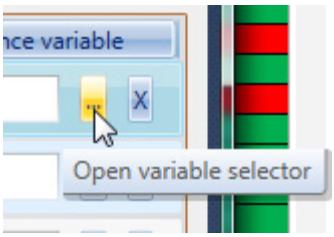
Wire a Status Variable to an I/O Module

You can use a Status Variable to get data such as health, ready, and run conditions from an I/O module. The AADvance Workbench provides a structure (T9K_TA_GROUP_STATUS) for the status data.

To wire a Status Variable to an I/O module, do the following:

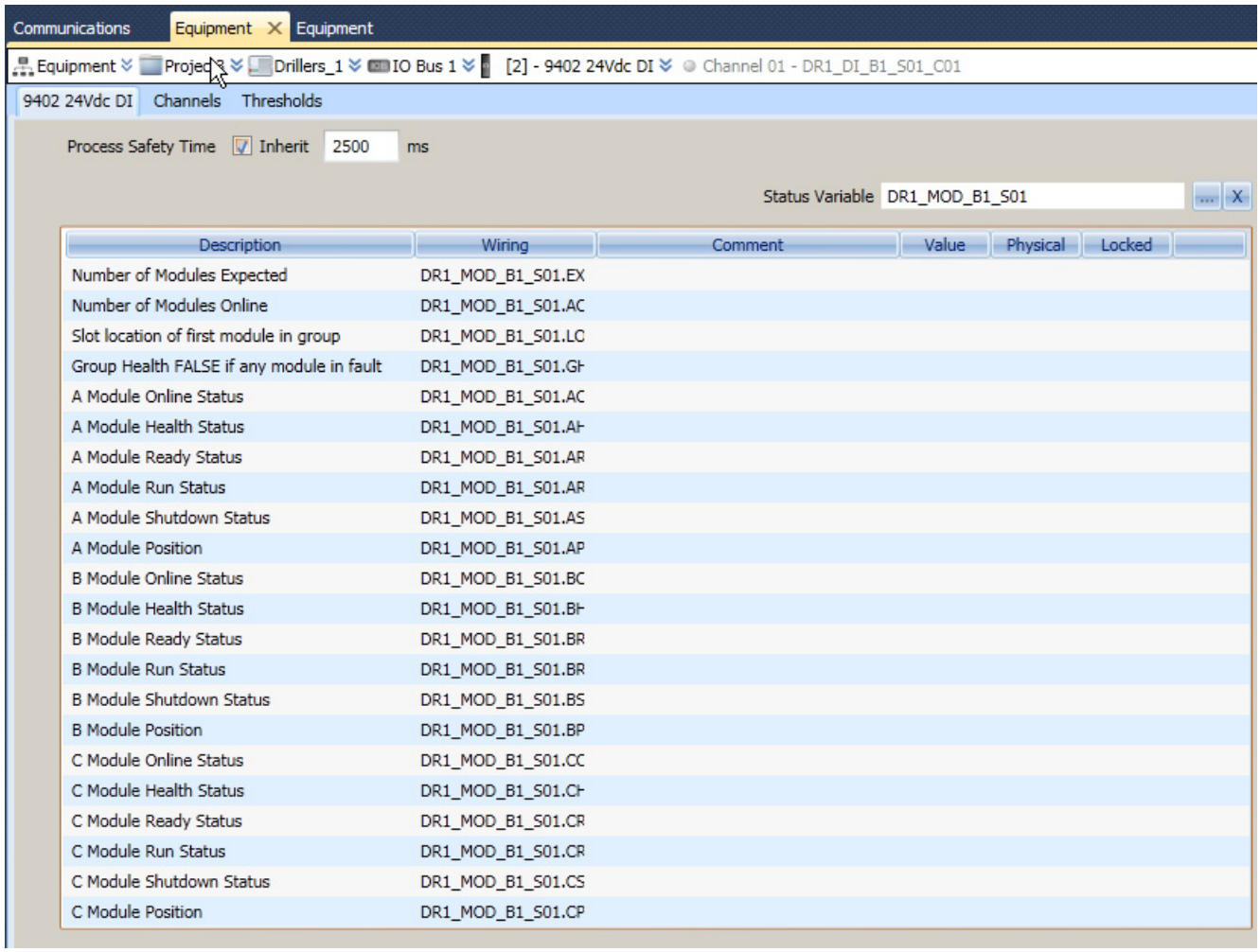


1. Declare a variable. Use the type T9K_TA_GROUP_STATUS and make sure that the direction is set to input.
2. Go to the Equipment view and select the I/O module on the I/O bus.
3. Go to the **Equipment** tab on the content window and select the tab with the name of the module ('9401 24Vdc DI' in the illustration).
4. Go to the Status Variable field and click the button adjacent to it.



   - The Variable Selector window is displayed.
5. Select the variable from the list, then click **OK**.
6. Repeat this procedure for other I/O modules.

A Typical Status Variable



T9K_TA_GROUP_STATUS (I/O Module Status Data)

The data structure for Module Status Data (T9K_TA_GROUP_STATUS) has the elements given in the table.

The controller examines an I/O module by referring to the physical arrangement of the module and its location in a group:

- A simplex module is called module A; a duplex module is module A or B and a triplicated module is module A, B or C.
- The structure always includes the status fields for modules A, B and C, even if the configuration has a simplex or dual termination assembly.
- The module is shown as 'X' in the table.

Table 3 - Structure for I/O Module Status Data

Identifier	Type	Description	Remarks
<tagname>.EXPC	INT	Modules expected	Retrieves the quantity of modules that are defined in the configuration for the group (1, 2 or 3)
<tagname>.ACT	INT	Modules online	Retrieves the quantity of modules in a group that are installed, powered, locked and communicating over the I/O bus (1, 2 or 3)
<tagname>.LOC	INT	Slot location	Retrieves the number of the slot allocated to the first module in a group. (1 to 18 or 1 to 24) [1](†) but does not identify the bus.
<tagname>.GH	BOOL	Group health	Retrieves the general health status of all modules in a group • TRUE: all modules are healthy • FALSE: one or more modules in the group is online and reporting a fault
<tagname>.XONL	BOOL	Online status	Retrieves the online status of module X • TRUE: the module is installed, powered, locked and is communicating over the I/O bus, otherwise FALSE
<tagname>.XHLY	BOOL	Health status	Retrieves the general health of module X • TRUE: the module is online and has no faults, otherwise FALSE
<tagname>.XRDY	BOOL	Ready status	Retrieves the ready status of module X • TRUE: the module is online and ready to send channel values, otherwise FALSE
<tagname>.XRUN	BOOL	Run status	Retrieves the run status of module X • TRUE: the module is online and reporting channel values, or requires manual intervention (pressing the **Fault Reset** button) before values can be reported, otherwise FALSE
<tagname>.XSDN	BOOL	Shutdown status	Retrieves the shutdown status of module X • TRUE: it is necessary to press the **Fault Reset** button before the module can send values, otherwise FALSE
<tagname>.XPOS	INT	Position	Retrieves the slot number of module X (1 to 18 or 1 to 24) [1] but does not identify the bus.

(1) The quantity of slots available on each bus depends on the type of the controller. The standard controller has 24 slots on each bus; a Eurocard version has 18 slots.

Resize a Group of Modules

You can decrease the size of a redundant group of modules from a triple to a double then from a double to a single module.

To decrease the group size:

1. Select the **Equipment View**.
2. Right click to select the module arrangement to change and delete the modules.
3. Select the empty slot and assign the reduced combination of modules to the slot.
4. Set the PST for the modules and configure the status and channel variables for the new group.

Move a Module to a Different Slot

You can move an assigned I/O module, or a group of modules, to a different slot (s).

To move a module:

- Select the **Equipment View**.
- Click on the module on the I/O bus and drag it to the new slot. You can also cut and paste a module to move it.

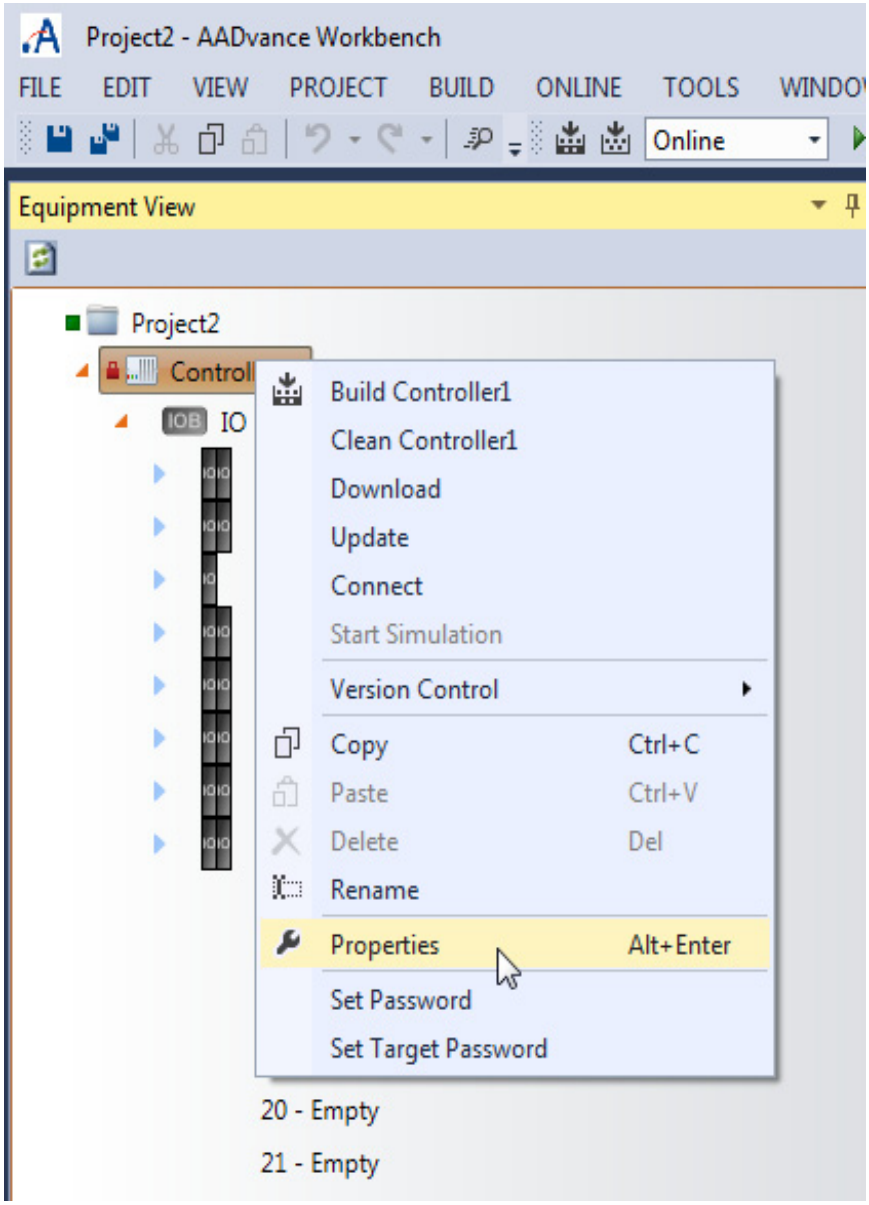
Remove an I/O Module

To remove an I/O module do the following:

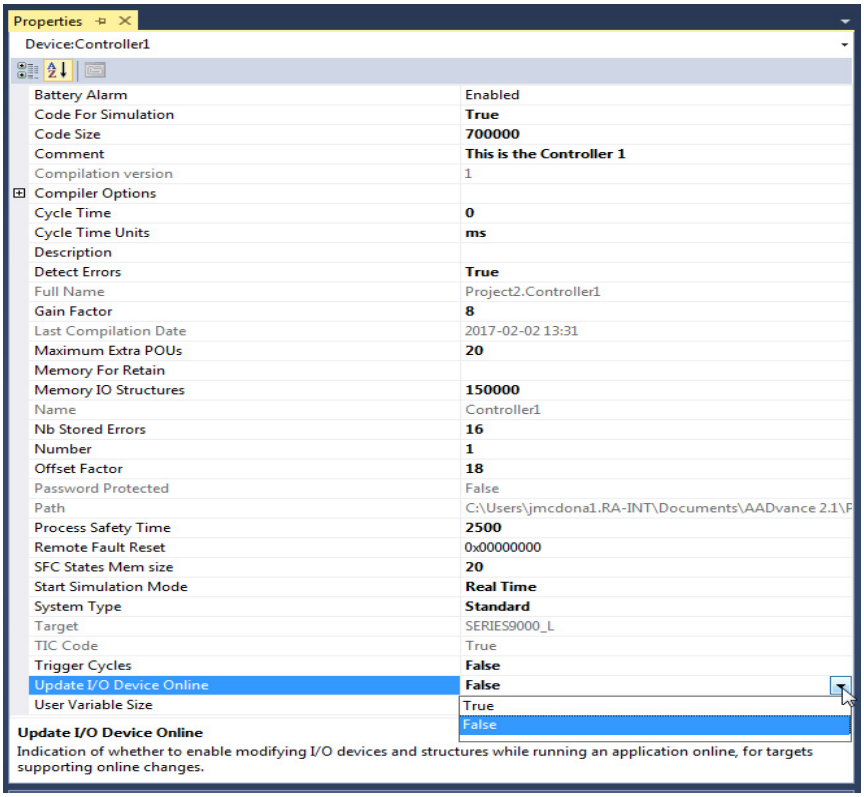
- Select the configuration in the **Equipment View**.
- Go to the unwanted module on the I/O bus and right-click, then select **Delete**.
- The module is removed from the slot and the channel variable wiring is unwired.
- The slot now displays as Empty.

Enable I/O Online Update

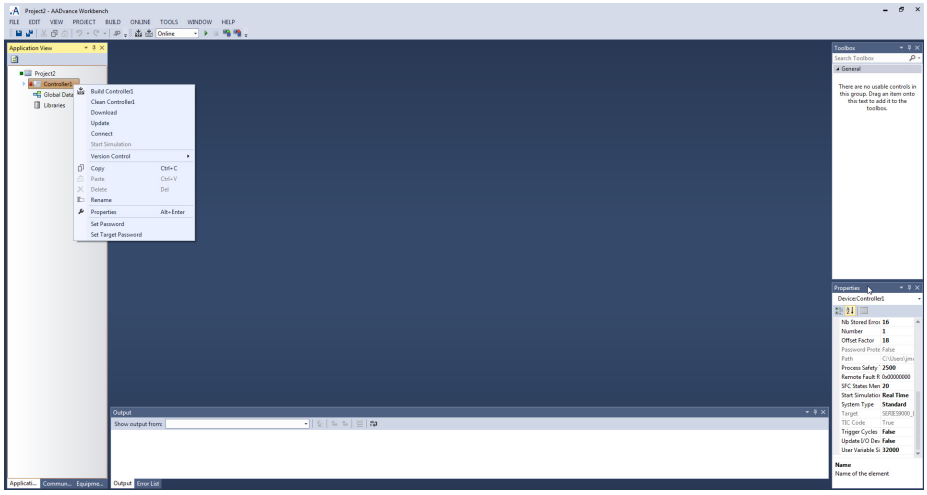
1. To enable the I/O Online Update feature, select the **Equipment View**.
2. Right-click the controller and select the **Properties** option. The Controller (Config1) Properties page is displayed.



3. Controller (Config1) Properties page is displayed.



The properties frame will appear in the bottom right corner of the AADvance Workbench 2.1 window default layout.



4. One of the Properties is Enable Online I/O Device Update. The default value is False (disabled).
5. To enable the update feature, set the Enable Online I/O Device Update.

Change the I/O Configuration with an Online Update

WARNING: Before changing the I/O configuration and doing an online update alternative safety measures should be set up and be available for the duration of the update.Doing an online update of a Safety Implemented System is the responsibility of the user. It is recommended that an online update only be performed if it absolutely necessary as it could reduce the safety integrity of the system while doing the change. If the changes are not done correctly it, can stop the running application.

Perform an I/O Configuration Change using an Online Update

- set up the alternative measures before making the hardware changes
- carry out the hardware changes
- ensure the online update is enabled in the controller properties
- write and download the online update program.

About Wiring to and from I/O Channels

The AADvance Workbench has two variable structures (compact and full) for the variables you wire to I/O channels. When declaring the variables, you can declare one of these structures, or you can assign the primary I/O variable directly to the base variable type.

When using one of the structures to declare the variable, the AADvance Workbench creates a set of variable elements having the same tagname. The system wires a set of I/O variables to the channel, depending on your chosen structure.

The syntax for a structure variable is <tagname>.XX where XX represents the reporting element of the variable. For example, <tagname>.DI is a Boolean that reports the digital input state for a channel.

Configuring Digital Inputs

Module Settings

You can configure the following settings for the Digital Input modules:

- **Process Safety Time**
  Use the procedure described earlier "Configure the I/O Module Process Safety Time" and either choose the default value or enter a custom value for the module.
- **Status Variable**
  Use the procedure described earlier in the topic "Wire a Status Variable to an I/O Module".
- **Threshold Values**
  A check box enables specifying custom values for the threshold settings or you can accept the default values.

Wire Input Channels for Reported Values

You can wire digital input channels to the following variable types and data structures to receive the reported values:

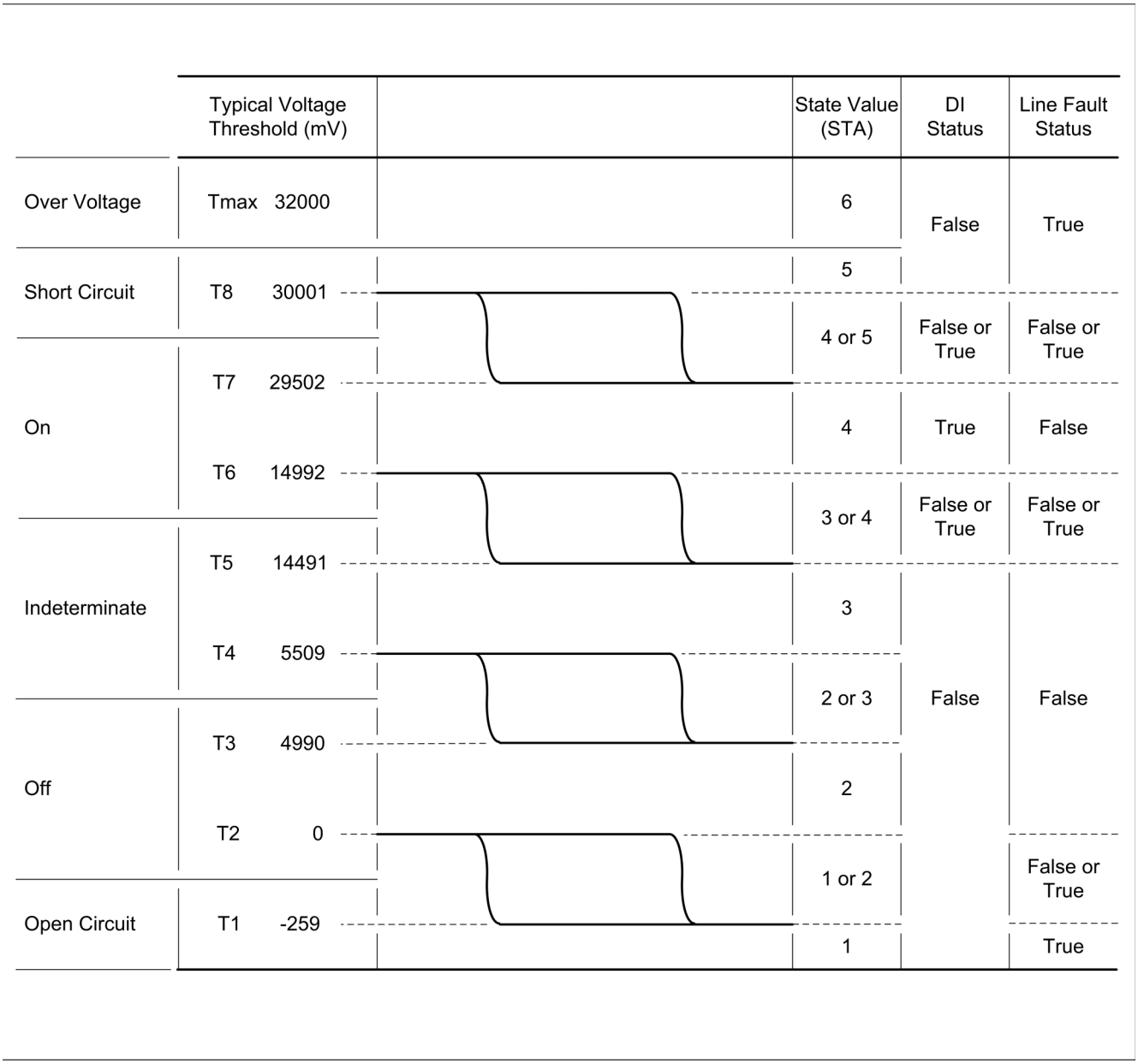
- BOOL (the <variable_name> retrieves the input module state)

- T9K_DI_Compact (provides three elements)
- T9K_DI_Full (provides six elements).

Threshold Values for Digital Inputs

The module determines the channel state and the line fault status by comparing the channel input voltage with a set of threshold values. You can make your own threshold values or use the default values. The values you choose for the module are inherited by each channel. You can subsequently define different thresholds for individual channels.

An indeterminate region is defined between the closed and open regions to detect marginal faults in either the sensor or external wiring. The AADvance controller provides hysteresis on the thresholds for increasing and decreasing values to help prevent chatter. The AADvance Workbench updates the reporting values during each application cycle.



Define Thresholds for a Digital Input

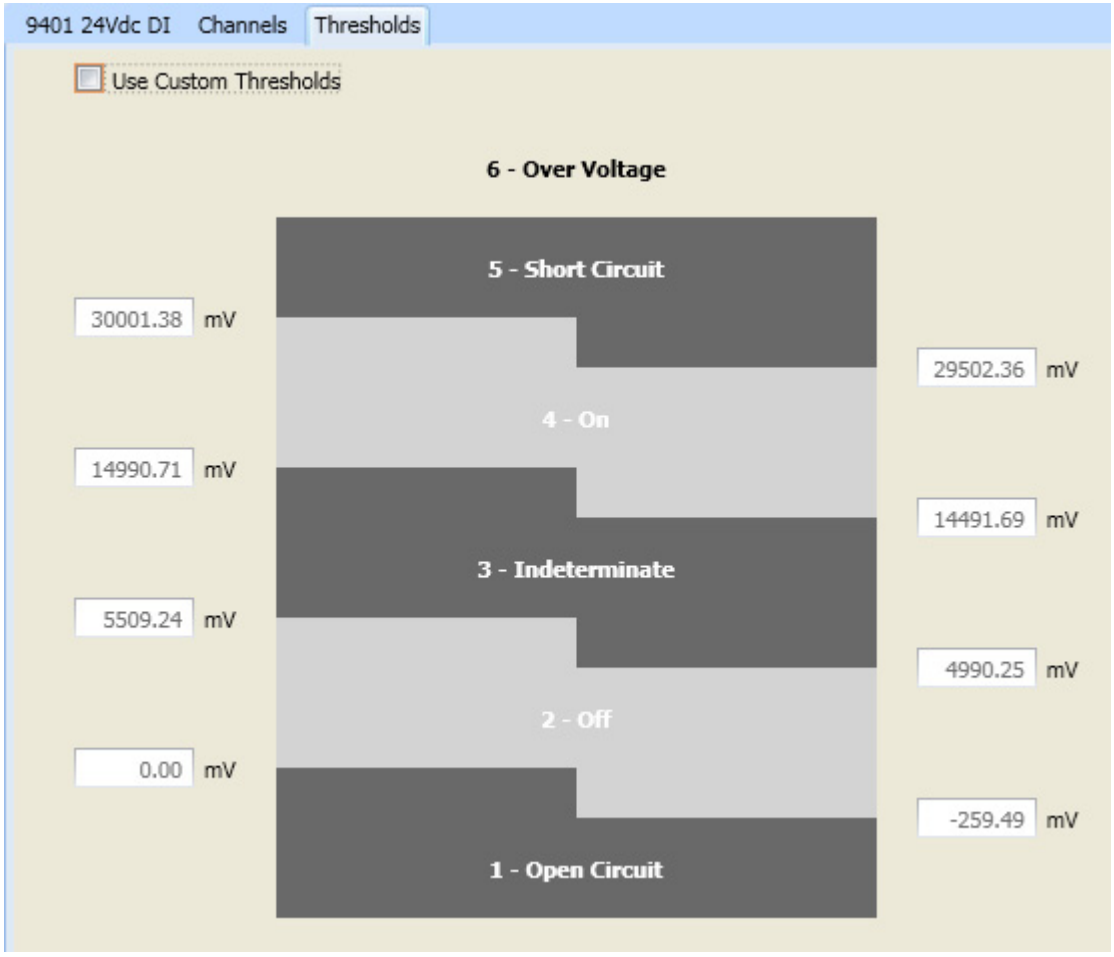
You can set custom thresholds for a digital input module. The custom thresholds become the default thresholds for each channel on the module. You can then also define custom thresholds for each individual input channel.

1. To define custom threshold values do the following:
2. Go to the Equipment view and select the input module on the I/O bus.
3. Select the **Equipment** tab on the content window, then choose the **Thresholds** tab.
   - The thresholds fields display a set of default values is shown in the threshold fields.
4. Select the Use **Custom Thresholds** check box, then specify your own values in the threshold fields.
5. To restore the default values, click **Default** and clear **Use Custom Thresholds,** then click **Apply**.
6. If necessary to define custom thresholds for an individual channel on the module, select the channel in the Equipment view and then repeat step 2 and step 3.

Default Thresholds for Digital Inputs

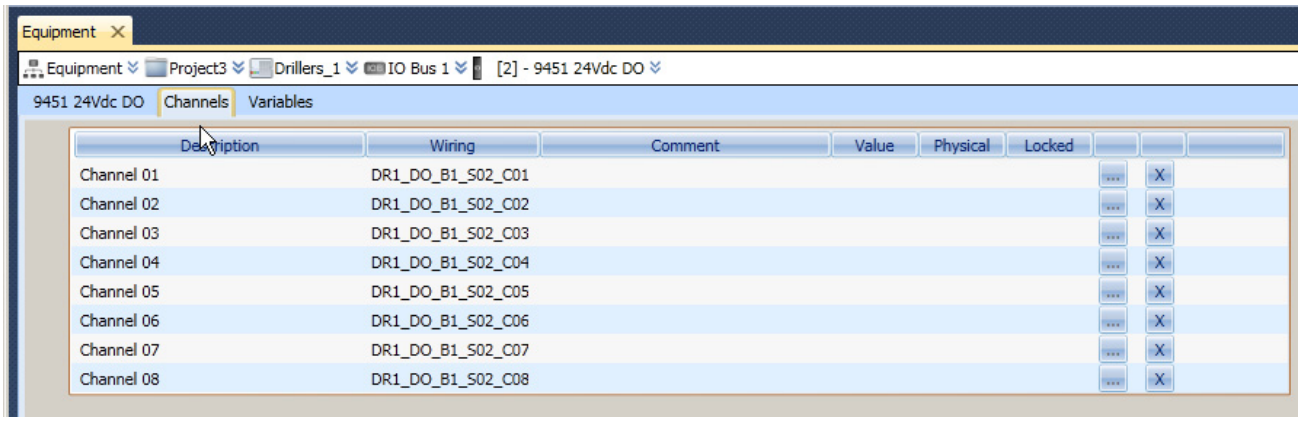
The default threshold values for digital inputs are for a standard (online monitored) 24 Vdc digital input channel.

The default values are as follows:

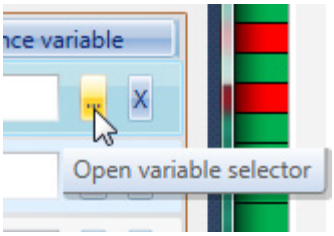


Wire a Channel Variable to a Digital Input

You wire a variable to a digital input channel to enable the application to use the input. Do the following:



1. Select the input module in the Equipment View and then select the channel that you want to wire to a variable.
2. Go to the **Equipment** tab on the content window and select the tab with the name of the channel (for example 'Channel 01' in the illustration).
3. Click the button beside the Channel Variable field.



   - The Variable Selector window is displayed.
4. Select the variable from the list, then click **OK**.
5. Repeat this procedure for other input channels.

T9K_DI_Compact and T9K_DI_FULL (Digital Inputs)

You wire the channel variables so the controller can receive the reported input values for the channels. The elements for both data structures for digital input channels (T9K_DI_COMPACT and T9K_DI_FULL) are indicated in the following tables:

Table 4 - T9K_DI_COMPACT Structure for Digital Inputs

Identifier	Type	Description	Remarks
<tagname>.DI	BOOL	Input state	TRUE: input voltage above threshold T6 FALSE: input voltage below threshold T5
<tagname>.LF	BOOL	Line fault	TRUE: input voltage above threshold T8; between T5 and T4; or below T1 FALSE: input voltage between thresholds T2 and T3; or between T6 and T7
<tagname>.DIS	BOOL	Discrepancy	TRUE: there is a discrepancy in voltage larger than 20 % exists between the channels of two or three modules in a redundant configuration (†)

Table 5 - T9K_DI_FULL Structure for Digital Inputs

Identifier	Type	Description	Remarks
<tagname>.DI	BOOL	Input state	TRUE: input voltage above threshold T6 FALSE: input voltage below threshold T5
<tagname>.LF	BOOL	Line fault	TRUE: input voltage above threshold T8; between T5 and T4; or below T1 FALSE: input voltage between thresholds T2 and T3; or between T6 and T7
<tagname>.DIS	BOOL	Discrepancy	TRUE: a discrepancy in voltage larger than 8 % (of 24 V) exists between the channels of two or three modules in a redundant configuration (1)
<tagname>.CF	BOOL	Channel fault	TRUE: module diagnostics found a fault in the channel electronics or firmware (state = 7)
<tagname>.V	UINT	Voltage	Reports the channel voltage in units of millivolts with an accuracy of ± 500 mV (2)
<tagname>.STA	USINT	State	Reports a state value for the channel: • 1 = open circuit • 2 = de-energized • 3 = indeterminate • 4 = energized • 5 = short-circuit • 6 = over voltage • 7 = faulted

(1) Discrepancy can only be reported TRUE when two or three modules are active in a group.

(2) The voltage element cannot report values below 0 mV.

Faulted State Value for Digital Inputs

A digital input channel is faulted when the channel cannot report a voltage within a safety accuracy specification of 10 % of the full scale measurement of the 24 V dc supply (2.4 V). The reported faulted state is a value of 7.

When the state reports the value 7, the following 'safe' values are reported by the other variables:

- Input State = FALSE
- Line Fault = TRUE
- Discrepancy = TRUE
- Channel Fault = TRUE
- Voltage = 0 mV.

Configuring Analogue Inputs

Settings

You can configure the following settings for the Analogue Input Modules:

- **Process Safety Time**
  Use the procedure described earlier "Configure the I/O Module Process Safety Time" and either choose the default value or enter a custom value for the module.
- **Status Variable**
  Use the procedure described earlier in the topic "Wire a Status Variable to an I/O Module".
- **HART Variables**
  You can enable HART for each channel and wire a HART variable.
- **Threshold Values**
  You can set custom values for the threshold settings or accept the default values.

Wire Input Channels for Reported Values

You can wire analogue input channels to the following variable types and data structures:

- REAL (the <variable_name> Retrieves a floating-point value representing 4 to 20 mA)
- T9K_AI_Compact (provides three elements)
- T9K_AI_Full (six elements)

The structures contain more information about the input, such as discrepancy status. You can also set up analogue inputs to operate with HART devices and define custom thresholds.

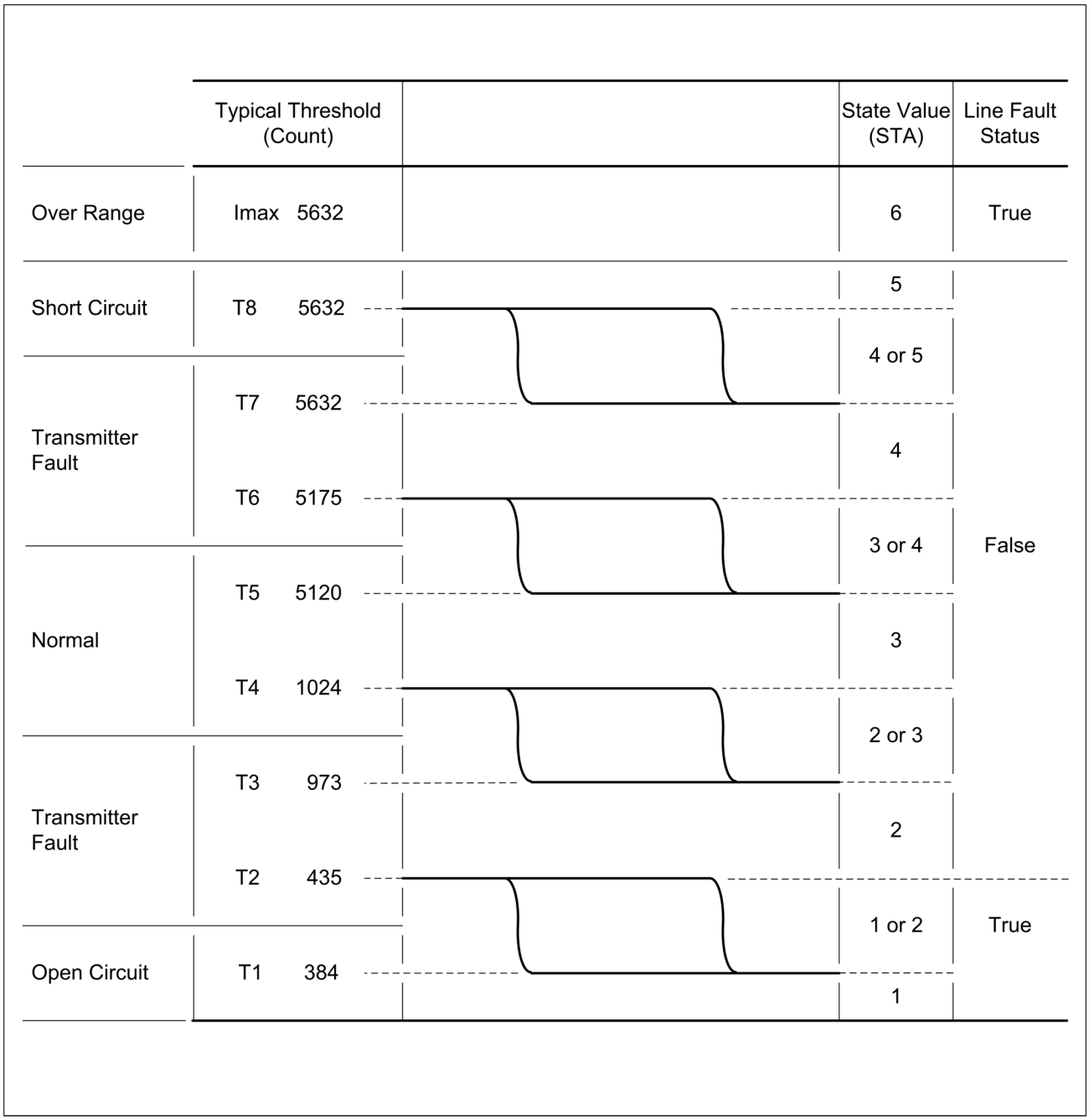
Threshold Values for Analogue Inputs

The module determines the channel state and the line fault status by comparing the channel input current with a set of threshold values. You can make your own threshold values or use the default values. The values you choose for the module are inherited by each channel. You can subsequently define different thresholds for individual channels.

Thresholds are specified in counts, with 0 (zero) being 0 mA, 1,024 being 4 mA, and 5,120 being 20 mA.

IMPORTANT	When the system is operational, you must change these values only using an online update.

The AADvance controller provides hysteresis on the thresholds for increasing and decreasing values to help prevent chatter. The AADvance Workbench updates the reporting values during each application cycle.



To define the custom threshold values do the following:

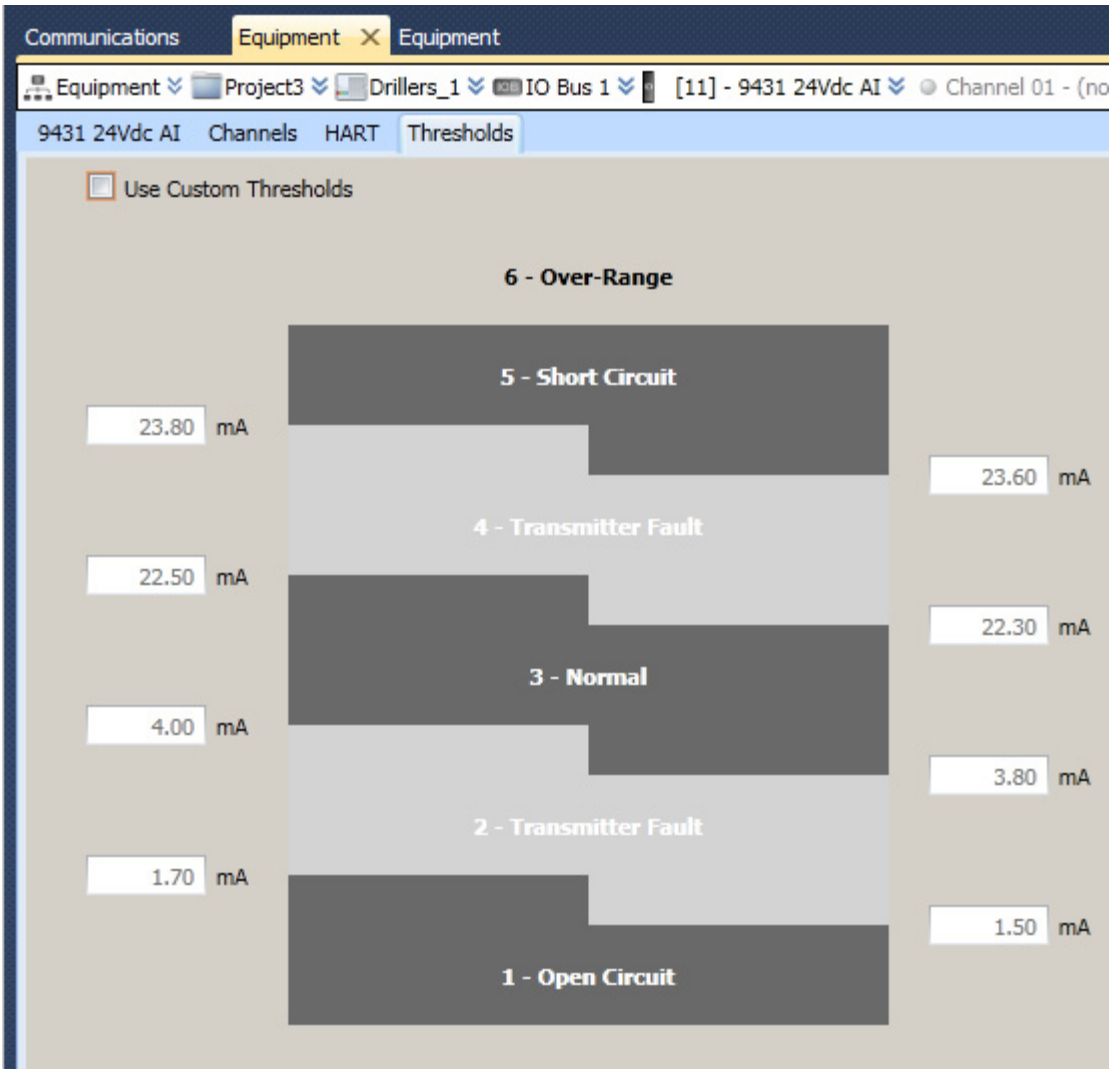
1. Go to the Equipment View and select the input module on the I/O bus.
2. Select the **Equipment** tab on the content window and select the **Thresholds** tab.
   - The thresholds fields display a set of default values.
3. Select the **Use Custom Thresholds** check box, then specify your own values in the threshold fields.
4. To restore the default values, click **Default** then clear the **Use Custom Thresholds** and click **Apply**.
5. If it is necessary to define custom thresholds for an individual channel on the module, select the channel in the Equipment View and repeat step 2 and step 3.

Default Thresholds for Analogue Inputs

The default threshold values for analogue inputs are for a standard (online monitored) 24 Vdc analogue input channel. The values agree with the lower limit and higher limits for an analogue signal given in the NAMUR NE43 standard.

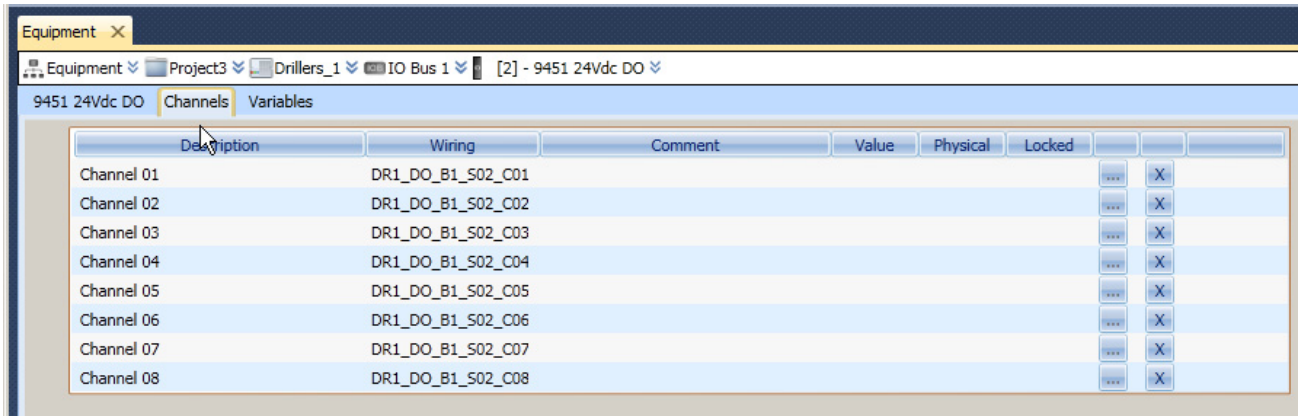
The default values are indicated in the illustration:

Figure 3 - Default Threshold Values for the 9431 Analogue Input Module

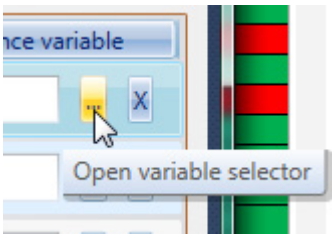


Wire a Channel Variable to an Analogue Input

Wire a variable to an analogue input channel to enable the application to use the input.



1. Select the input module in the Equipment View and then select the channel that you want to wire to a variable.
2. Select the **Equipment** tab on the content window, then the tab with the name of the channel (for example 'Channel 01' in the illustration).
3. Click the button beside the Channel Variable field.



   - The Variable Selector window is displayed.
4. Select the variable from the list, then click **OK**.
5. Set the Low and High counts for the Raw values received from the field elements.
6. Set the scaling factor as an Engineering Value in the range from 0 % to 100 %. With the Raw value set as shown, the resulting Process Values are as follows:
   - 1024 = 0 = 4 mA
   - 3072 = 50 = 12 mA
   - 5120 = 100 = 20 mA.
7. Repeat this procedure for other input channels.

T9K_AI_COMPACT and T9K_AI_FULL

The elements for both data structures for analogue inputs (T9K_AI_COMPACT and T9K_AI_FULL) are indicated in the following tables.

Table 6 - T9K_AI_COMPACT Structure for Analogue Inputs

Identifier	Type	Description	Remarks
<tagname>.PV	REAL	PV	Process Value. A scaled, floating-point value representing the analogue loop current. Default scaling factor is 0 to 100 % representing 4 to 20 mA.
<tagname>.CNT	INT	Raw count	A count representing the current on the channel in units of 1/256 mA • 0 represents 0 mA • 5,120 represents 20 mA Accurate to within ± 13 counts, equivalent to ± 0.05 mA
<tagname>.DIS	BOOL	Discrepancy	TRUE: there is a discrepancy in current larger than 2 % exists between the channels of two or three modules in a redundant configuration (1)

Table 7 - T9K_AI_FULL Structure for Analogue Inputs

Identifier	Type	Description	Remarks
<tagname>.PV	REAL	PV	Process Value. A scaled, floating-point value representing the analogue loop current. Default scaling factor is 0 to 100 % representing 4 to 20 mA
<tagname>.CNT	INT	Raw count	A count representing the current on the channel in units of 1/256 mA • 0 represents 0 mA • 5,120 represents 20 mA Accurate to within ± 13 counts, equivalent to ± 0.05 mA
<tagname>.LF	BOOL	Line fault	TRUE: state (.STA) is 1, 5, 6 or 7 FALSE: state (.STA) is 2, 3 or 4
<tagname>.DIS	BOOL	Discrepancy	TRUE: there is a discrepancy in current larger than 2 % exists between the channels of two or three modules in a redundant configuration (1)
<tagname>.CF	BOOL	Channel fault	TRUE: module diagnostics detect a fault in the channel electronics or firmware (state = 7)
<tagname>.STA	USINT	State	Reports a state value for the channel: • 1 = open circuit • 2 = transmitter fault (low) • 3 = normal • 4 = transmitter fault (high) • 5 = short-circuit • 6 = over range • 7 = faulted

(1) Discrepancy can only be reported TRUE when two or three modules are active in a group.

Faulted State for Analogue Inputs

An analogue input channel is faulted (the state reports a value of 7) when the channel cannot report a count in a safety accuracy specification of 1 % of the full scale measurement range of 5,120 (51 counts, 0.2 mA).

When the state reports the value 7 then the following 'safe' values are reported by the other variables:

- Process Value = a calculated value based on a Count value of 0
- Line Fault = TRUE
- Discrepancy = TRUE
- Channel Fault = TRUE
- Count = 0.

HART

Highway Addressable Remote Transducer (HART) is an open protocol for process control instrumentation. HART combines digital signals with analogue signals to supply control and status data for field devices.

The AADvance controller supports the use of HART on each analogue input and output channel. The application can use HART data to monitor and respond to device conditions and to supply diagnostic information such as data comparison and error reporting. This is a useful increase in the level of safety system diagnostics.

HART Features

The support for HART in the AADvance controller has the following features:

- HART support on each analogue input and output channel
- Variables for each analogue input and output channel to monitor HART device information.

HART

HART variables can be configured on each analogue input and output channel to monitor the HART field device.

Make sure that your HART field devices support HART command 0 ('read unique ID') and HART command 3 ('read current and four dynamic variables'). The AADvance controller uses these commands to communicate with the HART devices.

The AADvance analogue input and output modules use HART command #03 to collect data from the field device as specified by Revision 5 of the HART specification. The extra data available from HART-enabled field devices is reported to the application in custom data structures: T9K_AI_HART and T9K_AI_HART_FULL.

The structures supply the following data:

- loop current in milliamps
- process measurement in engineering units
- errors on HART communication seen by the field device
- status of the field device
- time since the most recent update, in milliseconds.

You can use the loop current variable for diagnostic checks in the application, to compare the value of the variable with the value on the 4 to 20 mA loop and react if there is a discrepancy. You can also monitor the status of the field device and use this to report diagnostic errors and manual configuration changes.

IMPORTANT	The update rate for HART data from field devices is slower than the update rate for the 4 to 20 mA analogue signal itself. HART data can take a maximum of 4 seconds to update, depending on the device type and configuration.

Precautions for HART in a Safety System

WARNING: If using HART in a safety system, follow these precautionary guidelines:
- Do not use HART variables as the primary initiator for a Safety Instrumented Function (SIF). The HART protocol does not satisfy the required safety integrity levels for Safety Instrumented Functions.
- Make sure to disable HART for field devices not having a locked configuration. This helps prevent the use of HART from changing a device configuration.
- Make sure that the custom data for the device (this is the data given in response to HART command #03) is used in accordance with the manufacturer recommendations.

T9K_AI_HART and T9K_AI_HART_FULL

The two data structures for HART variables (T9K_AI_HART and T9K_AI_HART_FULL) have the elements in the following tables:

Table 8 - T9K_AI_HART Data Structure

Identifier	Type	Description	Remarks
<tagname>.I	REAL	Current	Loop current in mA
<tagname>.V1	REAL	Variable	Primary loop current variable
<tagname>.U1	BYTE	Variable Units	Primary loop current variable units code
<tagname>.V2	REAL	Variable	Second loop current variable
<tagname>.U2	BYTE	Variable Units	Second loop current variable units code
<tagname>.V3	REAL	Variable	Third loop current variable
<tagname>.U3	BYTE	Variable Units	Third loop current variable units code
<tagname>.V4	REAL	Variable	Fourth loop current variable
<tagname>.U4	BYTE	Variable Units	Third loop current variable units code
<tagname>.COMMS	BOOL	Communication Status	HART Communication Status • TRUE: Communication OK • FALSE: Communication Stopped
<tagname>.DEVICE	BYTE	Device Status	Field Device Status: • Bit 7: field device malfunction • Bit 6: configuration changed • Bit 5: cold start • Bit 4: more status available • Bit 3: analogue output current fixed • Bit 2: analogue output saturated • Bit 1: non-primary variable out of limits • Bit 0: primary variable out of limits

The HART_FULL data structure is used for HART Pass-Through communication in WB2.1. This data structure has two additional elements - Elapsed Time data and the Pass-Through Communication Status.

Table 9 - T9K_AI_HART_FULL Data Structure

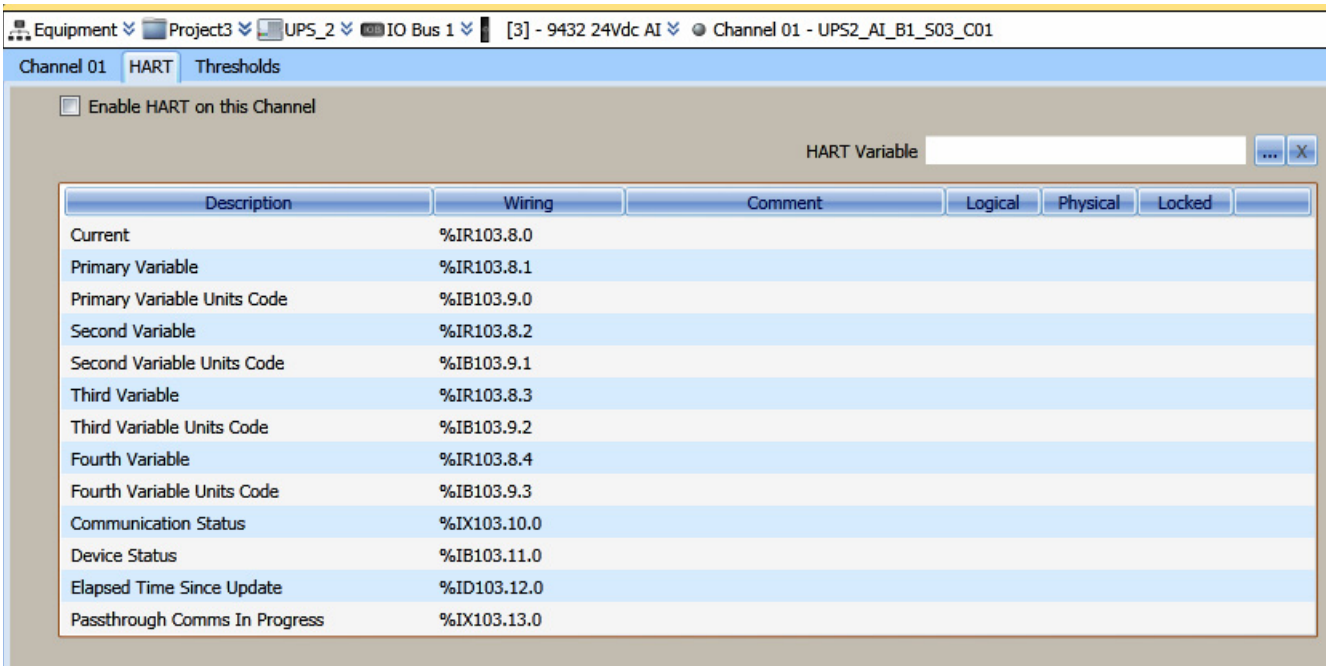
Identifier	Type	Description	Remarks
<tagname>.I	REAL	Current	Loop current in mA
<tagname>.V1	REAL	Variable	Primary loop current variable
<tagname>.U1	BYTE	Variable Units	Primary loop current variable units code
<tagname>.V2	REAL	Variable	Second loop current variable
<tagname>.U2	BYTE	Variable Units	Second loop current variable units code
<tagname>.V3	REAL	Variable	Third loop current variable
<tagname>.U3	BYTE	Variable Units	Third loop current variable units code
<tagname>.V4	REAL	Variable	Fourth loop current variable

Table 9 - T9K_AI_HART_FULL Data Structure

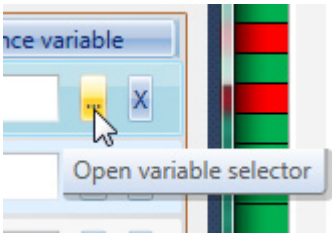
Identifier	Type	Description	Remarks
<tagname>.U4	BYTE	Variable Units	Third loop current variable units code
<tagname>.COMMS	BOOL	Communication Status	HART Communication Status • TRUE: Communication OK • FALSE: Communication Stopped
<tagname>.DEVICE	BYTE	Device Status	Field Device status: • Bit 7: field device malfunction • Bit 6: configuration changed • Bit 5: cold start • Bit 4: more status available • Bit 3: analogue output current fixed • Bit 2: analogue output saturated • Bit 1: non-primary variable out of limits • Bit 0: primary variable out of limits
<tagname>.TIME	DINT	Time in ms	Elapsed time since last non-Pass-Through communication. This parameter is reset to zero when data is received.
<tagname>.TIME	BOOL	Communication Status	Pass-Through Communication Status: • TRUE: Communication OK • FALSE: Communication stopped

Configure HART for Analogue Input Field Device Monitoring

To configure an analogue channel to use HART to monitor a field device wire the T9K_AI_HART variable for the channel. You can wire a set of HART variables to each channel, or you can wire a variable for one channel. However, you need to access a channel to enable HART for that channel.



1. Create a HART variable for the channel and set the type to T9K_AI_HART.
2. Select the module in the Equipment View and then select the channel that you want to configure for HART field device monitoring.
3. Select the **HART** tab.
4. On the channel, select the check box **Enable HART on this Channel**.
5. Click the button adjacent to the HART Variable field.



- The Variable Selector window is displayed.

6. Select the variable from the list, then click **OK**.
7. Repeat this procedure.

HART Pass-Through

The HART Pass-Through feature enables using an external asset management system to manage HART compatible field devices connected to an AADvance controller.

HART Pass-Through uses the device type manager (DTM) standard, defined by the HART Communication Foundation, to enable using any asset management system which using the generic 'Frame' standard. Examples of compatible tools are the Fieldcare application by Endress+Hauser and the FactoryTalk® Asset Centre by Rockwell Automation.

Using HART Pass-Through

HART variables can be configured on each analogue input and output channel to monitor HART Pass-Through data as follows:

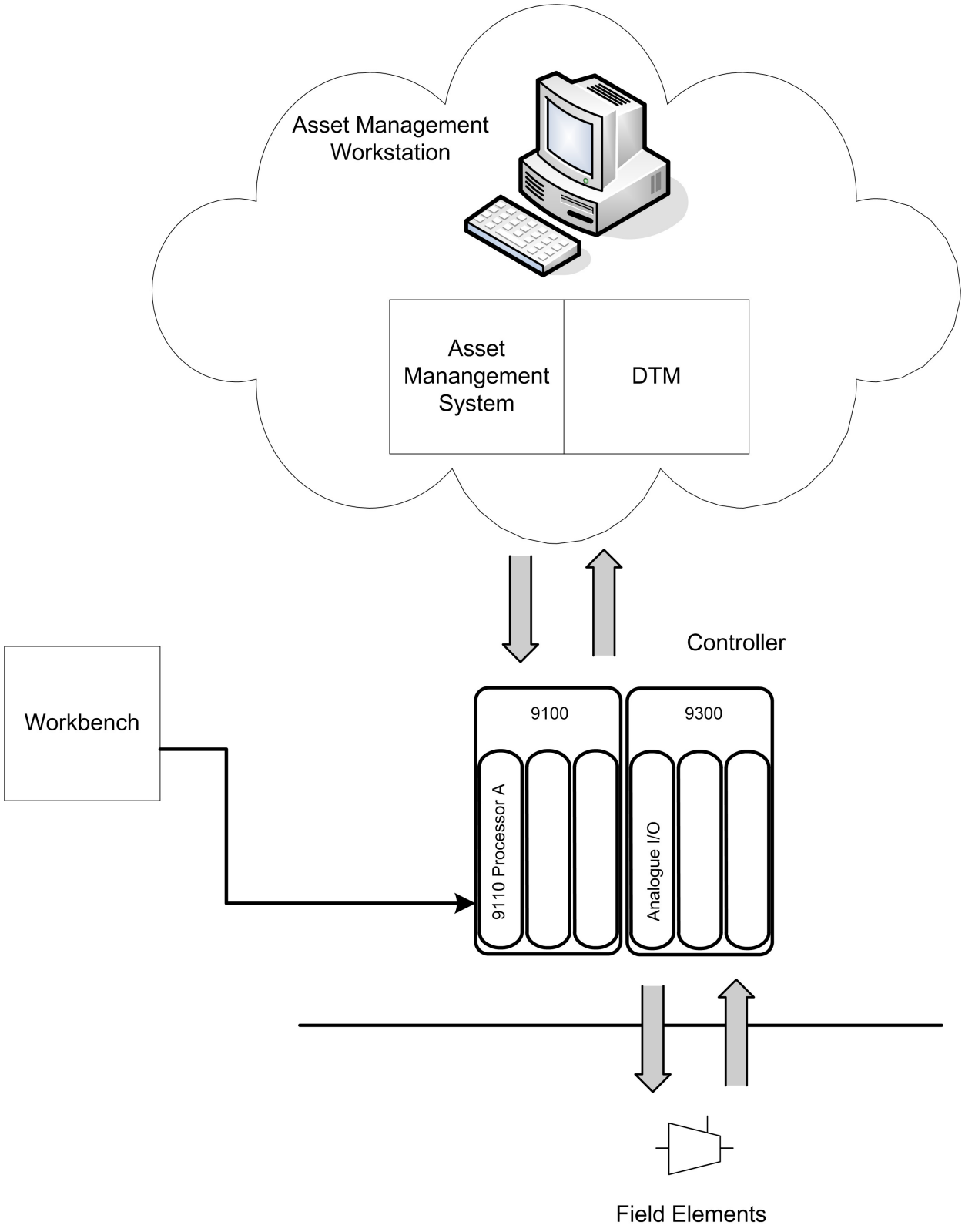
- They can be enables per channel.
- HART Pass-Through is enabled at the controller level.
- HART must be enabled on the channel to support HART (and HART Pass-Through must be enabled for the system).
- HART Pass-Through data isn't monitored. The HART feature in AADvance provides a controller capability to pass through requests/responses.

ATTENTION: HART Pass-Through has limitations on the maximum size HART message that can be processed:
For an Analogue input module the maximum message size is 0x5F (plus between 5 and x14 bytes preamble) for the command to remote HART device and 0x5F for a response (including up to 14 bytes for the preamble).

For the AOM the limits are 0x31 bytes for the command (plus between 5 and 0x14 bytes preamble) and 0x63 bytes for a response (including up to 0x14 bytes for the preamble).

Typical HART Arrangement

A typical arrangement of a system using HART Pass-Through is shown in the illustration:



To use HART Pass-Through you have to install the AADvance T9033 DTM software on the computer running the asset management system. You can then enable and disable the HART Pass-Through capability of the controller.

You can also set up the application to get status data for individual analogue channels. The supply data is provided by two elements in the T9K_AI_HART_FULL data structure which supply the data:

- <tagname>.ELAPSED (DINT), showing the time in milliseconds since the most recent valid non-Pass-Through communication. The value of this element resets to 0 (zero) when the application passes new HART data on the channel.
- <tagname>.PASS-THROUGH (BOOL), showing when the channel is carrying HART Pass-Through data.

The application can use these elements, for example, to let it make the decision when to permit HART Pass-Through communications.

HART Pass-Through Features

The support for HART Pass-Through in the AADvance controller has these features:

- Pass-Through support for HART standards 5, 6 and 7.
- Dedicated Ethernet port for HART Pass-Through communication.
- Supports the AADvance DTM supplied by Rockwell Automation.

NOTE HART Pass-Through should not be used on a safety application loop but is considered non-interfering on the SIL 3 loop.

Precautions for HART Pass-Through in a Safety System

WARNING: If you use HART Pass-Through in a safety system, follow these precautionary guidelines:
- Make sure that HART Pass-Through is enabled only under control of the application.
- Make sure that HART Pass-Through is enabled only when necessary.
- Configure the application to start an alarm if HART Pass-Through is enabled on any safety-critical channel of any module.

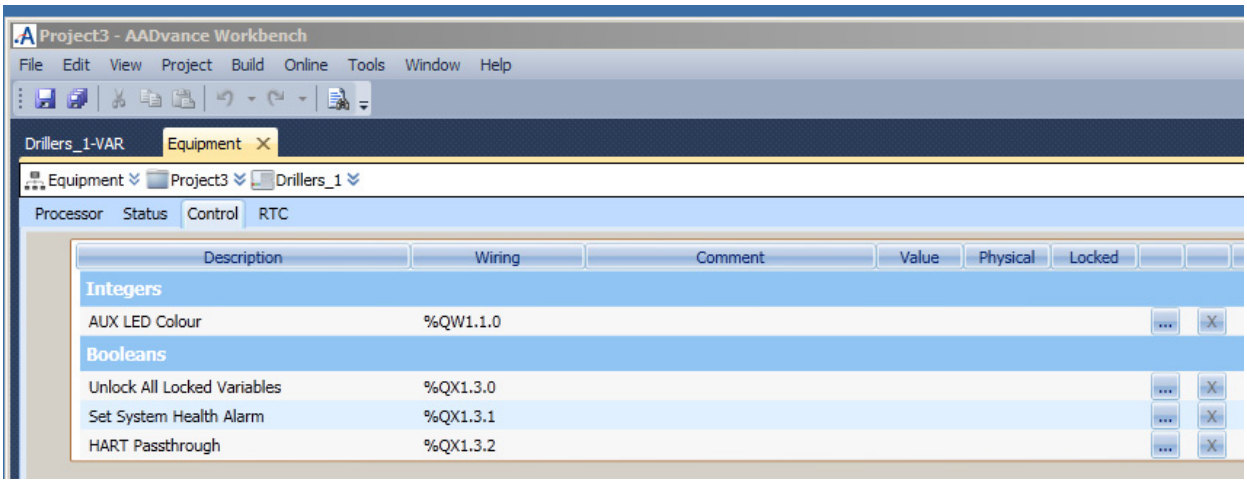
Install the T9033 DTM

The T9033 DTM is the device type manager for the HART Pass-Through feature in an AADvance system. It must be installed on the Windows computer which is running the asset management system. To install the DTM, do the following:

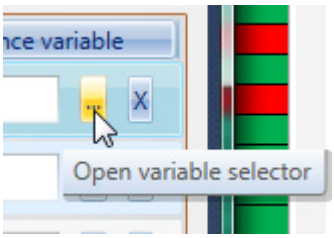
1. Un-install the existing version of the DTM installed on your Workstation.
2. Re-boot the system
3. Locate and run the file named AADvance DTM 1.xxx Setup.exe and run it.
   - The setup wizard is displayed.
4. Click **Next**.
5. Accept the terms of the license agreement, click **Next**.
6. Wait for the setup wizard to install the DTM.
7. The installation of the DTM is now completed.
8. Re-boot the system.

Enable HART Pass-Through in the Controller

IMPORTANT	To use HART Pass-Through at least one CIP Produce variable must be defined. If there are no CIP Produce and Consume variables the CIP stack does not activate. Refer to the CIP Chapter in this manual for instructions on setting up CIP Produce and Consume variables.



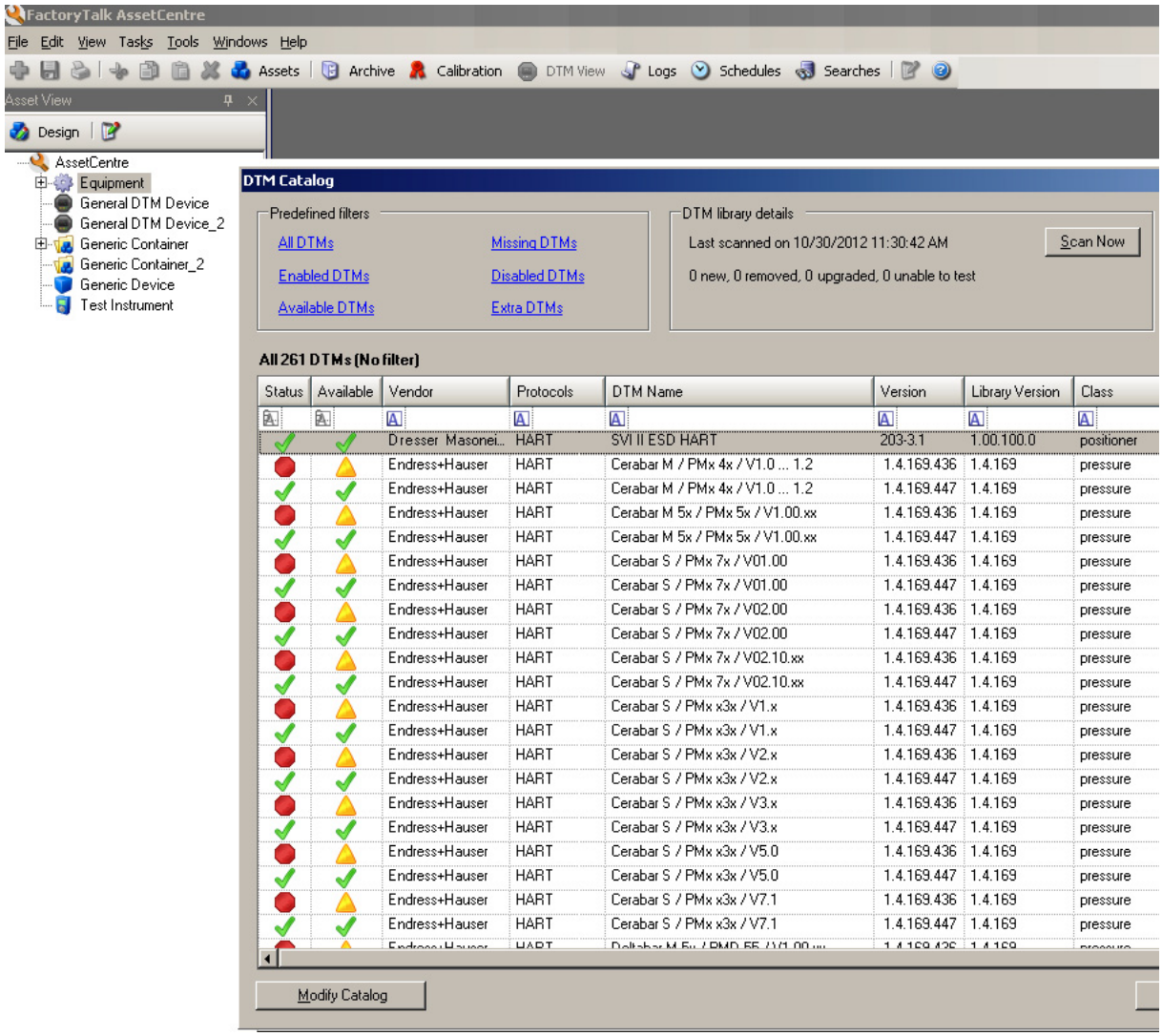
1. In the Dictionary create a variable named HART_CONTROL which the application uses to enable and disable HART Pass-Through. The variable must be a BOOL type with its direction set to output.
2. Select the HART Pass-Though variable selector button.



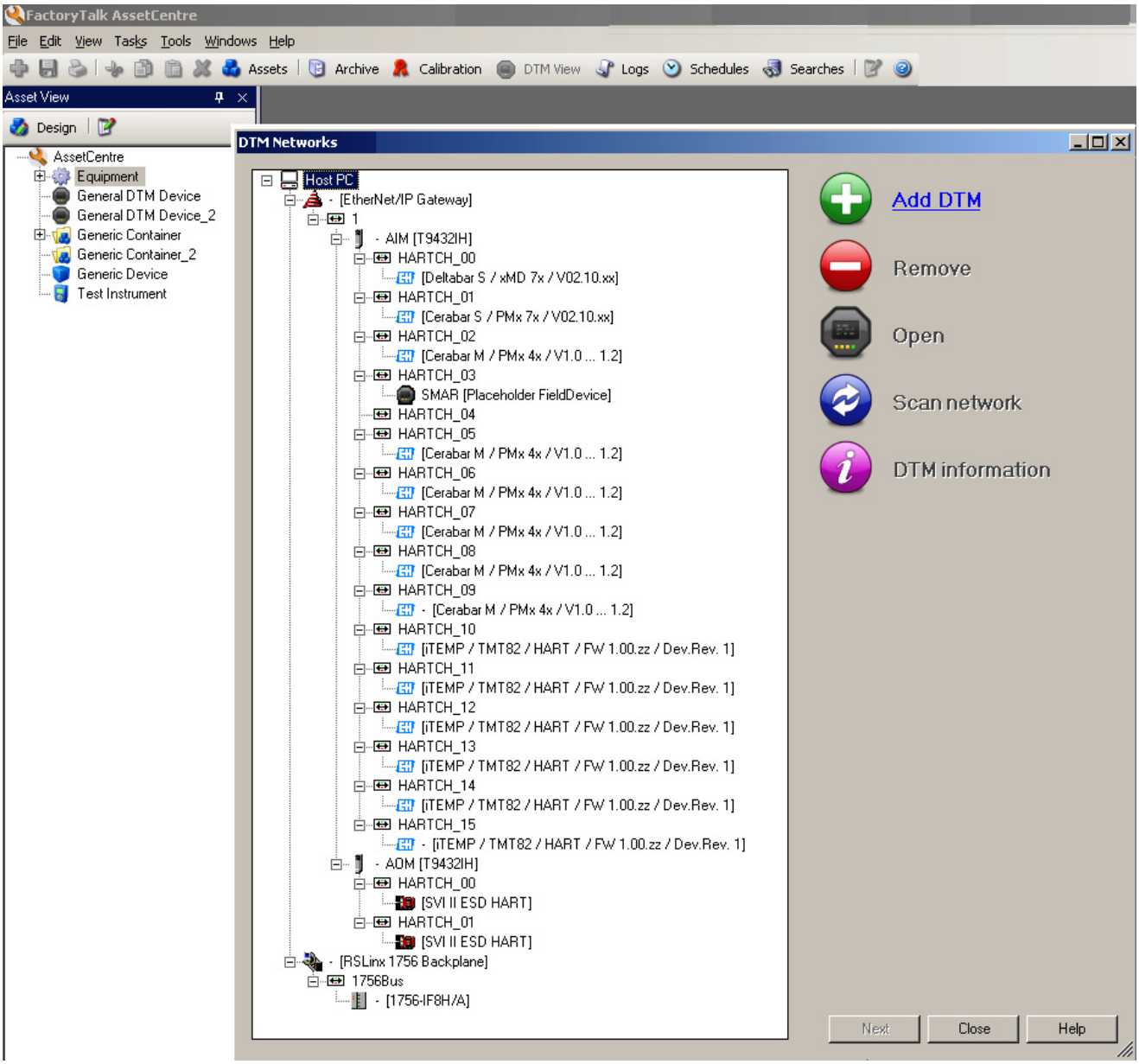
   - The Variable Selector window opens.
3. Select the HART_CONTROL variable to wire it to the processor control variable.

Use FactoryTalk Asset Center with an Analogue Module

This example shows how to configure the FactoryTalk Asset Center to let it use the AADvance HART Pass-Through feature with an analogue input module as well as an analogue output module.



1. Open the FactoryTalk Asset Centre and select **Tools** ➔ DTM Catalog.
2. Click **Scan Now**.
   - All DTMs are scanned and automatically updated.

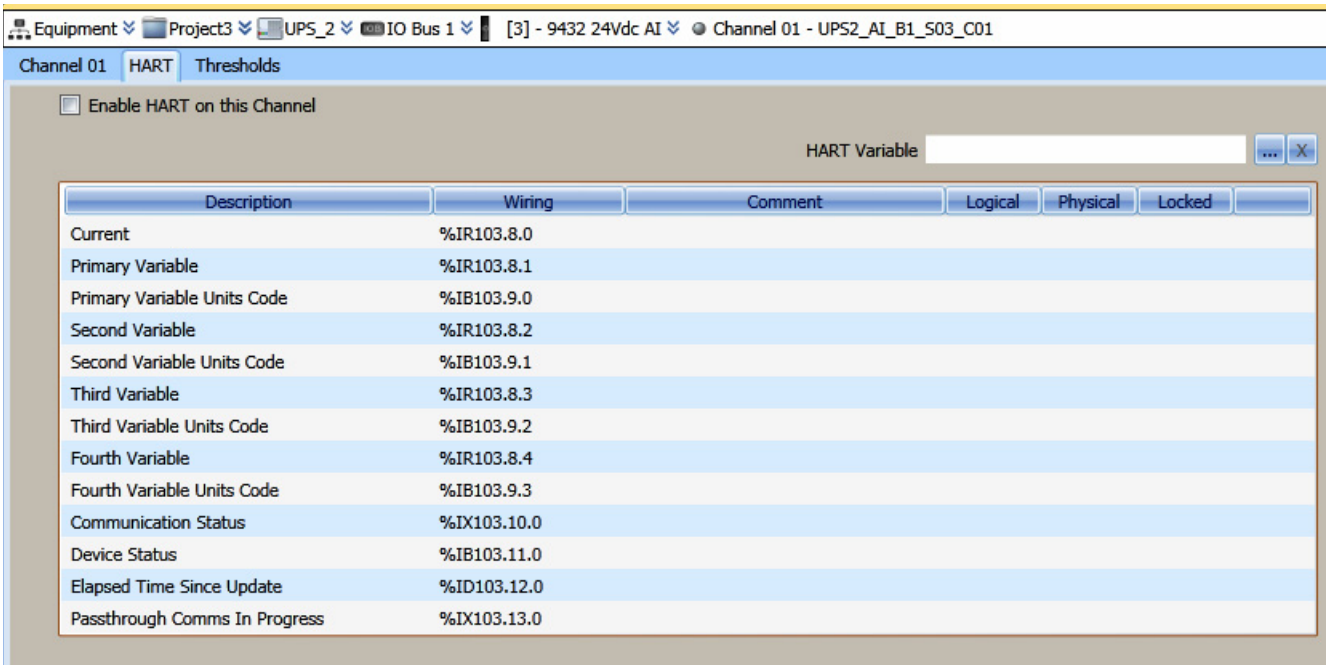


3. Un-install the existing version of the DTM installed on your Workstation.
4. Re-boot the system.
5. Right-click on the Host computer item at the top of the tree, then click **Add DTM**.
6. Select EtherNet/IP Gateway, then click **OK**.
   - The tree structure under Host computer is displayed.
7. Right-click on 1 below Ethernet/IP Gateway and click **Add DTM**.
8. Select T9432IH.
   - The tree structure below Ethernet/IP Gateway is displayed.
9. Name it AIM [T9432IH]
10. Right-click on AIM [T9432IH] and click Configuration.
11. Enter the IP address for the AADvance controller and the slot number for the analogue input module, then click **Next**.
12. Right-click on AIM [T9432] and click Scan network.

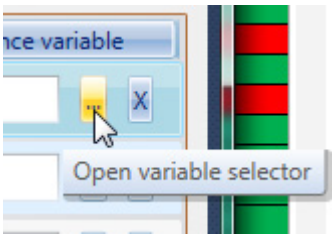
13. Select all of the channels in the Select Communication Channel window and click **OK**.
    - all the input channels on the AADvance controller are now set up.

Configure HART Pass-Through Monitoring

To configure an analogue input channel to supply HART and use Pass-Through communication monitoring:



1. Create a HART variable for the channel and set the type to T9K_AI_HART_FULL.
2. In the Equipment view, select the module, then the channel to configure for field device monitoring.
3. Select the **HART** tab.
4. Click the check box **Enable HART on this Channel**.



- Click the button adjacent to the HART Variable field.

5. The Variable Selector dialog box is displayed.
6. Select the T9K_AI_HART_FULL variable from the list, then click **OK**.
7. Repeat this procedure for other channels that will use HART enabled devices.

Configuring Digital Outputs

Settings

You can configure the following settings for Output Modules:

- **Process Safety Time**
  Use the procedure described earlier "Configure the I/O Module Process Safety Time" and either choose the default value or enter a custom value for the module.
- **Module Status Variables**
  Use the procedure described earlier in the topic "Wire a Status Variable to an I/O Module" to wire the Status Booleans and the Field Power Status Integers.
- **Channel Advanced Settings**
  Set the Advanced options.

Wire Output Channels

You can wire digital output channels to the following variable type and data structures:

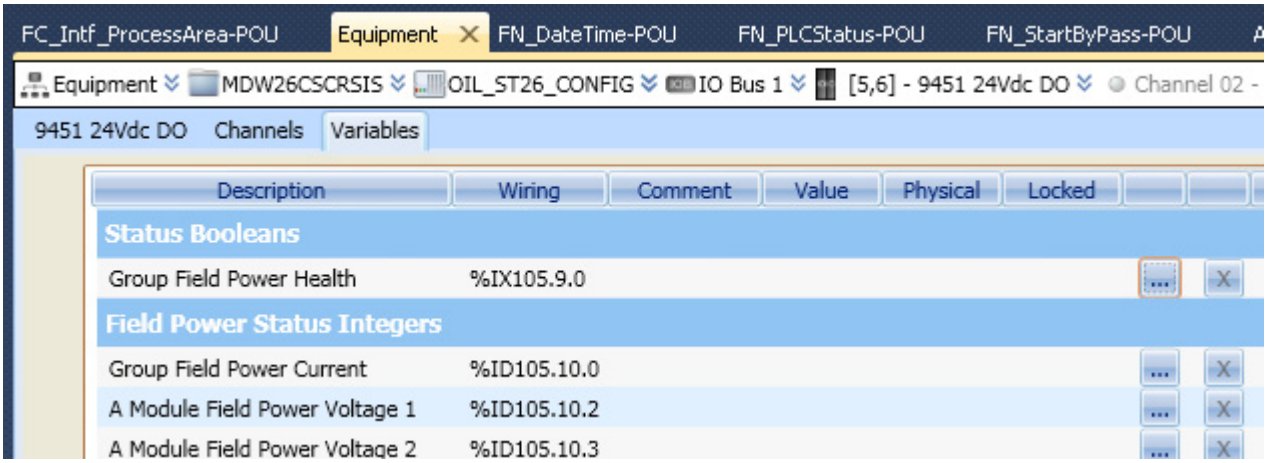
- BOOL (<variable_name> gives the commanded state)
- T9K_DO_Compact (provides three elements)
- T9K_DO_Full (seven elements).

The structures provide data about the output such as line fault status and discrepancy status.

IMPORTANT The controller writes its digital outputs one time in each application cycle and the digital output variables are updated one time in each application cycle.

Variables for a Digital Output Module

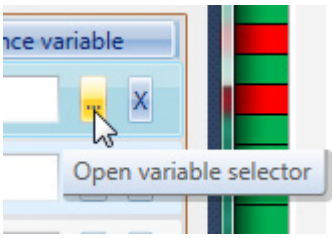
The 9451 digital output module provides Status Variables that are available to the application. The variables are Status Booleans and Field Power Status Integers.



Wire Status Variables to a Digital Output Module

To wire a Status Variable to a digital output module:

1. Select the output module in the Equipment View, then select the channel to wire to a variable.
2. Select the **Equipment** tab on the content window, then select the **Variables** tab.
3. Click the button adjacent to the variable to be wired.



   - The Variable Selector dialog box is displayed.
4. Select the variable from the list, then click **OK**.
5. Repeat this procedure for other Status Variables.
6. To unwire a Status Variable, click the X button adjacent to it.

Status Booleans

The Status Booleans supply data to the application about the field power supplies to a group of digital output modules.

Group Field Power Health

Direction: input to application from controller

Type: Boolean

Values:

- TRUE = all field power supplies for all active digital output modules in the group are in the range 18 V to 32 V dc inclusive
- FALSE = one or more field power supplies to an active module is less than 18 V dc or more than 32 V dc.

Description:
Provides a top level indication of the health of field power supplies to active digital output modules.

IMPORTANT	The controller incorporates a 0.5 V hysteresis on these thresholds to help prevent chatter. The controller declares a fault when a supply falls below 18 V but does not clear the fault until the supply rises to 18.5 V. Similarly the controller declares a fault when a supply exceeds 32 V but does not clear the fault until the supply falls below 31.5 V.

Field Power Status Integers

The Field Power Status Integers (all DINT) supply data to the application, about the field power supplies to a group of digital output modules.

Group Field Power Current

Direction: input to application from controller

Type: DINT

Values:

- 0 to 8,000 mA or larger (limited by capacity of DINT variable)

Description:
Reports the total current that all active digital output modules in a group are drawing from the field power supply. Accuracy is ± 10 %.

A Module Field Power Voltage 1:

Direction: input to application from controller

Type: DINT

Values:

- 0 to 48,000 mV or larger (limited by capacity of DINT variable).

Description:
Reports the voltage from the field power supply for the specified module and field power input. Accuracy is ± 500 mV.

A Module Field Power Voltage 2:

Direction: input to application from controller

Type: DINT

Values:

- 0 to 48,000 mV or larger (limited by capacity of DINT variable).

Description:
Reports the voltage from the field power supply for the specified module and field power input. Accuracy is ± 500 mV.

B Module Field Power Voltage 1:

Direction: input to application from controller

Type: DINT

Values:

- 0 to 48,000 mV or larger (limited by capacity of DINT variable).

Description:
Reports the voltage from the field power supply for the specified module and field power input. Accuracy is ± 500 mV.

B Module Field Power Voltage 2:

Direction: input to application from controller

Type: DINT

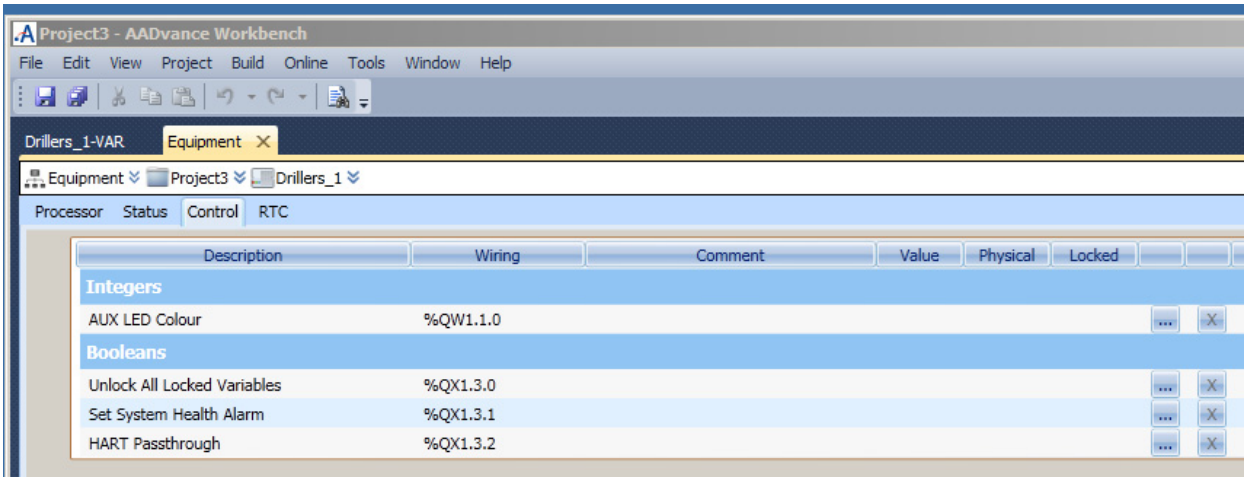
Values:

- 0 to 48,000 mV or larger (limited by capacity of DINT variable).

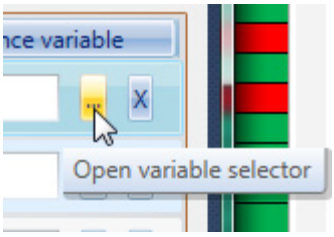
Description:
Reports the voltage from the field power supply for the specified module and field power input. Accuracy is ± 500 mV.

Wire a Channel Variable to a Digital Output

Wire a variable to a digital output channel to let the application use the output:



1. Select the output module in the Equipment View, then select the channel that you want to wire to a variable.
2. Select the **Equipment** tab on the content window, then select the tab with the name of the channel (for example 'Channel 01' in the illustration).
3. Click the button adjacent to the Channel Variable field.



   - The Variable Selector window is displayed.
4. Select the variable from the list, then click **OK**.
5. Repeat this procedure for other output channels.

T9K_DO_COMPACT and T9K_DO_FULL

The data structures for digital inputs (T9K_DO_COMPACT and T9K_DO_FULL) supply the elements shown in the tables.

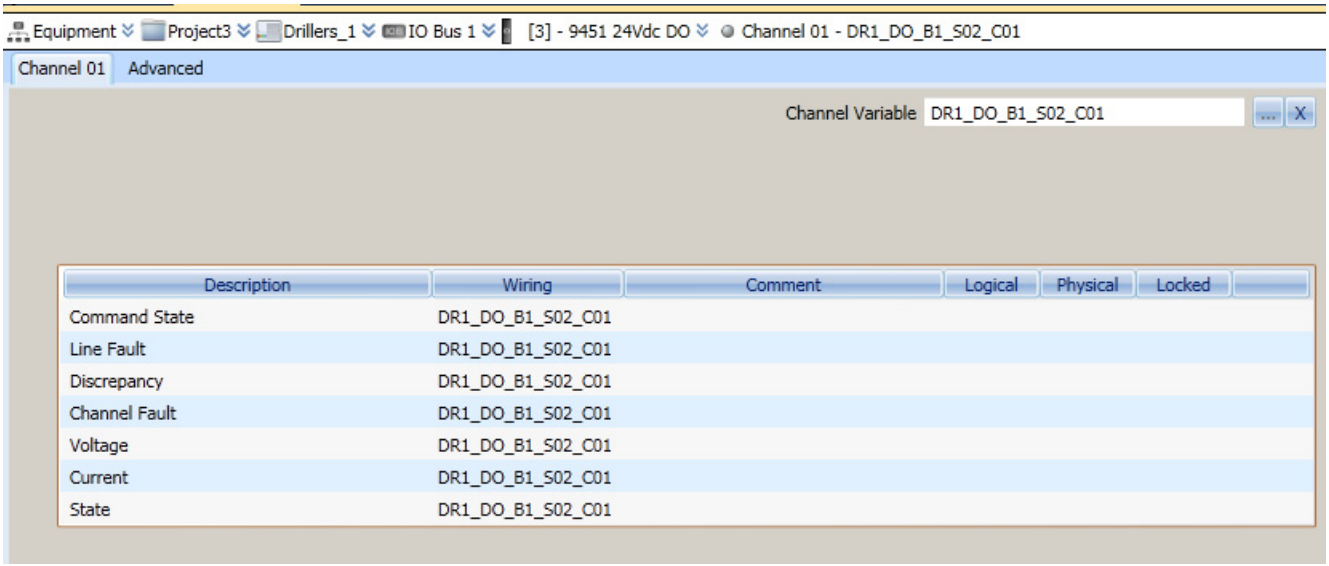


Table 10 - T9K_DO_COMPACT Structure for Digital Outputs

Identifier	Type	Description	Remarks
<tagname>.DOP	BOOL	Commanded State	The output demand to be passed to the output channel • Set to TRUE to energize • Set to FALSE to de-energize
<tagname>.LF	BOOL	Line fault	TRUE: there is no field supply, no load is connected, or a short circuit is detected
<tagname>.DIS	BOOL	Discrepancy	TRUE: there is a discrepancy in current larger than 1 % exists between the channels of two modules in a redundant configuration (1)

Table 11 - T9K_DO_FULL Structure for Digital Outputs

Identifier	Type	Description	Remarks
<tagname>.DOP	BOOL	Output demand	The commanded state to be passed to the output channel • Set to TRUE to energize • Set to FALSE to de-energize
<tagname>.LF	BOOL	Line fault	TRUE: there is no field supply, no load is connected, or a short circuit is detected
<tagname>.DIS	BOOL	Discrepancy	TRUE: a discrepancy in current larger than 1 % exists between the channels of two modules in a redundant configuration (1)
<tagname>.CF	BOOL	Channel Fault	TRUE: module diagnostics identify a fault in the channel electronics or firmware (state = 7)

Table 11 - T9K_DO_FULL Structure for Digital Outputs

Identifier	Type	Description	Remarks
<tagname>.V	UINT	Voltage	Reports the channel voltage at the output terminals, in units of millivolts with an accuracy of ± 500 mV(2)
<tagname>.I	INT	Current	Reports the current for the channel in milliamps with an accuracy of ± 2 mA and ± 10 % of measurement
<tagname>.STA	USINT	State	Reports a state value for the channel: • 1 = no vfield • 2 = de-energized • 3 = no load • 4 = energized • 5 = short-circuit • 6 = field fault • 7 = faulted

(1) Discrepancy can only be reported TRUE when two modules are active in a group.

(2) The voltage element cannot report values below 0 mV.

The State Variable for Digital Outputs

The state variable for a digital output is an unsigned integer with a value from 1 to 7 representing the following:

- 1 = no vfield: the field supply voltage is at or below 18 V dc for that channel.

IMPORTANT When the state variable is 1, the field voltage (<tagname.V>) is reported as 0 mV.

- 2 = de-energized: the commanded state is FALSE and the channel is de-energized.
- 3 = no load: the controller cannot identify a load connected to the channel field wiring or the load is below the minimum required channel load of 10 mA when commanded TRUE.
- 4 = energized: the commanded state is TRUE and the channel is energized.
- 5 = short-circuit: the controller has detected a short-circuit condition, irrespective of the channel drive state.
- 6 = field fault: an external source is driving the channel to an energized state or a voltage larger than 18 Vdc, irrespective of the channel drive state.
- 7 = faulted.

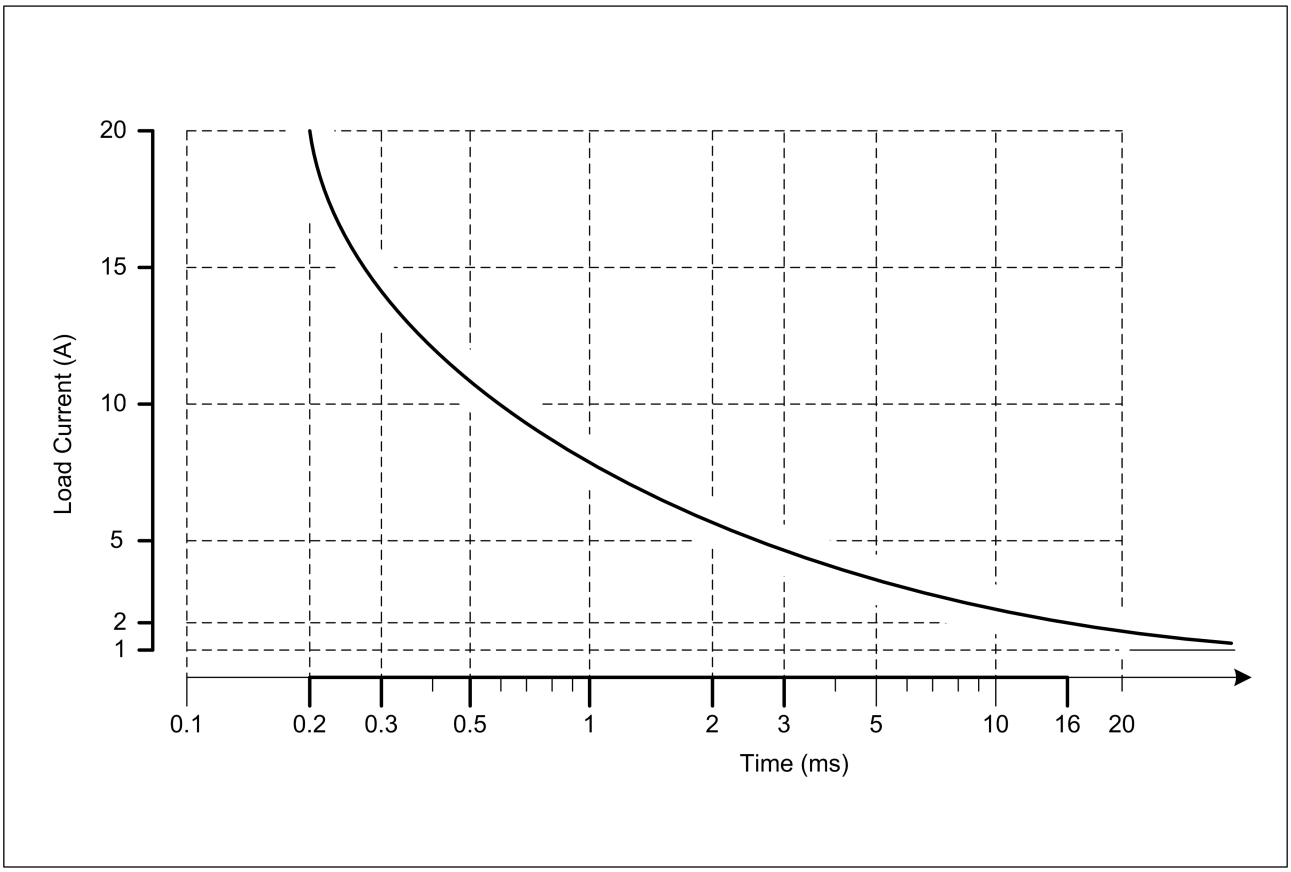
Protection for Digital Outputs

The AADvance controller has three mechanisms to protect its digital output channels:

- inrush current protection
- short circuit protection for energized channels
- short circuit protection for de-energized channels.

The controller tolerates inrush currents so that its digital outputs can energize capacitive loads without causing the controller to report a short-circuit. The illustration shows the characteristics of the maximum load currents that the controller tolerates when a digital output is commanded on. If the load current

enters the region above the curve on the graph, the controller applies its inrush current protection.



After allowing for inrush, the controller engages its short-circuit protection for an energized channel when the loop current reaches 2 A.

- Short-circuit detection on an energized channel is immediate and the channel is de-energized. The controller reports the condition until the short-circuit is cleared.
- When the short circuit is removed, the channel re-energizes. The short-circuit report is then cleared by pressing the **Fault Reset** button on the 9110 processor module or by setting the commanded state to FALSE.

The controller checks de-energized digital output channels for possible short-circuits. Periodically, the controller partially turns on each de-energized output in turn and measures the loop current. If the loop current shows a loop resistance of less than approximately 10 ?, the controller reports a short-circuit.

Faulted State for Digital Outputs

A digital output channel is faulted when normal operation or diagnostic tests find a known fault condition.

When a channel is faulted, the state reports the value 7, and the other variables report the following 'safe' values:

- Line Fault = TRUE
- Discrepancy = TRUE

- Channel Fault = TRUE
- Voltage = 0 mV
- Current = 0 mA

NOTE For details of how the AADvance digital output module detects field faults, see Knowledgedase Document ID OA23147 AADvance/ bulletin 1715: Digital output channel diagnostic test.
Sign in to your Rockwell Automation account to view Knowledgebase articles.

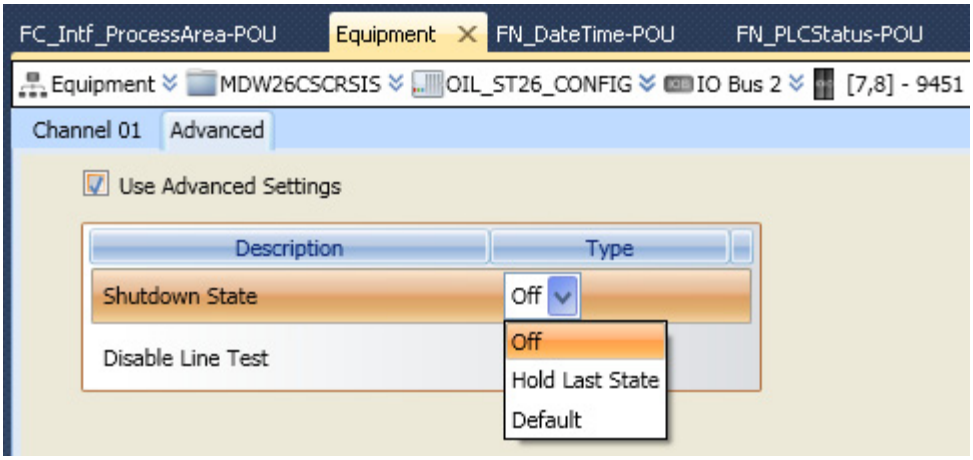
Configure the Channel Advanced Settings

You can configure the following Advanced settings:

- Shutdown State
- Disable Line Test.

Shutdown State

The Shutdown State defines how the output behaves when its parent 9451 digital output module suffers a loss of communications with the processor or a failure in the module and goes into its shutdown mode.



To configure the Shutdown State:

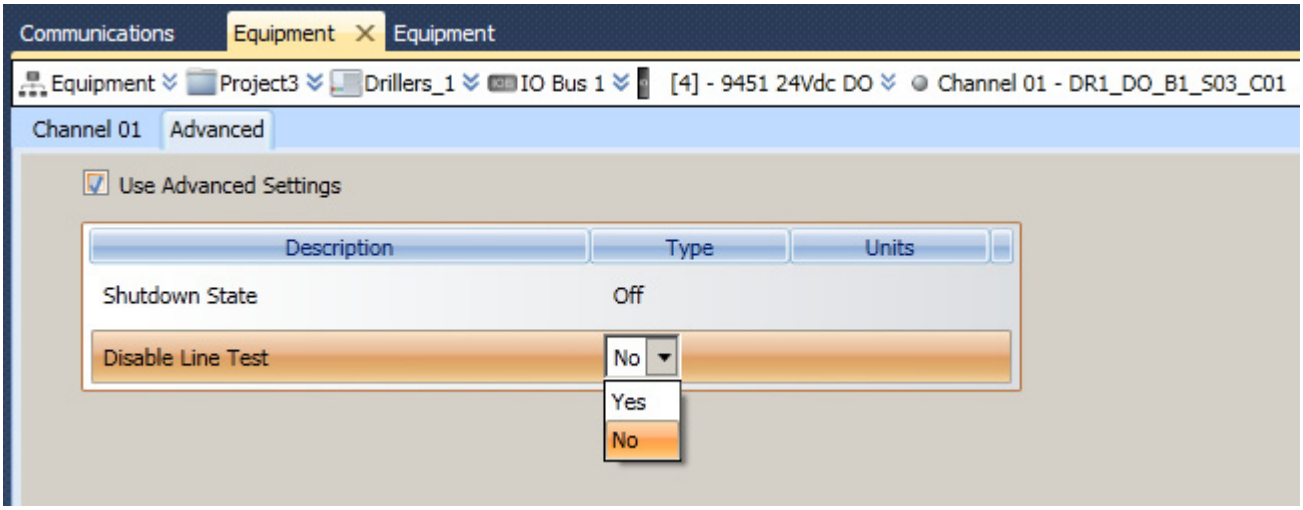
1. Select the output module in the Equipment View and then necessary channel to configure.
2. Go to the **Equipment** tab on the content window and select the **Advanced** tab.
3. Select **Use Advanced Settings**.
4. Set the Shutdown State to one of these options in the pull down list:
   - off (default) disables the output when the module is shutdown
   - Hold Last State forces the output to stay in its most recent commanded state during a module shutdown
   - the Default option in the pull down list is not used.

WARNING: If you use Hold Last State for a safety-related output you must restore the communications in less than the MTTR or use other compensating measures during a module failure to keep the safety rating of the system.

5. Repeat this procedure for other output channels.

Disabling the Line Test for a Digital Output

You can set the Disable Line Test to Yes or No.



The 9451 digital output module performs a test for a no-load condition on each output. The test is called the 'Disable Line Test'. A no-load condition occurs when the controller cannot find a load connected to the field wiring or the load current is below 20 mA when the output is commanded TRUE. You will disable the line test if you want to connect a low load to an output or if the output is not used and you prefer not to fit a dummy load.

When the line test is enabled, the module reports a no-load condition by setting the state variable (<tagname>.STA) to the value 3 and by setting the channel LED to amber. After disabling the line test, then assuming there are no other faults present, the state variable indicates 2 or 4 (depending on the commanded value) rather than 3 and the channel LED indicates off or green rather than amber.

IMPORTANT If you disable the line test, there are other scenarios, such as no field voltage that set the channel LED to amber.

Configuring Analogue Outputs

Settings

You can configure the following settings for the Analogue Input Modules:

- **Process Safety Time**
  Use the procedure described earlier "Configure the I/O Module Process Safety Time" and either choose the default value or enter a custom value for the module.

- **Module Status Variables**
  Use the procedure described earlier in the topic "Wire a Status Variable to an I/O Module".
- **HART**
  Wire HART variables to the channels.
- **Channel Advance Settings**
  Set the Advance options.

Wire Output Channels

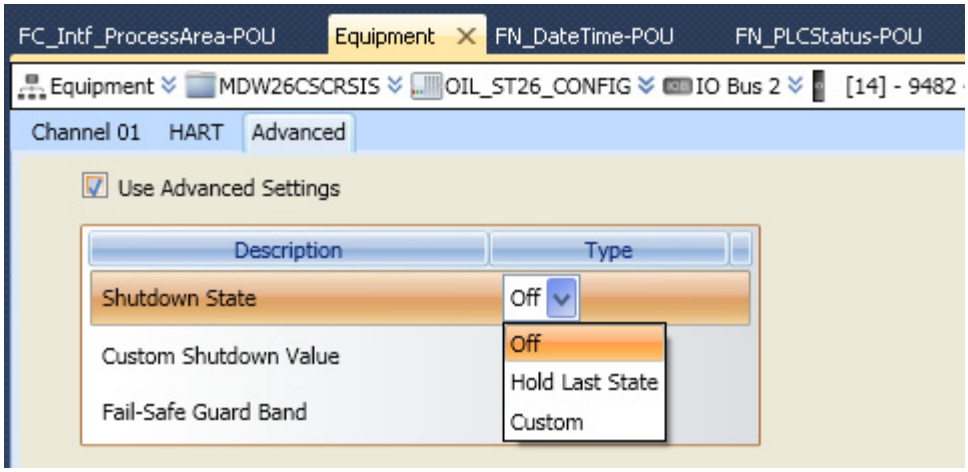
The controller writes its analogue outputs one time in each application cycle. The analogue output variables are also updated one time in each application cycle. You can wire analogue output channels to the following variable type and data structures:

- REAL (the <variable_name> Retrieves the commanded state)
- T9K_AO_Compact (provides three elements)
- T9K_AO_Full (seven elements).

The structures supply data about the output, such as line fault status and discrepancy status.

Configure the Channel Advanced Settings

You can configure both Advanced shutdown states for an analogue output channel. The shutdown state defines how the output behaves when its parent 9481 or 9482 analogue output module suffers a loss of communications with the processor or a failure in the module, and goes into its shutdown mode.



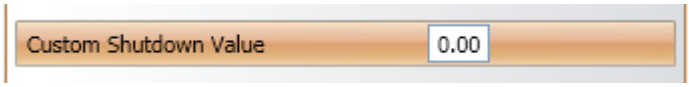
To configure the shutdown state do the following:

1. Select the output module in the Equipment View and select the necessary channel to configure.
2. Go to the **Equipment** tab on the content window and select the **Advanced tab**.
3. Select **Use Advanced Settings**.
4. Set the Shutdown State to one of the options in the drop-down list:
   - Off (default) disables the output when the module is shutdown.

- Hold Last State forces the output to stay in its most recent commanded state during a module shutdown.
- Custom forces the output to go to the value set in the custom shutdown value field during a module shutdown

WARNING: If you use Hold Last State for a safety-related output you must restore the communications in less than the MTTR or use other compensating measures during a module failure to maintain the safety rating of the system.

5. If you set the Shutdown State to Custom, set the custom shutdown value to a value of current using the same scaling as a command value (CV).



   - the default scaling factor is 0 to 100 %, representing 4 to 20 mA
   - the default value is 0.00, representing 4 mA.
6. Repeat this procedure for other output channels.

Analogue Output Advanced Channel Settings

Each output channel from the 9481/9482 analogue output module supports the set of control parameters shown in the following table:

Table 12 - Analogue Output Control Parameters

Description	Value(s)	Default	Remarks
Shutdown State	Off, Hold Last State, Custom	Off	'Off' de-energizes the output during a shutdown 'Hold Last State' forces the output to stay in its most recent commanded current during a shutdown 'Custom' forces the output to go to the value set in 'Custom Shutdown Value' during a shutdown
Custom Shutdown Value	As scaled range of command value (CV)	-25	Output set when 'Shutdown State' is set to 'Custom', during a shutdown
Fail-Safe Guard Band	0-100 % of full scale 20 mA (1 % = 0.2 mA)	1 %	Threshold for discrepancy alarm between command value (CV) and Count (CNT). A discrepancy alarm is reported on parameter DIS.

T9K_AI_COMPACT and T9K_AI_FULL

The data structures for analogue outputs (T9K_AI_COMPACT and T9K_AI_FULL) have the elements shown in the following tables. Use only the FULL structure for WN.

Table 13 - T9K_AI_COMPACT Structure for Analogue Outputs

Identifier	Type	Description	Remarks
<tagname>.CV	REAL	Command Value	Demanded current. A scaled, floating-point value showing the analogue loop current. Default scaling factor is 0 to 100 % representing 4 to 20 mA
<tagname>.LF	BOOL	Line fault	TRUE: no field supply is present, no load is connected, the commanded output current cannot be met, the wiring polarity is reversed, or channel fault (states 1,3,5,6,7)
<tagname>.DIS	BOOL	Discrepancy	TRUE: measured current and commanded current differ by more than the Fail-safe Guard Band

Table 14 - T9K_AI_FULL Structure for Analogue Outputs

Identifier	Type	Description	Remarks
<tagname>.CV	REAL	Command Value	Demanded current. A scaled, floating-point value showing the analogue loop current. Default scaling factor is 0 to 100 % representing 4 to 20 mA
<tagname>.LF	BOOL	Line fault	TRUE: no field supply is present, no load is connected, the commanded output current cannot be met, the wiring polarity is reversed, or channel fault (states 1,3,5,6,7)
<tagname>.DIS	BOOL	Discrepancy	TRUE: measured current and commanded current differ by more than the Fail-safe Guard Band
<tagname>.CF	BOOL	Channel fault	TRUE: module diagnostics detect a fault in the channel electronics or firmware (state = 7)
<tagname>.V	INT	Voltage	Reports the channel voltage at the output terminals, in units of millivolts and with an accuracy of ± 500 mV.
<tagname>.CNT	INT	Raw count	Reports the current for the channel in raw units scaled 256 per mA (from 0 mA = 0 to 24 mA = 6,144)
<tagname>.STA	USINT	Channel state	Reports a state value for the channel: • 1 = no vfield (< 18 Vdc) • 2 = off (demand < 0.4 mA) • 3 = no load/open circuit • 4 = on (demand > 0.4 mA) • 5 = compliance fault (demand cannot be met) • 6 = reverse polarity (< -1 Vdc) • 7 = faulted

The State Variable for Analogue Outputs

The state variable for an analogue output is an unsigned integer with a value from 1 to 7 showing the following:

- 1 = no vfield: the field supply voltage is at or below 18 V dc and the commanded current is less than 0.4 mA for that channel.

IMPORTANT When the state variable is 1, the field voltage (<tagname.V>) is shown as 0 mV.

- 2 = off: the raw count value is less than 102 (0.4 mA).
- 3 = no load: the controller cannot identify a load connected to the channel field wiring, or the loop voltage cannot be detected. This happens when the commanded current is more than 0.4 mA, the raw count value is less than 51 (0.2 mA) and the measured voltage is less than 1000 mV.
- 4 = on: the raw count value is 102 or more (>=0.4 mA).
- 5 = compliance fault: there is not sufficient loop voltage available to sustain the commanded output current to within the safety accuracy specification (i.e. 1 % full scale or 0.2 mA) and the commanded current is

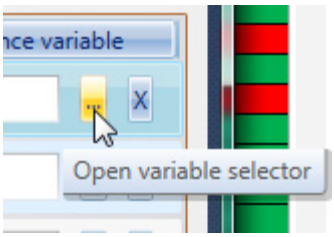
more than 102 Counts (0.4 mA). The output supplies as much current as is available and does not start a fail-safe action or report a channel fault.

- 6 = reverse polarity: the measured voltage is less than -1000 mV.
- 7 = faulted.

Wire a Variable to an Analogue Output Channel

You wire a variable to an analogue output channel to let the application use the output. Do the following:

1. Select the output module in the Equipment View and then select the channel that you want to wire to a variable.
2. Go to the Equipment window and select the tab with the name of the channel (for example 'Channel 01').
3. Click the button adjacent to the Channel Variable field.



   - The Variable Selector window is displayed.
4. Select the variable from the list, then click OK.
5. Repeat this procedure for other outlet channels.

Faulted State for Analogue Outputs

An analogue output channel is faulted when normal operation or diagnostic tests find a known fault condition.

When a channel is faulted, the state reports the value 7, and the other variables report the following 'safe' values:

- Line Fault = TRUE
- Discrepancy = TRUE
- Channel Fault = TRUE
- Voltage = 0 mV
- Raw Count = 0 mA.

Sharing Data between ControlLogix and AADvance

The AADvance® system can share non-safety data with the Allen-Bradley® family of ControlLogix® controllers programmed by RSLogix 5000® software. This chapter describes how to configure this data sharing feature between AADvance and ControlLogix using CIP over Ethernet.

WARNING: Do not use CIP over EtherNet/IP for safety-related data. You must use the SNCP network for safety-related data shared between AADvance controllers.

Using AADvance for CIP

The Common Industrial Protocol (CIP) over Ethernet /IP protocol enables AADvance to exchange data with ControlLogix controllers programmed by RSLogix 5000. The exchange of data uses the produce/consume tag method currently used for sharing data between RSLogix™ controllers.

AADvance supports produce and consume communications to redundancy systems. This support for produce/consume variables is non-interfering, a failure of the Ethernet/IP stack does not interfere with the safe operation of an AADvance system.

To use AADvance for CIP over Ethernet:

- Define a produce link which allows you to specify both the AADvance controller variables to be produced and the AADvance Ethernet ports that will accept consume connections for those variables.
- Define multiple consumer links each of which allows the identification of a Logix controller path to a Logix controller that is producing variable(s) of interest and individual consumer variables (each of which identifies the produced variable it is to be associated with).

Connect the AADvance system to the ControlLogix system.

At runtime AADvance with a producer variable sends values to the remote controller. Whereas the AADvance consumer variable pulls values from the remote controller.

The following applies for using AADvance for CIP over Ethernet/IP:

- You can only use the CIP network to exchange data using producers and consumers, you cannot use the CIP network for downloading to a target or monitoring a target.
- Do not use CIP for exchanging data between AADvance controllers as part of a safety network. Use the SNCP Network for safety data communication.

- You can configure a variable as a consumer or a producer but not both.
- Only Global variables are available for use with controllers supporting a connection to a CIP network.

NOTE When connecting to a controller not supporting CIP to a CIP network, the workbench displays a warning and does not implement the connection.

The default IP address for a connection between a controller and a CIP network is blank.

When a valid IP address is not specified, the Workbench reports an error when building the network- "Network CIP: invalid connection properties for xxx network".

Connection Requirements of a Produced or Consumed Tag

Produced and consumed tags, all require connections. Each connection uses memory and processing resources. The demands are as follows:

Type of Tag	Number of Connections Used
Produced Tag	number of consumers + 1
Consumed Tag	1

Example:

Connection Requirements of a Produced or Consumed Tag:

- An AADvance controller producing 4 tags for 1 controller uses 8 connections
- Each tag uses 2 connections (1 consumer + 1 = 2)
- 2 connections per tag x 4 tags = 8 connections
- Consuming 4 tags from a controller uses 4 connections (1 connection per tag x 4 tags = 4 connections).

The number of connections rather than size of the data transferred over the connection is the significant factor.

It is far more efficient to include multiple items in a single connection than have individual connections for each one. (e.g. send an array of 10 DINTs rather than 10 individual DINTs)

An individual AADvance processor slice has a finite capability to process the amount of communications traffic that passes through its associated Ethernet ports.

Memory available to CIP is available per controller rather than per slice. (see memory requirements below)

A connection defines a single Producer or Consumer that can be as small as a DINT or as large as a structure up to 500 bytes.

A consumer or producer is associated with a specific AADvance Ethernet interface. This association is formed by virtue of the network path that a consumer is using to access a producer or the network interface through which a consume connection from another controller arrives.

Each AADvance interface, when used for CIP Produce/consume, must be assigned a unique subnet.

The number of consumer connections per slice is **C**

The number of producer connections per slice is **P**

'Load' **L** Per Slice = (**C**+2**P**)

L must be <= 32

RPIs can be calculated as a function of **L**

L <= 16 Minimum RPI for any producer or consumer is 200 ms

16 < **L** < 24 Minimum RPI for any producer or consumer is 300 ms

L > 24 Minimum RPI for any producer or consumer is 500 ms

Memory Considerations.

CIP Produce/Consume variables utilize memory from a pool that is shared with other communications features of the AADvance controller. Specifically:-

- MODBUS
- Sequence Of Events
- Variable Bindings

The total amount of space available is 131072 bytes

A Produced variable of 120 DINTS will consume 1552 bytes of this space

A Consumed variable of 120 DINTS will consume 1120 bytes of this space.

NOTE This memory is used by every slice e.g. 48 producers (each slice producing the maximum of 16 producers) will consume 48 * 1152 = 74496 bytes.

Data Types for CIP

You can use the following IEC 61131-3 data types (or a compound type constructed from these) as producers or consumers with CIP over Ethernet/IP, as long as the data type is of size DINT (4 bytes / 32 bits) or larger (maximum is 500 bytes):-

- BOOL
- SINT
- INT
- DINT
- LINT
- REAL
- LREAL.

You can use array or structure variables, but not the member elements of an array or structure. There are some limitations:

- The CIP tab is available for array or structure variables, but not for the member elements of an array or structure.
- You cannot use the STRING data type with CIP over Ethernet/IP in an AADvance system.

To share data types smaller than four bytes, you have to declare a user-defined data type and pad it accordingly. This is like the restriction defined in RSLogix 5000. The instruction for sending items smaller than four bytes states that you must create a user-defined data type which is padded to four bytes. The last member of the data type of the variable being produced/consumed must be either a structure or a 4 byte type, e.g. DINT, REAL. Arrays must be 4 byte aligned; aligned either, by placing them inside a structure type or by ensuring they are preceded by a 4 byte type. Structures are always 4 byte aligned unless they contain LREAL/LINT values which must be 8 byte aligned. The size of a structure containing LINT/LREAL members must also be a multiple of 8 bytes.

Setting Up Variables for CIP

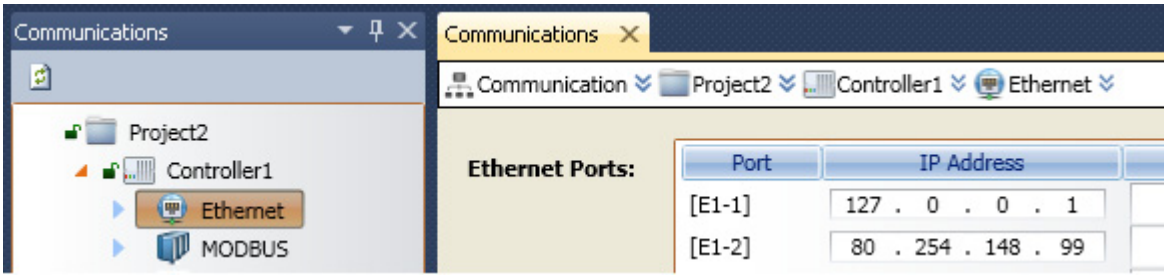
You can declare the variables to use with CIP over Ethernet/IP before you give them CIP producer and consumer links, or as you need them when you set up the individual links:

- Use the VAR window for the controller (found in the content window) to declare variables before giving them CIP links.
- Use the Variable Selector window to declare variables when you give them CIP links.

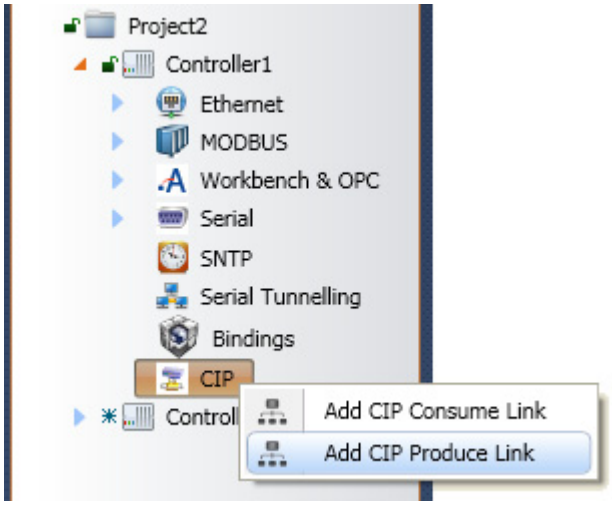
In both cases you have to first create a CIP link (producer or consumer) and then use the Communications window for your link to supply variables to the link. You also use the Communications window to give extra properties (which are only related to CIP over Ethernet/IP) to the variables. You can give a variable a producer link or a consumer link, but not both.

Define the CIP Producer Link

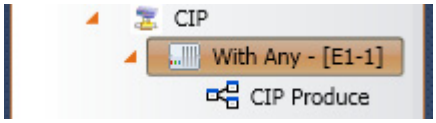
An AADvance controller can have one CIP producer link. To define the CIP producer link:



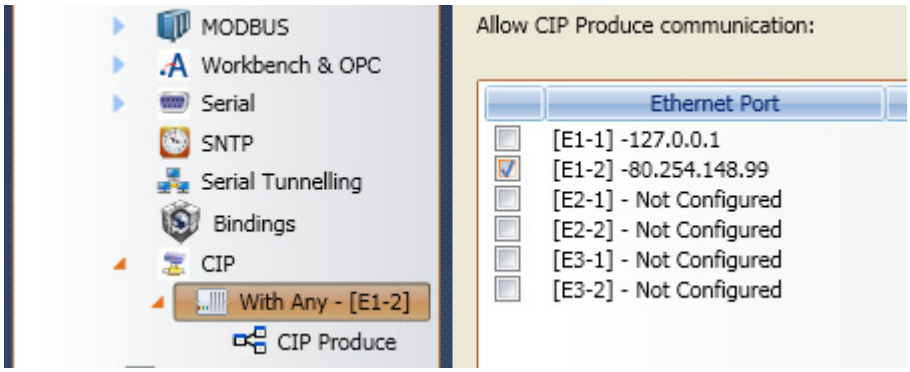
1. Go to the Communications view and select the **Ethernet** node.
2. Add the IP address on which the controller offers its producer services to a spare Ethernet port. This example uses 80.254.148.99 on port E1-2.
3. Staying in the Communications view, select the **CIP** node for the controller.



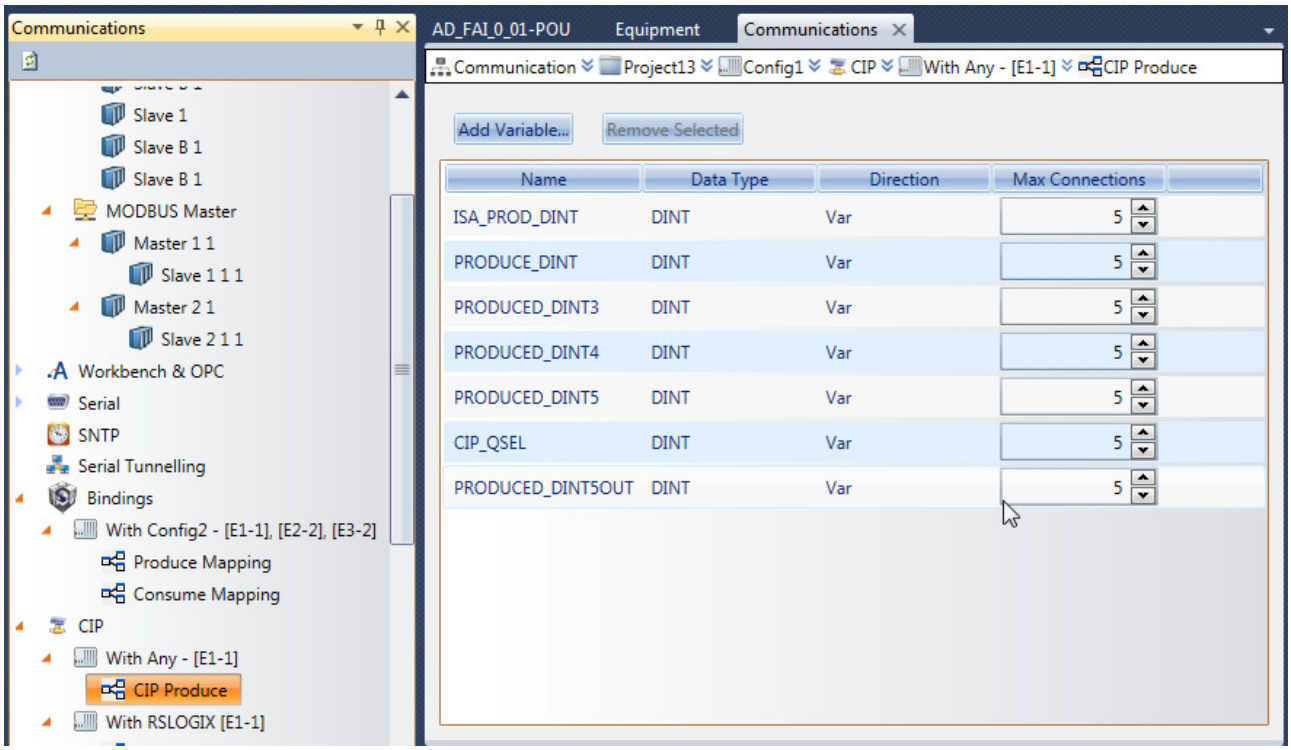
4. Open the contextual menu and select **Add CIP Produce Link**.
   - The AADvance Workbench adds a node named **With Any** below the CIP node.
   - The AADvance Workbench adds a node named CIP Produce below:



5. Expand the CIP node to see the two nodes:
   - the node labeled **With Any** holds the communications settings
   - the node labeled CIP Produce holds the producer variables.



6. Go to the **With Any** node and use the communications window to select the Ethernet Port chosen at step 2. In this example, this is port E1-2.
7. Select **Add Variable** to add a CIP Produce Variable. You can also drag a Producer Variable from the VAR (dictionary) into the Produce list.



8. Add the producer variables.

   The Producer variable parameters are:

   - **Name**: Variable name from the variables list
   - **Data Type**: DINT
   - **Direction**: for I/O wiring indicates whether a variable in an input or output or internal.
   - **Max Connections**: Values 1 to 10; Default = 5; the maximum number of simultaneous consumers of particular variable.

NOTE The AADvance controller will accept more connections than the values defined in the Connection Requirements of a Produced or Consumed Tag on page 154 - however exceeding the limits defined may result in link instability especially with low RPI values.

CIP and its Producer and Consumer Variables

You can set up one set of CIP producer variables for an AADvance controller, and one or more sets of CIP consumer variables. Each set of CIP consumer variables is related to its own ControlLogix controller.

The AADvance controller sets its consumer variables to the most recent received value at the start of its application scan before executing the logic. The controller updates its producer variables at the end of its application scan after executing the logic. The AADvance controller uses the most recent value of a producer variable when sending it to a consumer. Produced values are sent at the Requested Packet Interval (RPI) expected by the ControlLogix controller.

You cannot define a default value for a consumer variable. If the connection fails (typically because the communications link fails), the most recently received value of the consumer variable is retained.

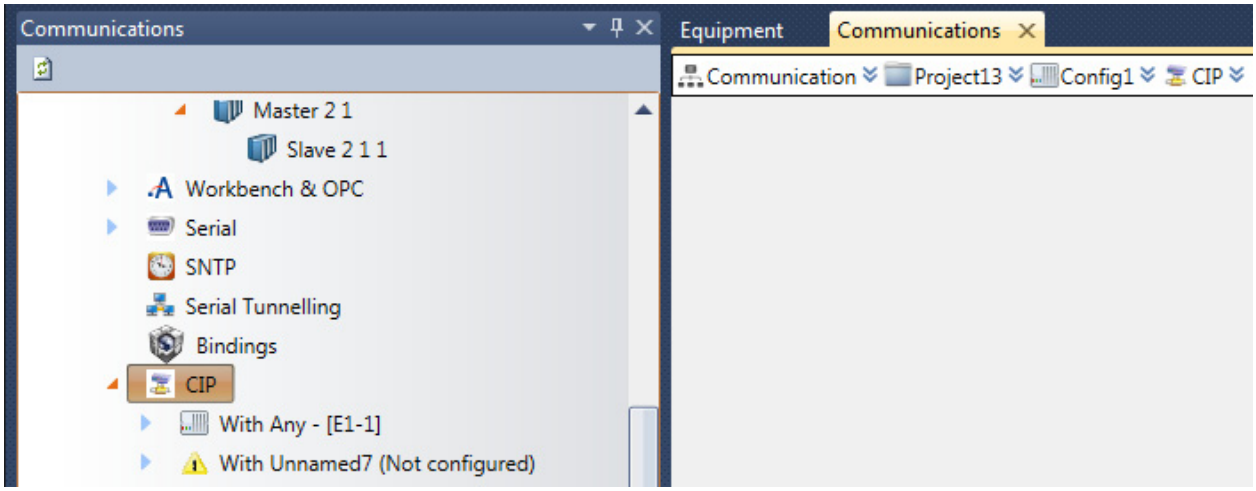
The maximum size of a CIP variable is 500 bytes. If the variable is a structure having a mixture of element types, then each element starts on a new byte or word depending on its size. For example a DINT will start on a new 4 byte boundary even though the previous element may be a single bit BOOL.

IMPORTANT Use the SNCP network to share EtherNet/IP data between AADvance controllers: do not use CIP to share data between controllers.

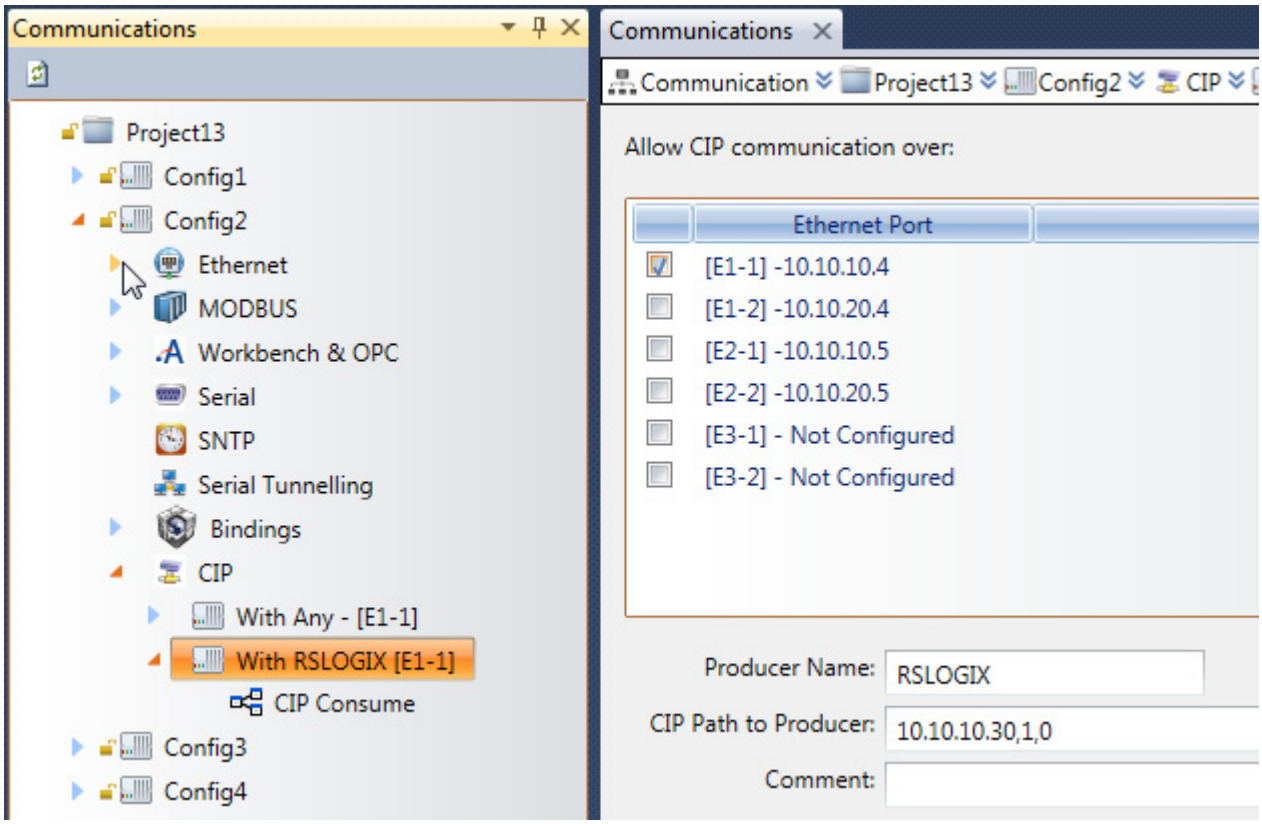
Define a CIP Consumer Link

When you configure a variable as a consumer for CIP over Ethernet/IP, you have to identify the producing ControlLogix controller and identify the name of the production variable it is associated with (the remote tag name) and the requested packet interval.

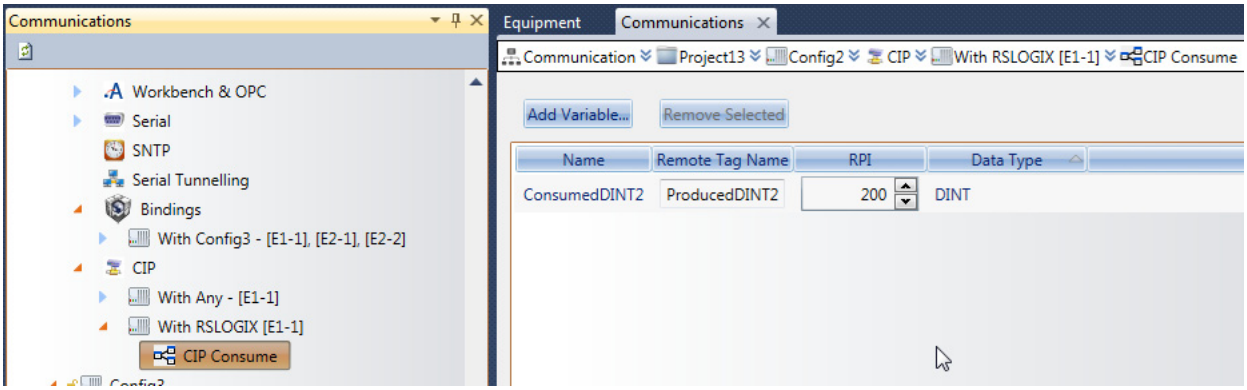
NOTE The AADvance controller can support the number of consumer connections defined in Connection Requirements of a Produced or Consumed Tag on page 154 but will allow configuration of up to 128 consumers. Exceeding the defined limits may result in link instability especially with low RPI values.



1. Select the Communications View, then the **CIP** node for the controller.
2. Right-click on the **With Unnamed (Not Configured)** Node.



3. Select an **Ethernet Port** for CIP communication.
4. Enter the Producer Name: (name of the Producer device, i.e., RSLOGIX
5. Enter the valid Ethernet address of the RSLOGIX controller in the CIP Path to Producer field.
   - This will be of the form x.x.x.x,n1,n2 where: x.x.x.x = IP address, n1 = the Slot id, n2 = port number.
6. Enter a comment if required: this is a text string.
7. Select **Add Variable** to add a CIP Consume variable. You can also drag a Consume variable from the Controller VAR (dictionary) into the Consume list.



The CIP Consumer variable parameters are:

- **Name**: Variable name from variables list (non-editable field)
- **Remote tag Name**: Name of the Producer variable in Logix controller being consumed from

- **RPI**: Requested Packet Interval; defines the frequency at which the Logix controller offers the variable to the AADvance controller. The RPI should not be set below 200 ms for the current release
- **Data Type**: DINT (non-editable field).

With Status Produce and Consume Variables

AADvance firmware 1.4 and later supports 'With Status' produce and consume variables.

Obtaining the Connection Status for a Consumed Variable

Your application can obtain the connection status for a consumed variable. To do this, make sure that the variable is a member of a structure data type that has two elements: a first element of the type CONNECTION_STATUS and a second element to match the data type of the variable.

IMPORTANT Produce and Consume with status tags are not supported prior to AADvance firmware R1.4

CONNECTION_STATUS is a pre-defined type.

The CONNECTION_STATUS type is itself a structure comprising two BOOL elements, RunMode and ConnectionFaulted:

- When the structure data type for a variable includes CONNECTION_STATUS as its first element then these fields will be automatically filled in by the AADvance controller, ready for your application to read them.
- The RunMode value of a consumed variable with status indicates the Run Mode of the Logix controller producing the value being consumed.
- The Connection Faulted value of a consumed with status variable will be TRUE if the connection is not working correctly or FALSE if it is.

If the connection fails, the value of the consumption variable remains at its last received value; you cannot define an 'error' value to revert to if the communications link fails.

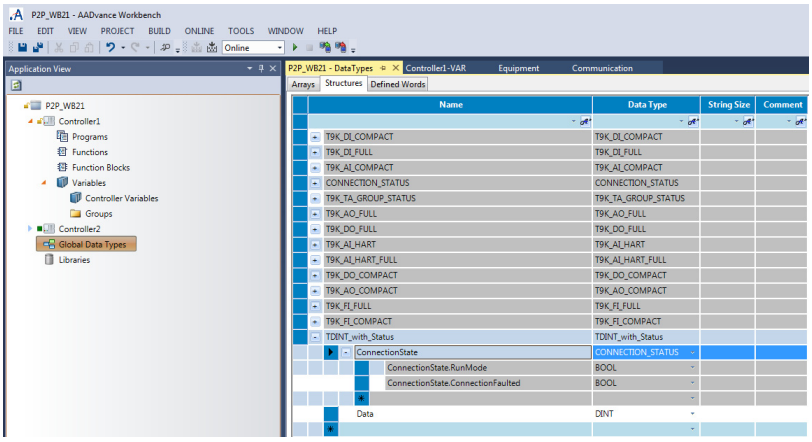
Example

As an example, consider the consumer of a remote variable named 'T' that is a DINT. The variable will be named 'T' to match.

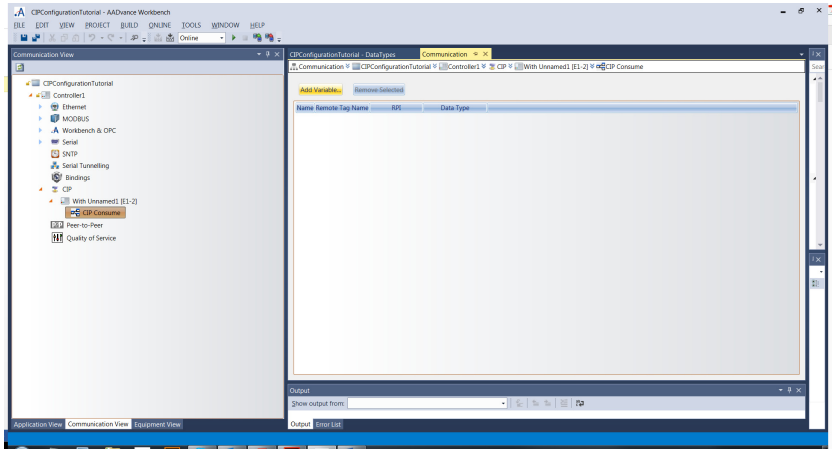
You have to define a structure data type of the form:

- CONNECTION_STATUS (the connection status information)
- DINT (the value of the data)

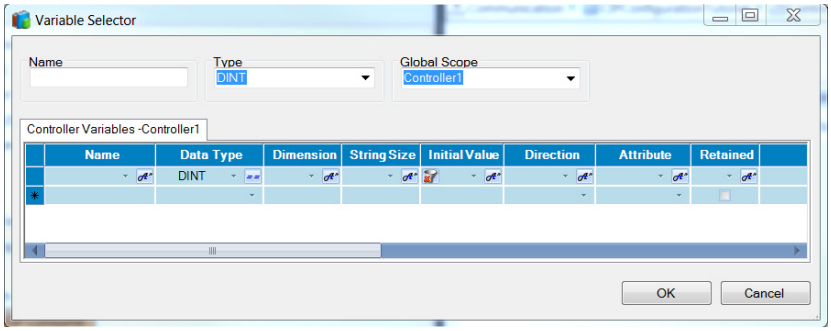
Give the structure data type a name, suitable for re-use within the project, such as TDINT. But first we must define the structure, which for this example we will call, "TDINT_with_Status".



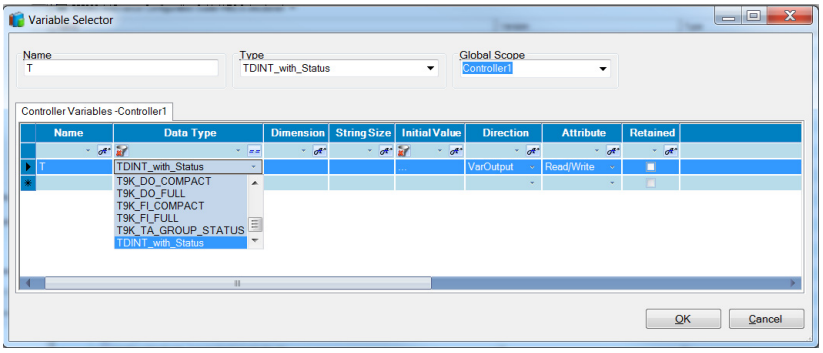
Click on the **CIP Consume** node to open the following dialog box.



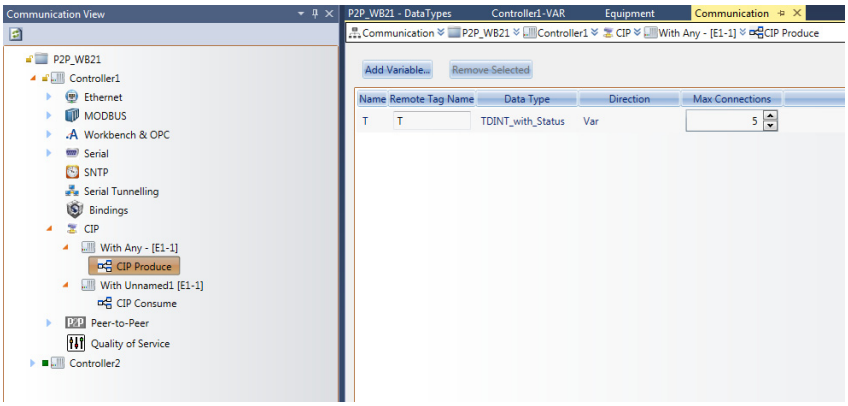
Click on the **Add Variable** button to open the Variable Selector



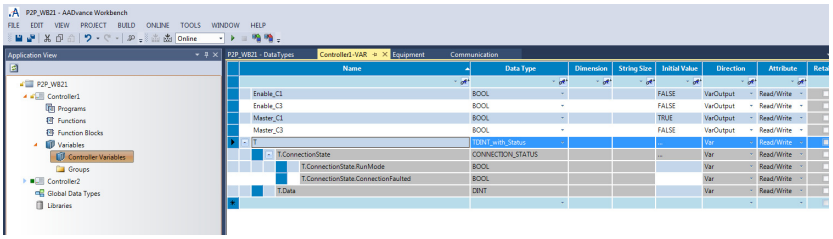
When you create the variable 'T' in the Dictionary, declare its type to be TDINT_with_Status:



The Remote Tag 'T' now has Data Type "TDINT_with_Status".



You can now examine the structure of 'T' within the controller Variables (Previously referred to as "The Dictionary"):



The AADvance controller sets the values of the elements T.ConnectionState.RunMode and T.ConnectionState. ConnectionFaulted based upon the run mode of the remote Logix controller and the presence of a communications fault.

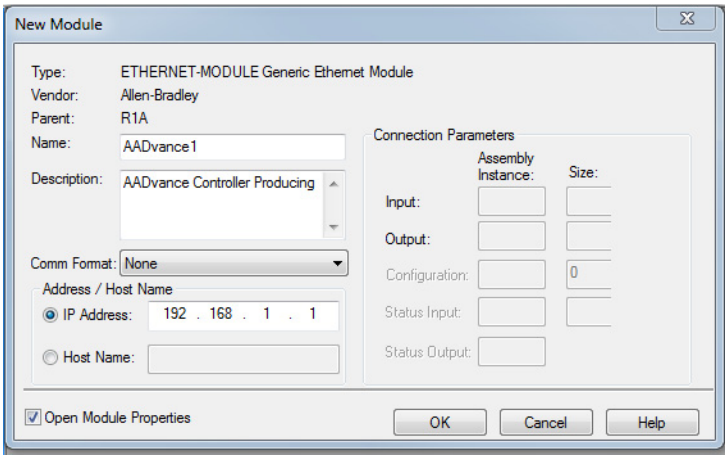
AADvance Producer RunMode

The RunMode of a produced with status variable can be set by the user's application to True/False. AADvance can only produce/consume when applications are running so this flag can be used to indicate some boolean condition in the AADvance controller application e.g. is the security dongle fitted.

IMPORTANT	When using 'with status' connections the **name** of the variable's datatype and the **structure** that type defines **must be identical** in both controllers.

About the RSLogix 5000 Configuration

Within the RSLogix configuration it is necessary to configure the AADvance controller as a CIP provider. This is achieved by adding a Generic Ethernet module configured with the IP address of the AADvance controller. Input/ Output sizes shall be set to zero and the Comm Format set to None.



IMPORTANT When the ControlLogix architecture has redundant processors, the ControlLogix Ethernet card handling CIP has to be on each of the local racks (where the processors are located) and not in the remote racks. The cards must be set up with the same IP, subnet mask, link speed etc.

Set the RSLogix UNICAST Configuration

The AADvance controller uses the Unicast mode for CIP You will need to configure the Unicast option in the RSLogix CLX configuration.

Produced Variable

1. Select a CLX configuration Produced CIP variable.
2. Check the Allow Unicast Consumer Connections option.

Consumed Variable

For a Consumed Variable select a Consumed CIP variable.

1. Set the RPI value to 500 ms (the default value is 20.0 ms).
2. Check the Use Unicast Connection over Ethernet/IP option.

Configuring MODBUS

The AADvance® controller supports MODBUS and can be set up to operate as a MODBUS Master or a MODBUS Slave element. This chapter describes the procedures to set up and configure the AADvance controller for MODBUS communication.

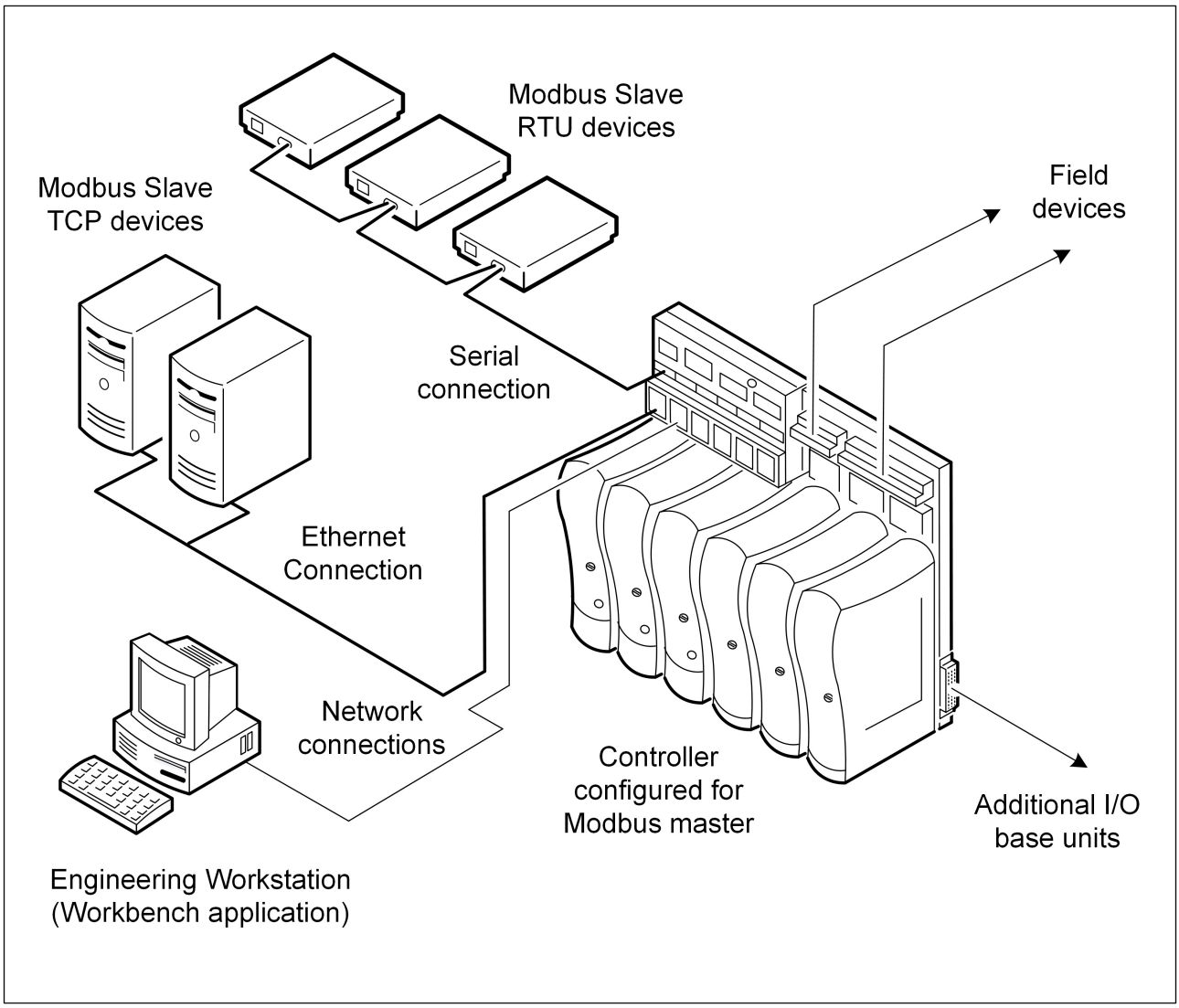
MODBUS Standards

The MODBUS functionality implemented by the AADvance controller meets the following standard:

- MODBUS Application Protocol Specification, version 1.1b. December 2006. The MODBUS Organization.

MODBUS Master Hardware and Physical Connections

The MODBUS Master functionality is built into the 9110 Processor Module. The physical communication ports are located on the 9100 Processor Base Unit. You do not need to add any other hardware to the AADvance controller apart from other components to make the physical connections to the processor base unit. The following illustration shows some possible arrangements for MODBUS Master connections.



The MODBUS RTU slave devices are connected to one or more of the serial ports on the controller; a usual arrangement uses a multi-drop (RS-485) arrangement. The engineering workstation and the MODBUS TCP devices are shown connected to the Ethernet ports on different networks. Alternatively, these devices can be combined onto one network. Refer to the AADvance System Build Manual for more details about physical connections.

Planning for MODBUS Master

To use the AADvance controller as a MODBUS Master, you have to complete three activities:

- Define the physical connections from the controller to the slave devices.
- Configure the characteristics for the serial ports (MODBUS RTU only).
- Set up the project for MODBUS Master, and configure the application.

Each of these activities is independent. Therefore, you can execute these in any order, but you have to complete all three activities.

MODBUS Master Command Set

The AADvance MODBUS Master supports the subset of MODBUS commands listed in the following table.

Table 15 - AADvance MODBUS Master Commands

Command Code	Command	Remarks
01	Read coil status	Maximum 512 coils for each message
02	Read input status	Maximum 512 inputs for each message
03	Read holding registers	Maximum 125 registers for each message
04	Read input registers	Maximum 125 registers for each message
05	Force single coil	
06	Preset single register	
15	Write multiple coils	Maximum 512 coils for each message
16	Write holding registers	Maximum 123 registers for each message
08	Diagnostics	Sub-function 00, query used (RTU only)

MODBUS Data Types and Addressing

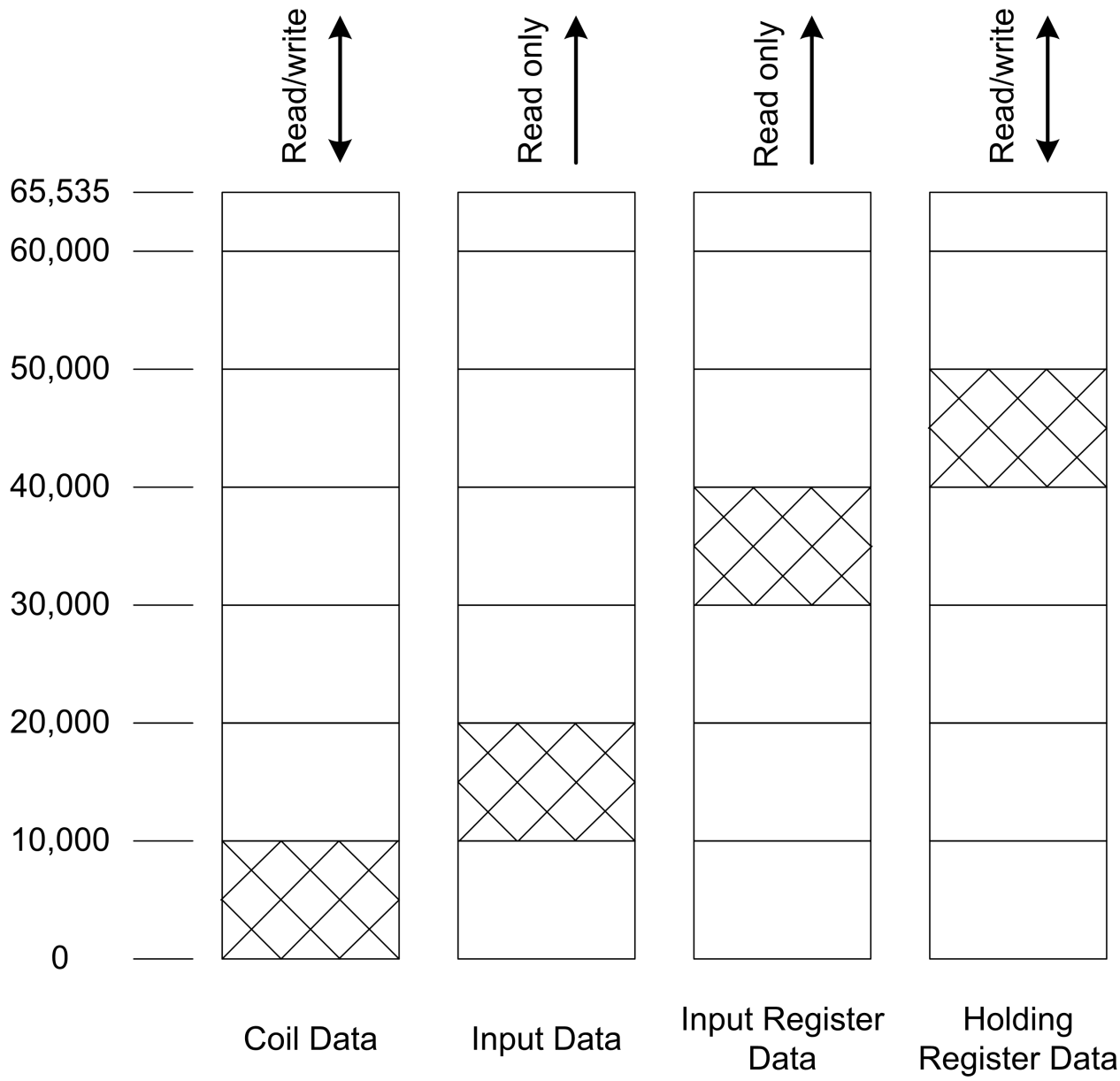
The MODBUS Master supports the MODBUS data types listed in the table.

Table 16 - MODBUS Data Types

Data Type	Address Range
Coils	1 to 65,535
Inputs	1 to 65,535
Input registers	1 to 65,535
Holding registers	1 to 65,535

MODBUS uses a numeric addressing scheme to move data between devices. AADvance controllers provide a dedicated area for each of the four MODBUS data types and follow the model specified in the MODBUS Application Protocol Specification.

The original MODBUS standard defines the address field as a four-digit field with a prefix relating to the data type. The areas shown hatched in the following illustration show how original-style, five-digit MODBUS addresses (for example, a holding register at 40,001) are related to the AADvance memory map.



The addresses used for MODBUS data transfer listings start at one. The first AADvance Workbench variable network address is 1 and the first coil is 00001. An address of 0 for either field is illegal.

MODBUS Message Scheduling

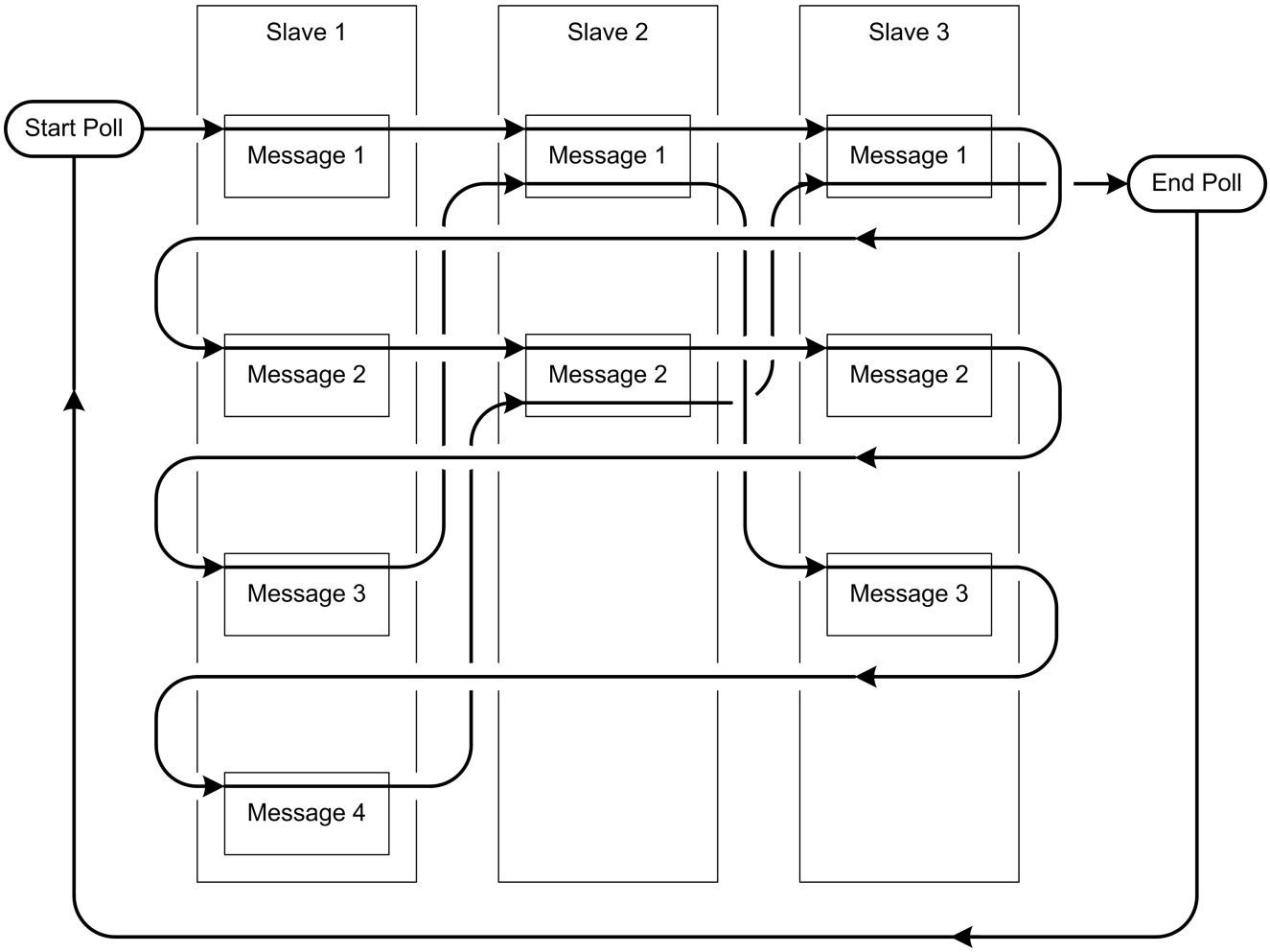
In operation, all of the MODBUS Master objects specified for a project (MODBUS RTU or MODBUS TCP) function independently of each other. Each master polls its related slave devices to send the messages scheduled for the slaves.

For MODBUS RTU, a master object polls its slave devices one at a time, executing one message for each slave in turn. Different slaves attached to the same master will usually need different numbers of messages. Therefore, the master proceeds to work through the messages listed for the slaves, one message for each pass, returning to the first message for a slave after it has sent the last one. This means that some messages (for slaves having fewer

scheduled messages) are sent more than once during a polling cycle. The polling cycle completes when the MODBUS Master has performed an equal number of polls to each slave and has sent every message. The cycle then starts again.

Example

The illustration shows the message scheduling for a master having three slaves needing four, two and three messages.



The total number of messages that the master has to send in a complete polling cycle is equal to the largest number of messages scheduled for any one slave, multiplied by the total number of slaves. In the following example the total number of messages is 4 ? 3 = 12. Nine messages are specified for the three slaves. The message scheduling mechanism sends three of these messages twice making a total of twelve messages. The sequence of messages for the serial communications link in this example is listed in the table.

Table 17 - Message Sequence for Example MODBUS Master

Sequence Number	Slave	Message	Remarks
1	Slave 1	Message 1	
2	Slave 2	Message 1	
3	Slave 3	Message 1	End of first pass through all slaves

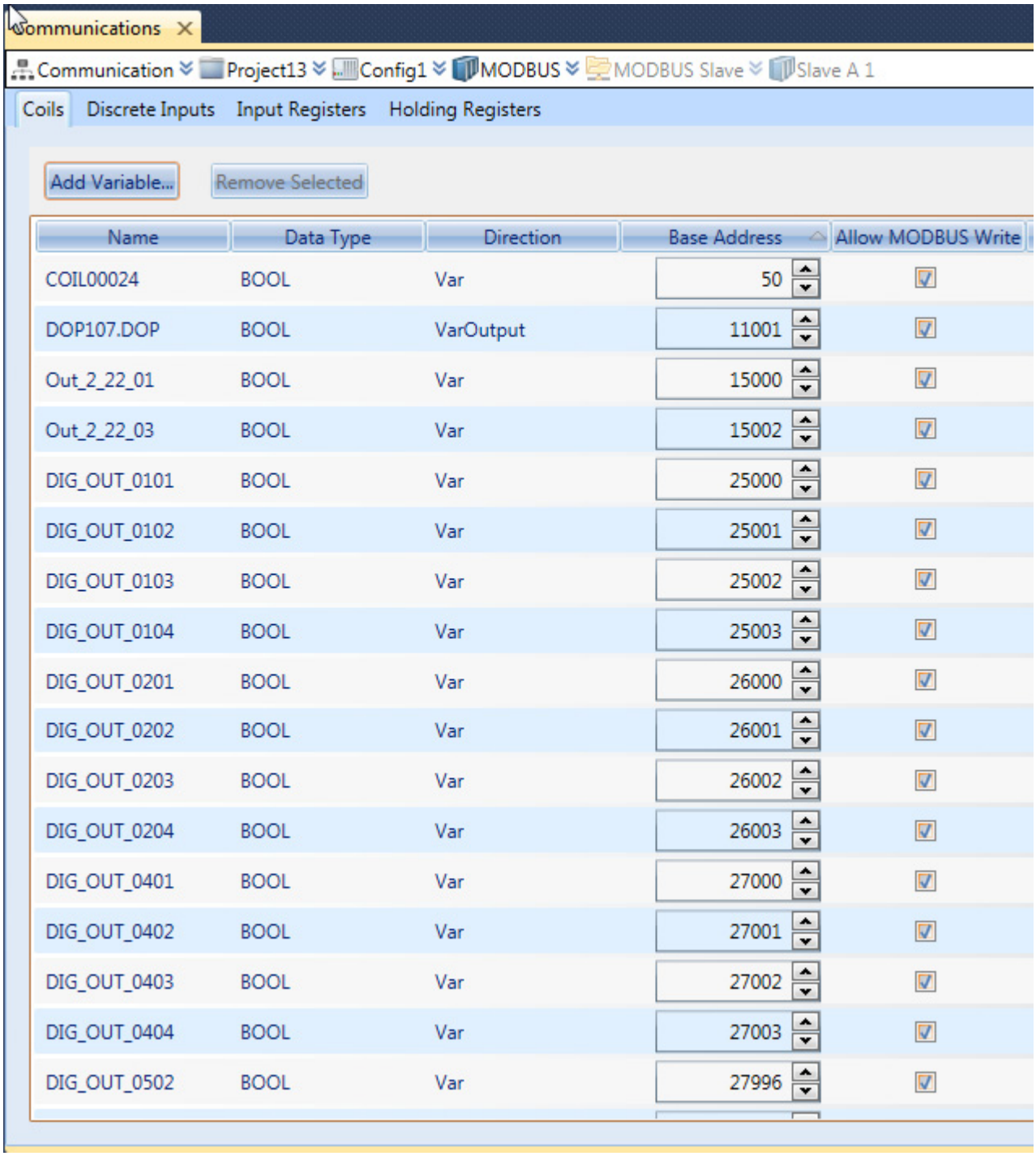
Table 17 - Message Sequence for Example MODBUS Master

Sequence Number	Slave	Message	Remarks
4	Slave 1	Message 2	
5	Slave 2	Message 2	End of pass through slave 2
6	Slave 3	Message 2	
7	Slave 1	Message 3	
8	Slave 2	Message 1	Repeat message
9	Slave 3	Message 3	End of pass through slave 3
10	Slave 1	Message 4	End of pass through slave 1
11	Slave 2	Message 2	Repeat message
12	Slave 3	Message 1	Repeat message; end of poll

Configure MODBUS Addresses

To configure MODBUS Addresses:

1. Add MODBUS addresses to an existing variable in the dictionary of WB 2.0 or define a new variable which will be saved automatically in the dictionary of WB 2.0.
2. Select the Communications view and select a MODBUS icon below a controller:
   - The MODBUS variable parameter screen is displayed.
3. Select the **Add Variable** button.
   - The Variable Selector screen is displayed.
4. Select the required variable and click **OK**.
5. To allow MODBUS to write to a variable click on the box **Allow MODBUS Write** in the row showing the variable.



Physical Connections for MODBUS RTU

MODBUS Master RTU communications use the serial ports on the 9100 processor base unit. The following fundamental topologies are possible:

- Point to point
- Point to multi-point (multi-drop).

Each of these topologies can use a full-duplex, 4-wire connection or a half-duplex, 2-wire connection. For details refer to the AADvance System Build Manual.

Physical Connections for MODBUS TCP

MODBUS Master TCP communications use the Ethernet ports on the 9100 processor base unit.

MODBUS Names

The MODBUS object Name identifies a MODBUS Master object or slave link in the AADvance Workbench, and is included in printed reports. However, the application does not use the MODBUS object name. We recommend that you use a different name for each object. Alternatively, accept the default name.

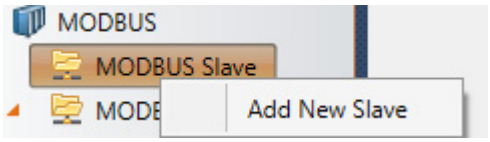
Create a MODBUS Slave Object

The AADvance controller can operate as a MODBUS Slave, supporting up to five MODBUS Slaves on each processor module. This provides a capacity of fifteen MODBUS Slaves for a controller having three processor modules.

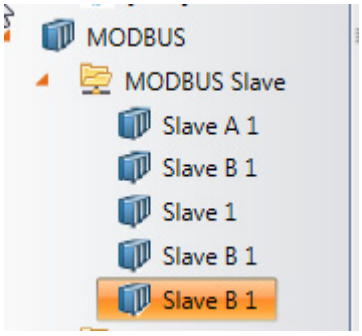
NOTE As a MODBUS Slave device, the controller only transmits data if necessary according to an indication from a MODBUS Master. It does not communicate with other slaves.

Create a MODBUS Slave object under a Controller MODBUS tab.

1. Select the Communications view, then choose the Controller that will support the MODBUS Master object.
   - This is the module related to the Serial port or the Ethernet port that you intend to use.
2. Right-click on a **MODBUS Slave** node, then choose Add New Slave.



   - The system adds a MODBUS Slave object.



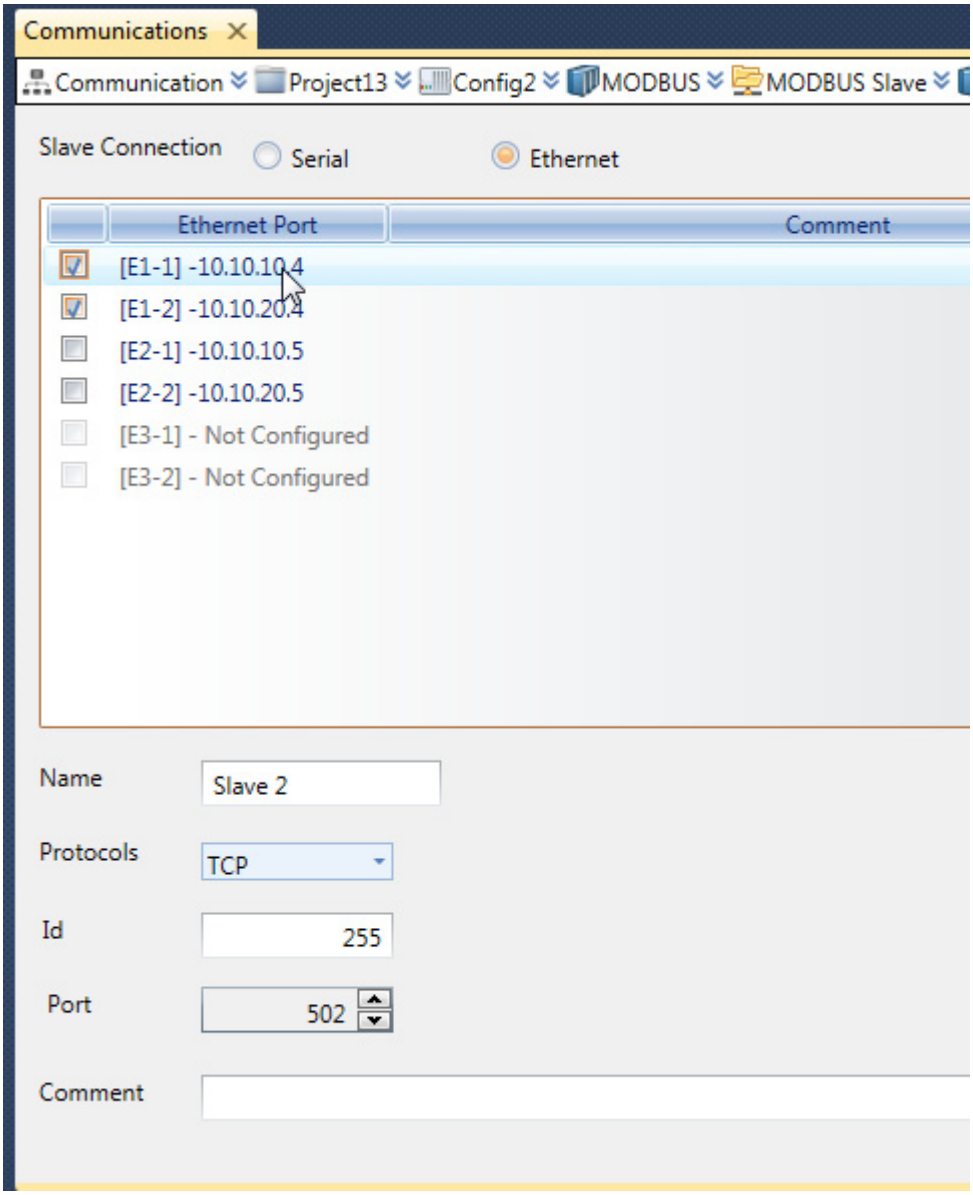
   - You can add up to 5 Slave objects under a MODBUS Slave icon.
3. Rename the Slave object as required by right-clicking and choosing the Rename option.
4. Select the Slave object to display the Slave parameter screen.

Configure a MODBUS Slave Object

Configure a MODBUS Slave object.

Ethernet Slave Connection

1. Right-click the Slave object and select Open.



2. Select the Serial or Ethernet options for the Slave Connection.
3. Enter the parameters.

MODBUS Slave Parameters

Each Slave has a choice of Slave Connections - Serial or Ethernet.

Table 18 - Serial Connection Parameters

Description	Value(s)	Default	Remarks
Slave Connection	Serial	Not Applicable	The Serial option automatically selects the RTU protocol.
Serial Port	S1-1, S1-2; S2-1, S2-2; S3-1, S3-2		Selections for the Serial ports on the three processor modules.
Comment	String	empty	Not used.
Name	String	Slave <n>	Enter a suitable unique name for the object.

Table 18 - Serial Connection Parameters

Description	Value(s)	Default	Remarks
Protocols	RTU	RTU	Serial communication protocol.
Id	1 to 247	1	Represents the Slave identity for a serial connection.
Port	Not used		Ethernet connections only.
Comment	string	empty	Enter a comment string as required.

Table 19 - Ethernet Connection Parameters

Description	Value(s)	Default	Remarks
Slave Connection	Ethernet	Not applicable	The Ethernet option selects the Ethernet ports and protocols.
Serial Port	E1-1, E1-2; E2-1, E2-2; E3-1, E3-2		Selections for the Ethernet ports on the three processor modules.
Comment	String	empty	Not used.
Name	String	Slave <n>	Enter a suitable unique name for the object.
Protocols	RTU, TCP	TCP	Select the required communication protocol.
Id - TCP	255	255	Represents the slave identity.
Port - RTU		2000	Ethernet port default value is always 2000 for RTU protocol.
Port - TCP		502	Ethernet port default value is always 502 for TCP protocol.
Comment	string	empty	Enter a comment string as required.

MODBUS Slave Exception Codes

When the AADvance controller operates as a MODBUS Slave, the controller can report the following exception codes through the application:

Code	Name	Comments
Code 01	Illegal Function	The function code received in the query is not a permitted action for the slave. If a Poll Program Complete command was issued, this code indicates that no program function preceded it. Code 01 represents a function that the AADvance controller does not know or support.
Code 02	Illegal Data Address	The data address received in the query is not a permitted address for the slave. The AADvance controller raises Code 02 when a request specifies an address outside the 16-bit range 0 to 65,535. The exception occurs if the request specifies the address implicitly ('give me the 20 registers from address 65,530') or explicitly (give me the register at address 65,536').
Code 03	Illegal Data Value	A value contained in the query data field is not a permitted value for the slave. The AADvance controller can raise Code 03 only on Boolean (coil) writes.
Code 04	Slave Device Failure	An unrecoverable error occurred while the slave was trying to do what it was instructed to do. Code 04 represents an internal error in the AADvance controller.
Code 06	Slave Device Busy	The slave is engaged in processing a long-duration program command. The master must retransmit the message later when the slave is free. The AADvance controller can be 'busy' and thus raise Code 06 while it is waiting for its application to download or to start. The controller can report itself to be busy

MODBUS Master

The AADvance controller can be used as a MODBUS Master to one or more MODBUS Slave devices. Slave devices can include programmable logic

controllers, remote devices (typically with little or no processing ability) and, more rarely, other functional safety controllers (Trusted or AADvance).

The controller supports the MODBUS RTU and MODBUS TCP protocols, as well as a subset of MODBUS commands. You can use MODBUS RTU with point-to-point and multi-drop serial links, and MODBUS TCP with Ethernet.

NOTE The AADvance controller does not support the MODBUS ASCII protocol.

You can set up a list of messages (commands) for each slave device. MODBUS read commands cause data to read from the slave device to the MODBUS Master, while MODBUS write commands cause data to be copied from the MODBUS Master to the slave device. You can also define a sequence of broadcast write commands, which a MODBUS Master can send to multiple MODBUS RTU slaves without requiring an acknowledgment. The AADvance controller can control and monitor each of the MODBUS Master objects and their slave links.

WARNING: The MODBUS Master functionality has a safety integrity level of zero (SIL 0) and must only be used for non-safety applications.

MODBUS Statistics

The AADvance controller maintains a cycle timer and counters to generate statistics for each slave. The statistics are available to the application to enable reacting accordingly or reporting whether a link is operating correctly. Diagnostic commands also use these statistics. The application can read the statistics as variables (the address of each variable must be specified) and can also reset the statistics to zero. Statistics are available for all point-to-point message transfers, but not for broadcast messages.

Three types of statistics are available:

- Last rate, reporting the length of the most recent scan time
- Maximum rate, reporting the longest scan time
- Average rate, reporting the average scan time.

Statistics are reported in hundredths of a second; these are not reported in milliseconds.

During operation, the statistics are updated on the first message of each slave and on the first slave link of each master to provide the slave and master cycle times. The software also checks the application interface to verify whether a statistics reset is required.

Meanwhile, the MODBUS Master measures the overall scan time. This scan time runs from the first slave start, to round to the first slave to start again. Again, the statistics are updated on the first message of each slave and the first slave of each master to provide the slave and master cycle times and the application interface is checked to see whether a statistics reset is required.

Handling MODBUS Communication Errors

If a MODBUS Master encounters communication errors to a slave device, the master suspends polling of that slave for a length of time. This suspension allows the polling of other slaves on the same master to continue without pausing for the communications time-out on each cycle through the poll. At regular intervals, the controller can ping the non-communicating slave (this is a configurable setting); when the non-communicating slave responds to a ping, then polling of the slave restarts on the next cycle.

NOTE If the controller is receiving error messages in the replies from a slave, the slave remains in the polling cycle; the controller does not treat this as a communication failure.

MODBUS Diagnostics

The AADvance Workbench makes diagnostic information available as application variables. The diagnostic information includes the status of a communication link and of the MODBUS Slave devices connected to it. You can provide the location of the variables during programming in the Workbench.

MODBUS Exception Responses

The AADvance controller uses the absence of a response from a slave device to identify an error on a MODBUS link. Also the slave can return a code for a special exception in some cases.

The MODBUS protocol allows for these errors by returning an error frame to the master. The error frame consists of the original requested function code with the high bit set and a data field consisting of the error code. The AADvance MODBUS Master does not explicitly remember these codes but the master does locate these codes and regard them as a general error to the device. The slave status variable is set to 'Slave Error' enabling the application to respond to the exception.

AADvance Objects for MODBUS Master

The AADvance controller implements its MODBUS Master functionality through one or more MODBUS Master objects, which you created within the AADvance Workbench. Each MODBUS Master object can connect to one MODBUS TCP device or to one or more MODBUS RTU devices.

MODBUS Master objects are allocated to 9110 processor modules. A MODBUS Master object can use one of the two Ethernet ports or one of the two serial ports related to the processor module. If you need more than one MODBUS Master object, and the controller has two or three processor modules, you can share out your MODBUS Master objects between the processor modules to optimize the use of the ports. This provides flexibility where, for example, the

connections to your MODBUS TCP devices span across more than one network.

Failure of a processor module causes the failure of the MODBUS communications allocated to that processor module. When a controller has more than one processor module, you can share out your MODBUS Master objects between two or three of the processor modules. You set up such sharing in the AADvance Workbench when you create the MODBUS Master objects. If a processor module fails, slave communications continue to operate on the channels related to the remaining processor modules.

The AADvance Workbench represents each slave link with a slave object, i.e., one object for each slave device. You can share out slave objects between MODBUS Master objects as desired, subject to a constraint of no more than one slave link for a master object configured for MODBUS TCP Master/slave relationships are configured below the master objects.

MODBUS Master Capacities

MODBUS Master, slave and message configurations are subject to the limits shown in the table.

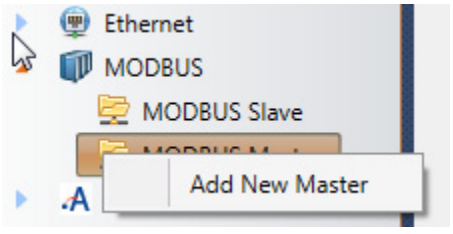
Table 20 - MODBUS Master Capacities per processor module

Item	Limit
Maximum number of masters	32
Maximum number of slaves	5
Maximum number of messages	400

Create a MODBUS Master Object

A MODBUS Master object provides the configuration settings to add a MODBUS Master capability to the AADvance controller. A new MODBUS Master object can be created for each group of multi-drop MODBUS RTU slaves that use the same serial port on the controller and a MODBUS Master object can be created for each MODBUS TCP slave on the network.

MODBUS Master objects can be shared between two or three of the processor modules. You set up sharing when creating MODBUS Master objects. If a processor module fails, slave communications continue to operate on the channels related to the remaining processor modules.

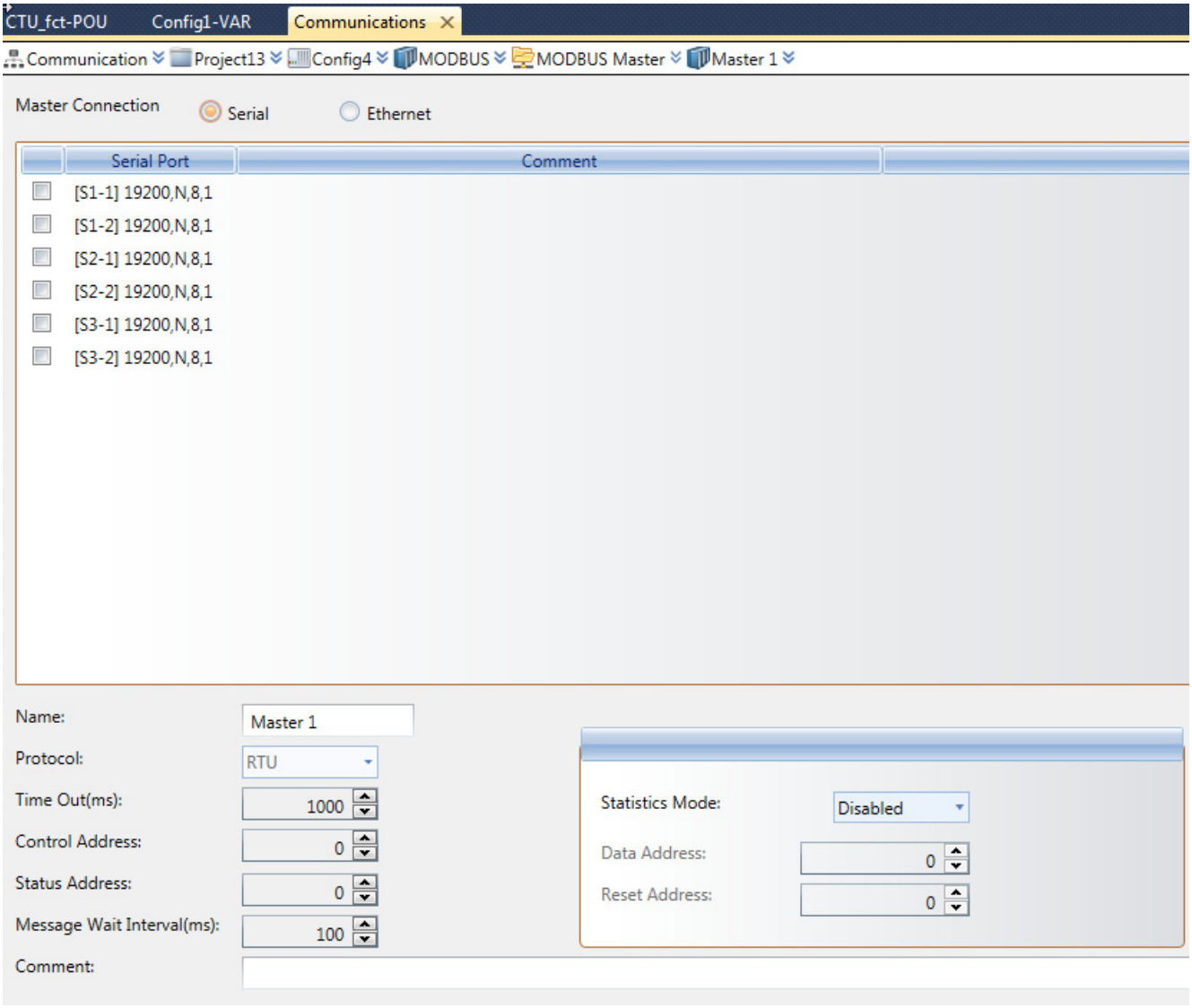


1. Select the Communications view, then choose the Controller that will support the MODBUS Master object.
2. This is the module related to the Serial port or the Ethernet port that you mean to use.
3. Right-click on the **MODBUS Master** tab below a Controller.

4. Select Add New Master.
   - The AADvance Workbench inserts a MODBUS Master object below the chosen processor item.
   - The object is configured for MODBUS RTU; you can subsequently change this to MODBUS TCP.
   - The AADvance Workbench allocates a default name to the object; the name includes the number of the processor module, with a serial number for the object. You can change the name.
5. Right-click on the Master object and rename the Master object.

Configure for MODBUS RTU/TCP

Configure the communications and control settings for a MODBUS Master object for MODBUS RTU or MODBUS TCP.



1. Select the new MODBUS Master object.
2. Select the Serial icon and one of the Serial Port S1-1 to S3-2.

Alternately to set up a TCP MODBUS Master.

1. Select the Ethernet icon and select the Ethernet port E1-1 to E3-2.
2. Enter the following parameters:

MODBUS Master Parameters

Each MODBUS Master object has a set of communication and control settings as shown in the table.

Table 21 - MODBUS Master Communication and Control Settings

Description	Value(s)	Default	Remarks
Protocol	RTU or TCP	RTU	Use MODBUS RTU with serial links. i.e., point-to-point and multi-drop and MODBUS TCP with Ethernet
Port Type	Serial	RTU	Selects the communication port/s used for the MODBUS communication links.
	Ethernet	TCP	Selects the communication port/s used for the MODBUS communication links.
Port Id	Sn-1 .. Sn-2 En-1 .. En-2	where n identifies controller	Controller port identity on the processor base unit. Processor A = 1, B = 2 and C = 3
Timeout	0 to 60,000 ms	1000 ms	This is the time interval the MODBUS Master waits for a response from a slave device before retrying the communication or assuming the slave is unavailable.
Control Address	1 to 65,536, or 0 to disable	0	A control register is specified to control a master object. This parameter specifies the address of the control register (a holding register). • 0 = inactive: Disables the MODBUS Master; The link is inactive, no communication activity; All connected slaves are inactive. • 1 = standby: Forces the MODBUS Master to operate in its standby mode; The link is active but no data is transferred. • 2 = active: Forces the MODBUS Master to operate in its active mode; link is active and transferring data.
Status Address	1 to 65,536, or 0 to disable	0	A status register is specified to report on the status of a MODBUS Master. It is an unsigned integer value (UINT) returned by the MODBUS driver to let the application monitor and act on MODBUS Master faults. This parameter is the address of a status register (a holding register). 0 = healthy: The MODBUS Master is operating normally; the link is active and no errors reported. • 1 = initializing: the MODBUS Master is initializing. • 2 = error: The MODBUS Master has an error and is disabled; Unable to make a link.
Message Wait Interval (ms)	0 to 65,535 ms	100 ms	Enables AADvance to support legacy MODBUS Slave devices with slow communications responses and is used only for Serial RTU links. The time interval sets a short time to see if the response from an earlier MODBUS Master request is not a new request targeted to the legacy slave device. The interval should not be used as a timing control. It applies to all master transmissions and is common to all slave links configured to a master.

Table 21 - MODBUS Master Communication and Control Settings

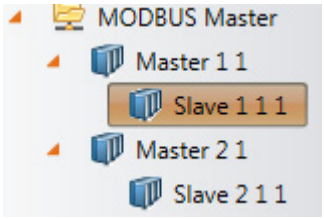
Description	Value(s)	Default	Remarks
Comment	Text string	empty	An optional text string entry.
Statistics Mode			Sets the application statistics to report unless set to disabled. • Last rate, reports the length of the most recent scan time • Maximum rate, reports the longest scan time • Average rate, reports the average scan time. The statistics are reported in hundredths of a second (not milliseconds).
Data Address	1 to 65,536, or 0 to disable	0	Data variable is reported in hundredths of a second.
Reset Address	1 to 65,536, or 0 to disable	0	Write any non-zero value to the reset variable to reset the data variable to zero.

Create Links to MODBUS Slaves

A slave link represents the connection between a MODBUS Master object and a server. The AADvance Workbench represents each slave link by a slave object, one object for each server.

NOTE If you have configured the master object for MODBUS TCP (with Ethernet communications), create only one slave link for the master object.

1. Select the Communications view.
2. Right-click on the **MODBUS Master** object and select Add Slave.



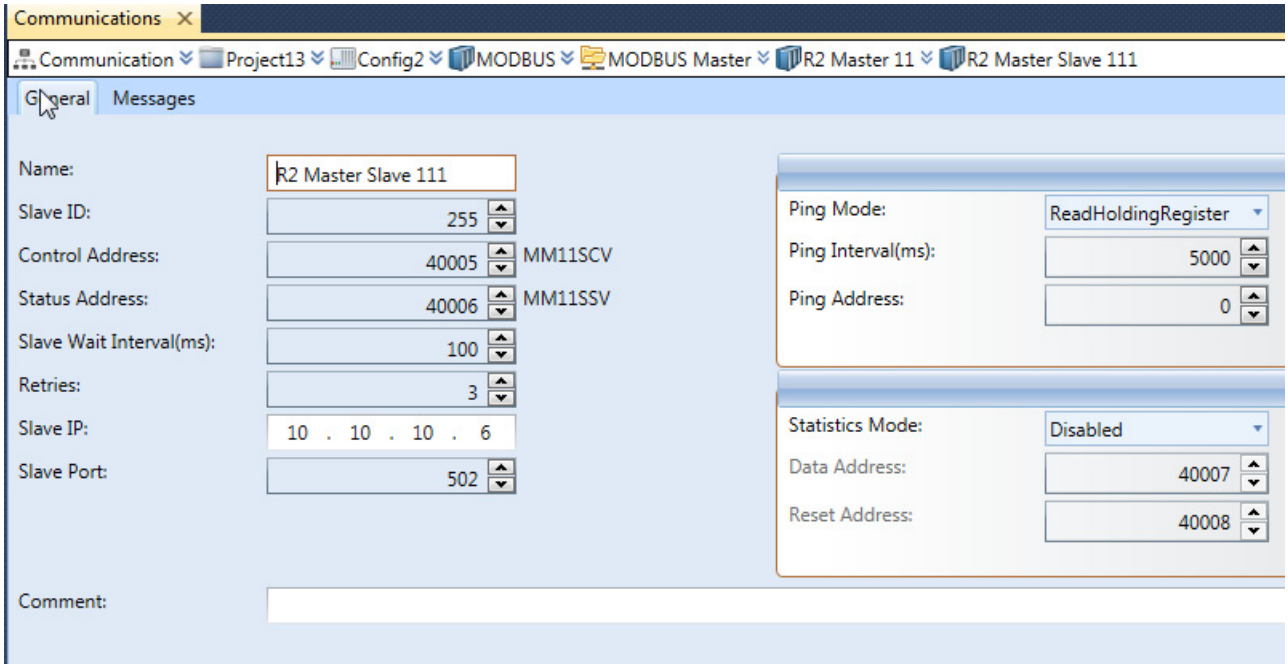
- A default name is allocated to the object; the name includes the number of the processor module and the number of the master, with a serial number for the slave object. You can change the name when you configure the object.

3. Create more slave links if necessary for master objects configured to MODBUS RTU, one for each Slave device.

- You can now configure the MODBUS Slave link objects.

IMPORTANT If you create multiple slave links for a MODBUS RTU master and then choose to reconfigure the master for MODBUS TCP, you must keep one slave link and delete all others. Right-click on each unwanted slave link in turn and select Remove Slave.

Configure a MODBUS Slave - General Tab



The following settings apply to both TCP and RTU master objects operating protocols unless stated.

Select and open the Slave object under a Master and select the **General Tab** and enter the Slave link parameters.

Every MODBUS Slave link object has a pair of registers (holding registers), enabling the application control the link and retrieve status information. The registers are a 'control register' and a 'status register'. Each register is found at a unique address — the Control Address and the Status Address.

The use of the registers is optional. To use a register, you have to declare a variable for the register in the application variables; and the address must also be specified in the holding register map.

Table 22 - MODBUS Slave Link Identification and Control Settings

Description	Value(s)	Default	Remarks
Name	Slave n		Use any alphanumeric characters to assign a unique name.
Slave ID- RTU	1 to 247	1	Allocate a unique Slave ID for each slave configured to a particular master
Slave ID - TCP	255	255	Default Value only for TCP

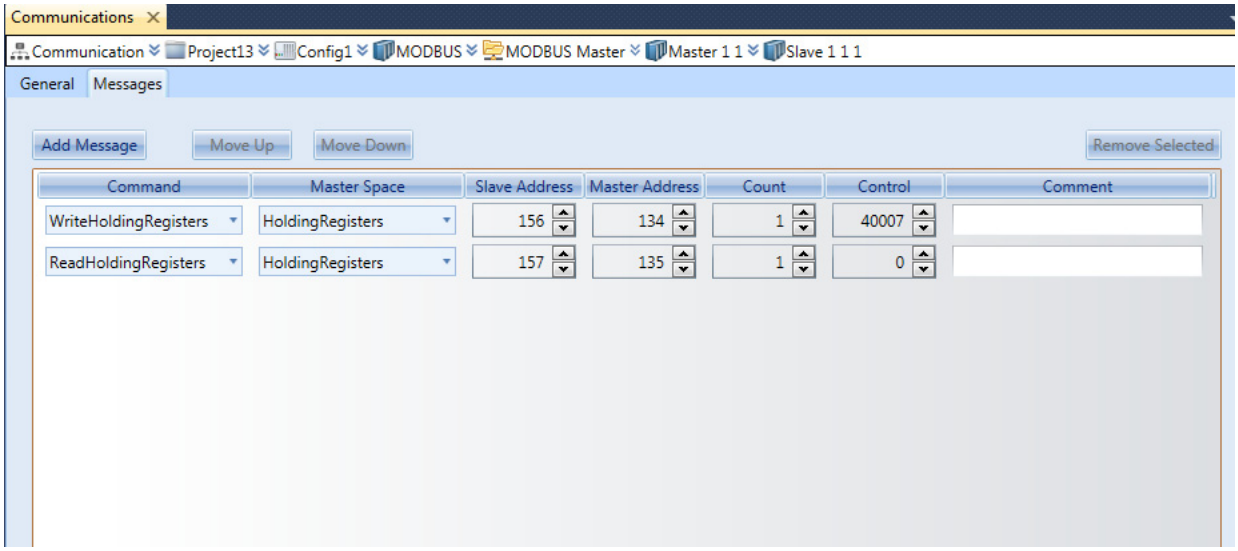
Table 22 - MODBUS Slave Link Identification and Control Settings

Description	Value(s)	Default	Remarks
Control Address	1 to 65,536, or 0 to disable	0	The address of a control register. The application can use the control register to control a slave link. If a control variable address is not specified, then the link is enabled automatically when the controller is started. The variable in the application has the functions listed in the table. • 0 = inactive: Disables the slave link and the slave is not polled or pinged. • 1 = standby. Forces the slave to operate in a standby mode. MODBUS continues to ping the slave to make sure that communications are possible. • 2 = active. Forces the slave to operate in its active mode. The slave is polled for data.
Status Address	1 to 65,536, or 0 to disable	0	The address of a status register. The status register is an unsigned integer value (UINT) returned by the MODBUS driver to let the application monitor and act on faults. If a status register address is not specified, the slave link does not report its status information to the application. The variable in the application has these values: • 0 = healthy: Slave is operating normally, link is active and no errors are reported. • 1 = initializing: The slave link is initializing. This value is also reported if the slave device has failed to respond to a ping on a link declared as standby; if the MODBUS Master is in error; or if the MODBUS Master is set to inactive. • 2 = communications failure: Slave is being polled for data and is not responding. • 3 = error: The last message to the slave resulted in an exception response.
Slave Wait Interval (ms)	0 to 65,535ms	100 ms	Specified individually for each MODBUS Slave; it is the minimum length of time the master will waits from the start of one message to a slave to the start of the next message to the same slave.
Retries	1 to 10	3	Used in conjunction with the master message timeout value. The number of retries the master will retransmits a message to the slave if it does not receive a response and then initiate the Ping mode.
Slave IP (TCP only)	0.0.0.0 to 255.255.255.255	0.0.0.0	The network address used by the slave device.
Slave Port (TCP only)	0 to 65,535	502	The TCP port the slave device is offering its services.
Ping Mode	RTU Options: DoNotPing; FunctionCode08; ReadHoldingRegister	ReadHoldingRegister	The Ping Mode tests the communication between the Master and the Slave. The action depends upon the protocol mode of the Master. The options are different for RTU and TCP. Do Not Ping: Do not ping or test the link. Function Code 8: sends a diagnostic message to the slave echoing operational. ReadHoldingRegister: reads data from the holding register.

Table 22 - MODBUS Slave Link Identification and Control Settings

Description	Value(s)	Default	Remarks
Ping Mode	TCP Options: DoNotPing; ReadHoldingRegister	ReadHoldingRegister	Do Not Ping: do not ping or test the link. **Note:** If you choose Do Not Ping for MODBUS TCP it will help prevent it from communicating. ReadHoldingRegister: reads data from the holding register.
Ping Interval	0 to 65,535 ms	5,000 ms	The ping interval is specified individually for each slave; this interval defines the maximum time between each initiation of a Function Code 08 diagnostic message.
Ping Address	1 to 65,536, or 0 to disable	0	The ping address is specified individually for each slave; this is the address of the slave holding register. It is disabled by DoNotPing and Function Code 08 mode.
Statistics Mode	Disabled; Last Rate; Maximum Rate; Average rate	Disabled	Sets the application statistics to report unless set to disabled. • Last rate, which reports the length of the most recent scan time • Maximum rate, which reports the longest scan time • Average rate, which reports the average scan time. The statistics are reported in hundredths of a second (not milliseconds).
Data Address	1 to 65,536 or 0 to disable	0	Data variable reports in hundredths of a second.
Reset Address	1 to 65,536 or 0 to disable	0	Write any non-zero value to the reset variable to reset the data variable value to zero.

Configure a MODBUS Master Slave - Messages



Select and open the Slave object under a Master and select the **Messages** Tab and enter the Slave link parameters.

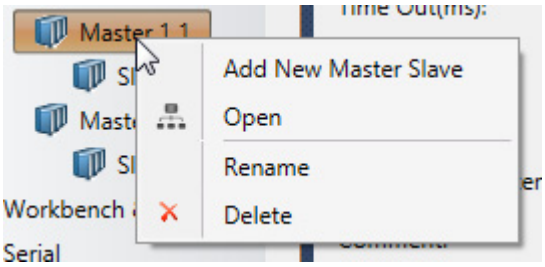
Table 23 - Slave Messages Tab Parameters

Description	Value(s)	Default	Remarks
Add Message			Adds a new message.
Move Up/Move Down			Moves a selected message up or down the in the list.
Remove Selected			Deletes a selected message.
Command	Read Coils	Coils	Reads 1 to 2,000 contiguous status of coils in a remote slave device.
	ReadDiscreteInputs	Coils	Reads 1 to 2,000 contiguous status of discrete inputs (coils) in a remote slave device.
	ReadHoldingRegisters	HoldingRegisters	Reads the contents of a contiguous block of holding registers in a remote slave device
	ReadInputRegisters	HoldingRegisters	Reads from 1 to 125 contiguous input registers in a remote slave device.
	WriteCoils	Coils DiscreteInputs	Forces each coil in a sequence of coils On and Off in a remote slave device. Writes to each discrete input in a remote slave device
	WriteHoldingRegisters	HoldingRegisters InputRegisters	Writes to a block of contiguous holding or input registers (1 to 123) in a remote slave device.
Master Space	Coils		Defines the Command as Coils
	HoldingRegisters		Defines the Command as Holding Register
	DiscreteInputs		Defines the Command as Discrete Inputs
	InputRegisters		Defines the Command as Input Registers
Slave Address	1 to 65536	1	Defines the slave address.
Master Address	1 to 65536	1	Defines the Master address
Count	1 to 125 for ReadHoldingRegisters ReadHoldingRegisters WriteHoldingRegisters	1	The Count value only applies to the Holding Registers.
Control	0 to 65536	0	Specifies the Control Address.
Comment		empty	Optional text string.

Remove a MODBUS Object

You can remove an unwanted MODBUS object.

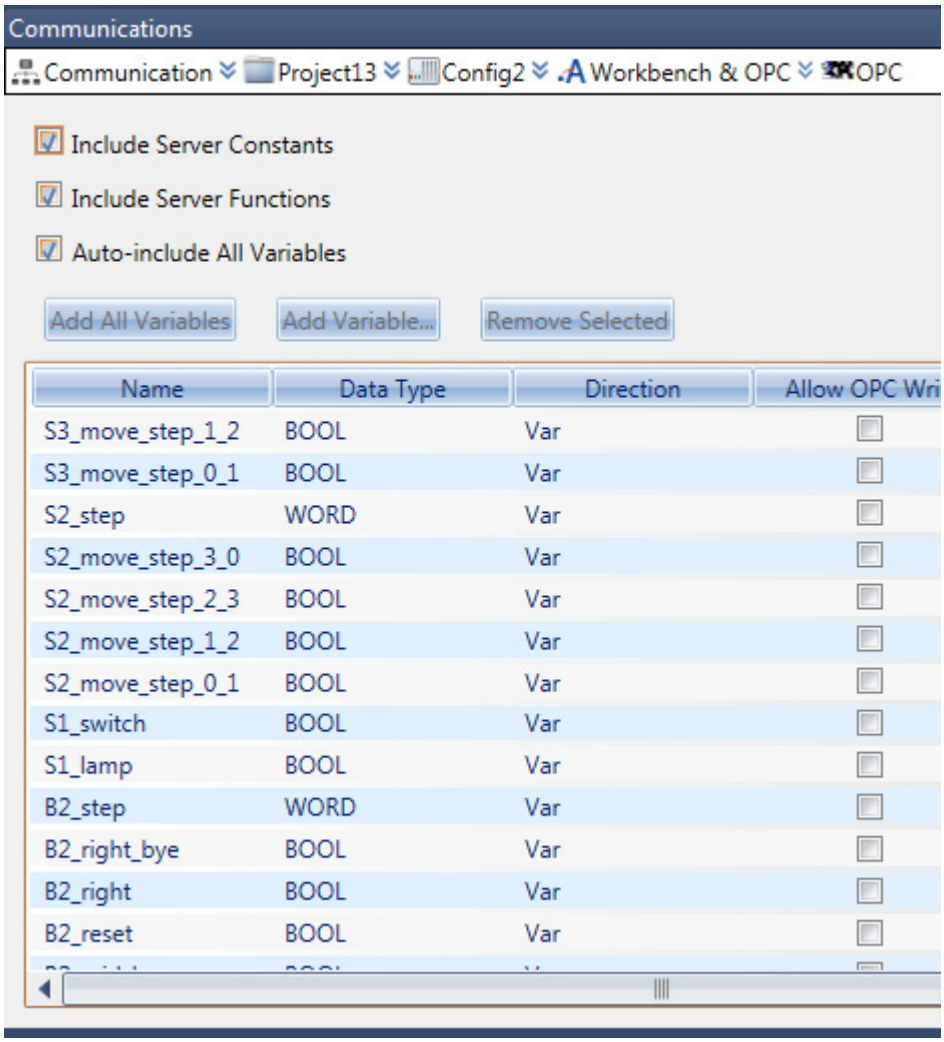
1. Right click on the object, then choose Delete.



2. The object is removed.

OPC

AADvance provides access to an OPC portal when configuring communication with the portal.



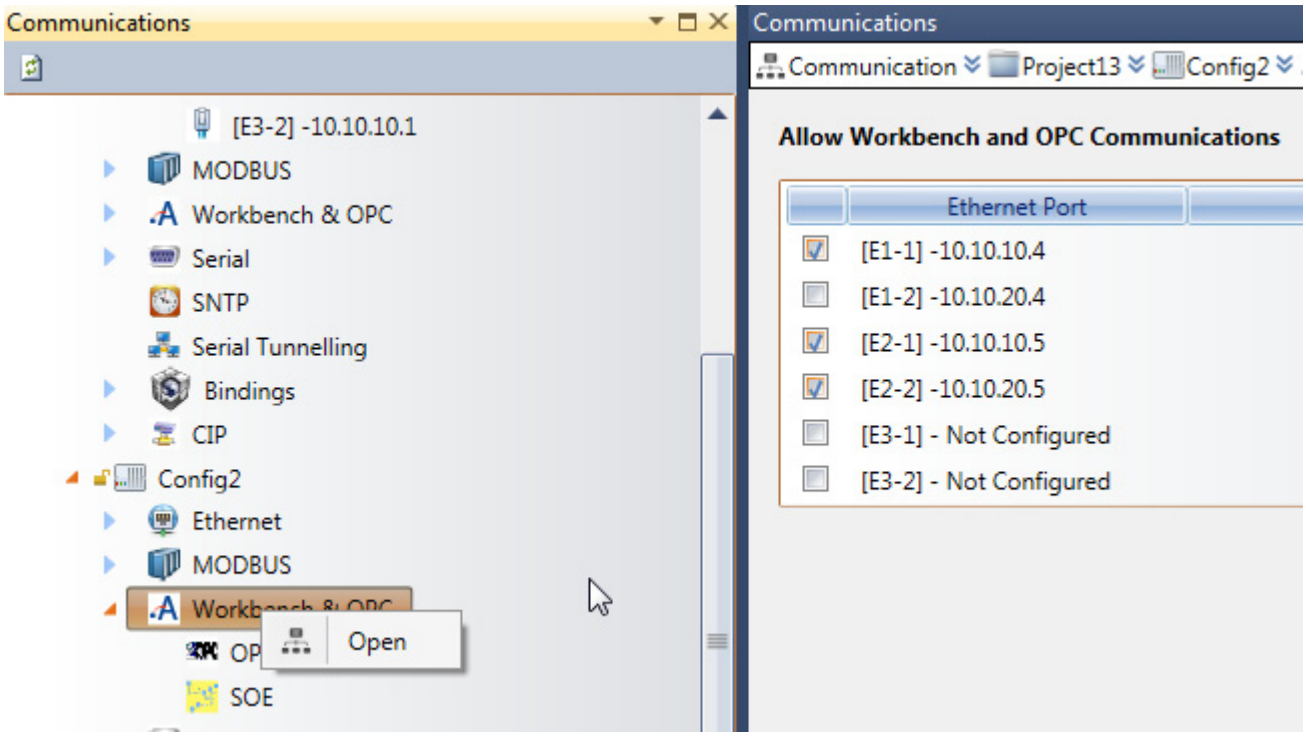
The following properties are available to configure this feature

Property	Value	Default	Description
Include Server Constants	True or False	True	Includes the OPC server constants
Include Server Functions	True or False	True	Includes the OPC server functions
Auto-include All Variables	True or False	True	Automatically adds all variables
Add All Variables	True or False	False	Adds all variables in the Library when manually selected
Add Variable	True or False	False	Opens the Variable Selector window to enable selecting an individual variable from the Library
Remove Selected	True or False	False	Removes a selected variable

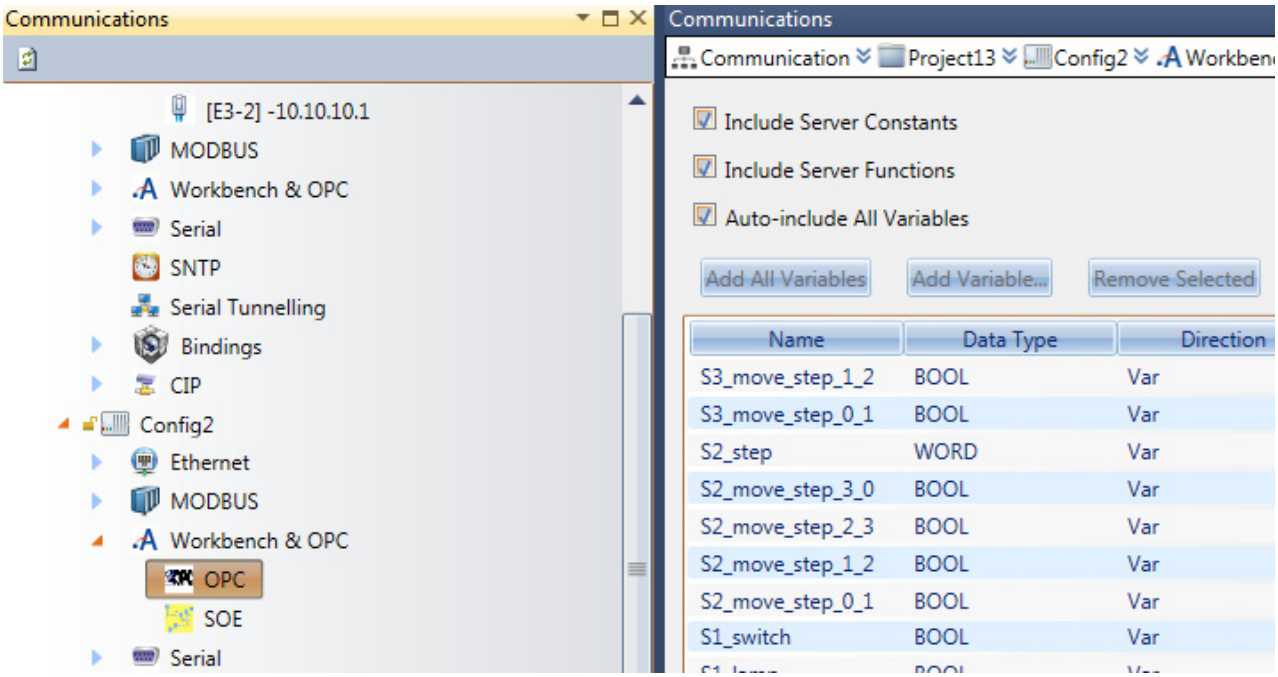
Configure OPC Parameters

You configure the OPC from the Communications View.

1. In the Communication view, select the **Workbench & OPC** Node. The Allow Workbench and OPC Communications window opens.



2. Select the **Ethernet port** to communicate with the OPC server.
3. Select the **OPC** icon. The OPC parameter window is displayed.



4. All three include check boxes are checked selected to set a default value of true for each option. Select these to exclude the values from being included in the OPC mapping.
5. For example, de-selecting the Auto-include All Variables removes the un-selected variables from the mapping list.
6. To add all variables to the list select **Add All Variable** button.
7. To add a single variable select the Add Variable Button. The Variable Selector window is displayed. Choose a Variable and click **OK**.

8. To remove a single variable select the variable on the list and then click Remove Selected.

Sequence of Events (SOE)

AADvance provides SOE for Boolean variables. When the variables are chosen you can define their SOE properties.

Add Variable: Opens the Variable Selector list to choose a variable and add it to the list. You must use this button to add the first variable after which you can select the variable, right-click on the variable and set of options are provided to add or remove new variables.

Remove Selected: Removes the selected variable from the list.

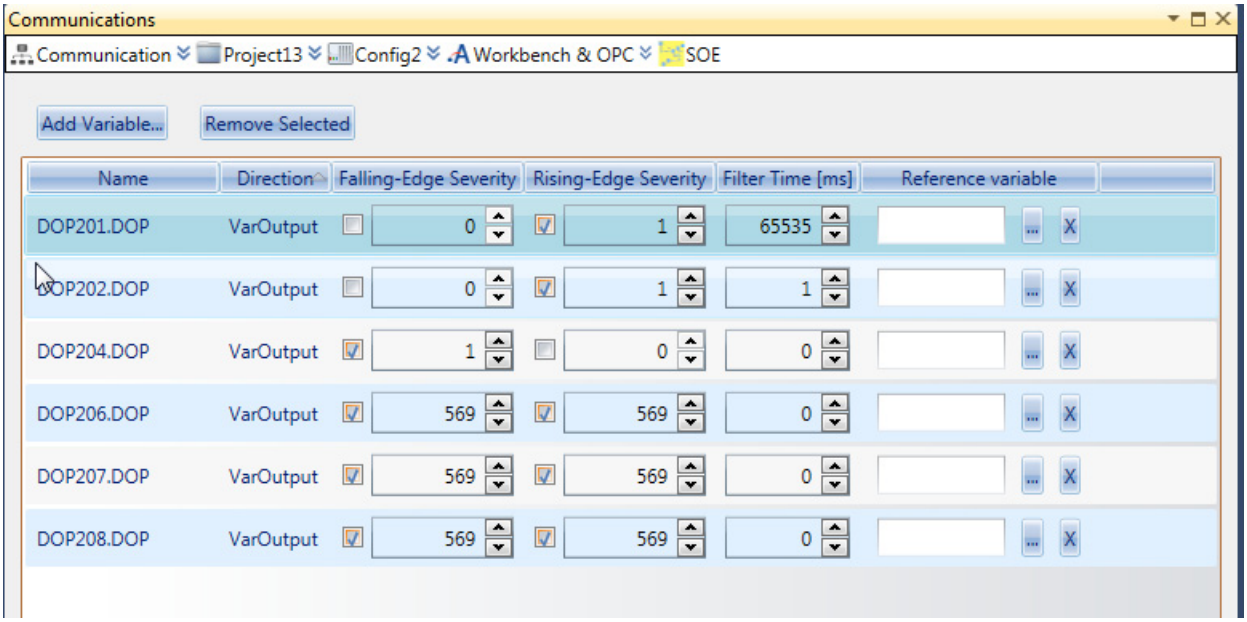
Table 24 - Variable Properties

Property	Range	Default	Description
Name	text string	Blank	Variable name.
Direction	text string	varout	Variable direction - output.
Falling-Edge Severity	1 to 1000	1	Indicates whether the service detects a fall from TRUE to FALSE. When set to TRUE, the service detects a falling edge; when set to FALSE ignores the transition.
Rising-Edge Severity	1 to 1000	1	Indicates of whether the service detects a rise from FALSE to TRUE. When set to TRUE, the service detects a rising edge; when set to FALSE, the service ignores the transition. Possible values are 1 to 1000.
Filter Time [ms]	0 to 65535	0 ms	The minimum time lapse between two events. Possible values range from 0 to 65535 ms.
Reference Variable	text string	blank	Variable of any elementary data type.

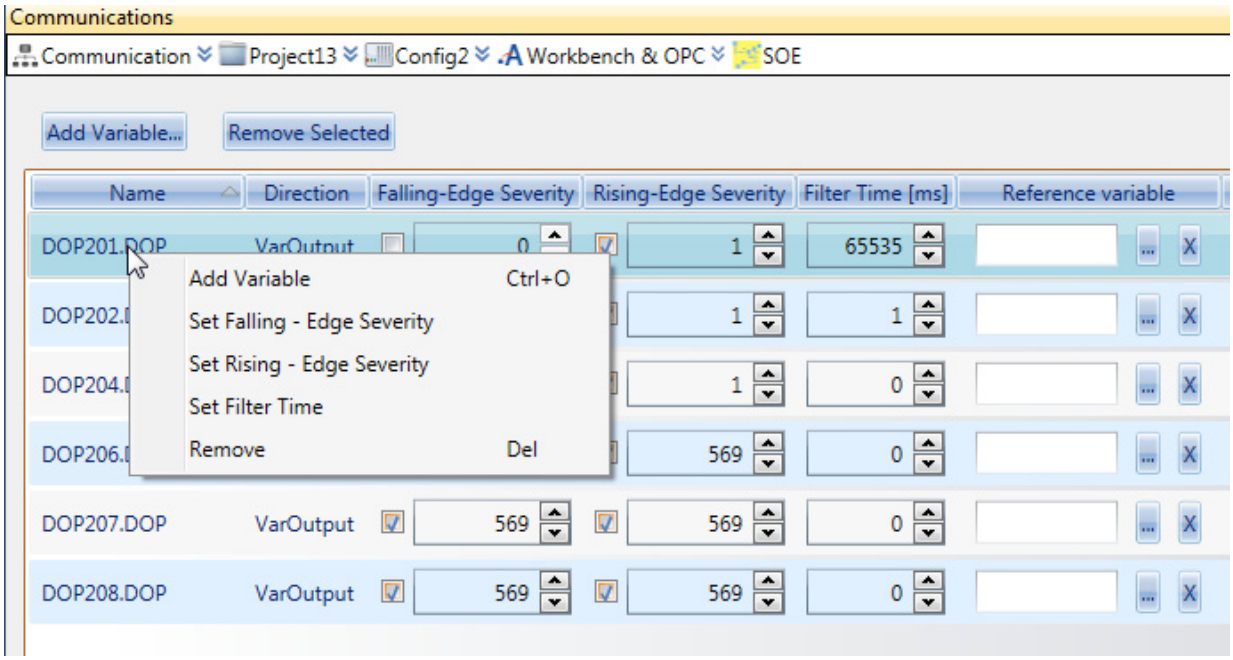
Configure SOE Variables

You set up the SOE variables from the Communications View.

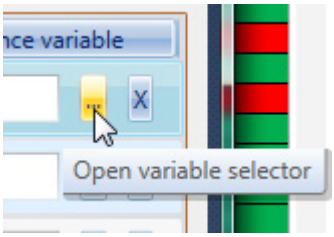
1. Select the **Workbench & OPC** node, then select the **Ethernet** ports if they are not shown as selected.
2. Select **SOE**.



3. To add the first variable to the list, use **Add Variable** button to add the first variable to the list.
4. To set the properties Falling-Edge Severity or Rising-Edge Severity select the variable and choose the relevant option.



5. Alternatively you can select the check box adjacent to the option and select/enter a value into the field.
6. Set the Filter time by entering a value into the field or choose the option.
7. To open the Variable Selector, use the options or click the Variable Selector button and choose a variable for the Reference variable.


nce variable
X
Open variable selector

Notes:

SNCP

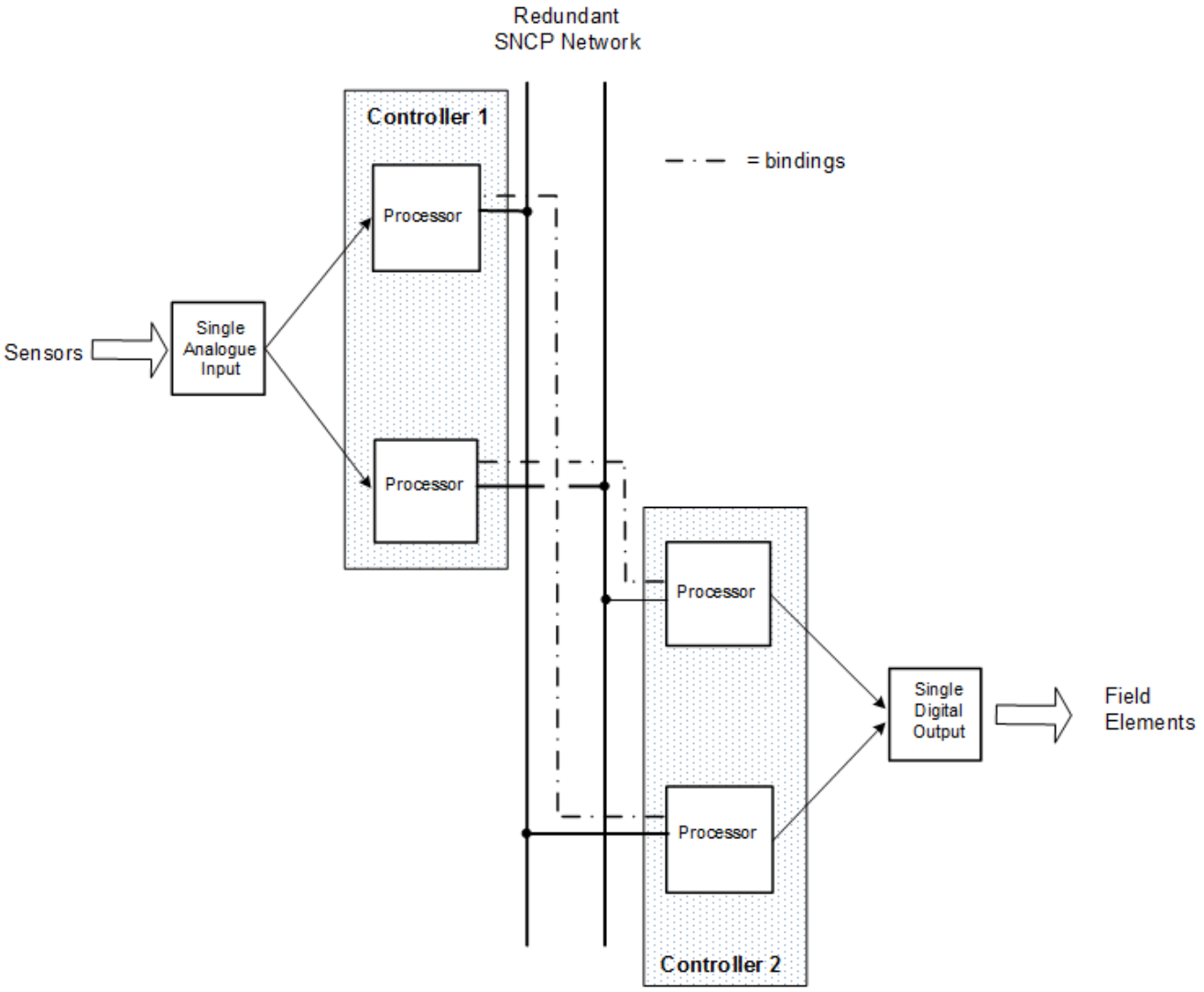
SNCP (Safety Network Control Protocol) is a SIL 3 certified protocol supplying a safety layer for the Ethernet network making the network connection a "Black Channel". Data is exchanged by creating a relationship between variables in different AADvance® controllers; this is called "Binding Variables". This chapter describes the process for setting up SNCP variable bindings.

Bindings and the SNCP Network

Bindings are based on a producer/consumer model. The controller consuming the data establishes a binding link with the controller producing the data and manages all of the sending and receiving of data. It schedules the sending and receiving of data, sending the diagnostic data, managing the safety response if faults occur and managing the communications redundancy. An SNCP network is illustrated in the diagram.

There must be a physical connection between the two controllers. The design of the Ethernet network and the equipment used does not impact the SIL rating of the communications interface, but the design of the network does change the reliability of the network and does impact the spurious trip rate. SNCP Network data can be combined on a common network resulting in safety and non-safety data sharing a common physical network. This does not compromise the SIL rating of the network but again does introduce failure modes and possibly security risks which can increase the spurious trip rate. Therefore, careful consideration must be given to the network topology during the applications specification and design phase.

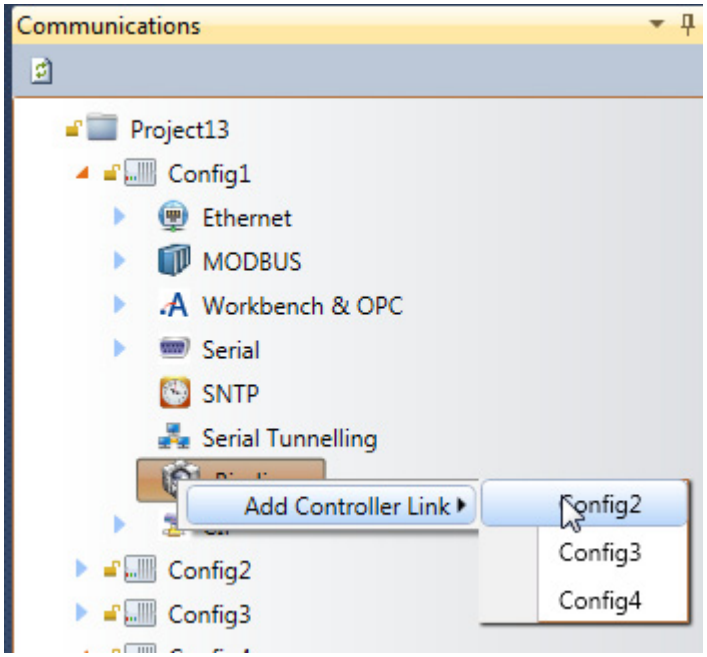
SNCP Networks can be configured as Simplex (Fail Safe) or Redundant (Fault tolerant). The network configuration is dependent on the applications safety and availability requirements. The giving and receiving of data occurs independently from the physical network configuration as the connection between the controllers is treated as a logical network.



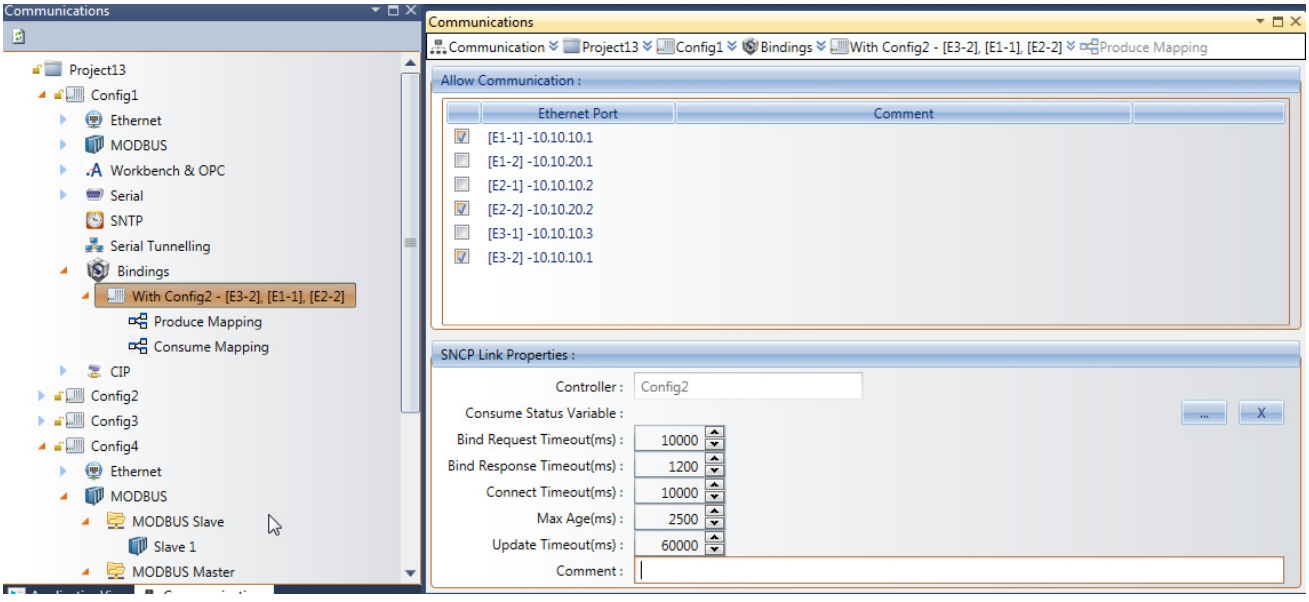
Create a Binding Link

To set up a binding link you must link two controller nodes using the Communications View. Below each controller node are the binding produce and consume elements. You can also rename controllers as required.

1. Select the **Communications** view.
2. Select a **Bindings** node under the controller you want to connect from.



3. Right-click the node and from the **Add Controller Link** options select the controller to link to (Config 2 for example). Repeat this to link to more than one controller if required.
   - A controller link node is displayed under the Bindings node indicating that the link is not configured.
4. Select **Open** at the controller link node.
5. The Allow Communications window is displayed.
6. Select the Ethernet ports to use for the link.



7. The Bindings node changes to With Config n - [E1-1], [E2-1]
8. Set the SNCP Link Properties:

Variable	Range	Default
Bind Request Timeout	0 to 65535 ms	10000 ms
Bind Response Timeout	0 to 65535 ms	1200 ms
Connect Timeout	0 to 65535 ms	10000 ms
Max Age	0 to 65535 ms	2500 ms
Update Timeout	0 to 65535 ms	60000 ms

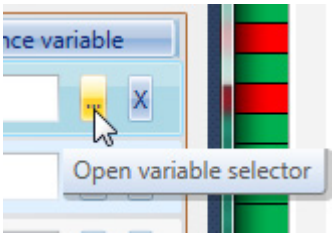
The Consumer Status Variable can be optionally configured to provide binding link status to the application. The variable is a DINT.

NOTE Linked controllers must have the same SNCP Link Properties.

SNCP Link Properties

A Consume Status Variable is used to report the status of the link. For a Healthy binding link, the value is 0, with a non-zero value indicating a fault. The fault values are described in the topic SNCP Binding Error Values.

- Click the variable selector button and select a variable from the Variable Selector.



The SNCP key time out values are the MaxAge and BindRespTimeout. These settings govern the number of binding requests a consumer sends without getting a response before considering the bindings link as failed and disconnecting.

ConnectTimeOut

The ConnectTimeOut timeout serves two purposes:

- The time the resource waits to connect to a producer when a consumer resource starts up.
- Provides a timeout for a producer's response once a consumer has sent a "connect" request to a producer. If the producer does not respond within the timeout, the consumer sends another "connect" request. The consumer continues to send "connect" request at intervals given by this timeout until a connection is established. For a healthy binding link the ConnectTimeOut will be 0, with a non-zero value indicating a fault. The fault values are described in "SNCP Binding Error Values".

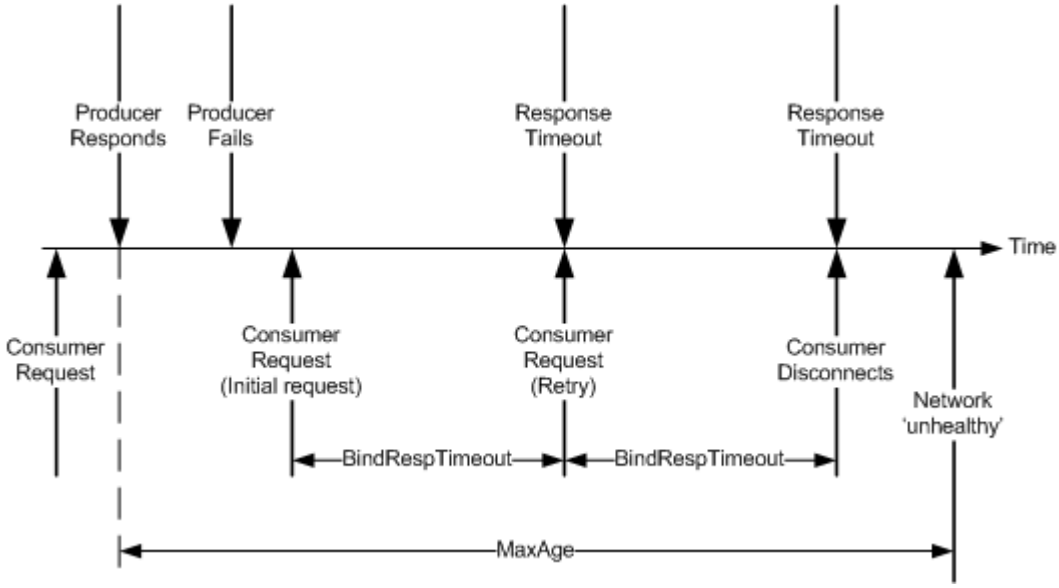
BindRespTimeout

This BindRespTimeout timeout is used in a consumer to timeout a binding data response from a producer. Once a consumer has established a binding connection with a producer, it sends a request for binding data; this timeout value timeouts out the producer response.

If no response is received, either another request (retry) is sent, or the consumer disconnects from the producer. The number of retries that are attempted before a consumer disconnects depends on the value of this parameter and MaxAge. The consumer will continue to send requests until MaxAge expires.

In general the number of requests sent before disconnecting is calculated as follows: MaxAge / BindRespTimeout.

The following diagram illustrates (not to scale) the timeout behavior when the MaxAge is configured as 2500 ms and BindRespTimeout is 1000 ms. The total number of requests sent before disconnecting is 2.



Once disconnected, the consumer attempts to re-establish a connection to the producer by sending a connection request at 'ConnectTimeOut' intervals.

MaxAge

This consumer resources uses the MaxAge timeout serving two purposes. Firstly, this timeout is the time during which a consumer must receive a valid binding message from the producer before the physical network is considered 'unhealthy'. Stale, corrupt, or out of sequence messages are not considered valid. Continued reception of such messages can be indicative of underlying network problems, in which case this will be reflected by an 'unhealthy' network status (as reported by the KVB status function blocks).

Secondly this timeout value is also used to examine the age of binding data message received by the consumer. If a message contains data that is older than this value, the message is discarded. This can occur if the message is delayed because of network problems.

The number of requests is determined by MaxAge/BindRespTimeout. To increase the number of retries you must increase MaxAge or decrease BindRespTimeout.

BindReqTimeout

The producer uses the BindReqTimeout. Once a binding connection is established with a consumer, the producer uses this timeout value to timeout binding data requests from the consumer. If producer does not receive a request for binding data from the consumer within this timeout period, the producer disconnects the consumer.

Subsequent requests for data from the consumer are ignored until the consumer establishes a new connection.

NOTE The Consumers Network bindings parameters (i.e. timeout values) are those located in the Producing Resource.

UpdateTimeout

The consumer and producer resources uses the UpdateTimeout during an online update. During an online update all binding connections are closed. The SNCP binding driver then restarts with the potentially new binding configuration. This timeout value is the time during which the consumer redefines its binding connections.

During this time any configured binding error variables will continue to indicate 'healthy', even though there is no connection between exists consumer and producer.

Failure to make the connection again within the timeout results in binding error variables becoming 'unhealthy'.

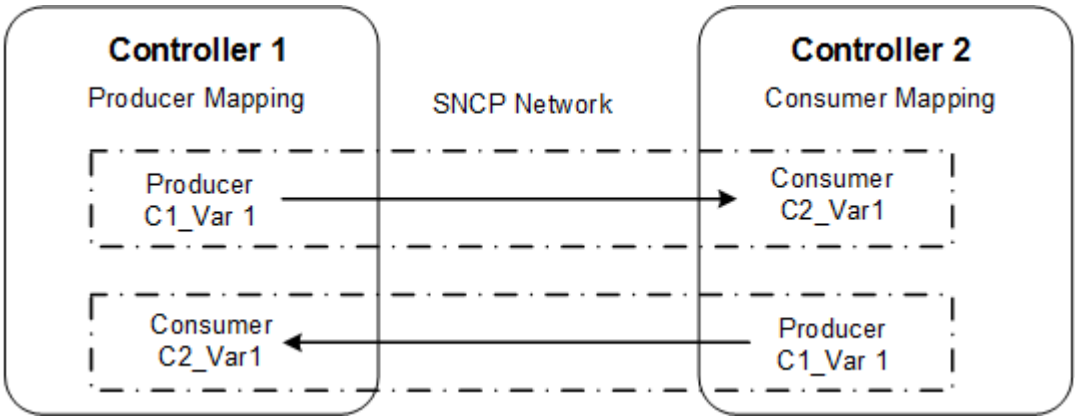
IMPORTANT From the Workbench perspective, it is important to do online updates for both the producer and consumer resources as quickly as possible, to help prevent this timeout from occurring.

Bindings

Bindings are directional links and provide access paths between variables located in different controllers that are part of the same project. Bindings are between variables of the same type and are defined as a producer variable sending values to a consumer variable. Individual controllers have variable binding definitions from producer mapping and consumer mapping perspectives. For example, you can view the C1_Var1 (producer) to C2_Car2 (Consumer) binding between Controller1 and Controller2 from the Producer Mapping view for Controller1 or the Consumer Mapping section for Controller2.

Bindings use the parameters defined for the producer controller. Multiple consumers can connect to a single producer. A single consumer can connect to multiple producers. A producer can also be a consumer and vice versa.

Variables are not updated in the consuming controller until the producing controller sends the values through the binding media.



NOTE When importing or adding existing controllers having bindings, these controllers must keep the same number and name as when exported to retain binding definitions. Therefore, before importing or adding such controllers, make sure to renumber and rename conflicting controllers.

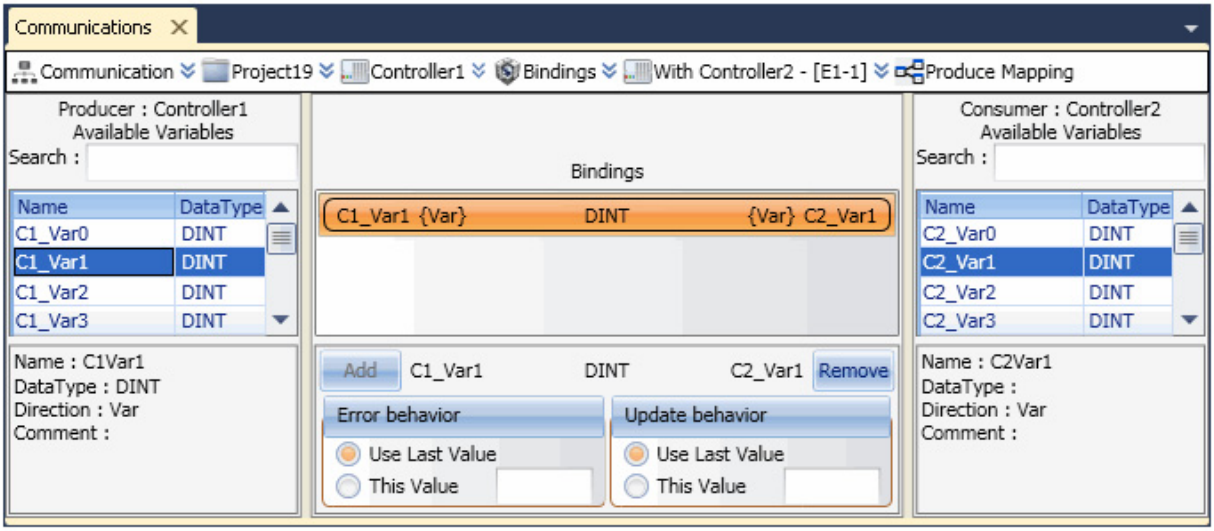
Configure Bindings

When creating a variable binding, you define the binding either from the Producer element to the Consumer element or from the Consumer element to the Producer element. The following procedure uses the Producer to Consumer direction:

Using the Controllers linked in the SNTP network set up procedure.

1. Right-click on the controller link node (With Config"n" - [En-En]) under the Bindings node on the Communications view to display the Producer and Consumer Mapping elements.
   - The Bindings Produce Mapping and Consume Mapping elements are displayed. The Producer Mapping editor lists the available Producer variables on the left and the available Consumer variables on the right.
2. Select the Produce Mapping element. The mapping editor is displayed.

Variable binding from controller produce mapping perspective



- The Produce Mapping editor lists the available producer variables on the left and available consumer variables on the right.

3. Select a variable from the Consumer list. The details are displayed below the list and are shown next to the Remove button. Verify that the variables are the correct type (same type).
4. Enter the Error behavior and the Update behavior. You can select Use Last Value or enter a specific value for the variable when an error occurs.
5. Select the **Add** button. The bindings are moved to the top of the Bindings list.
6. To delete a binding, select a binding in the Bindings list then press the Delete key.

SNCP Binding Error Variables

To report binding errors create the following defined words in the dictionary for your project:

Error Message	Value (Code)	Description
SNCP_KVB_ERR_BINDING_READY	0x00	Normal Steady State Data Exchange
SNCP_KVB_ERR_BINDING_IN_PROCESS	0x01	Producer and consumer are in the process of connecting
SNCP_KVB_ERR_UPD_IN_PROCESS	0x02	Producer and Consumer are in the process of re-connecting following an online-change update
SNCP_KVB_ERR_UPD_TIMEOUT	0x03	Producer and consumer did not re-connect within Update Timeout milliseconds following an online change update.
SNCP_KVB_ERR_NO_PRODUCER	0x04	At the beginning of a connection or following BindRespTimeout milliseconds during data exchange.
SNCP_KVB_ERR_BAD_CRC	0x05	Producer and consumer binding tables CRC mismatch during connection
SNCP_KVB_ERR_IMPOSSIBLE_TO_BIND	0x06	Conversion of structures not supported on heterogeneous binding link

Error Message	Value (Code)	Description
SNCP_KVB_ERR_IP_DENIED	0X07	Producer only error, indicates that a consumer with an unknown/unexpected IP Address is requesting a bindings link. The link is denied.
SNCP_KVB_ERR_BAD_PRODUCER_ID	0X08	Producer only error, indicates that the producer has received a bindings link request from a consumer but the request does not contain the expected producer identity. The link is denied.
SNCP_KVB_ERR_BAD_GROUP_ID	0X09	Producer only error, indicates that the producer has received a bindings link request from a consumer but the request does not contain a known bound variable group identity. The link is denied.
SNCP_KVB_ERR_BAD_BUFF_SIZE	0X0A	Producer only error, indicates that either the producer or consumer has insufficient buffer space to translate the bound variables. Unable to establish the link.
SNCP_ERR_BAD_CONFIG	0X0B	Producer only error, indicates that the producer is unable to obtain the consumers IP address information from the network configuration file. The link is denied.

Notes:

Peer-to-Peer Network

AADvance® provides the options to set up a Peer-to-Peer network. The Peer-to-Peer protocol enables you to communicate application data between up to 40 AADvance or Trusted® systems for each peer network. Data can be transferred between individual systems or from one to many systems at the same time using multicast network communication.

IMPORTANT The current AADvance controller only supports two Multicast IP Addresses, one fixed address can be configured for each Ethernet port.

Peer-to-Peer Features

A Peer-to-Peer network consists of one or more Ethernet networks connecting together a series of AADvance and/or Trusted controllers to enable application data to be passed between them. A Trusted controller with four communication interfaces has eight Ethernet ports and Peer-to-Peer network can use all eight physical Ethernet networks (referred to below as subnets) to provide redundant data paths via eight separate physical routes.

Both network interfaces of an AADvance processor module can be used for peer communications at the same time. A TMR controller with three processor modules provides a maximum of six physical connections that can be divided between different peer networks, or assigned to the same network as required.

Network subnets may be assigned to the processor Ethernet ports. Normally, subnets of the same redundant network would use different processor modules.

The information to be transferred over the peer network is defined within the application using input and output blocks of the standard form. The blocks configure data blocks of 16 or 128 Boolean points, 16 or 128 analogue points, and network status information.

Peer-to-Peer Configuration Process

Configuring the AADvance controller to communicate with a Trusted Peer-to-Peer network requires the same parameters as a Trusted configuration. The configuration for AADvance is arranged in a tree in the I/O Configuration view. The recommended process for setting up a Peer-to-Peer network is as follows:

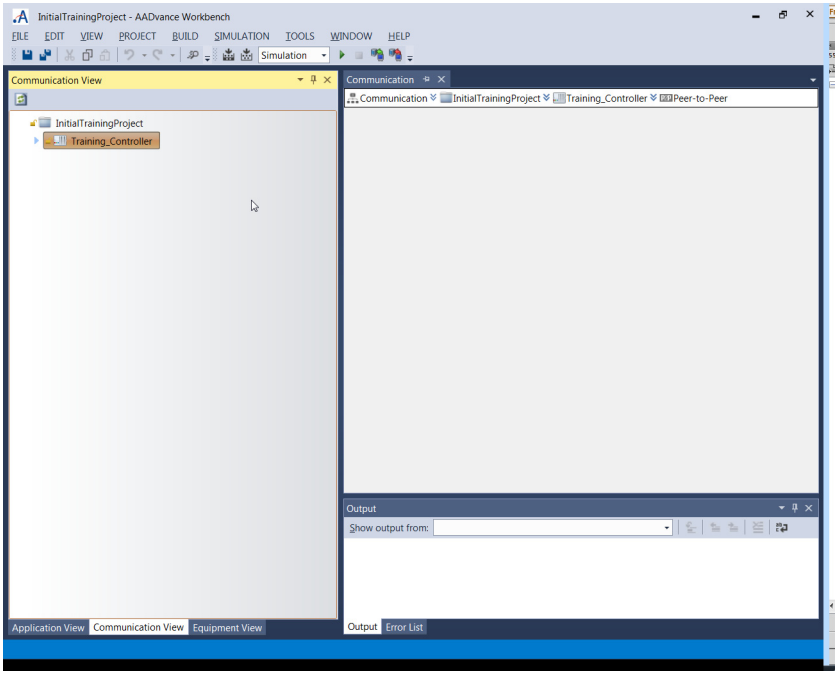
- create a Peer-to-Peer network and give it an identity
- select a Peer-to-Peer subnet configuration and enter the subnet data
- enter the peer IP Address information
- configure the I/O blocks

- wire the I/O block status parameters to variables
- wire the I/O block channel data to variables.

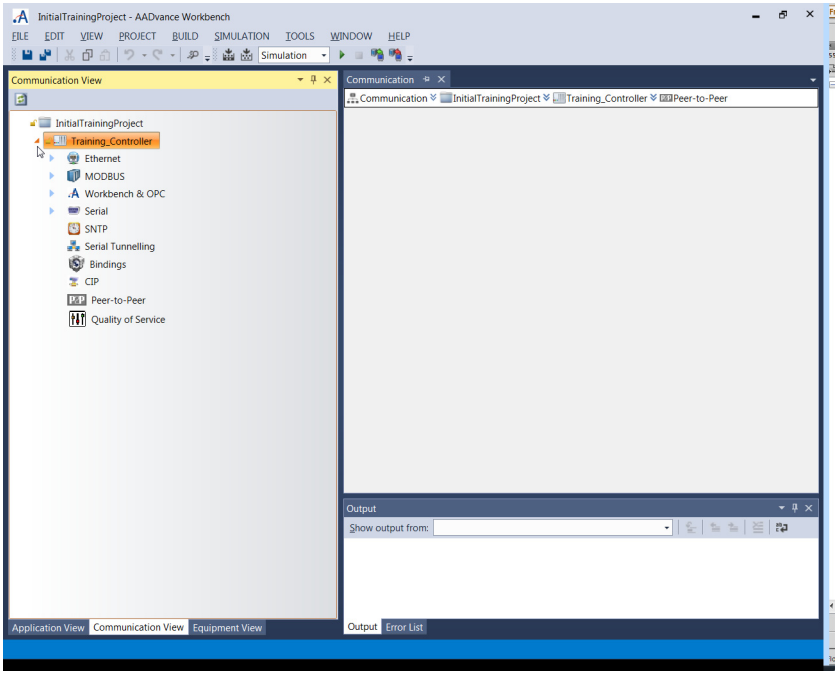
Create a Peer-to-Peer Network

Create a Peer-to-Peer network as follows:

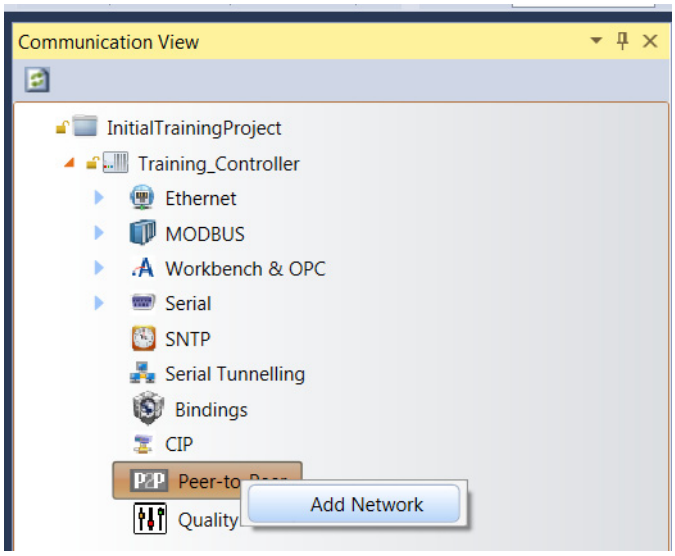
1. Select the **Communication View** tab.



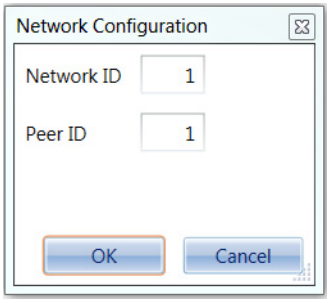
2. Click the expansion arrow to display the drop-down box menu.



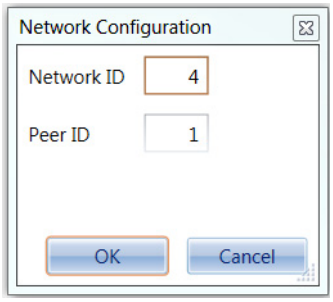
3. Right-click on the **Peer-to-Peer** node to reveal the **Add Network** option.



4. Click on the **Add Network** option, then set the required Network ID and Peer ID options.

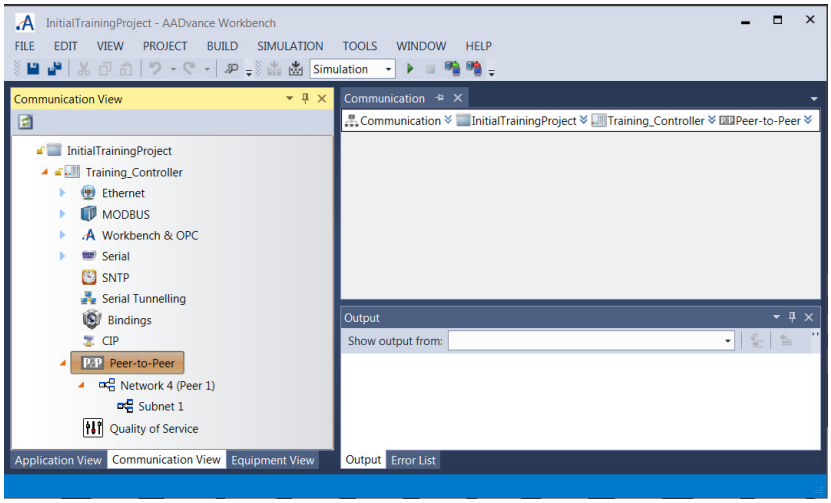


5. In the following illustration, Network 4 and Peer Identity 1 has been selected.



   - The peer identity gives the position of the controller's IP address in the Peer List

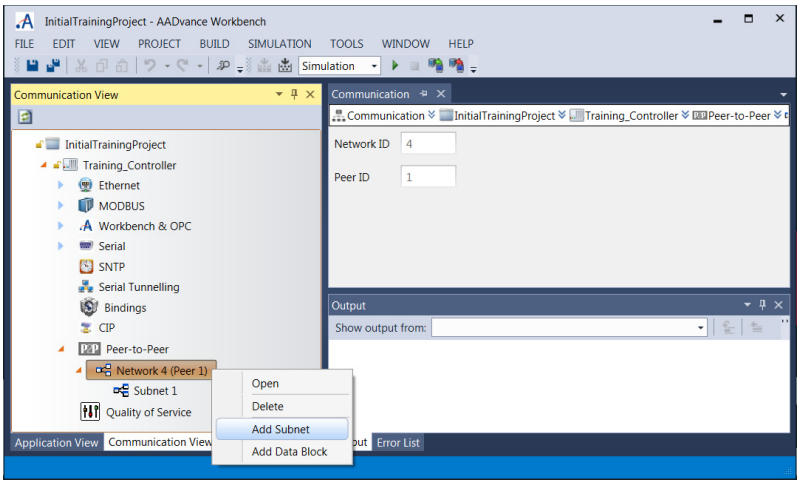
6. Click **OK** button on the Network Configuration dialog box.
   - A peer network Network 4 and Peer Identity 1 is set up.



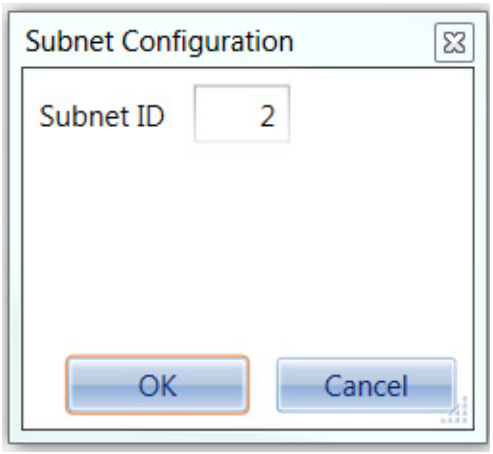
- The first Subnet, labeled Subnet 1 is also added. For addition of further subnets See Peer-to-Peer Subnet Controller Configuration below.

Peer-to-Peer Subnet Controller Configuration

Configure the subnets as follows:Right-click on **Network 4 (Peer 1)** and select **Add Subnet** from the drop-down menu box.

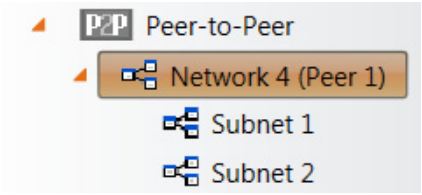


7. Click on **Add Subnet** to reveal dialogue box.

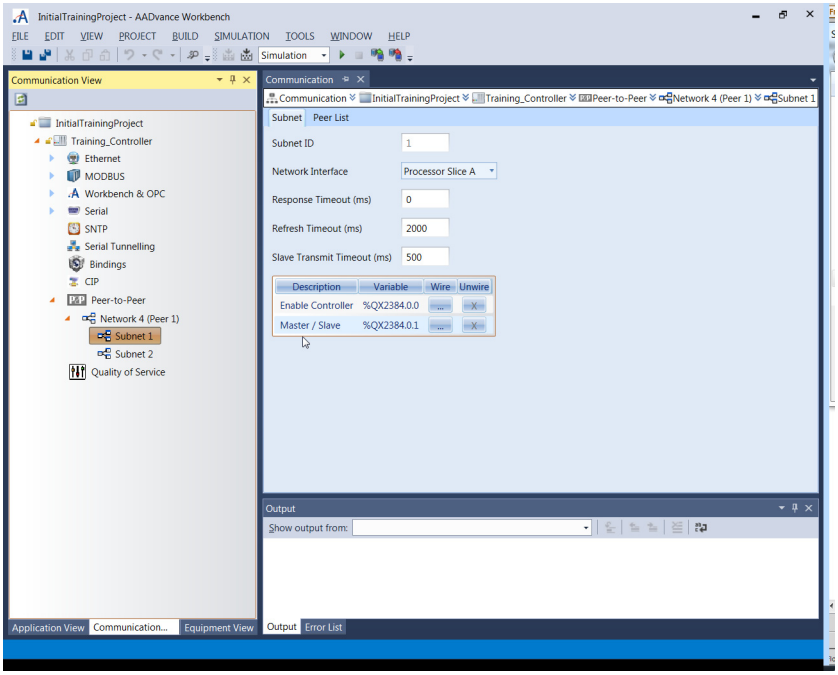


8. Click **OK** button.

- Additional subnet, Subnet 2 is set up.



9. For this example, select Subnet 1.
   - The subnet set up screen is displayed:



10. Enter the following data in the required fields.
    - Network Interface - enter which processor slice (Processor A, B or C) the peer network will be configured on, i.e. which processor module the network is physically connected to.
    - Response Timeout - Enter a value in the range 0 to 10000 ms. This value sets the time allowed for the peer to acknowledge a data packet. If the field is set to zero this indicates that no acknowledgment is required. It should only be set to a value above zero to avoid network packet sequence errors in networks where the propagation delay between any two nodes could exceed 1 ms.
    - If the entire Peer-to-Peer network is on a local Ethernet network with no routers or gateways, this setting may be zero. If the network is not a simple local network, set this parameter to exceed the longest expected propagation delay between any two nodes. The 'ping' command may be used to measure delays once the network is built. If the parameter is not zero, an acknowledgment is expected for every packet, to verify it has been delivered. The Response Timeout acts as a watchdog trip for a lost packet, and it should be the shortest of the timeout settings.
    - Refresh Timeout - Enter a value in the range 0 to 10000 ms. This value sets the time that the controller will wait for its turn in the cycle to send its output data. The setting is used by both masters and slaves.
    - The Refresh Timeout should be set to the Slave Transmit Timeout (below) x the number of peers that will be lost on a worst-case network break.

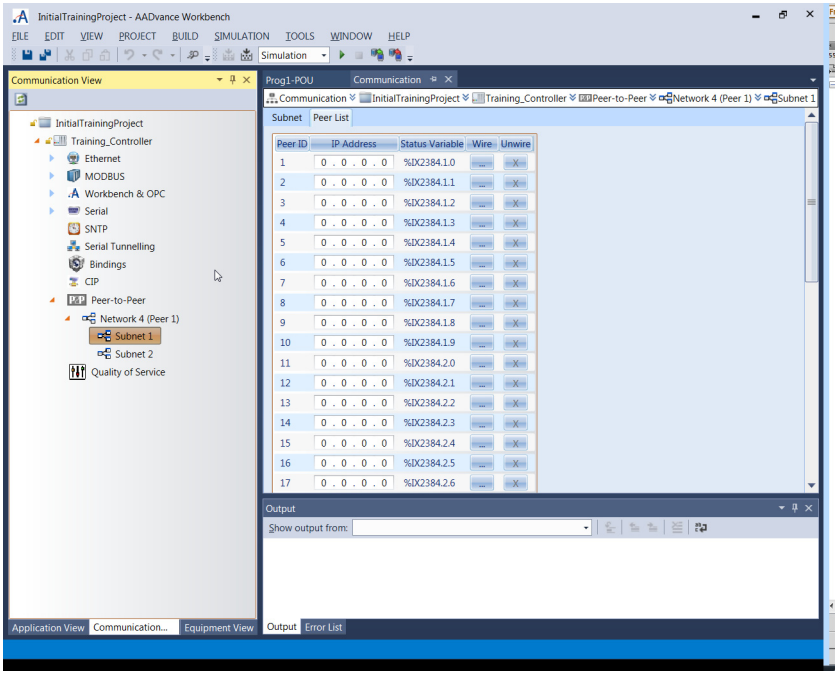
- The Slave Transmit Timeout should trip before the Refresh Timeout if the network is broken. This will minimize the loss of data transfer.
- Slave Transmit Timeout - Enter a value in the range 1 to 10000 ms. This value sets the time a network master controller will wait for a slave to complete transmission of its data and return control of the network before declaring the slave absent. This parameter will be ignored during slave mode.
- The Peer-to-Peer network operates in a cycle. Each controller in turn is asked to send all its configured output data. The Slave Transmit Timeout is only used by the master of a network. It acts as a watchdog for a lost slave, and it has to allow time for the slave to send all its outputs even if the slave is itself waiting for lost packets for its Response Timeout setting. Therefore the Slave Transmit Timeout should be set to:

  (The slave's Response Timeout) x (The maximum number of output data blocks in any controller) + 16 ms
- If the Response Timeout is 0 ms, use 2 ms in the calculation above. A minimum Slave Transmit Timeout of 64 ms is recommended.

11. Wire Enable Controller and Master/Slave to variables you have set up for this purpose.
    - Enable Controller - This variable starts or stops the peer communications using this controller, it should be set to TRUE to enable the controller.
    - Master/Slave - This is a Boolean value that selects the controller's role, it should be set it to TRUE to make the controller act as a Master or FALSE to act as a Slave.

Set up the Peer IP Addresses and Status Variable

1. Select the **Peer List** tab.



2. Enter the IP Address for each peer.
    - You must ensure that when you enter an IP Address the row number you select from the Peer column must correspond to the peer identity

number; e.g. for Network 4 (peer 1) enter the IP Address in row 1 of the peer column.

NOTE When entering an IP address for each peer, there cannot be any blank spaces between successive peers. For example, Peer 1 and Peer 3 cannot be configured without entering an IP address for Peer 2 also

- To configure a Multicast IP Address, enter the following IP Addresses:

  Ethernet Port 1: 224.1.2.3

  Ethernet Port 2: 224.4.5.6

3. Select a Boolean input Variable for the same Peer number.
   - This variable reports the peer status as TRUE when the peer is active and working or FALSE when the peer is inactive or faulty. Except for multicast peers which are always set FALSE, irrespective of whether the multicast peer is active or inactive.

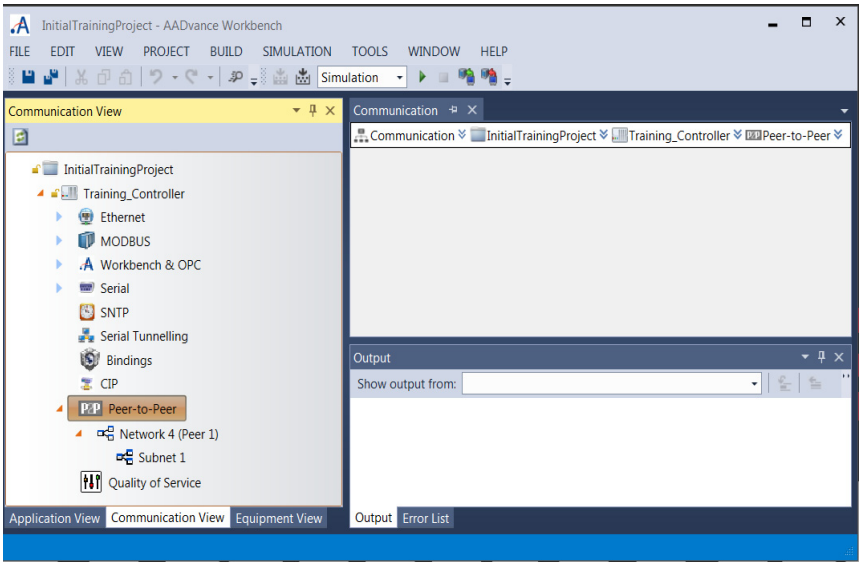
Peer-to-Peer Data Blocks

You can configure two types of input and output blocks - Analogue or Digital - and they can be large or small versions. Select the type you require to verify that the optimum communication packet size can be used for your application. Each input block has a corresponding output block that must be of the same type and channel capacity.

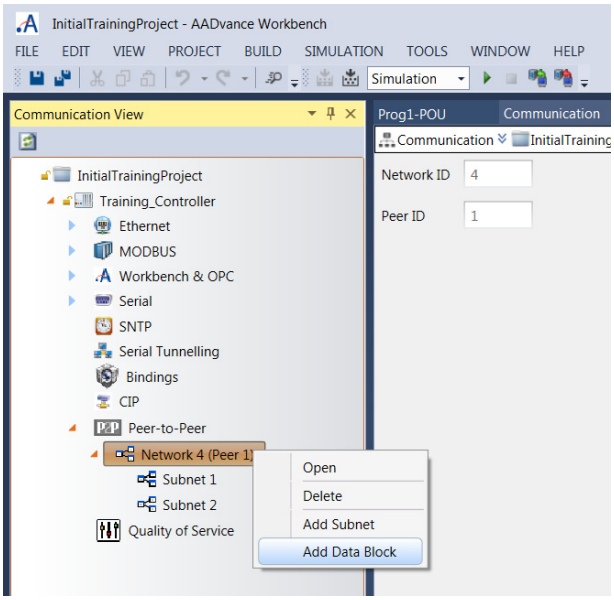
Each output block delivers data to one or more input blocks across one peer network. The subnet of the peer network used to send the data is transparent to the input and output blocks and more than one subnet can be defined to provide redundant communications.

Configure Data Blocks

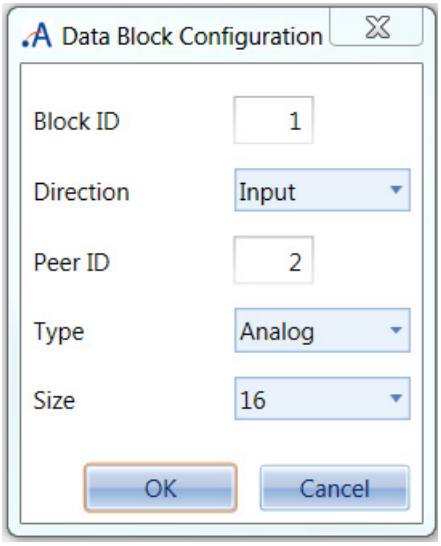
1. Right-click on the **Network 4 (Peer 1)** node.



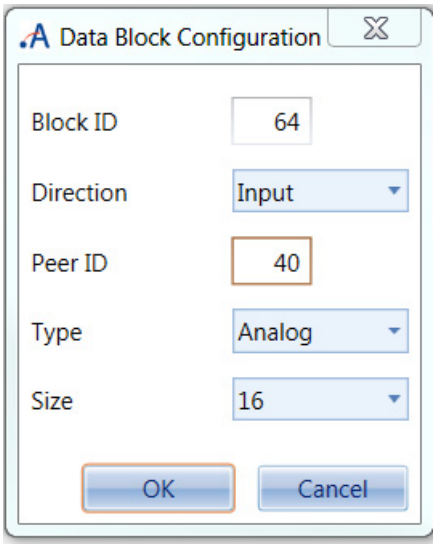
   - The following drop-down menu is revealed
2. Click on **Add Data Block** option.



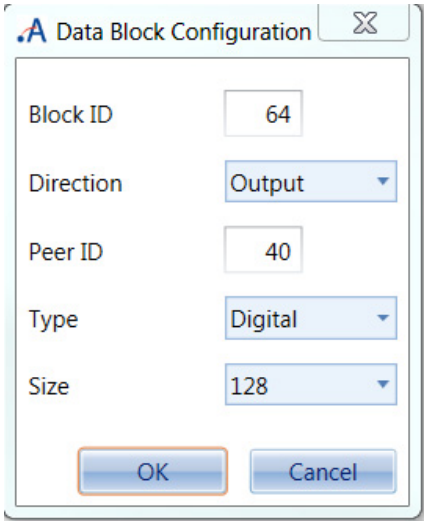
- The following dialogue box is revealed:



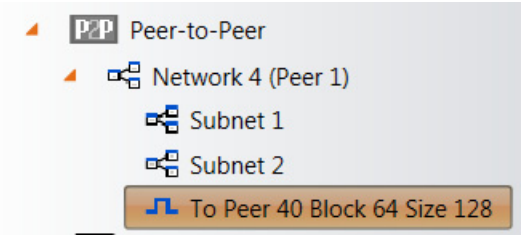
3. The Block ID has integer values from 1 to 64, the Direction has Input and Output Values, the Peer ID has values from 1 to 40, the Type is either Analog or Digital and the Size is either 16 or 128.



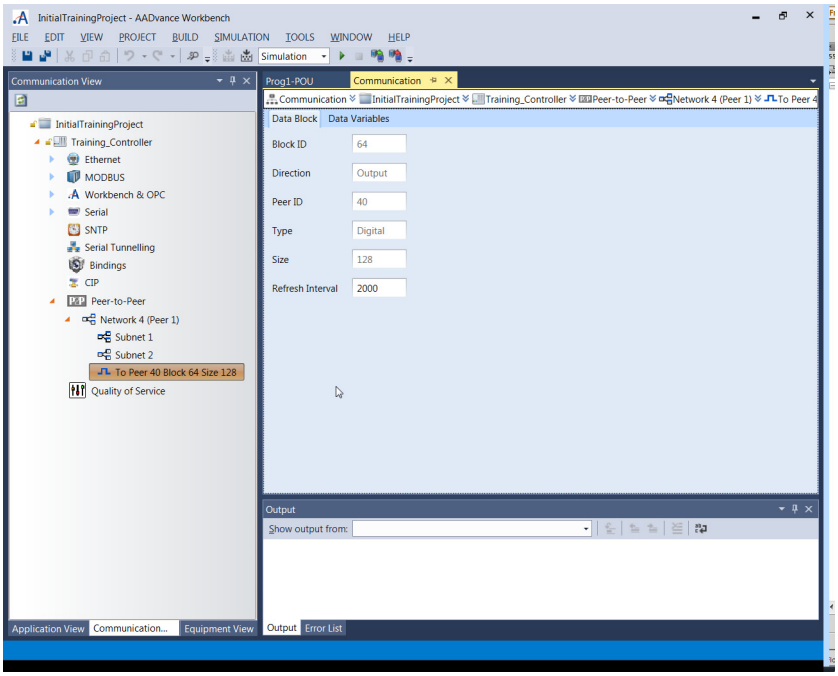
4. Below Shows the alternative Direction, Type and Size



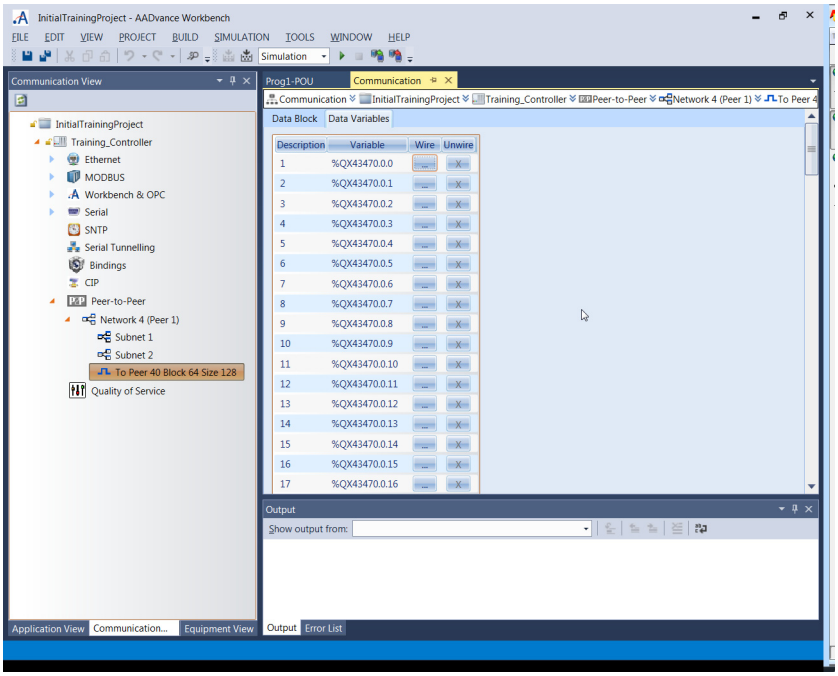
5. Selecting the **OK** button on the Data Block Configuration dialog box produces a virtual button to access the data block. This has details of the Peer ID Block ID and Size.



6. Click on the tab to select the **Data Block**.

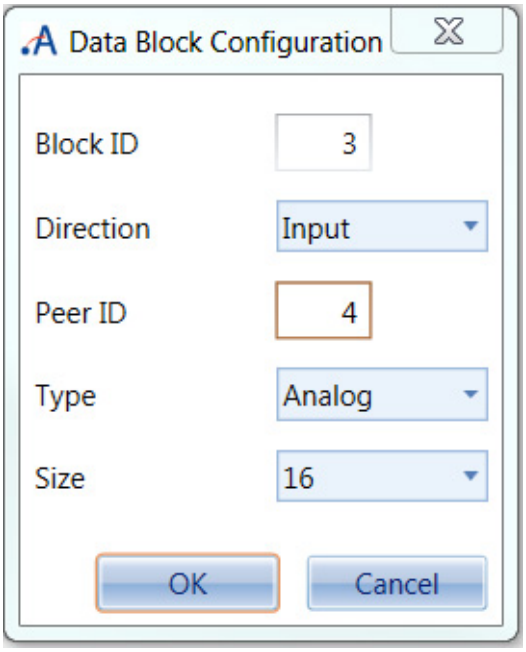


7. Selecting the **Data Variables** tab reveals the variables for the selected Data Block.



Configure Analog Input Blocks

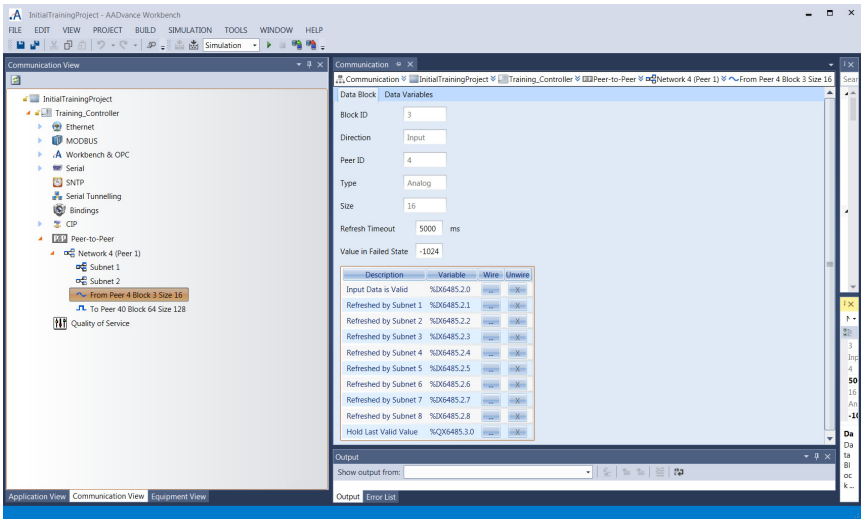
1. Create an Analog Input block.
   - Select “Input” for Direction and “Analog” for Type in the Data Block Configuration dialog box and click on **OK** button.



- The Peer-to Peer Analog Input board editor is displayed as shown below.

2. Select the **Data Block** tab.
3. Enter the following data
   - Refresh Timeout - Enter a value in the range 1 to 10000 ms. This is the maximum number of milliseconds allowed between successive refreshes of input data before the data is declared invalid. Should the time be exceed and the data declared invalid the input data will either retain the last received value or revert to a fail-safe condition according to the Hold Last Valid Value setting.
   - The Refresh Timeout should be set to the longest delay before a fail-safe action should be taken. This will be the Process Safety Time for the input data. The corresponding Output block has a Refresh Interval parameter; this controls how often data is sent to the Input block. The Refresh Interval should be calculated to deliver at least two updates within the Refresh Timeout, under worst case conditions. A Refresh Timeout which is too short for the network size and complexity will make it impossible to deliver fresh data often enough.
   - Value in Failed State - Enter a value in the range- 9.999999e+38 to + 9.999999e+38. This sets the control value to be adopted by the inputs when the input has not been refreshed within Refresh Timeout and is therefore declared invalid. Where the input corresponds to an integer the fractional part is truncated. The value is always adopted at application start-up, though it will not be used while the Hold Last Valid Value variable is TRUE.

The screen view below shows the configuration of a 16 channel analogue input block. The block being configured is for Peer Network 4, source peer is Peer 4 and the data block identity Block 3



Wire the Analogue Input Status Variables

1. Select the required row from the list shown.

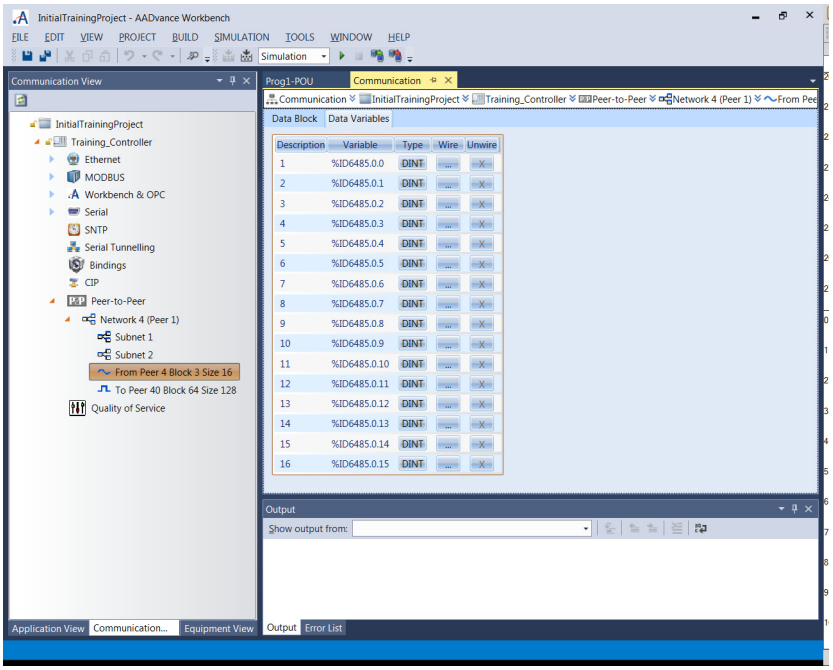
   - Select the **Wire** button ... to connect the status parameter to a variable.
   - Input Data is Valid - this status variable is used to report that the input data is refreshed within the Refresh Timeout value.
   - Refreshed by Subnet 1 to 8 - Each of these variables can be used to indicate that the data has been refreshed by Subnet 1 to 8 within the Refresh Timeout value. This status should be used to detect latent faults within a redundant network. The data is delivered over all available subnets simultaneously. If any variable goes FALSE for a programmed subnet, then the data has failed to arrive on that subnet within the Refresh Timeout value. The variables for programmed subnets may be combined through an AND gate to provide an indication of full redundancy on that particular data path.
2. Hold Last Valid Value - When it is set to FALSE it will force data to the fail safe state when the input data is invalid. When it is TRUE it will allow previous data to persist when the input data is invalid.

3. If you need to disconnect a variable select the **Unwire** X button.

Wire Analogue Input Channel Data Variables

Analogue values are received from the corresponding output of the selected output block in the sending system. The values are 32 bit and will assume either a 32 bit signed integer format or a 32 bit real format depending on the variable to which it is connected. Both the specific input channel and the corresponding output channel must be connected to the same variable type.

1. Select the **Data Variables** tab.



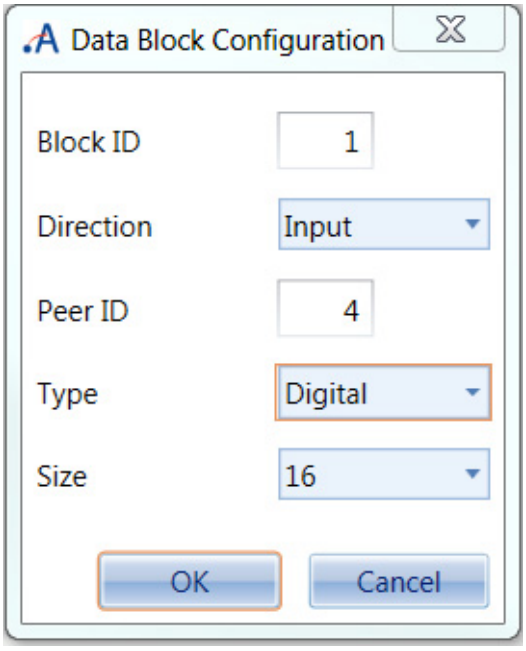
2. Select a Channel row.
3. Select the type DINT or REAL by clicking on **Toggle** button.
4. Select the **Wire** button  to connect to a variable.
5. If you need to disconnect a variable select the **Unwire** button.  .

The 128 input peer block supports 128 analogue inputs instead of 16 but is otherwise identical.

IMPORTANT The safety related data using 128 analogue channel blocks must be sent by two different input/output block pairs and compared at the receiving input end to verify safety integrity. Alternatively it may be broken into 16 channel blocks.

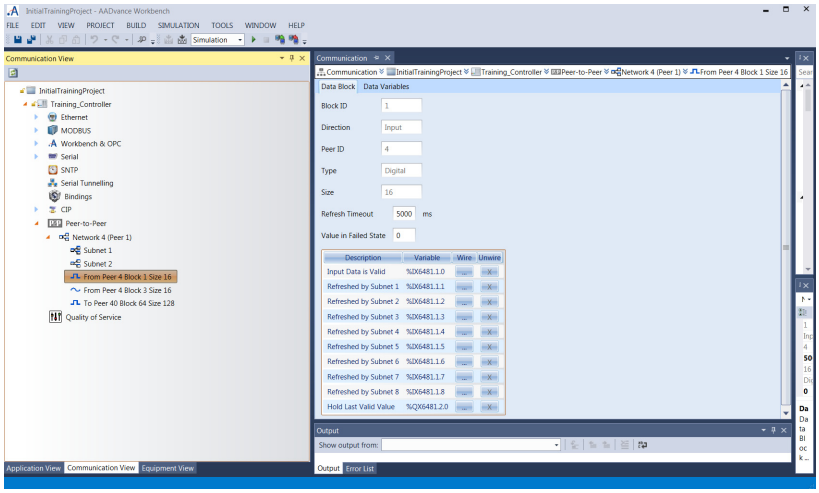
Configure Digital Input Blocks

1. Create a Digital Input Block
   - Select “Input” for Direction and “Digital” for Type in the Data Block Configuration dialog box and click on **OK** button.



- The Peer-to Peer Digital Input board editor is displayed as shown below.

2. Select the **Data Block** tab
3. Enter the data in the required fields.
   - Refresh Timeout - Enter a value in the range 1 ms to 10000 ms. This is the maximum number of milliseconds allowed between successive refreshes of input data before the data is declared invalid. Should the time be exceed and the data declared invalid the input data will either retain the last received value or revert to a fail-safe condition according to the Hold Last Valid Value setting.
   - The Refresh Timeout should be set to the longest delay before a fail-safe action should be taken. This will be the Process Safety Time for the input data. The corresponding Output block has a Refresh Interval parameter; this controls how often data is sent to the Input block. The Refresh Interval should be calculated to deliver at least two updates within the Refresh Timeout, under worst case conditions. A Refresh Timeout which is too short for the network size and complexity will make it impossible to deliver fresh data often enough.
   - Value in Failed State - Enter a value of either 0 or 1. This sets the control value to be adopted by the inputs when the input has not been refreshed within Refresh Timeout and is therefore declared invalid.



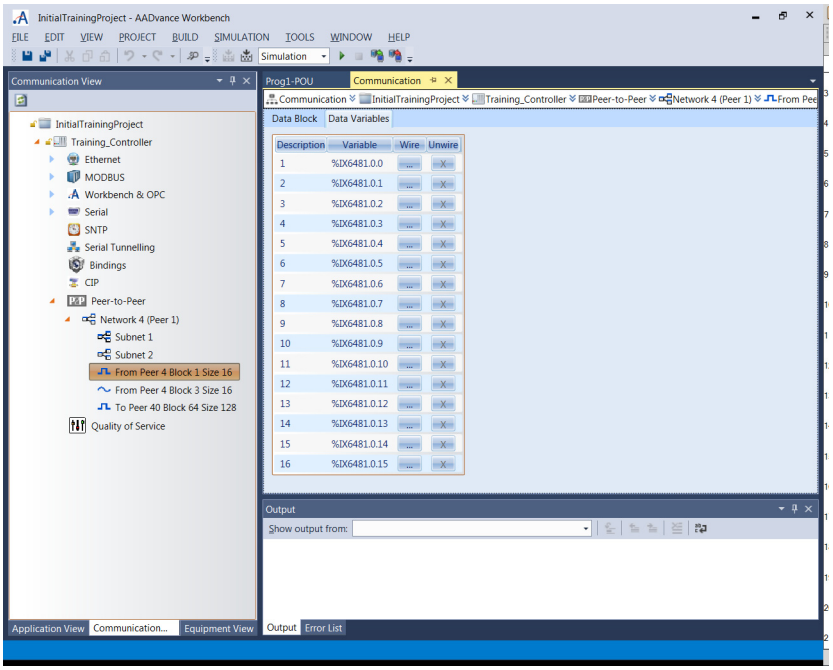
Wire Digital Input Block Status Variables

1. Select the required row from the list shown.
2. Select the **Wire** button ... to connect the status parameter to the required variables.
   - Input Data is Valid - this status variable is used to report that the input data is refreshed within the Refresh Timeout value.
   - Refreshed by Subnet 1 to 8 - Each of these variables can be used to indicate that the data has been refreshed by Subnet 1 to 8 within the Refresh Timeout value. This status should be used to detect latent faults within a redundant network. The data is delivered over all available subnets simultaneously. If any variable goes FALSE for a programmed subnet, then the data has failed to arrive on that subnet within the Refresh Timeout value. The variables for programmed subnets may be combined through an AND gate to provide an indication of full redundancy on that particular data path.
   - Hold Last Valid Value - When this variable is set to FALSE it will force data to the fail safe state when data is invalid. When it is TRUE it will allow previous data to persist when data is invalid.
3. If you need to disconnect a variable select the **Unwire** button. X .

Wire Digital Input Channel Data Variables

Digital input values are received from the corresponding output of the selected output block in the sending system; the values are Boolean values. Both the specific input channel and the corresponding output channel must be connected to the same variable type.

1. Select the **Data Variables** tab.

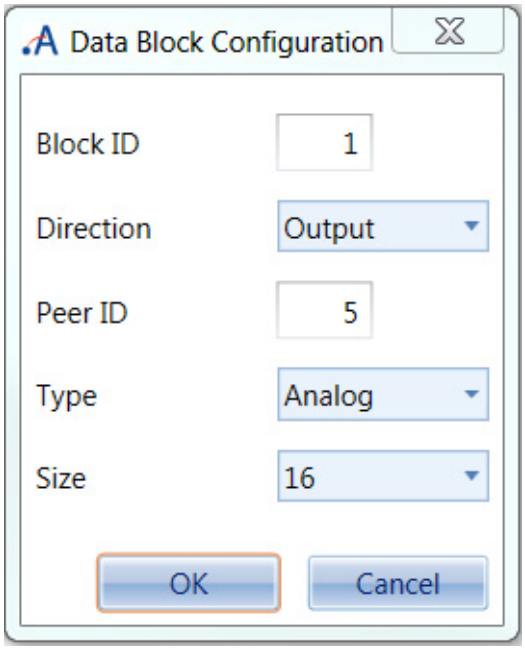


2. Select a Channel row.
3. Select the **Wire** button  to connect to a variable.
4. If you need to disconnect a variable select the **Unwire** button  .

The 128 input peer block supports 128 analogue inputs instead of 16 but is otherwise identical.

Configure Analogue Output Blocks

1. Create the Analog Output block.
   - Select “Output” for Direction and “Analog” for Type in the Data Block Configuration dialog box and click on **OK** button.

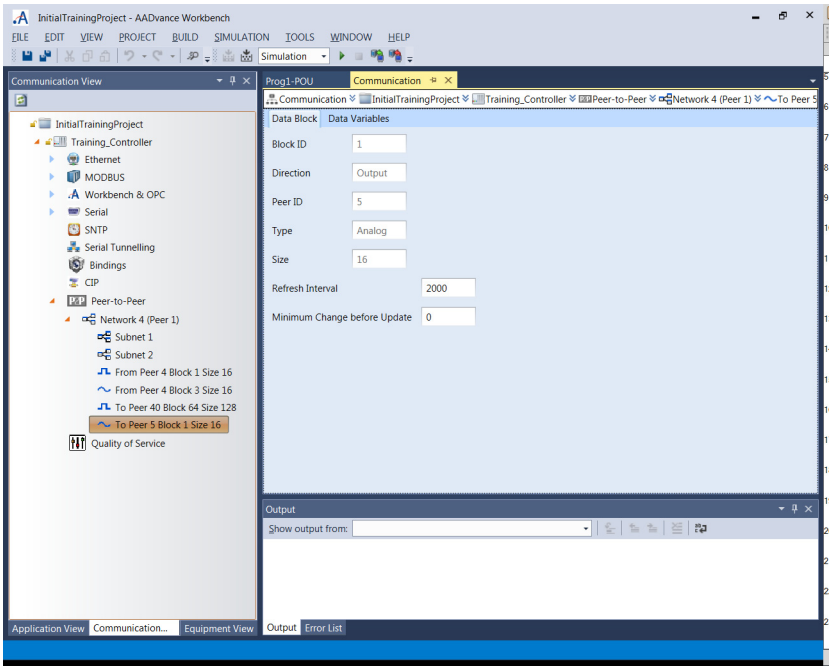


- The Peer-to Peer Analog Output board editor is displayed as shown below.

2. Select the **Data Block** tab.
3. Enter the following data.
   - Refresh Interval - Enter a value in the range 0 – 10000 ms. This is the maximum time allowed between transmissions of the output data.
   - Data will be sent immediately following any change of output state. If a value of zero is specified in this field then data will be refreshed every application scan regardless of output state change.
   - The Refresh Interval should be calculated to deliver at least two updates within the Refresh Timeout of the corresponding Input block, under worst case conditions. A Refresh Timeout which is too short for the network size and complexity will make it impossible to deliver fresh data often enough.
   - The setting for the Refresh Interval should be less than:

     (Input block Refresh Timeout - Worst case Delivery Delay) / 2
   - The Worst Case Delivery Delay is:

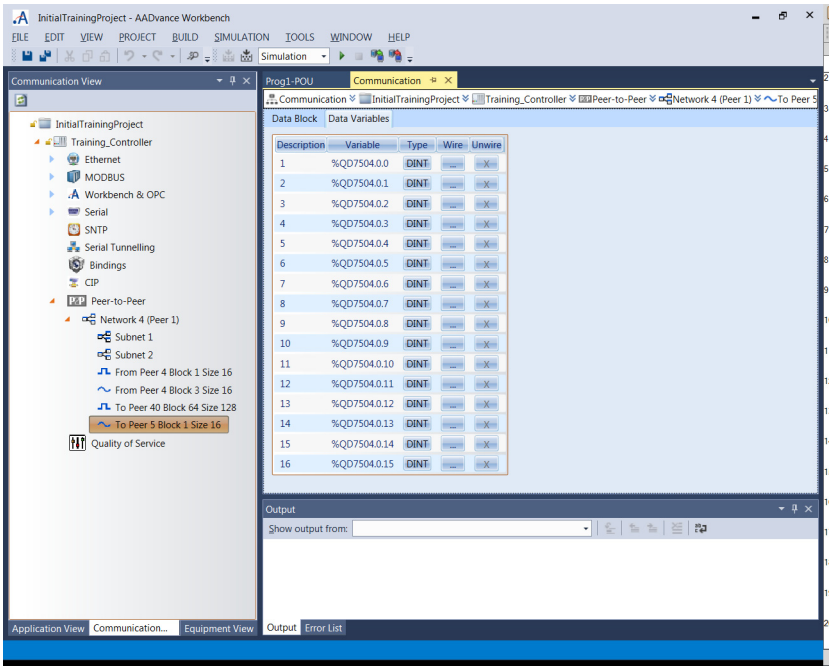
     Refresh Timeout (from the subnet control board) + Scan time of this controller + Scan time of receiving controller + 50 ms
   - Minimum Change before Update - Enter a value in the range 0 to 9.999999e+038. This value sets the minimum change in any output variable before the update is sent to the Peer input block (excluding any refresh interval). When applied to integers the fractional part is truncated.



Wire the Analogue Output Channel Variables

Analogue output channels send analogue data to the corresponding analogue input channels on the peer network.

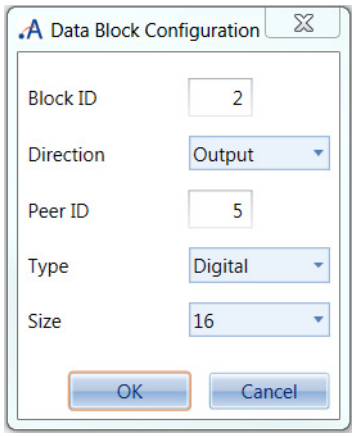
1. Select the **Data Variables** tab.



2. Select a Channel row.
3. Select the Type DINT or REAL by clicking the toggle button.
4. Select the **Wire** button ... to connect to a variable.
5. If you need to disconnect a variable select **Unwire** button X .

Configure Digital Output Blocks

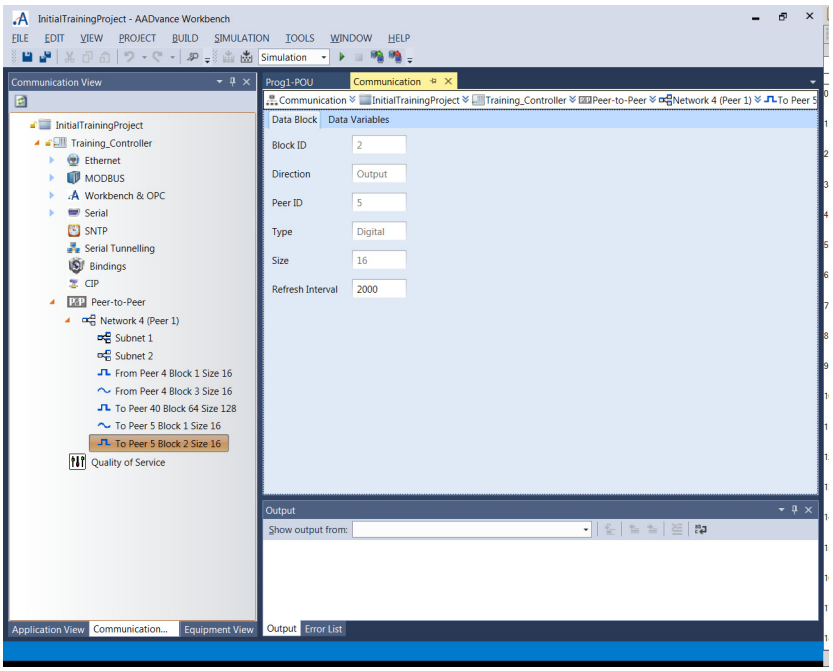
1. Create the Digital Output block
   - Select “Output” for Direction and “Digital” for Type in the Data Block Configuration dialog box and click on **OK** button



   - The Peer-to Peer Digital Output board editor is displayed as shown below.
2. Select the **Data Block** tab.
3. Enter the following data.
   - Refresh Interval - Enter a value in the range 0 – 10000 ms. This is the maximum time allowed between transmissions of the output data.
   - Data will be sent immediately following any change of output state. If a value of zero is specified in this field then data will be refreshed every application scan regardless of output state change.
   - The Refresh Interval should be calculated to deliver at least two updates within the Refresh Timeout of the corresponding Input block, under worst case conditions. A Refresh Timeout which is too short for the network size and complexity will make it impossible to deliver fresh data often enough.
   - The setting for the Refresh Interval should be less than:

     (Input block Refresh Timeout - Worst case Delivery Delay) / 2
   - The Worst Case Delivery Delay is:

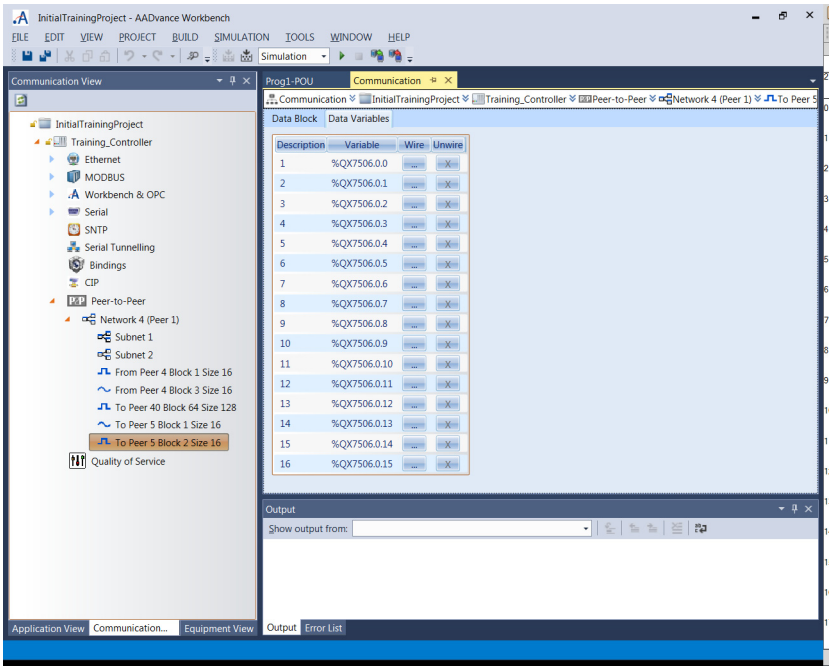
     Refresh Timeout (from the subnet control board) + Scan time of this controller + Scan time of receiving controller + 50 ms.



Wire the Digital Output Channel Data Variables

The digital output blocks send digital data to the corresponding digital input blocks on the peer network.

1. Select the **Data Variables** tab.

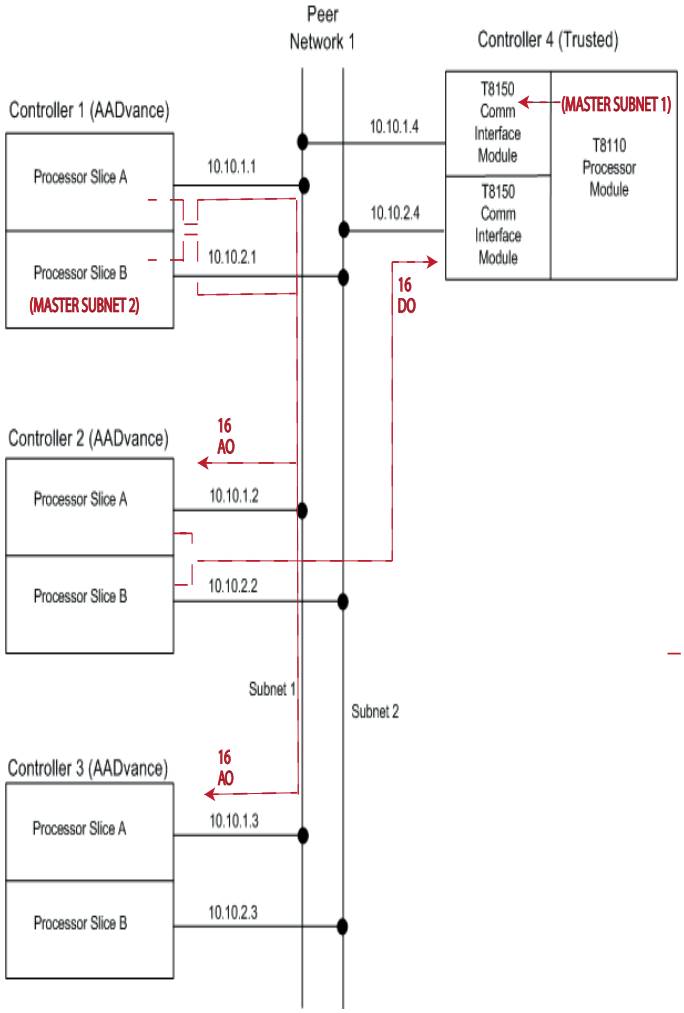


2. Select a Channel row.
3. Select a **Wire** button ... to connect to a variable.
   - The outputs are Boolean values.
4. To disconnect a variable select the **Unwire** button X .

Peer-to-Peer Configuration Example 1

This example uses 4 controllers connected to two subnets forming one Peer-to-Peer network.

- Controllers 1, 2 and 3 are dual processor AADvance controllers and Controller 4 is a Trusted Controller.
- Controller 4 is designated the "master" of Network 1 Subnet 1 and Controller 1 the "master" of Network 1 Subnet 2.



Peer-to-Peer Controller Setting Summary

You should follow the process described in this chapter.

The network configuration is as follows:

- 16 analogue values are sent from Controller 1 to Controller 2
- 16 analogue values are sent form Controller 1 to Controller 3
- 16 digital values are sent from Controller 2 to Controller 4

Table 25 - Controller 1 - Dual Peer-to-Peer Net Control, Network 1, Subnet 1

	Value	Comment
Subnet 1	x	
Network Interface	A	Network connected to processor A
Network Identity	1	Network 1
Subnet Identity	1	Subnet 1

Table 25 - Controller 1 - Dual Peer-to-Peer Net Control, Network 1, Subnet 1

	Value	Comment
Peer ID	1	Identity of this controller
Response Timeout	0	Default
Refresh Timeout	2000	Default
Slave Transmit timeout	500	Default
Enable Controller	TRUE	Enable Peer on this subnet
Master/Slave	FALSE	This is a slave connection
Peer List:		
Peer 1	10.10.1.1	Controller 1, network 1, subnet 1
Peer 2	10.10.1.2	Controller 2, network 1, subnet 1
Peer 3	10.10.1.3	Controller 3, network 1, subnet 1
Peer 4	10.10.1.4	Controller 4, network 1, subnet 1

Table 26 - Controller 1 - Dual Peer-to-Peer Net control, network 1 subnet 2

	Value	Comment
Subnet 2		
Network Interface	B	Network connected to processor B
Network Identity	1	Network 1
Subnet Identity	2	Subnet 2
Peer ID	1	Identity of this controller
Response Timeout	0	Default
Refresh Timeout	2000	Default
Slave Transmit timeout	500	Default
Enable Controller	TRUE	Enable Peer on this subnet
Master/Slave	TRUE	This is a Master connection
Peer List:		
Peer 1	10.10.2.1	Controller 1, network 1, subnet 2
Peer 2	10.10.2.2	Controller 2, network 1, subnet 2
Peer 3	10.10.2.3	Controller 3, network 1, subnet 2
Peer 4	10.10.2.4	Controller 4, network 1, subnet 2

Network Set Up - Network 1, Subnet 1

The following are the values that should be entered during the set up process to configure an example Peer-to-Peer network.

Table 27 - Controllers 1, 2 and 3

	Controller 1	Controller 2	Controller 3
Subnet 1			
Network interface	A	A	A
Network Identity	1	1	1
Subnet Identity	1	1	1
Peer ID	1	2	3
Response Timeout	0	0	0
Refresh Timeout	2000	2000	2000
Slave Transmit Timeout	500	500	500
Enable Controller	TRUE	TRUE	TRUE
Master Slave	FALSE	FALSE	FALSE
Peer List:			
Peer 1	10.10.1.1	10.10.1.1	10.10.1.1

Table 27 - Controllers 1, 2 and 3

	Controller 1	Controller 2	Controller 3
Peer 2	10.10.1.2	10.10.1.2	10.10.1.2
Peer 3	10.10.1.3	10.10.1.3	10.10.1.3
Peer 4	10.10.1.4	10.10.1.4	10.10.1.4

Table 28 - Controller 4

	Controller 4
dxpnc40 - Control Rack	
Chassis	1
Slot	7
Network Id	1
Subnet_ID	1
Peer ID	4
Response_TMO	0
Refresh_TMO	2000
TX_DATA_TMO	500
Variable 1	TRUE
Variable 2	TRUE
Peers_1 rack:	
Peer_IP_01	10.10.1.1
Peer_IP_02	10.10.1.2
Peer_IP_03	10.10.1.3
Peer_IP_04	10.10.1.4

Dual Peer-to-Peer Net control Network Set Up - Network 1, Subnet 2

Table 29 - Controllers 1, 2 & 3

	Controller 1	Controller 2	Controller 3
Subnet 2			
Network interface	B	B	B
Network Identity	1	1	1
Subnet Identity	2	2	2
Peer ID	1	2	3
Response Timeout	0	0	0
Refresh Timeout	2000	2000	2000
Slave Transmit Timeout	500	500	500
Enable Controller	TRUE	TRUE	TRUE
Master Slave	TRUE	FALSE	FALSE
Peer List:			
Peer 1	10.10.2.1	10.10.2.1	10.10.2.1
Peer 2	10.10.2.2	10.10.2.2	10.10.2.2
Peer 3	10.10.2.3	10.10.2.3	10.10.2.3
Peer 4	10.10.2.4	10.10.2.4	10.10.2.4

Table 30 - Controller 4

	Controller 4
dxpnc40 - Control Rack	
Chassis	1
Slot	8
Network Id	1
Subnet_ID	2

Table 30 - Controller 4

	Controller 4
Peer ID	4
Response_TMO	0
Refresh_TMO	2000
TX_DATA_TMO	500
Variable 1	TRUE
Variable 2	FALSE
Peers_1 rack:	
Peer_IP_01	10.10.2.1
Peer_IP_02	10.10.2.2
Peer_IP_03	10.10.2.3
Peer_IP_04	10.10.2.4

Peer-to-Peer Data Summary

Output Data

Table 31 - Controller 1 & 2

	Controller 1		Controller 2
Data Block Type	Analog Output	Analog Output	Digital Output
Network identity	1	1	1
Target peer identity	2	3	4
Source data identity	1	2	1
Refresh timeout	2000	2000	2000
Minimum change before update	20	20	N/A
Variable 1 -16	Analogue Data Output	Analogue Data Output	Digital Data Output

For input data see next page.

Input Data

Table 32 - Controller 2 & 3

	Controller 2	Controller 3
Data Block Type	Analog Input	Analog Input
Network identity	1	1
Source peer identity	1	1
Source data identity	1	2
Refresh timeout	5000	5000
Value in failed state	-1024	-1024
Input Data is valid	Input data valid	Input data valid
Refreshed by subnet 1...8	Refreshed on subnet 1...8	Refreshed on subnet 1...8
Host Last Valid value	Failure Action	Failure action
Data Variables Tab		
Variable 1 - 16	Analogue Data Input	Analogue Data Input

Table 33 - Controller 4

	Controller 4
dxpnc40 - DATA RACK	Digital Input
Network Identity	1
Source Peer Identity	2

Table 33 - Controller 4

	Controller 4
Source data identity	1
Refresh timeout	5000
Value in failed state	FALSE
STATUS RACK	
Variable 1	Boolean Data Input
Input Data is Valid	Input Data Valid
Refreshed by Subnet 1...9	Refreshed on Subnet 1...8
CONTROL RACK	
Variable 1	Failure Action

Notes:

History of Changes

This appendix contains the new or updated information for each revision of this publication. These lists include substantive updates only and are not intended to reflect all changes. Translated versions are not always available for each revision.

ICSTT-RM458D-EN-P, February 2021

Change
Updated for AADvance® system release 1.40 TÜV Rheinland certification
Added reference to AADvance®-Trusted® SIS Workstation Software User Guide, publication ICSTT-UM002
Updated publication template
Removed **NFPA 87** from Glossary.
Updated **proof test** definition in Glossary section.

ICSTT-RM458C-EN-P, July 2019

Change
Updated for Release 1.34 IEC 61508 Edition 2.0 certification

ICSTT-RM458B-EN-P, April 2018

Change
Update for R1.4 - Workbench 2.1 - Document No. ICSTT-RM458B-EN-P

Issue 09, March 2015

Change
Update for R1.34

Issue 08, February 2015

Change
Update for R1.32

Issue 07, June 2013

Change
Update to add information to online update topics

Issue 06, June 2012

Change
Release 1.3 & 1.3.1

Issue 05, April 2012

Change
Updated Release 1.2 version with Analogue Output Module information added.

Issue 04 October 2011

Change
Release 1.2

Issue 03, July 2010

Change
Update for CRs

Issue 02, November 2009

Change
Release 1.1.1

Issue 01B, August 2009

Change
Updated issue for per review comments

Issue 01A, August 2009

Change
Release 1.1 Issue

Issue 01, January 2009

Change
First Issue

The following terms and abbreviations are used throughout this manual. For definitions of terms not listed here, refer to the Allen-Bradley Industrial Automation Glossary, publication AG-7.1.

A

accuracy The degree of conformity of a measure to a standard or a true value. See also 'resolution'.

achievable safe state A safe state that is achievable.

> **NOTE** Sometimes, a safe state cannot be achieved. An example is a non-recoverable fault such as a voting element with a shorted switch and no means to bypass the effect of the short.

actuator A device which cause an electrical, mechanical or pneumatic action to occur when required within a plant component. Examples are valves and pumps.

AITA Analogue input termination assembly.

alarms and events (AE) An OPC data type that provides time stamped alarm and event notifications.

allotted process safety time The portion of the total process safety time allotted to a sub function of that process.

application software Software specific to the user application, typically using logic sequences, limits and expressions to read inputs, make decisions and control outputs to suit the requirements of the system for functional safety.

architecture Organizational structure of a computing system which describes the functional relationship between board level, device level and system level components.

asynchronous A data communications term describing a serial transmission protocol. A start signal is sent before each byte or character and a stop signal is sent after each byte or character. An example is ASCII over RS-232-C. See also 'RS-232-C, RS-422, RS-485'.

availability The probability that a system will be able to carry out its designated function when required for use — normally expressed as a percentage.

B

backplane clip A sprung, plastic device to hold together two adjacent AADvance® base units. Part number 9904. Used in pairs.

base unit One of two designs which form the supporting parts of an AADvance controller. See 'I/O base unit' and 'processor base unit'.

bindings Bindings describe a "relationship" between variables in different AADvance controllers. Once a variable is "bound" to another variable, a unique and strong relationships is created between the two variables and the SIL 3 Certified SNCP protocol is used to ensure that the consuming variable is updated with the data from the producing variable.

black channel A communication path whose layer (i.e. cabling, connections, media converters, routers/switches and associated firmware/software, etc.) has no requirement to maintain the integrity of safety critical data transferred over it. Measures to detect and compensate for any errors introduced into the black channel must be implemented by the safety critical sender and receiver (by software and/or hardware means) to make sure the data retains its integrity.

blanking cover A plastic moulding to hide an unused slot in an AADvance base unit.

boolean A type of variable that can accept only the values 'true' and 'false'.

BPCS Basic process control system. A system which responds to input signals and generates output signals causing a process and associated equipment to operate in a desired manner, but which does not perform any safety instrumented functions with a claimed safety integrity level of 1 or higher.

Refer to IEC 61511 or to ANSI/ISA—84.00.01—2004 Part 1 (IEC 61511-1 Mod) for a formal definition.

Equivalent to the Process Control System (PCS) defined by IEC 61508.

breakdown voltage The maximum voltage (AC or DC) that can be continuously applied between isolated circuits without a breakdown occurring.

BS EN 54 A standard for fire detection and fire alarm systems.

BS EN 60204 A standard for the electrical equipment of machines, which promotes the safety of persons and property, consistency of control response and ease of maintenance.

bus A group of conductors which carry related data. Typically allocated to address, data and control functions in a microprocessor-based system.

bus arbitration A mechanism for deciding which device has control of a bus.

C

CIP Common Industrial Protocol. A communications protocol, formally known as 'CIP over Ethernet/IP', created by Rockwell Automation for the Logix controller family, and which is also supported by the AADvance controller. AADvance controllers use the protocol to exchange data with Logix controllers. The data exchange uses a consumer/producer model.

clearance The shortest distance in air between two conductive parts.

coding peg A polarization key, fitted to the 9100 processor base unit and to each termination assembly, which verifies that only a module of the correct type may be fitted in a particular slot. Part number 9903.

coil In IEC 61131-3, a graphical component of a Ladder Diagram program, which represents the assignment of an output variable. In Modbus language, a discrete output value.

Compiler Verification Tool (CVT) The Compiler Verification Tool (CVT) is an automatic software utility that validates the output of the application compilation process. This process, in conjunction with the validated execution code produced by the AADvance Workbench, provides a high degree of confidence that there are no errors introduced by the Workbench or the compiler during the compilation of the application.

configuration A grouping of all the application software and settings for a particular AADvance controller. The grouping must have a 'target', but for an AADvance controller it can have only one 'resource'.

consumer The consuming controller requests the tag from the producing controller.

contact A graphical component of a Ladder Diagram program, which represents the status of an input variable.

continuous mode Where the Safety Instrumented Function in the Safety System is continually maintaining the process in a safe state.

controller A logic solver; the combination of application execution engine and I/O hardware.

controller system One or more controllers, their power sources, communications networks and workstations.

coverage The percentage of faults that will be detected by automated diagnostics. See also 'SFF'.

creepage distance The shortest distance along the surface of an insulating material between two conductive parts.

cross reference Information calculated by the AADvance Workbench relating to the dictionary of variables and where those variables are used in a project.

D

data access (DA) An OPC data type that provides real-time data from AADvance controllers to OPC clients.

de-energize to action A safety instrumented function circuit where the devices are energized under normal operation. Removal of power de-activates the field devices.

dictionary The set of internal input and output variables and defined words used in a program.

discrepancy A condition that exists if one or more of the elements disagree.

DITA Digital input termination assembly.

DOTA Digital output termination assembly.

E

element A set of input conditioning, application processing and output conditioning.

energize to action A safety instrumented function circuit where the outputs and devices are de-energized under normal operation. Application of power activates the field device.

EUC Equipment Under Control. The machinery, apparatus or plant used for manufacturing, process, transportation, medical or other activities.

expansion cable assembly A flexible interconnection carrying bus signals and power supplies between AADvance base units, available in a variety of lengths. Used in conjunction with a cable socket assembly (at the left hand side of a base unit) and a cable plug assembly (at the right hand side of a base unit).

F

fail operational state A state in which the fault has been masked. See 'fault tolerant'.

fail safe The capability to go to a pre-determined safe state in the event of a specific malfunction.

fault reset button The momentary action push switch located on the front panel of the 9110 processor module.

fault tolerance Built-in capability of a system to provide continued correct execution of its assigned function in the presence of a limited number of hardware and software faults.

fault tolerant The capability to accept the effect of a single arbitrary fault and continue correct operation.

fault warning receiving station A centre from which the necessary corrective measures can be initiated.

fault warning routing equipment Intermediate equipment which routes a fault warning signal from the control and indicating equipment to a fault warning receiving station.

field device Item of equipment connected to the field side of the I/O terminals. Such equipment includes field wiring, sensors, final control elements and those operator interface devices hard-wired to I/O terminals.

fire alarm device A component of a fire alarm system, not incorporated in the control and indicating equipment which is used to give a warning of fire — for example a sounder or visual indicator.

fire alarm receiving station A centre from which the necessary fire protection or fire fighting measures can be initiated at any time.

fire alarm routing equipment Intermediate equipment which routes an alarm signal from control and indicating equipment to a fire alarm receiving station.

function block diagram An IEC 61131 language that describes a function between input variables and output variables. Input and output variables are connected to blocks by connection lines. See 'limited variability language'.

functional safety The ability of a system to carry out the actions necessary to achieve or to maintain a safe state for the process and its associated equipment.

G

group A collection of two or three input modules (or two output modules), arranged together to provide enhanced availability for their respective input or output channels.

H

hand-held equipment Equipment which is intended to be held in one hand while being operated with the other hand.

HART HART (Highway Addressable Remote Transducer) is an open protocol for process control instrumentation. It combines digital signals with analogue signals to provide field device control and status information. The HART protocol also provides diagnostic data. (For more details of HART devices refer to the HART Application Guide, created by the HART Communication Foundation, and their detailed HART specifications. You can download documents from www.hartcomm.org.)

high demand mode Where the Safety Instrumented Function in the Safety System only performs its designed function on a demand, and the frequency of demands is greater than one per year.

hot swap See live insertion.

I

I/O base unit A backplane assembly which holds up to three I/O modules and their associated termination assembly or assemblies in an AADvance controller. Part number 9300. See 'I/O module' and 'termination assembly'.

I/O module A collation of interfaces for field sensors (inputs) or final elements (outputs), arranged in a self-contained and standardized physical form factor.

IEC 61000 A series of international standards giving test and measurement techniques for electromagnetic compatibility.

IEC 61131 An international standard defining programming languages, electrical parameters and environmental conditions for programmable logic controllers. Part 3, which is entitled 'Programming Languages', defines several limited variability languages.

IEC 61508 An international standard for functional safety, encompassing electrical, electronic and programmable electronic systems; hardware and software aspects.

IEC 61511 An international standard for functional safety and safety instrumented systems (SIS) for the process industry, encompassing electrical, electronic and programmable electronic systems, hardware and software aspects.

indicator A device which can change its state to give information.

input (Workbench variable) In the context of an AADvance Workbench variable, this term describes a quantity passed to the Workbench from a controller.

instruction list An IEC 61131 language, similar to the simple textual language of PLCs. See 'limited variability language'.

integer A variable type defined by the IEC 61131 standard.

IXL IXL stands for ISaGRAF® eXchange Layer. This is the communication protocol between ISaGRAF based components.

K

key connector The receptacle on the AADvance controller for the program enable key. A 9-way 'D' type socket, located on the 9100 processor base unit.

L

ladder diagram An IEC 61131 language composed of contact symbols representing logical equations and simple actions. The main function is to control outputs based on input conditions. See 'limited variability language'.

LAN Local area network. A computer network covering a small physical area, characterised by a limited geographic range and lack of a need for leased telecommunication lines.

live insertion The removal and then reinsertion of an electronic module into a system while the system remains powered. The assumption is that removal of the module and reinsertion will cause no electrical harm to the system. Also referred to as 'hot swap'.

low demand mode Where the Safety Instrumented Function only performs its designed function on demand, and the frequency of demands is no greater than one per year.

M

manual call point A component of a fire detection and fire alarm system which is used for the manual initiation of an alarm.

Modbus An industry standard communications protocol developed by Modicon. Used to communicate with external devices such as distributed control systems or operator interfaces.

Modbus object A representation of the configuration settings for a Modbus master or for its associated slave links, within the AADvance Workbench. The settings include communication settings and messages.

module locking screw The AADvance latch mechanism seen on the front panel of each module and operated by a broad, flat-blade screwdriver. Uses a cam action to lock to the processor base unit or I/O base unit.

N

NFPA 85 The Boiler and Combustion Systems Hazards Code. Applies to certain boilers, stokers, fuel systems, and steam generators. The purpose of this code is to contribute to operating safety and to help prevent uncontrolled fires, explosions and implosions.

NFPA 86 A standard for Ovens and Furnaces. Provides the requirements for the prevention of fire and explosion hazards in associated with heat processing of materials in ovens, furnaces and related equipment.

O

online The state of a controller that is executing the application software.

OPC A series of standards specifications which support open connectivity in industrial automation.

output (Workbench variable) In the context of an AADvance Workbench variable, this term describes a quantity passed from the Workbench to a controller.

P

peer to peer A Peer to Peer network consists of one or more Ethernet networks connecting together a series of AADvance and/or Trusted® controllers to enable application data to be passed between them.

pinging In Modbus communications, sending the diagnostic Query Data command over a link and by receiving a reply ensuring that the link is healthy and the controller is able to communicate with the master. No process data is transferred or modified. In the case of slave devices that will not support pinging then the Standby command will default to Inactive state, but no error will be returned.

portable equipment Enclosed equipment that is moved while in operation or which can easily be moved from one place to another while connected to the supply. Examples are programming and debugging tools and test equipment.

process safety time (PST) For equipment under control this represents the period of time a dangerous condition can exist without the protection of a safety instrumented system before a hazardous event occurs.

processor base unit A backplane assembly which holds all of the processor modules in an AADvance controller. Part number 9100. See also 'processor module'.

processor module The application execution engine of the AADvance controller, housed in a self-contained and standardized physical form factor.

producer A controller producing a tag to one or more consumers, at the request of the consumers.

program enable key A security device that protects the application from unauthorized access and change, in the form factor of a 9-way 'D' type plug. Part number 9906. Supplied with the processor base unit. See also 'key connector'.

project A collection of configurations and the definition of the linking between them. See 'configuration'.

proof test A periodic test performed to detect dangerous hidden faults in a safety instrumented system (SIS) so that, if necessary, a repair can restore the system to an 'as new' condition or as close as practical to this condition.

Proof tests are designed to reveal both Systematic and Random failures, Proof tests may be required depending on how the technology has been implemented.

AADvance product data is given for a Useful Life of 20 years. For a Mission Time of up to 20 Years, proof testing is not required. For Mission Times greater than 20 years, any products that are still in service once that time is reached should be replaced.

protocol A set of rules that is used by devices (such as AADvance controllers, serial devices and engineering workstations) to communicate with each other. The rules encompass electrical parameters, data representation, signalling, authentication, and error detection. Examples include Modbus, TCP and IP.

PST Process Safety Time. The process safety time for the equipment under control (denoted PSTEUC) is the period a dangerous condition can exist before a hazardous event occurs without a safety system as a protection.

R

real A class of analogue variable stored in a floating, single-precision 32-bit format.

redundancy The use of two or more devices, each carrying out the same function, to improve reliability or availability.

resolution The smallest interval measurable by an instrument; the level of detail which may be represented. For example, 12 bits can distinguish between 4096 values.

RS-232-C, RS-422, RS-485 Standard interfaces introduced by the Electronic Industries Alliance covering the electrical connection between data communication equipment. RS-232-C is the most commonly used interface; RS-422 and RS-485 allow for higher transmission rates over increased distances.

RTC Real-time clock.

RTU Remote terminal unit. The Modbus protocol supported by the AADvance controller for Modbus communications over serial links, with the ability to multi-drop to multiple slave devices.

S

safe state A state which enables the execution of a process demand. Usually entered after the detection of a fault condition; it makes sure the effect of the fault is to enable rather than disable a process demand.

safety accuracy The accuracy of a signal within which the signal is guaranteed to be free of dangerous faults. If the signal drifts outside of this range, it is declared faulty.

safety-critical state A faulted state which helps prevent the execution of a process demand.

sensor A device or combination of devices that measure a process condition. Examples are transmitters, transducers, process switches and position switches.

sequential function chart An IEC 61131 language that divides the process cycle into a number of well-defined steps separated by transitions. See 'limited variability language'.

SFF Safe Failure Fraction. Given by (the sum of the rate of safe failures plus the rate of detected dangerous failures) divided by (the sum of the rate of safe failures plus the rate of detected and undetected dangerous failures).

SIF Safety Instrumented Function. A form of process control that performs specified functions to achieve or maintain a safe state of a process when unacceptable or dangerous process conditions are detected.

SIL Safety Integrity Level. One of four possible discrete levels, defined in IEC 61508 and IEC 61511, for specifying the safety integrity requirements of the safety functions to be allocated to a safety-related system. SIL4 has the highest level of safety integrity; SIL1 has the lowest.

The whole of an installation (of which the AADvance system forms a part) must meet these requirements in order to achieve an overall SIL rating.

SNCP SNCP (Safety Network Control Protocol) is the Safety Protocol that allows elements of an AADvance System to exchange data. SNCP is a SIL 3 certified protocol which provides a safety layer for the Ethernet network making it a "Black Channel".

SNTP Simple Network Time Protocol. Used for synchronizing the clocks of computer systems over packet-switched, variable-latency data networks.

structured text A high level IEC 61131-3 language with syntax similar to Pascal. Used mainly to implement complex procedures that cannot be expressed easily with graphical languages.

synchronous A data communications term describing a serial transmission protocol. A pre-arranged number of bits is expected to be sent across a line per second. To synchronise the sending and receiving machines, a clocking signal is sent by the transmitting computer. There are no start or stop bits.

T

TA See 'termination assembly'.

target An attribute of a 'configuration' which describes characteristics of the AADvance controller on which the configuration will run. Includes characteristics such as the memory model and the sizes of variable types for the controller.

TCP Transmission control protocol. One of the core protocols of the Internet Protocol suite. It provides reliable, ordered delivery of a stream of bytes from a program on one computer to another program on another computer. Common applications include the World Wide Web, e-mail and file transfer and, for an AADvance controller, Modbus communications over Ethernet.

termination assembly A printed circuit board which connects field wiring to an input or output module. The circuit includes fuses for field circuits. The board carries screw terminals to connect field wiring to the controller, and the whole assembly clips onto the 9300 I/O base unit.

TMR Triple modular redundant. A fault tolerant arrangement in which three systems carry out a process and their result is processed by a voting system to produce a single output.

TÜV certification Independent third party certification against a defined range of international standards including IEC 61508.

U

U Rack unit. A unit of measure used to describe the height of equipment intended for mounting in a standard rack. Equivalent to 44.45mm (1-¾ inches).

V

validation In quality assurance, confirmation that the product does what the user requires.

verification In quality assurance, confirmation that the product conforms to the specifications.

voting system A redundant system (m out of n) which requires at least m of the n channels to be in agreement before the system can take action.

W

withstand voltage The maximum voltage level that can be applied between circuits or components without causing a breakdown.

Notes:

Rockwell Automation Support

Use these resources to access support information.

Technical Support Center	Find help with how-to videos, FAQs, chat, user forums, and product notification updates.	rok.auto/support
Knowledgebase	Access Knowledgebase articles.	rok.auto/knowledgebase
Local Technical Support Phone Numbers	Locate the telephone number for your country.	rok.auto/phonesupport
Literature Library	Find installation instructions, manuals, brochures, and technical data publications.	rok.auto/literature
Product Compatibility and Download Center (PCDC)	Download firmware, associated files (such as AOP, EDS, and DTM), and access product release notes.	rok.auto/pcdc

Documentation Feedback

Your comments help us serve your documentation needs better. If you have any suggestions on how to improve our content, complete the form at rok.auto/docfeedback.

Waste Electrical and Electronic Equipment (WEEE)

At the end of life, this equipment should be collected separately from any unsorted municipal waste.

Rockwell Automation maintains current product environmental information on its website at rok.auto/pec.

Rockwell Otomasyon Ticaret A.Ş. Kar Plaza İş Merkezi E Blok Kat:6 34752, İçerenköy, İstanbul, Tel: +90 (216) 5698400 EEE YÖnetmeliğine Uygundur

Connect with us.

rockwellautomation.com — expanding **human possibility**™

AMERICAS: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
EUROPE/MIDDLE EAST/AFRICA: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
ASIA PACIFIC: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication ICSTT-RM458D-EN-P - February 2021
Supersedes Publication ICSTT-RM458C-EN-P - July 2019