Advantys Configuration
Software 11.0
User Manual

04/2016

33003486.13

www.schneider-electric.com

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

Table of Contents

Safety Information

Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a Danger safety label indicates that an electrical hazard exists, which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

DANGER

DANGER indicates an imminently hazardous situation which, if not avoided, **will result in** death or serious injury.

WARNING

WARNING indicates a potentially hazardous situation which, if not avoided, **can result in** death or serious injury.

CAUTION

CAUTION indicates a potentially hazardous situation which, if not avoided, **can result in** minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

About the Book

At a Glance

Document Scope

This user manual is intended to support the configuration software for the Advantys distributed I/O system.

Validity Note

This documentation is valid for Advantys Configuration Software 11.0.

Related Documents

Title of Documentation	Reference Number
Advantys FTB CANopen IP67 monobloc input/output splitter box User guide	1606218 02 A04
Advantys FTM CANopen IP67 Modular Input/Output Splitter box User guide	1606224 02 A04
Advantys OTB CANopen Remote Inputs and Outputs User Guide	1606384 02
Advantys OTB Ethernet Remote inputs and outputs User guide	1606385 02
Advantys OTB Modbus Remote Inputs and Outputs User Guide	1606383 02
Advantys STB Hardware Components Reference Guide	31002952
Advantys STB Reflex Actions Reference Guide	31004635
Advantys STB System Planning and Installation Guide	31002947
Registration Wizard Online Help	-

You can download these technical publications and other technical information from our website at http://download.schneider-electric.com

Product Related Information

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Only persons with the appropriate expertise in control systems should design, program, install, alter, and apply this product.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Chapter 1
Hardware and Software Requirements

Overview

The Advantys Configuration Software is designed to run on various Windows-based operating systems. This chapter describes your computer system requirements. It also provides instructions for installing and uninstalling the software.

What Is in This Chapter?

This chapter contains the following topics:

System Requirements

Hardware Requirements

This table describes the minimum hardware requirements to run the Advantys Configuration Software on Microsoft Windows 7 Professional (32/64 bits), Windows 8 Pro (64 bits), Windows 8.1 (32/64 bits), Windows 10, Windows Server 2008 R2 (64 bits), and Windows Server 2012 R2 (64 bits):

Requirement	Minimum
System	Pentium Processor 2.4 GHz or higher, recommended 3.0 GHz
RAM	2 GB, recommended 3 GB
Hard Disk	Minimum available free space 2 GB, recommended 10 GB
Microsoft Internet Explorer 5.5 or higher	

This table describes the minimum hardware requirements to run the Advantys Configuration Software on Microsoft Windows XP Professional:

Requirement	Minimum
System	Pentium Processor 1.2 GHz or higher, recommended 3.0 GHz
RAM	1 GB, recommended 2 GB
Hard Disk	Minimum available free space 2 GB, recommended 5 GB
Microsoft Internet Explorer 5.5 or higher	

A CD-ROM drive is the required installation medium.

Virtual Machine

The Advantys Configuration Software runs on the following virtual machines:
- VirtualBox
- VMWare

Connection to a Physical Island

The Advantys Configuration Software runs on a PC that can connect to the network interface module (NIM) of the physical Advantys Island in various ways, depending on the product family and network.

You need a cable to make any connection. Schneider Electric offers a number of cables for the connection. In case, you use an Advantys NIM supporting Ethernet or serial Modbus or an Advantys STB NIM supporting Modbus Plus, you need an appropriate cable for the Ethernet connection. In case you use Advantys OTB NIMs, you need an appropriate cable to download firmware revisions. The type of cable therefore differs depending on the product family.

The following table lists the references for the cables:

Cable	References
Serial	2 m (6.2 ft) STB XCA 4002 programming cable It is delivered with the software.
USB	SR2 CBL 06 USB to serial (D-Sub 9) connector cable Use this cable if your computer has no serial (D-Sub 9) connector. This cable provides an USB to serial (D-Sub 9) connector. Additionally, use the STB XCA 4002 programming cable which is part of the Advantys Configuration Software. To download revisions to Advantys OTB firmware, a USB to RS-485 converter cable is required. Use the converter TSXCUSB485 along with the RJ45 network cable described below.
Ethernet	Use a ● Cat5 twisted-pair cable shielded or unshielded (STB/UTB) with a maximum length of 100 m (328 ft) for STB NIMs. ● RJ45 network cable for OTB NIMs.

For more information about the required hardware connections, refer to the *Advantys STB System Planning and Installation Guide*, the CFG port discussion in your *Advantys STB NIM Applications Guide* or the Advantys OTB, FTB or FTM hardware manuals.

Compatibility and Limitations

Introduction

The following overview gives you information about the compatibility with non-Advantys products and the limitations of older Advantys versions.

Compatibility List

The following list specifies the compatibility with other software products.

The output files created by the Advantys Configuration Software allow to interact with the following programming and network configuration products:

Software	Version	Description
TwidoSuite	2.2 or later	for FTB, FTM, and OTB Islands via exported Island description files in CANopen EDS or DCF formats
SyCon	2.8 or later	via exported Island description files in CANopen EDS and DCF formats
Unity Pro	2.*x*	via exported symbol description files in XSY format
Unity Pro	3.*x* or later	via command line interface and exported files in XML and XSY formats
PL7	4.2 or later	via exported symbol description files in SCY format
Concept	2.5 or later	via exported section description files in TXT format

Limitations of Advantys

A configuration including a V3.*x* Advantys STB CANopen network interface module (NIM) cannot be downloaded into a V2.*x* NIM or a V1.*x* NIM.

Installing and Removing the Advantys Configuration Software

Before You Start

Before you install the Advantys Configuration Software, close all Windows applications and deactivate any virus-protection software.

Installation

To install the Advantys Configuration Software perform the following steps:

Step	Action	Result
1	Insert the Advantys CD in the CD-ROM drive of your PC.	If the Autorun function is activated, the installation will start automatically.
2	If the installation does not start automatically, double-click *CD-Rom drive: \start.exe*.	The **Choose Setup Language** dialog box is displayed.
3	Choose a language and click **OK**.	The **Installation Wizard** dialog box is displayed.
4	Click **Next** to continue.	The **Release Notes** dialog box is displayed.
5	Click **Next** to continue.	The **License Agreement** dialog box is displayed.
6	Accept the license agreement and click **Next**.	The **Customer Information** dialog box is displayed.
7	Enter your user information and the following application settings, and then click **Next**.	The **Product Information** dialog box is displayed.
8	Select the desired Part Number from the list and click **Next**.	The **Destination Folder** dialog box is displayed.
9	Browse the destination folder or use the standard folder, and then click **Next** to continue.	The **Setup Type** dialog box is displayed.
10	Select the type of installation you want to use and click **Next** to continue. **Typical** - installs the Advantys standalone tool and the Advantys DTM **Complete** - installs all the features **Custom** - allows you to select the program features you want to install **NOTE:** You can cancel the installation of Advantys DTM in the **Custom** type.	The **Shortcuts** dialog box is displayed.
11	Select or clear the appropriate check boxes and click **Next**.	The **Ready to Install the Program** dialog box is displayed.

Step	Action	Result
12	Click **Install** to start the installation of the Advantys Configuration Software on your computer.	The installation status is displayed and then the **Installation Wizard Completed** dialog box is displayed.
13	Click **Finish**.	After the installation, an icon is displayed on your desktop: 

Removal

To remove the Advantys Configuration Software from your computer, choose **Start** → **Settings** → **Control Panel** → **Add/Remove Programs**.

Installing and Removing Advantys DTM

Installation

The installation of the Advantys Configuration Software includes the optional installation of the Advantys DTM. The Advantys DTM is supplied on the same CD as the Advantys Configuration Software. During installation of the Advantys Configuration Software you can decide whether you want to install the Advantys DTM or not. By default, the Advantys DTM will be installed automatically with the Advantys Configuration Software.

It is not possible to install Advantys DTMs separately, without Advantys Configuration Software.

Adding the Advantys DTM to Your Schneider Software Tool

After installing your Advantys DTM, it has to be added to the DTM catalog in your Schneider software tool before it can be used. Please refer to your Schneider software tool documentation how to do this.

Removal

Advantys DTMs are removed with the Advantys Configuration Software removing procedure.

It is possible to remove the DTM Library through **Maintenance** -> **Modify**. Select the **DTM library** or parts of the **DTM library** to remove.

NOTE: For details on the installation/removal processes, refer Installing and Removing the Advantys Configuration Software *(see page 17)*.

Chapter 2
Configuration Software Environment

Overview

This chapter provides an overview of the basic components of the Advantys Configuration Software.

What Is in This Chapter?

This chapter contains the following sections:

Section 2.1
Product Families and Fieldbuses

Introduction

This section provides an overview of the different hardware products that can be used in combination with the Advantys Configuration Software. Further, it contains a short description of the different fieldbus or network types that are supported by the Advantys hardware products and configuration software.

What Is in This Section?

This section contains the following topics:

Product Families

Introduction

The Advantys Configuration Software supports these 4 hardware product families:
- Advantys FTB family
- Advantys FTM family
- Advantys OTB family
- Advantys STB family

Each product family includes modules of different groups and types, offering various performances. You can select the product family that best fulfills your demands.

FTB Family Description

The Advantys FTB (field terminal block) family consists of I/O splitter boxes including a network interface for CANopen.

All FTB modules possess an Ingress Protection (IP) rating of 67 according to IEC 60529.

An Advantys FTB Island always consists of 1 FTB module. Depending on the module, the number of pre-configured and configurable digital inputs and outputs varies.

FTM Family Description

The Advantys FTM (field terminal module) family includes network interface modules (NIMs) for CANopen and various compact and extensible I/O splitter boxes.

As with the FTB modules, all FTM modules are IP 67 modules.

An Advantys FTM Island consists of 1 FTM network interface module and at least 1 FTM I/O splitter box. Each NIM is fitted with 4 M12-type connectors for connecting splitter boxes. This allows a star architecture that can consist of 4 segments. Each segment can contain up to 4 I/O splitter boxes, connected in a daisy chain (line architecture). Thus, an FTM Island can include a maximum number of 4 analog I/O splitter boxes, i.e. 1 per segment as they are non-extensible, or 16 digital I/O splitter boxes, i.e. 3 extensible and 1 compact per segment.

OTB Family Description

The Advantys OTB (optimized terminal block) family includes network interface modules with built-in I/Os and expansion I/O modules. OTB NIMs support the following fieldbuses or networks:
- CANopen fieldbus
- Modbus fieldbus
- Ethernet communication network

All OTB modules possess an IP rating of 20 according to IEC 60529.

An Advantys OTB Island consists of 1 OTB NIM. Every NIM has 12 built-in inputs and 8 built-in outputs and accepts up to 7 Twido or TM2 I/O expansion modules.

OTB NIMs provide the following specific functions:
- fast counter (RFC)
- very fast counter (RVFC)
- pulse generator (RPLS)
- pulse generator with pulse width modulation (RPWM)
- programmable input filter

STB Family Description

The Advantys STB family includes open fieldbus NIMs, power distribution modules, standard and special I/O modules, extension modules and special modules. These constitute the core Advantys STB modules. In addition, an STB Island can be extended to non-STB devices. These can be preferred modules and/or enhanced CANopen devices.

An Advantys STB Island contains at least 1 NIM, 1 STB I/O module, a power distribution module and a terminator. The NIM resides in the primary segment which is the mandatory part of an STB Island. In addition, every Island can consist of up to 6 extension segments. At most, it supports 32 I/O modules. All STB modules, except for the NIMs, are mounted in base units interconnected on DIN rails, thus forming the Island bus structure. NIMs are directly attached to DIN rails.

The following NIMs provide different levels of operation:
- basic
- standard
- premium

There is a NIM type to support each of the following fieldbus networks:
- CANopen
- DeviceNet
- Ethernet and Ethernet/IP
- Fipio
- Interbus
- Modbus Plus
- Profibus DP

CANopen Fieldbus Protocol

Communication Model

Based on a serial bus system, CANopen operates within a producer/consumer model. All nodes listen on the network for messages that apply to their functionality. Messages sent by producer devices will be accepted only by particular consumer devices. CANopen also employs client/server and master/slave models.

Each message is given a priority, the one with the higher priority is transmitted first.

Physical Layer

CAN employs a differentially driven (common return), 2-wire bus line. A CAN signal is the difference between the voltage levels of the CAN-high and CAN-low wires. Bus wires can be routed in parallel or twisted or shielded, depending on EMC requirements. A single line structure minimizes reflection.

Node Limitations

A CANopen network is limited to 127 nodes (node IDs 1 to 127).

Register Limits

The maximum data image size for CANopen NIMs is

- 120 words input and
- 120 words output.

For standard and premium Advantys STB CANopen NIMs, the maximum data image size includes the following maximum sizes for HMI-PLC data that can be reserved:

- 120 words HMI-to-PLC data
- 120 words PLC-to-HMI data

Transmission Rate and Network Length

Depending on the transmission rate, the following table lists the maximum network/cable lengths that FTB, FTM, OTB and STB NIMs support:

Transmission Rate	Max. Length for STB	Max. Length for FTB, FTM, OTB
1 Mbit/s	25 m	20 m
800 kbit/s	50 m	40 m
500 kbit/s	100 m	100 m
125 kbit/s	500 m	500 m
50 kbit/s	1000 m	1000 m
10 kbit/s	5000 m	5000 m

For the limitations concerning the topology of an Island, refer to the STB, OTB, FTB and FTM hardware manuals.

Bit-Packing

Bit-packing is performed on the basis of byte boundaries.

DeviceNet Fieldbus Protocol

Communication Model

DeviceNet is based on the serial bus system CAN and operates within a producer/consumer model. All nodes listen on the network for messages that apply to their functionality. Messages sent by producer devices will be accepted only by particular consumer devices. DeviceNet allows users to implement a master/slave, multi-master, or peer-to-peer network architecture.

Each data packet's identifier field defines the data priority. DeviceNet supports strobed, polled, cyclic, change of state and application-triggered data exchange.

Physical Layer

DeviceNet's data link layer is defined by the CAN specification. CAN implements a differentially driven (common return), 2-wire bus line. DeviceNet's physical layer contains 2 twisted pairs of shielded wires, 1 for transferring data and 1 for supplying power. This results in simultaneous support for self-powered devices and those receiving power. Devices can be added/removed without powering off the fieldbus.

Node Limitations

A DeviceNet network is limited to 64 addressable nodes (node IDs 0 to 63).

Register Limits

The maximum data image size for DeviceNet NIMs is

- 128 words input and
- 128 words output.

For standard and premium DeviceNet NIMs, the maximum data image size includes the following maximum sizes for HMI-PLC data that can be reserved:

- 32 words HMI-to-PLC data
- 32 words PLC-to-HMI data

Transmission Rate and Network Length

DeviceNet supports a trunk line/drop line network configuration. This table lists the transmission rates DeviceNet NIMs support for CAN devices and the resulting maximum lengths of DeviceNet networks depending on the cable type:

Cable Type	125 kbit/s	250 kbit/s	500 kbit/s
Thick Trunk	500 m	250 m	100 m
Thin Trunk	100 m	100 m	100 m
Flat Trunk	420 m	200 m	75 m
Maximum Drop Length	6 m	6 m	6 m
Cumulative Drop Length	156 m	78 m	39 m

Bit-Packing

Bit-packing is performed on the basis of byte boundaries.

Ethernet Network

Introduction

Ethernet is a frame-based networking technology for local area networks (LANs). It can use a bus or star topology to connect different nodes on a network. Advantys NIMs residing on the Ethernet LAN use TCP as the transport layer and IP as the network layer. The NIM's fieldbus (Ethernet) port is configured as Modbus over TCP/IP.

Advantys STB Ethernet/IP NIMs are based on DeviceNet.

Communication Model

Modbus TCP/IP is based on a client/server model. Each Modbus master has an array of registers from which the clients can read or to which they can write data. The Modbus server routinely polls each field device and looks for changes in the data. First, data from the Ethernet host are written to the output data image area in the NIM's process image. Then, status, echo output and input data information from the I/O modules on the Island is placed in the input data image area. In this location, the Modbus server can access them over the TCP/IP network or over the STB CFG port.

Physical Layer

The Advantys NIMs supporting Ethernet LAN are based on the 10Base-T standard. This requires a twisted pair cable with a maximum segment length of 100 m, terminating with an RJ-45 connector. Exchanges on an Ethernet network use a multiple access protocol with carrier sense and collision detection. Advantys OTB Ethernet NIMs additionally support the 100Base-T standard.

Node Limitations

The Ethernet network enables communication with a wide range of devices.

Register Limits

The maximum data image size for Ethernet NIMs is

- 4096 words input and
- 4096 words output.

For standard and premium STB Ethernet NIMs, the maximum data image size includes the following maximum sizes for HMI-PLC data that can be reserved:

- 512 words HMI-to-PLC data
- 512 words PLC-to-HMI data

Transmission Rate and Network Length

The Advantys Ethernet NIMs support a transmission rate of 10 Mbit/s, which is the 10Base-T standard. Advantys OTB Ethernet NIMs additionally support a transmission rate of 100 Mbit/s, which is the 100Base-T standard.

Bit-Packing

Ethernet does not support bit-packing.

Fipio Fieldbus Protocol

Introduction

On a Fipio network, every device (node) is associated with a unique identifier that is its global address. Neither the device owning a specific identifier nor the device receiving data from an identifier need to know one another's physical address.

Communication Model

Fipio is based on a master/slave model. As a time-critical protocol, it frequently uses a producer/consumer model. There is only 1 producer of a variable and all of the other devices on the network are potential consumers of the variable. Upon request from the Fipio fieldbus master, the producer of a variable advertises its value. Those consumers requiring the value capture it. No acknoledgement by the consumer devices is required.

Data exchange between the fieldbus master and the Avantys Island bus is cyclical.

Physical Layer

Fipio's physical layer consists of a shielded twisted pair cable.

Node Limitations

A Fipio network is limited to 128 addressable nodes (node IDs 0 to 127, except for 63, which is reserved for the programming and diagnostics terminal).

Register Limits

The maximum data image size for Fipio NIMs is
- 32 words input and
- 32 words output.

For standard and premium Fipio NIMs, the maximum data image size includes the following maximum sizes for HMI-PLC data that can be reserved:
- 32 words HMI-to-PLC data
- 32 words PLC-to-HMI data

Transmission Rate and Network Length

The Advantys Fipio NIMs support a transmission rate of 10 Mbit/s for the following maximum network lengths:
- 1 km for a single fieldbus segment
- 15 km with repeaters between the segments

Bit-Packing

Bit-packing is performed on the basis of word boundaries.

Interbus Fieldbus Protocol

Introduction

Interbus is a serial bus system with an active ring topology, having all devices integrated in a closed transmission path and addressed according to their sequence in the ring.

Communication Model

Interbus is based on a master/slave network model. Each network slave has an in connector for receiving data and an out connector for transmitting data on the ring. Each device amplifies the incoming signal and sends it on. The ring structure uses a distributed shift register. In a single bus cycle, data from the master to the slaves (and from the slaves to the master) are transferred. The cycle ends when the loop-back word is returned to the master. Each node is a component on the shift register ring on which data are circulated.

Advantys STB NIMs only support the remote bus structure. Data exchange between the fieldbus master and the Advantys Island bus is cyclical and 16 words of cyclical data are supported by Advantys Interbus NIMs.

Physical Layer

Interbus' physical layer consists of a single twisted pair of shielded wires.

Node Limitations

An Interbus network is limited to 512 addressable nodes.

Register Limits

The maximum data image size for Interbus NIMs is
- 16 words input and
- 16 words output.

For standard and premium Interbus NIMs, the maximum data image size includes the following maximum sizes for HMI-PLC data that can be reserved:
- 15 words HMI-to-PLC data
- 15 words PLC-to-HMI data

Transmission Rate and Network Length

The Advantys Interbus NIMs support a transmission rate of 500 kbit/s at a maximum network length of 12.8 km and a maximum distance of 400 m between devices.

Bit-Packing

Bit-packing is performed on the basis of byte boundaries.

Modbus Plus Fieldbus Protocol

Communication Model

Modbus Plus is based on a master/slave network model. Its protocol uses a logical token bus. A token is a grouping of bits that is passed in sequence from 1 device to another on a single network in order to grant access for sending messages. Each node on the network can access the network once it receives the token. While holding the token, a node initiates message transactions with other nodes. Global data are transferred within the token frame.

Your application can be laid out as 1 large network or several smaller ones with each one having its own token passing. Data exchange between the fieldbus master and the Advantys Island bus is cyclical.

Physical Layer

The physical layer of Modbus Plus consists of a twisted pair shielded cable that is run in a direct path between successive nodes.

Node Limitations

A Modbus Plus network is limited to 64 addressable nodes (node IDs 1 to 64).

Register Limits

The maximum data image size for Modbus Plus NIMs is
- 125 words input and
- 125 words output.

For standard and premium Modbus Plus NIMs, the maximum data image size includes the following maximum sizes for HMI-PLC data that can be reserved:
- 125 words HMI-to-PLC data
- 125 words PLC-to-HMI data

Transmission Rate and Network Length

A Modbus Plus network consists of 1 or more cable sections, with any section supporting up to 32 nodes. The Advantys Modbus Plus NIMs support a transmission rate of 10 Mbit/s at a maximum network section length of 450 m and a minimum distance of 3 m between devices.

Bit-Packing

Modbus Plus does not support bit-packing.

Profibus DP Fieldbus Protocol

Communication Model

Profibus DP is a serial fieldbus, based on a master/slave network model. Only the fieldbus master has access rights to the bus, the slaves can only respond to prompts and requests. Interactions between a Profibus DP fieldbus master and any node on its network comprise a series of service access points (SAPs) that are defined in Profibus Standard DIN 19245.

Data exchange between the fieldbus master and the Advantys Island bus is cyclical. For predictable results, the bus cycle time has to be shorter than the cycle time of the master's program.

Physical Layer

The physical layer of Profibus DP consists of a shielded twisted pair line.

Node Limitations

A Profibus DP network is limited to 125 addressable nodes (node IDs 1 to 125).

Register Limits

The maximum data image size for Modbus Plus NIMs is 120 words for the sum of inputs and outputs.

For standard and premium Profibus DP NIMs, the maximum data image size includes the following maximum sizes for HMI-PLC data that can be reserved:
- 120 words HMI-to-PLC data
- 120 words PLC-to-HMI data

Transmission Rate and Network Length

Depending on the transmission rate, the following table lists the maximum network lengths that Advantys Profibus DP NIMs support:

Transmission Rate	Maximum Network Length
12 Mbit/s	100 m
6 Mbit/s	100 m
3 Mbit/s	100 m
1.5 Mbit/s	200 m
500 kbit/s	500 m
187.5 kbit/s	1000 m
93.75 kbit/s	1200 m
45.45 kbit/s	1200 m

Bit-Packing

Bit-packing is performed on the basis of byte boundaries.

Section 2.2
Advantys Configuration

Introduction

This section provides an overview on the components used to set up an Advantys configuration.

What Is in This Section?

This section contains the following topics:

Introduction

Overview

The Advantys Configuration Software supports the Advantys distributed I/O system, an open, modular system designed for the machine industry, with a migration path to the process industry.

The software is an optional feature of the system that can be used for the following activities:

- creating, modifying, and saving the configuration descriptions of all the physical devices on an Island
  These tasks are performed mainly in offline mode, although some modifications may be done online.
- monitoring Island performance, adjusting data values, and building a binary file describing the Island configuration

Configuration Check

The Advantys Configuration Software checks for correctness (numeric limitations on modules and I/O points, compatibility between power supply modules and I/O modules, and so on.) whenever possible during the editing process. Otherwise, these checks are made when a complete project build is performed.

The software provides features that help you plan

- logic and field power consumption
- I/O image area consumption

Editors

The table below describes the software features of the 4 main editors:

Editor	Software Features
Island Editor	provides a graphical display of the Island segments and the order in which the modules are installed
Module Editor	allows you to customize the operating parameters of the individual input and output modules
User Defined Label Editor	allows you to modify the user defined labels of all the I/O image data items in the current Island
Reflex Editor	allows you to program reflex actions and map their results to individual output modules on the Island

Software Outputs

A list of the modules available for your Island configuration is displayed in the Catalog Browser of the Advantys Configuration Software. Use this browser to select and install modules in the Island Editor.

The configuration software provides 3 major outputs:
- a device description file
  (You may generate this device description file in several formats depending on the intended use and the fieldbus format.)
- a binary image of the configuration suitable for loading into the STB Island
- printout of project documentation

Further, the Advantys Configuration Software automatically assigns labels to certain data items.

Starting the Advantys Configuration Software

Starting the Software from the Desktop

The normal installation process places an Advantys program icon onto your desktop. To launch the Advantys Configuration Software, double-click this icon:

Starting the Software in the Windows Environment

Launch the Advantys Configuration Software in the Windows environment as follows:

Step	Action
1	Click the **Start** button.
2	From the **Programs** option, select **Schneider Electric**, followed by **Advantys**, and, once again, by **Advantys**.
3	From the dialog box, select the type of physical Island you want to configure.
4	Select the language you want to use and click **OK**.

Starting the Software from the Command Line

Start the Advantys Configuration Software from a command line editor as follows:

Step	Action
1	Open the command line editor.
2	When the command prompt appears, type `advantys.exe -w=` followed by a path to the Workspace. For example: `"Program Files\Schneider Electric\Advantys\Advantys.exe" -w="C:\Program Files\Schneider Electric\Advantys\Projects\test\test.aiw"` **Note:** To open the Advantys Configuration Software with the previous Workspace as if in Windows, omit the -w option.
3	Press ENTER.
4	From the dialog box, select the type of physical Island you want to configure.
5	Select the language you want to use and click **OK**.

If you enter a wrong argument as a command parameter, a notification will be displayed, suggesting valid arguments.

What Is an Island?

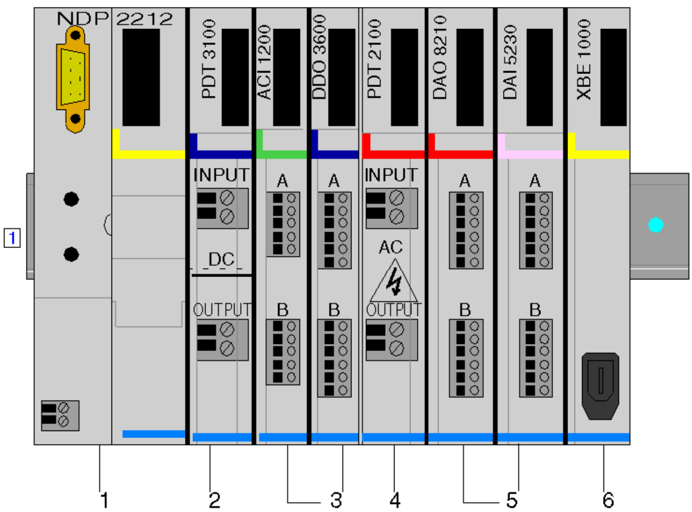
Introduction

In the Advantys Configuration Software, a distinction is drawn between a *physical Island* in the real world of your application and a *logical Island* in the context of the software.

Physical Island

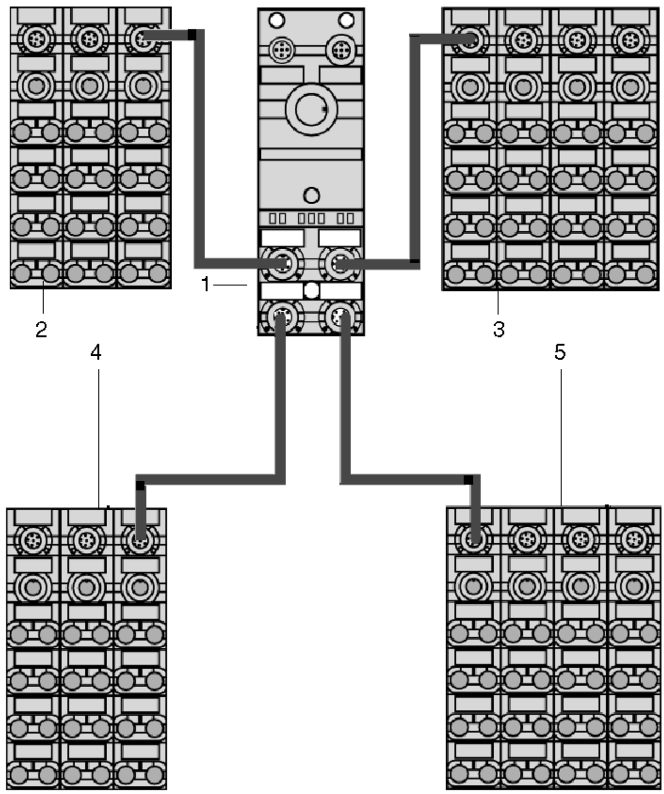
An Island is an assembly of distributed I/O, power distribution and Island bus communication/extension modules that function together as 1 node on a fieldbus. Depending on type and product family, an Island can contain up to 32 I/O modules plus a NIM. STB Islands additionally require 1 or more power distribution modules (PDMs) and offer modules that let you extend the bus to multiple segments (or rails) of Advantys STB I/O modules, to Advantys STB preferred modules, and to enhanced CANopen devices.

This illustration shows an example of a segment on a physical STB Island:



1 NIM
2 PDM
3 Voltage Group of I/O Modules
4 Second PDM Supporting the Second Group of I/O Modules (5)
5 Second Group of I/O Modules, Requiring a Different Field Power Voltage or Additional Current
6 EOS Module for Extending the Physical Island to Another Segment of Advantys I/O Modules or to a Preferred Module

While OTB and FTB Islands only consist of 1 segment, FTM Islands can contain up to 4 segments, arranged in a star architecture as shown in the figure below. Each segment can contain up to 4 I/O modules:



1 NIM
2 Segment 1
3 Segment 2
4 Segment 3
5 Segment 4

Logical Island

The Advantys Configuration Software lets you model a physical Island so that it can be tested against our design rules and customized to meet your application requirements. The software model is called *logical Island* *(see page 154)*.

The logical Island is a file in the software program with a *.isl* extension. It contains a description of the physical Island: all the modules on the Island and all the operating parameters associated with each module that may be defined in the software.

As you develop a logical Island, the software will provide notifications about any mistakes you have made in the model, and it will not assist you in creating an invalid configuration. For example, it stops and notifies when you place a DC module in a location where AC field power is received (and vice versa).

Island Types

Depending on the product family, Islands can be of the following 4 types:
- FTB
- FTM
- OTB
- STB

All of these Island types can be configured using the Advantys Configuration Software.

Workspace

Introduction

The Workspace is a project environment in the Advantys Configuration Software. The Workspace is where you design a logical Island configuration. Within the Workspace, you can create a new configuration, output supporting files and, in case of Advantys STB Islands, download it into the physical Island. You can also upload configuration data from a physical Island to a logical Island in the Workspace *(see page 159)*.

A Workspace is saved as a file with an *.aiw* extension.

Relationship of the Workspace to an Island

Within a Workspace, you can create and manage 1 or more logical Islands, up to a maximum of 10. These Islands can be of different types. The configuration data associated with each Island are stored in its own *.isl* file within the Workspace.

Customizing Your Workspace

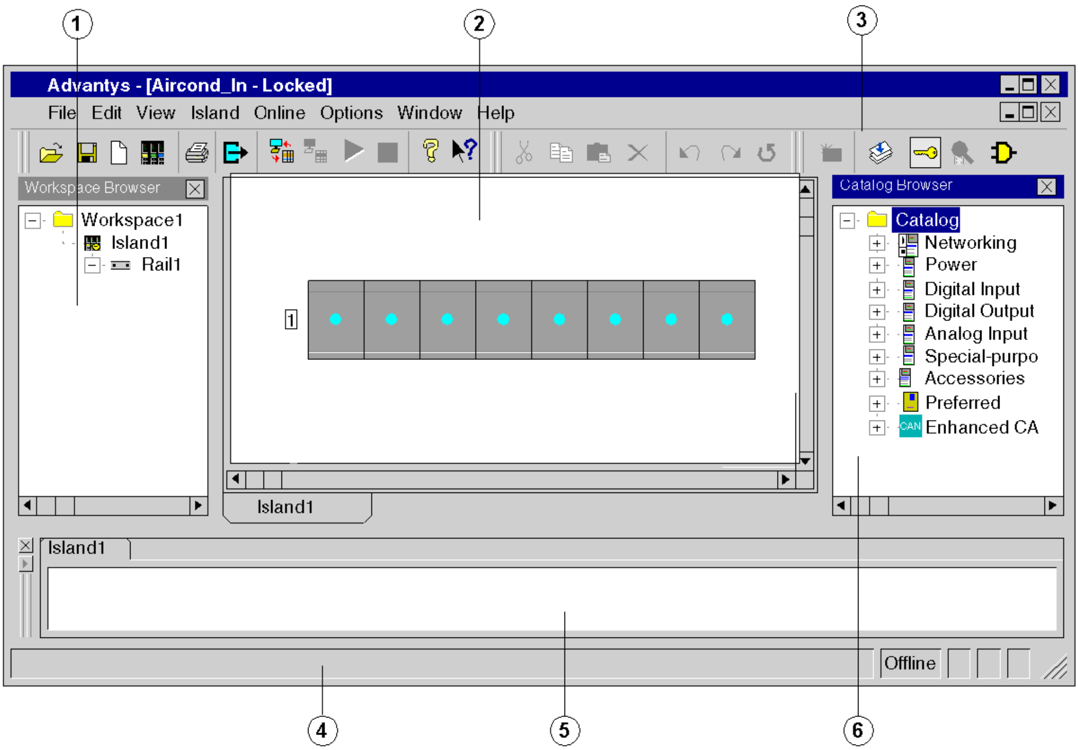
You can customize your Workspace settings by selecting **Settings** from the **Options** menu.

These settings include the following features:
- interface language (**English**, **French**, **German**, **Spanish** or **Italian**)
- foreground and background colors
- default directory path

If you have more than 1 Workspace on your computer, you can define different settings for different Workspaces.

Workspace Window

A Workspace window consists of the following areas:



1 Workspace Browser
2 Island Editor
3 Toolbars
4 Status Bar
5 Log Window
6 Catalog Browser

All these areas may be hidden; except the bars, they may also be enlarged and reduced. Further, all areas but the status bar may be moved or docked on the Workspace window. If you modify the layout of a Workspace window, the layout definition will be saved. Each time you reopen that Workspace, the window is displayed with the layout you used the last time you saved the Workspace.

The figure above shows the default locations of each of the 6 Workspace areas. The functions of these areas will be described in more detail on the following pages.

Island Segments

Primary Segment

Each Island has to include at least 1 segment, called the primary segment. The primary segment is always the first segment in the Island configuration. It is where the network interface module (NIM) resides.

Modules in the Primary Segment

The NIM is always located in the first (leftmost) slot of the primary segment. The power supply built into the NIM converts 24 VDC into a 5 V logic power signal that supports all other modules in the primary segment.

In STB Islands, the NIM is immediately followed by a power distribution module (PDM), which will distribute field power to the input and output modules on your Island. Depending on the type of I/O modules in the segment, you will use an STB PDT 310x PDM (to distribute 24 VDC), an STB PDT 210x PDM (to distribute 115 or 230 VAC) or some combination of the 2 PDM types.

The power supply in the NIM supports 1.2 A of current to be drawn by the I/O modules in the segments.

Auxiliary Power Supply

For STB Islands, an auxiliary power supply is available. The STB CPS 2111 auxiliary power supply provides 5 VDC logic power to the modules installed to its right in an Advantys STB Island segment. It works together with the NIM (in the primary segment) or with a BOS module (in an extension segment) to provide logic power when the I/O modules in the segment draw current in excess of 1.2 A. The auxiliary power supply module can only be preceded by I/O modules.

The module converts 24 VDC from an external power source to an isolated 5 VDC of logic power, providing up to 1.2 A of current to the modules on its right.

Last Device in a Primary Segment

Each Advantys STB system has to be terminated at the last module. If the Island comprises only the primary segment, use the STB XMP 1100 termination plate as the last module in this segment. If the Island is extended, the type of extension defines how the primary segment is terminated.

Depending on your requirements, you can choose 1 of the following methods to terminate a primary segment in an STB Island:

If the Island bus ...	Then...
comprises just 1 segment with no extensions	terminate the primary segment with an STB XMP 1100 termination plate.
is extended to another segment of Advantys STB I/O modules	install an STB XBA 2400 base to hold an STB XBE 1000 or an STB XBE 1100 EOS (end-of-segment) module at the end of the segment. Do not use a termination plate in the primary segment.

If the Island bus ...	Then...
is extended to a preferred module	install an STB XBA 2400 base to hold an STB XBE 1100 EOS module at the end of the segment. Do not use a termination plate in the primary segment.
is extended to an enhanced CANopen device	install an STB XBA 2000 base to hold an STB XBE 2100 CANopen extension module at the end of the segment, followed by an STB XMP 1100 termination plate.

Extension Segments

You might want to extend your Island configuration beyond the primary segment for the following reasons:

- You might want to position the I/O modules as close as possible to the sensors and actuators they control.
- You might want to extend an STB Island bus to devices other than Advantys STB I/O modules (preferred modules and/or enhanced CANopen devices).

Advantys FTM NIMs, Advantys STB I/O modules and preferred modules can be followed by extension segments, enhanced CANopen devices are always the last devices on an Island bus.

Terminating the Island Bus

The last module on an STB Island determines how the bus has to be terminated:

If the last module is ...	Then the STB Island bus is terminated using ...
an Advantys STB I/O module	an STB XMP 1100 termination plate.
a preferred module	a TeSys U LU9 RFL15 termination device.
an enhanced CANopen device	an STB XMP 1100 termination plate that follows the STB XBE 2100 CANopen extension module at the end of the segment and a physical termination following the last CANopen device.

Extending Islands to Further Segments

Introduction

You can extend

- Advantys FTM Islands to Advantys FTM I/O splitter boxes
- Advantys STB Islands to
  - Advantys STB I/O modules
  - preferred modules
  - enhanced CANopen devices

OTB Islands consist of only 1 segment, FTB Islands of only 1 splitter box.

Extending FTM Islands

Each FTM NIM is fitted with 4 M12-type connectors for connecting splitter boxes. This allows a star architecture that can consist of 4 segments. Each segment can contain up to 4 I/O splitter boxes, connected in a daisy chain (line architecture). Thus, an FTM Island can include a maximum number of 4 analog I/O splitter boxes, i.e. 1 per segment as they are non-extensible, or 16 digital I/O splitter boxes, i.e. 3 extensible and 1 compact per segment.

The length of a segment can amount to a maximum of 5 m.

Extending STB Islands to STB Modules

You can extend an STB Island bus to 1 or more segments of STB I/O modules. These segments are called extension segments. Extension segments have to be preceded by 1 primary segment.

The first (leftmost) module in each extension segment depends on the module type immediately preceding it:

If preceded by ...	Then the first module in the extension segment has to be ...
a segment of Advantys STB I/O modules	• either an STB XBE 1200 BOS (beginning-of-segment) • or an STB XBE 1300 BOS module.
a preferred module	an STB XBE 1300 BOS module.

The BOS module is followed by a PDM and 1 or more STB I/O modules. It has a built-in power supply like the one used in the NIM. It provides 1.2 A of current to support the STB I/O modules in its extension segment. Auxiliary power supply can provide additional logic current if necessary.

The BOS is connected to the previous segment or to a preferred module by an Island bus extension cable. The cable and the BOS module extend the Island's communication bus and auto-addressing line to the new segment.

An Island bus may support up to 6 extension segments with a maximum number of 32 STB I/O modules. In the Advantys Configuration Software, each segment is shown on a separate DIN rail. In a real physical installation, more than 1 segment may be placed on the same DIN rail.

Extending STB Islands to Preferred Modules

You may also extend an STB Island bus to 1 or more preferred modules. In most respects, the Island bus handles them just as other STB I/O modules. There are, however, 2 key differences:

- A preferred module is not designed in the Advantys STB form factor and does not fit into 1 of the standard base units. It therefore does not reside in a segment.
- It may require its own power supply.

A preferred module has an input connection to receive an Island bus extension cable from the upstream Island module. It is designed with an extension cable output connection that allows it to send Island bus signals to a downstream module or segment.

An Island can support a maximum of 31 preferred modules. Each segment has to include at least 1 auto-addressable Advantys STB module. You may use Island bus extension cables to daisy-chain multiple preferred modules together.

Extending STB Islands to Enhanced CANopen Devices

You may also extend an STB Island bus to 1 or more CANopen devices. CANopen devices have to be addressed manually, via a set of address switches built onto the devices. Via the **Options** tab, the Module Editor for the NIM also allows you to set up the maximum node ID value to be used on a CANopen extension. Any manually set address switch has to match the automatically assigned node ID. The address assignment for the CANopen modules starts with this value, counting downwards to avoid any overlap with addresses automatically assigned to the Advantys STB modules. The default value is 32; however, it may be modified in order to enforce the use of lower node IDs for CANopen devices. Indeed, some of these devices may have a restricted configurable address range.

Whenever a CANopen device is part of the Island bus, the bus has to be configured to operate at 500 kBaud. The default baud rate is 800 kBaud, so change it selecting from the **Island** menu *Baud Rate Tuning, page 283*.

When you are using CANopen devices, do not push the RST button on the NIM. The RST button will cause the baud rate to be set to 800 kBaud, and the Island bus will not operate properly.

The Island bus generally supports up to 12 CANopen devices. However, there are limitations. In case you use CANopen NIMs, only 7 of the 12 supported modules are allowed to be drives. CANopen devices have to always be the last devices on the Island bus. Each segment has to include at least 1 auto-addressable Advantys STB module (the NIM in the primary segment and at least 1 I/O in any other segment).

Maximum Length of the Island Bus

The total length of an STB Island bus, from the NIM to the last device, has to be less than 15 m (49.2 ft). This length includes both the sum of the lengths of all bus extension cables and CANopen cables connecting devices as well as the widths of the hardware modules themselves.

Creating a Project with the Advantys Configuration Software

Introduction

The Advantys Configuration Software provides a set of Windows-based tools that enable you to plan, model, customize, and test Island bus designs and, depending on the product family, to download custom configurations into physical Islands.

Advantages of Using the Software

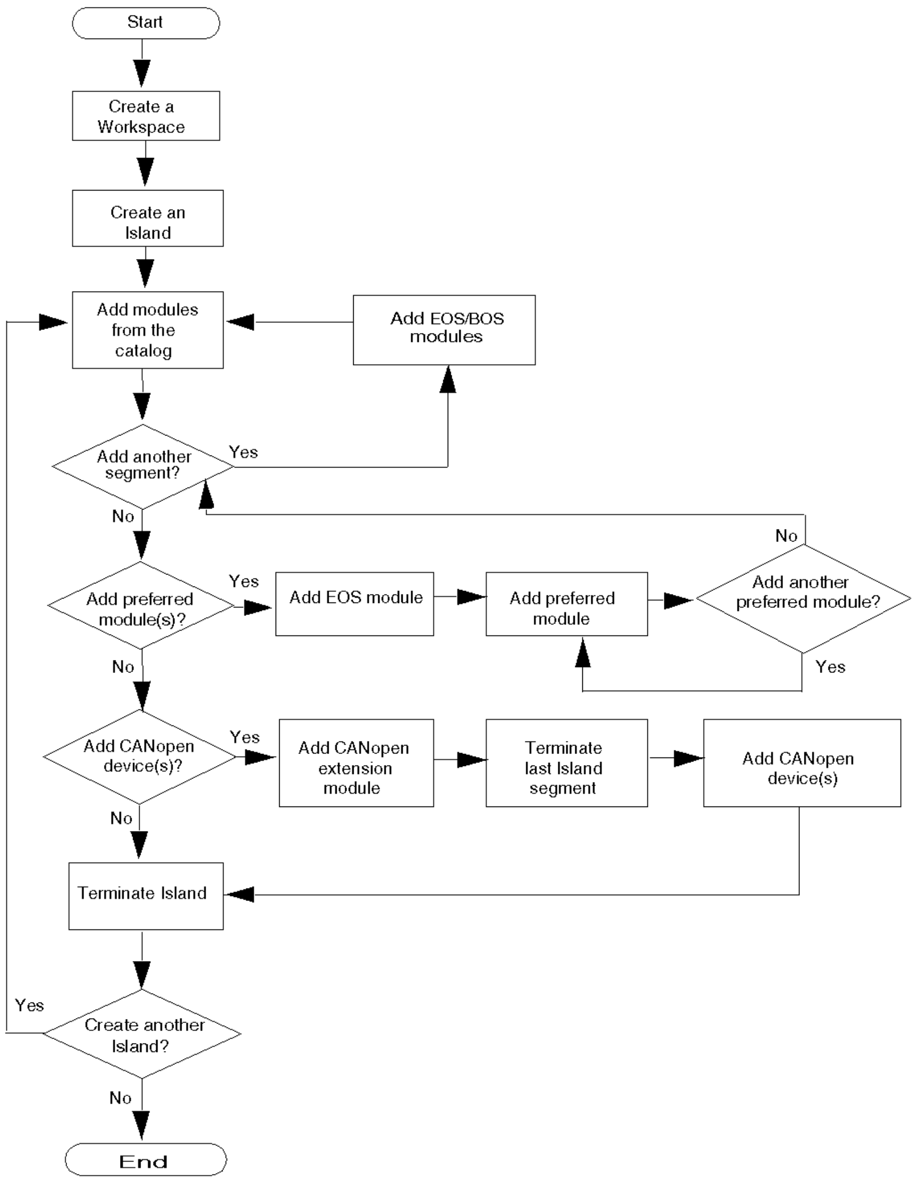
All the Advantys I/O modules have factory-default parameter settings that allow them to be operational directly out of the box. If you want to customize your Island's operational capabilities, however, you need to use the Advantys Configuration Software.

Depending on the Island type, the software allows you the following:

- customizing the operating parameters of the I/O modules
- creating and implementing reflex actions
- optimizing the Island performance by assigning priority to certain modules
- designating certain application-critical modules as mandatory
- adding preferred modules and/or enhanced CANopen devices to the Island configuration
- validating that your STB Island configuration adheres to Advantys STB design guidelines *(see page 154)*

Project Work Flow

The following flowchart describes the work flow associated with a valid STB Island configuration:



Section 2.3
Configuration Environment

Introduction

This section provides information about the editors and browsers of the Advantys Configuration Software.

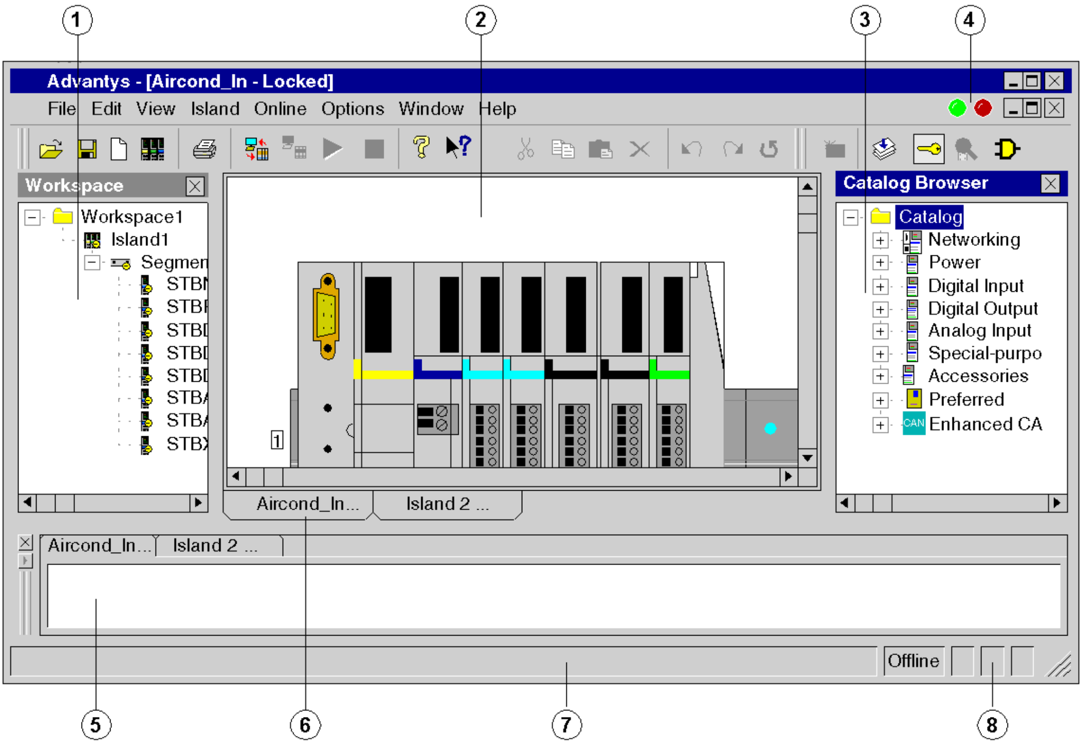
What Is in This Section?

This section contains the following topics:

General

Software Windows

When you create a new Workspace or open an existing Workspace in the Advantys Configuration Software, you will find the general components by default. See the figure below for an Advantys Configuration Software window which includes an Island configuration:



1 Workspace Browser
2 Island Editor
3 Catalog Browser
4 Status LEDs
5 Log Window
6 Island Editor Pane
7 Status Bar
8 Status Indicators

Opening the General Components

You can open or hide the general components of the Advantys Configuration Software using the following elements:

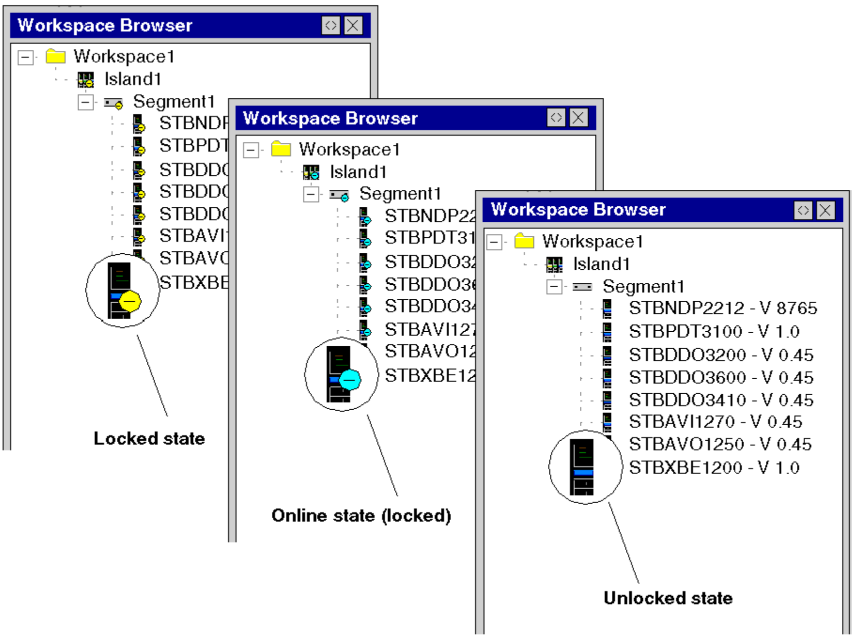
Elements	Description
Toolbar Icons	You can open or hide the Workspace Browser, Catalog Browser, or Log Window using the following icons on the toolbar of the Advantys Configuration Software: 
Software Menu	You can find the menu to open or hide the Workspace Browser, Catalog Browser or Log Window using the following menu entries: ● **View → Workspace Browser** ● **View → Catalog Browser** ● **View → Log Window Browser**
Shortcuts	You can open or hide the Workspace Browser, Catalog Browser or Log Window using the following shortcuts: ● CTRL+W for the Workspace Browser ● CTRL+T for the Catalog Browser ● CTRL+L for the Log Window

Workspace Browser

Introduction

The Workspace Browser displays the contents of the currently open Workspace in a hierarchical or tree-structured fashion. The browser displays all the Islands currently residing in the selected Workspace. Each Island can be expanded to its segment level, and each segment can be expanded to the module level. Unlike the Island Editor, the Workspace Browser allows you to look simultaneously at the contents of multiple Islands.

The browser also indicates graphically whether the Island is locked or unlocked and whether it is online or offline:



NOTE: Note the yellow lock symbol in the left window above; it indicates that the Island is locked. In online mode, the symbol color is blue.

By default, this browser is displayed on the left side of the screen. You may resize it vertically and horizontally, hide or open it. You can also place the Workspace Browser window wherever you are within the Advantys Configuration Software.

Operating in the Browser

The following tables describe the different buttons and commands you may use to operate in a Workspace Browser:

If you right-click or press a context-sensitive key on...	Then a shortcut menu is displayed, featuring the following options:
a Workspace label	• **Add Island** You can add a new or existing Island to the Workspace. • **Properties ...** You can edit the Workspace properties. For example, the logical name, the author´s name, a comment and the version information.
an Island label	• **Add Rail** If the primary DIN rail was deleted for any reason, add a new rail before configuring the Island. **Note:** The menu is only activated if there is no rail in the Island Editor. • **Remove** Removes your Island from the Workspace. You can also delete the files from your hard disk. • **Build** Validates the software configuration you have created. • **I/O Image Overview** Invokes the **I/O Image Overview** dialog box. • **Connect** You can connect your logical Island to your physical hardware. • **Disconnect** You can disconnect your logical Island from your physical hardware. • **Properties ...** You can edit the Workspace properties. For example the logical name, the author´s name, a comment and the version information.
a segment label	• **Cut** • **Copy** • **Paste** • **Delete** You can edit the segments as you know from other MS Windows applications.
a module label	• **Copy** • **Paste** • **Delete** • **Module Editor ...** You can edit the modules' information in the Module Editor *(see page 66)*. You can edit the modules as you know from other MS Windows applications.

If you press ENTER or double-click ...	Then ...
a Workspace label	A collapsed tree expands to the Island level; an expanded tree collapses to the Workspace level.
an Island label	A collapsed tree expands to the segment level; an expanded tree collapses to the Island level. If the Island Editor is closed, it opens.
a segment label	A collapsed tree expands to the module level; an expanded tree collapses to the segment level. The segment and the modules it carries are selected in the Island Editor. If the Island Editor is closed, it opens.
a module label	If the Island Editor is closed, it opens. The module is selected in the Island Editor.

Editing Labels

You can edit the label names of the different elements by clicking the label twice or selecting the label and pressing F2.

Displaying or Hiding the Workspace Browser

By default, the Workspace Browser is displayed on the left side of the window when you create a new Workspace. You have the option of hiding this browser if you do not need to use it.

To hide the Workspace Browser, simply click the following icon on the **View** toolbar:

If you close the Workspace when the Workspace Browser is hidden, it will still be hidden the next time you open the Workspace. If you open an existing Workspace and do not see the Workspace Browser, click once more the following icon on the **View** toolbar to display the browser *(see page 53)*:

Catalog Browser

Introduction

The Catalog Browser lists all available modules for assembling an Island in a tree structure. According to the product families, it contains the following 4 catalogs:

- FTB
- FTM
- OTB
- STB

By default, the Catalog Browser is displayed on the right side of the window. However, you may resize it vertically and horizontally, hide or open it, or move it to another window location.

Module Groups

According to the modules' function, each catalog contains different module groups. The following table lists the different module groups of the STB catalog:

Group	Description
Networking	a group of standard network interface modules (NIMs)
Networking (Basic)	a group of basic network interface modules
Networking (Legacy)	a group of standard network interface modules with older versions
Power	a group of power distribution modules (PDM)
Digital Input	a group of the Advantys digital input modules
Digital Output	a group of the Advantys digital output modules
Analog Input	a group of the Advantys analog input modules
Analog Output	a group of the Advantys analog output modules
Special-Purpose	a group of modules for special purposes, e.g. high speed counters
Accessories	a group of termination, BOS and EOS modules
Preferred	a group of preferred Advantys STB modules
Enhanced CANopen	a group of different modules with CANopen interfaces, e.g. Altivars
Obsolete	a group of modules that are obsolete or soon will be

NOTE: The OTB product family contains TWD as well as TM2 modules. TWD modules are displayed in the obsolete folders of each analog and digital I/O group.

Unregistered Catalogs

An unregistered catalog is indicated by a red mark next to its icon and the tooltip is extended with a remark about the registration status. The sub-tree cannot be expanded and therefore an Island of the associated family cannot be edited. It remains in locked state but viewing Island data and online services are possible.

Operating in the Catalog Browser

Right-clicking the different labels causes the following results:

Right-click the	Result
catalog label.	The **Catalog Properties** dialog box can be invoked.
group label.	Nothing is displayed.
the module label.	The **Module Properties** dialog box can be invoked.

Double-clicking the different labels causes the following results:

Double-click the	Result
catalog label.	A collapsed tree expands to the functional family level. An expanded tree collapses to the catalog level.
functional family label.	A collapsed tree expands to the module level. An expanded tree collapses to the functional family level.
a module label.	In the Island Editor, the module is inserted to the right of a selected module. **Note:** The software will verify whether in terms of power and function the module fits logically.

Alternatively, you can place a module into the Island Editor by selecting the desired module in the Catalog Browser, and then drag the module to position it into the segment of the Island Editor.

NOTE: The Advantys Configuration Software applies standard placement rules to the drag-and-drop feature. It does not allow you to drop a module into an invalid location on the segment.

Modules selected from the Catalog Browser are displayed as graphical modules in the Island Editor and as module labels in the Workspace Browser.

Displaying or Hiding the Catalog Browser

By default, the Catalog Browser is displayed on the right side of the window when you create a new Workspace. You have the option of hiding this browser if you do not need to use it.

To hide the Catalog Browser or to display it if it was hidden before, click the following icon on the **View** toolbar *(see page 53)*:

Island Editor

Introduction

The Island Editor provides a graphical representation of the logical Islands that you are building using the Advantys Configuration Software. Each opened logical Island has its own Island Editor. You can change the active Island Editor with the Island Editor panes below the Island Editor window. By default, the Island Editor is displayed in the top-center pane of the Workspace when you create a new Island. When you open an existing Island, the Island Editor is displayed, maximized or minimized according to the way the Island file was saved in the last work session.

Creating a New Island

When you create a new Workspace or a new Island, the Island Editor appears as an empty DIN rail where the primary segment will be built. You can drag modules from the Catalog Browser into this segment. This is the method by which you establish an Island bus configuration in the software. The Island Editor automatically imposes a set of connectivity constraints on the Island you are designing, by not allowing you from installing modules in invalid locations.

You can enlarge or reduce the view of the Island segments using the **Zoom** option.

Module Addresses

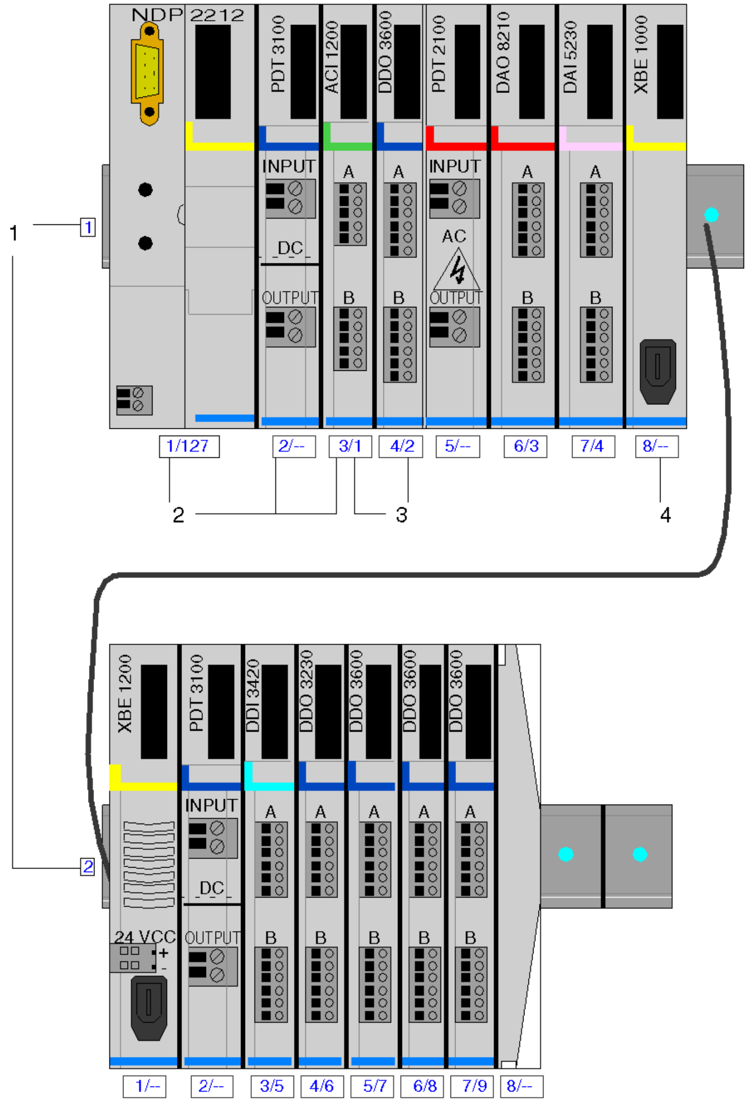
The modules that you add to an Island are automatically addressed. Each address is displayed in the Island Editor. All modules have a segment number, which is displayed in front of the segment. Further, there are slot and node numbers, which are displayed below the modules. Slot numbers represent the physical location of modules. Node numbers represent the logical addresses of the modules on the Island bus.

STB and OTB Islands may contain modules that do not have a logical address, such as STB power distribution modules and the OTB commons block. Therefore, slot numbers and node numbers may not be identical. As a result, the addresses of STB and OTB modules consist of a slot number and a node number in addition to the segment number (see example below).

FTM modules have identical slot and node numbers because all of them are logically addressable. As a result, the addresses of FTM Islands only consist of a segment number and a node number.

Addressing Example

The figure below represents the addressing of STB modules in the Island Editor:



1 Segment Number
2 Slot Number (Physical Location)
3 Node Number (Logical Address of the module on the Island Bus)
4 Node Number Not Available (Non-Addressable Module)

Operating in the Island Editor

The following tables describe the different keystrokes available in the Island Editor:

If you right-click or press context-sensitive key on ...	**Then a shortcut menu is displayed, with the following options:**
a module	• **Cut** • **Copy** • **Paste** • **Delete** • **Module Editor ...** You can edit the different information of the modules within the Module Editor *(see page 66)*. You can edit the modules as you know from other MS Windows applications.
a segment (DIN rail)	• **Cut** • **Copy** • **Paste** • **Delete** You can edit the modules as you know from other MS Windows applications.
the Island Editor	• **Add Annotation** You can add any comment to the Workspace. • **Paste** You can paste copied or cut annotations to your Workspace.

If you press ENTER or double-click ...	**Then ...**
a module	the Module Editor window will be displayed. You can edit the different information of the modules within the Module Editor *(see page 66)*.

Log Window

Introduction

A Log Window displays the results of any operation performed by the Advantys Configuration Software. In online mode, it displays additional health information of the physical Island including upstream fieldbus diagnostic messages. The Log Window provides a separate tab for each Island in the current Workspace. By default, the Log Window is displayed at the bottom of the screen when you open a new Workspace. When you open an existing Workspace, the Log Window may be displayed or hidden, depending upon the Island's last saved state.

Operating in a Log Window

The following table describes the different operations you may perform in the Log Window:

If you right-click or press a context-sensitive key on ...	Then a shortcut menu is displayed, featuring the following options:
the Log Window	• **Save Log File** You can save the Log Window information as a log file. These files are formatted in standard text and can be edited using every text editor. • **Clear** You can clear the Log Window. You are asked to save the log information before. Click **Yes** or **No** to proceed.

Displaying or Hiding the Log Window

You have the option to hide the Log Window if you do not need to use it.

To hide the Log Window, simply click the following icon on the **View** toolbar:

If you close the Workspace when the Log Window is hidden, it will still be hidden the next time you open the Workspace.

If you open an existing Workspace and do not see the Log Window, click once more the following icon on the **View** toolbar to display the Log Window *(see page 53)*:

Status Indicators

Introduction

The Advantys Configuration Software provides 2 status indicators:
- status bar
- status LEDs

Status Bar

A status bar is displayed in the bottom panel of the Workspace display.

The status bar indicates the following information:
- status messages
- offline/online status
- physical Island status
- test or non-test mode of the software

If you open an existing Island, the status bar may not be visible. To reveal it, go to the **View** menu and click **Status Bar**.

Status LEDs

The 2 status LEDs are located on the right side of the menu bar and are active when the software is in online mode.

The LEDs represent the RUN and ERROR LED on the NIM module.

The following blink codes are used:
- blink R - blinks regularly
- blink N - blinks N times (N = 1 to 8)

For a description of the different Island states, refer to *Different STB Island States, page 158*.

Chapter 3
Software Functions

Overview

The Advantys Configuration Software runs either in offline or online mode. In offline mode, you can design the Island configuration and set the operating parameters of your I/O modules. For the Advantys STB product family, there is an online mode where you can download the configuration into a physical Island and monitor an operational Island.

The software's online/offline status is specific to the logical Island currently active in the Island Editor.

At any given time, 1 Island connection may be online.

What Is in This Chapter?

This chapter contains the following sections:

Section 3.1
Module Editor

Introduction

This section provides a general overview of the Module Editor. Because its functions vary depending on the product family, a detailed description of the Module Editor is provided for STB, OTB, FTM and FTB modules separately in the subsequent sections.

What Is in This Section?

This section contains the following topics:

Module Editor Introduction

Introduction

The Module Editor provides information about a selected module, allows you to modify some of its operating parameters and, for the STB product family, to view live I/O data when the software is in online mode.

The Module Editors are specific to both the product families and the module groups. For FTB, FTM and OTB modules, configuration parameters are assigned to each single data item, which is displayed as superordinated. In contrast, parameters of STB modules are listed as superordinated and the data items are assigned to them. The STB Module Editor can also be used for mapping I/Os, whereas FTB, FTM and OTB I/Os are mapped using the **I/O Image Overview** function.

Accessing the Module Editor

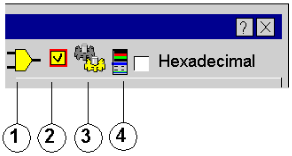
To invoke the Module Editor, choose 1 of the following methods:

If you want to open it ...	Then ...
from the Island Editor	double-click the desired module.
from the Workspace Browser	right-click the module label and select **Module Editor** from the shortcut menu.
from the toolbar	select the desired module in either the Island Editor or the Workspace Browser and click the following icon on the **Island** toolbar: 

NOTE: It is not possible to access the Module Editor from the Catalog Browser.

Status Icons

The following figure shows the status icons displayed to the right of the tabs:



Number	Description
1	The **Reflex Action** icon appears if the module hosts at least 1 reflex action.
2	The **Checkmark** icon appears if at least 1 I/O mapping has been modified.
3	The **Parameter Change** icon appears if at least 1 parameter has been modified. **Note:** Additionally, a dog ear is displayed in the upper right corner of the corresponding value field.
4	The **Locked/Online** status is shown by this icon. The icon states and their respective meanings are similar to those in the Workspace Browser.  **Locked state**  **Unlocked state**  **Online state (locked)**

Hexadecimal Check Box

Check this box to select the hexadecimal display format for all values on the **Parameters** tab. If the box is unchecked, the values in the table are displayed in the decimal format. The default display format is decimal, and the box is unchecked.

Module Help

The Module Editor provides a specific help for the selected module. You can get different information about the module from this **Help** function, for example for wiring diagrams and so on.

Module Editor for STB Modules

Overview of the Tabs

The table below describes the different tabs of the STB Module Editor and for which modules they are available:

Tab	Description
General *(see page 72)*	This read-only tab displays an illustration of the selected module and provides a brief hardware and functional description. It is available for all STB modules.
Parameters *(see page 74)*	This tab displays the operating parameters of the selected module which are currently not mapped. It is accessible for all STB NIMs and standard I/O modules.
Ethernet Parameters *(see page 79)*	This tab displays Ethernet specific parameters of the selected module. It is only accessible for the STB NIP2311 Ethernet NIM. The tab contains the following subordinate tabs: ● **IP Address** for Ethernet IP and port parameters ● **Master IP** for master IP addresses and timeout parameters ● **SNMP** for SNMP parameters ● **Redundancy** for parameters specific to RSTP settings In online mode, the parameters cannot be changed.
Ports *(see page 84)*	This tab displays the actual operating values of certain port parameters. The tab is accessible in online mode only for the: ● STB NIP2311 Ethernet NIM ● STB NCO2212 CANopen NIM version 3.05 or later
I/O Image *(see page 86)*	This tab displays the selected module's I/O data that are currently mapped. It is accessible for all STB NIMs and I/O modules except for the STB NIP2311 Ethernet NIM. In online mode, the live I/O data of the selected module are dynamically displayed.
Diagnostics *(see page 89)*	This read-only tab displays any diagnostic message generated by the selected module. It is only accessible in online mode for STB NIMs and I/O modules.
Options *(see page 91)*	This tab displays optional parameters for the current module. It is accessible for STB NIMs and I/O modules. In online mode, the parameters cannot be changed.
I/O Mapping *(see page 96)*	This tab displays the I/O mapping for the current module. It is available for STB standard I/O modules.
Information	This read-only tab displays device parameters in online mode. It is only available for ATV and Tesys U modules in online mode.

Overview of the Functions

It is possible to
- view general module information,
- edit parameters in offline mode,
- assign labels to parameters and I/O data in offline mode,
- modify the I/O mapping in offline mode,
- monitor I/O data, port parameters and module diagnostics in online mode and
- set I/O data when the Island is online and in test mode.

NOTE:
Keep the following in mind:
- The **Diagnostics** and the **Ports** tab can be accessed only when the software is online.
- Some tabs are not available for all modules.
- The settings or type of information provided on the tabs can differ for each selected module.

Module Editor for OTB, FTM and FTB Modules

Overview of the Tabs

The table below describes the different tabs of the OTB Module Editor and for which modules they are available:

Tab	Description
General *(see page 72)*	This read-only tab displays an illustration of the selected module and provides a brief hardware and functional description. It is available for all OTB, FTM and FTB modules.
Parameters *(see page 102)*	This tab displays all input and output data objects of the selected module including bit level information. It is available for all OTB and FTB modules as well as for all FTM modules except for FTM NIMs. See also *Parameters Tab for FTM and FTB Digital Modules, page 111*, *Parameters Tab for FTM Analog Input Modules, page 113* and *Parameters Tab for FTM Analog Output Modules, page 114* for detailed information on the parameters of FTM and FTB modules.
Counters *(see page 106)*	This tab displays the configuration parameters for the counters of the NIMs. Therefore, it is only available for OTB NIMs.
Pulse Generator *(see page 107)*	This tab displays the configuration parameters for the pulse generators of the NIMs. Therefore, it is only available for OTB NIMs.
Options *(see page 108)*	This tab displays the global configuration parameters for accessing the registers of NIM running on a Modbus protocol-based upstream fieldbus network. It is only available for OTB Ethernet and Modbus NIMs, not for the OTB CANopen NIM. It is not available for FTM and FTB modules.

Overview of the Functions

In offline mode, it is possible to

- view general module information,
- edit parameters and
- assign labels to I/O data.

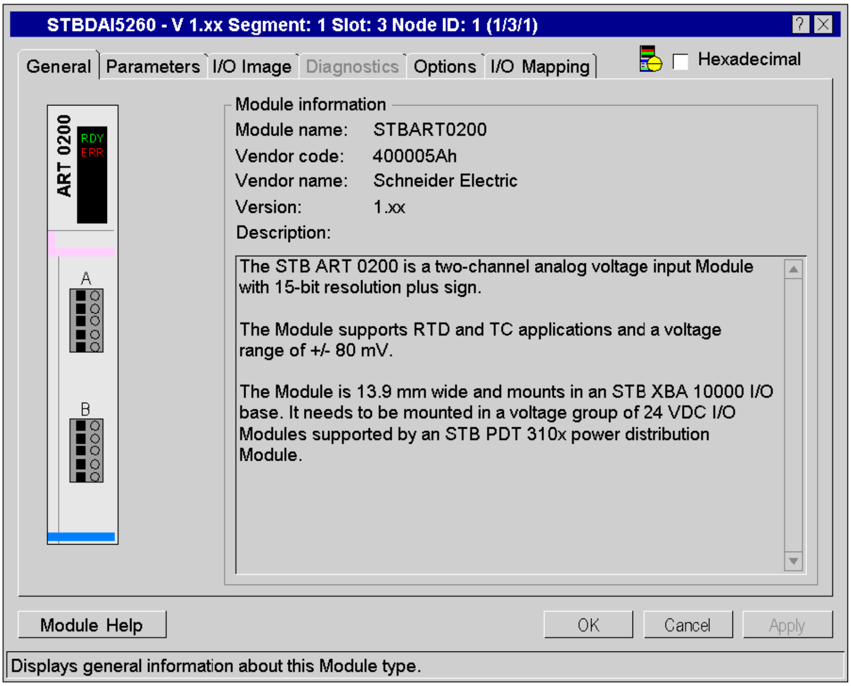
General Tab

Introduction

The **General** tab is common for the modules of all product families. It displays descriptive information on the selected module. The module's name and its exact location on the Island bus are displayed in the title bar of the Module Editor window.

General Tab Figure Example

The following figure shows the information displayed on the **General** tab:



General Tab's Information

The module information provided by the **General** tab is read-only and includes the module and the manufacturer name as well as the manufacturer code for the module. Further, the firmware version is displayed. Finally, the tab contains hardware information and a brief functional description of the module.

Section 3.2
Module Editor for STB Modules

Introduction

This section contains a detailed description of the Module Editor for STB modules. All available tabs are explained.

What Is in This Section?

This section contains the following topics:

Parameters Tab for STB Modules

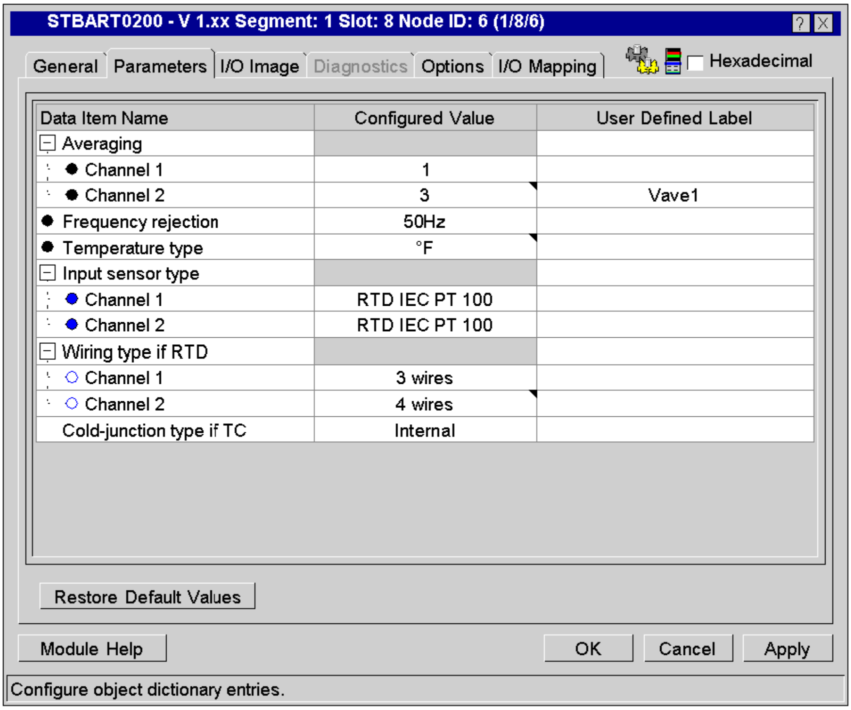
Introduction

You can customize the selected module's operating parameters using the **Parameters** tab in the Module Editor. The name of the module and its exact location on the Island bus are displayed in the title bar at the top of the Module Editor window.

NOTE: The **Parameters** tab appears dimmed (not accessible) for accessories, power supply modules and basic modules.

Parameters Tab Figure Example

The following figure shows the **Parameters** tab information:



NOTE: You cannot configure values or labels when the Island is locked or online. For all editable parameters, the valid value range is displayed in the status bar of the Module Editor.

Data Item Name Column

Operational parameters that characterize the selected module are listed in the **Data Item Name** column. There are essentially 2 types of module parameters:

- Structured parameters containing subordinated objects, mostly channels. To the left of these parameters, you will notice a **+/-** box that allows you to expand the list down to the channel level and collapse it up to the parameter level. **Polarity** and **Input sensor type** are examples of this type.
- Plain parameters applicable only at the module level. When this parameter value is set, it applies to all the channels on the module. This type of parameter, for instance the **Frequency rejection**, is preceded by a bullet.

There are 3 different types of bullets:

Bullet	Description
●	The black bullets indicate normal, independent parameters.
●	The blue bullets indicate master parameters.
○	The blue circles indicate slave parameters.

Configured Value Column

The **Configured Value** column displays the operational values for the parameters of the selected module. Some of these parameter values are read-only; others can be edited. Read-only parameter values are displayed in the dimmed cells.

Some parameter values are text strings, and others are integer values. If the value is integer, you can either choose to display it in hexadecimal format by checking the **Hexadecimal** box, or to display it in decimal format by clearing the same box.

Some user-configurable parameter values have a limited set of acceptable values (e.g. a fallback mode value of either **Hold last value** or **Predefined**, a boolean **0** or **1**). In this case, the value of the parameter is selected from a pull-down list box displayed when you click the appropriate cell in the **Configured Value** column. Other user-configurable parameter values may be integers from a wide range of values (e.g. an analog integer value in the range **0...32,000**). When you click the cell of a parameter that accepts a range value, you need to type the desired integer value (in the appropriate hexadecimal or decimal format). For module-specific information on the acceptable range of values, click the **Module Help** button.

If you have modified a parameter value, a dog ear is displayed in the upper right corner of the corresponding value field. In addition, the following icon appears to the left of the **Hexadecimal** check box, visible on all tabs of the Module Editor:

Example of a Digital Module

If you are editing parameter values that may be changed at the individual channel level, for instance of the digital module DDO 3230, you can change these parameters either at the channel level or at the module level. **Polarity**, for instance, can be set independently on each channel to 1 of 2 possible values: **0 - Normal** (the default) or **1- Reverse**. A setting of **0 - Normal** means that the polarity of the input on the selected channel is normal; a setting of **1- Reverse** means reverse. Suppose you want to change the polarity of input channel 2 from normal to reverse (from **0 - Normal** to **1- Reverse**).

To make the change at the channel level:

Step	Action
1	Expand the **Polarity** list by clicking the **+/-** box in the **Data Item Name** column so that you see **Channel 1** and **Channel 2**.
2	In the **Configured Value** column, click **Channel 2**. **Result:** A drop-down list box appears.
3	Select the value **1 - Reverse** from the list box and click outside of the column.

Notice that the value associated with **Channel 2** is now **1** and that (after pressing RETURN or clicking at another channel) the value associated with **Polarity** at **Module level** in the list is now **1**. The configuration software handles the channel values as bits in a byte that defines the polarity parameter for the module.

That means the following:

If...	Then ...
Channel 1 = 0 - Normal and **Channel 2 = 0 - Normal**	**Polarity = 0 - Normal**
Channel 1 = 1 - Reverse and **Channel 2 = 0 - Normal**	**Polarity = 1 - Reverse**
Channel 1 = 0 - Normal and **Channel 2 = 1 - Reverse**	**Polarity = 2**
Channel 1 = 1 - Reverse and **Channel 2 = 1 - Reverse**	**Polarity = 3**

You can change the polarity of channel 2 at the module level, even if the **Polarity** list is collapsed.

Step	Action
1	In the **Configured Value** column, click **Polarity**.
2	Type `2` in this column.

If you expand the **Polarity** list, you will see that the value associated with **Channel 2** is now **1 - Reverse**, including enumeration values.

Parameter Dependencies

The **Parameters** tab is supporting parameter dependencies. If a master parameter value is changed, the rules for all slave parameters depending on this master parameter will be re-evaluated, that is

- the slave parameter will be enabled/disabled or,
- the slave parameter's range of selectable values will be reduced/enlarged.

If a slave parameter is disabled due to the re-evaluation of a rule, the value of this slave parameter will be reset to the associated default value.

User Defined Label Column

You can write a custom label or note for any of the parameters listed on the **Parameters** tab. This feature allows you to pre-symbolize important memory locations in the Island before the application is written.

Double-click in the appropriate cell in the **User Defined Label** column to enter the label text.

The labels should not be duplicates and they have to be compliant to the IEC61131 rules:

- Only alphanumeric and underscore characters can be used.
- The first character has to be an alphabetic character.
- Blanks and non-ASCII characters are not allowed.
- The overall length of the label should not exceed 24 characters.

Restore Default Values

If you click the **Restore Default Value** button, all values that can be modified on this tab will be set to their default values.

NOTE: All user-defined values modified on this tab will be deleted, but not the labels.

Saving or Canceling Parameter Changes

If you have modified any operational parameter values displayed on the **Parameters** tab, the **Apply** button at the bottom of the Module Editor window becomes enabled. To validate your modifications for the selected module

- click the **Apply** button to accept the changes, or
- click the **OK** button to apply the changes and close the dialog.

If you decide not to save the changes you have made to the **Parameters** tab, click the **Cancel** button.

Ethernet Parameters Tab

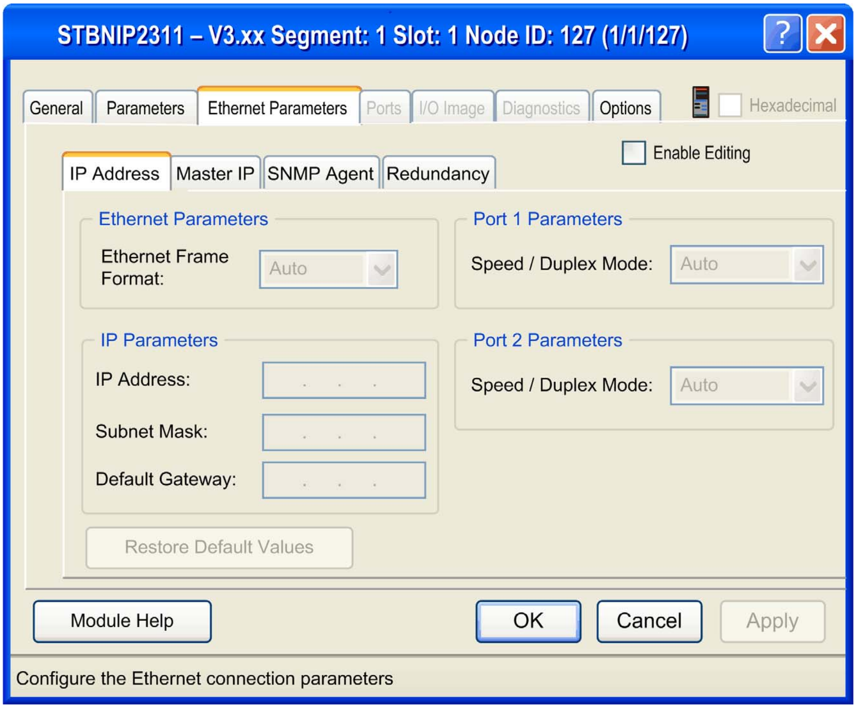
Introduction

You can customize Ethernet specific parameters using the **Ethernet Parameters** tab. It contains 4 subordinate tabs. The name of the module and its exact location on the Island bus are displayed in the title bar at the top of the Module Editor window.

NOTE: The **Ethernet Parameters** tab is only accessible for the STB NIP2311 Ethernet NIM.

Figure Example

This figure shows the **Ethernet Parameters** tab with the **IP Address** subtab selected:



Enable Editing Check Box

Depending on the status of the **Enable Editing** check box, you can use the web pages of the NIM or the Advantys Configuration Software to edit the Ethernet specific parameters:

If ...	Then you are ...
the check box is deactivated (default value)	allowed to use the web pages of the NIM for editing. The actual parameter values are read from the NIM and updated at least once every five seconds.
the check box is activated	allowed to use the Advantys Configuration Software for editing.
the corresponding Island is in online mode	not allowed to edit any parameters independent of the status of the check box. In case the **Enable Editing** check box is disabled, the current parameter values are read from the NIM and updated at least once every five seconds.
you disable the check box and changes are pending	allowed to apply, discard or even correct your changes.

Restore Default Values

Every subtab has a **Restore Default Values** button. If you click this button, all values that can be modified on this subtab will be set to their default values.

NOTE: All user-defined values modified on this tab will be deleted, but not the labels.

Hexadecimal Check Box

The global **Hexadecimal** check box has no effect on the contents of this tab. It is grayed out when this tab is selected.

Subtabs of the Ethernet Parameters Tab

IP Address Subtab

The **IP Address** subtab is the default subtab that opens when the **Ethernet Parameters** tab is selected. In offline mode, you can use the **IP Address** subtab to configure the parameters associated with attaching the NIM to a network, such as IP and port parameters. In online mode, this subtab displays the current values of these stored parameters.

For a figure example, refer to the Ethernet Parameters tab *(see page 79)*.

You can configure the following parameters:

Parameter Name	Description
Frame Type	Use the pull-down list box to select 1 of the following: ● Ethernet II ● IEEE802.3 ● Auto (default value)
IP Address	Leave the field empty or enter 4 octets. Each can be 0...255. The default value is 0.
Subnet Mask	If you have filled in the **IP Address** field, enter 4 octets. Otherwise, you can leave this field empty. Each octet can be 0...255. The default value is 0.
Gateway	Leave the field empty or enter 4 octets. Each can be 0...255. The default value is 0. If given, these data has to match the IP address network.
Speed / Duplex Mode	Use the pull-down list box for each port to select 1 of the following: ● Auto (default value) ● 10 MBit/s - Half ● 10 MBit/s - Full ● 100 MBit/s - Half ● 100 MBit/s - Full

Master IP Subtab

In offline mode, you can use the **Master IP** subtab to configure the parameters associated with master (PLC) connections to the NIM. In online mode, this subtab displays the current values of these parameters.

You can configure the following parameters:

Parameter Name	Description
Master x IP Address	Leave the field empty or enter 4 octets. Each can be 0...255. The default value is 0.
Reservation Time	Enter a value in the range 0...120 000. The default value is 60 000 ms. **NOTE:** The reservation time is the value displayed divided by 10. Therefore, enter a value that is a multiple of 10 or use the arrow buttons to modify it in steps of 10.
Holdup Time	Enter a value in the range 300...120 000. The default value is 1000 ms. **NOTE:** The holdup time is the value displayed divided by 10. Therefore, enter a value that is a multiple of 10 or use the arrow buttons to modify it in steps of 10.
Link Failure Mode	Use the pull-down list box to select the behavior in case of a detected error. The default value is the fallback mode.

SNMP Subtab

In offline mode, you can use the **SNMP** subtab to configure the parameters associated with SNMP settings. In online mode, this subtab displays the current values of these parameters.

You can configure these parameters as follows:

Parameter Name	Description
System Name, System Contact, System Location	These fields are read-only.
Manager 1, Manager 2	Leave the fields empty or enter IP addresses of 4 octets each. Each octet can be 0...255.
GET, SET, TRAP Community Name	Leave the fields empty or enter up to 16 printable ASCII characters. The default value is *public*.
Trap Enables features	Use the check box to enable or disable the corresponding feature. By default, all these features are disabled.

Redundancy Subtab

In offline mode, you can use the **Redundancy** subtab to enable or disable RSTP and to set the bridge priority value. In online mode, this subtab displays the current values of these parameters.

Use the check box to enable or disable RSTP.

The parameters associated with the RSTP setting are configured using the bridge priority value. Select the appropriate value in the **Bridge Priority** list.

NOTE: The **Redundancy** subtab is accessible only for STB NIP2311 Ethernet NIM version 3.0 or later.

Ports Tab

Introduction

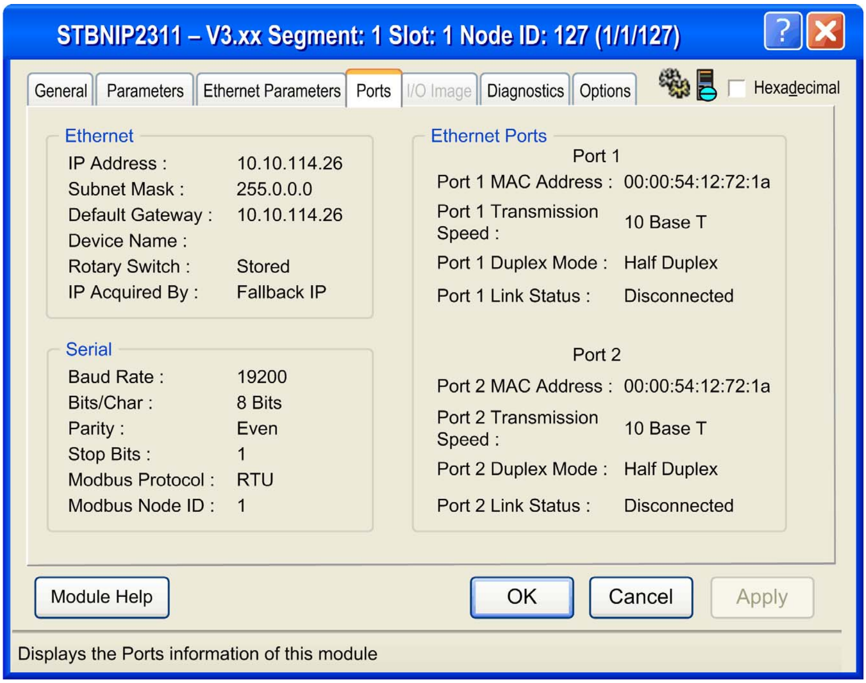
In online mode, the **Ports** tab displays the current operating values of certain port parameters. In offline mode, this tab is grayed out. The name of the module and its exact location on the Island bus are displayed in the title bar at the top of the Module Editor window.

NOTE:
The **Ports** tab is only accessible for the

- STB NIP2311 Ethernet NIM
- STB NCO2212 CANopen NIM version 3.05 or later

Ports Tab of STB NIP2311

The following figure shows the **Ports** tab information of STB NIP2311 Ethernet NIM:



Parameters

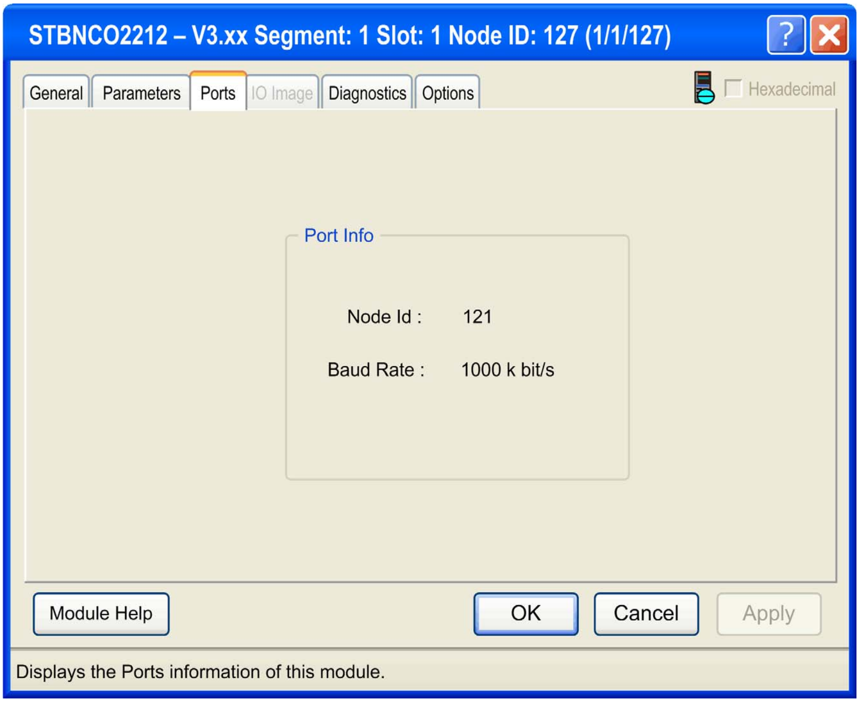
Most of the parameters displayed are self-explanatory. The **Rotary Switch** field contains the current rotary switch position, which is not necessarily the switch position at boot time. The **IP Acquired By** field displays the method used by the NIM to obtain its IP address information.

Hexadecimal Check Box

The global **Hexadecimal** check box has no effect on the contents of this tab. It is grayed out when this tab is selected.

Ports Tab of STB NCO2212

The following figure shows the **Ports** tab information of STB NCO2212 CANopen NIM version 3.05 or later:



Parameters

The current values of **Node Id** and **Baud Rate** are displayed only in the online mode.

Hexadecimal Check Box

The global **Hexadecimal** check box has no effect on the contents of this tab. It is grayed out when this tab is selected.

I/O Image Tab

Introduction

On the **I/O Image** tab, you can read and modify the I/O data of a module.

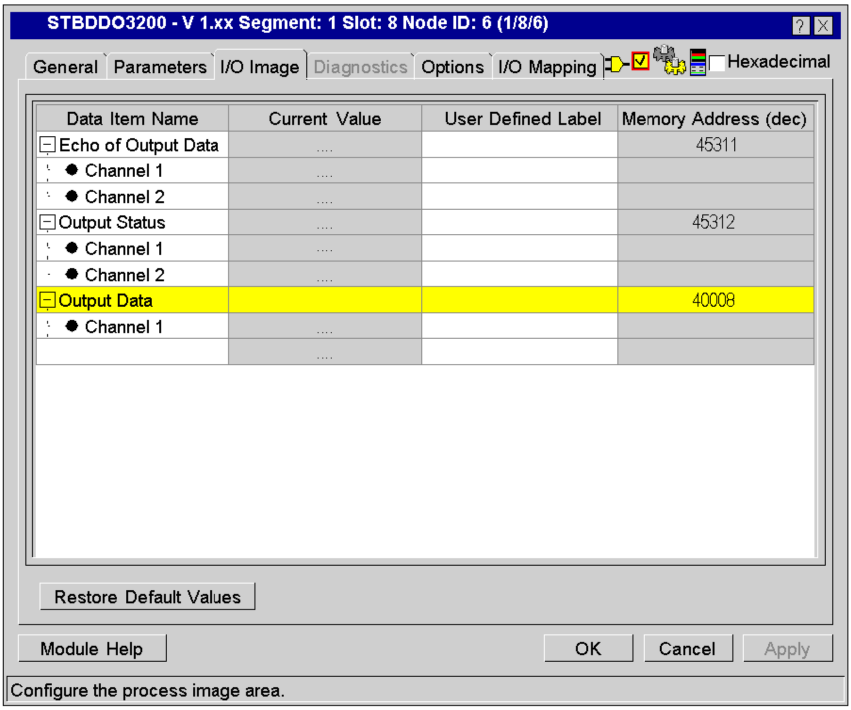
You can change the format of the values from the decimal to the hexadecimal format by checking or clearing the **Hexadecimal** check box.

The name of the module and its exact location on the Island bus are displayed in the title bar at the top of the Module Editor window.

NOTE: The **I/O Image** tab is not available for accessories and power supply modules.

I/O Image Tab Figure Example

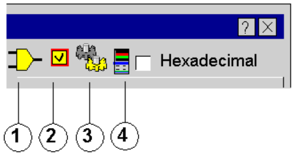
The following figure shows the **I/O Image** tab information:



On the **I/O Image** tab, complete rows of output controlled by reflex actions are marked in yellow.

Status Icons

The following figure shows the status icons:



Number	Description
1	The **Reflex Action** icon appears if the module hosts at least 1 reflex action.
2	The **Checkmark** icon appears if at least 1 I/O mapping has been modified.
3	The **Parameter Change** icon appears if at least 1 parameter has been modified.
4	The **Locked/Online** status is shown by an icon. The icon states and their respective meanings are similar to the Workspace Browser.

Data Item Name Column

Operational parameters that characterize the selected module are listed in the **Data Item Name** column.

Current Value Column

The **Current Value** column displays the actual values for the I/O points. If you check the **Hexadecimal** box, the values are displayed in hexadecimal format. If you clear the same box, the values will be displayed in decimal format.

NOTE: The actual values are only displayed if the Island is in either the operational state or in the non-mandatory module mismatch state. In all other states, **---** is displayed.

User Defined Label Column

You can write a custom label or note for any of the data items listed on the **I/O Image** tab. This feature allows you to pre-symbolize important memory locations in the Island before the application is written.

In the **User Defined Label** column, double-click in the appropriate cell to enter the label text. Each label can be only up to 24 characters long.

Predefined Labels

If you configure the PLC-to-HMI area, the HMI-to-PLC area or use virtual modules for reflex actions, the Advantys Configuration Software automatically generates user-defined labels for these data items. These labels are displayed in the **User Defined Label** column. They are grayed out and you cannot edit them. The software uses these generated labels for the export.

Memory Address

The displayed memory address represents the Modbus register.

NOTE: Values in the **Memory Address** column will only be displayed for parent nodes. The memory address is **read-only**.

Modifying Module Output Data Online

Output data can only be modified in test mode.

Proceed as follows to change the output data values:

Step	Action
1	Open the **I/O Image** tab in the Module Editor.
2	Double-click to enter any valid value in the **Current Value** column of the corresponding I/O data. **Note:** The test mode has to be activated. You cannot modify the values specified in the dimmed cells. For I/O data specified as channels and enumerated strings, a combo box is displayed when you double-click to enter any valid value to modify the value. Select the specified value from the listed values of the combo box. When you double-click to enter any valid value in the **Current Value** column, the status bar displays the minimum and maximum configuration limits. **Note:** They will be not displayed for channels and enumerated strings. Enter the value based on the information specified in the status bar. **Note:** Yellow cells display values from reflex actions.
3	Press ENTER or RETURN to set the cell value to the physical Islands. You can also set the modified value by moving on to another cell.

NOTE: Values in the **Memory Address** column will only be displayed for parent nodes. The memory address is **read-only**.

Diagnostics Tab

Introduction

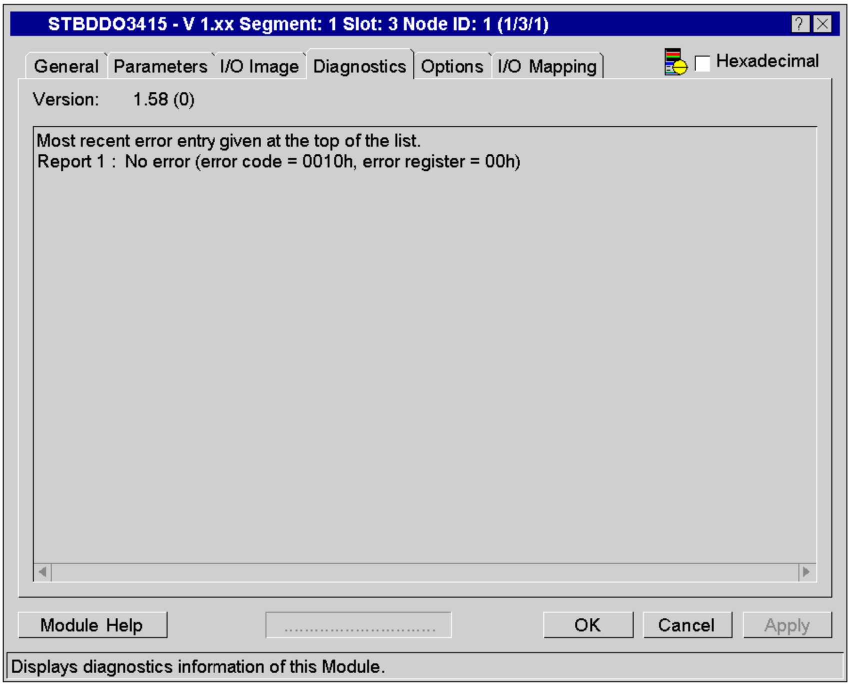
The **Diagnostics** tab displays any detected error reported by the selected module. The name of the module and its exact location on the Island bus are displayed in the title bar. The module version is displayed at the top of the **Diagnostics** tab.

Data displayed in the **Diagnostics** tab are read-only. For I/O modules, a maximum of 4 entries is displayed. For a NIM, 16 device status bits are displayed in binary format: 8 general and 8 fieldbus-specific diagnostic bits with textual evaluation.

NOTE: The **Diagnostics** tab is only available when the software is connected to a physical Island and running in online mode. It is dimmed in offline mode and for modules that do not communicate on the Island bus, such as power supplies.

Diagnostics Tab Figure Example

The following figure shows the **Diagnostics** tab information:



Diagnostic Codes

The following table describes some of the diagnostic codes reported by the modules and the corrective actions if any that are required:

Diagnostic Code	Meaning	Most Probable Cause of Detected Error	Suggested Corrective Action(s)
0x0000	no detected error or 1 detected error cleared	for information only	none required
0x0010	node started	for information only, no detected errors	none required
0x2320	short-circuit	output short	Remove the output short. **Note:** The outputs may remain latched off.
0x3000	24 VDC detected error	24 VDC field power shorted or missing or not within valid range	Remove field power short or add field power or check it is within the correct range.
0x3110	over-range	out-of-range signal	Check that the signal is within the range.
0x3120	under-range	out-of-range signal	Check that the signal is within the range.
0x6000	internally detected error	internally detected error in bus due to inoperable module(s), improper module setup	Check the Island setup, check the module setup, replace the inoperable module(s), reset the Island, power cycle the Island.
0x8110	CAN overrun		
0x8120	detected error in CAN		
0x8140	CAN bus off		
0x8210	detected error in data packet		
0x8130	inoperable heartbeat	detected loss of fieldbus communication, NIM communication interruption or internally detected error in bus	Check the Island setup and the NIM, check the fieldbus connection, replace the inoperable device(s), reset the Island, power cycle the Island.
0xF010	detected error in reflex action	detected loss of fieldbus communication, detected loss of communication with a peer module, internally detected error in bus, inoperable module(s)	Check the Island setup, check the fieldbus connection, replace the inoperable module(s), reset the Island, power cycle the Island.
0xFF00	inoperable device	inoperable module, improper module configuration	Check the device setup, replace the inoperable device(s), reset the Island, power cycle the Island.

Options Tab for STB Modules

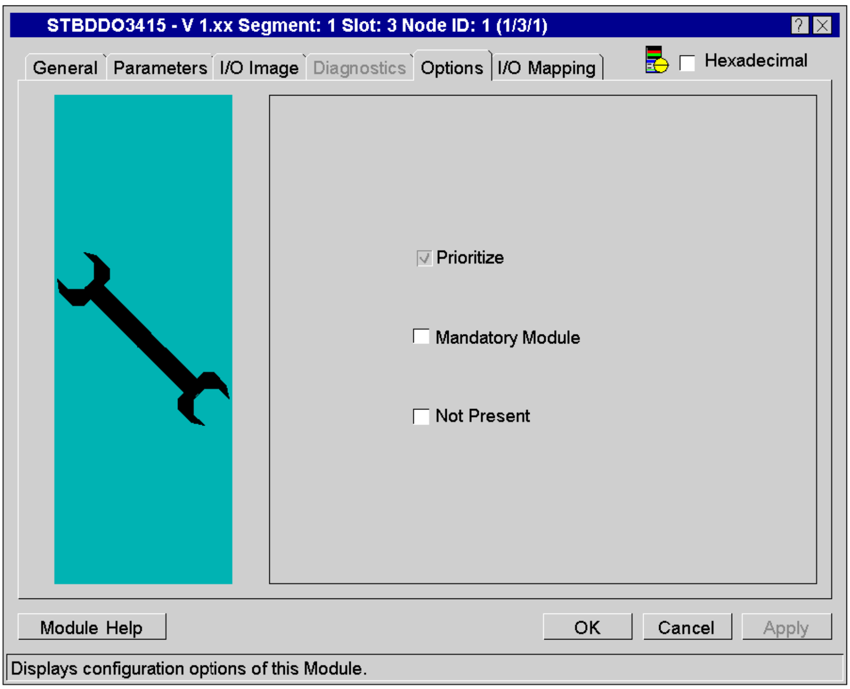
Introduction

You can find optional settings on the **Options** tab. Depending on the module, the following check boxes are displayed:

Module Group	Check Boxes
I/Os	• Prioritze • Mandatory Module • Not Present
NIMs	• Configure run-time parameters • Max. node ID on the CANopen extensions (dec) **Note:** The **Options** tab is not available for basic NIMs, accessories and power supply modules.

Options Tab Figure Example

The following figure shows the **Options** tab information of an output module:



Prioritize

Check this box to include the selected I/O module in a group of fast-solve modules that are scanned by the NIM more frequently than other modules. Typically, this operational parameter can only be applied to digital I/O modules with a fast response time. Most Advantys STB analog I/O modules cannot be prioritized. By default, the software automatically prioritizes the first 10 prioritizable modules. If the Island consists of more than 10 prioritizable modules, prioritize the modules manually.

NOTE: The **Prioritize** check box is not present in online mode.

Mandatory Module

Check this box to designate the selected I/O module as mandatory. If a mandatory module stops working or is removed from the Island, the entire Island bus will switch to pre-operational mode and stop. It will return to its operational state only if you reinstall the same functional module, or a new module of the same type, at this exact location on the bus. By default, no module is mandatory, and this box is clear.

NOTE: The **Mandatory** check box is not available for all modules.

NOTE: The **Mandatory** check box is not present in online mode.

Not Present

Check this box to mark the module as Virtual Placeholder.

The Virtual Placeholder allows you to remove certain physical Island I/O modules from a base configuration while keeping the identical process image. Thus, you can define an Island with various options removed without changing the PLC program which controls the Island. If you have removed some STB modules, the remaining ones should be plugged physically next to each other because spare slots are not allowed.

In the Module Editor, the Virtual Placeholder modules are marked with crossed red lines.

NOTE: If signals from a Virtual Placeholder module which is physically not present are used as input data for a reflex action, **no** reflex action in that module will function.

For more information on Virtual Placeholders, please refer to Virtual Placeholders *(see page 236)*.

Making an I/O Module Mandatory

You may want to configure an I/O module as mandatory if it performs a function that is critical to the system's operation. If, for any reason, the NIM does not detect a healthy mandatory module at the expected location on the Island bus, the Island will stop functioning and all the modules on the Island will default to their fallback states.

NOTE: If you hot-swap a mandatory I/O module, the Island will not continue to run.

You can configure a module as mandatory only when the configuration software is offline. The **Mandatory** option is disabled when the software is online.

By default, all Advantys STB modules are configured as standard (not mandatory). There is no limit to the number or type of I/O modules that can be configured as mandatory.

NOTE: If the island contains the enhanced CANopen modules, selecting **Mandatory Module** causes build interruption.

Configuring Run-Time Parameters: An Example

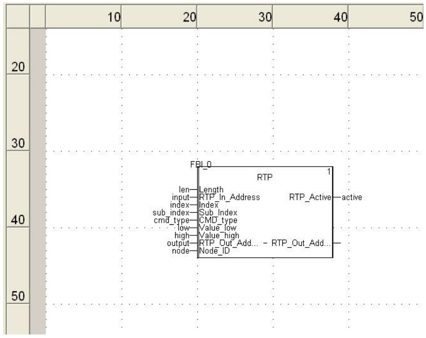
This option is available for STB NIMs version V 2.*xx* or later. If this box is checked, a set of registers in the fieldbus image is reserved. These registers allows to control the transfer of parameters at application program level using normal I/O operations. These registers are indicated in the **I/O Image** as **RTP** *(see page 123)*.

NOTE: To use this application example, you should be familiar with Unity Pro and Advantys configuration software.
Ensure that:
- Advantys STB modules are fully assembled and installed according to particular system, application, and network requirements.
- you have the required devices.

The following table shows the procedure to create a project for Quantum PLC in Unity Pro:

Step	Action	Comment
1	On the **File** menu, select **New**.	–
2	From the **New Project** window, select the **Quantum CPU**.	For example, 140 CPU 651 60.
3	Click **Project browser** -> **Configuration** -> **Local Bus**.	**Local Bus** window opens.
4	Right-click the empty slot of rack and select **New device** from the context menu.	**New device** window opens.
5	From the **New device** window, select the communication module **140NOC 771 01**.	**Properties of Device** dialog opens.
6	In the **Properties of Device** dialog, click **OK**.	–
7	In the **Local Bus** window, double-click the CPU module.	New dialog with CPU data opens.

Step	Action	Comment
8	Go to the **Configuration** tab -> **State RAM** area.	Change the **%MW** value based on requirement.
9	Go to the **Local Bus** window, double-click the **NOC 771 01** module.	New dialog with NOC module data opens.
10	Go to the **Configuration** tab and enter the values in **Inputs** and **Outputs** areas.	For example, Inputs **%MWindex**: 1000 and **Max size**: 100 Outputs **%MWindex**: 2000 and **Max size**: 100
11	Validate the changes.	–
12	Click **Update application**.	Variables are created.
13	Select the **DTM Browser** option from the **Tools** menu of Unity Pro.	**DTM Browser** opens.
14	Right-click the **Communication DTM Q_NOC77101** and select **Add** option from the context menu.	**Add** dialog opens with list DTMs avaialable.
15	Select the **STB NIC2212** device DTM from the list of DTMs available.	–
16	Click **Add DTM**.	–
17	In the **Properties of Device** window, click **OK**.	–
18	Double-click the **STB NIC 2212 DTM**.	**STB DTM** window opens.
19	Click **Start Advantys**.	Advantys configuration software launches.
20	Create an island in Advantys configuration software with inputs and outputs module.	–
21	Double-click the **STBNIC2212** module in Advantys configuration software.	Module editor window opens.
22	Go to **Options** tab of the module editor window, select the **Configure Run-Time Parameters** check box and close the Module editor.	–
23	Save the island in Advantys configuration software and close the application.	–
24	Go to Unity Pro. Click **Apply** in the **STB DTM** window.	–

Step	Action	Comment
25	Create an FBD to configure the run-time parameters as shown in the following figure:  	Refer to the Run-Time Parameters *(see page 124)* for the valid RTP commands.

Max. Node ID on the CANopen Extension

Enhanced CANopen devices are connected as last devices on the Island using a CANopen extension module. The addressing of these modules starts with 32 counting down while the internal modules start with 1 counting up.

Should the addressing capability of your CANopen device allow only an address below 32, enter this address in this field. Bear in mind that this will reduce the number of possible internal Island modules.

I/O Mapping Tab

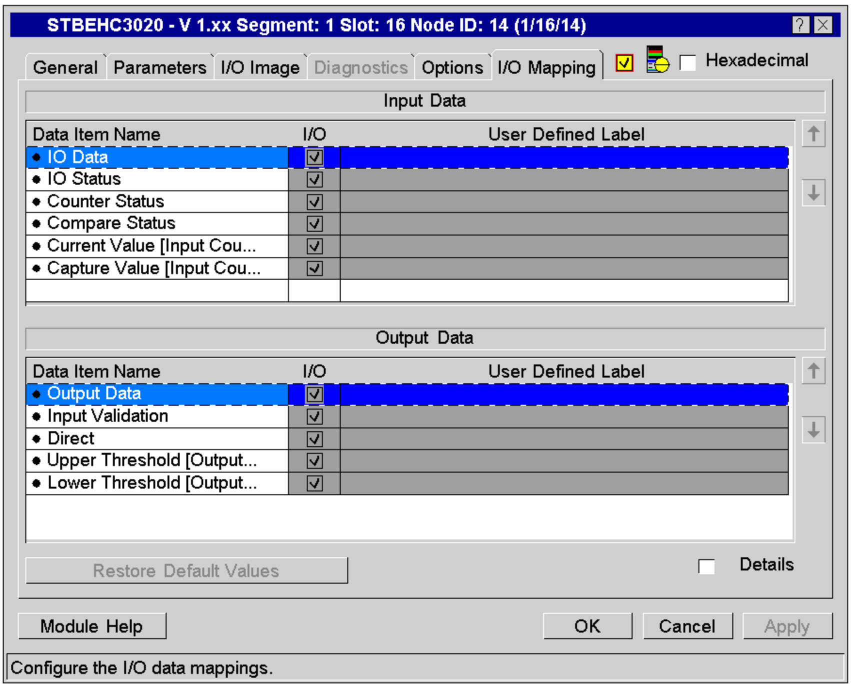
Introduction

You can monitor and edit the I/O mapping of the selected module using the **I/O Mapping** tab in the Module Editor. This dynamic I/O mapping allows you to optimize the Island's process image on a module-by-module basis.The name of the module and its exact location on the Island bus are displayed in the title bar at the top of the Module Editor window.

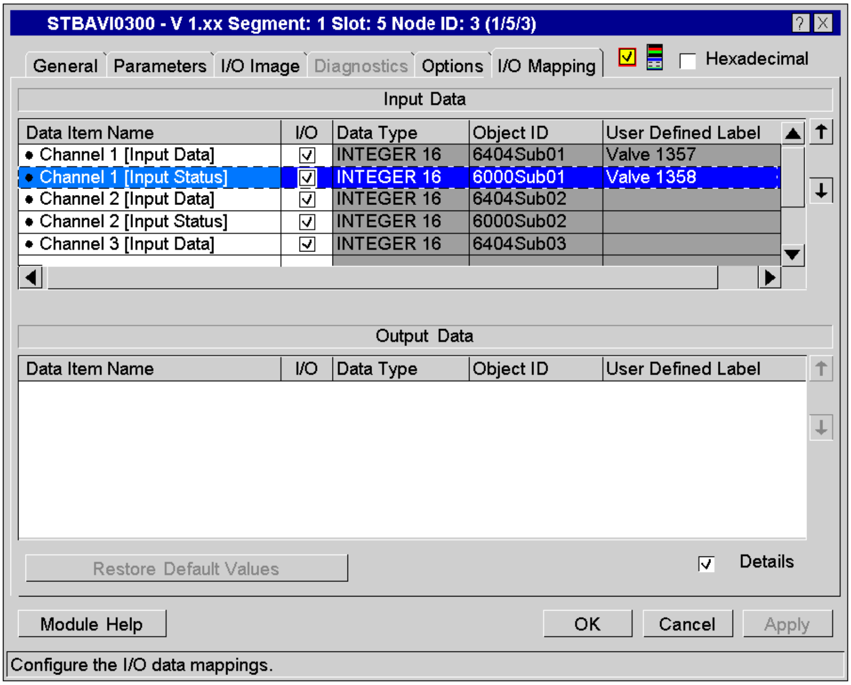
NOTE: The **I/O Mapping** tab is not available for NIMs, basic I/Os, digital VAC input modules, accessories and power supply modules.

I/O Mapping Tab Figure Examples

The following figure shows the **I/O Mapping** tab information:



The following figure shows the **I/O Mapping** tab information with the **Details** check box selected:



Input/Output Data Areas

In the **Input Data** and the **Output Data** areas, all data items that can be mapped are listed. Whether a data item is displayed in the **Input Data** or in the **Output Data** area depends on the access type of the object, for instance read-only vs. write-only.

Data Item Name Column

In the **Data Item Name** column, all mapped and unmapped data items are displayed. They are sorted so that the mapped data items are shown in a block at the beginning of the respective list. The sequence of the data items within the block resembles the sequence of the data items in the I/O image.

I/O Column

The **I/O** column shows the current I/O mapping status of an item. If an Island produces a process image that contains more information than you need for your application, then you can suppress data items by deselecting the check box. If a module provides more information than the default process image supports, then you can add data items by selecting the check box.

Dynamic I/O Mapping

As soon as you deselect data items in the **I/O** column, they are removed from the **I/O Image** tab. As soon as you select additional data objects in the **I/O** column, they are displayed in the **I/O Image** tab.

NOTE: If you deselect an output data item, it is displayed in the **Parameters** tab. If the deselected output data item has a read-only attribute, then it is not displayed in the **Parameters** tab.

Dynamic I/O mapping allows you to optimize the size of your process image on a module-by-module basis. If your island is producing a process image that contains more information than you need for your application, or if a module on the island actually provides more information than the default process image supports, dynamic I/O mapping lets you to suppress the information that is not needed and to add the information that was previously missing.

An **I/O Mapping** tab appears in the Module Editor for any Advantys STB modules that support dynamic I/O mapping (some I/O modules that produce very small amounts of process image information do not show this tab in the Module Editor). When you click this tab, the information that appears on the screen directly reflects the types of information that appear on the **I/O Image** screen, that is, module status, data, echo output data, and so on. A check box appears next to each information type and a check mark appears in each check box.

To suppress an information type from the process image, simply clear the associated check box. For example, your application may not use echo output data from the digital output modules. Open the Module Editor for each of the digital output modules, go to the **I/O Mapping** screen, and clear **Echo Data** information type. This action suppresses the information from the process image and reduces the overall size of the image.

In other cases, you can use a module such as the ATV61 variable frequency drive, which provides additional diagnostic information that is not automatically captured in the process image. For these modules, the **I/O Mapping** screen gives the ability to add more registers to the process image so that this information can be delivered to the fieldbus master.

NOTE: Added information is read-only. You cannot use dynamic I/O mapping for adding write-control registers to the process image.

After making adjustments on the **I/O Mapping** screen, the unchecked information is removed from the **I/O Image** screen.

When you see the resulting process image on the **I/O Image Overview** screen, all unchecked information types are no longer present and the new registers that you added appear in the image.

Additional Diagnostic Information

ATV31x, ATV61, and ATV71 drives on the STB CANopen extension provide the capability to include the following additional diagnostic information:

- high speed and low speed setting
- output frequency
- frequency reference
- motor current and motor voltage
- power supply voltage
- thermal state of the drive
- motor torque and so on

Indication of Changes

Changes from the default are indicated as follows:

If you ...	Then ...
• deselect default data • add data that was not part of the default image	a solid dog ear is displayed in the upper right corner of the corresponding field.
• change the I/O mapping and thereby change the data item sequence • move a data item within the sequence using the arrow buttons	a hollow dog ear is displayed in the upper right corner of the fields of those data items affected by the sequence change.

In addition, the following icon is displayed to the left of the **Hexadecimal** check box each time a default value of the I/O mapping has been changed:

User Defined Label Column

In this column, the labels associated with the data items are shown. The label can be assigned in the **I/O Image** tab or in the **Parameters** tab of the Module Editor.

Arrow Buttons

The currently selected mapping can be moved within the list of mapped data items using the following buttons or keys:

- the arrow buttons on the right side of the **Input Data** and the **Output Data** areas
- the keys CTRL+UP ARROW and CTRL+DOWN ARROW

Restore Default Values

If you have modified any entries displayed on the **I/O Mapping** tab, these are marked and the **Restore Default Values** button at the bottom of the Module Editor window becomes enabled. If you click this button, all values will be reset to their respective default values including the sequence of the data items and the check boxes. In online mode, the **Restore Default Values** button is not available.

Saving or Canceling Changes

If you have modified any entries displayed on the **I/O Mapping** tab, the **Apply** button at the bottom of the Module Editor window becomes enabled. To validate your modifications for the selected module

- click the **Apply** button to accept the changes, or
- click the **OK** button to apply the changes and close the dialog.

If you decide not to save the changes you have made to the **I/O Mapping** tab after you have lastly used the **Apply** function, click the **Cancel** button.

Hexadecimal Check Box

The global **Hexadecimal** check box has no effect on the contents of this tab.

Details Check Box

If you select the **Details** check box, the following additional CANopen information concerning the local Island bus mapping is displayed in the **Input Data** and the **Output Data** areas:

Column	Description
Data Type	displays the data type of the CANopen I/O data objects mapped to the local Island bus
Object ID	displays the index / subindex of the CANopen I/O data objects mapped to the local Island bus

Handling Changes

Mappings may be defined as fixed. In this case, the I/O data item can neither be changed, for instance moved within the list, nor can it be removed from the I/O image of the module.

After each change of the mapping, the tab is updated and the resulting I/O view and mapping order is displayed. Changes have to be confirmed in a modal notification box that informs about the effects on a PLC application program context.

Section 3.3
Module Editor for OTB Modules

Introduction

This section contains a detailed description of the Module Editor for OTB modules. All available tabs are explained.

What Is in This Section?

This section contains the following topics:

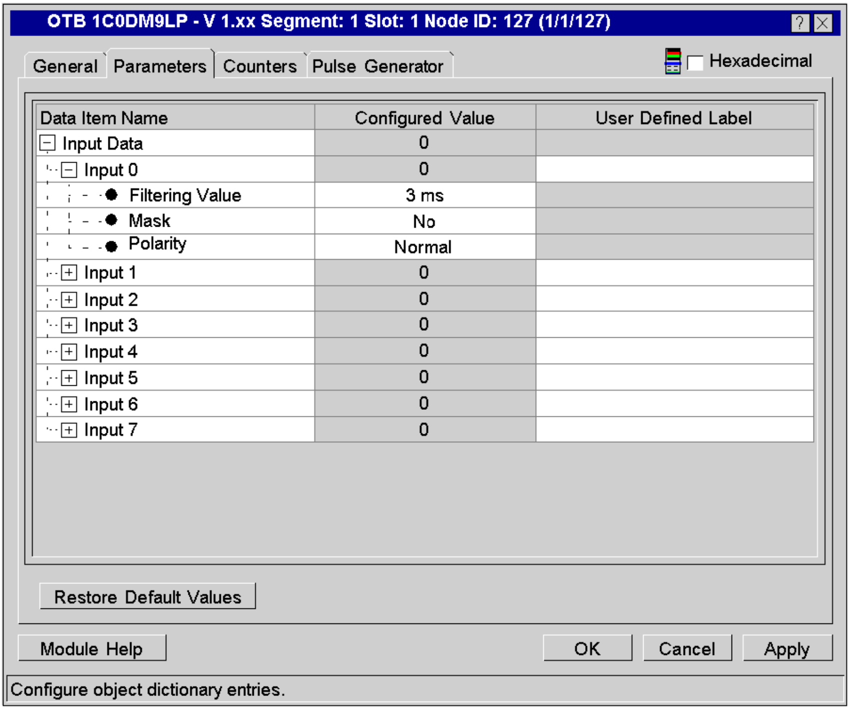
Parameters Tab for OTB Modules

Introduction

The **Parameters** tab displays all input and output data items of the module selected. For each data item, you can customize the parameters available. The name of the module and its exact location on the Island bus are displayed in the title bar.

Parameters Tab Figure Example

This figure shows the **Parameters** tab information of an OTB CANopen NIM:



Tab Structure

The **Parameters** tab is organized and structured in the same way the **Parameters** tab of the STB Module Editor is. That means that the configuration parameters are displayed in a hierarchical tree view comprising master parameters and slave parameters. The only difference is that the parameters of OTB modules are assigned to each single data item, that is the data item is displayed as superordinated. In contrast, parameters of STB modules are listed as superordinated and the data items are assigned to them. For details, refer to *Parameters Tab for STB Modules, page 74*.

Which parameters are available depends on the NIM type and on the module type.

Parameters for Digital I/Os

This table contains a description of the digital I/O parameters and their availability for the different OTB modules:

Parameter	Description
Data Item Name	name of the data item
Mask	enabling or disabling the ● input filtering (only for inputs within Islands containing OTB CANopen NIMs) ● output propagation (only for outputs within Islands containing OTB CANopen NIMs)
Polarity	selecting between normal or reverse polarity (only for inputs and outputs within Islands containing OTB CANopen NIMs)
Filtering Value	setting the value for the input filter constant (only for the inputs of OTB CANopen NIMs)
Fail State	setting the value for the output fallback state
User Defined Label	user-defined label of the parameter

Parameters for Analog Inputs

This table contains a description of the parameters for analog input modules:

Parameter	Description
Data Item Name	name of the data item
Mode	selecting the range mode for the input
Upper Limit	enabling and selecting the value for the upper limit
Lower Limit	enabling and selecting the value for the lower limit
Delta	enabling and selecting the value for the difference
Range	selecting the value range that is used by the module during A/D conversion (standard or custom)
Min	setting the minimum value for the custom range

Parameter	Description
Max	setting the maximum value for the custom range
User Defined Label	label of the parameter

The limit and delta parameters define conditions for the transfer of updated input data on the Island bus. They are only available for Islands with OTB CANopen NIMs. Before you can edit the values, these parameters have to be enabled by selecting **Checked**. For the thermocouple TWDARI8HT, 3 additional parameters (**R**, **T** and **B**) are displayed for entering nominal resistance values.

Parameter Dependencies for Analog Inputs

For some analog input modules, the mode settings have to be the same for all channels. If you change the setting of 1 channel, the Module Editor will adapt the setting of the other ones. Further, the availability of parameters depends on the state of the mode and range parameters as follows:

If you select ...	Then you can edit these parameters:
the mode **Disable**	User Defined Label
a mode other than **Disable** and the range **Standard**	• Upper Limit • Lower Limit • Delta • Range • User Defined Label
a mode other than **Disable** and the range **Custom**	• Upper Limit • Lower Limit • Delta • Range • Min • Max • User Defined Label

Parameters for Analog Outputs

This table contains a description of the parameters for analog output modules:

Parameter	Description
Data Item Name	name of the data item
Mode	selecting the range mode for the input
Range	selecting the value range that is used by the module during A/D conversion (standard or custom)
Min	setting the minimum value for the custom range
Max	setting the maximum value for the custom range
Error Mode	setting the diagnostic mode (hold last value or predefined)

Parameter	Description
Error Value	setting the diagnostic value for the predefined diagnostic mode
User Defined Label	label of the parameter

Parameter Dependencies for Analog Outputs

The availability of parameters depends on the state of the mode and range parameters as follows:

If you select ...	Then you can edit these parameters:
the mode **Disable**	User Defined Label
a mode other than **Disable** and the range **Standard**	• Range • Error Mode • Error Value if you have selected **Predefined** for the diagnostic mode • User Defined Label
a mode other than **Disable** and the range **Custom**	• Range • Min • Max • Error Mode • Error Value if you have selected **Predefined** for the diagnostic mode • User Defined Label

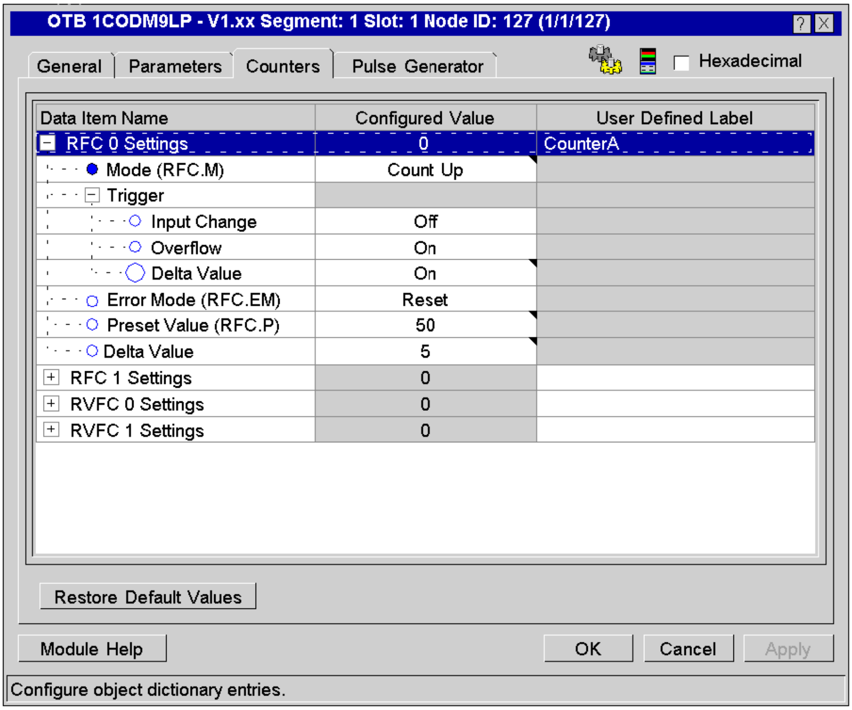
Counters Tab

Introduction

The **Counters** tab displays all configuration parameters for the fast and the very fast counters and is therefore available only for OTB NIMs. The name of the module and its exact location on the Island bus are displayed in the title bar.

Counters Tab Figure Example

The following figure shows the **Counters** tab information:



Tab Structure

The **Counters** tab is organized and structured in the same way the **Parameters** tab of the STB Module Editor is. That means that the configuration parameters are displayed in a hierarchical tree view comprising master parameters and slave parameters. For details, refer to *Parameters Tab for STB Modules, page 74*.

Which parameters are available depends on the NIM type and the counter type.

Pulse Generator Tab

Introduction

The **Pulse Generator** tab displays all configuration parameters for the pulse generators and is therefore available only for OTB NIMs. The name of the module and its exact location on the Island bus are displayed in the title bar.

Tab Structure

The **Pulse Generator** tab is organized and structured in the same way the **Parameters** tab of the STB Module Editor is. That means that the configuration parameters are displayed in a hierarchical tree view comprising master parameters and slave parameters. For details, refer to *Parameters Tab for STB Modules,* *page 74*.

Which parameters are available depends on the NIM type. The trigger parameter is available only for Islands with OTB CANopen NIMs.

Which parameters are editable depends on the pulse generator type. The following pulse generator types are available:
- remote pulse genrators (RPLS)
- remote pulse generators with modular function (RPWM)

Options Tab for OTB Modules

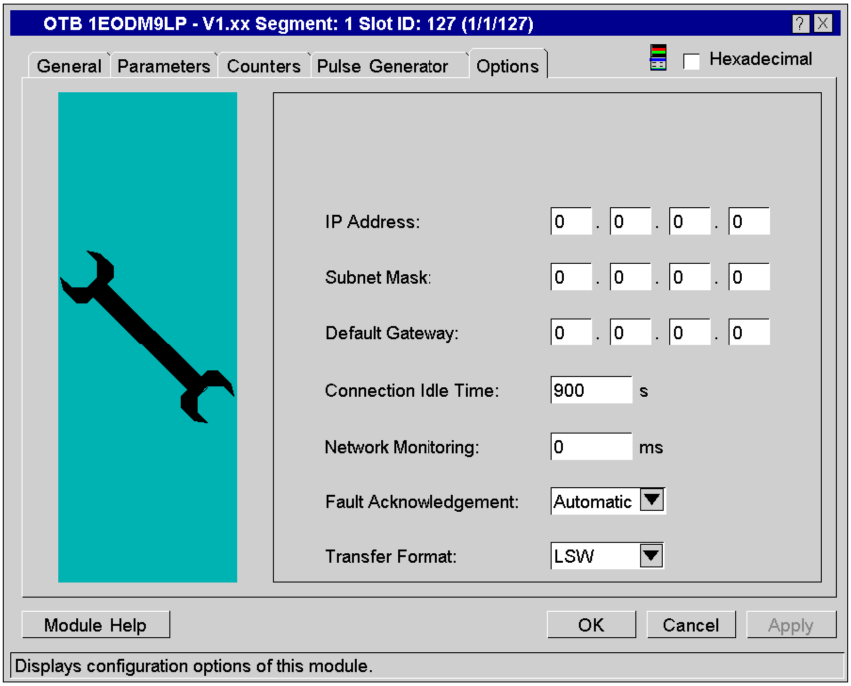
Introduction

The **Options** tab lists global configuration parameters for accessing NIM registers and connection settings. The name of the module and its exact location on the Island bus are displayed in the title bar.

NOTE: The **Options** tab differs for OTB Ethernet and OTB Modbus NIMs. It is not available for OTB CANopen NIMs.

Options Tab Figure Example

The following figure shows the **Options** tab information:



Parameters Description

This table contains a description of the global configuration parameters available for OTB Ethernet and OTB Modbus NIMs:

Parameter	Description
IP Address	connection parameters of the physical communication port only available for OTB Ethernet NIMs
Subnet Mask	
Default Gateway	
Connection Idle Time	configured idle time until connection is closed only available for OTB Ethernet NIMs
Network Monitoring	intervall for network monitoring
Fault Acknowledgement	mode for detected error acknowledgement (automatic or manual)
Transfer Format	LSW/MSW order for 32 bit values

Section 3.4
Module Editor for FTM and FTB Modules

Introduction

This section contains a detailed description of the Module Editor for FTM and FTB modules. All available tabs are explained.

What Is in This Section?

This section contains the following topics:

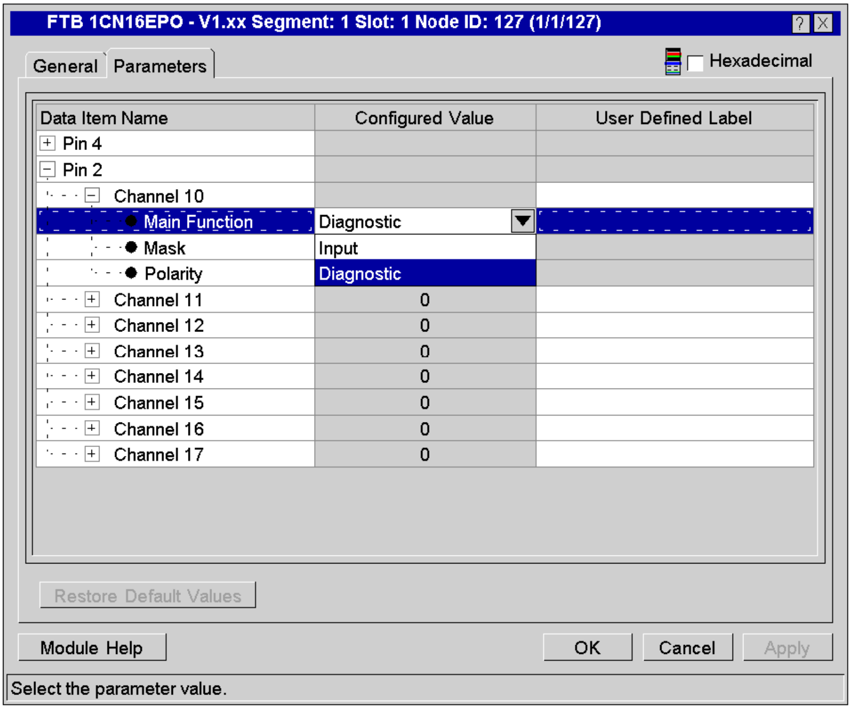
Parameters Tab for FTM and FTB Digital Modules

Introduction

Many of the digital I/O modules of the FTM product family and of the NIMs of the FTB product family are configurable with respect to the function associated with a physical connector pin. That means that you can configure if a pin shall be used for input data, output data or diagnostics. This configuration is performed in the **Parameters** tab, where all I/O data objects of the module concerned are listed including bit level information. The name of the module and its exact location on the Island bus are displayed in the title bar.

Parameters Tab Figure Example

The following figure shows the **Parameters** tab information of an FTB NIM:



Tab Structure

The **Parameters** tab provided for FTM and FTB digital modules is organized and structured in the same way the **Parameters** tab of the OTB Module Editor is. That means that the configuration parameters are displayed in a hierarchical tree view comprising master parameters and slave parameters. Further, they are assigned to each single data item, that is the data item is displayed as superordinated. For details, refer to *Parameters Tab for OTB Modules,* *page 102*.

Parameters Description

This table contains a description of the digital I/O parameters and their availability for the different FTM and FTB modules:

Parameter	Description
Data Item Name	name of the data item (grouped according to their physical assignment to either pin 4 or 2)
Main Function	selecting the data category for the channels of configurable modules Depending on the module type, you can choose between ● input and output ● input and status ● input, output and diagnostic For some modules, the function is preset and not editable.
Mask	enabling or disabling the ● input filtering (for preset or configured inputs) ● output propagation (for preset or configured outputs)
Polarity	selecting between normal or reverse polarity
Fail State	setting the value for the output fallback state (only available if the main function is set to **Output**)
User Defined Label	user-defined label of the parameter

Parameters Tab for FTM Analog Input Modules

Introduction

For customizing the input data objects of FTM analog inputs, use the **Parameters** tab. The name of the module and its exact location on the Island bus are displayed in the title bar.

Tab Structure

The **Parameters** tab provided for FTM analog input modules is organized and structured in the same way the **Parameters** tab of the OTB Module Editor is. That means that the configuration parameters are displayed in a hierarchical tree view comprising master parameters and slave parameters. Further, they are assigned to each single data item, that is the data item is displayed as superordinated *(see page 102)*.

Parameters Description

This table contains a description of the parameters for analog input modules:

Parameter	Description
Data Item Name	name of the data item
Range	selecting the range mode for the input
Filter	selecting the filter for pre-filtering the analog signal
Diagnostic	enabling or disabling the diagnostics
Delta Value	defining a deadband within which any modification of the input signal is not indicated
Min / Max	minimum / maximum value for the custom range
User Defined Label	label of the parameter

Parameter Dependencies

The availability of parameters depends on the state of the range parameter:

If you select ...	Then you can edit these parameters:
the mode **Disable**	User Defined Label
a mode other than **Disable**	• Filter • Diagnostic • Delta Value • User Defined Label **Note:** The minimum and maximum values for the custom range are not editable.

Parameters Tab for FTM Analog Output Modules

Introduction

For customizing the output data objects of FTM analog outputs, use the **Parameters** tab. The name of the module and its exact location on the Island bus are displayed in the title bar.

Tab Structure

The **Parameters** tab provided for FTM analog output modules is organized and structured in the same way the **Parameters** tab of the OTB Module Editor is. That means that the configuration parameters are displayed in a hierarchical tree view comprising master parameters and slave parameters. Further, they are assigned to each single data item, that is the data item is displayed as superordinated. For details, refer to *Parameters Tab for OTB Modules,* *page 102*.

Parameters Description

This table contains a description of the parameters for analog output modules:

Parameter	Description
Data Item Name	name of the data item
Range	selecting the range mode for the input
Diagnostic	enabling or disabling the diagnostics
Error Mode	setting the diagnostic mode (hold last value or predefined)
Error Value	setting the diagnostic value for the predefined error mode
User Defined Label	label of the parameter

Parameter Dependencies

The availability of parameters depends on the state of the range parameter as follows:

If you select ...	Then you can edit these parameters:
the mode **Disable**	User Defined Label
a mode other than **Disable**	• Diagnostic • Error Mode • Error Value if you have selected **Predefined** for the diagnostic mode • User Defined Label

Section 3.5
User Defined Label Editor

Introduction

This section provides an overview of the User Defined Label Editor.

What Is in This Section?

This section contains the following topics:

User Defined Label Editor Introduction

Introduction

The **User Defined Label Editor** is a single location where you can associate labels with the I/O image data items for all the I/O modules present in the current Island. It is enabled only for STB Islands and when the Island has atleast 1 I/O module.

The **User Defined Label Editor** also supports importing and exporting of labels in the *.csv* format.

The user defined labels that are modified in the **User Defined Label Editor** are also displayed in the **IO Image** tab of the Module Editor.

NOTE: You can modify the user defined labels only when the Island is offline and unlocked. But you can launch the **User Defined Label Editor** when the Island is online or offline, either in locked or unlocked mode.

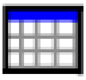
Overview of the Functions

It is possible to:

- view all the I/O image data items of the current Island in a single location
- assign labels to all the I/O image data items of the current Island when the Island is offline and unlocked
- modify the user defined labels *(see page 120)* when the Island is offline and unlocked
- cut, copy, or paste multiple labels when the Island is offline and unlocked
- import labels *(see page 121)* as *.csv* file when the Island is offline and unlocked
- export labels *(see page 122)* to a *.csv* file when the Island is online or offline, either in locked or unlocked mode
- navigate between labels using the UP ARROW key and the DOWN ARROW key

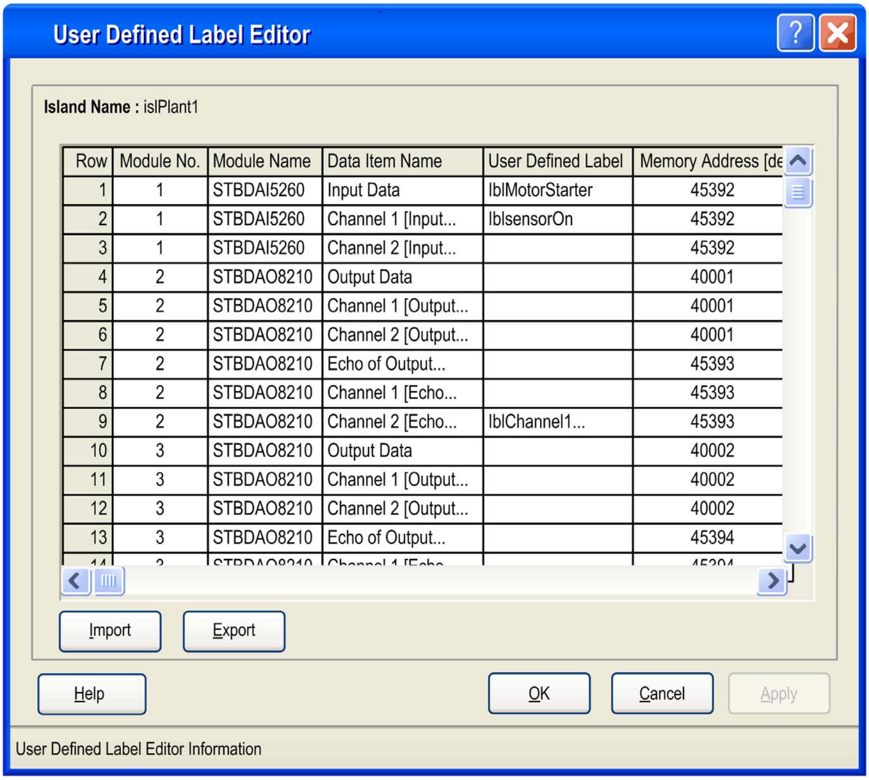
Accessing the User Defined Label Editor

To invoke the **User Defined Label Editor**, choose 1 of the following methods:

If you want to open it...	Then...
from the **Island** menu	select **Label Editor**. 
from the **Island** toolbar	click the following icon: 

Example for the User Defined Label Editor

The figure below shows an example of the **User Defined Label Editor**:



Row	Module No.	Module Name	Data Item Name	User Defined Label	Memory Address [de
1	1	STBDAI5260	Input Data	lblMotorStarter	45392
2	1	STBDAI5260	Channel 1 [Input...	lblsensorOn	45392
3	1	STBDAI5260	Channel 2 [Input...		45392
4	2	STBDAO8210	Output Data		40001
5	2	STBDAO8210	Channel 1 [Output...		40001
6	2	STBDAO8210	Channel 2 [Output...		40001
7	2	STBDAO8210	Echo of Output...		45393
8	2	STBDAO8210	Channel 1 [Echo...		45393
9	2	STBDAO8210	Channel 2 [Echo...	lblChannel1...	45393
10	3	STBDAO8210	Output Data		40002
11	3	STBDAO8210	Channel 1 [Output...		40002
12	3	STBDAO8210	Channel 2 [Output...		40002
13	3	STBDAO8210	Echo of Output...		45394

The **Island Name** field displays the current Island name.

The **User Defined Label Editor** displays the following information of all the I/O modules present in the current Island:

1. **Row**
2. **Module No.**
3. **Module Name**
4. **Data Item Name**
5. **User Defined Label**
6. **Memory Address (dec)**

The I/O modules appear in the same order as they are placed in the current Island.

Row

The identification of each row with a sequential number is listed in the **Row** column.

NOTE: This column is **read-only**.

Module No. Column

The addressable numbers of all the I/O modules in the current Island are listed in the **Module No.** column.

NOTE: This column is **read-only**.

Module Name Column

The names of all the I/O modules in the current Island are listed in the **Module Name** column.

NOTE: This column is **read-only**.

Data Item Name Column

The operational parameters that characterize the I/O modules in the current Island are listed in the **Data Item Name** column.

NOTE: This column is **read-only**.

User Defined Label Column

You can write a custom label or note for any of the I/O image data items listed on the **User Defined Label Editor**. This feature allows you to pre-symbolize important memory locations in the Island before the application is written.

In the **User Defined Label** column, click in the appropriate cell to enter the label text. Each label can be only up to 24 characters long.

Memory Address (dec) Column

The displayed memory address represents the Modbus register.

NOTE: Values in the **Memory Address** column are displayed only for parent nodes. This column is **read-only**.

User Defined Label Editor Help

You can get information about the **User Defined Label Editor** by clicking the **Help** button.

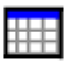
Modifying Labels

Description

The **User Defined Label Editor** allows you to modify the user defined labels of all the I/O image data items in the current Island. This feature is enabled only for STB Islands. The user defined labels can be modified only when the Island is offline and unlocked.

Modifying User Defined Labels

To modify labels in the **User Defined Label Editor**, the Island should be offline and unlocked, then:

Step	Action
1	Select **Label Editor** from the **Island** menu, or click the following icon on the **Island** toolbar:  **Result**: The **User Defined Label Editor** is displayed.
2	In the **User Defined Label** column, click the user defined label you want to modify, or select multiple/all labels using either of the 2 methods: ● drag the mouse pointer on the required cells, to select multiple labels or ● right-click and choose **Select All**, to select all labels in the column
3	Type a new label, or right-click and select the options **Cut**, **Copy**, **Paste**, or **Delete**.
4	Press ENTER, or select another label to modify and continue until all the labels you wanted to modify are complete.
5	● Click **Apply**, to apply the modifications. or ● Click **OK**, to save and close the **User Defined Label Editor**. **NOTE:** The label modifications are neither applied nor saved, if duplicate user defined labels exist.

Overview of the User Defined Labels

The user defined labels should not be duplicates and they have to be compliant to the IEC61131 rules:

- Only alphanumeric and underscore characters can be used.
- The first character has to be an alphabetic character.
- Blanks and non-ASCII characters are not allowed.
- The overall length of the label should not exceed 24 characters.

Importing Labels

Description

The **User Defined Label Editor** supports importing labels as *.csv* file. The import option is enabled only when the Island is offline and unlocked.

Importing Labels

To import labels from the **User Defined Label Editor**:

Step	Action
1	Select **Label Editor** from the **Island** menu or click the Label Editor icon in the Island toolbar to open the **User Defined Label Editor**.
2	Click **Import**. **Result:** The **Import** dialog box is displayed.
3	Browse the *.csv* file to import and click **Open**. **Result:** The selected file is imported and the status **File imported successfully** is displayed.

NOTE:
The import action is cancelled if the selected file:
- does not match with the expected *.csv* format
- modules name and I/O image data items does not match with the current Island
- has duplicate user defined labels
- has labels that are not compliant to the IEC61131 rules

Exporting Labels

Description

The **User Defined Label Editor** supports exporting labels to a *.csv* file. The export option is enabled when the Island is online or offline, either in locked or unlocked mode.

Exporting Labels

To export labels from the **User Defined Label Editor**:

Step	Action
1	Select **Label Editor** from the **Island** menu or click the User Defined Label Editor icon in the Island toolbar to open the **User Defined Label Editor**.
2	Click **Export**. **Result:** The **Export** dialog box is displayed with the default file name and *.csv* extension. The default file name is the current Island name followed by *_labels*. **NOTE:** The export action is cancelled if duplicate labels exist.
3	Click **Save**. **Result:** The user defined label data of the current Island is saved and the status **File exported successfully** is displayed.

Section 3.6
Run-Time Parameters

Introduction

This section provides an overview of the run-time parameters of the STB network interface modules.

What Is in This Section?

This section contains the following topics:

Run-Time Parameters

Introduction

For STB modules, the Advantys Configuration Software provides the RTP (run-time parameters) feature. It can be used for monitoring and modifying selected I/O parameters and Island bus status registers of the NIM while the Island is running. This feature is available only in standard STB NIMs with firmware version 2.0 or later.

RTP has to be configured using the Advantys Configuration Software before it can be used. RTP is not configured by default. Configure RTP by selecting **Configure run-time Parameters** in the **Options** tab of the NIM Module Editor. This allocates the necessary registers within the NIM's data process image to support this feature.

Request and Response Blocks

Once configured, use the RTP feature by writing up to 5 reserved words in the NIM's output data process image (the RTP request block) and by reading the value of 4 reserved words in the NIM's input data process image (the RTP response block). The Advantys Configuration Software displays both blocks of reserved RTP words in the Island's **I/O Image Overview** dialog box, both in the **Modbus Image** tab and (for NIMs with a separate fieldbus image) in the **Fieldbus Image** tab. In each tab, the blocks of reserved RTP words appear after the block of process I/O data and before the block of HMI data (if any).

NOTE: The Modbus address values of the RTP request and response blocks are the same in all standard NIMs. The fieldbus address values of the RTP request and response blocks depend upon the network type. Use the **Fieldbus Image** tab of the **I/O Image Overview** dialog box to obtain the location of the RTP registers. For Modbus Plus and Ethernet networks, use the Modbus register numbers.

Exceptions

Any parameter you modify using the RTP feature does not retain its modified value if one of the following events occurs:

- Power is cycled to the NIM.
- A **Reset** command is issued to the NIM using the Advantys Configuration Software.
- A **Store to SIM Card** command is issued using the Advantys Configuration Software.
- The module whose parameter has been modified is hot-swapped.
  If a module is hot-swapped, as indicated by the HOT_SWAP indicator bit, you can use the RTP feature to detect which module has been hot-swapped and to restore the parameters to their previous values.

Test Mode

When the NIM is operating in test mode, the NIM's output data process image (including the RTP request block) can be controlled either by the Advantys Configuration Software or by an HMI (depending upon the test mode configured). Standard Modbus commands can be used to access the RTP words. If the NIM is in test mode, the fieldbus master cannot write to the RTP request block in the NIM's output data process image.

RTP Request Block Words Definitions

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Write all bytes in the RTP request block before you set the `toggle+CMD` and `toggle+length` bytes to the same new value.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

The following table lists RTP request block words:

Modbus Address	Upper Byte	Lower Byte	Data Type	Attribute
45130	sub-index	`toggle + length`	unsigned 16	RW
45131	index (high data byte)	index (low data byte)	unsigned 16	RW
45132	data byte 2	data byte 1 (LSB)	unsigned 16	RW
45133	data byte 4 (MSB)	data byte 3	unsigned 16	RW
45134	`toggle + CMD`	Node ID	unsigned 16	RW
NOTE: The RTP request block is also presented in the manufacturer specific area of the CANopen fieldbus as an object with a dedicated index of 0x4101 and sub-index 1 to 5 (data type = unsigned 16, attribute = RW).				

The NIM performs range checking on the above bytes as follows:

- index (high / low byte): 0x2000 to 0xFFFF for write; 0x1000 to 0xFFFF for read
- `toggle + length`: length = 1 to 4 bytes; the most significant bit contains the toggle bit
- `toggle + CMD`: CMD = 1 to 0x0A (see the table *Valid Commands*, below); most significant bit contains toggle bit
- Node ID: 1 to 32 and 127 (the NIM itself)

The `toggle+CMD` and `toggle+length` bytes are at either end of the RTP request register block. The NIM processes the RTP request when the same value is set in the respective toggle bits of these two bytes. The NIM processes the same RTP block again only when both values have changed to a new identical value. We recommend that you configure new matching values for the two toggle bytes (`toggle+CMD` and `toggle+length`) only after you have constructed the RTP request between them.

RTP Response Block Words Definitions

The following list shows RTP response block words:

Modbus Address	Upper Byte	Lower Byte	Data Type	Attribute
45303	status (the most significant bit is used to indicate whether RTP service is enabled: MSB=1 means enabled)	`toggle + CMD echo`	unsigned 16	RO
45304	data byte 2	data byte 1 (LSB)	unsigned 16	RO
45305	data byte 4 (MSB)	data byte 3	unsigned 16	RO
45306	-	`toggle + CMD echo`	unsigned 16	RO
NOTE: The RTP response block is also presented in the manufacturer specific area of the CANopen fieldbus as an object with a dedicated index of 0x4100 and sub-index 1 to 4 (data type = unsigned 16, attribute = RO).				

The `toggle+CMD echo` bytes are located at the end of the register range to let you validate the consistency of the data wrapped within these bytes (in case RTP response block words are not updated in a single scan). The NIM updates the status byte and the 4 data bytes (if applicable) before updating the `toggle+CMD echo` bytes in Modbus register 45303 and 45306 to equal the value of the `toggle+CMD` byte of the corresponding RTP request. First check that both `toggle+CMD` bytes match the `toggle+CMD` byte in the RTP request block before making use of the data inside the RTP response block.

Valid RTP Commands

The following list shows valid commands (CMDs):

Command (CMD)	Code (Except the msb)	Valid Node IDs	Allowed State of the Addressed Node	Data Bytes
Enable RTP (Only After RTP Has Been Configured Using the Advantys Configuration Software)	0x08	127	N/A	-
Disable RTP	0x09	127	N/A	-
Reset Hot-Swap Bit	0x0A	1-32	N/A	-

Command (CMD)	Code (Except the msb)	Valid Node IDs	Allowed State of the Addressed Node	Data Bytes
Read Parameter	0x01	1-32, 127	pre-operational operational	data bytes in response, length to be given
Write Parameter	0x02	1-32	operational	data bytes in request, length to be given

The most significant bit of an RTP request block's `toggle+CMD` byte is the toggle bit. A new command is identified when the value of this bit changes and matches the value of the toggle bit in the `toggle+length` byte.

A new RTP request is processed only if the preceding RTP request has finished. Overlapping RTP requests are not allowed. A new RTP request made before the completion of a preceding request is ignored.

To determine when an RTP command has been processed and its response is complete, check the values of the `toggle+CMD echo` bytes in the RTP response block. Continue to check both `toggle+CMD` bytes in the RTP response block until they match the RTP request block's `toggle+CMD` byte. Once they match, the contents of the RTP response block is valid.

Valid RTP Status Messages

The following list shows valid status messages:

Status Byte	Code	Comment
Success	0x00 or 0x80	0x00 for successful completion of a Disable RTP command
Command not Processed due to Disabled RTP	0x01	-
Illegal CMD	0x82	-
Illegal Data Length	0x83	-
Illegal Node ID	0x84	-
Illegal Node State	0x85	Access is denied because a node is absent or not started.
Illegal Index	0x86	-
RTP Response Has More Than 4 Bytes	0x87	-
No Communication Possible on the Island Bus	0x88	-
Illegal Write to Node 127	0x89	-

Status Byte	Code	Comment
SDO Aborted	0x90	If there is a detected error in SDO protocol, the data bytes in the response contain the SDO abort code according to DS301.
General Exception Response	0xFF	This is a status event of a type other than those specified above.

The most significant bit of the status byte in the RTP response block indicates whether RTP is enabled (1) or disabled (0).

Using RTP

Introduction

Typical use of RTP includes the following:
- configuring RTP
- enabling RTP
- executing an RTP read request
- executing an RTP write request

Configuring RTP

Configure RTP as follows:

Step	Action
1	In the Advantys Configuration Software, confirm that the NIM is using firmware version 2.0 or later.
2	On the physical Advantys STB Island, confirm that the NIM on the rail is using firmware version 2.0 or later.
3	In the Advantys Configuration Software, in the **Options** tab of the NIM's Module Editor, select the **Configure run-time Parameters** check box, and then click **OK**. **Result:** 5 words (Modbus Registers 45130 to 45134) are added to the NIM's output data process image, and 4 words (Modbus Registers 45303 to 45306) are added to the NIM's input data process image
4	After all Island configuration settings have been completed, download the configuration into the NIM.

Enabling RTP

Before you can execute RTP commands, first enable RTP using the Advantys Configuration Software.

Enable RTP as follows:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Verify that any prior RTP command has been completed.
Note: This step is necessary only if RTP has been used since the NIM was last reset or powered-up.</td></tr>
<tr><td>2</td><td>Check a `toggle+CMD echo` byte in the RTP response block to obtain the previous value of the toggle bit.</td></tr>
<tr><td>3</td><td>Write the following values to the RTP request block:
<table>
<tr><th>Address</th><th>Upper Byte</th><th>Lower Byte</th></tr>
<tr><td>45130</td><td>sub-index: 0</td><td>`toggle + length`:
toggle: If the value of the toggle retrieved in step 2 above is 0, use 0x80; otherwise, use 0x00.</td></tr>
<tr><td>45131</td><td>index (high data byte): 0</td><td>index (low byte): 0</td></tr>
<tr><td>45132</td><td>data byte 2: 0</td><td>data byte 1: 0</td></tr>
<tr><td>45133</td><td>data byte 4: 0</td><td>data byte 3: 0</td></tr>
<tr><td>45134</td><td>`toggle + CMD`:
toggle: If the value of the toggle retrieved in step 2 above is 0, use 0x88; otherwise, use 0x08.</td><td>Node ID: 127</td></tr>
</table>
Result: Upon completion of this step, RTP is enabled.</td></tr>
<tr><td>4</td><td>Continue to check both `toggle+CMD echo` bytes in the RTP response block until they match the RTP request block’s `toggle+CMD` by byte.</td></tr>
<tr><td>5</td><td>Check the status byte of the RTP response block to confirm that RTP is enabled.</td></tr>
</table>

Executing an RTP Read Request

Obtain the following information for the parameter you wish to read:

- Node ID
  (**Note:** The Node ID value is displayed at the bottom of a module in the Advantys Configuration Software Island Editor, on the right.)
- index
- sub-index
- length

(Refer to the *Hardware Reference Manual* for these values.)

Execute an RTP read request as follows:

<table>
<tr><th>Step</th><th colspan="3">Action</th></tr>
<tr><td>1</td><td colspan="3">Verify that any prior RTP command has completed.</td></tr>
<tr><td>2</td><td colspan="3">Check a <code>toggle+CMD echo</code> byte in the RTP response block, to obtain the previous value of the toggle bit.</td></tr>
<tr><td rowspan="8">3</td><td colspan="3">Issue an RTP read request by setting up the 5 words in the RTP request block.</td></tr>
<tr><td>Address</td><td>Upper Byte</td><td>Lower Byte</td></tr>
<tr><td>45130</td><td>sub-index: 0- 255</td><td><code>toggle + length</code>:
toggle: the inverse of the value retrieved in step 2 above
length: 1-4 bytes</td></tr>
<tr><td>45131</td><td>index (high data byte):
0x10 to 0xFF</td><td>index (low byte):
0x00 to 0xFF</td></tr>
<tr><td>45132</td><td>data byte 2: 0</td><td>data byte 1: 0</td></tr>
<tr><td>45133</td><td>data byte 4: 0</td><td>data byte 3: 0</td></tr>
<tr><td>45134</td><td><code>toggle + CMD</code>:
If toggle was 0, use 0x81; if toggle was 1, use 0x01.</td><td>Node ID:
1-32 and 127</td></tr>
<tr><td colspan="3">Note: Although these 5 words can be modified in any order, be aware that the NIM will execute the RTP command when the toggle bits in words 45130 and 45134 have been changed from their previous values and set equal to each other. The last word you modify should be a word 45130 or 45134 to ensure that the desired RTP request is executed.</td></tr>
<tr><td>4</td><td colspan="3">Continue to check both <code>toggle+CMD echo</code> bytes in the RTP response block until they match the RTP request block’s <code>toggle+CMD</code> byte.</td></tr>
<tr><td>5</td><td colspan="3">Check the status byte of the RTP response block.
Result: If the status byte = 0x80, the requested data is available in the data registers of the RTP response block. Refer to the valid status messages table above in case of detected error.</td></tr>
</table>

Executing an RTP Write Request

Obtain the following information for the parameter you wish to write:

- Node ID
  (**Note**: The Node ID value is displayed in the Advantys Configuration Software Island Editor at the right bottom of a module on the right.)
- index
- sub-index
- length

(Refer to the *Hardware Reference Manual* for these values.)

Execute an RTP write request as follows:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Verify that any prior RTP command has completed.</td></tr>
<tr><td>2</td><td>Check a <code>toggle+CMD echo</code> byte in the RTP response block, to obtain the previous value of the toggle bit.</td></tr>
<tr><td>3</td><td>Issue an RTP write request by setting up the 5 words in the RTP request block:
<table>
<tr><td>Address</td><td>Upper Byte</td><td>Lower Byte</td></tr>
<tr><td>45130</td><td>sub-index: 0- 255</td><td><code>toggle + length</code>:
toggle: the inverse of the value retrieved in step 2 above
length: 1-4 bytes</td></tr>
<tr><td>45131</td><td>index (high data byte):
0x20 to 0xFF</td><td>index (low byte):
0x00 to 0xFF</td></tr>
<tr><td>45132</td><td>data byte 2:
0x00 to 0xFF</td><td>data byte 1
0x00 to 0xFF</td></tr>
<tr><td>45133</td><td>data byte 4
0x00 to 0xFF</td><td>data byte 3
0x00 to 0xFF</td></tr>
<tr><td>45134</td><td><code>toggle + CMD</code>:
toggle: If the value of the toggle retrieved in step 2 above is 0, use 0x82; otherwise, use 0x02.</td><td>Node ID:
1-32</td></tr>
</table>
Note: Although these 5 words can be modified in any order, be aware that the NIM will execute the RTP command when the toggle bits in words 45130 and 45134 have been changed from their previous values and set equal to each other. The last word you modify should be a word 45130 or 45134 to ensure that the desired RTP request is executed.</td></tr>
<tr><td>4</td><td>Continue to check both <code>toggle+CMD echo</code> bytes in the RTP response block until they match the RTP request block’s <code>toggle+CMD echo</code> byte.</td></tr>
<tr><td>5</td><td>Check the status byte of the RTP response block.
Result: If the status byte = 0x80, the write request has been successfully executed. Refer to the valid status messages table above in case of detected error.</td></tr>
</table>

Hot-Swap Diagnostics

Introduction

Standard STB NIMs with firmware version 2.0 or later implement a HOT_SWAP indicator bit as part of the NIM device status, which is included in the Island's fieldbus image. This indicator bit will be set if 1 or more modules have been hot-swapped since the NIM was last reset or powered-up.

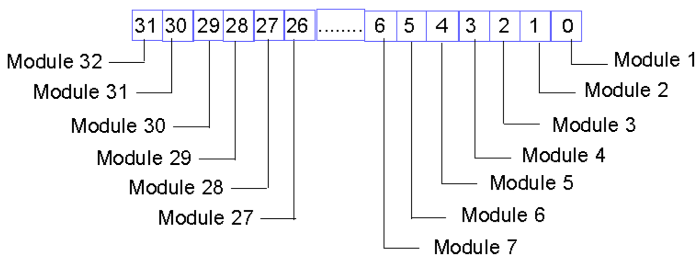
You can use the RTP feature to determine which Island module has been hot-swapped. Standard NIMs with firmware version 2.0 or later implement a read-only object that contains a bit for each Island module indicating that module's hot-swap status (1 = hot-swapped, 0 = reset). When all bits in the hot-swap object are reset, the HOT_SWAP indicator bit is also reset.

The content of this object is accessible using RTP to node 127 (the NIM).

Hot-Swap Object

The hot-swap object is located at index 0x4106, sub-index 1 and is of data type unsigned32 (4 bytes). Each bit in this object corresponds to a module, as follows:

Hot-Swap Object



Obtaining the Hot-Swap Status of All I/O Modules

Proceed as follows to obtain the hot-swap status of all I/O modules:

<table>
<tr><th>Step</th><th colspan="3">Action</th></tr>
<tr><td>1</td><td colspan="3">Verify that any prior RTP command has been completed.</td></tr>
<tr><td>2</td><td colspan="3">Check a `toggle+CMD echo` byte in the RTP response block, to obtain the previous value of the toggle bit.</td></tr>
<tr><td rowspan="8">3</td><td colspan="3">Issue an RTP read request by setting up the 5 words in the RTP request block.</td></tr>
<tr><td>Address</td><td>Upper Byte</td><td>Lower Byte</td></tr>
<tr><td>45130</td><td>sub-index: 1</td><td>`toggle + length`:
toggle: If the value of the toggle retrieved in step 2 above is 0, use 0x84; otherwise, use 0x04.</td></tr>
<tr><td>45131</td><td>index (high data byte):
0x41</td><td>index (low byte):
0x06</td></tr>
<tr><td>45132</td><td>data byte 2: 0</td><td>data byte1: 0</td></tr>
<tr><td>45133</td><td>data byte 4: 0</td><td>data byte 3: 0</td></tr>
<tr><td>45134</td><td>`toggle + CMD`:
toggle: If the value of the toggle retrieved in step 2 above is 0, use 0x81; otherwise, use 0x01.</td><td>Node ID: 127</td></tr>
<tr><td colspan="3">**Note:** Although these 5 words can be modified in any order, be aware that the NIM will execute the RTP command when the toggle bits in words 45130 and 45134 have been changed from their previous values and set equal to each other. The last word you modify has to be word 45130 or 45134 to ensure that the desired RTP request is executed.</td></tr>
<tr><td>4</td><td colspan="3">Continue to check both `toggle+CMD echo` bytes in the RTP response block until they match the RTP request block's `toggle+CMD` byte.</td></tr>
<tr><td>5</td><td colspan="3">Check the status byte of the RTP response block.
Result: If the status byte = 0x80, the hot-swap status of the modules is available in the data registers of the RTP response block. Refer to the valid status messages table above in case of detected error.</td></tr>
</table>

Obtaining the Hot-Swap Status of Individual I/O Modules

All bits in the hot-swap object are reset whenever power is cycled to the NIM, a `Reset` or a `Store to SIM Card` command is issued, or a new configuration is downloaded into the NIM. You can also use RTP command 0x0A to reset the hot-swap status of individual modules.

Proceed as follows to reset the hot-swap status of an I/O module:

<table>
<tr><th>Step</th><th colspan="3">Action</th></tr>
<tr><td>1</td><td colspan="3">Verify that any prior RTP command has completed.</td></tr>
<tr><td>2</td><td colspan="3">Check a <code>toggle + CMD echo</code> byte in the RTP response block, to obtain the previous value of the toggle bit.</td></tr>
<tr><td rowspan="8">3</td><td colspan="3">Issue an RTP reset hot-swap bit request by setting up the 5 words in the RTP request block:</td></tr>
<tr><td>Address</td><td>Upper Byte</td><td>Lower Byte</td></tr>
<tr><td>45130</td><td>sub-index: 0</td><td><code>toggle + length</code>:
toggle: If the value of the toggle retrieved in step 2 above is 0, use 0x80; otherwise, use 0x00.</td></tr>
<tr><td>45131</td><td>index (high data byte): 0</td><td>index (low byte): 0</td></tr>
<tr><td>45132</td><td>data byte 2: 0</td><td>data byte1: 0</td></tr>
<tr><td>45133</td><td>data byte 4: 0</td><td>data byte 3: 0</td></tr>
<tr><td>45134</td><td><code>toggle + CMD</code>:
toggle: If the value of the toggle retrieved in step 2 above is 0, use 0x8A; otherwise, use 0x0A.</td><td>Node ID: 1 - 32</td></tr>
<tr><td colspan="3">Note: Although these 5 words can be modified in any order, be aware that the NIM will execute the RTP command when the toggle bits in words 45130 and 45134 have been changed from their previous values and set equal to each other. The last word you modify has to be word 45130 or 45134 to ensure that the desired RTP request is executed.</td></tr>
<tr><td>4</td><td colspan="3">Continue to check both <code>toggle+CMD echo</code> bytes in the RTP response block until they match the RTP request block’s <code>toggle + CMD</code> byte.</td></tr>
<tr><td>5</td><td colspan="3">Check the status byte of the RTP response block.
Result: If the status byte = 0x80, the reset hot-swap bit request has been successfully executed. Refer to the valid status messages table above in case of detected error.</td></tr>
</table>

Section 3.7
Offline Features

Introduction

This section provides an overview of the offline features of the Advantys Configuration Software.

What Is in This Section?

This section contains the following topics:

Offline Features Introduction

Introduction

By default, the logical Island in the configuration software is not connected to the physical Island, and the Island is set to offline mode. If the Island is in offline mode, you may perform the following:
- adding modules to the Island configuration
- deleting modules from the Island configuration
- modifying the operating parameters of the I/O modules using the Module Editor
- adding annotation text in the Island Editor
- configuring reflex actions
- building the Island configuration
- exporting symbols, variables and communication parameters

NOTE: Which functions can be performed depends on both the product family and the NIM type.

Planning an Island

Introduction

An Island comprises different modules with at least 1 segment called the primary segment. Any Island consists of 1 NIM (network interface module); each segment has to contain 1 auto-addressable module (NIM or I/O). All Advantys STB and OTB modules of an Island are mounted on a DIN rail. Preferred modules and enhanced CANopen devices are positioned off the rail and connected by extension cables. FTM and FTB modules are surface mounted.

Preparation

Before installing modules, you need to define a clear scheme including:
- number and type(s) of I/O modules on the Island
- power requirements
- resulting order in which they are mounted

It is essential that you establish and follow a clear plan. For STB Islands, the backplane is composed of a series of interconnected, module-specific base units. The structure of the Island backplane is therefore determined by the type(s) and sequence of modules that will reside in it. These decisions has to be made anticipatively in order to build an appropriate backplane.

OTB I/O modules are connected in a row to the NIM. Determine their order in advance, before the whole Island is mounted on a DIN rail. FTM modules are mounted on a stretch. Nevertheless, a certain order has to be clear in advance because not all modules are extensible. FTB Islands consist of only 1 NIM.

Island Beginning

The first module installed in each Island is the NIM. Each Island can contain only 1 NIM. It has to be installed in the leftmost slot of the primary segment.

Depending on the product family, the following various NIM models are available, each supporting a different open fieldbus protocol:
- CANopen
- Profibus DP
- Interbus
- DeviceNet
- Fipio
- Ethernet
- Ethernet/IP
- Modbus
- Modbus Plus

Check that the NIM you select is appropriate to your fieldbus environment. Neither FTB and FTM nor STB and OTB NIMs are mounted in its own distinct base. All NIMs already include all necessary Island bus contacts and, in case of STB and OTB NIMs, the DIN rail connectors.

I/O Modules

Key factors in determining the Island layout are the number and type(s) of I/O modules required. When you have made those 2 decisions, it becomes easy to determine the Island's external power and power distribution requirements. Depending on the product family, an Island may be able to support up to 32 addressable I/O modules. These I/O modules may be combinations of digital, relay, analog and expert Advantys modules, preferred modules and/or enhanced CANopen devices.

Segments

The Advantys I/O modules has to be installed in structures called segments. A segment consists of a series of interconnected I/O modules and/or power supplies, power distribution modules, termination or extension devices. In case of STB Islands, these interconnected modules need to be inserted in bases, which snap together on the DIN rail.

These interconnected bases form the backplane over which the Island conveys

- logic power,
- Island bus communications,
- module sensor and actuator field power.

OTB Islands are interconnected and then mounted on DIN rails. Thus, power and communications are conveyed. FTM and FTB Islands are surface mounted. Separate cables are used to provide their modules with power.

Each Island has to include at least the primary segment. Depending on the product family, up to 6 extension segments may be added. Each segment can support up to 1.2 A of current draw on the logic power signal. If more is needed in a segment, you can add an auxiliary power supply. You are allowed to install as few as 1 I/O module per extension segment, and you may use multiple segments as a way to position the I/O modules as close as possible to the field sensors and actuators they service. The maximum length of an STB Island, from the NIM to the last module, is 15 m (49.21 ft). This length includes all extension cables between device segments as well as the widths of the devices themselves. It also applies to the length required to support any preferred modules and enhanced CANopen devices. The maximum length of an OTB Island is 5 m (16.4 ft).

Power Requirements

For STB Islands, the Island backplane, sometimes referred to as Island bus, is designed to distribute field power to all its I/O modules over separate sensor buses (to the input modules) and actuator buses (to the output modules). There are modules for distributing field power, which are called power distribution modules (PDM). Analog I/O and relay output modules generally depend on 24 VDC PDMs for power distribution. Some digital modules depend on 24 VDC PDMs, while other digital modules depend on 115/230 VAC PDMs. The PDM has to be installed directly to the left of the I/O modules to which it distributes field power. If you intend to support both DC I/O modules and AC I/O modules in the same segment, you need to install different PDMs in the segment to support the different voltage groups. Please note that PDM and PDT (power distribution terminal) are synonymous; we recommend the term PDM.

As you plan the Island, it is important to remember that all I/O modules requiring 24 VDC has to be clustered together in a voltage group separately from any 115 or 230 VAC modules. Likewise, all I/O modules requiring 115 VAC has to be clustered together away from any 24 VDC modules and 230 VAC modules in the segment. For best noise immunity, install 115/230 VAC clusters closest to the NIM or BOS.

Island End

STB and OTB Islands are mounted on 1 or more DIN carrier rails. The DIN rail provides the functional ground point across the Island.

Each STB Island ends with a 120 Ω terminator resistor. If the STB Island ends at the last module on the primary segment (if the Island bus is not extended), then the segment has to end with the termination plate. If the STB Island bus is extended to either another segment of Advantys modules or to a preferred module, do not install the termination plate at the end of the primary segment.

Matching STB Modules to the Proper Bases

When the Island is correctly planned, select the proper sequence of module bases, taking into account the types of modules you want to include in the Island. The first module inserted in the primary segment of the Island is always the NIM. Module bases interlock with the NIM to form a backplane over which the Island conveys power and data.

NOTE: It is imperative to fully understand the complete Island layout before mounting module bases onto the DIN rail. Remember that module bases are interconnected from left to right along the DIN rail. If you insert an incorrect module base in any location on the bus, you will need to remove all the bases to its right before you are able to remove the incorrect module base.

The appropriate module base has to be inserted in each position along the DIN rail, in a sequence matching the succession of modules making up the Island.

There are 7 different types of STB bases, listed in the table below:

Module Types	Base Model	Width of Base
Any 24 VDC or 115/230 VAC Power Distribution Module (PDM)	STB XBA 2200	18.4 mm (0.73 in.)
Any 24 VDC Digital I/O Module Any Analog I/O Module	STB XBA 1000	13.9 mm (0.55 in.)
Any 115/230 VAC Digital I/O Module, Relay Module, CANopen Extension Module, and some Specialty Modules	STB XBA 2000	18.4 mm (0.73 in.)
End-of-Segment (EOS) Module	STB XBA 2400	18.4 mm (0.73 in.)
Beginning-of-Segment (BOS) Module	STB XBA 2300	18.4 mm (0.73 in.)
Some Specialty Modules	STB XBA 3000	28.1 mm (1.11 in.)
Auxiliary Power Supply	STB XBA 2100	18.4 mm (0.73 in.)

Constructing an Island

Introduction

Before proceeding with the Island construction, check that you have a clearly defined scheme to construct the Island. Once you have successfully completed the planning of the Island, physically constructing it should be an easy task.

Constructing the Island involves the following steps:
- using the Workspace to construct an Island
- constructing a valid segment
- extending the Island to multiple segments
- terminating the Island
- matching the Island modules to the proper bases

You can construct a new Island in a new or existing Workspace.

Creating New Islands in a New Workspace

Proceed as follows to create a new Island in a new Workspace:

Step	Action	Result
1	On the **File** menu, select **New Workspace**.	The **New Workspace** dialog box is displayed.
2	Type the name of the Workspace in the **Name:** field.	-
3	Type a path to the file in the **Location:** field or browse to the desired directory for the file using the button next to the field.	Once you have completed step 2 and 3, the name of the Workspace file and its file path will be displayed in the read-only **Name with Path:** field.
4	In the **Island File** section of the dialog box, name the Island in the **Name:** field.	The Island name and file path are displayed in the read-only **Name with Path:** field.
5	Click **OK**.	A new Island appears in the new Workspace. **Note:** The Island type is not defined until you have selected a NIM and thus a product family. A Workspace can include Islands of different product families.

Creating New Islands in an Existing Workspace

Proceed as follows to create a new Island in an existing Workspace:

Step	Action	Result
1	On the **File**, select **Open Workspace**.	The **Open Workspace** dialog box is displayed.
2	Select the Workspace file from the directory and click **Open**.	The Workspace and any pre-existing Islands appear in the Workspace Browser. **Note:** Any Workspace can support a maximum of 64 Islands.
3	On the **File** menu, select **Add New Island**.	The **New Island** dialog box is displayed.
4	In the **Name:** field, type the name of the Island you wish to create.	-
5	Click **OK**.	A new Island appears in the existing Workspace. **Note:** The Island type is not defined until you have selected a NIM and thus a product family. A Workspace can include Islands of different product families.

Appearance of the Workspace

The following windows are displayed if you open a new or existing Workspace:

- By default, the Workspace Browser is displayed on the left pane of the screen with the name you have assigned as the node. To rename the Island, select the node, click the Island node label and enter a new name, or press F2.
- By default, the Catalog Browser is displayed on the right pane of the screen with all available catalogs and modules.
- An Island Editor with a default DIN rail is displayed in the center pane of the screen.

Constructing a Logical Island in the Workspace

When you have opened a new Island in a new or existing Workspace, you need to physically construct the Island. To construct the Island, select successive modules from the Catalog Browser and insert them in the Island Editor.

To insert a module in the Island Editor you may

- double-click the module in the Catalog Browser,
- press ENTER on a selected module in the Catalog Browser or
- drag the module from the Catalog Browser.

NOTE: Once you have selected a NIM from 1 of the available catalogs, only the catalog concerned is still accessible.

Double-Click Method

To insert a module in the DIN rail of the Island Editor, double-click the desired module name in the Catalog Browser. The desired module will be added on the right side of the selected module in the Island Editor.

If you try to insert a module in an invalid location in the Island Editor, the application will display a notification.

Drag-and-Drop Method

To drag a module from the Catalog Browser to the Island Editor, click the desired module in the Catalog Browser and drag the selected module into position in the Island Editor.

As you drag the module, the following icon appears:

As you continue to drag the module to the DIN rail, the following icon appears:

If you attempt to drop a module onto an invalid location on the DIN rail, the system will display the following icon:

The dropped module will be added right of the position where you have dropped it.

NOTE: When you open an Island in an existing Workspace, an existing Island is always locked by default. You are not allowed to perform operations on the Island while it is locked. Unlock the Island before attempting any operation in the Island Editor.

Labeling Objects

Introduction

The Advantys Configuration Software allows you to assign meaningful names to any object in your configuration. This includes the Workspace, the Island and its segments as well as module parameters and I/O data objects.

The names you assign either replace the generic names completely (as for Workspaces, Islands and segments) or are appended to the generic names (as for module parameters and I/O data objects). The names for the selected data objects are displayed in the **I/O Image Overview**, **I/O Image Animation**, **Module Editor**, and **User Defined Label Editor**.

Naming of Workspaces, Islands and Segments

Proceed as follows to assign a name to a Workspace, an Island or the segments of an Island:

Step	Action
1	Open the Workspace Browser.
2	Select the object you want to give a name by clicking it.
3	Either click the object once more or press F2. **Note:** This is only possible if the Island is not locked or in offline mode.
4	Enter the name you want to assign to the object. **Note:** The names cannot exceed 24 characters.
5	Press ENTER.

Naming of Data Objects from the Module Editor

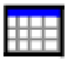
Proceed as follows to assign a name to a module parameter, an I/O status or I/O data element:

Step	Action
1	Select the module whose data elements you want to assign names to.
2	Open the Module Editor.
3	If the data object is ● a module parameter, select the **Parameters** tab. ● an I/O data object, select the **I/O Image** tab.
4	Double-click the **User Defined Label** cell of the object you want to assign a name to. **Note:** This is only possible if the Island is unlocked and in offline mode.
5	Type the name you want to assign to the object. **Note:** The names cannot exceed 24 characters.
6	Press ENTER.

Naming of Data Objects from the User Defined Label Editor

The **User Defined Label Editor** is a single location where you can associate labels with the I/O image data items for all the I/O modules in the current Island.

Proceed as follows to assign a user defined name to an I/O image data object:

Step	Action
1	Select **Label Editor** from the **Island** menu, or click the following icon on the Island toolbar to open the **User Defined Label Editor**: 
2	Click the **User Defined Label** cell of the I/O image data object you want to assign the name to. **Note:** This is only possible if the Island is offline and unlocked.
3	Type the name you want to assign to the I/O image data object. **Note:** The name cannot exceed 24 characters.
4	Press ENTER, or select another cell of the I/O image data object you want to assign the name. Continue until all the I/O image data objects you wanted to assign names are complete.
5	● Click **Apply**, to apply the assigned names. or ● Click **OK**, to save and close the **User Defined Label Editor**. **NOTE:** The assigned names are neither applied nor saved if you have typed duplicate names.

Extending and Terminating an Island

Introduction

You can extend the following Islands beyond the primary segment:
- Advantys FTM Islands to Advantys FTM I/O splitter boxes
- Advantys STB Islands to
  - Advantys STB I/O modules
  - preferred modules
  - enhanced CANopen devices

Depending on the module types to be added, the procedures for extending and terminating your Island bus vary.

Extending FTM Islands

An FTM Island bus is able to support up to 4 extension segments with 1 analog or up to 4 digital I/O splitter boxes each. Proceed as follows to extend an FTM Island:

Step	Action
1	Click the segment in which you want to insert the desired modules.
2	In the Catalog Browser, double-click the desired modules. **Note:** Place extensible I/Os first because compact I/Os cannot be extended.

Extending STB Islands to STB Modules

An STB Island bus is able to support up to 32 STB I/O modules in up to 6 extension segments in addition to the primary segment. Proceed as follows to extend an STB Island bus from an existing segment to a new extension segment:

Step	Action
1	Delete any termination plate or XBE you may have inserted at the end of the existing segment.
2	From the Catalog Browser, select the STB XBE 1000 EOS or STB XBE 1100 EOS extension module and place it at the end of the existing segment.
3	Double-click the STB XBE 1200 BOS module (if you have selected the XBE 1000 EOS module in step 2) or STB XBE 1300 BOS module (if you have selected the XBE 1100 EOS module in step 2) in the Catalog Browser. **Result:** The Advantys Configuration Software creates a new DIN rail in the Island Editor and places the BOS module in the leftmost position of the new extension segment. The Island Editor also shows the extension cable running from the EOS module to the BOS module of the next segment.
4	From the Catalog Browser, select a suitable PDM and insert it directly to the right of the BOS module in the new extension segment.
5	Populate the new extension segment with the desired I/O modules, and then end the segment with either a termination plate or another extension module.

Extending STB Islands to Preferred Modules

A preferred module is a module from another Schneider Electric catalog, or potentially from a third-party developer that complies with the Island protocol. A preferred module is not designed to fit in the standard form factor of an Advantys STB I/O module and does not fit into a standard base. It therefore does not reside in an Island segment. A preferred module requires its own power supply, and does not receive any logic power from the Island bus.

An Island can support a maximum of 31 preferred modules. Each segment has to include at least 1 auto-addressable Advantys STB module (the NIM in the primary segment and at least 1 I/O in any other segment). You may use Island bus extension cables to daisy-chain multiple preferred modules together.

Proceed as follows to extend an Island bus from an existing segment to 1 or more preferred modules:

Step	Action
1	Delete any termination plate you may have inserted at the end of the existing segment.
2	From the Catalog Browser, select the STB XBE 1100 EOS extension module and place it at the end of the existing segment.
3	Double-click the desired preferred module in the Catalog Browser. **Result:** The Advantys Configuration Software positions the selected module to the right of the existing segment, off the DIN rail.
4	To add another preferred module, double-click the module in question in the Catalog Browser. **Result:** The Advantys Configuration Software positions the selected module to the right of the previous preferred module, off the DIN rail.
5	To extend the Island bus to another segment of Advantys STB I/O modules, double-click the STB XBE 1300 BOS module in the Catalog Browser. **Result:** The Advantys Configuration Software creates a new DIN rail for the extension segment.
6	To designate a preferred module as the last module on the Island bus, add the TeSYS U LU9 RFL15 termination device.

Extending STB Islands to Enhanced CANopen Devices

An Island bus is able to support a maximum of 12 CANopen devices. Each segment has to contain at least 1 auto-addressable Advantys STB module (the NIM in the primary segment and at least 1 I/O in any other segment).

NOTE: Enhanced CANopen devices are installed as last devices on the Island after all Advantys STB I/O module segments and preferred modules have been placed.

Proceed as follows to extend the Island bus to enhanced CANopen devices:

Step	Action
1	Delete any termination plate that may be at the end of the existing segment.
2	From the Catalog Browser, select the STB XBE 2100 CANopen extension module and place it at the end of the last segment.
3	Double-click the desired CANopen device in the Catalog Browser. **Result:** The selected CANopen module is placed below the CANopen extension module. The Island Editor also shows a CANopen extension cable running from the extension module to the CANopen device.
4	To add another enhanced CANopen device to the Island bus, double-click the name of that device in the Catalog Browser. **Result:** The new CANopen device is displayed to the right of the previous CANopen device. Both are connected by a CANopen extension cable.
5	Install a termination plate to the right of the CANopen extension module on the last segment and physically terminate the Island. **Note:** An Island bus cannot be extended back to Advantys segments from a CANopen device. Remember that it is your responsibility to properly terminate the last CANopen device on the Island bus.

Terminating an Island

Each Advantys STB Island bus has to be properly terminated at its beginning and end. The beginning of an Island is integrated into every NIM. If you position the NIM correctly, that is in the leftmost slot of the primary segment, you provide the required terminator for the beginning of the Island bus. The end terminator for an Advantys STB system has to be positioned as the final component of an Island bus assembly. No module may be installed to the right of the Island's end terminator. The type of end termination required depends on the Island configuration.

Observe the rules listed below to ensure the proper termination of the Island:

Last Module Type	Required Termination
Advantys STB I/O Module	STB XMP 1100 termination plate
Preferred Module	TeSys U LU9 RFL 15 termination device
Enhanced CANopen Device	STB XMP 1100 terminator right of the CANopen extension module; physical termination of the last CANopen device

Locking and Protecting an Island

Locking and Unlocking an Island

When you create a new Workspace or a new Island in an existing Workspace, the new Island opens in an unlocked state.

When an Island is unlocked in a Workspace, you can perform all actions available in the Advantys Configuration Software. After you have saved an Island configuration and closed it, it will be in a locked state whenever it is reopened. In addition, when the software is in online mode, you cannot modify the configuration stored in the project file.

Locked Island

If an Island is locked, you can only view the configuration. You can

- view the Island configuration in the Island Editor.
- view the parameter values assigned to the modules on the Island using the Module Editor.
- view the reflex actions configured for the Island using the Reflex Editor.
- export a definition file describing the Island for the following fieldbuses:
  - CANopen
  - DeviceNet
  - Ethernet
  - Interbus
  - Modbus Plus

The Workspace Browser will display the Island tree and its branches with a lock icon next to each item. The lock status is also displayed on the title bar. All menu commands that would enable you to modify the Island configuration are grayed out.

Optionally, you may set a password to limit write-access to the configuration. If the lock is password protected, enter the password to unlock the Island and edit it.

Unlocking an Island

Proceed as follows to unlock an Island:

Step	Action	Result
1	Click the following icon: 	If the lock is not password protected, the Island unlocks. The lock icons disappear and the menu commands and toolbar buttons are enabled.
2	If the lock is password protected, enter the password into the dialog box that appears and click **OK**.	The lock icons disappear, menu commands and toolbar buttons become enabled. Now, you can edit the configuration.

Setting and Changing Passwords

Proceed as follows if you are working offline with an unlocked Island and want to apply a password to the lock or change an existing password:

Step	Action
1	While the new *.isl* file is active in the Workspace, click the following icon:  **Result:** A message is displayed asking you if you want to set a password.
2	Click **Yes**. **Result:** The **Set Password** dialog box is displayed.
3	In the **New password:** field, ● type a password if you want to set a password. ● type a new password if you want to change an existing one. **Note:** Valid characters for the password are alphanumeric characters.
4	Retype the password in the **Confirm:** field. **Note:** If the text strings do not match, you are prompted to restart the process.
5	Click **OK**. **Result:** A message is displayed prompting you to save the file with the new password.
6	Click **OK**. **Result:** The password just set is now enabled and the Island is locked. Users will be required to enter it to unlock the Island configuration. Any existing old password no longer applies.

NOTE: Passwords are case sensitive.

Building an Island Configuration

Introduction

For Islands containing STB, OTB Ethernet and OTB Modbus NIMs, the **Build** command is available. After you have completed an Island configuration and expect that it is valid, use this command to verify its correctness. The build process analyzes the layout for conformity to design rules and creates a downloadable configuration binary file for the Island.

NOTE: The **Build** command is not available for Islands containing OTB CANopen, FTB and FTM NIMs.

Before Building an Island

Before you issue a **Build** command, check that

- the last module on the Island bus is terminated,
- any reflex action that you have configured has valid inputs and is mapped to a valid action module,
- your module count and current draw do not exceed the Island's absolute limits,
- all customized module parameters are correctly configured,
- you have not exceeded the Island's resource utilization limitations.

Building the Island Configuration

To build an Island, click the following icon on the **Island** toolbar.

The build process will configure:

- communication objects for the Island
- the reflex actions
- the modules' customized operating parameter values
- the synchronization objects on the Island
- the process image area in the NIM
- an HMI area in the Island's process image

It will also create

- intermediate log files for the Island, if the **Generate intermediate lock files** option is checked in the **Settings** dialog box and
- a downloadable binary file.

Validating the Build

If the configuration is not valid, the build will not be successful. The Log Window will display the associated diagnostic messages. If the configuration is valid, the build will succeed and the following message will be displayed: `Build completed successfully`.

Diagnostics Logging During the Build Process

The software performs analysis while building. During the analysis phase, the software will:

- detect errors that can block the build process.
  Diagnostic messages are posted in the Log Window. The detected errors are caused by the violation of an absolute limit that renders the configuration unusable.
- provide notifications about detected errors that do not block the build process.
  A notification will be posted in the Log Window.

During the binary file generation, the software may encounter diagnostics caused by

- the unavailability of objects for the modules in the Island.
- an inconsistency in some of the database objects.

Design Rules for Building Island Configurations

Introduction

As the software builds your *.isl* file, it checks the data to check that it conforms to the Advantys STB design rules.

Design Rules for Module Placement

The software checks these Advantys STB design rules concerning module placement:

- Exactly 1 NIM has to be present on the Island, and it has to be the first module on the primary rail.
- A PDM has to be directly to the right of the NIM.
- There shall be no more than 7 rails (1 primary and 6 extensions) on the Island.
- A BOS module shall be the first module on every extension rail.
- A PDM shall be directly to the right of each BOS module on the extension rail(s).
- An EOS or preferred module shall be directly to the left of a BOS module
- There shall be no more than 32 I/O modules across all the rails on the Island.
- The last module on the Island shall be terminated with a 120 Ω resistor.

Design Rules for Power Supply

The software checks the following Advantys STB design rules concerning power distribution and consumption:

- There shall be no more than 16 I/O modules per PDM.
- The voltage distributed by all PDMs on the Island has to be appropriate for the I/O modules they support.
- The I/O modules on each rail may not consume more than a total of 1.2 A of logic power.
- The input modules in a voltage group should not draw a total of more than 4 A of sensor power from the PDM that supports them. The software will produce an alert if this condition is detected, and the build will continue.
- The output modules in a voltage group should not draw a total of more than 8 A of actuator power from the PDM that supports them. The software will produce an alert if this condition is detected, and the build will continue.

Design Rules for the Project Configuration

The software checks the following Advantys STB design rules concerning your project configuration:

- There shall be no more than 10 prioritized input modules on the Island.
- There shall be no more than 10 reflex action blocks in the Island configuration.
- No more than 2 reflex blocks shall be configured for any 1 output module on the Island.
- If an output module has reflex blocks mapped to it, it shall be able to support those reflex type(s).
- If an HMI panel is used on the Island, the maximum size of the HMI input/output table shall not exceed the size that you have configured. The HMI table size is a user-configurable parameter for the NIM, which you set in the Module Editor.
- The size of the HMI input/output table plus the block required for standard I/O data exchange shall not exceed the maximum data exchange size imposed by your fieldbus. Refer to your NIM documentation for this limit.

Design Rules for CANopen Devices

The software checks the following design rules concerning CANopen devices:

- If 1 or more standard CANopen devices are used on an Island, the last module on the last rail of the Island shall be a CANopen extension module. A termination plate shall be installed on the rail after the CANopen extension module.
- The address of any standard CANopen device shall not duplicate the address of an Advantys STB module or a preferred module on the Island.
- There shall be at least 1 CANopen device and no more than 12 CANopen devices after the CANopen extension module.
- The Island baud rate has to be 500 kBaud if a CANopen device is present.
- Mandatory modules are not allowed if a CANopen device is present.

Design Rules That May Not Be Checked by the Software

There are 2 important design rules that are not checked in the build process by the software. Be aware of these rules when designing your physical Island:

- If the last module on the Island is a preferred module or a standard CANopen device, you shall provide 120 Ω termination on that module/device.
- The maximum length of the physical Island shall be no more than 15 m (49.2 ft) if standard CANopen devices are not used and no more than 6.5 m (21.3 ft) if standard CANopen devices are used. Standard CANopen devices and cables shall be calculated as part of the total Island length.

Section 3.8
Online Features

Introduction

This section provides an overview of the online features of the Advantys Configuration Software.

What Is in This Section?

This section contains the following topics:

Online Feature Introduction

Introduction

When the logical Island active in the Island Editor is connected to the physical Island, the software is switched to online mode. Online mode is supported by the following product and fieldbus combinations:

- STB
  - all NIMs via the local serial port
  - Ethernet NIMs via the upstream network
- OTB Ethernet and Modbus NIMs via the upstream network

Available Features for STB NIMs

Depending on the fieldbus type, the following online operations are possible:

- connecting to the physical Island
- disconnecting from the physical Island
- downloading the configured data into the Island
- saving a downloaded configuration to a removable memory card in the NIM
- uploading the configured data from the Island
- reading global diagnostics data from the Island
- forcing auto-configurations
- changing configuration port settings
- controlling the Island (run, stop, reset)
- selecting and monitoring the process image animation data
- forcing outputs from the test mode

Available Features for OTB NIMs

OTB NIMs support a restricted number of online operations, which are

- connecting to the physical Island
- disconnecting from the physical Island
- downloading the configured data into the Island
- uploading the configured data from the Island
- changing the connection settings

STB Island States

Different STB Island States

This table lists the possible Island states, shown in the status bar in online mode:

Island State	Green	Red	Description
	blink 2	blink 2	The Island bus is powering up (self test in progress).
0x00-Initializing	off	off	The Island bus is initializing but not started or without power.
0x40-Reset	blink 1	off	The Island bus has been put in the pre-operational state by the RST button, or it has been reset by the software.
	off	blink 8	The contents of the removable memory card is invalid.
0x60-Configuration 0x62-Auto-Addressing	blink R	off	The NIM is configuring or auto-configuring the Island bus, which is not started.
0x80-Pre-Operational	blink 3	off	Initialization is complete, the Island bus is configured, the configuration matches – the Island bus is not started.
0x81-Non-Mandatory Module Mismatch	blink 3	blink 3	Configuration mismatch – non-mandatory or unexpected modules in the configuration do not match; the Island bus is not started.
0x82-Mandatory Module Mismatch	blink 3	blink 2	Configuration mismatch – at least 1 mandatory module does not match; the Island bus is not started.
0x83-Configuration Mismatch	off	blink R	Detected error; so severe that no more communication with the Island bus is possible and the NIM stops the Island. These are the detected errors: ● internally detected error ● module ID mismatch ● incorrect auto-addressing ● incorrect configuration of mandatory module ● detected error in process image ● incorrect auto-configuration/incorrect configuration ● detected error in Island bus management ● detected error in application parameter ● detected error in receive/transmit queue software overrun
A0-Running	on	off	The Island bus is operational.
A1-Running (...)	on	blink 3	At least 1 standard module does not match – the Island bus is operational with a configuration mismatch.
A2-Stopped (...)	on	blink 2	Serious configuration mismatch. The Island is in pre-operational mode because of 1 or more mismatched mandatory modules.
C0-Stopped	blink 4	off	The Island bus is stopped, no more communication with it possible.

Transferring a Configuration

Introduction

Once the Advantys Configuration Software has established the communication to a physical Island, you can either transfer the logical configuration from the software to the Island (**Download into the Island**) or load the existing configuration of the physical Island into the software (**Upload from the Island**).

For more information on how to establish the communication to a physical Island, please refer to *Connect, page 286*.

Download

The **Download into the Island** command allows you to transfer a configuration file previously built in the Advantys Configuration Software to the connected physical Island. You can download the Island configuration into the physical Island only if the Island is in online mode. The configuration file is downloaded into the NIM's FLASH, where it can then be optionally saved to a removable memory card.

To perform a configuration download, select the **Download into the Island** option from the **Online** menu.

When you download a configuration into a physical Island, the Island automatically enters reset state. You are notified that the physical Island will reset.

NOTE: You cannot perform a download when the Island is in protect mode unless you enter the correct protect mode password. During the download process, a progress bar is displayed, tracking the status of the download.

Upload

The **Upload from the Island** command allows you to upload a configuration file to the Advantys Configuration Software from a physical Island. The configuration is uploaded to the logical Island that is currently open in the Island Editor of the Advantys Configuration Software. You may upload the physical Island's configuration only when the software is in online mode.

The process will abort while uploading the configuration from the physical Island if 1 of the module IDs or the major firmware version of 1 of the modules do not match or any detected errors occur. Diagnostic messages will be displayed in the Log Window.

If you use the **Connect** option from the **Online** menu, the software performs a configuration consistency check (see *Connection Process, page 287*).

Uploading an Island Configured Using Auto-Configuration

When an auto-configured physical Island is uploaded, any module that does not communicate on the Island bus is not uploaded. The software inserts necessary modules, such as power distribution or termination modules. However, the modules inserted may not exactly match the ones configured.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION
After an upload, check the configuration of Islands that have been constructed using auto-configuration. If necessary, insert or exchange the modules concerned using the Catalog Browser.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

The following description explains the procedure for uploading such a configuration step by step.

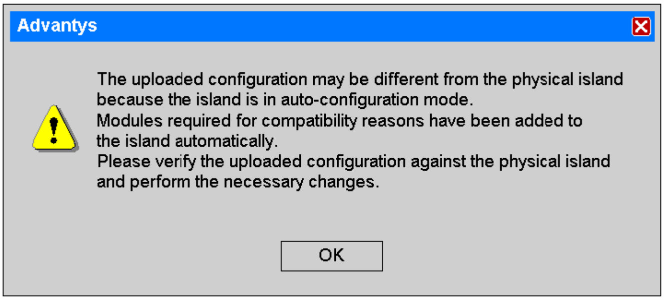
The table shows a physical STB Island consisting of 2 segments and a possible result of an upload into the Advantys Configuration Software:

Physical Island		Uploaded Logical Island		Type of Module
Slot	Module	Slot	Module	
1	STB NDN 2212	1	STB NDN 2212	network interface
2	STB PDT 3100		**STB PDT 3105**	power distribution
3	STB ACI 1230	6	STB ACI 1230	I/O module
4	STB AVI 1270	7	STB AVI 1270	I/O module
5	STB DRC 3210	8	STB DRC 3210	I/O module
6	STB XBE 1000		-	end of segment
Second Segment				
1	STB XBE 1200		-	beginning of segment
2	STB PDT 2100		**STB PDT 2105**	power distribution
3	STB DAI 7220	10	STB DAI 7220	I/O module
4	STB DAO 8210	11	STB DAO 8210	I/O module
5	STB XMP 1100		STB XMP 1100	BUS termination

The resulting configuration does not equal the one configured in the physical Island.

Correcting the Uploaded Configuration

After the configuration has been uploaded, the following message is displayed reminding you to compare the configuration uploaded and the one configured:



To get a configuration equivalent to the original physical Island, perform the following steps:

Step	Action
1	Exchange the power distribution modules using the Catalog Browser.
2	To select the modules you need to transfer to the second segment, click the first module to select and with the SHIFT key pressed click the last module to select.
3	Cut the modules using the CTRL+X key combination or clicking the following icon on the **Edit** toolbar: 
4	Add an end-of-segment module XBE1000 and a beginning- of-segment module XBE1200. **Result:** A new segment is automatically created.
5	Drag the previously cut modules to the new segment using the CTRL+V key combination or clicking this icon on the **Edit** toolbar: 

NOTE: Your configuration now reflects the actual physical Island.

Protect Mode

Introduction

You can limit the access to the available online functions for STB Islands by putting the physical Island into protect mode. When online, you can toggle between protect and protect edit mode. The mode of the Island is displayed in the title bar adjacent to the Island's name and indicated by a check mark next to the **Protect** option.

If the Island is protected, enter a user-assigned password in order to:
- change configuration port settings
- change the state of the physical Island
- download configurations
- store to SIM card
- unprotect the Island
- activate/deactivate test mode
- force output data in test mode
- force auto-configuration

NOTE: After you have downloaded a configuration into a physical Island being in protect edit mode, the Island will no longer be protected.

Connecting to Protected Islands

When connecting to a physical Island which is in protect mode, you are prompted to enter a user-assigned password:

If you ...	Then you can ...
enter the correct password	perform all online operations. The title bar will indicate this by displaying *Protected (Edit Mode)*.
click **Cancel**	only monitor the Island in online mode because it remains protected.
enter an incorrect password	enter the password again by selecting **Online** → **Protect**. After 3 attempts, you are disconnected.

Setting Passwords

When you change the Island to protect mode, you are prompted to set a password:

If you select ...	Then ...
No	the Island will be protected without a password (empty password).
Yes	you will be prompted to enter a password. **Note:** A valid password comprises between 0 and 6 alphanumeric characters. An empty password is also valid. When the protect mode is released, the password is deleted.
Cancel	the Island remains unprotected.

Test Mode

Introduction

Only 1 device is allowed to control the physical Island's process image at any given time. The master device is the fieldbus master. In order to control Island outputs locally by the Advantys Configuration Software or by an HMI, the test mode needs to be activated. The temporary test mode option in the Advantys Configuration Software allows the software to obtain or release control over the process image. In the temporary test mode, the upstream fieldbus is not able to write the outputs but still able to read the inputs and diagnostic data. The activation of the test mode is indicated by the test LED on the NIM switched on and also by a bit (CTM_PIO) in the NIM device status. While the software has mastery over the process image, the fieldbus/PLC can read from but not write to the physical Island.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION
Use test mode only for testing your Island I/O. Do not put the Island into test mode when the system is in normal operation. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

Test Mode Settings

There are 3 test mode options available:
- temporary test mode (available for all STB NIMs)
- persistent test mode (only available for standard STB NIM of V2.*x* or later)
- password test mode (only available for standard STB NIM of V2.*x* or later)

Temporary Test Mode

The temporary test mode is the default configuration supported by all STB NIMs. It is a toggle option allowing you to obtain or relinquish control over the process image:

If you want ...	Then ...
to activate the temporary test mode	click **Test Mode** on the **Online** menu. **Note:** You have to be in online mode.
the software to control the Island’s process image	click **OK**. **Result:** The status bar reads **Test ON** in the test mode pane.
the fieldbus to retain control of the Island’s process image	click **Cancel**. **Result:** The status bar reads **Test OFF** in the test mode pane.

Deactivating the Temporary Test Mode

If you disconnect from the physical Island in test mode, the software automatically attempts to relinquish its control of the process image and displays a message box asking whether you want to deactivate the test mode.

The temporary test mode can also be deactivated by
- deselecting the **Test Mode** command on the **Online** menu.
- power cycling the Island.
- changing the configuration.

Persistent Test Mode

This mode has to be configured using a config tool. After the download of this configuration has been completed, the NIM enters test mode automatically. The test mode option as part of the configuration data is stored non-volatile in the NIM. After the power cycle has been completed, the NIM reenters the persistent test mode. The persistent test mode can only be released after the configuration has been changed.

Password Test Mode

The password test mode allows an HMI to control the Island using public Modbus commands. This mode has to be configured using config tool. A respective password has to be configured. After downloading the configuration, an HMI device can activate the password test mode by using a public Modbus command to write the password value to register 45120. This write has to be a single register write command. When this register is read, a dummy value (0000) will be returned. The password test mode can only be exited by power cycling the Island or by changing the configuration. Some NIMs provide additional ways to exit password test mode (see below and the corresponding NIM documentation for details). The password test mode can only be entered via the NIM's configuration port. Attempts to enter the password test mode via the fieldbus will be rejected.

If your Island contains an STB NIC NIM, you have the following possibilities to exit the password test:
- power cycling the NIM
- selecting **Online** → **Reset**
- performing an auto-configuration
- downloading a new Island configuration to the NIM (or inserting a SIM card with a new Island configuration into the NIM and power cycling the NIM)
- using an HMI to issue a single Modbus register write command to send the password value to Modbus Register 45121 (for STB NIC 2212 NIM only)

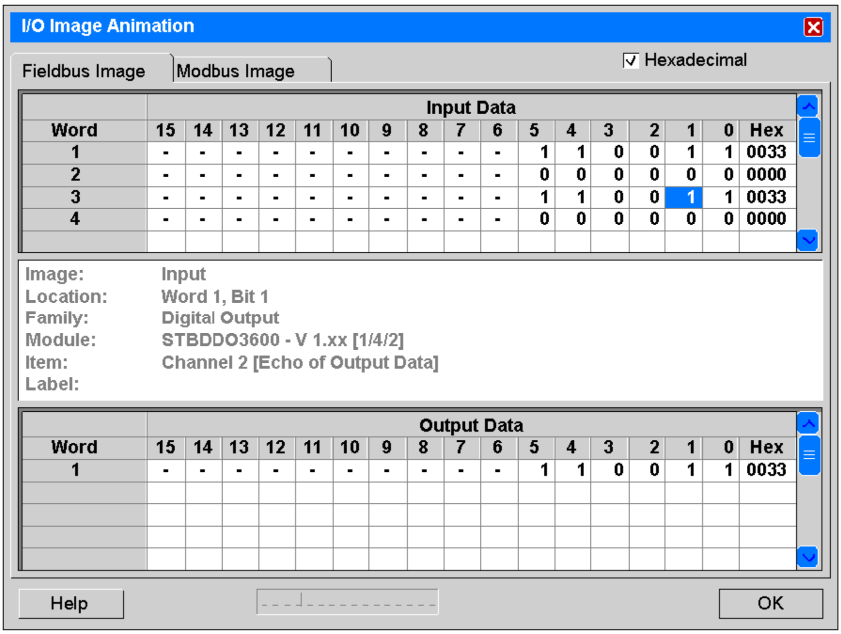
I/O Image Animation

Introduction

The I/O image animation is a dynamic display of I/O data exchanged between the physical Island and the fieldbus master (and between an HMI panel on the Island and the fieldbus). It is available only for STB Islands and when the software is online.

To open the **I/O Image Animation** dialog box, select **I/O Image Animation** from the **Online** menu or click the following icon on the **Island** toolbar:

The figure below shows an example of the **I/O Image Animation** dialog box:



Views of the I/O Data

The I/O image animation feature displays the process image data in 2 formats:
- Modbus, in which data are exchanged inside the physical Island and to the HMI
- the appropriate format for your fieldbus protocol, determined by the NIM type

An exception are Ethernet and Modbus Plus Islands. Because their Modbus view equals the fieldbus view, the I/O image animation contains only 1 view.

In addition, there are an **HMI <-> PLC** tab for Ethernet, Ethernet/IP and Modbus Plus Islands and **TxPDOs** and **RxPDOs** tabs for CANopen Islands.

Input and Output Tables

The **I/O Image Animation** dialog box is divided into 2 tables:
- **Input Data**
- **Output Data**

Input data are read from the Island by the fieldbus master, and output data are written to the Island by the fieldbus master.

The tables are organized as follows:
- Each row displays 1 word of data.
- Each row has 17 columns, 16 of which correspond to the bits in each word and 1 (the rightmost) that displays the total value of the word. The bit columns are numbered 0 to 15 from right to left. Bit 0 represents the least significant bit (LSB), and bit 15 represents the most significant bit (MSB). The total value of the word value will be displayed in either hexadecimal or decimal format, depending on your choice of integer display. Hexadecimal is the default.
- When you view the tables in the **Fieldbus Image** view, the rows contain word numbers, for instance **1**, **2**, **3**, and so on, or CANopen object references of the fieldbus interface of the NIM, for instance **6200sub1**, in case a CANopen NIM is used. Such a CANopen object can occupy several rows.
- When you view the tables in the **Modbus Image** view, the rows are named as the register numbers, for instance **45303**, **45304**, **45305**, and so on.

Each bit position contains a boolean **1** or **0** if the bit is used, or a **-** symbol if the bit is not used. The total value of the word is displayed in the selected format of hexadecimal or decimal. The word value column header is **Hex** if the integers are to be displayed in hexadecimal format or **Dec** if the integers are to be displayed in decimal format.

Outputs controlled by reflex action blocks will be highlighted with a yellow background. The value displayed is what is written by the fieldbus master and not necessarily what was written by the reflex block.

Output values controlled by reflex actions are not reflected in the **Output Data** area. You can monitor these values if the **Input Data** area contains echo bits of the output data concerned (this is the case if the Island contains digital outputs). In this case, these echo bits reflect the output values controlled by reflex actions.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Do not depend on data values displayed with yellow background in the I/O animation table.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Bit Arrangements for NIM Objects

This figure shows how a sequence of an 8 bit and a 24 bit object is displayed with bit numbers designated by a leading **b** in the table:

	Input Data															
NIM Object	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
6000sub8	-	-	-	-	-	-	-	-	b7	b6	b5	b4	b3	b2	b1	b0
2000sub1	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	-	-	-	-	-	-	-	-	b23	b22	b21	b20	b19	b18	b17	b16

Cell-Related Information

The information corresponding to the cell is displayed in the area between the 2 tables. It is displayed when you select a cell. No information will be displayed if you select a cell containing **-**.

When you select a cell containing a number, the following information is displayed:
- image table selected
- bit location (word, bit number for fieldbus; register, bit number for Modbus)
- module family
- module’s name, and its topological information (rail/slot/node)
- channel number and type of information (data or status)
- label (symbol) assigned in the Module Editor

Data Transfer and Fieldbuses

The high byte and the low byte of each word may be used to display the data from different modules. Refer to *Data Transfer and Fieldbuses, page 176*.

Modifying the Values in the Tables (Test Mode Only)

You can modify the boolean values in the used bit positions from **1**s to **0**s and vice versa. You cannot modify the bit positions marked with **-**.

Depending on the data item type, edit the values in the bit positions as follows:

Data Item	Description
Digital Output	Values are edited on the bit level by modifying the single bits.
Analog Output	Values are edited on the object level by modifying the value of the whole object. An object can occupy several rows and has to be modified consistently.
RTP Request Register	
HMI-to-PLC Data	

The values in the 16 bit positions of a row determine the total word value in the rightmost column. Each word can have a total value in the range 0x0000 to 0xFFFF hex (0 to 65,535 dec). The modified word values will be set to the physical Island.

NOTE: The cell values can be modified only when the NIM is in test mode and in run state. Only the output table values, the RTP request registers and the HMI-to-PLC data of the physical Island can be modified.
The values of all the tables are updated dynamically from the physical Island.

Editing on the Bit Level

To edit data items on the bit level, perform the following steps:

Step	Action
1	• Double-click the bit value cell concerned or • single-click the corresponding bit value cell concerned and press F2.
2	Type the alternative boolean value.
3	To set the value to the physical Island, • click the next cell to be edited or • press ENTER or • click **OK** if you have finished editing.

Editing on the Object Level

To edit data items on the object level, perform the following steps:

Step	Action
1	• Double-click the word value cell concerned or any of the correspondig bit value cells • single-click the word value cell concerned or any of the corresponding bit value cells and press F2. **Note:** If you use a word value cell for editing and this I/O image word contains more than 1 data item, you are asked which one you want to edit. **Result:** A helper dialog is displayed. It provides information on the data type to ensure data consistency.
2	In the **Value** area, type the new word value. **Note:** It is indicated if you type an invalid value regarding the data type.
3	Click **OK** to close the helper dialog and set the value to the physical Island.

Keyboard Navigation

Keep the following points in mind for the keyboard navigation:

Keyboard Navigation	Description
Keyboard Arrows	to move up, down, left and right through the cells in the input and output tables
F2	to modify a value in a cell
TAB	to navigate through the controls in the window
CTRL+TAB	to toggle between the **Fieldbus Image** view and the **Modbus Image** view **Note:** NIP and NMP modules allow the **Modbus Image** view only. In that case, the **Fieldbus Image** view is hidden.

To close the **I/O Image Animation**, click **Close**

Section 3.9
Online/Offline Features

Introduction

This section describes the features available in online and offline mode.

What Is in This Section?

This section contains the following topics:

Resource Analysis

Introduction

Using the **Resource Analysis** option, you can view a bar graph representing various aspects of the Island resource consumption. It is available only for STB Islands.

The bar graph provides information regarding the following:
- I/O Image area
- HMI Image area
- logic power
- field power

Resource Consumption

The consumption report on each of the resources is detailed in the table below:

Resource	Description
I/O Image Area	separate values for input and output data images
HMI Image Area	HMI-to-PLC and PLC-to-HMI data images
Logic Power	power consumption for each Island segment
Field Power	sensor and actuator power consumption for each PDM

Accessing the Resource Analysis

The **Resource Analysis** dialog box displays a report pertaining to the consumption of resources by the Island.

To access the **Resource Analysis** dialog box, select **Island** → **Resource Analysis** or click the following icon on the **Standard** toolbar:

The **Resource Analysis** option leads to a window including 2 tabs:
- **Power**
- **Configuration**

Power Tab

The power delivered by each module is displayed in the **Power** tab of the **Resource Analysis** dialog box. The **Power** tab is the default tab when the **Resource Analysis** dialog box is invoked. Bars are dynamic for this tab and are updated as you construct the Island.

The table below lists the different types of power and resources:

Type of Power	STB Islands
logic power	• NIM • BOS • auxiliary power supply
actuator power	PDM
sensor power	PDM

Logic Power

The Logic Power graph illustrates the 5 V supply consumption by various I/O modules. Depending on the Island type, sources for this power are the NIM (network interface module), the BOS modules and the auxiliary power supply. The Island does not build successfully if the logic power usage for the Island exceeds the logic power available. In this case, the bar color switches to red to denote that the build is not successful.

Field Power

The sensor/actuator power graphs represent the field power consumption by various I/O modules. Depending on the island type, the source for this power is the PDM or the NIM. The power is supplied to field devices connected through the I/O modules cluster to the right of the resource module concerned. When you attempt to build, the Island displays a notification if the actuator or sensor power for any of the PDMs or NIMs exceeds the available power, as indicated by the yellow bar.

Different colors identify the consumption ranges of the resources. Each grid line represents 20% of the power consumption of the resource.

Configuration Tab

The **Configuration** tab graphically represents the following parameters:

- input process image
- output process image
- HMI-to-PLC area
- PLC-to-HMI area

NOTE: The **Configuration** tab is not dynamic: it displays the data which had been gathered when the **Resource Analysis** dialog box was invoked.

Input Process Image

This is the area where all input data are mapped. Normally, the input data are transferred from the process image to the upstream system. Each NIM imposes a limit on the amount of data that can be transferred to the upper level network during the I/O data exchange process. The input process image bar displays the percentage of the allowable fieldbus packet size that is transferred to the upstream system. If the configured input process image size exceeds the fieldbus packet size, the build is not successful. The bar color switches to red to indicate that the build is not successful.

Output Process Image

This is the area where all output data are mapped. Normally, the output data are transferred to the process image from the upstream system. Each NIM imposes a limit on the amount of data that can be transferred to the upper level network during the I/O data exchange process. The output process image bar displays the percentage of the allowable fieldbus packet size that is transferred from the upstream system. If the configured output process image size exceeds the fieldbus packet size, the build is not successful. The bar color switches to red to indicate that the build is not successful.

The Profibus fieldbus protocol is the exception to these 2 rules: For Profibus NIMs, the build is not successful if the **sum** of input and output data exceeds the fieldbus packet size limitation. Both the input and output bar turn red to indicate that the build is not successful.

HMI-to-PLC Area

You may also configure a Modbus HMI by connecting to the configuration port on the NIM, in order to communicate with the upstream PLC. These data are transferred along with the input data to the upstream system. This is the area where the HMI-to-PLC data are mapped. The HMI-to-PLC bar graph indicates whether a resource is available and displays a percentage corresponding to the current usage of each resource.

PLC-to-HMI Area

You may also configure a Modbus HMI by connecting to the configuration port on the NIM in order to communicate with the upstream PLC. These data are received along with the output data from the upstream system. This is the area where the PLC-to-HMI data are mapped. The PLC-to-HMI bar graph indicates whether a resource is available and displays a percentage corresponding to the current usage of each resource.

I/O Image Overview

Introduction

The Advantys Configuration Software provides a utility providing an overview of the I/O data and status allocation for all modules on the Island. It also gives you a view of any data that may be written to the Island bus or read by the fieldbus master.

To access the **I/O Image Overview** dialog box, select **Island** → **I/O Image Overview** or click the following icon on the **Island** toolbar:

In online mode, the I/O image overview for STB Islands can be animated using the I/O image animation feature.

Views of the I/O Data

The **I/O Image Overview** dialog box provides 2 tabs containing different views of the I/O data and status information:

- **Fieldbus Image** tab containing the fieldbus view
- **Modbus Image** tab containing the Modbus view

The **Fieldbus Image** tab is displayed by default when you select the **I/O Image Overview**. In case of Ethernet and Modbus Plus NIMs, the fieldbus view equals the Modbus view. Therefore, only 1 tab is displayed. In case the Island contains an Ethernet/IP NIM, both tabs are displayed because this type of NIM is based on DeviceNet.

Both I/O image views display data and status bits in tabular form. Each view has:

- input table
- output table

Help Function

When using the **I/O Image Overview** or the **I/O Image Animation** options, you can access the corresponding sections in the *Advantys Configuration Software Online Help* by clicking the **Help** button.

Input and Output Tables

The input table displays the data and status associated with the input modules on the Island bus and the RTP data, if selected. Input modules send information from the sensors to the fieldbus master. The output table displays the data associated with the output modules on the Island bus and the RTP and virtual modules data, if selected. Output modules receive application data from the fieldbus master and use it to update field actuator devices.

HMI-to-PLC and PLC-to-HMI data are also displayed in the input and output tables except for Islands containing STB Ethernet, Ethernet/IP and Modbus Plus NIMs. In these cases, the **I/O Image Overview** dialog box contains a separate tab for displaying the HMI-to-PLC and PLC-to-HMI data, see *I/O Image of STB Ethernet and Modbus NIMs, page 178*.

The input and output tables are organized as follows:
- Each row displays a word of data.
- Each row has 16 cells, each word holds 16 bits. Bits are numbered 0 to 15, where bit 0 is the rightmost bit and bit 15 is the leftmost bit.
- In the **Fieldbus Image** tab, each row is identified by a word number. In the **Modbus Image** tab, each row is identified by a Modbus register number.

The numbers that appear in certain cells correspond to a specific module's address on the Island bus. The data or status bit in that cell is associated with that module.

NOTE: Output data controlled by reflex actions are indicated by a yellow background of the corresponding cell. Data areas corresponding to a module set as **Not Present** (Virtual Placeholder) are disabled.

Cell-Related Information

Specific Island information corresponding to a selected cell is displayed in the area between the 2 tables in both the fieldbus and the Modbus view.

When you click a cell containing a number or a letter, the following information will be displayed:

Information	Meaning
Image:	input or output information which the selected bit contains
Location:	word number, Modbus register number, bit position in that word
Family:	type of module in which the selected bit operates (e.g. digital input, analog output, preferred module, and so on)
Module:	module name and its topological information (segment/slot/node)
Item:	channel in which the selected bit resides and whether it contains data or status information
Label:	label (symbol) assigned to the data element in the Module Editor

Keyboard Navigation

The keyboard navigation works as follows:

Key	Description
Keyboard Arrows	move left, right, up, down through the input and output table cells
F2	modify a value in a cell
TAB	navigate through the controls on the window
CTRL+TAB	toggle between the **Fieldbus Image** and the **Modbus Image** tab

Data Transfer and Fieldbuses

The fieldbus image view in the software provides a word-wide display. Some fieldbus networks, such as CANopen, Profibus DP, and DeviceNet handle data transfers of bytes in form of words. When you are viewing data from 1 of these fieldbuses in the fieldbus image view, the high byte and the low byte of each word may be used to display the data from different modules.

For example, if your configuration includes a STBDDO3400, STBDDO3600 and a STBAVO1250 with a STBNDP2212 Profibus DP NIM in that order, the output image will contain the following:

Address		Content
Word 1	low byte	I/O data for the STBDDO3400
	high byte	I/O data for the STBDDO3600
Word 2	low byte	I/O data for the STBAVO1250 (channel 1, low-byte)
	high byte	I/O data for the STBAVO1250 (channel 1, high-byte)
Word 3	low byte	I/O data for the STBAVO1250 (channel 2, low-byte)
	high byte	I/O data for the STBAVO1250 (channel 2, high-byte)

Depending on your configuration, the data from a word-wide channel such as the STBAVO1250 analog module may sometimes be split over 2 words in the fieldbus image view. For example, if the STBNDP2212 Profibus DP configuration includes only 1 STBDDO3400 followed by an STBAVO1250 module, the output image contains:

Address		Content
Word 1	low byte	I/O data for the STBDDO3400
	high byte	I/O data for the STBAVO1250 (channel 1, low-byte)
Word 2	low byte	I/O data for the STBAVO1250 (channel 1, high-byte)
	high byte	I/O data for the STBAVO1250 (channel 2, low-byte)
Word 3	low byte	I/O data for the STBAVO1250 (channel 2, high-byte)
	high byte	no data bits

Additional Functions

Irrespective of the fieldbus and the Modbus view, which is available for all fieldbuses, these additional functions are available depending on the fieldbus:

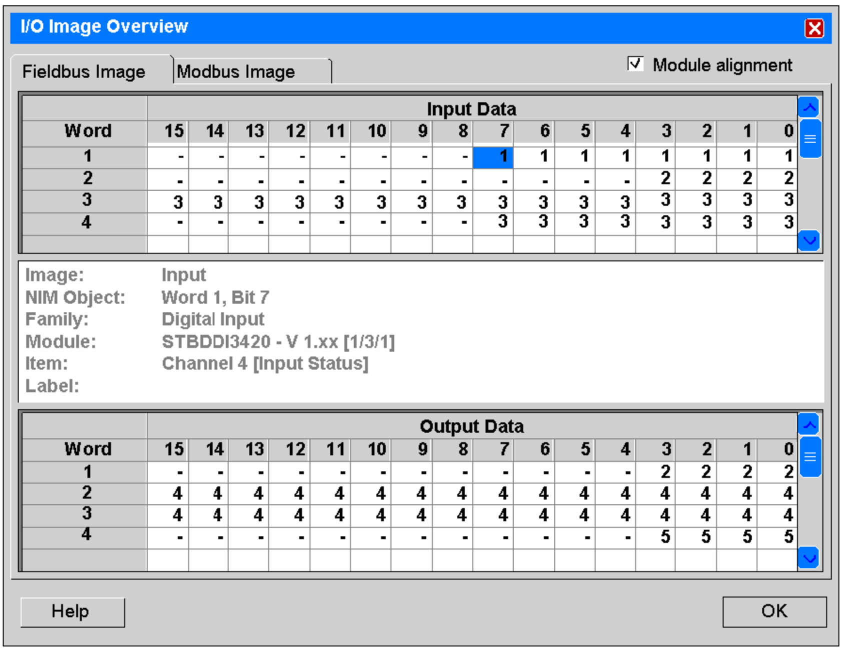
- **Module alignment** option for Islands containing Profibus DP NIMs
- **HMI <-> PLC** tab for STB Ethernet, Ethernet/IP and Modbus Plus NIMs
- **TxPDOs** and **RxPDOs** tabs for STB CANopen NIMs
- **TxPDOs**, **RxPDOs**, **PDO Configuration** and **Data Ranges** tabs for Islands containing FTB, FTM and OTB CANopen NIMs
- **Registers (read-only)**, **Registers (writeable)** and **Data Ranges** tabs for Islands containing OTB Ethernet and Modbus NIMs

Module Alignment for Profibus DP NIMs

The **Module alignment** option is enabled by default. With this option, the data of each I/O module are realigned on a word boundary in the I/O image. The resulting view matches the I/O data layout of a TSX Premium Profibus master.

You can disable the **Module alignment** option by deselecting the **Module alignment** check box.

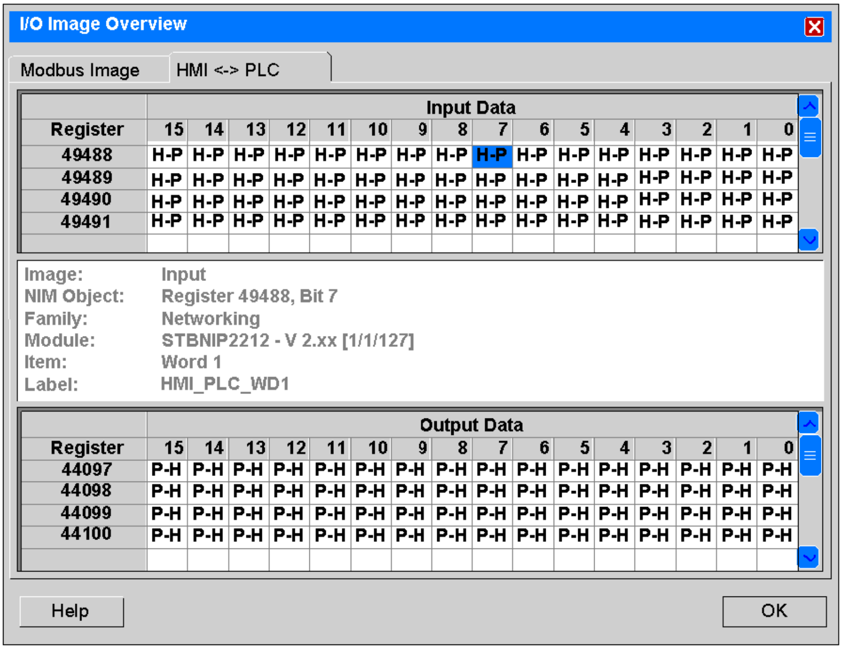
This figure shows an example of the module alignment for Profibus modules:



In this view, word-oriented analog data are typically aligned on word boundaries, that is analog data are not split over 2 different I/O words of the fieldbus I/O image.

I/O Image of STB Ethernet and Modbus NIMs

The following figure shows an example of an STB Ethernet NIM I/O image overview with the **HMI <-> PLC** tab selected:

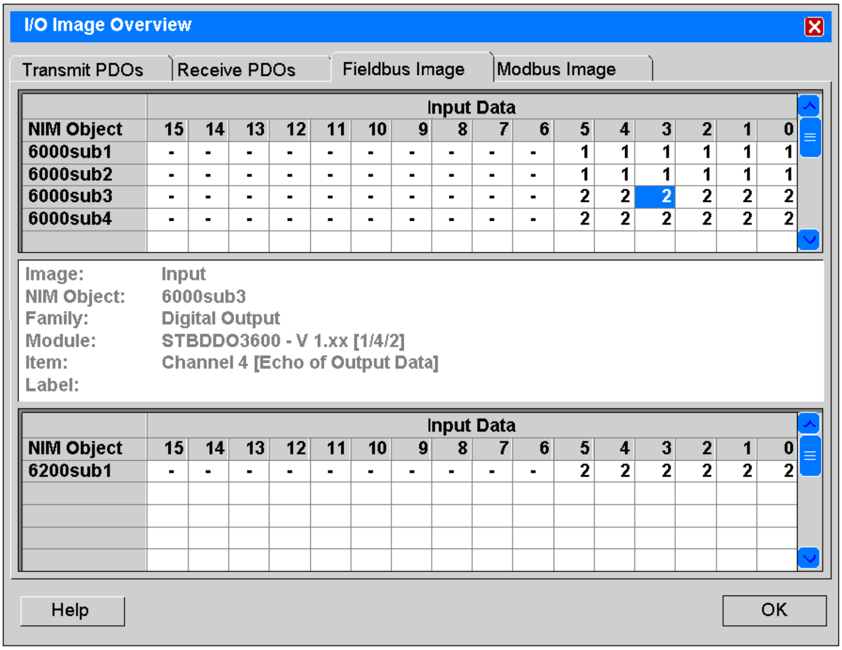


The **HMI <-> PLC** tab is available only for Ethernet, Ethernet/IP and Modbus Plus NIMs of the STB product family. It lists HMI-to-PLC data, displayed in the input table, and PLC-to-HMI data, displayed in the output table. If these data are not configured, the tables remain empty. For STB Ethernet and Modbus Plus NIMs, the leftmost column contains the registers belonging to the HMI-to-PLC and PLC-to-HMI data. In case of STB Ethernet/IP NIMs, this column contains words, which are counted up starting with the number *1*.

I/O Image of STB CANopen NIMs

The CANopen I/O image overview is object-based. Each input CANopen I/O data object is shown in 1 line. Objects exceeding a size of 16 bits are shown in subsequent lines.

The following figure shows an example of an STB CANopen I/O image overview:

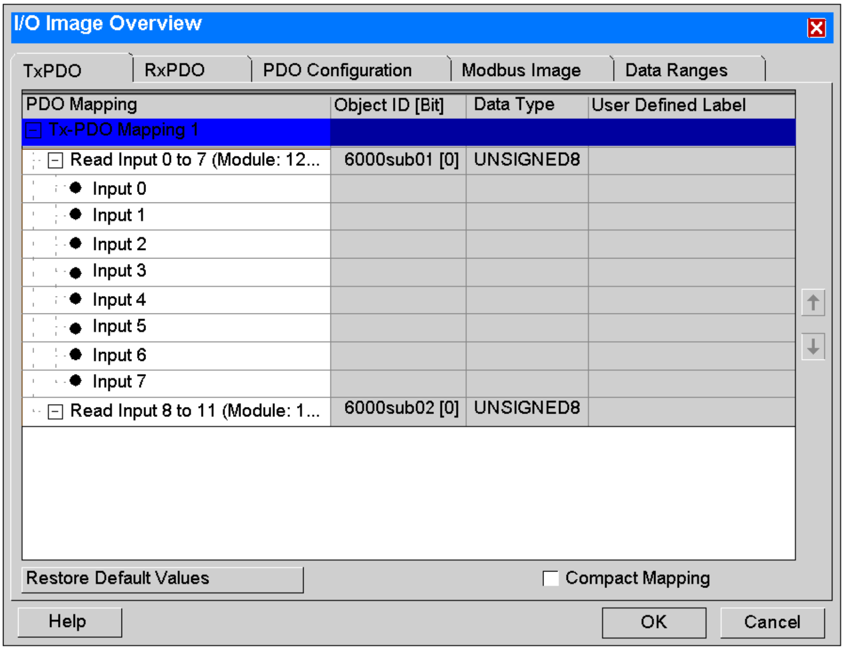


The **NIM Object** column contains the CANopen identification of the NIM I/O object, for example **6000sub03**.

The **TxPDOs** and **RxPDOs** tabs display the current PDO layout presented by the NIM on the upstream fieldbus.

I/O Image of FTB, FTM, OTB CANopen NIMs

The following figure shows an example of an OTB CANopen I/O image overview with the **TxPDOs** tab selected:

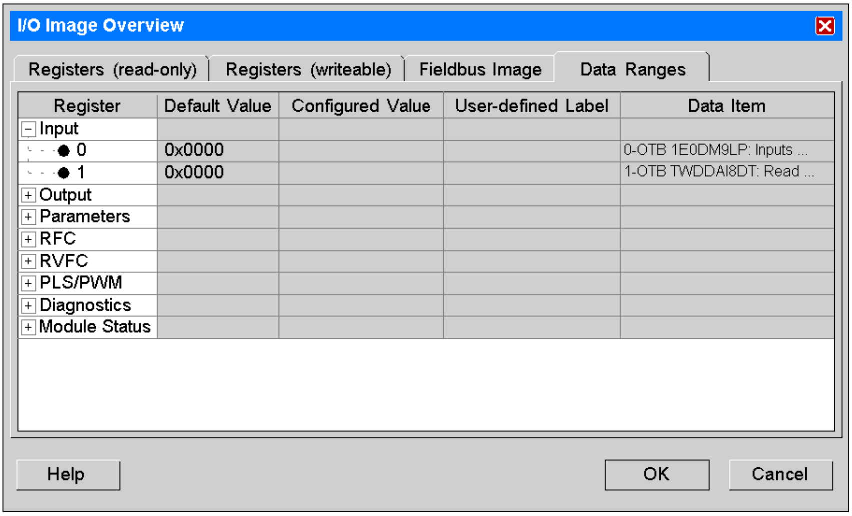


The **TxPDOs** and **RxPDOs** tabs display the current PDO layout presented by the NIM on the upstream fieldbus. You can use these tabs to modify the I/O mapping of FTB, FTM and OTB Islands. For detailed information, see *I/O Mapping of FTB, FTM and OTB Islands,* *page 182*.

The **PDO Configuration** tab is for configuring the transmission parameters of the PDO items. The **Data Ranges** tab lists mandatory, manufacturer and optional objects, sorted as CANopen objects. If you select **Compact Mapping**, the data items are automatically arranged by the software and the PDOs are filled from first to last.

I/O Image of OTB Ethernet and Modbus NIMs

The following figure shows an example of an OTB Ethernet I/O image overview with the **Data Ranges** tab selected:



The **Registers (read-only)** and **Registers (writeable)** tabs display the read-only and writeable registers. You can use these tabs to modify the I/O mapping of OTB Islands. For detailed information, see *I/O Mapping of FTB, FTM and OTB Islands, page 182*.

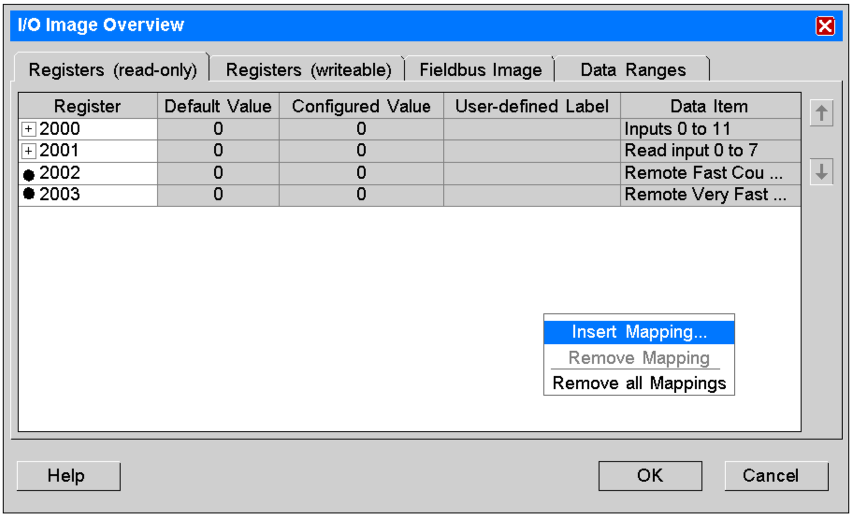
The **Data Ranges** tab contains input, output, parameter and diagnostics registers as well as I/O modules status registers. Further, special function registers (RFC, RVFC, PLS/PWM) are included.

I/O Mapping of FTB, FTM and OTB Islands

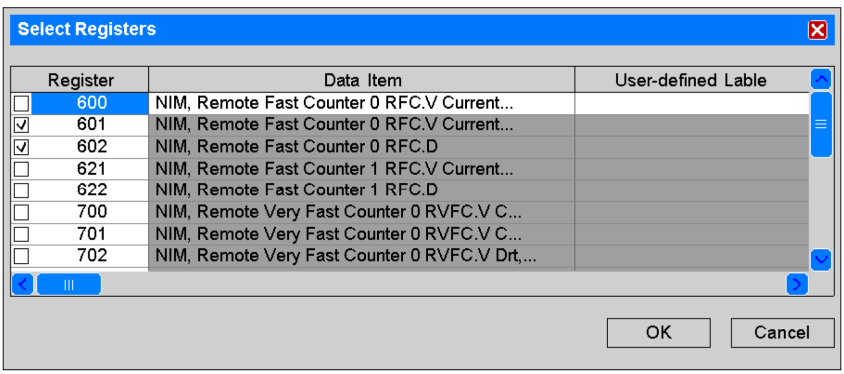
The I/O mapping is performed in the

- **TxPDOs** and **RxPDOs** tabs for FTB, FTM and OTB CANopen NIMs.
- **Registers (read-only)** and **Registers (writeable)** tabs for OTB Ethernet and Modbus NIMs.

To modify the I/O mapping, right-click the tab window and select an option from the shortcut menu that is then displayed as shown in this figure:



If you select **Insert Mapping**, the **Select Registers** dialog box is displayed, providing a list of all mappable data items as shown in this figure. To select an item, click the corresponding check box and then **OK**:



Section 3.10
Export Function

Introduction

This section provides an overview of the **Export** function for symbols, variables, and communication parameters into PLC programs.

What Is in This Section?

This section contains the following topics:

Export Function Basics

Introduction

You have the possibility to create symbols for data items and the hardware configuration (communication parameters) in the Advantys Configuration Software. The **Export** function enables you to provide these symbols together with the related address information to your PLC software.

The Schneider Electric software products can import the information:
- CoDeSys (communication parameters)
- Concept (symbols, communication parameters)
- PL7 (symbols)
- SyCon (communication parameters)
- TwidoSuite (communication parameters)
- Unity Pro (symbols)

NOTE: You can use the DDXML format to extract data from the export file to your specific applications. You can use transformation files or .java script to extract or convert the data.

Export File Formats

These export file formats are available (if not stated otherwise, an *x* means that the format is available for all NIMs which support the fieldbus concerned):

Format	CANopen	DeviceNet	Ethernet	Ethernet/ IP	Fipio	Interbus	Modbus	Modbus Plus	PROFIBUS DP
CSV	x (*)	-	x (*)	-	-	-	x	-	-
DCF	x	-	-	-	-	-	-	-	-
DDXML	x	-	x	-	-	-	x	x	-
EDS	x	x	-	-	-	x	-	-	-
GSD	-	-	-	-	-	-	-	-	x
LIST	-	-	x (*)	-	-	-	x	-	-
MDC	-	-	-	-	-	x	-	-	-
SCY	x	-	x	-	x	x	x	x	x
TXT	-	-	x (**)	-	-	x	-	x	-
XDB	-	-	x (*)	-	-	-	x (*)	-	-
XSY	x	-	x	x	x	x	x (*)	x	x
(*) = not available for STB islands (**) = only available for STB islands									

Additional EDS Export Options for CANopen

If you have selected the EDS export format for a CANopen Island, you have to additionally define for which Schneider Electric software the EDS file shall be exported. Therefore, another dialog box is displayed once you have clicked **OK**. Depending on the module family, you have the following options:

Module Family	EDS Export Options
OTB, FTM, FTB	• **Network Configurator Sycon** • **Network Configurator CodeSys >= V2.0** • **Network Configurator TwidoSoft or TwidoSuite**
STB	• **Network Configurator Sycon or CodeSys >= V2.0** • **Network Configurator TwidoSoft or TwidoSuite**

The import possibilities depend on controller and CANopen configurator:

Controller	CANopen Configurator	EDS	DCF	Comment
ATV71	CoDeSys V2	x	-	check **Send all SDO** check box
Elau	EPAS-4 based on CoDeSys V2	x	-	check **Generate all SDO** check box
Lexium Motion	MotionPro based on CoDeSys V2	x	-	check **Create all SDO** check box
M238	SoMachine based on CoDeSys V3	x	-	check **Create all SDO** check box
Twido	TwidoSoft V3	x	-	-
	TwidoSuite V1/V2	x	-	-
Premium	Sycon	x	-	All SDOs are automatically sent to the device.
M340	Unity 3.0 (fixed catalog of CANopen devices including STB, OTB, FTB)	-	only STB	-
	Unity 4.0 (fixed catalog of CANopen devices including STB, OTB, FTB, FTM plus EDS import via catalog manager)	-	not for FTB	-

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Use the *.xsy* file exported by the Advantys Configuration Software only with Unity for Premium PLCs.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

The CANopen *.xsy* file exported from the Advantys Configuration Software uses flat addressing and will not work for Schneider Electric PLCs that use topological addressing.

General Behavior

The Advantys Configuration Software generates files for the PLC and/or communication software. These files can be imported by the software. You can start the **Export** function under **File → Export <Island name>**.

This table shows the allocation of the export file format to the different available software applications:

Export File Format	Software Application
CSV	TwidoSuite
DCF	Unity Pro
DDXML	general XML format
EDS	SyCon, TwidoSuite, CoDeSys
GSD	SyCon
LIST	TwidoSuite
MDC, TXT	Concept
SCY	PL7
XDB, XSY	Unity Pro

The **Export** function allows you to use the Advantys system information and the variable comments and to create various files (for instance CSV, DCF, DDXML, EDS, GSD, LIST, MDC, SCY, TXT, XDB, and XSY files) for importing device descriptions, device configurations, or symbols (for instance into TwidoSuite, CoDeSys, SyCon, Unity Pro, Concept, and PL7). Using the MDC format, you can dynamically import device descriptions into Concept to be able to use the devices concerned in your application as other devices.

Notes

If the Profibus configuration (GSD format) is exported in form of a compact device, the PBY100 Profibus master of the Premium family will not perform a module alignment and the default I/O image view will not be in line with the data view of the Premium PLC programming tools *(see page 193)*.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Use the default modular system export format with Premium PLC programming tools.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Labels for PLC-to-HMI and HMI-to-PLC data items are not exported when you use a Modbus Plus or an Ethernet NIM. Labels entered in the **Parameters** tab of the Module Editor will not be exported because they are not assigned to I/O data items.

GSD Export

For STB Profibus NIMs V4.*xx* or later, the GSD export function allows the export of channel-level diagnostics information for selected modules. Only Advantys STB digital I/O and analog I/O modules capable of reporting status are able to report channel-related data. For these modules, the following applies:

If the I/O Data Mapping ...	Then the Corresponding Parameter Records ...
is at its default	include channel-level diagnostics information.
has been changed from its default	contain only a single-byte entry with a zero value. **Note:** This also applies to modules not capable of reporting status.

XSY Export Format

The variables with size greater than 16 bits is converted to multiple integers by default and the variable is exported to 32 bit format by choosing the option **4 byte variables**. This data type is supported by Unity Pro V7.0 and above.

DDXML Export Format

The DDXML export file can be used for the transfer of the following data:

- device configuration
- communication parameters
- labels/symbols

You can transform the XML data by using a transformation file or java script.

You can use *.xsl* style sheets directly during the export procedure. In this case, you need to select the *.xsl* file in the **Transformation file** text box of the **Export** dialog box. You can use the ellipsis button to select the style sheet file. Refer to Export *(see page 190)* for a detailed description of the **Export** dialog box elements.

In the **Export** dialog box, you can change the directory for storing and the file name for the export file. By default, the files are stored in the project directory. You will find a folder with the Workspace name in the project directory. The default name of the export file is the name of the Island. The extension of the export file is *.xml*.

NOTE: The export file is following the XML standard and can be edited with every XML or ASCII editor.

XML File Contents

The result of a DDXML export is an extensive XML file, containing information that is grouped as follows:

- device identity
- device manager including the CANopen object list
- device function
- application process including
  - data type list
  - function type list
  - function instance list
  - parameter list
  - module description list

XML File Example

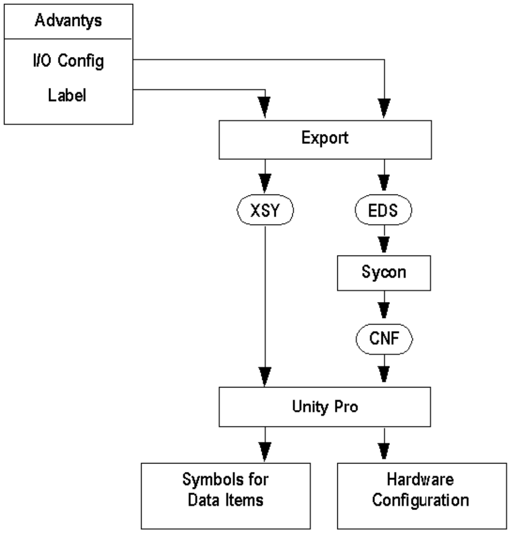
The figure below shows that an extract of a data type list included in the XML file:

```
-<ApplicationProcess>
 -<dataTypeList>
  -<enum name=”Fault Recovery Response” uniqueID=”DT_ENUM_STBDDO3600-
V1.xx_FaultRecoveryResponse_ID”>
    - <enumValue value=”0”
        <label lang=”en-us”>Latched Off</label>
      </enumValue>
    - <enumValue value=”1”
        <label lang=”en-us”>Auto Recovery</label>
      </enumValue>
    <BYTE />
   </enum>
  -<enum name=”BIT Values” uniqueID=”DT_ENUM_BITVALUES_ID”
    - <enumValue value=”0”
        <label lang=”en-us”>0</label>
```

Using the Export Function

In the following example, there will be used 2 export files for different needs: 1 for the labels and 1 for the hardware configuration. You may need to use the export files with different software tools to integrate the information into your PLC software.

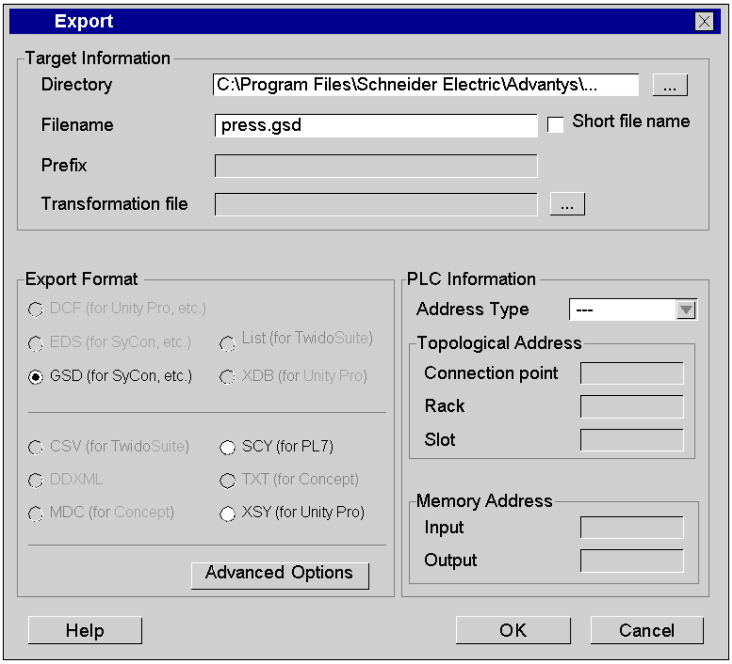
You can use the **Export** function as shown in the figure below (exemplary):



Export Dialog Box

Introduction

This figure shows an example of the **Export** dialog box for an STB Profibus DP NIM:



NOTE: The **Advanced Options** button is not available for every NIM.

Advanced Options

The **Advanced Options** button is enabled if supported by the format currently selected. With these settings, you are able to select how the information is exported. STB configurations can be exported separately for each module or in a compact way for the whole Island. OTB configurations can be exported using subprograms *(see page 194)*.

NOTE: Module alignment is available for STB Profibus DP NIMs only and can be selected in the **I/O Image Overview** dialog box. For the export, selecting **Compact System** disables the module alignment function independent of whether it is selected or not *(see page 193)*.

Target Information

The **Target Information** area contains the following fields:

Field	Comment
Directory	Directory for the export file. The default directory is defined by the current Workspace directory.
Filename	The default file name for the export file is derived from the Island file name. You can edit the file name by clicking the field.
Short file name	If this check box is selected, the associated short file name will be used. It consists of 8 plus 3 characters.
Prefix	This field is available if you select DDXML, SCY, TXT or XSY as export file format. You can enter a string of up to 5 characters, which is used as a prefix for each label in the export file. For an application consisting of multiple identical Islands, you are now able to export the same Island configuration with a different prefix multiple times instead of individual adjusting all labels. **Note:** This option is not available if the software is started from another application such as Unity Pro.
Transformation file	This field is available if you select DDXML as export file format. You can then transform the exported data using a style sheet.

Export Format

In the **Export Format** area, you can select the specific export format you want to generate dependent on the used network interface module (NIM). All possible formats are selectable, the others are not active. The network is automatically specified by the used NIM.

PLC Information

In the **PLC Information** area, you can enter the addresses in the topological or memory address format. Dependent on the fieldbus, the **Memory address** field or **Topological address** field is active. The other one is automatically not available.

Start Address Description

The following table describes the input and output address types for the different selected PLC programming systems and the used fieldbus:

NIM	Unity Pro	Concept	PL7
CANopen	**Memory address mode** input start address: `%MWi` output start address: `%MWj`	not available	**Memory address mode** input start address: `%MWi` output start address: `%MWj`
DeviceNet	not available	not available	not available
Ethernet Modbus TCP	**Memory address mode** input start address: `%MWi` output start address: `%MWj`	**Memory address mode** input start address: (4x reference) output start address: (4x reference)	**Memory address mode** input start address: `%MWi` output start address: `%MWj`
Fipio	**Topological address mode** connection point number rack number of module slot number of module	not available	**Topological address mode** connection point number rack number of module slot number of module
Interbus	**Topological address mode** rack number of module slot number of module input start address: `%IWi` output start address: `%QWj`	**Memory address mode** input start address: (3x reference) output start address: (4x reference)	**Topological address mode** rack number of module slot number of module input start address: `%IWi` output start address: `%QWj`
Modbus Plus	**Memory address mode** input start address: `%MWi` output start address: `%MWj`	**Memory address mode** input start address: (4x reference) output start address: (4x reference)	**Memory address mode** input start address: `%MWi` output start address: `%MWj`
Profibus DP	**Topological address mode** rack number of module slot number of module input start address: `%IWi` output start address: `%QWj`	not available	**Topological address mode** rack number of module slot number of module input start address: `%IWi` output start address: `%QWj`

NOTE: The lowercase letters i and j are used as placeholders for numerical values.

Advanced Options

Introduction

You can set advanced options for the following:
- STB Profibus DP Islands
- OTB Ethernet TCP/IP Modbus and Modbus Serial Line Islands

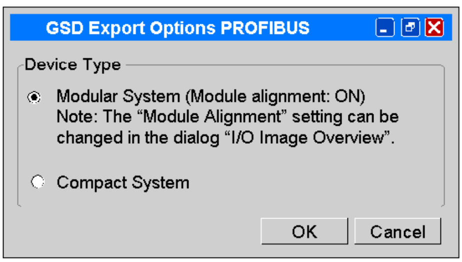
Advanced Options for STB Profibus DP Islands

The following table gives you an overview of the available advanced options:

Option	Description
Modular System	The export file for the specific Island contains every module of the Island as different devices. This is the default setting. **Note:** If you use the module alignment in the **I/O Image Overview** dialog box, you have to use the **Modular System** setting for the export. If you use the **Compact System** setting, the module alignment will be disabled and you will get a notification in the Log Window.
Compact System	The export file for the specific Island contains all information of the whole Island and the different modules as 1 device.

Selecting between a modular or compact export is possible for these export formats:
- GSD (for Sycon, etc.)
- SCY (for PL7)
- XSY (for Unity Pro)

The following figure shows the advanced options dialog box for GSD:



NOTE: The advanced setting affects all exports of the specific fieldbus and export format. This setting affects also older projects or other Islands in your project.
If you change the setting from modular to compact system, this setting is also effective after restarting the Advantys Configuration Software, until you change the setting in the advanced options dialog box again.

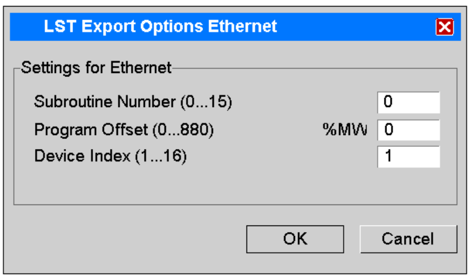
Advanced Options for OTB Ethernet Modbus and Modbus Serial Islands

For importing OTB Ethernet Modbus and Modbus Serial Islands into TwidoSuite, the LST export format is available. This file format contains a subprogram which allows the PLC to send the configuration to the Island. The subprogram is written in IL programming language.

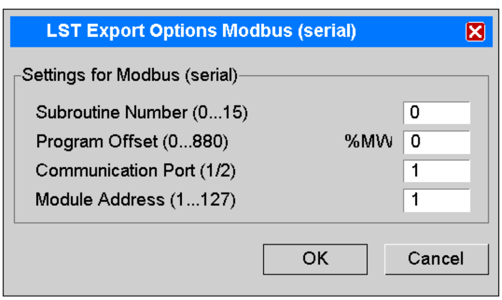
The following table gives you an overview of the available advanced options:

Option	Description
Subroutine Number	The number of the subroutine is used to modify the start sequence of the IL program. The default value is 0.
Program Offset	The first word is used to customize the addresses where the program operates. The default value is 0.
Device Index	This is the index of the distant device. The default value is 1.
Communication Port	This is the number of the serial port. The default value is 1.
Module Address	This is the slave address used to target the messages on the fieldbus. The default value is 1.

This dialog box contains the advanced options for OTB Ethernet Modbus NIMs:



This dialog box contains the advanced options for OTB Modbus Serial NIMs:



Example of Exporting Data

Introduction

The following example provides an introduction of how to export data from the Advantys Configuration Software step by step.

In the example, a Premium PLC is connected to an Advantys STB Island using the Profibus DP fieldbus protocol.

The Island consists of the following modules:
- STBNDP2212 Profibus network interface module (NIM)
- STBPDT3100 power distribution module (PDM)
- STBDDI3420 4-channel digital input module
- STBDDO3410 4-channel digital output module
- STBAVI1270 2-channel analog input module
- STBAVO1250 2-channel analog output module
- STBXMP1100 termination plate

The Island is node number 2 on the Profibus DP fieldbus network.

Step 1

Start the Advantys Configuration Software.

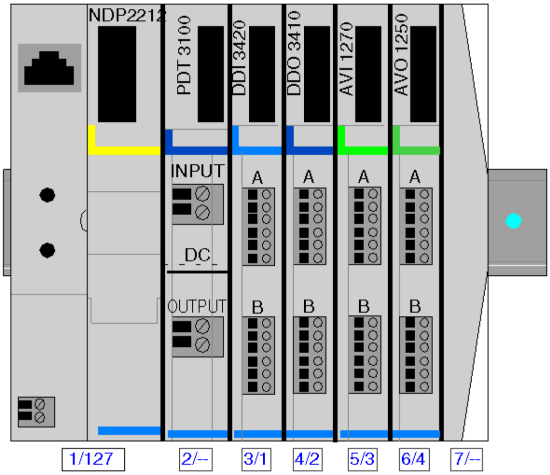
Step 2

Create a new Island in a new Workspace using the following names and path:

Step	Action
1	From the **File** menu, select **New Workspace**. **Result:** The **New Workspace** dialog box is displayed.
2	In the **Name:** field of the **Workspace File** area, type `Quick Start`.
3	In the **Location:** field of the **Workspace File** area, browse to or enter the path *c:\Program Files\Schneider Electric\Advantys\Projects*.
4	In the **Name:** field of the **Island File** area, type `Node_2`.
5	Click **OK**. **Result:** A new Workspace screen is displayed containing the new Island, which is displayed in the Island Editor as an empty DIN rail.

Step 3

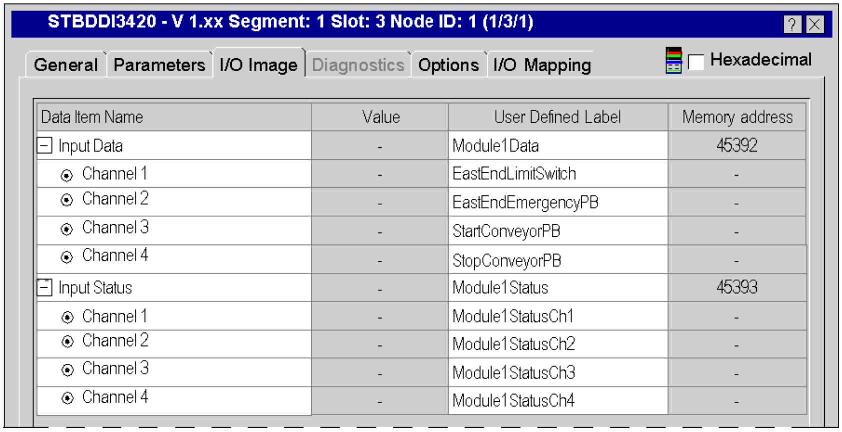
Configure the Island *Node_2* according to its hardware configuration defined above by dragging the modules from the Catalog Browser and placing them in the correct order on the DIN rail:



Step 4

Double-click a module in your configuration to open the Module Editor. Go to the **I/O Image** tab and fill the **User Defined Label** column with the labels/symbols that you want to use on your PLC platform.

The following figure shows the first module in the example configuration (STBDDI3420 4-channel digital input module) properly symbolized:



Conditions

The Advantys Configuration Software exports the symbols entered into the **User Defined Label** column of the Module Editor.

To obtain a reasonable export, at least 1 label for an I/O data item has to be available if you use Modbus Plus and Ethernet NIMs. In case of all other NIMs, automatically assigned labels are sufficient for a successful export.

The following data items have automatically assigned labels:
- run-time parameters
- HMI-to-PLC and PLC-to-HMI data
- status and control words of Interbus NIMs
- data of virtual analog and digital modules

The labels should not be duplicates and they have to be compliant to the IEC61131 rules:
- Only alphanumeric and underscore characters can be used.
- The first character has to be an alphabetic character.
- Blanks and non-ASCII characters are not allowed.
- The overall length of the label should not exceed 24 characters.

NOTE: If any of these rules is ignored, the **Export** function will not execute and will not create an output file.

Step 5

Design the fieldbus/network because some data are needed for the export file. Depending on the fieldbus/network type, use the following tools:

Network/Fieldbus	Tool
• Ethernet • Modbus Plus • Fipio • CANopen • DeviceNet	PLC programming software
Interbus	CMD tool
• CANopen • Profibus	Sycon tool

NOTE: Files have to be created sequentially for Ethernet and Modbus Plus because the base addresses of a subsequent node depend on the data of the previous node.

Step 6

Start the export function selecting **File** → **Export Node_2**.

Result: The **Export** dialog box will be displayed.

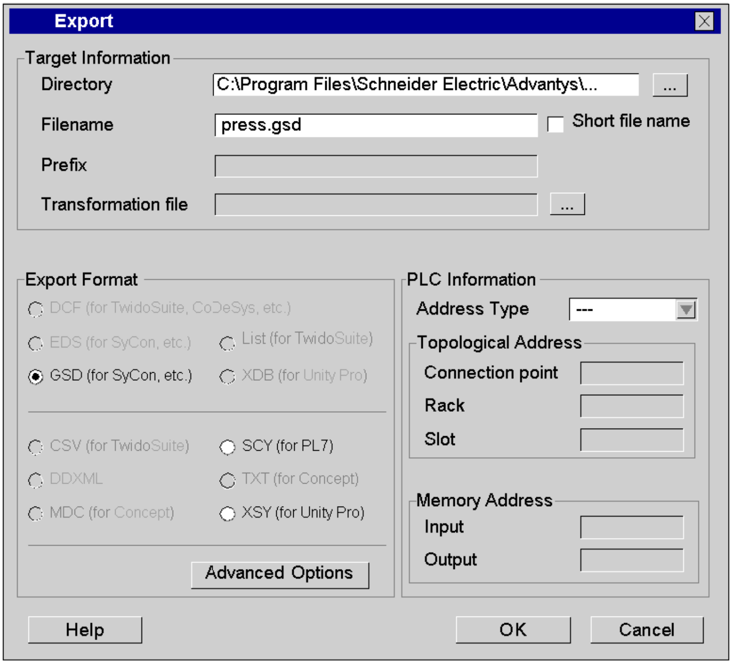
NOTE: The settings are automatically adjusted to the selected Island.

Step 7

Enter the export information into the dialog box.

First, select the export format (for instance GSD) because it influences the options offered in the **Target Information** and **PLC Information** areas.

The following figure shows the export form for the example configuration, filled with all necessary information:

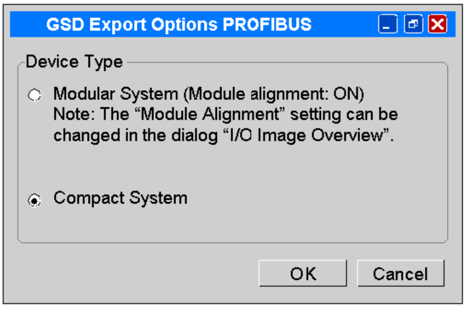


NOTE: Which fields and buttons are available differs for every NIM.

Step 8

Click the **Advanced Options** button to check which setting is selected.

The following figure shows the advanced options dialog box for the GSD export:



The dialog box displays the setting last selected for this fieldbus and this export format, independent of the fact for which Island in your project the setting has been selected *(see page 193)*.

Step 9

Change the setting if desired and click **OK**.

Result: The advanced options dialog box is closed.

Step 10

In the **Export** dialog box, click **OK** to start the export.

Result: A file of the selected type will be generated.

NOTE: If an STB Island contains a Profibus DP NIM V4.*xx* or later, the export file includes channel-level diagnostics information for modules capable of reporting status.

Import

General Import

Import the DCF, EDS, or GSD file into the PLC programming software as required.

Import into Unity Pro

Open the variable editor and import the XDB or XSY file generated by the Advantys Configuration Software and enter the network parameters (Ethernet, Modbus Plus, etc.) corresponding to the node inside the PLC programming software. Refer to the table below. For a successful import, check that the information you have entered in the Advantys Configuration Software and the configuration in Unity Pro is consistent.

The XDB file, used for the export of OTB Ethernet and Modbus Islands, contains the configuration for the `OTB_LoadConf` DFB. The input parameters of the DFB are the following:

- address
- rack
- module
- channel

The output parameters of the DFB are the following:

- error
- end

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION Use the *.xsy* file exported by the Advantys Configuration Software only with Unity for Premium PLCs. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

The CANopen *.xsy* file exported from the Advantys Configuration Software uses flat addressing and will not work for Schneider Electric PLCs that use topological addressing.

Import into Concept

Import the MDC and/or TXT file generated by the Advantys Configuration Software. Choose Program: IEC text as the **Select Source File Type**. Select **Convert to FDB/SFC** in the conversion options.

Enter the network parameters (Ethernet, Modbus Plus) corresponding to the node inside the PLC programming software. Refer to the table below.

For a successful import, check that the information you entered in the Advantys Configuration Software and the configuration in Concept is consistent.

NOTE: The import functionality works only if the configuration supports IEC functionality.

Import into PL7

Open the variable table and import the SCY file generated by the Advantys Configuration Software. Enter the network parameters (Ethernet, Modbus Plus, etc.) corresponding to the node inside the PLC programming software. Refer to the table below.

For a successful import, check that the information you enter in the Advantys Configuration Software and the configuration in PL7 is consistent.

Import into SyCon

In case the Island exported contains an STB Profibus DP NIM V4.*xx* or later, the export file includes channel-level diagnostics information for modules capable of reporting status. For a successful import of a compact mode GSD into SyCon, delete the entry for compact mode in Sycon and then re-enter it.

Import into TwidoSuite

Import the CSV, DCF or LIST file generated by the Advantys Configuration Software using the **Import** option available on the functions bar. Enter the network parameters (Ethernet, Modbus Plus, etc.) corresponding to the node inside the PLC programming software.

For a successful import, check that the information you enter in the Advantys Configuration Software and the configuration in TwidoSuite is consistent.

Starting Address Description

The following table describes the Information that has to be entered in the fieldbus configuration editor of the PLC programming tool after the generation of the export file:

	Unity Pro	Concept	PL7
CANopen	read length read ref slave (always 5391) write length write ref slave (always 0)	not available	read length read ref slave (always 5391) write length write ref slave (always 0)
DeviceNet	not available	not available	not available
Ethernet	read length read ref slave (always 5391) write length write ref slave (always 0)	read length read ref slave (always 45392) write length write ref slave (always 40001)	read length read ref slave (always 5391) write length write ref slave (always 0)
Fipio	nothing	not available	nothing
Interbus	nothing	read length read ref slave (always 30001) write length write ref slave (always 40001)	nothing

	Unity Pro	Concept	PL7
Modbus Plus	read length read ref slave (always 5391) write length write ref slave (always 0)	read length read ref slave (always 45392) write length write ref slave (always 40001)	read length read ref slave (always 5391) write length write ref slave (always 0)
Profibus DP	nothing	not available	nothing

Use with Unity Pro

Topological Addresses for CANopen, Ethernet and Modbus Plus Network

The input and output topological addresses are
- `%MWi` for word and `%MWi.n` for the bits and
- where `i` is the word address and `n` represents the bit location in the word. Enter the starting input address and starting output address.

Topological Addresses for Fipio Fieldbus

The input and output topological addresses are
- `%IW\2.y\0.0.0.w` for word and `%IW\2.y\0.0.0.w.n` for the bits.
- `%QW\2.y\0.0.0.w` for word and `%QW\2.y\0.0.0.w.n` for the bits.

The parameters represent the following:

Parameter	Description
y	connection point number in PL7 (1 to 127)
w	rank of the word (0 to 31)
n	bit location in the word

Topological Addresses for Interbus and Profibus DP Fieldbus

The input and output topological addresses are
- `%IWr.s.0.w` for word and `%IW\r.s.0. w.n` for the bits.
- `%QWr.s.0.w` for word and `%QW\r.s.0. w.n` for the bits.

The parameters represent the following:

Parameter	Description
r	rack number containing the Interbus master TSX IBY 100
s	slot number containing the Interbus master TSX IBY 100
w	rank of the word
n	bit location in the word

Use with Concept

Topological Addresses for Ethernet and Modbus Plus Network

The input and output topological addresses are

- `4YYYY` for word (4x reference) and
- where YYYY is a number from 0001 to 9999.

The Advantys Configuration Software generates an export file, which can be imported into Concept (Program: IEC Text as the source variable type). It creates an FBD or ST section code with the Advantys configuration file name as the name of the section. The program section has been automatically generated with `Word_to_Bit` and `Bit_to_Word` functions. All the bit-level symbols are configured as memory variables.

Topological Addresses for Interbus Fieldbus

The input topological addresses are

- `3YYYY` for word (3x reference) and
- where YYYY is a number from 0001 to 9999.

The output topological addresses are

- `4YYYY` for word (4x reference) and
- where YYYY is a number from 0001 to 9999.

The Advantys Configuration Software generates an export file, which can be imported into Concept (Program: IEC Text as the source variable type). It creates an FBD or ST section code with the Advantys configuration file name as the name of the section. The program section has been automatically generated with `Word_to_Bit` and `Bit_to_Word` functions. All the bit-level symbols are configured as memory variables.

Use with PL7

Topological Addresses for CANopen, Ethernet and Modbus Plus Network

The input and output topological addresses are

- `%MWi` for word and `%MWi.n` for the bits and
- where `i` is the word address and `n` represents the bit location in the word. The user has to enter the starting input address and starting output address.

Topological Addresses for Fipio Fieldbus

The input and output topological addresses are

- `%IW\x.2.y\0.0.w` for word and `%IW\x.2.y\0.0.w:Xn` for the bits.
- `%QW\x.2. y\0.0.w` for word and `%QW\x.2.y\0.0.w:Xn` for the bits.

The parameters represent the following:

Parameter	Description
x	Fipio processor's slot number in PL7 (0 or 15)
y	connection point number in PL7 (1 to 127)
w	rank of the word (0 to 31)
n	bit location in the word

Topological Addresses for Interbus and Profibus DP Fieldbus

The input and output topological addresses are

- `%IWrs.0.w` for word and `%IW\rs.0.w:Xn` for the bits.
- `%QWrs.0.w` for word and `%QW\rs.0.w:Xn` for the bits.

The parameters represent the following:

Parameter	Description
r	rack number containing the Interbus master TSX IBY 100
s	slot number containing the Interbus master TSX IBY 100
rs	calculated using the formula (100 x r)+s
w	rank of the word
n	bit location in the word

Use with TwidoSuite

Topological Addresses for CANopen, Ethernet and Modbus Network

The input and output topological addresses are

- `%MWi` for word and `%MWi:Xk` for the bits and
- where `i` is the word address and `k` represents the bit location in the word. The user has to enter the starting input address and starting output address.

Modifications of the Island Configuration

Introduction

A modification of the Island configuration may have an impact on the I/O data layout and consequently on the assignments referred to by the PLC program. The procedures described above have to be repeated to update the PLC application. Please bear in mind the behavior of the PLC programming platform.

Unity Pro Application

When importing the file in Unity Pro, differences are displayed and you have the choice to change. In order to update the application correctly, select **replace all** in the wizard. During the import, the addresses are updated in the Data Editor. Because the labels/symbols in the application are used as references, the program does not change and is up to date.

Concept Application

To perform the application update correctly, check the settings **allow modification of existing sections** and **allow modification of existing variables** in the Concept import utility. The import replaces the existing symbols with the new symbols and the program is automatically updated.

PL7 Application

In PL7, the references in the program are always the addresses. An import of new symbols does not update the mapping of the addresses. Consult your PL7 programming documentation for how to update your address mapping.

Section 3.11
Reflex Editor

Introduction

This section provides an overview of the Reflex Editor.

What Is in This Section?

This section contains the following topics:

Reflex Action Types

Introduction

Reflex actions are small routines that perform dedicated logical functions directly on the Island bus. They allow output modules on the Island to act on data and drive field actuators directly, without requiring the intervention of the fieldbus master.

The Island bus optimizes the reflex response time by assigning the highest transmission priority to its reflex actions. Reflex actions take some of the processing workload off the fieldbus master, and they offer a faster, more efficient system bandwidth.

The Reflex Editor is available for STB modules only.

Available Reflex Action Types

Before adding reflex actions, you should be familiar with the different types of the reflex actions and their functionality.

The following reflex action types are available:
- boolean logic *(see page 338)*
- integer compare *(see page 352)*
- unsigned compare *(see page 369)*
- counter *(see page 388)*
- timer *(see page 401)*
- analog latch *(see page 422)*
- digital latch *(see page 437)*

Working with the Reflex Editor

Introduction

The Reflex Editor offers the following functions:

- *Adding a Reflex Action, page 212*
- *Modifying a Reflex Action, page 212*
- *Deleting a Reflex Action, page 213*

Opening the Reflex Editor

Proceed as follows to open the Reflex Editor:

Step	Action
1	On the **Island** menu, click **Reflex Editor** or click the following icon on the **Island** toolbar: 
2	Add, modify, or delete the reflex actions.

NOTE: You can access the Reflex Editor only when the Island is offline and unlocked. By default, the Reflex Editor is displayed with only the **New** and **Close** command buttons enabled. All other buttons are disabled.

Indicating a Reflex Module

In the Island Editor, modules for which a reflex action has been configured are marked in the graphical representation with the following reflex icon:

Output data which are controlled by a reflex action are marked with a yellow background in the Island's I/O image overview and in the module's I/O image and also in the **I/O Animation** dialog box.

Adding a Reflex Action

Proceed as follows to add a reflex action:

Step	Action
1	Open the Reflex Editor.
2	Click **New**. **Note:** You can access the Reflex Editor only when the Island is offline and unlocked. By default, the Reflex Editor is displayed with only the **New** and **Close** command buttons enabled. All other buttons are disabled.
3	From the **Action group:** list, select the reflex action group.
4	From the **Action type:** list, select the reflex action type.
5	From the **Action module:** list, select the reflex action module.
6	Go to the second pane of the **Reflex Editor** dialog box, and configure all appropriate parameters.
7	Click **OK** to validate the changes.

NOTE: The **Action no.:** field is automatically updated by the Reflex Editor. The numbering is adjusted to the slot number and the output channel number of the action module. The reflex action of the action module with the lowest slot number will receive action number 1 even if you configure it after other reflex actions. If more than 1 action is defined for an action module, the action with the lower output channel number will receive the lower action number.

Modifying a Reflex Action

Proceed as follows to modify a reflex action:

Step	Action
1	Open the Reflex Editor.
2	Double-click the reflex action in the list you wish to modify. **Note:** The **Modify** button is enabled when you double-click the corresponding row of the reflex action.
3	Click **Modify**.
4	Modify the parameters in the Reflex Editor.
5	Click **OK** to accept the changes. **Result:** The parameters you have modified will be reflected in the Reflex Editor.
6	Click **Close** to close the **Reflex Editor** dialog box.

Deleting a Reflex Action

Proceed as follows to delete a reflex action:

Step	Action
1	Open the Reflex Editor.
2	Double-click the reflex action in the list you wish to delete. **Note:** The **Delete** button is enabled when you double-click the corresponding row of the reflex action.
3	Click **Delete**. **Result:** A message box is displayed for confirmation.
4	Click **Yes**. **Result:** The action you have deleted no longer appears in the Reflex Editor.
5	Click **Close** to close the **Reflex Editor** dialog box.

Nested Reflex Actions

The Advantys Configuration Software allows you to create 1 level of nesting for reflex actions. You can nest 2 reflex blocks, where the output from the first block is used as an operational input to the second block. Both reflex blocks should be nested within the same action module. Proceed as follows to configure a nested reflex action:

Step	Action
1	For the first reflex action, select **None** in the channel field of the **Physical output:** list.
2	For the second reflex action, now select the same action module. As module for the input, select **Nested**.
3	For the **Channel**, select the output signal configured in step 1 from the list.

For further information on nested reflex actions, please refer to *Nesting 2 Reflex Blocks, page 214*.

Displaying Labels

If you have entered names for modules and I/O data elements (channels), those labels are displayed in 2 different ways:

Object	Way of Displaying Object
Module	The name is displayed in all fields and lists where the module appears.
Channel	The name is only displayed as tooltip and only for the channels already entered. To see the tooltips, click **Modify** for the selected reflex action and move the mouse cursor onto the entered channel data.

Nesting 2 Reflex Blocks

Summary

The Advantys Configuration Software allows you to create 1 level of nesting for reflex actions. You can nest 2 reflex blocks, where the output from the first block is used as an operational input to the second block. Both reflex blocks should be nested within the same action module.

Action Module

In a nested reflex action, the output from the first reflex block is used internally as an operational input to the second reflex block. The output from the second reflex block is used to update the physical output channel of the action module.

When you nest a pair of reflex blocks, you need to map the outputs from both to the same action module. Choose the action module type that is appropriate for the output from the second nested action block. In some cases, this means that you may need to choose an action module for the first block that does not seem to be appropriate for its output.

For example, say you want to nest a counter-compare action. To do this, you need to configure 2 action blocks using the Reflex Editor. The first block belongs to the action group counter action *(see page 388)*, and the second block to the action group unsigned compare action *(see page 369)*.

The output from a counter is always a 16-bit word value, and the output from the unsigned compare is always a binary (boolean) value. Intuitively, you might assume that because the counter produces a word as its output it should be mapped to an analog action module. However, since the counter is the first block in the nested action and since the output from the second action, the unsigned compare, is a boolean, you need to select a digital output module as the action module.

Physical Outputs

The Reflex Editor requires that you specify the physical and logical output of each reflex block that you configure. Generally, the physical output is the channel on the action module to which the output of the action will be written. The physical output is always mapped this way when an action is not part of a nesting; it is also how the output from the second action block in a nested action is mapped. For the first block in a nested action, however, the physical output is sent to a temporary memory buffer. Instead of specifying an output channel on the action module, you need to specify the physical output as **None**.

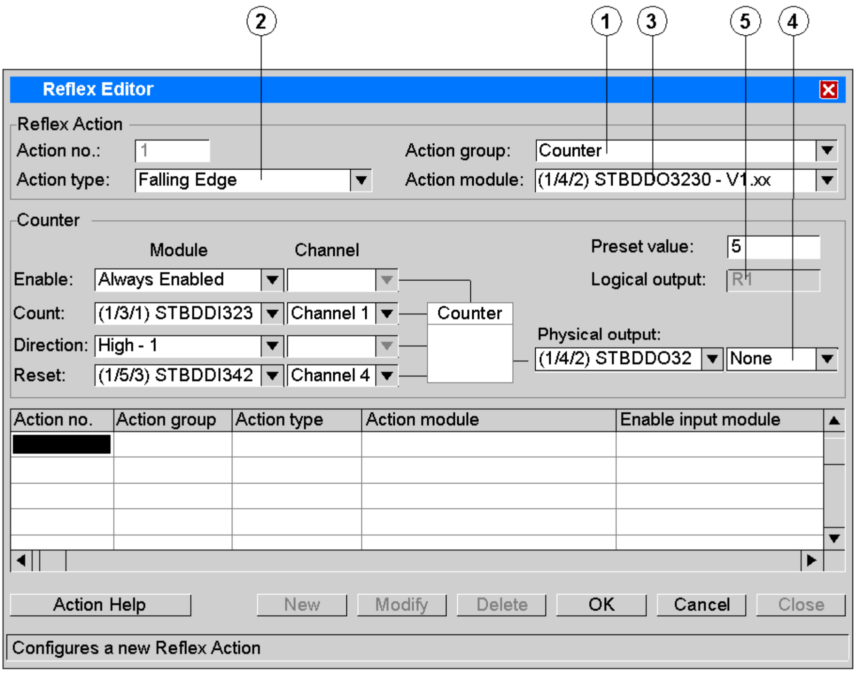
Logical Outputs

The output from each block also needs to be assigned a logical output. The logical output is a tag name for the output – a text string between 1 and 8 characters long. The characters may be any combination of standard keyboard characters – alphanumerics, underscores, and/or standard symbols (!,?, /, >, etc.).

The logical output can be particularly useful in a nested reflex because the text string of the first action block will appear in the drop-down menu as an input to the second action block.

Counter-Compare Configuration

To clarify the process of configuring a nested action, let us look at the way you might configure the first of the 2 action blocks in the Reflex Editor of the Advantys Configuration Software:

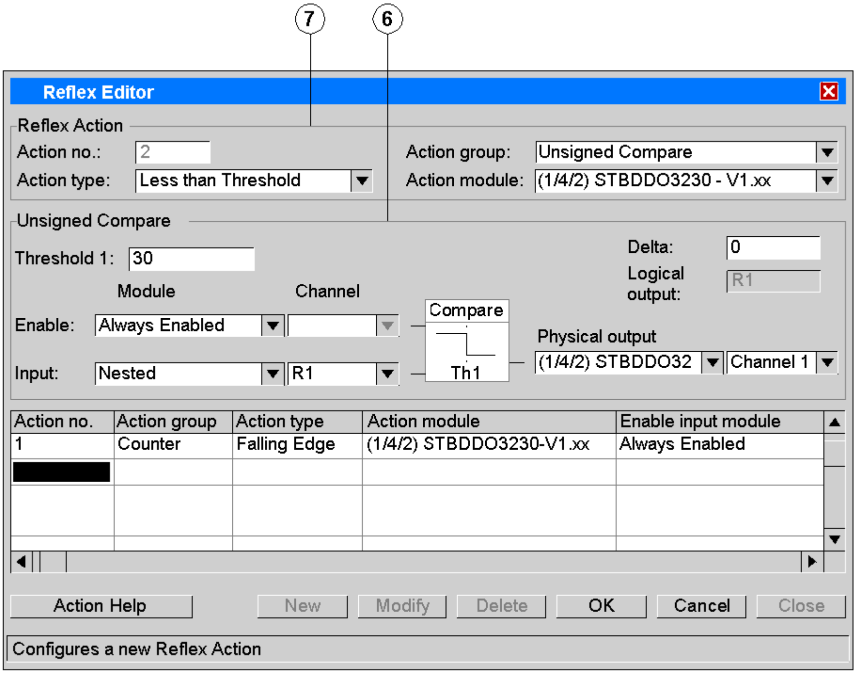


1+2 Action No 1 = Falling-Edge Counter
3 Action Module = STBDDO3230 Digital Output Module
4 Physical Output Channel of the Action Module = None
5 Logical Output = R1

Action number 1 is a falling-edge counter, as the items 1 and 2 above indicate. The action module is the STBDDO3230 digital output module at Island bus address 2 (item 3 above). The action module needs to be a digital output module because the ultimate result of the nested action will be boolean. The action module selected is automatically displayed in the **Physical output:** field.

Item 4 above shows the output channel on the action module as **None**. The output from the falling-edge counter is sent to a temporary memory buffer. The string **R1** (in this example) is automatically assigned to the output value in this temporary memory buffer, as shown in the **Logical output:** field (item 5 above).

The logical output from the first block will be used as the operational input to the second block, as shown in the following illustration:



6 Action No 2 = Unsigned Less than Threshold Compare Block
7 Action Module = STBDDO3230 Digital Output Module

Action number 2 is an unsigned less-than-threshold compare block. Item 6 (**Input** row, **Channel** list) shows that the operational input to the compare block is **R1**, the logical output from action number 1. The action module (item 7 above) for the less-than-threshold compare block is the STBDDO3230 digital output module at Island bus address 2, which the same action module as the one for the falling-edge counter.

Chapter 4
Creating an Island Bus Configuration

Overview

This chapter describes how a logical Island configuration can be created in an active Workspace.

What Is in This Chapter?

This chapter contains the following sections:

Section 4.1
Basic Island Configuration

Introduction

This section describes the steps necessary to configure an Island.

What Is in This Section?

This section contains the following topics:

Creating a Workspace

Introduction

Before you can create a *.isl* file for a logical Island, you need to open an existing Workspace or create a new one. In the Advantys Configuration Software, an Island can exist only inside a Workspace.

The first time you start the configuration software, first create a Workspace. When you create the Workspace, a new Island will be created inside the Island. You can add additional Islands to the Workspace. A Workspace can contain up to 10 Islands.

Creating a Workspace

Create a new Workspace as follows:

Step	Action	Result
1	On the **File** menu, select a **New Workspace**.	The **New Workspace** dialog box appears.
2	In the **Workspace File** field of the dialog box, enter a name for the Workspace.	A Workspace name can be up to 50 characters long and can comprise alphanumerics plus SPACE, MINUS SIGN, and UNDERSCORE keyboard characters.
3	In the **Island File** field of the dialog box, enter a name for the Island.	An Island file name can be up to 24 characters long and can comprise alpha-numerics, spaces and other keyboard characters.
4	Click **OK**.	A new Workspace screen appears showing the new Island. All that appears in the Island Editor as an empty DIN rail. **Note:** The Island type is not defined until you have selected a NIM and thus a product family. A Workspace can include Islands of different product families.

STB Basic NIMs

Introduction

Besides standard and premium NIMs, the STB catalog of the Advantys Configuration Software contains the following basic NIMs:

- STB NCO 1010 (NIM supporting CANopen)
- STB NDP 1010 (NIM supporting Profibus DP)
- STB NDN 1010 (NIM supporting DeviceNet
- STB NIB 1010 (NIM supporting Interbus)

In contrast to standard NIMs, which support configuration download as well as auto-configuration, basic NIMs only support auto-configuration. That means it is neither necessary nor possible to configure these NIMs manually.

As with standard NIM Islands, you can export the configuration file of Islands containing basic NIMs.

Basic NIMs support up to six extension segments with a maximum number of 12 STB I/O modules.

Disabled Software Features of the Basic NIM

If the active Island contains a basic NIM, the following module/family entries are not displayed in the Catalog Browser:

- CANopen extenders in the accessories family
- preferred modules
- enhanced CANopen modules

If the active Island contains a basic NIM, the following menu entries are disabled:

- on the **Island** menu
  - **Reflex Editor** entry
  - **Baud Rate Tuning** entry
  - **Temperature Range** entry
  - **Test Mode Settings** entry
- on the **Online** menu
  - all menu entries

Accordingly, the respective buttons on the toolbar are disabled.

If the active Island contains a basic NIM, the following tabs are disabled in the Module Editor:

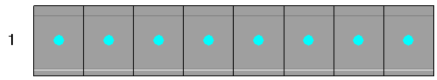
- **Parameters** tab
- **Diagnostics** tab
- **Options** tab
- **I/O Mapping** tab (if applicable to the module)

Rails

Introduction

In the Advantys Configuration Software, a segment is referred to as a **Rail**. Each segment in a logical Island appears on its own rail.

An empty rail appears in the Island Editor as soon as a new Island has been created:



This rail will support the Advantys modules in the primary segment of the new Island bus configuration. All the modules in the primary segment of the Island (the NIM, PDMs, AUX power supply, I/O modules, extension modules or termination plate) will be inserted on this default rail.

Deleting and Adding the Primary Rail

If you delete the primary rail from the Island Editor and then want to replace it, use the **Add Rail** command from the **Island** menu. You should have the primary rail to configure a logical Island.

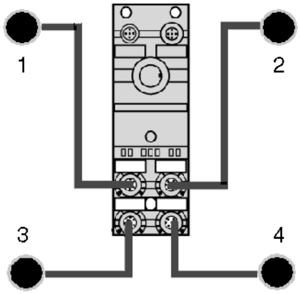
Adding More Rails

STB and FTM Islands can be extended beyond the primary segment. For STB Islands, the following applies:

- The maximum number of rails in an Island configuration is 7, 1 for the primary segment and up to 6 for extension segments.
- Preferred modules *(see page 227)* and CANopen devices *(see page 228)* do not appear on separate rails in the Island Editor. They appear beside or below the rail from which they are extended.

Exception

An FTM Island can consist of 4 segments. All of them are connected the NIM. As soon as you have inserted an FTM NIM into a newly created Island, the NIM and the segments will be displayed according to their star topology as follows:



Adding Modules to an Island Segment

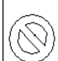
Introduction

There are 3 ways to add modules to a rail:
- using the drag-and-drop function
- double-clicking the module
- selecting the module and pressing ENTER

If you try to place a module on the rail into an invalid location, a notification appears and the software does not allow the module to be dropped into that location.

Drag-and-Drop Method

Proceed as follows to add a module to a rail using the drag-and-drop function:

Step	Action	Result
1	Select the module name in the Catalog Browser *(see page 57)*.	The module name is highlighted.
2	Drag the module to the desired location on the rail in the Island Editor.	As the module is dragged across the Workspace, the following icon is displayed:  When the module crosses over the rail, 1 of the following icons appear: •  indicates a valid position •  indicates an invalid position
3	Release the mouse button on a valid location.	A graphical version of the module drops into the location on the rail.

Double-Click Method

The double-click method is the quickest way to add a module to the configuration:

If you want to ...	Then ...	Result
add a module to the end of the last rail	double-click the module name in the Catalog Browser.	A graphical version of the module will appear at the end of the rail.
place a module between 2 modules that are already on the Island	select the leftmost of the 2 existing modules in the Island Editor, and then double-click the new module name in the Catalog Browser.	A graphical version of the new module will appear between the 2 existing modules on the rail.

ENTER Key Method

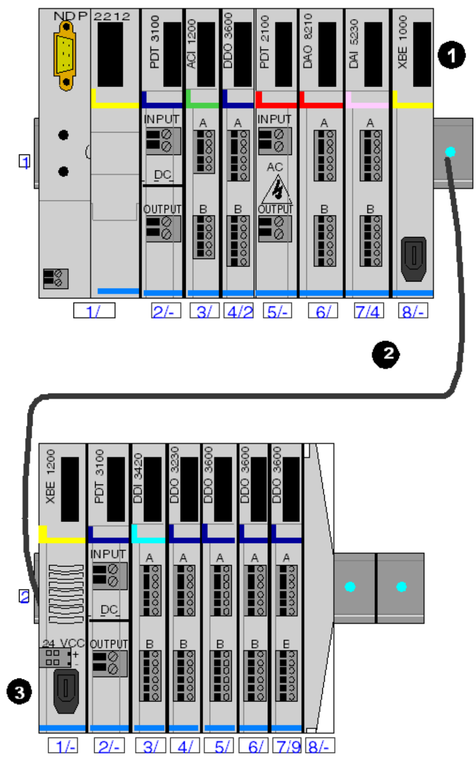
The ENTER key method is similar to the double-click method:

If you want to ...	Then ...	Result
add a module to the end of the last rail	double-click the module name in the Catalog Browser, and then press ENTER.	A graphical version of the module will appear at the end of the rail.
place a module between 2 modules that are already on the Island	select the leftmost of the 2 existing modules in the Island Editor, double-click the new module name in the Catalog Browser, and then press ENTER.	A graphical version of the new module will appear between the 2 existing modules on the rail.

Adding Extension Rails to the Island Configuration

Procedure for STB Islands

Proceed as follows to extend an STB Island configuration by adding rails:

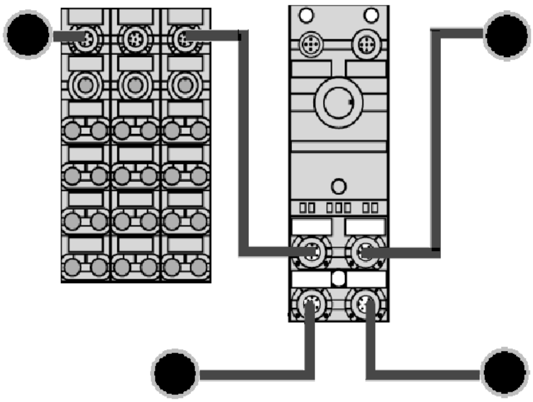
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>If there is a termination plate at the end of the last existing rail, remove it. Also remove the XBE2100 module, if there is one, and the modules attached to it.</td></tr>
<tr><td>2</td><td>Pick an EOS module from the Catalog Browser and drop it into the Island Editor at the end of the last rail.</td></tr>
<tr><td>3</td><td>Double-click a BOS module in the Catalog Browser.
Result: A new rail appears in the Island Editor with the BOS module as the first module. An extension cable connects the EOS and the BOS module:


1 EOS Module
2 Extension Cable
3 BOS Module</td></tr>
<tr><td>4</td><td>Pick a PDM from the Catalog Browser and drop it next to the BOS module.</td></tr>
</table>

Step	Action
5	Add the desired I/O and PDM modules from the Catalog Browser.
6	Add a termination plate or another extension module at the end of the new rail.

Procedure for FTM Islands

As soon as you have inserted an FTM NIM into a newly created Island, the NIM and all segments will be displayed according to their star topology as shown below. You can place modules without having the follow a certain segment order:



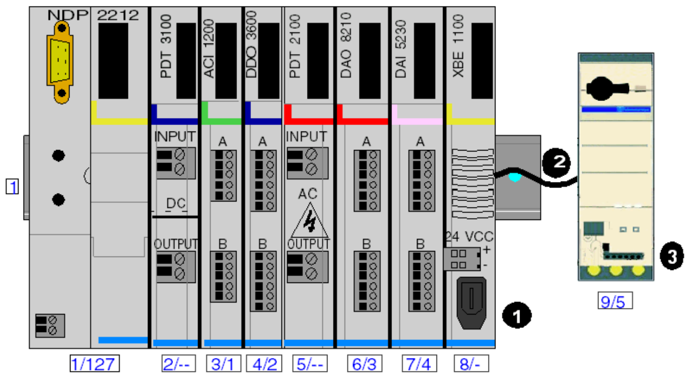
Proceed as follows to extend an FTM Island configuration:

Step	Action
1	If the only module on the segment to which you want to add a module is a compact one, remove it.
2	Select the segment or, if present, the module to which you want to add a new module. **Note:** The module should be an extensible module.
2	Select the new module from the Catalog Browser and double-click it. **Note:** The new module should be an extensible module. **Result:** The new module is added and, if present, right to the module selected before.
3	If you have removed a compact module before, replace it as last module of the segment.

Extending the Configuration to a Preferred Module

Procedure

Proceed as follows to extend an Island configuration from a rail to a preferred module:

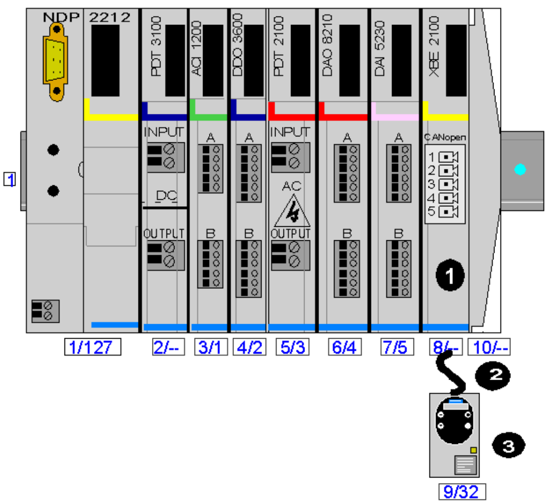
Step	Action
1	If there is a termination plate at the end of the last existing rail in the Island Editor, remove it.
2	Pick the STB XBE 1100 EOS module from the Catalog Browser and drop it into the Island Editor at the end of the rail.
3	Double-click a preferred module in the Catalog Browser. **Result:** The preferred module appears in the Island Editor beside the rail. An extension cable connects the EOS module and the preferred module:  1 EOS Module 2 Extension Cable 3 Preferred Module
4	If you want to add another preferred module, repeat step 3. **Result:** Each additional module is placed to the right of the previous module, with a cable connection between them. If you want to extend to a new Advantys I/O rail, go to step 5. If the preferred module shall be the last module on the Island, go to step 6.
5	Double-click the STB XBE 1300 BOS module in the Catalog Browser. **Result:** A new rail will appear below the existing one. The BOS module is the first module on the new rail. An extension cable connects the preferred module and the BOS module *(see page 225)*.
6	Add a TeSys U LU9 RFL15 termination device.

Extending the Configuration to Enhanced CANopen Devices

Procedure

An Advantys STB Island does not auto-address enhanced CANopen devices. Install them as the last devices on your Island and set their Island addresses manually on the devices. Install all your auto-addressable modules first.

Proceed as follows to extend the configuration to enhanced CANopen devices:

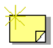
Step	Action
1a	If there is a terminator at the end of the last rail, pick a CANopen extension module from the Catalog Browser and drop it in front of the terminator.
1b	If a terminator is missing at the end of the last rail, pick a CANopen extension module from the Catalog Browser and drop it into the last position on the rail. Then, pick a termination plate from the Catalog Browser and drop it at the end.
2	Double-click a CANopen device in the Catalog Browser. **Result:** The device appears in the Island Editor below the CANopen extension module and off the rail, connected by an extension cable:  **1** CANopen Extension Module **2** Extension Cable **3** CANopen Device
3	If you want to add another CANopen device, repeat step 2. **Result:** Each additional device is placed to the right of the previous device and is connected by a CANopen extension cable.
4	Apply 120 Ω termination to the CANopen device. **Note:** There is no graphical element in the software to indicate termination on an enhanced CANopen device. Provide this termination on the physical device.

Adding and Deleting Annotations

Adding Annotations

Text comments can be placed in the Island Editor with the annotation feature. There are 3 ways to annotate a logical Island:

- clicking the following button on the **Island** toolbar:



- right-clicking a location in the Island Editor, and then selecting **Add Annotation** from the menu
- selecting **Add Annotation** from the **Island** menu

Resizing the Annotation Box

The **Annotation** box can be resized to accommodate any amount of text as follows:

Step	Action
1	Select the **Annotation** box. **Result:** When it is selected, handles appear on the corners and sides of the box.
2	Position the mouse cursor over a handle until the cursor changes to the following icon: 
3	Drag the handle until you achieve the desired size.

Moving the Annotation Box

An **Annotation** box can be moved anywhere within the Island Editor as follows.

Step	Action
1	Resize the **Annotation** box slightly, as described above. (The color of the handles on the box should be green.)
2	Drag the selected box to the desired location in the Island Editor.

Deleting Text from an Annotation Box

Delete text from an **Annotation** box as follows:

Step	Action
1	Double-click the **Annotation** box.
2	Select the text to be deleted.
3	Press ENTER or DELETE.

Retrieving Text for an Annotation Box

You can retrieve text that has just been deleted from an **Annotation** box as follows:

Step	Action
1	Click the empty **Annotation** box.
2	Click the following button: 

Deleting an Annotation Box

You can delete an annotation box along and its contents as follows:

Step	Action
1	Select the **Annotation** box.
2	Delete the box by performing 1 of the following commands: ● On the **Island** menu, click **Delete Annotation**. ● Right-click the annotation box, and then select **Delete** from the shortcut menu. ● Press DELETE.

Replacing NIMs

Introduction

You have the possibility to exchange the NIM type or the NIM version in an existing Island configuration by using the replace NIM feature. This feature is available for STB and OTB Islands.

Replacing the NIM

To replace the NIM type or the NIM version in an existing Island, the Island concerned has to be offline, open and unlocked. Proceed as follows:

Step	Action
1	Select the NIM module. **Result:** The NIM module is selected.
2	• Either click the **Island** menu and select **Replace NIM** • or right-click the NIM and select **Replace NIM** from the shortcut menu. **Result:** A selection dialog is displayed.
3	Depending on the NIM type currently used in the Island concerned, select from the list • either a NIM of the same type • or a higher-featured NIM type. **Note:** It is not possible to downgrade from a standard to a basic NIM because all configuration data would be lost.
4	Click **OK**. **Result:** A confirmation message is displayed.
5	Confirm your selection by clicking **Yes**. **Result:** The NIM is replaced, the Island configuration kept, and the Island concerned displayed with the new NIM.

NOTE: Configuration settings not related to the NIM module are preserved. HMI settings and Virtual Placeholder settings are lost if the target NIM type does not support them. This is indicated by a notification. Modified parameters may not be supported by the new NIM or may exceed the limits specific to it. In these cases, notifications are displayed in the Log Window.

Island Migration

Introduction

When a project file is opened, the Advantys Configuration Software automatically checks and detects if the project was created using

- an older tool version, resulting in an out-of-date project file format or
- an older catalog, resulting in out-of-date module data or
- the Advantys Configuration Tool.

In any of these cases, a project update including an Island migration is performed.

Island Migration

During migration,

- the Island database is read,
- the Island is re-created in memory using the appropriate module entries from the catalog,
- all modified parameter settings are applied,
- default values are used where out-of-range conditions are met,
- the old project file is saved as *.bak* file with the original project file name, and
- a new project file, an *.isl* file, is created with the original project file name.

After a successful migration, the new *.isl* project file is opened automatically.

NOTE: Your old project file is preserved and renamed with a BAK extension. After a successful migration of *Project1.isl* for instance, the Workspace contains the files *Project1.isl* (migrated file) and *Project1.bak* (original file).
Rename the *.bak* file as soon as possible if you want to keep the original project file. The *.bak* file will be overwritten if the Island is migrated again.

In case of a detected error, a diagnostic message is displayed, prompting you to confirm by clicking **OK**. Contact Schneider Electric and provide information on the type of detected error.

Migration of Advantys Configuration Tool Projects

Projects created using the Advantys Configuration Tool may contain parameters which value exceed the valid range or which associated parameter object may not exist in the module description given in the catalog. In these cases, the software displays a notification but continues the migration process.

The software may encounter labels/symbols that exceed the limit of 24 characters. These labels/symbols are transformed as follows:

1. The first 20 characters of the original label are extracted.
2. An underscore and a 3-digit number based on a global counter value are appended.
3. The global counter value is incremented. If the counter exceeds the maximum value of 999, the migration is not successful and a diagnostic message is displayed.

Offline Protection

Introduction

Whenever you open an existing Island (*.isl*) file, it comes up locked. You can monitor it on the Workspace screen, but you cannot edit it. Editing is possible only when the file is unlocked. Optionally, you can apply a password to the offline lock. If you do this, you will not be able to unlock the file without first entering the password.

Applying and Changing Passwords

To apply a password to the lock on a new *.isl* file or to change an existing one, perform the following steps:

Step	Action
1	While the new *.isl* file is active in the Workspace, click the following icon:  **Result:** A message is displayed asking you if you want to set a password.
2	Click **Yes**. **Result:** The **Set Password** dialog box is displayed.
3	In the **New password:** field, ● type a password if you want to apply a password to the lock. ● type a new password if you want to change an existing one. **Note:** Valid characters for the password are alphanumeric characters.
4	Retype the password in the **Confirm:** field. **Note:** If the text strings do not match, you are prompted to restart the process.
5	Click **OK**. **Result:** You are prompted to save the file with the new password.
6	Click **OK**. **Result:** The password just set is now enabled and the Island is locked. Users will be required to enter it to unlock the Island configuration. Any existing old password no longer applies.

Unlocking

To unlock a password protected *.isl* file, proceed as follows:

Step	Action
1	While the *.isl* file is active in the Workspace, click the following icon: 
2	Enter the password and click **OK**.

Online Protection

Introduction

An online protection capability is available to stop unauthorized changes or overwrites to the configuration data in the physical Island. When online protection is enabled, the RST button on the NIM is disabled and data on the removable memory card is ignored. This feature is available for STB Islands only.

You will be asked to apply a password to the online protection. When a password is applied, you need to know the password in order to remove the protection feature or get into (or out of) temporary test mode.

NOTE: Ensure to record the password. If you forget the password, you cannot use the RST button to reset the default configuration parameters or the removable memory card to load a new configuration. Also, you cannot change modes on the physical Island (test mode/run mode) without the password while online protection is enabled.

Online Protection Feature

The protection feature is available only in online mode (when the active *.isl* file in the Advantys Configuration Software is connected to a physical Island).

To enable online protection for the physical Island, perform the following steps:

Step	Action
1	On the **Online** menu, click **Protect**.
2	Enter a password.
3	Click **OK**.

The **Protect** command enables and disables the feature. When protection is applied, a check mark appears in the box next to the command in the menu.

Password

The password has to be an alphanumeric string between 0 and 6 characters long. An empty password is valid.

When protection is activated, you will be queried for a password if you try to execute a command online. If you are not using a password, simply click **OK** when the **Set Password** dialog box appears.

Unprotecting

To disable online protection, click **Online** → **Protect** again to disable the feature. When online protection is not applied, there is no check mark next to the command on the menu.

Section 4.2
Virtual Placeholders

Introduction

This section describes the use and configuration of Virtual Placeholders.

What Is in This Section?

This section contains the following topics:

Virtual Placeholders

Introduction

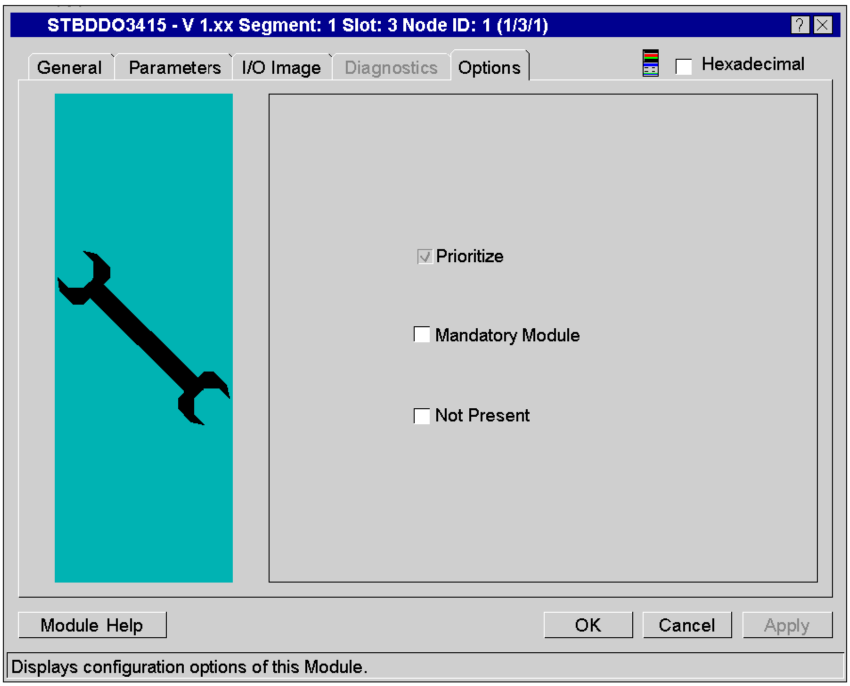
The Virtual Placeholder feature allows you to remove certain physical STB modules from a base configuration, while keeping the process image identical. Thus, you can create systems which have various options removed without changing the controlling PLC program.

The logical view of the Island, which reflects the I/O map for the user program, remains unchanged, whereas the physical view, which reflects the physically present STB modules, may change.

If you have removed some STB modules, the remaining ones will be plugged physically next to each other as no spare slots are allowed in an Island configuration.

Configuring Virtual Placeholders

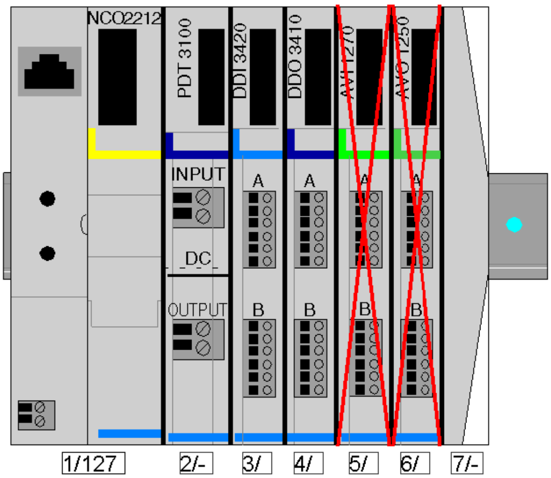
To configure a module as Virtual Placeholder, just check the **Not Present** box from the **Options** tab of the Module Editor:



Display of Virtual Placeholders

In the Module Editor, the Virtual Placeholder modules are marked with crossed red lines.

The following figure shows an example of an Advantys STB configuration with the 2 analog modules AVI 1270 and AVO 1250 configured as Virtual Placeholders:



NOTE: If signals from a Virtual Placeholder module, which is physically not present, are used as input data for a reflex action, **all** reflex actions in that module will not function.

I/O Image Overview and Animation

For modules set to **Not Present** in the Island configuration, the associated data items in the **I/O Image** and the **I/O Overview** are displayed with a gray background. This applies to both offline and online mode.

Resource Analysis for Virtual Placeholders

On modules set to **Not Present** in the Island configuration, the **Resource Analysis** function is disabled.

Remote Configuration of Virtual Placeholders

Introduction

With the standard **Virtual Placeholder** option, build an Island configuration for each version of the different options at configuration time. With a CANopen NIM of the STB product family, it is possible to build 1 single base-configuration and download it into the NIM.

The different options are then selected during runtime by downloading a Virtual Placeholder object through the CANopen fieldbus. For information about how to implement remote configuration of Virtual Placeholders in your controller application, please refer to Virtual Placeholders in the *NCO2212 User Manual*.

NOTE: The remote configuration of the **Virtual Placeholder** function is only available for the CANopen NIM STBNCO2212 - V3.*xx*.

NOTE: The remote configuration of the **Virtual Placeholder** function is only available for modules that support auto-addressing.

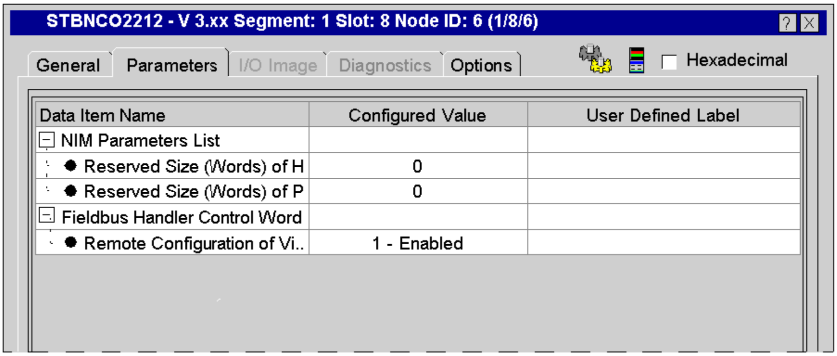
Enabling Remote Configuration of Virtual Placeholders

Proceed as follows to enable the remote configuration of Virtual Placeholders:

Step	Action
1	Double-Click the NIM to open the Module Editor.
2	On the **Parameters** tab, expand the parameter **Fieldbus Handler Control Word**.
3	Set the bit **Remote Configuration of Virtual Placeholders** to **1 - Enabled**. **Note:** If the remote configuration of Virtual Placeholders has been enabled, it is not possible to configure static Virtual Placeholders in the **I/O Module Editor** dialog box, as the **Not Present** check box is disabled. Any Virtual Placeholder settings defined before will be reset if you activate the remote configuration of Virtual Placeholders. In online mode, the bit is read-only. It can thus be checked but not changed by the configuration software. The setting of the bit is loaded into the project database after each build and is restored when the configuration software is opened.

Display in the Parameters Tab

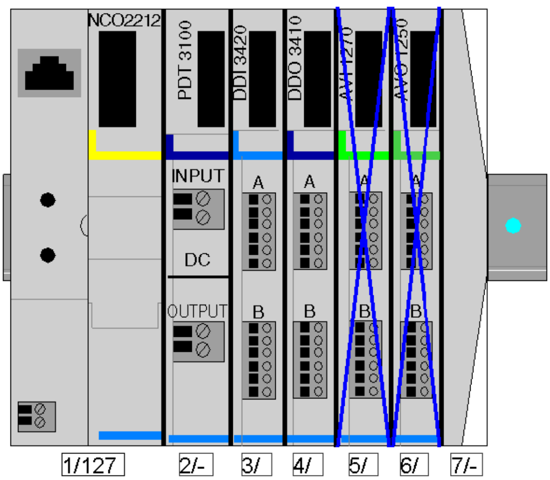
The following figure shows the information displayed on the **Parameters** tab:



Online Representation of Remote Configured Virtual Placeholders

In online mode, the actual Virtual Placeholder configuration is shown as it has been defined by the upstream fieldbus master. To distinguish this dynamic configuration from a static configuration where the Virtual Placeholders have been defined by the configuration software, the modules which are not present are marked with crossed blue lines instead of red ones.

The following figure shows the online view of an Advantys STB configuration with the 2 analog modules AVI 1270 and AVO 1250 configured as remote Virtual Placeholders by the fieldbus master:



Offline Representation of Remote Configured Virtual Placeholders

If you disconnect from the Island, all modules shown as **Not Present** will be reset to the **Present** state in the Module Editor so as to ensure that the base configuration is displayed in offline mode according to the configuration in the project database.

If the fieldbus master changes the Virtual Placeholder settings while the configuration tool is online, the Island will automatically be disconnected to avoid inconsistent visualization and diagnostics handling. An open Module Editor or an open I/O Image Overview/Animation will be closed immediately and a message will be displayed.

NOTE: A subsequent re-connect does not require a re-build and download because the base configuration has not been changed.

I/O Image Overview and Animation

If the remote configuration of Virtual Placeholders is enabled, the following display convention is used for both I/O image overview and I/O image animation:

Mode	Description
Offline Mode	All data items are in the standard way.
Online Mode	The actual Virtual Placeholder configuration is indicated by a gray background of the associated data item cells.

Resource Analysis for Virtual Placeholders

If the remote configuration of Virtual Placeholders is enabled, the **Resource Analysis** function displays different information for offline and online mode:

Mode	Description
Offline Mode	The **Resource Analysis** function displays available resources considering all modules of the base configuration because the project data contains no information about the dynamic settings defined by the fieldbus master.
Online Mode	The **Resource Analysis** function displays available resources taking into account dynamically or statically configured Virtual Placeholders.

Chapter 5
Configuration Software Structure

Introduction

This chapter provides a description of the basic user interface functions like navigation and working with menu items. The available menus and all contained items are presented as a reference.

What Is in This Chapter?

This chapter contains the following sections:

Section 5.1
User Interface

Introduction

This section describes the Advantys Configuration Software user interface structure. It provides an overview of the elements of the graphical user interface and how they are used.

What Is in This Section?

This section contains the following topics:

Windows Conventions

Introduction

There are 3 standard Windows conventions available for moving objects in the Advantys Configuration Software Workspace:

- drag-and-drop
- dock
- float

Drag-and-Drop

You can drag modules from the Catalog Browser and drop them in the Island Editor. Drag-and-drop a module as follows:

Step	Action	Indication
1	Select a desired module in the Catalog Browser.	-
2	Drag the selected module toward the DIN rail in the Island Editor.	The cursor should look like this: 
3	Drag the module into position on the DIN rail. **NOTE:** Annotation boxes can also be moved in the Island Editor using the drag-and-drop function.	The cursor should appear, indicating that the module location is valid:  If the cursor appears, the module location is invalid:  This means there is a violation of the Island's connectivity rules.

Dock

A standard Windows docking operation allows you to drag a window from its original position in the Workspace to any corner of the application area. Docking can be applied to the Workspace Browser, the Catalog Browser and the toolbars.

Float

A standard Windows float operation allows you to drag the window from its original position in the Workspace to any part of the application area. Floating can be applied to the Workspace Browser, the Catalog Browser and the toolbars.

Menus

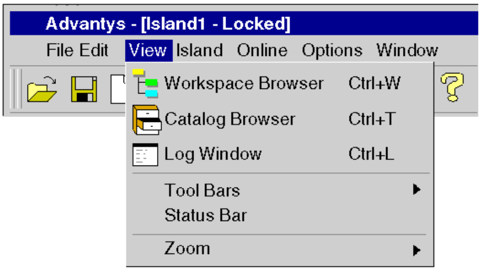
Introduction

There are 3 types of menus:
- main menus
- submenus
- shortcut menus

Main Menus

The titles of the individual menus are displayed on the menu bar. The individual menu commands are listed on the menus. A menu is opened by left-clicking the title of the menu or by pressing ALT+SELECTED LETTER.

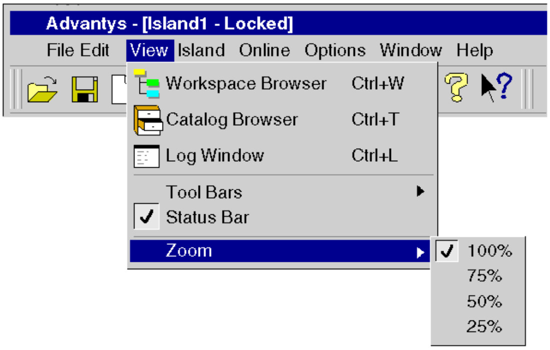
The following figure shows the menu bar with a menu:



Submenus

The title of a submenu is a menu command of the menu above it. The individual submenu commands are listed on the submenu. Menu commands which contain a submenu can be recognized by an arrow icon.

The following figure shows a submenu:

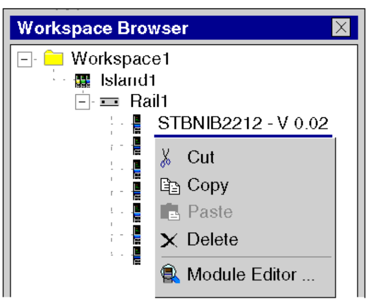


Shortcut Menus

Shortcut menus are menus which contain menu commands specific to the selected object. You can open a shortcut menu by clicking the object (right mouse button), selecting the object and confirming with SHIFT+F10, or pressing the context sensitive key.

Shortcut menus can also be invoked if several objects are selected. If this is the case, the menu only contains the menu commands which are valid for all objects.

The following figure shows an object with a shortcut menu:

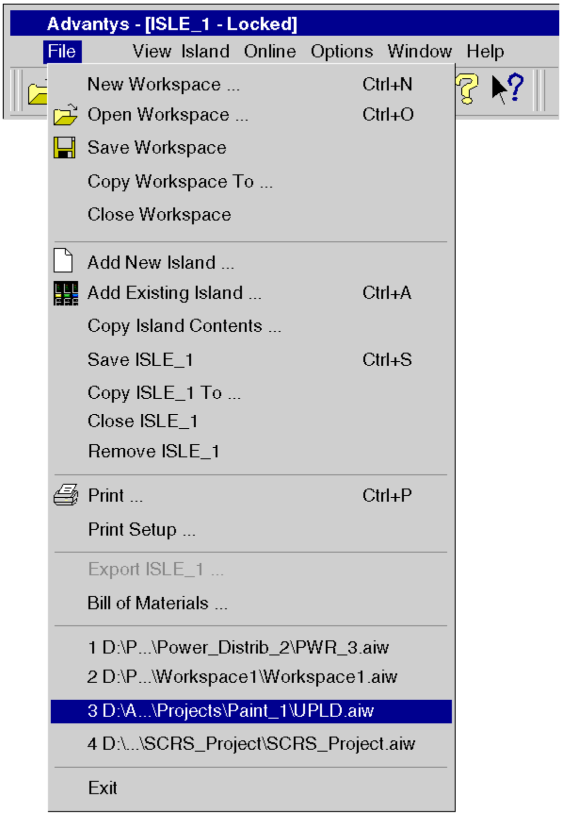


Menu Commands

Introduction

Menu commands are used to execute commands or to call dialog boxes.

The figure below shows an example of a menu with menu commands:



Keyboard Shortcuts or Mnemonics

Keyboard shortcuts (underlined letters) in menu commands allow you to select menu commands using the keyboard. A main menu (menu title) and subsequently a menu command can be selected by holding down ALT and simultaneously entering the underlined letter in the menu title and then the underlined letter of the menu command.

For example, you want to use the **File** menu **Save Workspace** menu command, press the ALT+F to open the menu, followed by ALT+V to execute the menu command.

Grayed Out Menu Command

If a menu command is not available, it is grayed out. Before the desired menu command can be executed, 1 or more other commands have to be executed.

Decimals (...) after the Menu Command

On execution of this menu command, a dialog box is opened with options which have to be selected before execution.

Check Mark before the Menu Command

The menu command is active. If the menu command is selected, the check mark disappears and the menu command is inactive. The check mark is mostly used to identify active modes (for instance that the Island is protected and so on).

Shortcut Keys

Shortcut keys (for instance F1) or key combinations (for instance CTRL+N) after the menu command are a shortcut way for executing the menu command. You can select the menu command using this shortcut key or key combination without having to open the menu. For example, CTRL+S can be pressed to perform the **Save** menu command.

Keyboard Navigation

Introduction

The following tables describe the possibility to access certain functions directly using keyboard shortcuts.

Globally Valid Key Combinations

The following shortcuts are available throughout the Advantys Configuration Software:

Function	Key Combination
Open Workspace	CTRL+O
Add Existing Island	CTRL+A
Save Island	CTRL+S
Print	CTRL+P
Undo	CTRL+Z
Redo	CTRL+Y
Cut	CTRL+X SHIFT+DEL
Copy	CTRL+C SHIFT+INS
Paste	CTRL+V CTRL+INS
Delete	DEL
Workspace Browser	CTRL+W
Catalog Browser	CTRL+T
Log Window	CTRL+L
Delete Annotation	CTRL+D
Help	F1
What' This?	SHIFT+F1
Change between windows of the Advantys Configuration Software, for instance from the Island Editor to the Catalog Browser	CTRL+F6 SHIFT+F6
Open the shortcut or context menu	SHIFT+F10 right mouse key application key

Key Combinations Inside the Browsers

The following shortcuts are available in the Workspace Browser and the Catalog Browser:

Function	Key Combination
Expand/collapse the object trees	SPACEBAR
Move down in the object tree and expand collapsed elements	RIGHT ARROW
Move up in the object tree and collapse expanded elements	LEFT ARROW
Move up/down in the object tree	UP/DOWN ARROW

Key Combinations Inside the Module Editor

The following shortcuts are available in the Module Editor:

Function	Key Combination
Expand an expandable tree node	PLUS SIGN
Collapse an collapsible tree node	MINUS SIGN
Expand/collpase an expandable/collapsible tree node	SPACEBAR
Select the cell on the left	LEFT ARROW
Select the cell on the right	RIGHT ARROW
Select the cell above	UP ARROW
Select the cell below	DOWN ARROW
Select the first row in the tree list control	PAGE UP HOME
Select the last row in the tree list control	PAGE DOWN END
Starts editing an editable	F2
Expand the complete tree list	F3
Collapse the complete tree list	F4

Key Combinations Inside the Island Editor

The following shortcuts are available in the Island Editor:

Function	Key Combination
Change between windows of different Islands	CTRL+TAB CTRL+F6
Select the next module on the right	RIGHT ARROW
Select the previous module on the left	LEFT ARROW
Change the status of a selected annotation box from editable to changes accepted	ESC
Select the next object (segment, NIM, annotation) in the Island Editor	TAB
Start the Module Editor	RETURN
Select the previous segment	SHIFT+TAB
Select the first module in the segment	HOME
Select the last module in the segment	END

Toolbars

Introduction

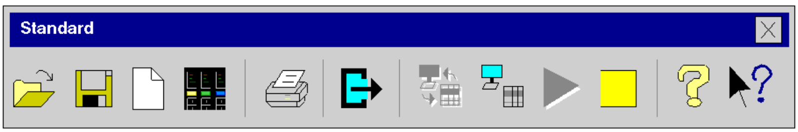
Toolbars display a collection of easy-to-use button images and/or menus that initiate different operations in the Advantys Configuration Software. Several icons are docked in form of independent toolbars on top of the screen.

The Advantys Configuration Software provides 4 toolbars:

- *Standard, page 254*
- *Edit, page 255*
- *View, page 256*
- *Island, page 257*

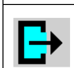
Standard

The **Standard** toolbar comprises 12 icon buttons:



You can move, dock, and/or float this toolbar to other locations in the Workspace. You may also hide the toolbar by right-clicking it and clearing the **Standard** option.

The **Standard** icon buttons perform the following tasks:

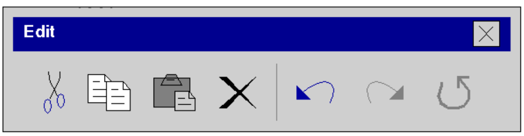
Icon	Task
	invokes a dialog box where you can open an existing Workspace
	invokes a dialog box where you can save the currently open Workspace and/or Islands
	invokes a dialog box where you can create a new Island in the open Workspace
	invokes a dialog box where you can add an existing Island to the open Workspace
	invokes a dialog box where you can select any of the items to be printed
	invokes a dialog box where you can select the format and file for the export

Icon	Task
	invokes a dialog box where you can connect the software to the physical Island
	disconnects the software from the physical Island
	in online mode: invokes a dialog box where you can put the physical Island in run mode
	in online mode: invokes a dialog box where you can stop the physical Island
	invokes a table of contents for the Advantys help system
	turns the cursor into a question mark so that you can invoke **What's This?** help on a selected item in the Workspace

Back to top *(see page 254)*

Edit

The **Edit** toolbar comprises 7 icon buttons:



You can move, dock, and/or float this toolbar to other locations in the Workspace. You may also hide the toolbar by right-clicking it and clearing the **Edit** option.

The **Edit** icon buttons perform the following tasks:

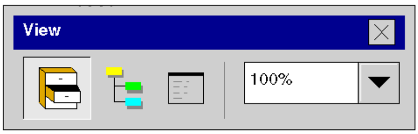
Icon	Task
	cuts the selected item
	copies the selected item
	pastes the copied item
	deletes the selected item

Icon	Task
	undoes the previous action
	redoes the previous **Undo**
	reverts to the previous saved state

Back to top *(see page 254)*

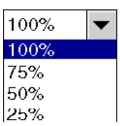
View

The **View** toolbar comprises 3 icon buttons and a list:



You can move, dock, and/or float this toolbar to other locations in the Workspace. You may also hide the toolbar by right-clicking it and clearing the **View** option.

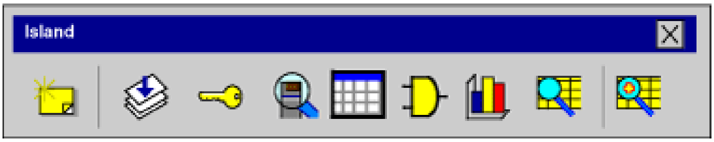
The **View** icon buttons perform the following tasks:

Icon	Task
	hides or shows the Catalog Browser
	hides or shows the Workspace Browser
	hides or shows the Log Window
	zooms the size of the Island Editor to 25%, 50%, 75% or 100% of the default view

Back to top *(see page 254)*

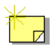
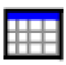
Island

The **Island** toolbar comprises 9 icon buttons:



You can move, dock, and/or float this toolbar to other locations in the Workspace. You may also hide the toolbar by right-clicking it and clearing the **Island** option.

The **Island** icon buttons perform the following tasks:

Icon	Task
	places an empty text window in the Island Editor where you can add an annotation
	initiates the build process for the selected Island configuration All edits to the configuration has to be saved before the build process can start.
	locks or unlocks the selected Island configuration and also gives you a sequence of prompts that enables you to assign or change a password
	opens the Module Editor for a selected I/O in the Island Editor
	opens the **User Defined Label Editor**
	opens the Reflex Editor
	displays the **Resource Analysis** dialog box
	opens the **I/O Image Overview** dialog box
	in online mode: opens the **I/O Image Animation** dialog box

Back to top *(see page 254)*

Section 5.2
Menu Structure

Introduction

This section describes the Advantys Configuration Software menu structure. It provides an overview of the available menu commands and how they are used.

What Is in This Section?

This section contains the following topics:

File Menu

Introduction

The **File** menu refers to the Island currently open, active and displayed in the Island Editor. It contains the following items:

- New Workspace *(see page 259)*
- Open Workspace *(see page 260)*
- Save Workspace *(see page 260)*
- Copy Workspace To *(see page 261)*
- Close Workspace *(see page 261)*
- Add New Island *(see page 262)*
- Add Existing Island *(see page 262)*
- Copy Island Contents *(see page 263)*
- Save <Island> *(see page 263)*
- Copy <Island> To *(see page 263)*
- Close <Island *(see page 264)*
- Remove <Island> *(see page 264)*
- Print *(see page 265)*
- Print Setup *(see page 266)*
- Export <Island> *(see page 267)*
- Bill of Materials *(see page 268)*
- Recent files list *(see page 268)*
- Exit *(see page 268)*

New Workspace

Before creating a new Workspace, check that the existing Workspace is closed. To create a new Workspace, perform the following steps:

Step	Action
1	From the **File** menu, select **New Workspace**.
2	In the **New Workspace** dialog box, type the name of the Workspace in the **Name:** field of the **Workspace File** area.
3	In the **New Workspace** dialog box, type the path for the Workspace in the **Location:** field of the **Workspace File** area.
4	In the **New Workspace** dialog box, type the name of the Island in the **Name:** field of the **Island File** area. **Note:** When you create a new Workspace, you automatically create a new Island.
5	Click **OK**.

Only in a Workspace, you are allowed to create and configure an Island.

Back to top *(see page 259)*

Open Workspace

Before opening an existing Workspace, close the Workspace that is currently open. To open an existing Workspace, perform the following steps:

Step	Action
1	Click either **Open Workspace** in the **File** menu or the following icon on the **Standard** toolbar: 
2	In the **Open Workspace** dialog box, ● either select the Workspace you need to open ● or type the Workspace name in the **File name:** field.
3	Click **Open**.

Back to top *(see page 259)*

Save Workspace

To save the open Workspace, perform the following steps:

Step	Action
1	Click either **Save Workspace** in the **File** menu or the following icon on the **Standard** toolbar: 
2	In the **Save Files** dialog box, select the names of 1 or more Islands contained in the Workspace that you want to save.
3	Click **OK**.

Back to top *(see page 259)*

Copy Workspace To

Use this option to save the existing Workspace under a different name. To copy a Workspace, perform the following steps:

Step	Action
1	From the **File** menu, select **Copy Workspace To**. **Result:** The **Copy Workspace To** dialog box is displayed.
2	In the **Name:** field, enter a new name for the Workspace.
3	In the **Location:** field, enter the path where you want to save the copy.
4	Click **OK**.

Back to top *(see page 259)*

Close Workspace

Only 1 Workspace may be open at any given time. Before opening another Workspace, close the Workspace that is currently open.

To close a Workspace, perform the following steps:

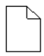
Step	Action
1	From the **File** menu, select **Close Workspace**.
2	If you have not saved the Workspace and Islands you are working on, the **Save Files** dialog box is displayed.
3	Click **Yes** if you want to save the Workspace and Islands.

Back to top *(see page 259)*

Add New Island

A Workspace may contain up to 10 Islands.

To add a new Island, perform the following steps:

Step	Action
1	To add a new Island, ● either select **File** → **Add New Island** ● or right-click the Workspace folder and select **Add Island** → **Add New Island** ● or click the following icon on the **Standard** toolbar: 
2	In the **New Island** dialog box, type the name of the Island in the **Name:** field of the **Island File** area.
3	Click **OK**.

The Advantys Configuration Software always creates a new Island unlocked and appends it to the bottom of the hierarchical tree in the Workspace Browser.

Back to top *(see page 259)*

Add Existing Island

Use the **Add Existing Island** command to replicate an already existing Island from a different Workspace and place it in the currently active Workspace. This option frees you from having to reconfigure an Island with the same configuration as an existing Island.

NOTE: You are not permitted to replicate an existing Island in the Workspace Browser where the original Island resides.

To add an existing Island to the active Workspace, perform the following steps:

Step	Action
1	To add an existing Island, ● either select **File** → **Add Existing Island** ● or right-click the Workspace folder in the Workspace Browser and select **Add Island** → **Add Existing Island** ● or select the following icon on the **Standard** toolbar: 
2	In the **Open Island** dialog box, select the name of an Island file from another Workspace.
3	Click **Open**.

Back to top *(see page 259)*

Copy Island Contents

Use the **Copy Island Contents** command to copy the contents and settings of an existing Island, even from another project, into the Island currently open in the Island Editor. Thus, you have extended possibilities when in restricted mode, where the software is allowed to manipulate only 1 Island.

To copy Island contents, perform the following steps:

Step	Action
1	From the **File** menu, select **Copy Island Contents**.
2	In the **Copy Island Contents** dialog box, select the Island which contents you want to copy.
3	Click **OK**.

Back to top *(see page 259)*

Save <Island>

Use the **Save <Island>** command to save the Island currently open in the Island Editor. The **Save <Island>** command automatically displays the name of the selected Island, for instance **Save My_Island**.

To save the selected Island, select **Save <Island>** from the **File** menu (where <Island> contains the name of the selected Island).

Back to top *(see page 259)*

Copy <Island> To

Use the **Copy <Island> To** command to save the Island currently open in the Island Editor to a new file. This file is not added to any Workspace. The **Copy <Island> To** command automatically displays the name of the selected Island, for instance **Copy My_Island To**.

To copy the selected Island, perform the following steps:

Step	Action
1	From the **File** menu, select **Copy <Island> To** (where <Island> contains the name of the selected Island).
2	In the **Copy <Island>.isl To** dialog box, select the path and the name under which the Island shall be stored.
3	Click **Save**.

Back to top *(see page 259)*

Close <Island>

Use the **Close <Island>** command to close the Island that is currently active in the Island Editor. The **Close <Island>** command automatically displays the name of the selected Island, for instance **Close My_Island**. This functionality is available only in offline mode.

To close the selected Island, select **Close <Island>** from the **File** menu (where <Island> contains the name of the selected Island).

If you have made any changes to the active Island since the last save, the Advantys Configuration Software prompts you to save the changes. To validate the changes you have made, click **Yes**. If you prefer to cancel the changes just made, click **No**.

Back to top *(see page 259)*

Remove <Island>

Because the Workspace can hold a maximum of 10 Islands, you may need to remove an Island from the Workspace in order to make room for a new one.

Use the **Remove <Island>** command to remove the Island currently open in the Island Editor. The **Remove <Island>** command automatically displays the name of the selected Island, for instance **Remove My_Island**. This functionality is available only in offline mode.

To remove the selected Island, perform the following steps:

Step	Action
1	Select the Island you want to remove by clicking in the Island Editor on the tab of the Island to be removed.
2	Remove the Island concerned by ● either selecting **File** → **Remove <Island>** (where <Island> contains the name of the selected Island) ● or right-click the Island concerned in the Workspace Browser and select **Remove**.
3	If you want to remove the Island files from the hard drive, check the box in the **Remove <Island>** dialog box.
4	Click **OK**.

Back to top *(see page 259)*

Print

Use the **Print** utility to print information about 1 or more Islands in the Workspace.

To print information:

Step	Action
1	From the **File** menu, ● either select **Print** ● or select the following icon on the **Standard** toolbar to open the **Print** dialog box:  **Result:** The **Print-<Workspace>** dialog box is displayed (where <Workspace> contains the name of the selected Workspace).
2	In the **Print** area, select 1 or more Islands from which you want information by clicking the appropriate option button: either **All**, **Active Island**, or **Selected Islands**.
3	In the **Print items** area, check the boxes that specify the type of information you want printed. If you want to print all types of information on the selected *.isl* files, check **Select All**.
4	For multiple copies of the 1 or more documents, type the desired number in the **Number of copies** field.
5	For multiple copies, to print the documents in binding order, check the **Collate** box.
6	If you want to print to a file instead of to a physical printer, check the **Print to file** box. You may store your file either in RTF or PDF format.
7	If you want to see how the 1 or more documents will look when printed, click **Print Preview**.
8	To customize your printer options, click **Setup**.
9	Click **OK**.

Back to top *(see page 259)*

Print Items

You can select the following items to be printed:

Item	Description
Workspace Information	prints Workspace information and list of Islands in the Workspace
Island Information	prints Island information, physical dimensions of segments, and list of modules in the Island
Island Image	prints graphical representation of the Island The graphic is identical to the display in the Island Editor.
Bill of Materials	prints the Bill of Materials for mandatory components It includes hints and alternative components if applicable.
Fieldbus I/O Image	prints input and output process image in the fieldbus view
Modbus I/O Image	prints input and output process image in the Modbus view
Reflex Actions	prints all configured reflex actions
Resource Utilization	prints utilized resources and maximum allowed values
Resource Power Details	prints utilized power in details and maximum allowed values
Resource Configuration Details	prints utilized configuration resources and available resources
Modules in Detail	prints technical data of the modules and configuration details
Annotations	prints all annotations for the Islands in the Workspace

Back to top *(see page 259)*

Print Setup

The **Print Setup** dialog box allows you to specify printer properties, paper size, and page layout characteristics.

To set these print options, perform the following steps:

Step	Action
1	From the **File** menu, select **Print Setup**. **Result:** The **Print Setup** dialog box is displayed.
2	In the **Name:** box, select the printer to be used.
3	In the **Size:** box, select the paper size.
4	In the **Source** box, select the paper source.
5	In the **Orientation** area, select the orientation by clicking **Portrait** or **Landscape**.
6	Click **OK**.

Back to top *(see page 259)*

Export <Island>

The **Export** function lets you export a device description file for the active Island. The Advantys Configuration Software determines the appropriate file formats based on the type of NIM used in the configuration. This functionality is available only in offline mode.

The **Export <Island>** command automatically displays the name of the selected Island, for instance **Export My_Island**.

To export the selected Island, perform the following steps:

Step	Action
1	Select the Island you want to export by clicking in the Island Editor on the tab of the Island to be exported.
2	From the **File** menu, select **Export <Island>** (where <Island> contains the name of the selected Island). **Result:** The **Export** dialog box is displayed.
3	In the **Export Format** area, select the export format. **Note:** Only the formats supported by the Island's NIM are available. The **Prefix** and the **Transformation file** field as well as the **Advanced Options** button are available only for certain NIMs and/or export formats. **Result:** The **Directory** field automatically contains the path for the configuration file. The **Filename** field automatically contains the Island's name and the appropriate extension. In the **PLC Information** area, the appropriate fields are automatically available.
4	If you want to change any information automatically displayed, click the appropriate field and enter the information desired.
5	If available, click the **Advanced Options** button if you want to set advanced export options. Select the desired option and click **OK**.
6	Click **OK**.

NOTE: The Advantys Configuration Software stores the Island, builds the configuration internally, and finally performs the export operation. If the application encounters any problem while saving or building the configuration, the export process is not executed. Any detected error generated during the export process is reported.

Back to top *(see page 259)*

Bill of Materials

The Bill of Materials provides the description of a selected Island with comments and alternatives if applicable. It contains mandatory and optional components. In addition to getting a printout of the Bill of Materials using the **Print** function, you have the possibility to export it to a separate file. This functionality is available only in offline mode.

To generate a Bill of Material of the selected Island, perform the following steps:

Step	Action
1	Select the Island of which you want to generate a Bill of Materials.
2	In the **File** menu, click **Bill of Materials**.
3	In the **Bill of Materials** dialog box, select the folder in which the CSV file shall be stored.
4	Type the name of the configuration file in the **File name:** field.
5	Click **Save**.

You can customize the output of the Bill of Materials (regarding kits, connector types, etc.) by selecting the **Settings** command from the **Options** menu, see *Settings, page 301* and *Bill of Materials, page 307*.

Back to top *(see page 259)*

Recent Files List

The Advantys Configuration Software maintains a list of Workspaces recently used. To open a recent Workspace, select 1 of the listed files. If any Workspace is already open, the Advantys Configuration Software closes the Workspace and prompts for saving changes if necessary. The maximum number of entries in the recent files list can be configured in the **Settings** dialog.

NOTE:
- This list displays the full path of the Workspace file but with an ellipsis (...) in the directory part if the text would exceed the maximum menu width. For example:
  <1. C:\...\Advantys\Projects\ws\ws.aiw>
- If the file is corrupted or deleted externally, the software displays the appropriate detected error message and removes the file in question from the recent files list.

Back to top *(see page 259)*

Exit

You can close the Advantys Configuration Software at any point of time. However, certain operating conditions can lead to a delay. Before closing, the application prompts you to save any changes in the active Workspace.

To close the Advantys Configuration Software, select **Exit** from the **File** menu.

Back to top *(see page 259)*

Edit Menu

Introduction

The **Edit** menu contains the following items:
- Undo *(see page 269)*
- Redo *(see page 270)*
- Revert *(see page 270)*
- Cut *(see page 271)*
- Copy *(see page 272)*
- Paste *(see page 273)*
- Delete *(see page 274)*

Undo

Issuing an **Undo** command cancels the last change you made in the Island Editor. You may undo any number of consecutive commands or actions; there is no limit on the maximum number of executions of an **Undo**. However, you cannot undo any command or action issued before the execution of a Revert *(see page 270)* command.

The following table shows how to issue an **Undo** command:

Step	Action
1	On the **Edit** menu, ● either click **Undo** ● or click the following icon on the **Edit** toolbar: 

For related topics, refer to the following sections:
- *Redo, page 270*
- *Revert, page 270*

Redo

Issuing the **Redo** command reapplies the last command or action that you have changed using the Undo *(see page 269)* command. You can execute a **Redo** for every **Undo** previously issued, in sequence; there is no limit on the maximum number of executions of a **Redo**. However, you cannot **Redo** any command issued before the execution of a Revert *(see page 270)* command.

Issue the **Redo** command as follows:

Step	Action
1	On the **Edit** menu, ● either click **Redo** ● or click the following icon on the **Edit** toolbar: 

For related topics, refer to the following sections:

- *Undo, page 269*
- *Revert, page 270*

Revert

If you issue a **Revert** command, the Island configuration will revert to its last saved state. All changes made on the Island after the last time it was saved are lost and irrecoverable.

Issue the **Revert** command as follows:

Step	Action
1	On the **Edit** menu, ● either click **Revert** ● or click the following icon on the **Edit** toolbar: 

For related topics, refer to the following sections:

- *Undo, page 269*
- *Redo, page 270*

Cut

If you want to move a module or a contiguous cluster of modules from their current location in a segment to another location in the same or on a different Island, begin by so to speak cutting the module(s) to a temporary paste buffer.

The following table shows how to cut the modules from the Island Editor:

Step	Action
1	Single-click the desired module in the Island Editor.
2	To cut a cluster of contiguous modules to the right or left of the selected module, hold down the SHIFT key and click the last module of the cluster you want to cut.
3	On the **Edit** menu, ● either click **Cut** ● or click the following icon on the **Edit** toolbar: 

NOTE: When you cut, paste, or delete modules, the software checks first if this action does not violate the connectivity constraints of the Island. An incompatibility message will be displayed if the action is not allowed.

For related topics, refer to the following sections:

- *Copy, page 272*
- *Paste, page 273*

Copy

If you want to reuse the same module or a cluster of contiguous modules from 1 segment to another location in the same or in a different Island, begin by copying the module(s) to a temporary paste buffer.

Copy the modules from the Island Editor as follows:

Step	Action
1	Single-click the desired module in the Island Editor.
2	To copy a cluster of contiguous modules to the right or left of the selected module, hold down the SHIFT key and click the last module of the cluster you want to copy.
3	On the **Edit** menu, ● either click **Copy** ● or click the following icon on the **Edit** toolbar:  ● or click the following icon on the **Edit** toolbar: 

NOTE: When you copy a module that has been custom configurated, the customized parameter values remain with the module when in the temporary past buffer.

For related topics, refer to the following sections:

- *Cut, page 271*
- *Paste, page 273*

Paste

After you have cut *(see page 271)* or copied *(see page 272)* a module or cluster of contiguous modules to a temporary paste buffer, you can go to a new location in the same or different Island configuration and insert the contents of the paste buffer there.

Paste the modules in a new location as follows:

Step	Action
1	In the Island Editor, open the Workspace containing the Island where you wish to insert the module(s) that are in the paste buffer.
2	Open or create the target Island.
3	Select a module in the target Island to serve as the insertion point. The module(s) in the paste buffer will be positioned directly to the right of this insertion point.
4	On the **Edit** menu, ● either click **Paste** ● or click the following icon on the **Edit** toolbar: 

NOTE: When you cut, paste, or delete modules, the software checks first if this action does not violate the connectivity constraints of the Island. An incompatibility message will be displayed if the action is not allowed.

For related topics, refer to the following sections:

- *Cut, page 271*
- *Copy, page 272*

Delete

The **Delete** command allows you to remove a selected module or cluster of contiguous modules in a segment from the Island configuration.

The following table shows how to delete a module or cluster of modules from a segment:

Step	Action
1	In the Island Editor, click the module you wish to delete.
2	To delete a cluster of contiguous modules to the right or left of the selected module, hold down the SHIFT key and click the last module of the cluster you want to delete.
3	On the **Edit** menu, • either click **Delete** • or click the following icon on the **Edit** toolbar: 

NOTE: When you cut, paste, or delete modules, the software checks first if this action does not violate the connectivity constraints of the Island. An incompatibility message will be displayed if the action is not allowed.

View Menu

Introduction

The **View** menu contains the following items:

- Workspace Browser *(see page 275)*
- Catalog Browser *(see page 276)*
- Log Window *(see page 276)*
- Toolbars *(see page 277)*
- Status Bar *(see page 277)*
- Zoom *(see page 277)*

Workspace Browser

By default, the Workspace Browser appears on the left side of the screen when you create a new Workspace. You have the option of hiding this browser if you do not need to use it.

To hide the Workspace Browser, clear the **Workspace Browser** check box on the **View** menu, or simply click once on the following icon on the **View** toolbar:

If you close the Workspace when the Workspace Browser is hidden, it will still be hidden the next time you open the Workspace. If you open an existing Workspace and do not see the Workspace Browser, select the **Workspace Browser** check box on the **View** menu, or click once more the icon on the **View** toolbar to display the browser:

Catalog Browser

By default, the Catalog Browser appears on the right side of the screen when you create a new Workspace. You have the option of hiding this browser if you do not need it.

To hide the Catalog Browser, clear the **Catalog Browser** check box on the **View** menu, or simply click once the following icon on the **View** toolbar:

If you close the Workspace with the Catalog Browser hidden, it will still be hidden the next time you open the Workspace. If you open an existing Workspace and do not see the Catalog Browser, select the **Catolog Browser** check box on the **View** menu, or click once more the icon on the **View** toolbar to display the browser:

Log Window

By default, a Log Window appears at the bottom of the screen when you create a new Workspace. You have the option to hide this window if you do not need to use it.

To hide the Log Window, un-select the **Log Window** check box on the **View** menu, or simply click once the icon on the **View** toolbar:

If you close the Workspace with the Log Window hidden, it will remain hidden when the Workspace is reopened. If you open an existing Workspace and do not see the Log Window, you can make it appear by selecting the **Log Window** check box on the **View** menu, or simply clicking once more the icon on the **View** toolbar to display the browser:

Toolbars

Toolbars display a collection of easy-to-use button images and/or menus that initiate different operations in the Advantys Configuration Software. Several icons are docked on independent toolbars at the top of the screen.

You can hide, show and move these 4 toolbars:
- **Standard** toolbar *(see page 254)*
- **Edit** toolbar *(see page 255)*
- **View** toolbar *(see page 256)*
- **Island** toolbar *(see page 257)*

If you open a work session and cannot find 1 or more of these toolbar menus on the screen, proceed as follows:

Step	Action
1	On the **View** menu, click **Toolbar** .
2	Check the missing toolbar menu name in the **Toolbar** list.

Status Bar

A status bar appears in the bottom panel of the Workspace display.

The status bar indicates
- status messages
- offline/online status
- physical Island status
- test mode status

If you open an existing Island, the status bar may not be visible. To reveal it, go to the **View** menu and click **Status Bar**.

Zoom

The **Zoom** function allows you to select the size of the images in the Island Editor display. If you need a close-up view of a set of modules in the Island Editor, you can view the display at 100% (the default view). If you need to view all the modules in a multi-segment Island configuration, zoom out to 75%, 50% or 25%.

Use the **Zoom** function as follows:

Step	Action
1	On the **View** menu, click **Zoom**, or click the **Zoom** list box in the **View** toolbar.
2	Click 1 of the **Zoom** options: **100%**, **75%**, **50%**, or **25%**.

Island Menu

Introduction

The **Island** menu contains the following items:

Add Rail

In the Island Editor, DIN rails act as the installation anchors for each segment of Advantys modules. When you create a new Island, an empty DIN rail appears in the Island Editor where you will configure the modules in the primary segment. If the primary DIN rail was deleted for any reason, add a new rail before configuring the Island.

For the extension segments, DIN rails are added automatically when you select a beginning-of-segment (BOS) module from the Catalog Browser.

To add the primary DIN rail to the Island Editor, select **Add Rail** from the **Island** menu.

Add Annotation

You can place text labels and comments in an Island Editor using the add annotation feature. To add an annotation to your Island Editor, click **Add Annotation** on the **Island** menu, or click the following icon on the **Island** toolbar:

A text frame appears in the Island Editor display. Simply type the desired text in this frame. The text area may be stretched to different dimensions and positioned at different locations in the Island Editor display. In the Island Editor, you may add as many of these text areas as you need.

NOTE: You can stretch an annotation with the so-called resizer points on the frame. You can move the annotation with the drag-and-drop operation when you click the resizer points so that they change to green color.

Delete Annotation

You can delete annotations from the Island Editor in 2 ways.

Delete an annotation using the menu as follows:

Step	Action
1	Select the annotation.
2	On the **Island** menu, click **Delete Annotation**.

Delete an annotation using the keyboard as follows:

Step	Action
1	Select the annotation.
2	Press ESC so that the frame turns green.
3	Press DELETE.

Replace NIM

You have the possibility to exchange the NIM type or the NIM version in an existing Island configuration by using the replace NIM feature.

To replace the NIM, the Island concerned has to be offline, open and unlocked. Proceed as follows:

Step	Action
1	Select the NIM module.
2	• Either click the **Island** menu and select **Replace NIM** • or right-click the NIM and select **Replace NIM** from the shortcut menu.
3	In the selection dialog, select the desired NIM and click **OK**.
4	Confirm your selection by clicking **Yes**.

Add Module

The **Add Module** command opens a submenu with all module families available in the Advantys catalog. By selecting 1 of these families the software opens an eventually hidden Catalog Browser with this family tree expanded. This feature can be used to simply find a family in the catalog for quick module adding.

Module Editor

The Module Editor provides information about a selected module, allows you to modify some of its operating parameters, and to view live I/O data when the software is in online mode.

To invoke the Module Editor, select the appropriate module from the Island Editor or from the Workspace Browser:

To open the Module Editor from the ...	Then ...
Island Editor	double-click the desired module.
Workspace Browser	right-click once the module label and select **Module Editor** from the shortcut menu.

Alternatively, you may access the Module Editor by selecting the desired module in either the Island Editor or Workspace Browser and selecting **Module Editor** from the **Island** menu, or clicking the following icon on the **Island** toolbar:

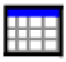
It is not possible to access the Module Editor from the Catalog Browser.

Refer to *Module Editor, page 66* for further information.

User Defined Label Editor

The **User Defined Label Editor** is a single location where you can associate labels with the I/O image data items for all the I/O modules in the current Island. This feature is applicable only for STB Islands.

The following table shows how to open the **User Defined Label Editor**:

Step	Action
1	From the **Island** menu, click **Label Editor**, or click the following icon on the **Island** toolbar:  **Result**: The **User Defined Label Editor** is displayed.
2	In the **User Defined Label** column, click the cell you want to modify.

NOTE: You can modify user defined labels in the **User Defined Label Editor** only when the Island is offline and unlocked. But you can launch the **User Defined Label Editor** when the Island is online or offline in either locked or unlocked mode.

Refer to User Defined Label Editor Introduction *(see page 116)* for further information.

Reflex Editor

Reflex actions are small routines that perform dedicated logical functions directly on the Island bus. They allow output modules on the Island to act on data and drive field actuators directly, without requiring the intervention of the fieldbus master.

The following table shows how to open the Reflex Editor:

Step	Action
1	On the **Island** menu, click **Reflex Editor**, or click the following icon on the **Island** toolbar: 
2	Add, modify or delete the reflex actions as per your convenience.

NOTE: You can modify a reflex action only when the Island is offline and unlocked. By default, the Reflex Editor is displayed with only the **New** and **Close** command buttons enabled. All other buttons are disabled.

For related topics, refer to the following sections:
- *Working with the Reflex Editor, page 211*
- *Adding a Reflex Action, page 212*
- *Modifying a Reflex Action, page 212*
- *Working with the Reflex Editor, page 211*

Build

To build an Island, click **Build** on the **Island** menu, or click the following icon on the **Island** toolbar:

The build process configures the following:
- communication objects for the Island
- reflex actions
- module's customized operating parameter values
- synchronization objects on the Island
- process image area in the NIM
- HMI area in the Island's process image

Furthermore, it creates
- intermediate log files for the Island
  (if the **Generate intermediate log files** option is checked in the application settings) and
- a downloadable binary file.

Lock

You can lock an Island to limit access to it.

If an Island is locked, you can only view the configuration. You are able to
- view the Island configuration in the Island Editor,
- view the parameter values assigned to the modules on the Island using the Module Editor,
- view the reflex actions configured for the Island using the Reflex Editor, and
- export a definition file describing the Island (for certains fieldbuses only).

To lock or unlock an Island Editor, click **Lock** on the **Island** menu, or click the following icon on the **Island** toolbar:

For further information on locking Islands, refer to *Locking and Protecting an Island, page 150*.

Resource Analysis

The **Resource Analysis** dialog box displays a report pertaining to the consumption of resources by the Island.

To access the **Resource Analysis** dialog box, select the **Resource Analysis** option from the **Island** menu, or click the following icon on the **Island** toolbar:

For further information on the resource analysis, refer to *Resource Analysis, page 171*.

I/O Image Overview

The Advantys Configuration Software provides a utility that gives you an overview of the I/O data and status allocation for all the modules on the Island. It also gives you a view of any HMI data that may be written to the Island bus or read by the fieldbus master.

To access the I/O Image Overview, select **I/O Image Overview** on the **Island** menu, or click the following icon on the **Island** toolbar:

For further information on the I/O image overview, refer to *I/O Image Overview, page 174*.

Baud Rate Tuning

The dialog box **Baud Rate** allows to select the baud rate for the internal Island bus. This is required if an STB Island is extended to enhanced CANopen devices. The default value is 800 Kbps; changes can only be made if the Island is unlocked.

Possible values range from 500...800 Kbps.

Temperature Range

Many Advantys STB modules are able to operate within an extended temperature range. To set the range, select **Temperature Range** from the **Island** menu. Generally, the following ranges are available:

- Normal (0...60° C)
- Low (-25...60° C)
- High (0...70°C)
- Extended (-25...70°C)

The temperature range is valid for the whole Island. Accordingly, you can only set a range that is supported by all modules of an Island. Further, you can add only modules that support the range that is selected for the Island concerned. Thus, only the modules supporting the range selected are available within the Catalog Browser and for replacing the NIM. For any other way of adding modules, a message is displayed and the operation is canceled in any case of non-conformance.

If the temperature range of an Island is anything but normal, an icon is displayed left to the NIM. The figure below shows the icons indicating the low, the high and the extended temperature range (from left to right):

Test Mode Settings

NOTE: This function is available on certain NIMs only. If the configured NIM does not support extended test modes, this topic is grayed out.

This dialog box allows you to select 1 of the 3 implemented test modes:

Test Mode	Description
Temporary Test Mode	It is the default setting. The test mode is activated by the user when online with the Advantys Configuration Software. After a power cycle, the test mode will no longer be activated, thus the test mode is not permanent.
Persistant Test Mode	It allows you to be permanently in test mode. Once set, the Island will remain in test mode until the test mode is reset (requires configuration change). It will remain in test mode through all power up/down scenarios.
Password Test Mode	It allows the switch to a test mode if the user enters a password in the configuration software. This password is then stored in the flash memory. Any Modbus HMI can come in and put the Island in test mode by writing the same password in the specific location in the Modbus area. This mode is not remembered upon a power cycle.

Island Properties

The Advantys Configuration Software provides a dialog box from which you can view the properties of an Island in the current Workspace. From this dialog box, you may also modify some of the Island properties.

To access the Island properties select **Island → Island Properties**, or from the Workspace Browser, right-click the label of the desired Island, and select **Properties** from the shortcut menu.

The **Island Properties** dialog box contains 9 fields. Some of these fields are read-only; others can be edited:

Field Name	Field Character	Field Content
Logical Name:	read-write	logical name of the selected Island
Island Name:	read-only	name of the file to which the Island configuration is saved
Last Modified:	read-only	date and time of the last file modification
Catalog version used:	read-only	version of the Advantys catalog database used to create the selected Island
Catalog revision used:	read-only	revision number, if any, of the Advantys catalog database used to create the selected Island
Author:	read-write	author's name
Comments:	read-write	any comments about the Island

Field Name	Field Character	Field Content
Version Number:	read-write	3 fields representing the major version, minor version and revision of the Island file
Auto Increment:	read-write	If this box is checked, the revision number of the Island file will automatically increment up to 99. If the revision number exceeds 99, it will restart at 0 and a minor number will automatically begin incrementing by 1 until it also reaches 99. If the minor number exceeds 99, the minor number will start back from 0, and the major number will automatically begin incrementing by 1 until it reaches 99.

Online Menu

Introduction

The **Online** menu contains the following items:

- Connect *(see page 286)*
- Disconnect *(see page 287)*
- Connection Settings *(see page 287)*
- Configuration Port Settings *(see page 290)*
- Run *(see page 290)*
- Stop *(see page 291)*
- Reset *(see page 291)*
- Download into the Island *(see page 291)*
- Upload from the Island *(see page 292)*
- Store to SIM *(see page 293)*
- Protect *(see page 293)*
- Force Auto-configuration *(see page 294)*
- Test Mode *(see page 294)*
- I/O Image Animation *(see page 295)*

Connect

The Advantys Configuration Software is considered online when it has been successfully connected to a physical Island that is under power and able to operate.

Connect the software to a physical Island as follows:

Step	Action
1	Make the physical connection with a Modbus cable between the programming panel running the configuration software and the configuration port on the Island's NIM. **Note**: If you are connecting to an Island that uses an Ethernet NIM, you may make the physical connection via Ethernet or Modbus. In all other cases, the physical connection to the Island has to be via Modbus.
2	From the **Online** menu, select **Connect**, or click the following icon on the **Standard** toolbar:  **Result:** If this is the first time you establish a connection in this session, the **Connection Settings** dialog box will appear *(see page 287)*.

Connection Process

When you have performed the above procedure, the software attempts to connect to the physical Island and does the following:

Phase	Description
1	Reads the signature of the physical Island. (If the signature of the logical Island mismatches, the software will prompt to upload or download the configuration.)
2	Checks the mode of the Island. (If the software is in protect mode, you will be prompted to enter a password to get connected.)

If a configuration mismatch between the logical and physical Island occurs, a dialog box will be displayed:

If you want to ...	Then click ...
copy the configuration from the physical Island to the software	**Upload**.
copy the configuration from the software to the physical Island	**Download**.
invoke a configuration consistency check (only for OTB)	**Compare**.
return to offline mode	**Cancel**.

The status of the online operation is displayed in the status bar. Further information on the configuration consistency check, refer to Configuration Consistency Check *(see page 296)*.

For related topics, refer to Connection Settings *(see page 287)*.

Disconnect

If a connection to the physical Island exists, you can terminate this connection. If the Island is in test mode, you will be asked whether you want to deactivate the test mode before disconnecting.

To disconnect from an Island, select **Disconnect** from the **Online** menu, or click the following icon on the **Standard** toolbar:

The status of the online operation is displayed on the status bar.

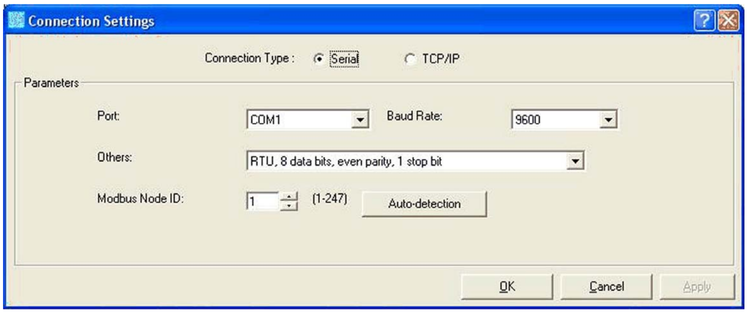
Connection Settings

Before connecting to the serial port on a physical Island (or possibly via TCP/IP if you are operating on an Ethernet bus), check that the connection parameters in the software are set to match those of the physical communication port.

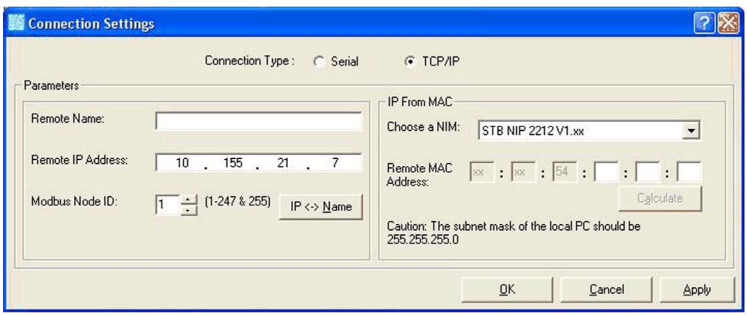
Proceed the following steps to enter the connection settings or use the auto-detection feature of the Advantys Configuration Software *(see page 299)*.

Set the connection parameters as follows:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>From the Online menu, select Connection Settings.
Result: The Connection Settings dialog box is displayed.</td></tr>
<tr><td>2</td><td>If you are connecting to the physical island through:
● the RS232 communications port on the NIM, click Serial.
● an Ethernet fieldbus, and you are using the fieldbus port as your connection point, click TCP/IP.</td></tr>
<tr><td>3</td><td>In the Serial connection type and under Parameters area, select the port, the baud rate and other connection settings so that they match with those set on the physical port through which you are connecting.
The following figure shows the Parameters area of the Serial connection type:



In the TCP/IP connection type and under Parameters area, type the IP address of the NIM.
The following figure shows the Parameters area of the TCP/IP connection type:

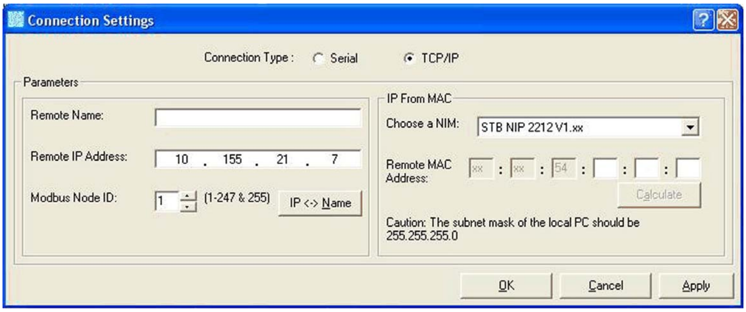


To set the IP From MAC parameters, refer to IP From MAC (see page 289).</td></tr>
<tr><td>4</td><td>Click Apply.</td></tr>
<tr><td>4</td><td>Click OK in the Connection Settings dialog box.</td></tr>
</table>

IP From MAC

To calculate the remote IP address from the MAC address (if your device is set up to do so), perform the following steps:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>In the Parameters area of the TCP/IP connection type and under IP From MAC, the Choose a NIM field automatically contains the name of the NIM of the logical Island.
The following figure shows the Parameters area of the TCP/IP connection type:



NOTE: If no NIM is displayed, the island contains a NIM that uses a serial connection.</td></tr>
<tr><td>2</td><td>If required, select another Ethernet NIM from the Choose a NIM list. This may be necessary, for example, if you want to connect to a physical Island that contains a different Ethernet NIM from your logical Island to upload the configuration.</td></tr>
<tr><td>3</td><td>In the empty segments of the Remote MAC Address field, enter the appropriate address values.
NOTE: In case of a STB NIC 2212 NIM, do not type <code>FF</code> in the last segment of the Remote MAC Address field. Otherwise, a calculation is not possible.</td></tr>
<tr><td>4</td><td>Click Calculate.
Result: The remote IP address is calculated from the remote MAC address and displayed in the Remote IP Address field.
NOTE: The remote IP address can also be entered manually.
To search the name of the remote IP address automatically, click IP <- > Name.</td></tr>
<tr><td>5</td><td>Click Apply.</td></tr>
<tr><td>6</td><td>Click OK.</td></tr>
</table>

Configuration Port Settings

Having successfully connected to the physical Island, you can change the following communication parameters of the NIM within available ranges:

- baud rate
- Modbus node ID of the NIM
- connection type
- parity
- stop bits
- data bits

This command is available in online mode only. To set communication parameters, click **Configuration Port Settings** on the **Online** menu.

Before the software can accept the change of the configuration port settings, it has to set the Island to reset state and prompts you for confirmation. After a change of the configuration port settings, the tool disconnects from the physical Island.

NOTE: The physical Island will stay in reset state and needs to be set to run mode manually after a change of the configuration port settings. From the next connection onwards, the new connection settings are used for the logical Island. The software reminds you by showing the **Connection Settings** dialog box.

NOTE: A reset does not cause the Island to auto-configure itself.

Run

Using this command, you can set the physical Island to run mode. This command is available in online mode only. It is disabled if the physical Island to which you are connected is already in run mode.

To set the physical Island to run mode, select **Run** from the **Online** menu, or click the following icon on the **Standard** toolbar:

Prior to running the Island, the software prompts you for confirmation during the transition between the Island states.

NOTE: When you run the Island, processing will be started on the Island.

For related topics, refer to the following sections:

- Stop *(see page 291)*
- Reset *(see page 291)*

Stop

Using this command, you can set the physical Island to stop state. This command is available in online mode only. It is disabled if the physical Island to which you are connected is already in stop state.

To set the Island to pre-operational state, select **Stop** from the **Online** menu, or click the following icon on the **Standard** toolbar:

Prior to stopping the Island, the software prompts you for confirmation during the transition between the Island states.

For related topics, refer to the following sections:
- Run *(see page 290)*
- Reset *(see page 291)*

Reset

Using this command, you can reset the physical Island. If the Island is reset, the input and output data are cleared and all the modules on the Island bus are auto-addressed. This command is available in online mode only. It is disabled if the Island is currently in reset state.

NOTE: A reset does not cause the Island to auto-configure itself.

To reset the Island, click **Reset** on the **Online** menu. Prior to resetting the Island, the software prompts you for confirmation during the transition between the Island states.

For related topics, refer to the following sections:
- Run *(see page 290)*
- Stop *(see page 291)*

Download into the Island

The **Download into the Island** command allows you to transfer a configuration file previously built in the Advantys Configuration Software to the connected physical Island. The configuration file is downloaded into the NIM's RAM and flash, where it can then be saved to a removable memory card. This command is only available when the Island is in online mode.

To perform a configuration download, click **Download into the Island** on the **Online** menu. When you download a configuration into a physical Island, the Island has to be in reset state.

If the Island is not already in reset state, a dialog box is displayed informing you that the Island is set to reset state:

If you want to ...	Then click ...
proceed with the download	**Yes**.
cancel the download	**No**.

NOTE: You cannot perform a download without entering the password when the Island is in protect mode.
During the download process, a progress bar is displayed, tracking the status of the download.
A reset does not cause the Island to auto-configure itself.

For related topics, refer to Upload from the Island *(see page 292)*.

Upload from the Island

The **Upload from the Island** command allows you to upload a configuration to the Advantys Configuration Software from a physical Island. The configuration is uploaded to the logical Island that is currently open in the Island Editor of the Advantys Configuration Software. You can only upload the physical Island's configuration when the software is in online mode.

If a mismatch between the module IDs and the database entries is detected or any detected errors occur during the upload process of the configuration from the physical Island, the upload will be aborted and diagnostic messages will be displayed in the Log Window.

NOTE: If an auto-configured physical Island is uploaded, modules that do not communicate on the Island bus are not included in that operation. For compatibility reasons, required modules are inserted by the software. However, the modules inserted may not exactly match the ones configured. Check the configuration that is uploaded and, if necessary, insert or exchange the modules concerned using the Catalog Browser.

The following table shows how to perform a configuration upload:

Step	Action
1	On the **Online** menu, click **Upload**. **Result:** A dialog box prompts you to confirm.
2	If you want to ● proceed with the upload, click **Yes**. ● cancel the operation, click **No**.

For related topics, refer to Download into the Island *(see page 291)*.

Store to SIM Card

After downloading a configuration into flash, you have the option of saving the new configuration to a removable memory card in the physical Island using the Advantys Configuration Software. This command is available only for STB Islands and can only be issued when the Advantys Configuration Software is in online mode.

From the **Online** menu, select **Store to SIM Card**. Prior to storing the configuration to SIM card, the software prompts you for confirmation during transition between the Island states.

NOTE: A reset does not cause the Island to auto-configure itself.

Protect

You can limit the access to the available online functions by setting the physical Island to protect mode.

If the Island is protected, enter a user-assigned password to perform the following:
- change configuration port settings
- change the state of the physical Island
- download configurations
- store to SIM card
- unprotect the Island
- activate/deactivate test mode
- force output data in test mode

When the software is online, you can toggle between protect and edit mode:

If you want to toggle the Island to ...	Then ...
protect mode	check **Protect** on the **Online** menu.
edit mode	clear **Protect** on the **Online** menu.

The mode of the Island is displayed on the title bar adjacent to the Island's name and indicated by the check mark left of the **Protect** option.

NOTE: If you change the status of the protect mode, the Island is set to reset state. This stops all processing on the Island.

NOTE: A reset does not cause the Island to auto-configure itself.

For related topics, refer to Test Mode *(see page 294)*.

Force Auto-Configuration

For STB Islands, you have the ability to force an auto-configuration on the physical Island to which the Advantys Configuration Software is connected. If you force an auto-configuration, all customized configuration parameters set by you are lost, and all modules on the Island revert back to their default configuration parameters. The connection settings and the password are not affected. This command can only be issued when the Advantys Configuration Software is in online mode.

On the **Online** menu, click **Force Auto-configuration**.

NOTE: When the Island is protected, you are not allowed to force an auto-configuration without entering the password. After auto-configuration, the Island will be disconnected.

Test Mode

Only 1 device is allowed to control the physical Island's process image at any given time. The master device is the fieldbus master. In order to control Island outputs locally by the software or by an HMI, the test mode needs to be activated. The **Test Mode** option allows the software to obtain or release control over the process image. The activation of the test mode is indicated by switching on the Test-LED on the NIM and also by a bit (CTM_PIO) in the NIM device status.

NOTE: While the software has mastery over the process image, the fieldbus/PLC can read from but not write to the physical Island.

The following table shows how to activate the test mode:

Step	Action
1	On the **Online** menu, click **Test Mode**. **Result:** A dialog box prompts you to confirm.
2	If you want to ● proceed, click **OK**. ● cancel the operation, click **Cancel**.

Whenever you disconnect from the physical Island in test mode, the software automatically attempts to relinquish its control of the process image. The software displays a dialog box asking whether you want to deactivate the test mode.

The status bar at the bottom of the screen reflects the control status. If the software is in control, the status bar reads **Test Mode ON** in the **Test Mode** pane. When the fieldbus is in control, the status bar reads **Test Mode OFF** in the **Test Mode** pane.

For related topics, refer to Protect *(see page 293)*.

Test Mode Settings

These are the various test mode options available *(see page 163)*:
- temporary test mode
- persistent test mode (not available for standard V1 NIMs)
- password test mode (not available for standard V1 NIMs)

I/O Image Animation

The I/O image animation is a dynamic display of I/O data communication between the physical Island and the fieldbus master (and between an HMI panel on the Island and the fieldbus). The I/O image animation is only accessible in online mode.

To open the I/O image animation, click **I/O Image Animation** on the **Online** menu, or click the following icon on the **Island** toolbar:

For further information on the I/O animation, refer to I/O Image Animation *(see page 165)*.

Configuration Consistency Check

Introduction

If you want to connect the software to your physical Island and the configurations of your logical and your physical Island mismatch, the **Data Transfer** dialog box is displayed. Before uploading or downloading a configuration, you have the possibility to compare both configurations with respect to the following:

- modules
- parameters
- I/O mappings

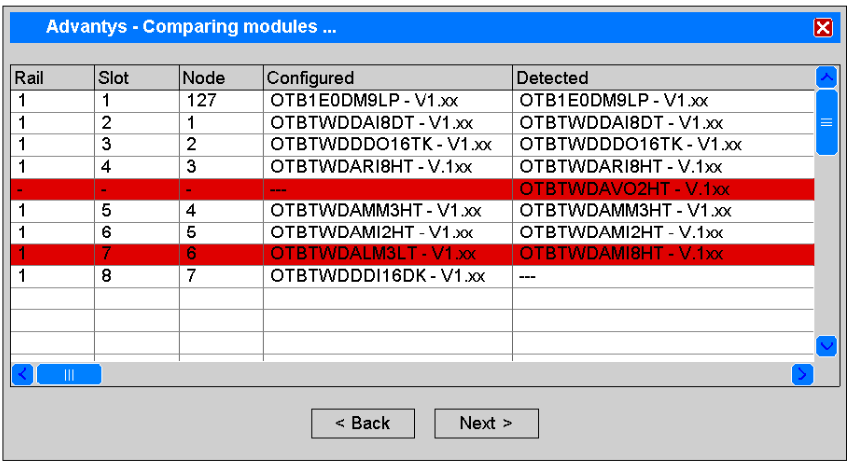
This comparison is based on an upload of the configuration from the physical Island. To compare the configurations, click **Compare** in the **Data Transfer** dialog box.

NOTE: The **Compare** option is only available for OTB Islands.

Comparing Modules

First, the modules configured in the logical Island are compared to those detected in the physical Island and the **Advantys - Comparing modules ...** dialog box is displayed. In this dialog box, all modules of both configurations are listed, independent of the result of the comparison.

This figure provides an example of the **Advantys - Comparing modules ...** dialog box, showing differences in the physical and logical Island configurations:



Rail	Slot	Node	Configured	Detected
1	1	127	OTB1E0DM9LP - V1.xx	OTB1E0DM9LP - V1.xx
1	2	1	OTBTWDDAI8DT - V1.xx	OTBTWDDAI8DT - V1.xx
1	3	2	OTBTWDDDO16TK - V1.xx	OTBTWDDDO16TK - V1.xx
1	4	3	OTBTWDARI8HT - V.1xx	OTBTWDARI8HT - V.1xx
-	-	-	---	OTBTWDAVO2HT - V.1xx
1	5	4	OTBTWDAMM3HT - V1.xx	OTBTWDAMM3HT - V1.xx
1	6	5	OTBTWDAMI2HT - V1.xx	OTBTWDAMI2HT - V.1xx
1	7	6	OTBTWDALM3LT - V1.xx	OTBTWDAMI8HT - V.1xx
1	8	7	OTBTWDDDI16DK - V1.xx	---

Comparison Process

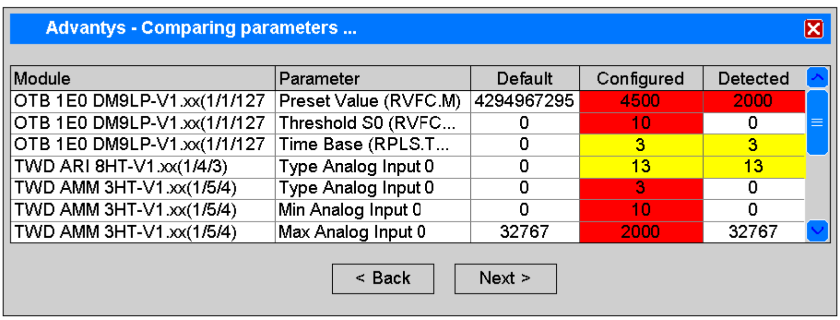
For the module comparison, the following applies:

If the modules ...	Then ...
mismatch	the fields with differences are marked with a red background color. If a mismatch is followed by a matching module, this is detected. An empty field is then assigned to the mismatching module, the matching ones are listed in the same line and the comparison is continued as shown in the figure above. A matching module in the next but one line is not detected. Using either the **< Back** or the **Next >** button, you can return to the **Data Transfer** dialog box.
are identical	they are listed without any fields being marked. You can access these dialog boxes successively by clicking **Next>**: ● **Advantys - Comparing parameters ...** for comparing configuration parameters ● **Advantys - Comparing I/O mappings ...** for comparing I/O mappings Using the **< Back** button, you can return to the dialog box previously displayed. Using the **Next >** button, you can access the following dialog box (the last comparison box is followed by the **Data Transfer** dialog box).

Comparing Parameters

The **Advantys - Comparing parameters ...** dialog box lists all mismatching parameters. Further, deviations from default values are indicated. Precondition is that at least 1 parameter mismatches. Otherwise, the table is empty.

This figure provides an example of the **Advantys - Comparing parameters ...** dialog box:



The parameter values are listed and marked as follows:

If a parameter value ...	If a parameter value ...
differs from its counterpart as well as from its default (first line in the figure above)	marked with a red background color.
differs from its counterpart but equals its default (e.g. second line in the figure above)	not marked (because the counterpart field is already marked to indicate the mismatch).
equals its counterpart but differs from its default (e.g. third line in the figure above)	marked with a yellow background color. **Note:** These differences are not displayed unless a parameter value mismatches, which is the precondition for the dialog box to list the differences at all.

Comparing I/O Mappings

The **Advantys - Comparing I/O mappings ...** dialog box lists all mismatching I/O mappings and indicates deviations from default values. The table is empty if the I/O mappings are identical and equal their defaults.

The I/O mappings are listed and marked as follows:

If an I/O mapping ...	Then the corrsponding field is ...
differs from its counterpart as well as from its default	marked with a red background color.
differs from its counterpart but equals its default	not marked (because the counterpart field is already marked to indicate the mismatch).
equals its counterpart but differs from its default	marked with a yellow background color.

Auto-Detection for Serial Parameters

Introduction

If you want to connect to the serial port on a physical Island, the connection parameters in the software has to match the physical communication port to which you will connect.

If necessary, you can adapt the connection settings

- either by manually choosing the correct parameters
- or by using the automatic detection feature.

For information on how to manually choose the connection settings, see *Connection Settings, page 287*. The automatic detection feature is explained below.

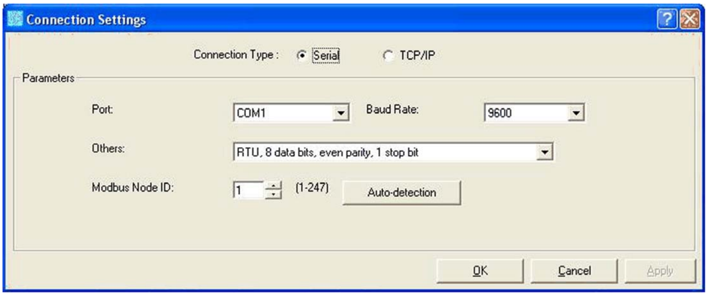
Automatic Detection Feature

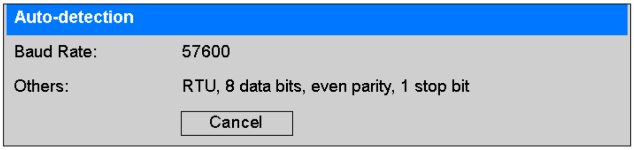
Automatic detection means that the software is searching for the correct connection settings. To do this, it enters a loop within which it is trying to connect to the Island via the chosen communication port for the Modbus address selected. In doing so, the software uses different baud rate and framing settings.

NOTE: The **Connection Settings** option is only available for STB and OTB Islands and if the Island is not connected. The **Auto-detection** option is only available for serial connections.

Activating the Automatic Detection

To activate the automatic detection feature, perform the following steps:

Step	Action
1	From the **Online** menu, select **Connection Settings**. **Result:** The **Connection Settings** dialog box is displayed as follows: 
2	Click **Serial** in the **Connection Type** area, and then select the Modbus address from the **Modbus Node ID** box. **Note:** The auto-detection will search for the serial port parameters for the Modbus port address selected from the **Modbus Node ID** box.

Step	Action
3	Click the **Auto-detection** button. **Result:** The automatic detection is activated and the **Auto-detection** message box is displayed as follows, updated with each attempt:  **Note:** Each attempt is logged in the Log Window, where it can be monitored.

While the automatic detection is running, the following situations can arise:

If the automatic detection ...	Then ...
is successful	the message box is updated to report success and displays the detected connection settings. The settings are retained and the **Serial Parameters** dialog box is updated accordingly. **Note:** The automatic detection feature detects the serial port parameters but does not connect to the Island. To do this, select **Online** → **Connect**.
is not successful	the automatic detection is canceled and a message box states that it was not succesful. **Note:** The connection settings are not changed. Check the physical connection and the communication port selection.
shall be stopped	click **Cancel** in the **Auto-detection** message box. **Result:** The connection settings are not changed.

Options Menu

Introduction

The **Options** menu contains the following items:
- Settings *(see page 301)*
- Workspace Properties *(see page 302)*
- Catalog Properties *(see page 303)*

Settings

The **Settings** dialog box is a multi-screen editor where you can define your preferences for the Advantys Configuration Software environment.

To open the settings, go to the **Options** menu and select **Settings**.

The **Settings** dialog box contains 3 tabs:

Use the ...	To customize ...
Environment tab	• the different **Display settings** by clicking check boxes. • the **Path settings** either directly by typing in a new path name or by clicking the ellipsis button adjacent to the text box and selecting the path by browsing through the directory structure in the **Set Path** dialog box. • an option in the **Other settings** area by simply selecting the option from the drop-down list. If you intend to generate intermediate log files during build, select the corresponding option.
Color tab	the colors of specific diagnostic messages. The default colors for the different detected errors are displayed. To modify the colors, click the ellipsis button next to the type of detected error and select a color.
Bill of Materials tab	the output of the Bill of Materials according to your preferred • calculation algorithm (based on kits or individual parts), • amount of module information, • type of connectors (spring or screw), and • extension cable length selection.

Back to top *(see page 301)*

Set Path

Indicate a path from the **Set path** dialog box. By default, the path displayed leads to the directory previously selected as default in the **Settings** dialog box.

Back to top *(see page 301)*

Workspace Properties

To view the Workspace properties, go to the **Options** menu and select **Workspace Properties**, or right-click the Workspace node from the Workspace Browser and select **Properties** from the shortcut menu.

The following table describes the fields in the **Properties** dialog box:

Field Name	Field Property	Description
Logical name:	read-write	logical name of the Workspace
Filename:	read-only	name of the file in which the Workspace configuration is saved
Last modified:	read-only	date and time of the last modifications to the file
Author:	read-write	name of the file's author
Comments:	read-write	your comments, if any, regarding the Workspace
Version Number:	read-write	There are 3 fields representing the major version, minor version, and revision of the Workspace file
Auto Increment:	read-only	Selecting this option causes the Advantys Configuration Software to auto-increment the major revision number of the *.aiw* file. If the revision number is incremented beyond 99, the major revision number wraps back to 0 and subsequent revisions will have a minor number that auto-incremented until it reaches 99. If the minor number increments beyond 99, the minor number wraps to 0 and the major number begins to auto-increment until it reaches 99.

Back to top *(see page 301)*

Catalog Properties

To view the catalog properties, go to the **Options** menu and select **Catalog Properties**, or right-click on the catalog node in the Catalog Browser and select **Properties** from the shortcut menu.

The following table describes the fields in the **Catalog** dialog box:

Field Name	Field Property	Description
Filename:	read-only	catalog file name
Database version:	read-only	version number of the catalog database
Last modified:	read-only	date and time of the last modifications to the catalog
Author:	read-only	name of the original author
URL for download:	read-only	URL for the website from which catalog updates can be downloaded
Comments:	read-only	comments entered by the original author

NOTE: Some of the changed settings will take effect only after restart of the tool.

Back to top *(see page 301)*

Window Menu

Introduction

The **Window** menu contains the following items:

- Maximize *(see page 304)*
- Minimize *(see page 304)*
- Tile Horizontally *(see page 304)*
- Tile Vertically *(see page 304)*
- Cascade *(see page 305)*
- Open Island List *(see page 305)*

Maximize

The **Maximize** option allows you to enlarge the Island Editor to full screen. To do this, click **Maximize** on the **Window** menu or this icon on the title bar:

Minimize

Use the **Minimize** option to reduce the Island Editor to a button in the Workspace. Click **Minimize** on the **Window** menu or this icon on the title bar:

Tile Horizontally

The **Tile Horizontally** option lets you display all the open Island Editors in separate windows, tiled horizontally without overlapping. To tile the Island Editor screens horizontally, click **Tile Horizontally** on the **Window** menu.

Tile Vertically

The **Tile Vertically** option allows you to display all the open Island Editors in separate windows, tiled vertically without overlapping. To tile the Island Editor screens vertically, click **Tile Vertivcally** on the **Window** menu.

Cascade

Use the **Cascade** option to rearrange all the open Island Editors so that they overlap. The title bar displaying each Island name and a portion of each of the Island Editors will be visible in the cascade. To cascade the view of Island Editor screens, click **Cascade** on the **Window** menu.

Open Island List

When you have multiple Islands opened in the Island Editor, the names of all open ISL files appear on the **Window** menu. Select the Island you wish to activate in the Island Editor by opening this list and selecting the name of the desired *.isl* file.

NOTE: You can also click the Island Editor panes to activate an Island Editor.

Help Menu

Introduction

The **Help** menu contains the following items:

- Contents *(see page 306)*
- Index *(see page 306)*
- What’s This? *(see page 306)*
- About *(see page 306)*

Contents

The **Contents** option provides you with a list of help topics available in the Advantys Configuration Software. On the **Help** menu, click **Contents**, or click the following icon on the **Standard** toolbar:

Index

The Index function allows you to search the Advantys Configuration Software's help system for specific words and phrases, or from a list of keywords. You can also streamline your searches by adding or deleting keywords.

On the **Help** menu, click **Index**.

What's This?

The **What’s This?** utility delivers quick, contextual help on a selected element in the Workspace with a simple mouse click:

Step	Action
1	Click the **?** icon embedded in many of the dialog boxes or click the following icon on the **Standard** toolbar: 
2	Move the **?** icon to the element on the menu or in the dialog box about which you want to obtain help.
3	Click the left mouse button to display the help in a tooltip window.

About

The **About** dialog box displays the version numbers of the Advantys Configuration Software components, its copyright, legal and licencing notices, the user and company name, and the software serial number.

On the **Help** menu, click **About**.

Bill of Materials

Introduction

The Bill of Materials provides the description of a selected Island including comments and alternatives if applicable. It contains mandatory and optional components. In addition to getting a printout of the Bill of Materials using the **Print** function, you have the possibility to export it to a separate file.

Exporting the Bill of Materials

The information for the Bill of Materials can be exported to a CSV (comma separated value) file by clicking **File** → **Bill of Materials**. This file can be imported by all prevalent spreadsheet applications. To avoid conflicts with floating point values, the separator character in this file is a semicolon.

You can customize the output of the Bill of Materials in the **Settings** dialog box according to your preferred

- calculation algorithm (based on kits or individual parts),
- amount of module information,
- type of connectors (spring or screw), and
- extension cable length selection.

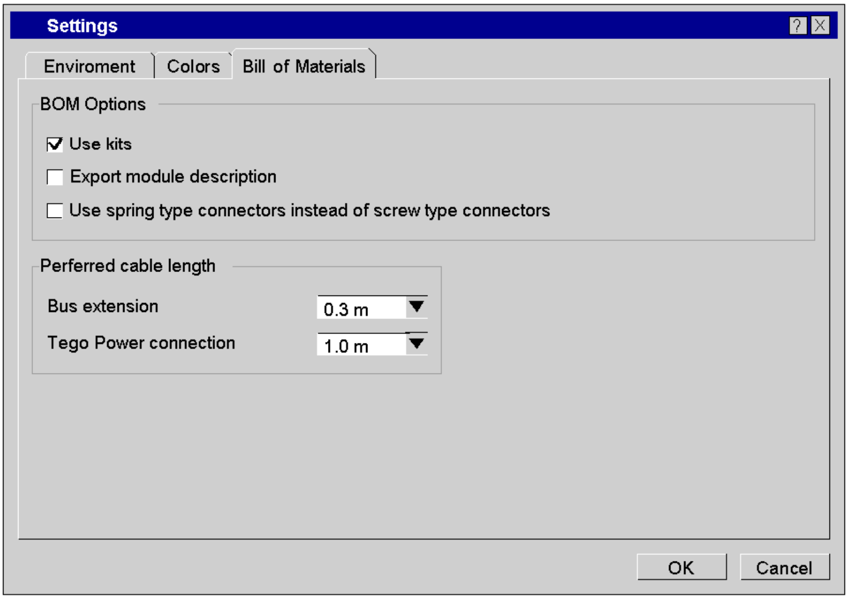
The default type of connectors is the screw one.

NOTE: Some modules do not support both connector types. In this case, the connector type supported by the module is listed in the Bill of Materials independent of the connector type you choose in the **Settings** dialog box.

You will find the **Settings** dialog box by selecting **Options** → **Settings**.

Settings Dialog Box

This figure shows the **Bill of Materials** tab in the **Settings** dialog box:



A change of the default information results in the following behavior:

If you select ...	Then ...
Use kits, which is checked by default	the components of the selected Island are summarized in form of kits for each module. This is indicated with a **K** after the module name. For the modules that do not support both connector types and thus kit types, the type is indicated with the following letters: ● **KS** (for screw type connectors) ● **KC** (for spring type connectors)
Export module description	the module descriptions are added to the module product references.
Use spring type Connectors ...	the default screw type connector product references are replaced by the equivalent spring type connector references.
a **Preferred cable length** that has a non-default value	the default cable product references are replaced by the appropriate ones.

Chapter 6
Reflex Actions Reference

Introduction

This chapter provides an overview of the reflex actions that can be created for STB Islands using the Reflex Editor of the Advantys Configuration Software.

What Is in This Chapter?

This chapter contains the following sections:

Section 6.1
General Information on Reflex Actions

Introduction

This section describes the general features and functions of the Advantys reflex actions. It lists the types and variations of reflex blocks that can be created using the Advantys Configuration Software and explains how 2 blocks may be combined in a nested reflex action.

What Is in This Section?

This section contains the following topics:

What Is a Reflex Action?

Summary

Reflex actions are small routines that perform dedicated logical functions directly on the Advantys Island bus. They allow output modules on the Island to act on data and drive field actuators directly, without requiring the intervention of the fieldbus master.

A typical reflex action comprises 1 or 2 function blocks that perform
- boolean AND or exclusive-OR operations,
- comparisons of an analog input value to user-specified threshold values,
- up- or down-counter operations,
- timer operations,
- the triggering of a latch to hold a digital value high or low,
- the triggering of a latch to hold an analog value at a specific value.

The Island bus optimizes the reflex response time by assigning the highest transmission priority to its reflex actions. Reflex actions take some of the processing workload off the fieldbus master, and they offer a faster, more efficient use of system bandwidth.

Behavior of Reflex Actions

Reflex actions are designed to control outputs independently from the fieldbus master controller. They may continue to turn outputs on and off even when the power is removed from the fieldbus master. The output state represented in the Island's network interface module (NIM) may not represent the actual states of the outputs. Use prudent design practices when you use reflex actions in your application.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION Do not depend on output values stored in the NIM when those values are controlled by reflex actions. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

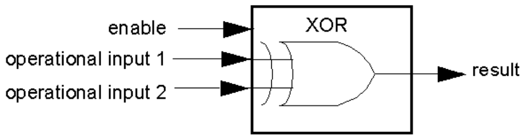
Configuring a Reflex Action

Each block in a reflex action has to be configured using the Advantys Configuration Software.

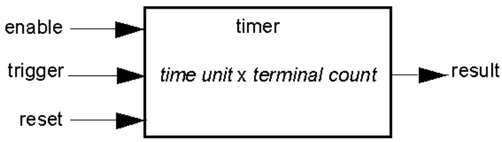
Each block has to be assigned a set of inputs and a result. Some blocks also require that you specify 1 or more user-preset values – a compare block, for example, requires that you preset threshold values and a delta value for hysteresis.

Inputs to a Reflex Action

The inputs to a reflex block include an enable input and 1 or more operational inputs. The inputs may be constants or they may come from other I/O modules on the Island, from virtual modules or outputs from another reflex block. For example, an XOR block requires 3 inputs – the enable and 2 digital inputs that contain the boolean values to be XORed:



Some blocks, such as the timers, require reset and/or trigger inputs to control the reflex action. The following example shows a timer block with 3 inputs:



The trigger input starts the timer at 0 and accumulates *time units* of 1, 10, 100 or 1000 ms for a specified number of counts. The reset input causes the timer accumulator to be reset.

An input to a block may be a boolean value, a word value, or a constant, depending on the type of reflex action it is performing. The enable input is either a boolean or a constant Always Enabled value. The operational input to a block such as a digital latch has to always be a boolean, whereas the operational input to an analog latch has to always be a 16-bit word.

You will need to configure a source for the block's input values. An input value may come from an I/O module on the Island or from the fieldbus master via a virtual module in the NIM.

NOTE: All inputs to a reflex block are sent on a change-of-state basis. After a change-of-state event has occurred, the system imposes a 10 ms delay before it accepts another change of state (input update).

Result of a Reflex Block

Depending on the type of reflex block that you use, it will output either a boolean or a word as its result. Generally, the result is mapped to an action module, as shown in the following table:

Reflex Action	Result	Action Module Type
Boolean Logic	boolean value	digital output
Integer Compare	boolean value	digital output
Counter	16-bit word	first block in a nested reflex action
Timer	boolean value	digital output
Digital Latch	boolean value	digital output
Analog Latch	16-bit word	analog output

The result from a block is mapped to an individual channel on an output module. Depending on the type of result that the block produces, this action module may be an analog channel or a digital channel.

When the result is mapped to a digital or analog output channel, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device.

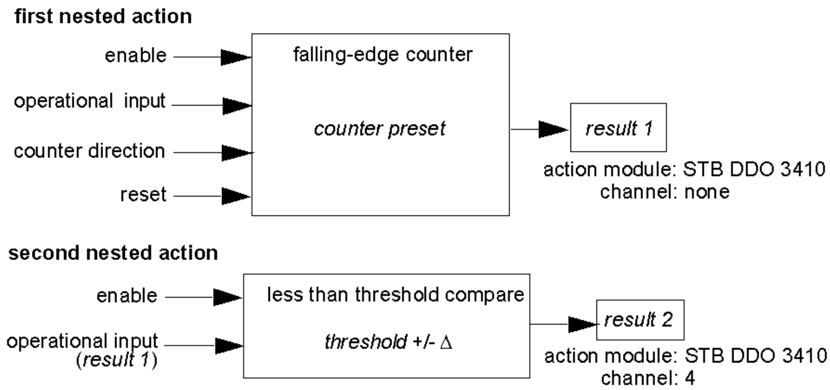
The exception is when a reflex block is the first of 2 actions in a nested reflex action.

Nesting

The Advantys Configuration Software allows you to create nested reflex actions. Only 1 level of nesting is supported, that means 2 reflex blocks, where the result of the first block is an operational input to the second block.

When you nest a pair of blocks, you need to map the results of both to the same action module. Choose the action module type that is appropriate for the result of the second block. This may mean that in some cases you will need to choose an action module for the first result that does not seem to be appropriate according to the table above.

For example, say you want to combine a counter block and a compare block in a nested reflex action. You want the result of the counter to be the operational input to the compare block. The compare block will then produce a boolean as its result:



Result 2 (from the compare block) is the result that the nested reflex action will send to an actual output. Because the result of a compare block needs to be mapped to a digital action module, *result 2* is mapped to channel 4 on an STB DDO 3410 digital output module.

Result 1 is used only inside the module. It provides the 16-bit operational input to the compare block. It is mapped to the same STB DDO 3410 digital output module that is the action module for the compare block.

Instead of specifying a physical channel on the action module for *result 1*, the channel is set to None. In effect, you are sending *result 1* to an internal reflex buffer where it is stored temporarily until it is used as the operational input to the second block. You are not really sending an analog value to a digital output channel.

Number of Reflex Blocks on an Island

An Island can support up to 10 reflex blocks. A nested reflex action consumes 2 blocks.

An individual output module can support up to 2 reflex blocks. Supporting more than 1 block requires that you manage your processing resources efficiently. If you are not careful with your resources, you may be able to support only 1 block in an action module.

Processing resources are consumed quickly when a reflex block receives its inputs from multiple sources (different I/O modules on the Island and/or virtual modules in the NIM).

The best way to preserve processing resources is to

- use the Always Enabled constant as the enable input whenever possible and to
- use the same module to send multiple inputs to a block whenever possible.

Overview of Reflex Action Types

Summary

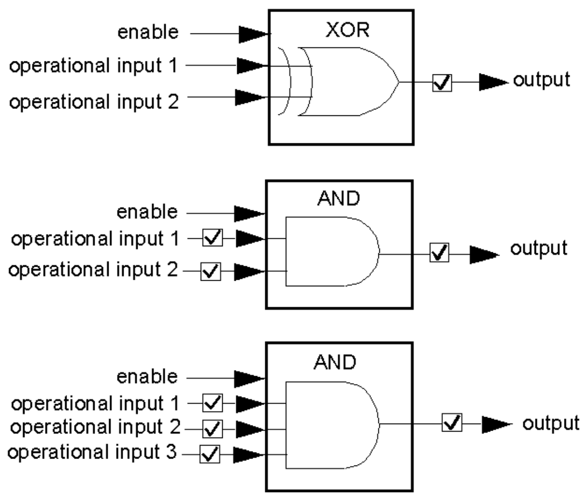
There are 7 types of reflex blocks available in the Advantys Configuration Software:
- boolean logic blocks *(see page 338)*
- integer compare blocks *(see page 352)*
- unsigned compare blocks *(see page 369)*
- counter blocks *(see page 388)*
- timer blocks *(see page 401)*
- digital latches *(see page 437)*
- analog latches *(see page 422)*

Each block supports a series of variations called *action types*.

Boolean Logic Action Types

The Advantys Configuration Software supports the following 3 fundamental boolean logic action types:
- exclusive-OR (XOR) block
- 2-input AND block
- 3-input AND block

The blocks are illustrated below:



Boolean logic blocks require 2 types of inputs, an enable input and 2 or 3 operational inputs. All the inputs need to be digital (boolean) values from sources that you have to specify in the Reflex Editor. The output from any of these action types is also a boolean value.

Notice the check boxes on the operational input lines to the AND blocks and on the output lines from all the boolean blocks. When you place a check mark in 1 or more of these boxes, you invert the input or output value(s). When you invert an input to an action block, a value of 0 is treated as a 1 and a value of 1 is treated as a 0. In other words, you turn a boolean false condition into a boolean true condition or vice versa. If you invert the output from an XOR block, it becomes an XNOR action; if you invert the output from an AND block, it becomes a NAND action.

Because of all the possibilities that can result from combinations of standard and inverted inputs and outputs, there are a large number of variations to the 3 basic boolean action types. These variations are illustrated in truth tables.

Compare Action Types

A compare block takes a word as its operational input and compares that value with a predefined threshold value or a window of values. An integer compare block accepts operational inputs with integer values in the range -32,768...+32,767. An unsigned compare block accepts operational inputs with integer values in the range 0...65,535.

Bear also the following points in mind:

- Integer compare blocks generally take their operational inputs from Advantys STB analog input modules. Advantys analog modules use the IEC format for handling data. In this format, the most significant bit is always a dedicated sign bit, and the remaining 15 bits are able to represent values up to 32,767.
- Unsigned compare blocks generally take their operational inputs from virtual modules *(see page 326)* or from the outputs produced by counter reflex actions *(see page 333)*. These input sources produce unsigned values with 16-bit resolution (values as high as 65,535).

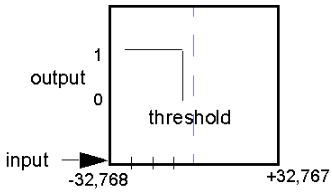
Integer compare blocks and unsigned compare blocks both support 4 action types:

Action Type	The output is a boolean 1 when the operational input value is ...
Less-than-Threshold Compare	less than a user-defined threshold value.
Greater-than-Threshold Compare	greater than a user-defined threshold value.
Inside-the-Window Compare	within the range of values bound by 2 user-defined thresholds.
Outside-the-Window Compare	outside the range of values bound by 2 user-defined thresholds.

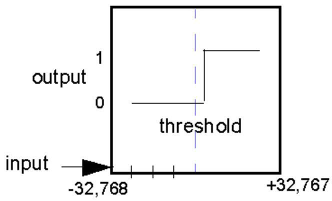
Behavior of the Compare Blocks

The following illustrations show how the 4 compare action types compare the input to the thresholds, using the integer compare block as an example:

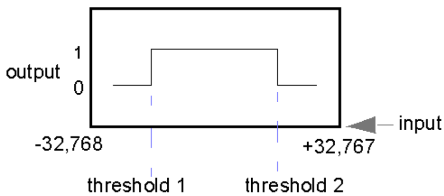
Less-than-threshold compare block:



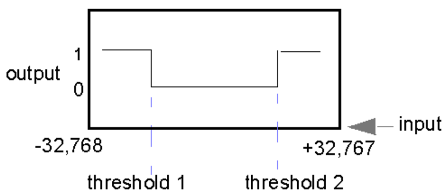
Greater-than-threshold compare block:



Inside-the-window compare block:



Outside-the-window compare block:



For all of the above action types, you may also specify a delta (Δ) value, which acts as an hysteresis around the threshold value(s).

The integer compare action types are described in *Integer Compare Reflex Blocks,* *page 352*. The unsigned compare action types are described in *Unsigned Compare Reflex Blocks,* *page 369*.

Counter Action Types

A counter block takes a series of digital inputs and accumulates a running count of the number of transitions either from 0 to 1 or from 1 to 0. You can configure the counter block to count up or down from a user-specified preset value. The output from the block is the current count – an unsigned integer value in the range 0...65,535.

Counter blocks support 2 action types:

Action Type	The counter increments or decrements each time the input value transitions from ...
Rising-Edge Counter	0 to 1
Falling-Edge Counter	1 to 0

NOTE: Counter blocks are different from other reflex actions in that they never map their output results to physical analog output channels. A counter block is designed to be coupled with an unsigned compare block in a nested reflex action *(see page 333)*. The counter block is always the first block in the nested action, and its output is used as the operational input to the compare block.

Timer Action Types

Timer blocks support 4 action types:
- delay-to-start timers
- delay-to-stop timers
- rising-edge timers
- falling-edge timers

The timer blocks respond to a digital trigger input. A block begins accumulating time units on either the rising edge or falling edge of the trigger input and accumulates counts until it reaches a user-specified terminal count.

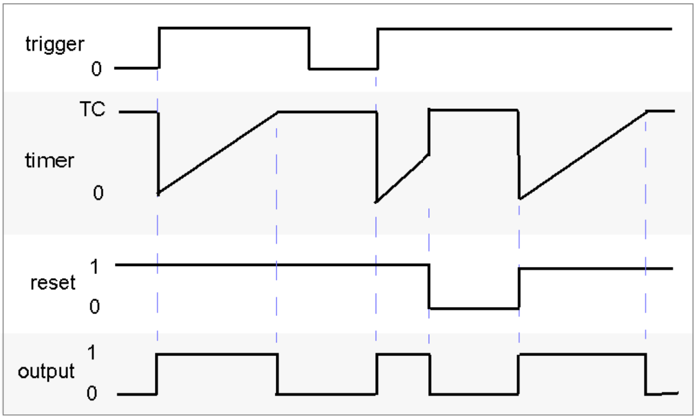
The outputs from all 4 timer action types may be inverted. When you invert an output from an action block, a value of 0 is treated as a 1 and a value of 1 is treated as a 0; in other words, you turn a boolean false condition into a boolean true condition or vice versa.

Behavior of the Edge Timers

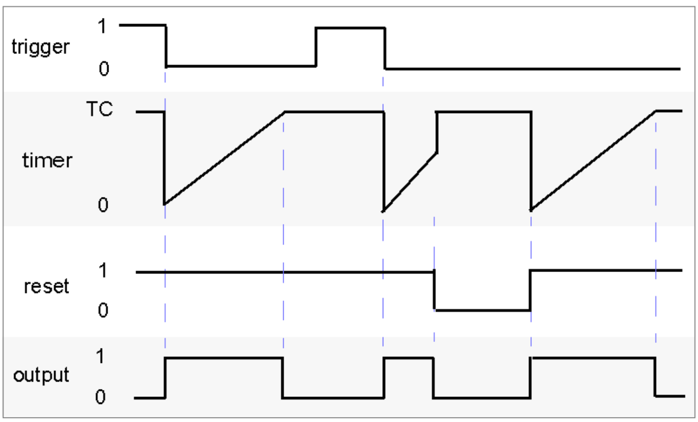
For a rising- or falling-edge timer, the accumulator holds the terminal count until the rising or falling edge of the trigger starts a new counting operation or until the block receives a reset input.

The output from an edge timer goes high while the timer is accumulating time counts and goes low when the terminal count is reached.

Rising-edge timer:



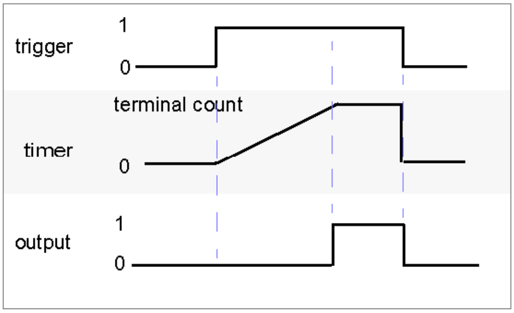
Falling-edge timer:



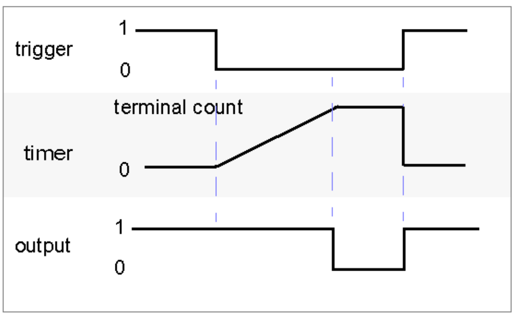
Behavior of the Delay Timers

The output from a delay timer goes high or low when the timer reaches its terminal count and stays high or low while the terminal count is being held.

Delay-to-start timer:



Delay-to-stop timer:



Latch Types

Latch blocks respond to a digital trigger input by latching to an operational input value on either the rising edge or falling edge of the trigger input. The block produces an output that is equal to value of the input at the moment it was latched, and that output remains until the trigger latches another value on its rising or falling edge. The operational input may be either of boolean values (digital latches) or word values (analog latches).

Digital and analog latches both support 4 action types:

Action Type	The block latches the output value to the value of the operational input when the trigger ...
Rising-Edge Latch	transitions from 0 to 1.
Falling-Edge Latch	transitions from 1 to 0.
Low-Level Latch	is at 0 and unlatches the output when the trigger is at 1.
High-Level Latch	is at 1 and unlatches the output when the trigger is at 0.

When an output is unlatched, the value of the output echoes the value of the operational input.

The output from a digital latch may be inverted. The output from an analog latch cannot be inverted. When you invert an output from a digital latch block, a value of 0 is treated as a 1 and a value of 1 is treated as a 0; in other words, you turn a boolean false condition into a boolean true condition or vice versa.

Configuring a Reflex Block

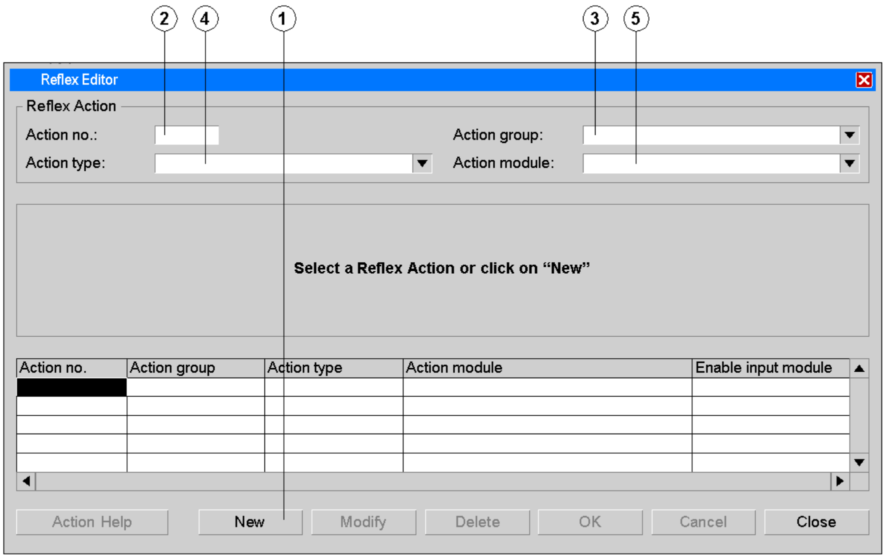
Summary

To create a reflex block and map it to an action module on your Island bus, you need to use the Reflex Editor in the Advantys Configuration Software. The following procedure describes the basic parameters that need to be specified in the editor.

Opening the Reflex Editor

To open the Reflex Editor, click the following icon on the **Island** toolbar:

If no reflex action is configured, the Reflex Editor is displayed as follows:



1 Button for Creating a New Reflex Action
2 Field for the Number of the Reflex Action
3 Field for the Action Group
4 Field for the Action Type
5 Field for the Action Module

NOTE: If the **New** button in the Reflex Editor is disabled, the Island selected in the Workspace is locked.

To unlock the Island, close the Reflex Editor and click the key icon on the **Island** toolbar:

Some Island configurations are password protected. If the configuration on which you are working is protected, you will need to enter the password to unlock the Island. If the configuration is not protected, it will unlock as soon as you click the key icon once.

Defining the Reflex Block

The following steps describe how to select and define a reflex block and its action module in the Reflex Editor:

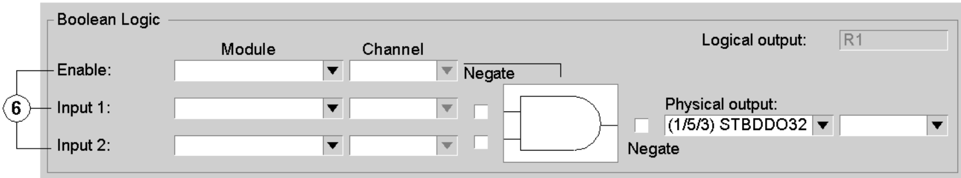
Step	Action	Result
1	Click the **New** button.	The **Action group:** field is selected.
2	From the **Action group:** box, select 1 of the seven reflex blocks *(see page 315)*.	The **Action type:** field is selected.
3	From the **Action type:** box, select the appropriate block type.	A block diagram appears in the center pane of the Reflex Editor with empty fields for the inputs, outputs, and any user-specified preset values. The **Action module:** field is selected.
4	From the **Action module:** box, select an output module from your Island bus configuration.	The module you specify here automatically appears in the **Physical output:** list box in the block diagram in the center pane of the editor.

NOTE: The number of the reflex action is automatically assigned when you finish configuring the reflex block by clicking **OK**, see Finishing *(see page 325)*.

Now, you can configure the action's input values and output destination. All reflex blocks require a set of input values, a physical output, and a logical output, as described in the following discussions.

Configuring the Inputs to a Reflex Block

Every block requires that you configure a set of input values. The block diagram that appears in the center pane of the Reflex Editor displays the input fields in a column on the right. The following example shows a 3-input AND block:



6 Input Fields

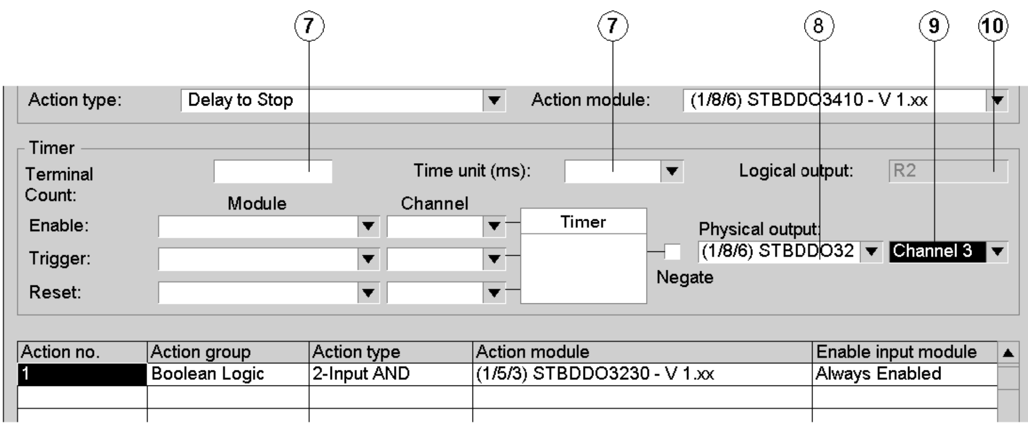
This example shows a block with 3 inputs: an **Enable** and 2 operational inputs (**Input 1** and **Input 2**). Each input has its own drop-down list, from which you will configure the source of each input.

Generally, inputs can be derived from 1 of 4 sources:
- from another input module on the Island bus
- from a constant value that you specify (for instance always enabled, up-count direction)
- from the fieldbus master, in the form of the virtual module *(see page 326)* or the action module *(see page 330)*
- from a reflex block

Configuring Preset Values for a Reflex Block

Some reflex actions also have some user-specified preset values that you will need to configure. For example, a timer block requires a timing unit and a terminal count preset. When preset values are required, the Reflex Editor displays them above the reflex block (as in item 7 below).

The following example shows the block display for a delay-to-stop timer:



7 Fields for Preset Values
8 Field for the Name of the Physical Output
9 Field for the Channel of the Physical Output
10 Field for the Logical Output

Notice the **Terminal Count:** and the **Time unit:** field for the preset values.

Configuring the Physical Output from a Reflex Block

Notice in both examples above that the module listed in the **Physical output:** field is the module you have chosen in the **Action module:** field. The physical output module is always the action module. You need to specify the channel on the action module to which the physical output will be written:

If you want to ...	Then choose ...	Result:
map the output from the action to a real physical output	a channel number.	**Result:** The physical channel will be dedicated to the reflex action.
configure the first of 2 blocks in a nested reflex action	**None**.	**Result:** The output will be written to a temporary memory buffer, and then used as an input to the second block in the nested reflex action.

Finishing the Reflex Block Configuration

If you have entered all information required to configure the reflex block, click **OK** in the Reflex Editor to finish the configuration.

Then, the software automatically assigns the following:
- a number to the reflex action
- a tag name to the output (ranging from R1 to R10)

The number of the reflex action is displayed in the **Action no.:** field, the tag name for the output in the **Logical output:** field. Both fields cannot be edited. The logical output is particularly useful in a nested reflex action because the string from the first reflex action appears in the drop-down lists as an input channel selection item for the inputs to the second reflex action.

Virtual Module

Summary

Because reflex actions are designed to operate independently from the fieldbus master, inputs to the reflex blocks generally come from local input modules. In some applications, however, you may want the fieldbus master to provide an input value to a block. You can realize this using the virtual module.

The Advantys Configuration Software provides 3 words in the output process image where the fieldbus master may write digital and/or analog values for use exclusively as inputs to the reflex actions. These 3 words comprise the *virtual module*.

Virtual Module Structure

If you choose to use the virtual module, it may be 1, 2 or 3 words in length:

If you want to use the virtual module ...	Then the virtual module will be ...
only for digital inputs	1 word long. It provides 16 bits where the fieldbus master can write up to 16 digital inputs to the reflex actions.
only for analog inputs	2 words long. It will provide 2 words where the fieldbus master can write up to 2 analog input values to the reflex actions.
for both digital and analog inputs	3 words long. The first word provides 16 bits for digital inputs and the second and third words are for 2 analog input values to the reflex actions.

NOTE: If you look at the virtual module data in the **Modbus Image** tab of the **I/O Image Overview** window, the 16 bits of digital virtual data will be displayed in the low bytes of 2 separate registers. If you look at the virtual module data in the **Fieldbus Image** tab, the 16 bits of digital virtual data may be displayed either in a single 16-bit word or in the low bytes of 2 contiguous words, depending on the fieldbus. The STB NMP 2212 Modbus Plus NIM and the STB NIP 2212 Ethernet NIM display virtual digital data the way they are displayed in the **Modbus Image**.

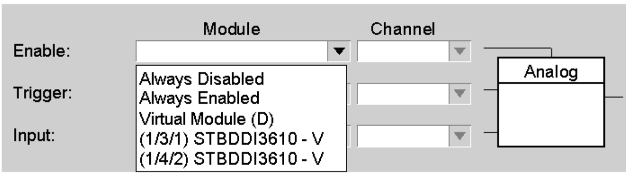
The word or words used for the virtual module are always the last words in the output process image. If all 3 words are used, the digital word will appear first followed by the 2 analog words.

Selecting the Virtual Module

The size of the virtual module in your process image is determined by your selection of inputs to the reflex actions in your Island configuration.

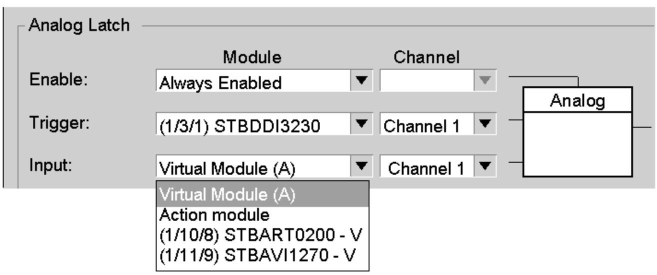
For example, suppose you are setting up a falling-edge analog latch *(see page 423)*. The latch has 3 inputs: an enable input, a latch trigger and an analog operational input. The enable input and the trigger need to be boolean (digital) inputs, and the operational input needs to be an analog word.

When you are selecting the source for the enable and the trigger inputs, 1 of the choices that will appear in both input list boxes is **Virtual Module (D)**:



If you select an input called **Virtual Module (D)**, the virtual-module word for digital inputs becomes part of the output process image. This means that the fieldbus master needs to write to 1 of the virtual module's 16 available bits to control the enable input and/or the trigger input.

Suppose you are configuring the operational input to that falling-edge analog latch. The operational input has to be an analog integer value. When you go to the list box to specify the source of the operational input, 1 of your choices will be **Virtual Module (D)**:



If you select **Virtual Module (A)** as the input, the 2 virtual-module words for analog inputs become part of the output process image. The fieldbus master needs to write an operational input to the first word in the analog latch block.

Action Module

Summary

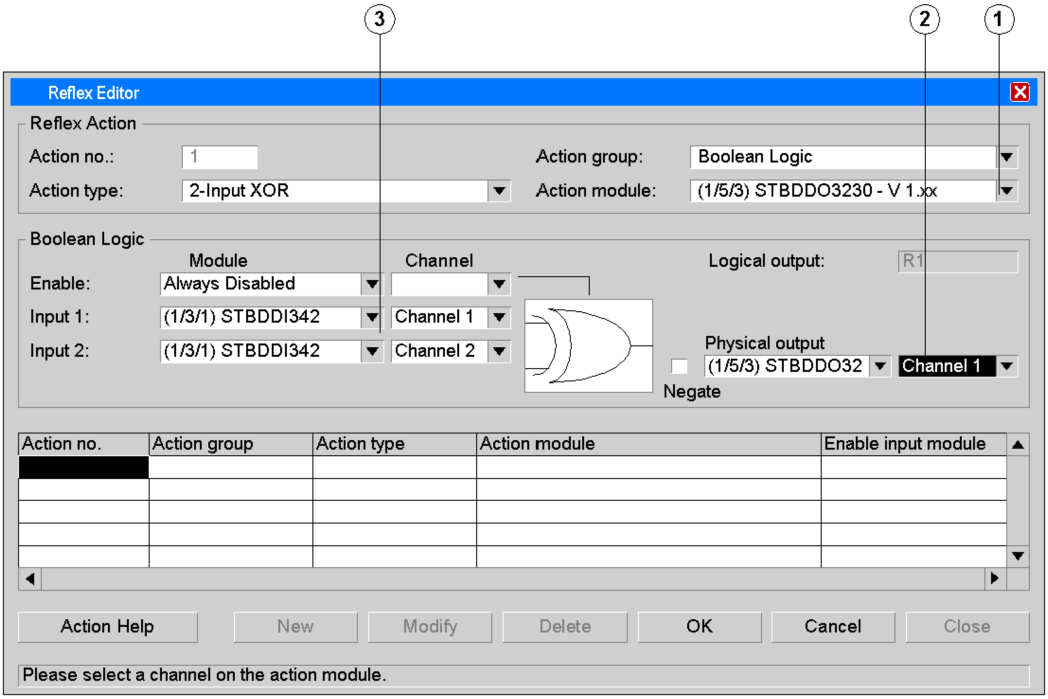
When you configure a reflex block, assign it to an action module. The action module is always 1 of the output modules in your Island configuration. There is a direct relationship between the action module that you select and the type of output that the reflex action will produce. If the reflex action produces a boolean result as its output, the action module is generally a digital output module. If the reflex action produces an analog output, an analog output module is generally the action module.

The exception is when you nest 2 reflex blocks together. In that case, both actions need to have the same action module, and the action module type needs to match the output expected from the second block in the nested action.

Mapping a Reflex Output to a Physical Output

When you configure a reflex block to write its output to a field actuator, choose an action module and specify the channel on the action module that will send the output to the actuator.

For example, suppose you want to configure a boolean XOR action that writes outputs to a field actuator connected to channel 1 on an STB DDO 3230 output module. Here is how the configuration might look in the Reflex Editor:



1 Action Module Selected from the List Box
2 Action Channel of the Physical Output
3 Boolean Inputs

This action is designed to XOR the boolean inputs produced on channel 1 and channel 2 of the STBDDI3420 digital input module at address 3 on the Island bus (item 3 above). The output from the action is written to channel 1 on the STBDDO3230 digital output module.

Item 1 above shows the action module selected from the list box. The entry lists 3 things:

- the model number of the action module
- the version of the module (V 1.xx)
- a position code (1/5/3)

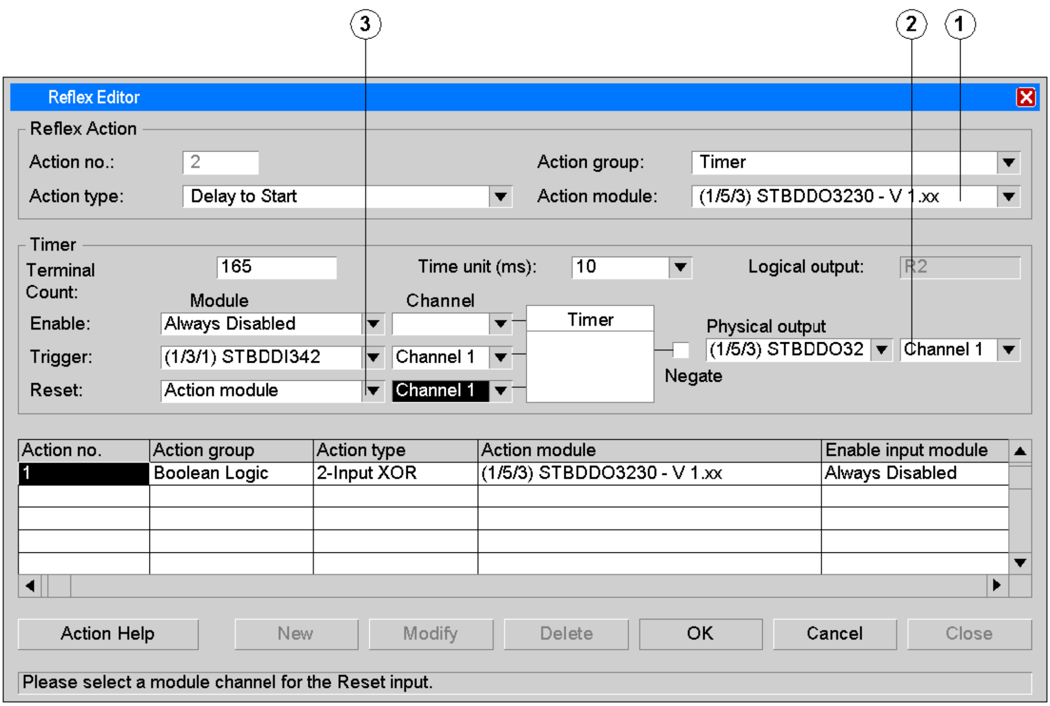
The position code tells you that the STBDDO3230 module that you have selected as your action module is located in the primary segment (1), at physical location 5 and logical address 3 on the Island bus. The discrepancy between the logical address and the physical location is caused by the presence on the Island bus of the NIM and a PDM, 2 modules that do not have logical addresses.

Item 2 above indicates exactly where the configuration will map the reflex output. The physical output module is the action module. The action channel is selected from the list box on the right. Since an STBDDO3230 is a 2-channel output module, the channel choices in the list box are **None**, **Channel 1** and **Channel 2**. For this configuration, channel 1 has been selected as the action channel.

Using the Action Module as an Input to a Block

Once you have mapped the output from a block to a physical channel on the action module and downloaded the configuration, this channel becomes dedicated to that reflex action. The fieldbus master can no longer drive the physical output. However, the channel address is still present in the output process image, and the fieldbus master can write data to this address location. You may use this channel address to deliver fieldbus data as an input to a reflex block.

For example, suppose you want to configure a delay-to-start timer *(see page 402)*, and you want the fieldbus master to provide the reset input to the reflex block. Here is how the configuration might look in the Reflex Editor:



1 Action Module
2 Action Channel of the Physical Output
3 Reset Input List Box

NOTE: This is 1 way of preserving processing resources *(see page 314)*, since you reuse resources from an output module that is already involved in the reflex action.

The action module is specified as the STBDDO3230 output module (item 1 above). The physical output is mapped to channel 1 of the action module (item 2 above).

For efficiency, you may reuse the bit in the output process image previously assigned to channel 1 of the action module as the reset input. Functionally, this means that the fieldbus master will be able to reset (stop) the timer accumulator by writing a value of 0 to that data bit in the output process image. To make this happen, select **Action module** from the **Reset** list (item 3 above), and then select **Channel 1** as the reset input channel.

Selecting None as the Physical Output Channel

In the 2 examples above, you always selected a channel (1, 2, 3, etc.) along with the action module as the physical output from the reflex action.

NOTE: Among the channel options in the list box is the entry **None**. Select this entry only when the action block you are configuring is the first block in a nested reflex action *(see page 333)*.

NOTE: When you select **None** as the physical output channel, the output from the block goes to a temporary storage buffer, and it can be used as an input to the second block in the nested reflex action.

Response of Action Modules to Fallback Conditions

Fallback Conditions

Advantys STB output modules are designed to send their output data to a predictable fallback state in the event of a communication interruption between the Island and the fieldbus. In this state, output data is replaced with pre-configured fallback values so that a module's output data values are known when the system recovers from a communication interruption.

Because reflex blocks are able to operate independently from the fieldbus master, there are some circumstances where the fallback scenario for an action module will be different from an output module that does not involve reflex actions. The following discussion points out these circumstances.

Action Module Behaviors

An action module is an Advantys STB output module that has at least 1 of its channels dedicated to the result of a reflex block. Typically, an action module will behave like any other output module on the Island. It will send its output channels to their configured fallback states when communication between the Island and the fieldbus master is lost.

An action module is not in a situation where its regular output channels are in their configured fallback states while a channel dedicated to the reflex action continues to operate.

The exception is a 2-channel action module where
- each output channel supports an independent reflex block and
- neither reflex block is receiving any inputs from the NIM; that means that inputs to the reflex blocks are not coming from the virtual module *(see page 326)* or from the action module itself *(see page 330)*.

If both these conditions are true, the action module will continue to run if communication between the Island and the fieldbus master is lost.

If the action module has more than 2 output channels, the behavior described above does not apply. An Advantys STB output module cannot be configured to support any more than 2 reflex blocks.

NOTE: Please consult the *Advantys STB Reflex Actions Reference Guide* for further information.

PDM Power Loss Detection

If an input module on the Island bus is providing an input to a reflex block and that input module loses sensor power from the PDM, the reflex block immediately acts upon a 0 value coming from that input. After a delay of up to 1.5 ms, the reflex action acknowledges that PDM power has been lost and puts the reflex channel to its fallback state.

NOTE: When an input module is used to control the enable input to a reflex action, the reflex block will act as if the enable has gone to 0 if the input module stops abruptly.

For more information about fallback conditions, refer to the output module descriptions in the *Advantys STB Hardware Components Reference Guide*.

Nesting 2 Reflex Blocks

Summary

The Advantys Configuration Software allows you to create 1 level of nesting for reflex actions. You can nest 2 reflex blocks, where the output from the first block is used as an operational input to the second block. Both reflex blocks have to be nested within the same action module.

Action Module

In a nested reflex action, the output from the first reflex block is used internally as an operational input to the second reflex block. The output from the second reflex block is used to update the physical output channel of the action module.

When you nest a pair of reflex blocks, you need to map the outputs from both to the same action module. Choose the action module type that is appropriate for the output from the second nested action block. In some cases, this means that you may need to choose an action module for the first block that does not seem to be appropriate for its output.

For example, say you want to nest counter-compare action. To do this, you need to configure 2 action blocks using the Reflex Editor. The first block is the counter action *(see page 388)*, and the second block is an unsigned compare action *(see page 369)*.

The output from a counter is always a 16-bit word value, and the output from the unsigned compare is always a binary (boolean) value. Intuitively, you might assume that because the counter produces a word as its output it should be mapped to an analog action module. However, since the counter is the first block in the nested action and since the output from the second action, the unsigned compare, is a boolean, you need to select a digital output module as the action module.

Physical Outputs

The Reflex Editor requires you to specify the physical and logical output of each reflex block that you configure. Generally, the physical output is the channel on the action module to which the output of the action will be written. The physical output is always mapped this way when an action is not part of a nesting; it is also how the output from the second action block in a nested action is mapped. For the first block in a nested action, however, the physical output is sent to a temporary memory buffer. Instead of specifying an output channel on the action module, you need to specify the physical output as None.

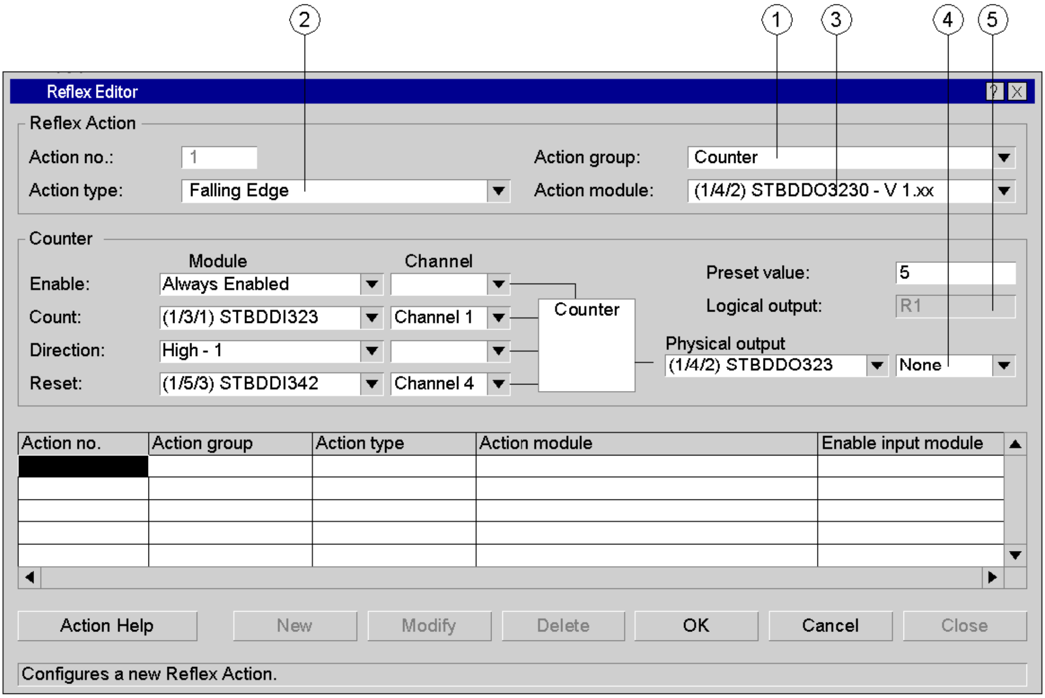
Logical Outputs

The output from each block also needs to be assigned a logical output. The logical output is a tag name for the output: a text string between 1 and 8 characters long. The characters may be any combination of standard keyboard characters: alphanumerics, underscores, and/or standard symbols (!,?, /, >, etc.).

The logical output can be particularly useful in a nested reflex because the text string of the first action block will appear on the menu as an input to the second action block.

Counter-Compare Configuration

To clarify the process of configuring a nested action, let us look at the way you might configure the first of the 2 action blocks in the Reflex Editor of the Advantys Configuration Software:

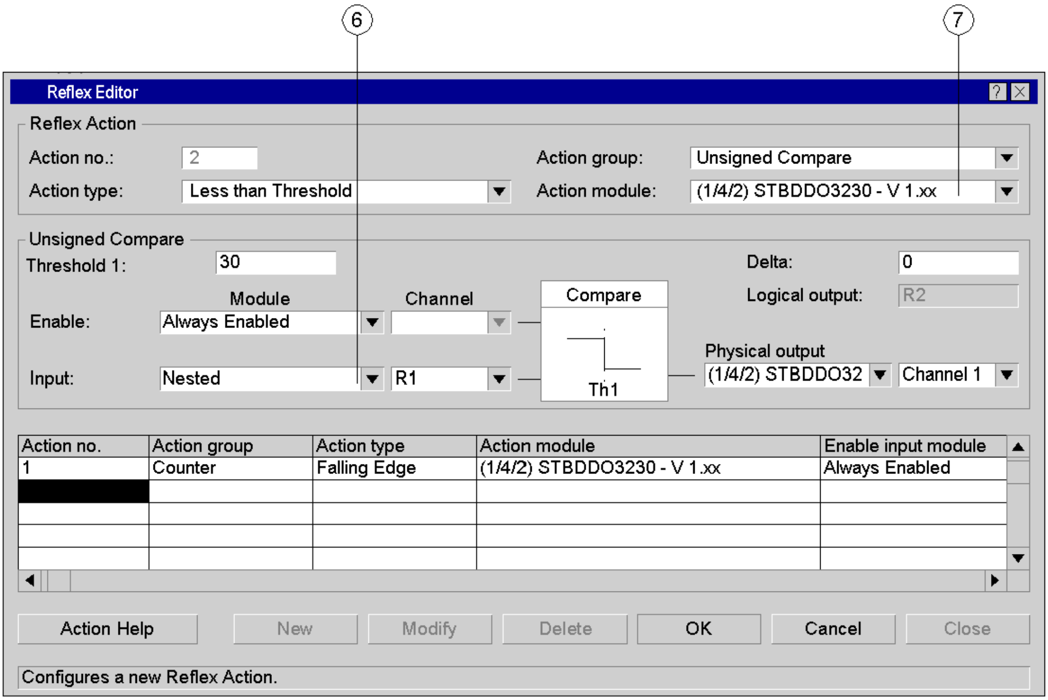


1+2 Action No 1 = Falling-Edge Counter
3 Action Module = STBDDO3230 Digital Output Module
4 Physical Output Channel of the Action Module = None
5 Logical Output = R1

Action number 1 is a falling-edge counter, as items 1 and 2 above indicate. The action module is the STBDDO3230 digital output module at Island bus address 2 (item 3 above). The action module needs to be a digital output module because the ultimate result of the nested action will be boolean. The action module selected is automatically displayed in the **Physical output:** field.

Item 4 above shows the output channel on the action module as **None**. The output from the falling-edge counter is sent to a temporary memory buffer. The string **R1** (in this example) is automatically assigned to the output value in this temporary memory buffer, as shown in the **Logical output:** (item 5 above).

The logical output from the first block will be used as the operational input to the second block, as shown in the following illustration:



6 Action No 2 = Unsigned Less than Threshold Compare Block
7 Action Module = STBDDO3230 Digital Output Module

Action number 2 is an unsigned less-than-threshold compare block. Item 6 (**Input** row, **Channel** list) shows that the operational input to the compare block is **R1**, the logical output from action number 1. The action module (item 7 above) for the less-than-threshold compare block is the STBDDO3230 digital output module at Island bus address 2, which is the same action module as the one for the falling-edge counter.

Reflex Action Start-Up States

Summary

All reflex blocks are initially at fallback when the Island starts up after a power cycle or any other reconfiguration sequence. However, the fallback mode and fallback value applied to each output channel is the factory-default (*predefined* state, *off*), not the user-configured parameters downloaded with the configuration. The user-configured parameters are applied only after all inputs have been received and a condition that triggers fallback occurs.

After all the reflex inputs have been received (even with a detected error in status), the reflex blocks will enter run state. If there is a detected error in status, that reflex block output channel will enter fallback with the user-configured value.

Enable inputs have the same effect as normal inputs for the purposes of entering and leaving the fallback mode.

Consequences

Some of the consequences of this start-up state behavior are the following:

- If 1 or more peer input module(s) is/are missing, no reflex blocks in the module will run. The factory-default fallback mode and fallback values will remain in effect.
- Issuing a **Stop** and a **Run** command from the Advantys Configuration Software will reset the reflex blocks so that the output channels will start in their user-configured fallback modes and states.
- Removing a mandatory module from the Island and then replacing it will also reset the reflex blocks so that the output channels will start in their user-configured fallback modes and states.

NOTE: Please consult the *Advantys STB Reflex Actions Reference Guide* for further information.

For detected errors in reflex to be cleared (as indicated by the LED and the Advantys Configuration Software described below), all configured reflex blocks have to be successfully executed. This requires that all input data are present (without any detected errors in status) and that the *enable* input is high at least once during the period when all input data are present.

Reflex Action LED Diagnostic State

When a reflex block is in inoperable or is not running because all its inputs have not been received, the green RDY LED on the action module will blink in a special pattern: 3 blinks followed by a pause, repeatedly until the condition is cleared.

Detected errors in reflex are also indicated by emergency messages and emergency diagnostic codes. These detected errors appear in the Advantys Configuration Software as a detected error in node (diagnostic register = 0*x*80 in the **I/O Module diagnostics** window).

Enable Behavior

The table below describes the enable behavior of reflex blocks:

<table>
<tr><th>If the enable input to a reflex action is...</th><th>Then the block ...</th></tr>
<tr><td>the Always Enabled constant</td><td>always becomes operational immediately at start-up.</td></tr>
<tr><td>the Always Disabled constant</td><td>always starts up in its fallback state. The action module flashes the fallback LED pattern described above for the action channel. Other non-reflex outputs remain operational. The module sends an emergency message indicated by the node diagnostics bit. If the Advantys Configuration Software is connected to the physical Island, the module image in the Island Editor will flash in red. The detected error in that block will never clear. The Always Disabled constant might be used while you are commissioning the Island.</td></tr>
<tr><td>the signal from an input module on the Island bus</td><td rowspan="2">starts up in its fallback state when the input value is 0. The action module flashes the fallback LED pattern described above for the action channel. Other non-reflex outputs remain operational. The module sends an emergency message indicated by the node diagnostics bit. If the Advantys Configuration Software is connected to the physical Island, the module image in the Island Editor will flash in red. As soon as the enable input is sensed, the reflex block becomes operational, the LED goes on steady, and the module image in the Island Editor stops flashing in red.</td></tr>
<tr><td>is either part of the action module or the digital virtual module</td></tr>
</table>

Section 6.2
Boolean Logic Reflex Blocks

Introduction

This section describes 3 boolean logic reflex blocks, an exclusive-OR (XOR) and 2 logical ANDs. XOR blocks operate on 2 input values; AND blocks can operate on either 2 or 3 inputs. Because the software allows you to invert the results of these blocks and sometimes their operational inputs, several variations of the 3 block types are supported.

What Is in This Section?

This section contains the following topics:

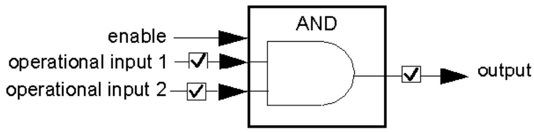
2-Input AND Blocks

Summary

A 2-input AND block performs a logical AND operation on 2 boolean operational inputs. The output is a boolean true or false, expressed as a value of 1 or 0. You may invert the value(s) of 1 or both inputs. You may also invert the value of the output, in which case the action becomes a logical NAND.

Structure of a 2-Input AND Block

A block diagram for a 2-input AND is shown below:



The AND block has 3 inputs:

Input	Description
Enable Input	turns the block on or off
Operational Input 1	sends a boolean value to the block
Operational Input 2	sends a boolean value to the block

All inputs enter into an AND operation when the block is enabled, and the result is a boolean output.

The check boxes on the 2 input lines and the output line provide the mechanism by which 1 or more of the values can be inverted. When you click 1 of these boxes, a check mark toggles on or off. When a box is checked, the value of the associated input or output is inverted, that means a 1 becomes a 0 and a 0 becomes a 1.

Enable Input

An AND block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)* or
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block was disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Operational Inputs

Every 2-input AND block requires 2 operational input values. Each input is a boolean 1 or 0.

These inputs may come from some combination of the following:
- constant values,
- digital inputs from modules on the Island,
- digital outputs from the virtual module *(see page 326)*,
- an output on the action module *(see page 330)* written to by the fieldbus master,
- the output from the first reflex block if the AND is the second block in a nested reflex action *(see page 333)*. In this case, 1 of the operational inputs may be the output of the first reflex block.

Physical Output

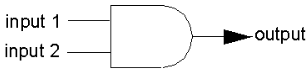
The output from a 2-input AND block is a boolean true (1) or false (0), as shown in the truth tables that follow. The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the AND is the first block in a nested reflex action	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

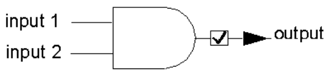
When the output of a reflex block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write to this bit address in the NIM, and the reflex action editor lets you use these data from the fieldbus master as an input to the block.

Truth Tables

In its simplest form, a 2-input AND block looks like this:



And an inverted AND (a NAND) block looks like this:



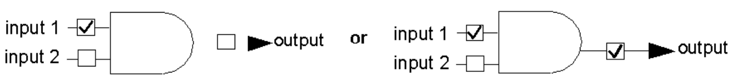
The following truth table shows the possible outputs of this AND operation:

If input 1 is...	and input 2 is...	Then the standard output is...	and the inverted output is...
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Inverted Operational Inputs

You can invert 1 or both of the operational inputs. An inversion is indicated in the Advantys Configuration Software as a check mark in a box on an input line.

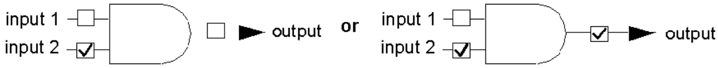
Input 1 is inverted:



When input 1 is inverted, the truth table yields the following:

If input 1 is...	and input 2 is...	Then the standard output is...	and the inverted output is...
0	0	0	1
0	1	1	0
1	0	0	1
1	1	0	1

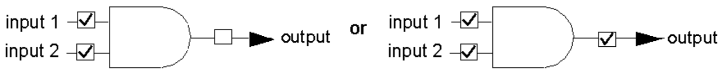
Input 2 is inverted:



When input 2 is inverted, the truth table yields the following:

If input 1 is...	and input 2 is...	Then the standard output is...	and the inverted output is...
0	0	0	1
0	1	0	1
1	0	1	0
1	1	0	1

Both inputs are inverted:



When both inputs are inverted, the truth table yields the following:

If input 1 is...	and input 2 is...	Then the standard output is...	and the inverted output is...
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	1

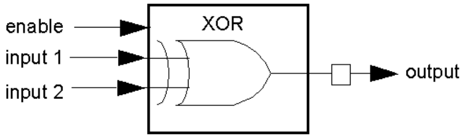
XOR Blocks

Summary

An XOR block performs an exclusive-OR operation on 2 boolean operational inputs. The output from the block is boolean true or false, expressed as a value of 1 or 0. You may invert the value of the output, in which case the action becomes an exclusive-NOR (XNOR).

Structure of an XOR Block

An XOR block diagram is shown below:



The XOR block has 3 inputs:

Input	Description
Enable Input	turns the block on or off
Operational Input 1	sends a boolean value to the block
Operational Input 2	sends a boolean value to the block

All inputs enter into an XOR operation when the block is enabled, and the result is a boolean output.

The check box on the output line provides the mechanism by which the output value may be inverted. When you click this box, a check mark toggles on or off. When the box is checked, the value of the associated output is inverted, that means a 1 becomes a 0 and a 0 becomes a 1.

Enable Input

An XOR block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Operational Inputs

Every XOR block requires 2 operational input values. These inputs may come from some combination of the following:

- constant values,
- digital inputs from modules on the Island,
- digital outputs from the virtual module *(see page 326)*,
- an output on the action module *(see page 330)* written to by the fieldbus master,
- the output from the first reflex block if the XOR is the second block in a nested reflex action *(see page 333)*. In this case, an operational input may be the output from the first reflex block.

Physical Output

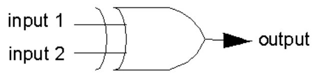
The output from an XOR block is a boolean true (1) or false (0), as shown in the truth tables that follow. The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the XOR is the first block in a nested reflex action	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

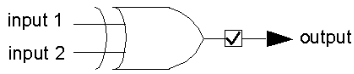
When the output of a reflex block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write to this bit address in the NIM, and the reflex action editor lets you use these data from the fieldbus master as an input to the block.

Truth Table

In its simplest form, a standard XOR block looks like this:



And an inverted XOR (an XNOR) block looks like this:



The following truth table shows the possible outputs:

If input 1 is...	and input 2 is...	Then the standard output is...	and the inverted output is...
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

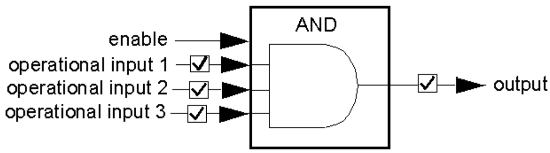
3-Input AND Blocks

Summary

A 3-input AND block performs a logical AND operation on 3 boolean operational inputs. The output is boolean true or false, expressed as a value of 1 or 0. Optionally, you may invert 1 or more inputs. You may also invert the value of the output, in which case the action becomes a logical NAND.

Structure of a 3-Input AND Block

A block diagram for a 3-input AND is shown below:



This AND block has 4 inputs:

Input	Description
Enable Input	turns the block on or off
Operational Input 1	sends a boolean value to the block
Operational Input 2	sends a boolean value to the block
Operational Input 3	sends a boolean value to the block

All inputs enter into an AND operation when the block is enabled, and the result is a boolean output.

The check boxes on the 3 input lines and the output line provide the mechanism by which 1 or more of the input/output values can be inverted. When you click these boxes, a check mark toggles on or off. When a box is checked, the value of the associated input or output is inverted, that means a 1 becomes a 0 and a 0 becomes a 1.

Enable Input

An AND block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Operational Inputs

Every 3-input AND requires 3 operational input values. Each input is a boolean 1 or 0.

These inputs may come from some combination of the following:
- constant values,
- digital inputs from modules on the Island,
- digital outputs from the virtual module *(see page 326)*,
- an output on the action module *(see page 330)* written to by the fieldbus master,
- the output from the first reflex block if the AND is the second part of a nested reflex action *(see page 333)*. In this case, 1 of the operational inputs may be the output from the first reflex block.

Physical Output

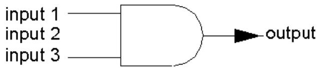
The output from a 3-input AND block is a boolean true (1) or false (0), as shown in the truth tables that follow. The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the AND is the first block in a nested reflex action	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

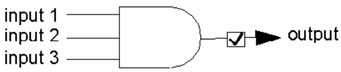
When the output from a reflex block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write to this bit address in the NIM, and the reflex action editor lets you use these data from the fieldbus master as an input to the block.

Truth Tables

In its simplest form, a 3-input AND block looks like this:



And an inverted AND (a NAND) block looks like this:



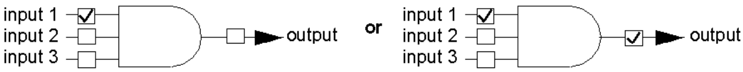
The following truth table shows the possible outputs of this AND operation:

If input 1 is...	and input 2 is...	and input 3 is...	Then the standard output is...	and the inverted output is...
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

Inverted Operational Inputs

You can invert 1 or more of the operational inputs. An inversion is indicated in the Advantys Configuration Software as a check mark in a box on an input line.

Input 1 is inverted:

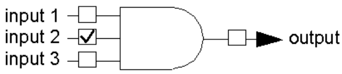


When input 1 is inverted, the truth table yields the following:

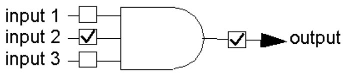
If input 1 is...	and input 2 is...	and input 3 is...	Then the standard output is...	and the inverted output is...
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1

If input 1 is...	and input 2 is...	and input 3 is...	Then the standard output is...	and the inverted output is...
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

Input 2 is inverted:



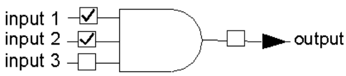
or



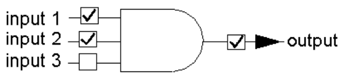
When input 2 is inverted, the truth table yields the following:

If input 1 is...	and input 2 is...	and input 3 is...	Then the standard output is...	and the inverted output is...
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	1	0
1	1	0	0	1
1	1	1	0	1

Inputs 1 and 2 are both inverted:



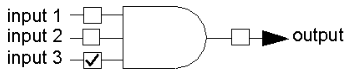
or



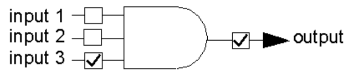
When the inputs 1 and 2 are inverted, the truth table yields the following:

If input 1 is...	and input 2 is...	and input 3 is...	Then the standard output is...	and the inverted output is...
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

Input 3 is inverted:



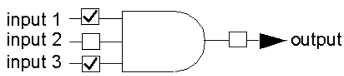
or



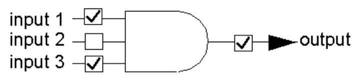
When input 3 is inverted, the truth table yields the following:

If input 1 is...	and input 2 is...	and input 3 is...	Then the standard output is...	and the inverted output is...
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

Inputs 1 and 3 are both inverted:



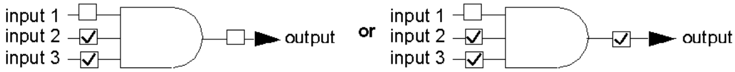
or



When the inputs 1 and 3 are inverted, the truth table yields the following:

If input 1 is...	and input 2 is...	and input 3 is...	Then the standard output is...	and the inverted output is...
0	0	0	0	1
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

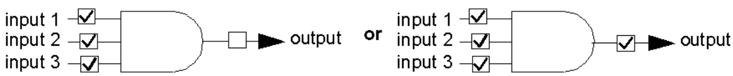
Inputs 2 and 3 are both inverted:



When the inputs 2 and 3 are inverted, the truth table yields the following:

If input 1 is...	and input 2 is...	and input 3 is...	Then the standard output is...	and the inverted output is...
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

All 3 inputs are inverted:



When all 3 inputs are inverted, the truth table yields the following:

If input 1 is...	and input 2 is...	and input 3 is...	Then the standard output is...	and the inverted output is...
0	0	0	1	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

Section 6.3
Integer Compare Reflex Blocks

Introduction

This section describes 4 integer compare reflex blocks. The first 2 of these blocks compare an analog input value to a single threshold value and produce a specific boolean result when the input is greater than or less than the threshold. The other 2 blocks compare an analog input value against a window defined by 2 threshold values and produce a specific boolean result when the input value is either inside or outside that window.

What Is in This Section?

This section contains the following topics:

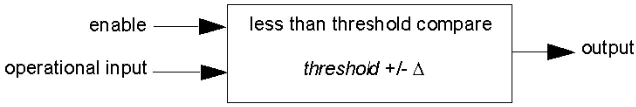
Less-than-Threshold Integer Compare Block

Summary

A less-than-threshold integer compare block performs a comparison between an analog input value and a threshold value that you specify. The analog input value is represented as an integer in the range -32,768...+32,767. The software allows you to assign a delta (Δ), which acts as an hysteresis around the threshold value. The block produces a boolean result as its output.

Structure of a Less-than-Threshold Compare Block

A block diagram for a less-than-threshold integer compare is shown below:



The block has 2 inputs:

Input	Description
Enable Input	turns the block on or off
Operational Input	sends a word value to the block that is compared against the threshold

The block also has 2 preset values *(see page 324)*:

Preset Value	Description
TH Value	threshold against which the operational input value is compared
Δ Value	delta value for hysteresis around the threshold

Specify these presets.

The output is a boolean 1 when the operational input value is less than TH - Δ and a boolean 0 when the input is greater than or equal to TH + Δ. The output remains unchanged when the operational input is greater than or equal to TH - Δ and less than TH + Δ.

Enable Input

A less-than-threshold compare block is enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Operational Input

A less-than-threshold integer compare block uses 1 operational input. This input needs to be a word that holds a signed integer value in the range -32,768...+32,767.

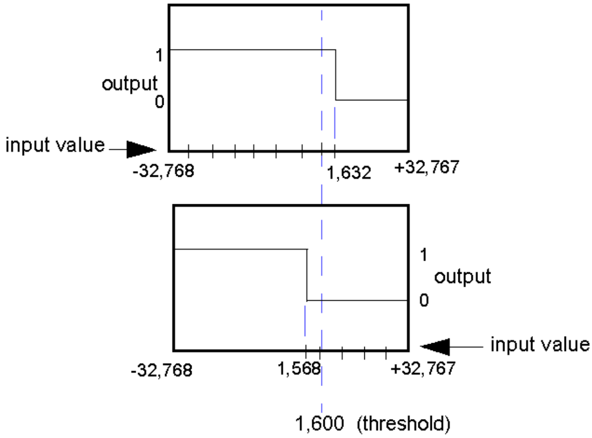
The input can come from the following:
- an analog input from a module on the Island
- an analog output from the virtual module *(see page 326)*
- the output of the first reflex block if the less-than-threshold compare is the second block in a nested reflex action *(see page 333)*

Threshold and Δ

You need to enter 2 preset values: a threshold and a Δ. The threshold is the value against which the operational input is compared. You can add a Δ value to the threshold, which acts as an hysteresis.

NOTE: TH + Δ and TH - Δ have to be integers in the range -32,768...+32,767 to be valid.

For example, say you assign a threshold value of 1,600 to the compare block. You then assign a Δ value of 32 to that threshold:



While the input value is within the 2Δ band, it holds its last value.

In the example above, the output behavior is as follows:

If the input value ...	Then the output ...
increases from a value less than TH - Δ (1,568)	is 1.
reaches TH + Δ (1,632) when increasing	set to 0.
decreases from a value greater than or equal to TH + Δ (1,632) after the output was set to 0	remains 0.
exceeds TH - Δ (1,568) when decreasing	is set to 1.

Physical Output

The block produces as its output a boolean 1 when the input value is less than TH - Δ and a boolean 0 when the input is greater than or equal to TH + Δ. The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the compare is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

When the output from a reflex block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write to this bit address in the NIM, and the reflex action editor lets you use these data from the fieldbus master as an input to the block.

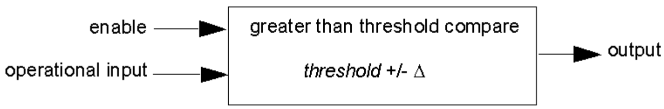
Greater-than-Threshold Integer Compare Block

Summary

A greater-than-threshold integer compare block performs a comparison between an analog input value and a threshold value that you specify using the Advantys Configuration Software. The analog input value is represented as an integer in the range -32,768...+32,767. The software allows you to assign a delta (Δ) value, which acts as an hysteresis around the threshold value. The action produces a boolean result as its output.

Structure of a Greater-than-Threshold Compare Block

A block diagram for a greater-than-threshold integer compare is shown below:



The block has 2 inputs:

Input	Description
Enable Input	turns the block on or off
Operational Input	sends a word value to the block that is compared against the threshold

The block also has 2 preset values *(see page 324)*:

Preset Value	Description
TH Value	threshold against which the operational input value is compared
Δ Value	delta value for hysteresis around the threshold

Specify these presets.

The output is a boolean 1 when the operational input value is greater than TH + Δ and a boolean 0 when the input is less than or equal to TH - Δ. The output remains unchanged when the operational input is greater than TH - Δ and less than or equal to TH + Δ.

Enable Input

A greater-than-threshold compare block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:

- digital input or output from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Operational Input

A greater-than-threshold integer compare block uses 1 operational input. This input needs to be a word with a signed integer value in the range -32,768...+32,767.

The input can come from the following:

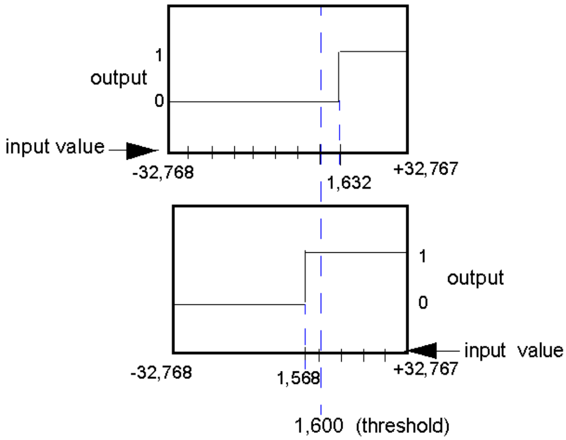
- an analog input from a module on the Island
- an analog output from the virtual module
- the output of the first reflex block if the compare is the second block in a nested reflex action *(see page 333)*

Threshold and Δ

You need to enter 2 values: a threshold and the Δ. The threshold is the value against which the operational input is compared. You can also add a Δ value to the threshold, which acts as an hysteresis.

NOTE: TH + Δ and TH - Δ have to be integers in the range -32,768...+32,767 to be valid.

For example, say you assign a threshold value of 1,600 to the block. You then assign a Δ value of 32 to that threshold:



While the input value is within the 2Δ band, it holds its last value.

In the example above, the output behavior is as follows:

If the input value ...	Then the output ...
increases from a value less than or equal to TH - Δ (1,568)	is 0.
exceeds TH + Δ (1,632) when increasing	set to 1.
decreases from a value greater than TH + Δ (1,632) after the output was set to 1	remains 1.
reaches TH - Δ (1,568) when decreasing	is set to 0.

Physical Output

The block produces a boolean 1 as its output when the input value is greater than TH + Δ and a boolean 0 as its output when the input value is less than or equal to TH - Δ. The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the compare is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

When the output from a reflex block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write to this bit address in the NIM, and the reflex action editor lets you use these data from the fieldbus master as an input to the block.

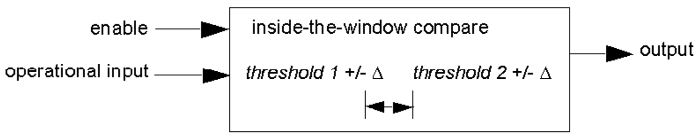
Inside-the-Window Integer Compare Block

Summary

An inside-the-window integer compare block performs a comparison between an analog input value and a window bounded by 2 thresholds. The input value is represented as an integer in the range -32,768...+32,767. The software lets you assign values to the 2 thresholds (TH1 and TH2) along with a delta (Δ) value, which acts as an hysteresis around TH1 and TH2. The block produces a boolean result as its output.

Structure of an Inside-the-Window Compare Block

A block diagram for an inside-the-window integer compare is shown below:



The block has 2 inputs:

Input	Description
Enable Input	turns the block on or off
Operational Input	sends a word value to the block that is compared against the thresholds

The block also has 3 preset values *(see page 324)*:

Preset Value	Description
TH1	threshold 1 against which the operational input value is compared
TH2	threshold 2 against which the operational input value is compared
Δ Value	delta value for hysteresis around the TH1 and TH2 values

The range of values between TH1 - Δ and TH2 + Δ comprises the window against which the operational input value will be compared. Specify these presets.

The output is a boolean 1 when the operational input value is inside the window (greater than TH1 + Δ but less than TH2 - Δ) and a boolean 0 when the input value is not inside the window (less than or equal to TH1 - Δ or greater than or equal to TH2 + Δ). The output remains unchanged when the operational input is greater than TH1 - Δ but less than or equal to TH1 + Δ, or when it is greater than or equal to TH2 - Δ but less than TH2 + Δ.

Enable Input

An inside-the-window integer compare block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- a digital input from a module on the Island
- a digital output from the virtual module *(see page 326)*
- an output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Thresholds

Inside-the-window compares require 2 threshold values, which define the upper and lower bounds of the window. Each TH value needs to be a signed integer value in the range -32,768...+32,767. TH1 defines the lower boundary of the window; TH2 defines the upper boundary.

NOTE: The value of TH2 has to be greater than the value of TH1.

Operational Input

An inside-the-window compare uses 1 operational input. It has to be a word with an integer value in the range -32,768...+32,767.

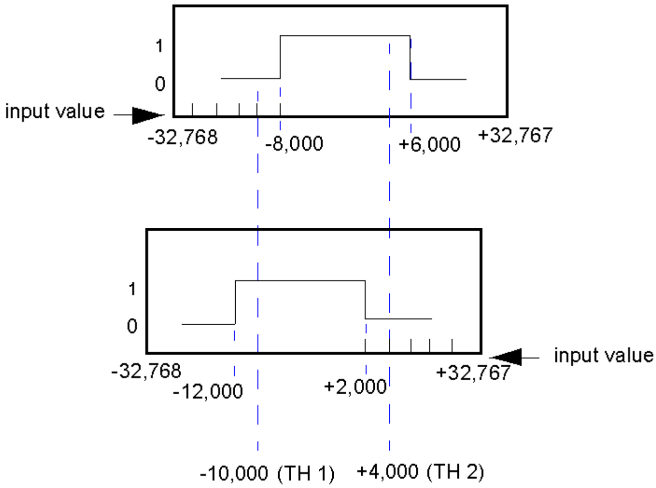
The input can come from the following:
- an analog input from a module on the Island
- an analog output from the virtual module *(see page 326)*
- the output of the first reflex block if the inside-the-window compare is the second block in a nested reflex action *(see page 333)*

Delta (Δ)

You can also add a Δ value to an inside-the-window compare. The Δ acts as an hysteresis around the 2 thresholds.

NOTE: To be valid, TH2 - TH1 has to be greater than 2Δ. For example, say that TH1 = -10,000 and TH2 = +4,000. The Δ value you assign to the reflex action has to therefore be less than 7,000.

Suppose you have a window defined by TH1 = -10,000 and TH2 = +4,000. To that window, you specify a Δ of 2,000:



While the input value is inside the area defined by the threshold and the Δ, it holds its last value.

In the example above, the output behavior is as follows:

If the input value ...	Then the output ...
is less than or equal to TH1 - Δ (-12,000) and increases	is 0.
exceeds TH1 + Δ (-8,000) when increasing	is set to 1.
reaches TH2 + Δ (+6,000) when increasing	is set to 0.
decreases from a value greater than or equal to TH2 + Δ (+6,000) after the output was set to 0	remains 0.
exceeds TH2 - Δ (+2,000) when decreasing	is set to 1.
reaches TH1 - Δ (-12,000) when decreasing	is set to 0.

Physical Output

The block produces a boolean 1 when the input value is inside the window and a boolean 0 when the input value is not inside that window. The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the compare is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

When the output from a reflex block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write to this bit address in the NIM, and the reflex action editor lets you use these data from the fieldbus master as an input to the block.

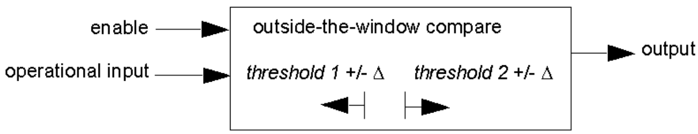
Outside-the-Window Integer Compare Block

Summary

An outside-the-window integer compare block performs a comparison between an analog input value and a window of values bounded by 2 thresholds. The input value is represented as an integer in the range -32,768...+32,767. The software lets you assign values to the 2 thresholds (TH1 and TH2) along with a delta (Δ) value, which acts as an hysteresis around TH1 and TH2. The block produces a boolean result as its output.

Structure of an Outside-the-Window Compare Block

A block diagram for an outside-the-window integer compare is shown below:



The block has 2 inputs:

Input	Description
Enable Input	turns the block on or off
Operational Input	sends a word value to the block that is compared against the thresholds

The block also has 3 preset values *(see page 324)*:

Preset Value	Description
TH1	threshold 1 against which the operational input value is compared
TH2	threshold 2 against which the operational input value is compared
Δ Value	delta value for hysteresis around the TH1 and TH2 values

The output is a boolean 1 when the operational input value is outside the window (less than TH1 - Δ or greater than TH2 + Δ) and a boolean 0 when the input value is not outside the window (greater than or equal to TH1 + Δ but less than or equal to TH2 - Δ). The output remains unchanged when the operational input is greater than or equal to TH1 - Δ but less than TH1 + Δ, or when it is greater than TH2 - Δ but less than or equal to TH2 + Δ.

Enable Input

An outside-the-window integer compare block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Thresholds

Outside-the-window compares require 2 threshold values, which define the upper and lower bounds of the window. Each TH value needs to be a signed integer in the range -32,768...+32,767. TH1 defines the lower boundary of the window; TH2 defines the upper boundary.

NOTE: The value of TH2 has to be greater than the value of TH1.

Operational Input

An outside-the-window compare block uses 1 operational input. It has to be a word with an integer value in the range -32,768...+32,767.

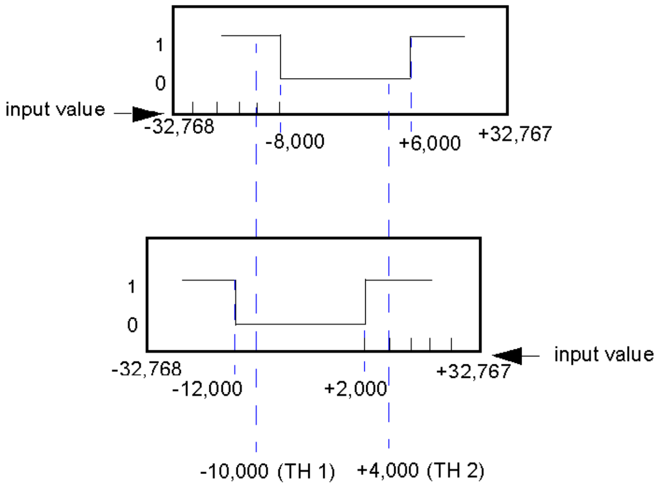
The input can come from the following:
- an analog input from a module on the Island
- an analog output from the virtual module
- the output of the first reflex block if the outside-the-window compare is the second block in a nested reflex action *(see page 333)*

Delta (Δ)

You can also add a Δ value to an outside-the-window compare, which acts as an hysteresis around the 2 thresholds.

NOTE: To be valid, TH2 - TH 1 has to be greater than 2Δ. For example, say that TH1 = -10,000 and TH2 = +4,000. The Δ value you assign to the reflex action has to therefore be less than 7,000.

Suppose you have a window defined by TH1 = -10,000 and TH2 = +4,000. To that window, you specify a Δ of 2,000:



While the input value is inside the area defined by the threshold and the Δ, it holds its last value.

In the example above, the output behavior is as follows:

If the input value ...	Then the output ...
is less than TH1 - Δ (-12,000) and increases	is 1.
reaches TH1 + Δ (-8,000) when increasing	is set to 0.
exceeds TH2 + Δ (+6,000) when increasing	is set back to 1.
decreases from a value greater than TH2 + Δ (+6,000)	remains 1.
reaches TH2 - Δ (+2,000) when decreasing	is set to 0.
exceeds TH1 - Δ (-12,000) when decreasing	is set back to 1.

Physical Output

The block produces a boolean 1 when the input value is outside the window and a boolean 0 when the input value is not outside that window. The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the compare is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

When the output from a reflex block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write to this bit address in the NIM, and the reflex action editor lets you use these data from the fieldbus master as an input to the block.

Section 6.4
Unsigned Compare Reflex Blocks

Introduction

This section describes 4 unsigned compare reflex blocks. The first 2 of these blocks compare an analog input value to a single threshold value and produce a specific boolean result when the input is greater that or less than the threshold. The other 2 blocks compare an analog input value against a window defined by 2 threshold values and produce a specific boolean result when the input value is either inside or outside that window.

What Is in This Section?

This section contains the following topics:

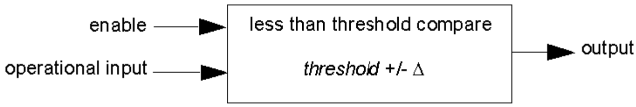
Less-than-Threshold Unsigned Compare Block

Summary

A less-than-threshold unsigned compare block performs a comparison between an analog input value and a threshold value. The input value is represented as an integer in the range 0...65,535. The software lets you assign the threshold value along with a delta (Δ) value, which acts as an hysteresis for the threshold. The action produces a boolean result as its output.

Structure of a Less-than-Threshold Comparison Block

A block diagram for a less-than-threshold unsigned compare is shown below:



The block has 2 inputs:

Input	Description
Enable Input	turns the block on or off
Operational Input	sends a word value to the block that is compared against the threshold

The block also has 2 preset values *(see page 324)*:

Preset Value	Description
TH Value	threshold against which the operational input value is compared
Δ Value	delta value for hysteresis around the threshold value

The output is a boolean 1 when the operational input is less than TH - Δ and a boolean 0 when the input is greater than or equal to TH + Δ. The output remains unchanged when the operational input is greater than or equal to TH - Δ but less than TH + Δ.

Enable Input

A less-than-threshold unsigned compare block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Operational Input

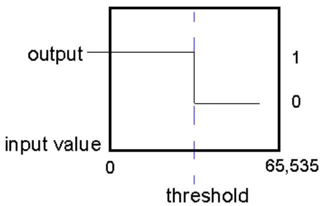
A less-than-threshold unsigned compare block uses 1 operational input. It has to be a word with an unsigned integer value in the range 0...65,535.

The input can come from the following:
- an analog input from a module on the Island
- an analog output from the virtual module *(see page 326)*
- the output of the first reflex block if the less-than-threshold compare is the second block in a nested reflex action *(see page 333)*

NOTE: Unsigned compare blocks are often nested together with counter blocks *(see page 388)*. The unsigned compare is always the second block in the nested action, and the analog output from the counter is used as its operational input. These 2 action types complement each other well because the output from a counter is always unsigned with 16-bit resolution.

NOTE: Do not use a word that contains a signed negative integer value as the operational input to this comparison. The reflex action will misinterpret a value of 1 in the sign bit position (bit 15) as part of the integer value. Avoid the use of Modules such as the STB AVI 1270 analog input module, which produces an input with a possible negative integer value, as the source for the operational input to your reflex action.

The illustration below shows a simple case of how the block works:



In the example illustrated above, the output behavior is as follows:

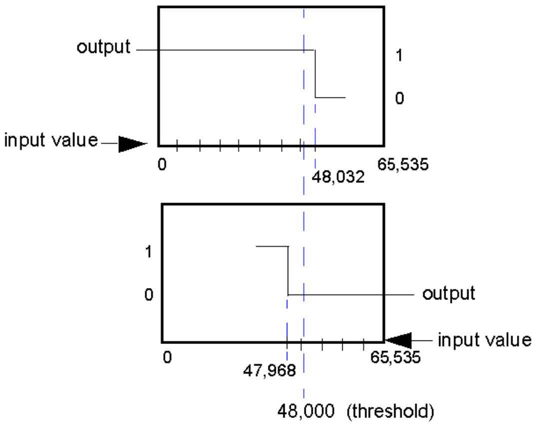
If the operational input is ...	Then the output is...
less than the threshold value	1
greater than or equal to the threshold value	0

Threshold and Δ

You need to enter 2 values in a compare action: the threshold and the Δ. The threshold is the value against which the operational input is compared, as shown in the examples above. The Δ value acts as an hysteresis around the threshold.

NOTE: To be valid, TH + Δ and TH - Δ have to be integers in the range 0...65,535.

For example, say you assign a threshold value of 48,000 to the comparison action, and then you assign a Δ value of 32 to that threshold:



While the input value is within the Δ band, it holds its last value.

In the example above, the output behavior is as follows:

If the input value ...	Then the output ...
increases from a value less than TH - Δ (47,968)	is 1.
reaches TH + Δ (48,032) when increasing	is set to 0.
decreases from a value greater than or equal to TH + Δ (48,032) after the output was set to 0	remains 0.
exceeds TH - Δ (47,968) when decreasing	is set to 1.

Physical Output

The block produces a boolean 1 when the input is less than TH - Δ and a boolean 0 when the input is greater than or equal to TH + Δ. The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the compare is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

When the output from a reflex block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write data to this bit address in the NIM, and the reflex action editor lets you use these data from the fieldbus master as an input to the block.

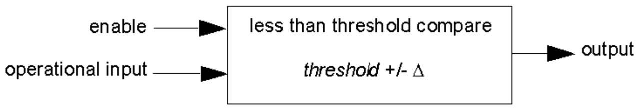
Greater-than-Threshold Unsigned Compare Block

Summary

A greater-than-threshold unsigned compare block performs a comparison between an analog input value and a threshold value (TH). The input value is represented as an integer in the range 0...65,535. The software lets you configure the TH value along with a delta (Δ) value, which acts as an hysteresis for the threshold. The action produces a boolean result as its output.

Structure of a Greater-than-Threshold Compare Block

A block diagram for a greater-than-threshold unsigned compare is shown below:



The block has 2 inputs:

Input	Description
Enable Input	turns the block on or off
Operational Input	sends a word value to the block that is compared against the threshold

The block also has 2 preset values *(see page 324)*:

Preset Value	Description
TH Value	threshold against which the operational input value is compared
Δ Value	delta value for hysteresis around the threshold value

The output is a boolean 1 when the operational input is greater than TH + Δ and a boolean 0 when the input is less than or equal to the TH - Δ. The output remains unchanged when the operational input is greater than TH - Δ but less than or equal to TH + Δ.

Enable Input

A greater-than-threshold unsigned compare block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Operational Input

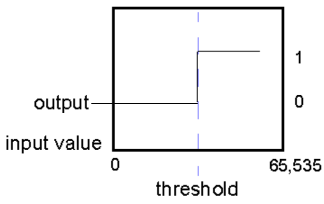
A greater-than-threshold unsigned compare block uses 1 operational input. It has to be a word with an unsigned integer value in the range 0...65,535.

The input can come from the following:
- an analog input channel on the Island
- an analog output from the virtual module *(see page 326)*
- the output of the first reflex block if the greater-than-threshold compare is the second block in a nested reflex action *(see page 333)*

NOTE: Unsigned compares are often nested together with counter blocks *(see page 388)*. The unsigned compare is always the second block in the nested action, and the analog output from the counter is used as its operational input. These 2 action types complement each other well because the output from a counter is always unsigned with 16-bit resolution.

NOTE: Do not use a word that contains a signed negative integer value as the operational input to an unsigned integer comparison. The block will misinterpret a value of 1 in the sign bit position (bit 15) as part of the integer value. Avoid the use of Modules such as the STB AVI 1270 analog input module, which produces an input with a possible negative integer value, as the source for the operational input to the block.

The illustration below shows the behavior of the block when Δ is 0:



In the example illustrated above, the output behavior is as follows:

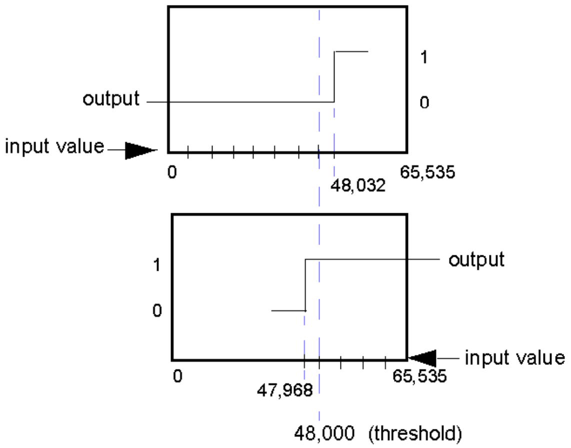
If the operational input is ...	Then the output is...
less than or equal to the threshold value	0
greater than the threshold value	1

Threshold and Δ

You need to enter 2 values: the threshold and the Δ. The threshold is the value against which the operational input is compared. You can also add a Δ value to the threshold, which acts as an hysteresis.

NOTE: To be valid, TH + Δ and TH - Δ have to be integers in the range 0...65,535.

For example, say you assign a threshold value of 48,000 to the comparison action, and then you assign a Δ value of 32 to that threshold:



While the input value is within the 2Δ band, it holds its last value.

In the example above, the output behavior is as follows:

If the input value ...	Then the output ...
increases from a value less than or equal to TH - Δ (47,968)	is 0.
exceeds TH + Δ (48,032) when increasing	is set to 1.

If the input value ...	Then the output ...
decreases from a value greater than TH + Δ (48,032) after the output was set to 1	remains 1.
reaches TH - Δ (47,968) when decreasing	is set to 0.

Physical Output

The block produces a boolean 1 when the input is greater than TH + Δ and a boolean 0 when the input is less than or equal to TH - Δ. The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the compare is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

When the output from a reflex block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write data to this bit address in the NIM, and the reflex action editor lets you use these data from the fieldbus master as an input to the block.

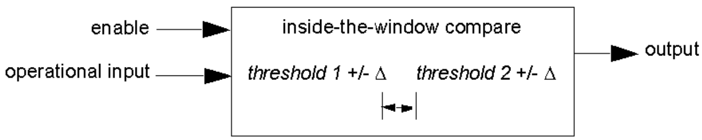
Inside-the-Window Unsigned Compare Block

Summary

An inside-the-window unsigned compare block performs a comparison between an analog input value and a window of values bounded by 2 thresholds. The input value is represented as an integer in the range 0...65,535. The software lets you assign values to the 2 thresholds (TH1 and TH2) along with a delta (Δ) value, which acts as an hysteresis around TH1 and TH2. The block produces a boolean result as its output.

Structure of an Inside-the-Window Compare Block

A block diagram for an inside-the-window unsigned compare is shown below:



The block has 2 inputs:

Input	Description
Enable Input	turns the block on or off
Operational Input	sends a word value to the block that is compared against the thresholds

The block also has 3 preset values *(see page 324)*:

Preset Value	Description
TH1	threshold 1 against which the operational input value is compared
TH2	threshold 2 against which the operational input value is compared
Δ Value	delta value for hysteresis around the TH1 and TH2 values

The range of values between TH1 - Δ and TH2 + Δ comprises the window against which the operational input value will be compared. Specify these presets.

The output is a boolean 1 when the operational input value is inside the window (greater than TH1 + Δ but less than TH2 - Δ) and a boolean 0 when the input value is not inside the window (less than or equal to TH1 - Δ or greater than or equal to TH2 + Δ). The output remains unchanged when the operational input is greater than TH1 - Δ but less than or equal to TH1 + Δ, or when its is greater than or equal to TH2 - Δ but less than TH2 + Δ.

Enable Input

An inside-the-window unsigned compare block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by

If the enable input is a boolean, it may be produced by the following:

- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Thresholds

Inside-the-window compare blocks require 2 threshold values, which serve as the upper and lower bounds of the window. Each TH value needs to be an unsigned integer value in the range 0...65,535. TH1 defines the lower boundary of the window; TH2 defines the upper boundary.

NOTE: The value of TH2 has to be greater than the value of TH1.

Operational Input

An inside-the-window compare block uses 1 operational input. It has to be a word with an unsigned integer in the range 0..65,535.

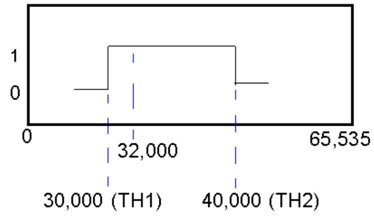
The input can come from the following:

- an analog input from a module on the Island
- an analog output from the virtual module *(see page 326)*
- the output of the first reflex block if the less-than-threshold compare is the second block in a nested reflex action *(see page 333)*

NOTE: Unsigned compare blocks are often nested together with counter blocks *(see page 388)*. The unsigned compare is always the second block in the nested action, and the analog output from the counter is used as its operational input. These 2 action types complement each other well because the output from a counter is always unsigned with 16-bit resolution.

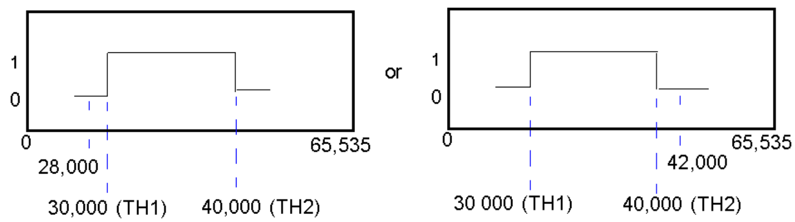
NOTE: Do not use a word that contains a signed negative integer value as the operational input to an unsigned integer compare. The block will misinterpret a value of 1 in the sign bit position (bit 15) as part of the integer value. Avoid the use of Modules such as the STB AVI 1270 analog input module, which produce an input with a possible negative integer value, as the source for the operational input to the block.

Suppose that you have 2 threshold values, where TH1 = 30,000 and TH2 = 40,000, and then suppose that the operational input is 32,000 and Δ = 0:



Because the value of the operational input is inside the window defined by TH1 and TH2, the block produces a boolean 1 as its output.

Alternately, suppose the value of the operational input is less than TH1 (say, 28,000) or greater than TH2 (say, 42 000):



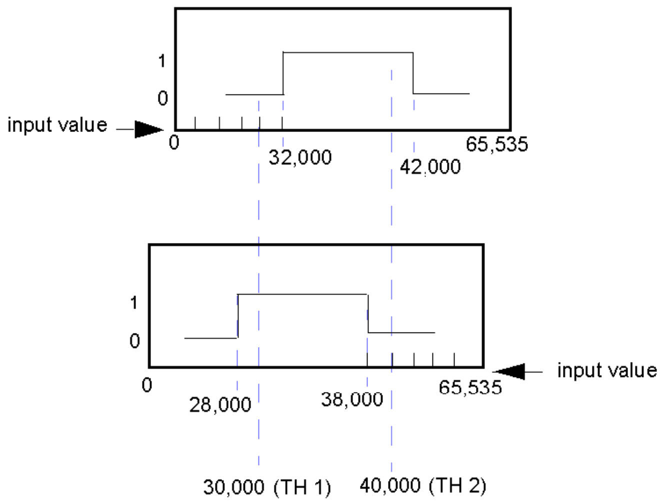
The block produces a boolean 0 as its output because the input value is outside the window.

Delta (Δ)

You can also add a Δ value to an inside-the-window compare block, which acts as an hysteresis around the 2 thresholds.

NOTE: To be valid, TH2 - TH 1 has to be greater than 2Δ. For example, say that TH1 = 30,000 and TH2 = 40,000. The Δ value you assign to the block has to therefore be less than 5,000.

Suppose you have a window defined by TH 1 = 30,000 and TH 2 = 40,000. To that window, you specify a Δ of 2,000:



While the input value is within the window defined by the threshold and the Δ, it holds its last value.

In the example above, the output behavior is as follows:

If the input value ...	Then the output ...
is less than or equal to TH1 - Δ (28,000) and increases	is 0.
exceeds TH1 + Δ (32,000) when increasing	is set to 1.
reaches TH2 + Δ (42,000) when increasing	is set back to 0.
decreases from a value greater than or equal to TH2 + Δ (42,000) after the output was set to 0	remains 0.
exceeds TH2 - Δ (38,000) when decreasing	is set to 1.
reaches TH1 - Δ (28,000) when decreasing	is set back to 0.

Physical Output

The block produces a boolean 1 when the input value is within the window and a boolean 0 when the input value is not inside that window. The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the compare is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

When the output from a reflex block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write data to this bit address in the NIM, and the reflex action editor lets you use these data from the fieldbus master as an input to the block.

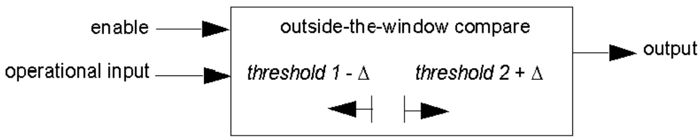
Outside-the-Window Unsigned Compare Block

Summary

An outside-the-window unsigned compare block performs a comparison between an analog input value and a window of values bounded by 2 thresholds. The input value is represented as an integer in the range 0...65,535. The software lets you assign values to the 2 thresholds (TH1 and TH2) along with a delta (Δ) value, which acts as an hysteresis around TH1 and TH2. The block produces a boolean result as its output.

Structure of an Outside-the-Window Compare Block

A block diagram for an outside-the-window unsigned compare is shown below:



The block has 2 inputs:

Input	Description
Enable Input	turns the block on or off
Operational Input	sends a word value to the block that is compared against the thresholds

The block also has 3 preset values *(see page 324)*:

Preset Value	Description
TH1	threshold 1 against which the operational input value is compared
TH2	threshold 2 against which the operational input value is compared
Δ Value	delta value for hysteresis around the TH1 and TH2 values

The range of values between TH1 - Δ and TH2 + Δ comprises the window against which the operational input value will be compared. Specify these presets.

The output is a boolean 1 when the operational input value is outside the window (less than TH1 - Δ or greater than TH2 + Δ) and a boolean 0 when the input value is not outside the window (greater than or equal to TH1 + Δ but less than or equal to TH2 - Δ). The output remains unchanged when the operational input is greater than or equal to TH1 - Δ but less than TH1 + Δ, or when it is greater than TH2 - Δ but less than or equal to TH2 + Δ.

Enable Input

An outside-the-window unsigned compare block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Thresholds

Outside-the-window unsigned compare blocks require 2 threshold values, which serve as the upper and lower bounds of the window. Each TH value needs to be an unsigned integer in the range 0...65,535. TH 1 defines the lower boundary of the window; TH 2 defines the upper boundary.

NOTE: The value of TH 2 has to be greater than the value of TH 1.

Operational Input

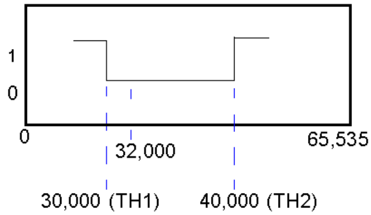
An outside-the-window unsigned compare block uses 1 operational input. It has to be a word with an unsigned integer in the range 0...65,535.

The input can come from the following:
- an analog input from a module on the Island
- an analog output from the virtual module *(see page 326)*
- the output of the first reflex block if the less-than-threshold compare is the second block in a nested reflex action *(see page 333)*

NOTE: Unsigned compare blocks are often nested together with counter blocks *(see page 388)*. The unsigned compare is always the second block in the nested action, and the analog output from the counter is used as its operational input. These 2 action types complement each other well because the output from a counter is always unsigned with 16-bit resolution.

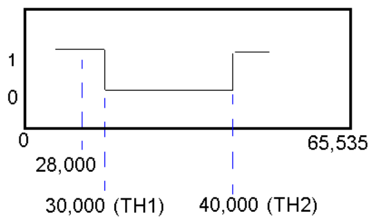
NOTE: Do not use a word that contains a signed negative integer value as the operational input to an unsigned integer compare. The block will misinterpret a value of 1 in the sign bit position (bit 15) as part of the integer value. Avoid the use of Modules such as the STB AVI 1270 analog input module, which produces an input with a possible negative integer value, as the source for the operational input to the block.

Suppose that you have 2 threshold values, where TH1 = 30,000 and TH2 = 40,000, and then suppose that the operational input is 32,000 and Δ = 0:

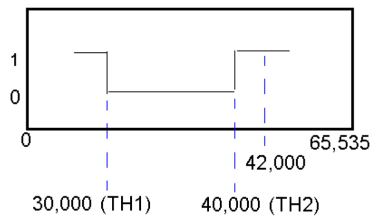


Because the value of the operational input is inside the window defined by TH1 and TH2, the block produces a boolean 0 as its result.

Alternately, suppose the value of the operational input is less than TH1 (say, 28,000) or greater than TH2 (say, 42,000):



or



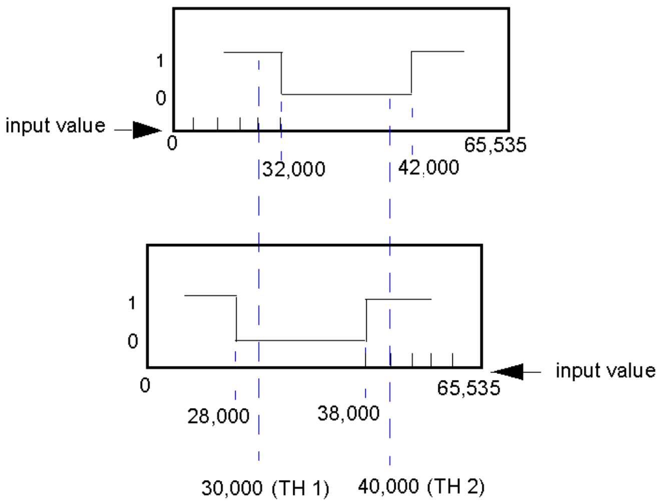
The block produces a boolean 1 as its result because the input value is outside the window.

Delta (Δ)

You can also add a Δ value to an outside-the-window unsigned compare block, which acts as an hysteresis around the 2 thresholds.

NOTE: To be valid, TH 2 - TH 1 has to be greater than 2Δ. For example, say that TH 1 = 30,000 and TH 2 = 40,000. The Δ value you assign to the block has to therefore be less than 5,000.

Suppose you have a window defined by TH 1 = 30,000 and TH 2 = 40,000. To that window, you specify a Δ of 2,000:



While the input value is within the window defined by the threshold and the Δ, it holds its last value.

In the example above, the output behavior is as follows:

If the input value ...	Then the output ...
is less than TH1 - Δ (28,000) and increases	is 1.
reaches TH1 + Δ (32,000) when increasing	is set to 0.
exceeds TH2 + Δ (42,000) when increasing	is set back to 1.
decreases from a value greater than TH2 + Δ (42,000)	remains 1.
reaches TH2 - Δ (38,000) when decreasing	is set to 0.
exceeds TH1 - Δ (28,000) when decreasing	is set back to 1.

Physical Output

The block produces a boolean 1 when the input value is outside the window and a boolean 0 when the input value is not outside that window. The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the compare is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

When the output from a reflex block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write data to this bit address in the NIM, and the reflex action editor lets you use these data from the fieldbus master as an input to the block.

Section 6.5
Counter Reflex Blocks

Introduction

This section describes 2 counter reflex blocks that count boolean inputs either up or down from a preset value. The result from these counter blocks is a word value.

The first counter increments or decrements on the rising edge of the operational input, and the other increments or decrements on the falling edge of the operational input.

What Is in This Section?

This section contains the following topics:

Falling-Edge Counter Block

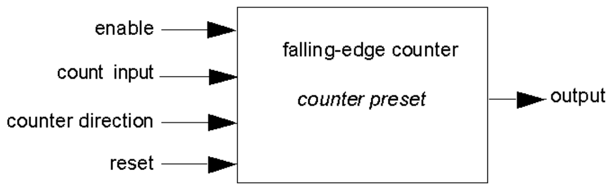
Summary

A falling-edge counter block counts up (increments) or down (decrements) each time its count input falls from 1 to 0. The count begins at a user-specified counter preset value and continues up or down until the block receives a reset input. A reset sends the counter back to its preset value and starts a new counting sequence. The block produces an unsigned analog word as its output.

NOTE: Unlike other reflex actions, a counter block is designed to act exclusively as the first block in a nested reflex action *(see page 333)*. The output from a counter block is used as an analog input to an unsigned compare block *(see page 369)*. As a result, the Reflex Editor lets you map the output only to a digital action module, even though the output value is analog.

Structure of a Falling-Edge Counter Block

A block diagram for a standard falling-edge counter is shown below:



The block has the following 4 inputs:

Input	Description
Enable	turns the counter on or off
Count	sends a boolean value to the block that will generate a count input when it transitions from 1 to 0
Counter Direction	defines whether the block will increment or decrement on each count
Reset	will restart the counting operation at the predefined counter preset value

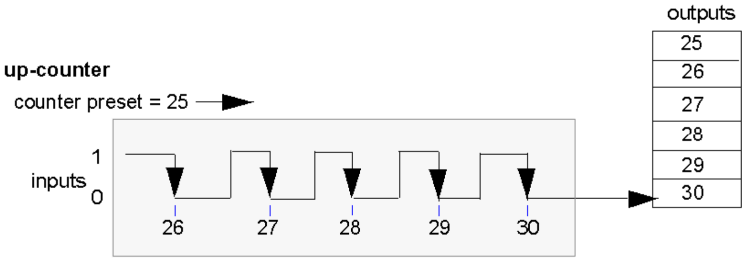
The block also has a counter preset value *(see page 324)*: an integer value that defines the starting point for each counting operation. Specify this preset.

The block produces a 16-bit word output on each count. The word holds an unsigned integer value in the range 0...65,535. On each count, the output equals the counter preset plus the incremented count or minus the decremented count.

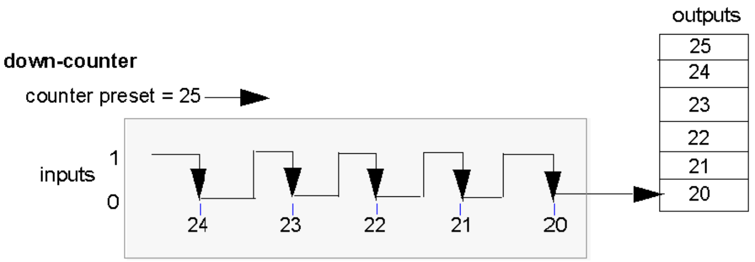
Counter Preset

Specify the counter preset value before implementing a counter operation. The preset has to be an unsigned integer in the range 0...65,535. A counting sequence always begins at this counter preset value, and then increments or decrements from it each time the count input value falls from 1 to 0.

For example, say you configure an up-counter with a counter preset at 25. The block will start a counting sequence at 25 and will increment by 1 each time the count input falls from 1 to 0:



If you are using a down-counter with a counter preset at 25, the counter will start a counting sequence at 25 and will decrement by 1 each time the count input drops from 1 to 0:



Enable Input

A falling-edge counter block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

NOTE: When the block is enabled, the counting starts depending on the value of the count input at enabling time, see the table below.

The following applies when the enable input transitions from 0 to 1:

If the count input is ...	Then ...
0	the counter assumes that a falling-edge transition has just taken place and increments or decrements once.
1	the block waits for the next falling-edge transition before it starts counting.

Count Input

A falling-edge counter block receives a stream of boolean 1 s and 0 s as a count input. The counter increments or decrements each time the input value falls from 1 to 0.

The inputs may come from the following:
- constant
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

NOTE: At start-up, check that the count input provides a 1 to the counter block. If the count input is 0 when the block is enabled, the counter will assume that a falling-edge transition has just taken place and will increment or decrement once, see the table above.

Count Direction Input

Every falling-edge counter block needs to count in a direction, either up or down. Using the Advantys Configuration Software, set the direction of the counter as a constant value of either 0 or 1, where the following applies:

- 0 = an up-counter
- 1 = a down-counter

The inputs may come from the following:

- constant
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

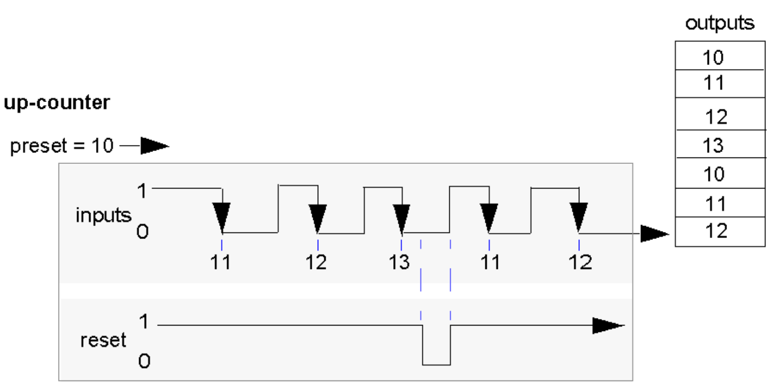
Reset Input

Every falling-edge counter action has a reset input. The reset input is a boolean value. A reset value of 0 returns the counter to the specified preset value. A reset value of 1 allows the counter to continue to increment or decrement. While reset is low, the block does not count.

The reset input can be configured to come from the following:

- constant
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

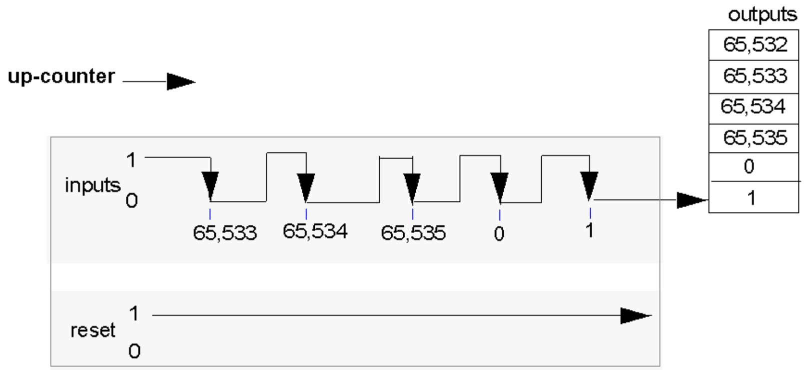
For example, say you have an up-counter with a preset value of 10. The counter will start its counting sequence at 10 and will increment by 1 each time the count input drops from 1 to 0. Suppose that the reset input drops to 0 after 3 counts:



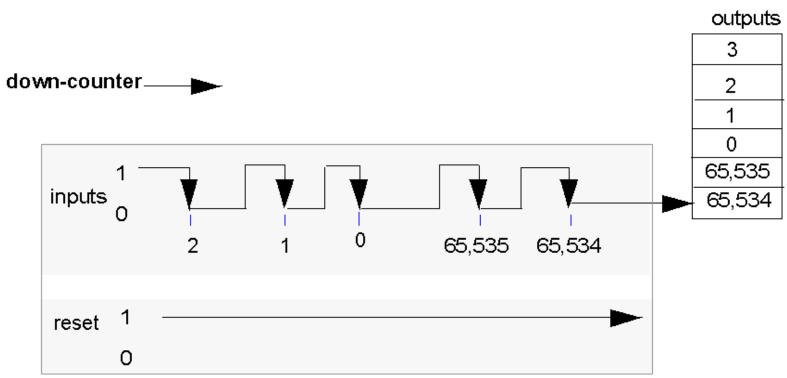
The reset input causes the counter to return to the preset value (10) and to start the up-counting process over again.

Wrap-Arounds

If an up-counter increments up to 65,535 and does not receive a reset input, it will wrap to 0 and continue to increment from there until it is reset. At reset, the counter will return to the preset value and start a new incremental up-count:



If a down-counter decrements down to 0 and does not receive a reset input, it will wrap to 65,535 and continue to decrement from there until it is reset. At reset, the counter will return to the preset value and start a new incremental down-count:



Physical Output

The output of a falling-edge counter is a word that holds an unsigned integer value in the range 0...65,535. The physical output *(see page 325)* needs to be mapped to a digital action module.

NOTE: A counter is always the first block in a nested reflex action. The action module has to always be a digital output module, which will be configured to perform a compare action (preferably an unsigned compare, since values greater than 32,767 would be misinterpreted by a standard integer compare action). The Reflex Editor does not allow you to map the counter's output to an analog module.

You need to specify the channel to which the counter output will be mapped as **None** so that the output will be stored temporarily in an internal reflex buffer, and then used as the count input to the compare block.

Power-Up and Fallback

Upon power-up, the counter's output data is set to the preset value (if the enable input is on).

If a detected error causes the counter block to go to its fallback state, the output freezes in its last active state. Upon removal of the detected error condition, the counter starts counting again at the point where the output was frozen.

Rising-Edge Counter Block

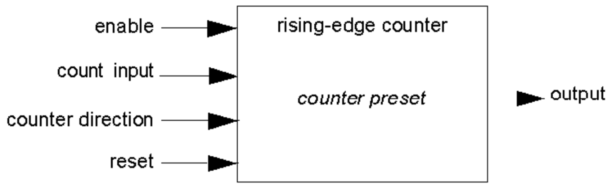
Summary

A rising-edge counter block counts up (increments) or down (decrements) each time an count input to the action rises from 0 to 1. The count begins at a user-specified preset value and continues counting up or down until the block receives a reset input. A reset sends the counter back to its preset value and starts a new counting sequence. The block produces an unsigned analog word as its output.

NOTE: Unlike other reflex actions, a counter block is designed to act exclusively as the first block in a nested reflex action *(see page 333)*. The output from a counter block is used as an analog input to an unsigned compare block *(see page 369)*. As a result, the Reflex Editor allows you to map the output only to a digital action module, even though the output value is analog.

Structure of a Rising-Edge Counter Block

A block diagram for a standard rising-edge counter is shown below:



The block has the following 4 inputs:

Input	Description
Enable	turns the counter on or off
Count	sends a boolean value to the block that will generate a count input when it transitions from 0 to 1
Counter Direction	defines whether the action will increment or decrement on each count
Reset	will restart the counting operation at the predefined counter preset value

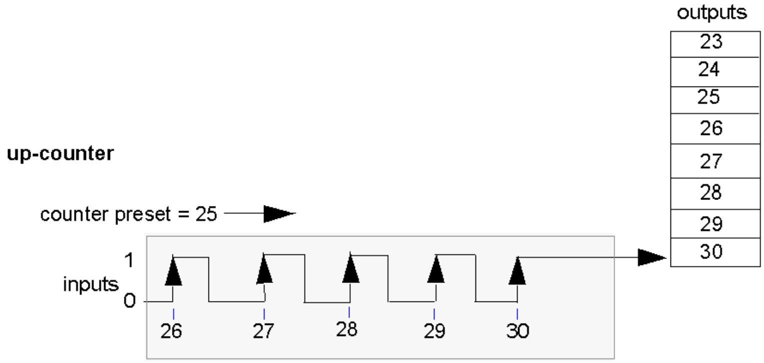
The block also has a counter preset value *(see page 324)*: an integer value that defines the starting point for each counting operation. Specify this preset.

The block produces a 16-bit word output on each count. The word holds an unsigned integer value in the range 0...65,535. On each count, the output equals the counter preset plus the incremented count or minus the decremented count.

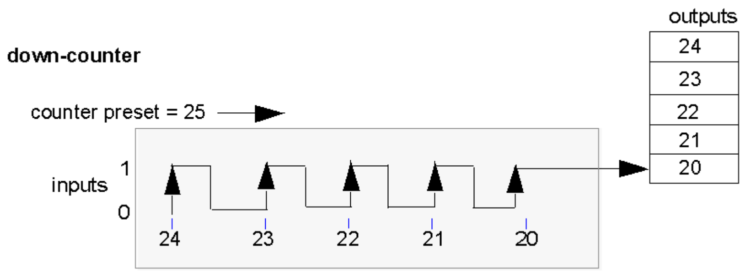
Counter Preset

Specify the counter preset value before implementing a counter operation. The preset has to be an unsigned integer in the range 0...65,535. A counting sequence always begins at the counter preset, and then increments or decrements from there each time the count input value transitions from 0 to 1.

For example, say you have an up-counter with a preset value of 25. The counter will start a counting sequence at 25 and will increment by 1 each time the count input rises from 0 to 1:



If you are using a down-counter with a preset value of 25, the counter will start a counting sequence at 25 and will decrement by 1 each time the count input rises from 0 to 1:



Enable Input

A rising-edge counter block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If a boolean input is used, its value may be produced by the following:
- digital input or output from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

NOTE: When the block is enabled, the counting starts depending on the value of the count input at enabling time, see the table below.

The following applies when the enable input transitions from 0 to 1:

If the count input is ...	Then ...
1	the counter assumes that a falling-edge transition has just taken place and increments or decrements once.
0	the block waits for the next falling-edge transition before it starts counting.

Count Input

A rising-edge counter block has 1 count input: a stream of boolean 1 s and 0 s. The counter increments or decrements each time the input value rises from 1 to 0.

The inputs can come from the following:
- constant
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

NOTE: At start-up, check that the count input provides a 0. If the count input is 1 when the counter becomes enabled, the block will assume that a rising-edge transition has just taken place and will increment or decrement once.

Count Direction Input

Every rising-edge counter block needs to count in a direction, either up or down. Using the Advantys Configuration Software, you can set the direction of the counter as a constant value of either 0 or 1, where the following applies:

- 0 = an up-counter
- 1 = a down-counter

The input can come from the following:

- constant
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

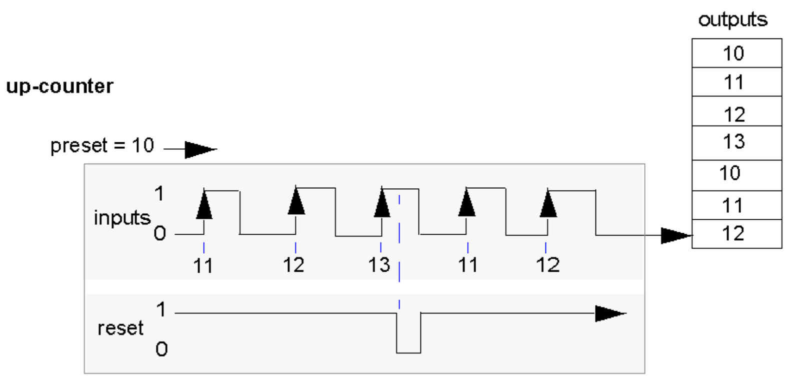
Reset Input

Every rising-edge counter block has a reset input. The reset input is a boolean value. A reset value of 0 returns the counter to the specified preset value. A reset value of 1 allows the counter to continue to increment or decrement. While reset is low, the counter will not count.

The reset input can be configured to come from the following:

- constant
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

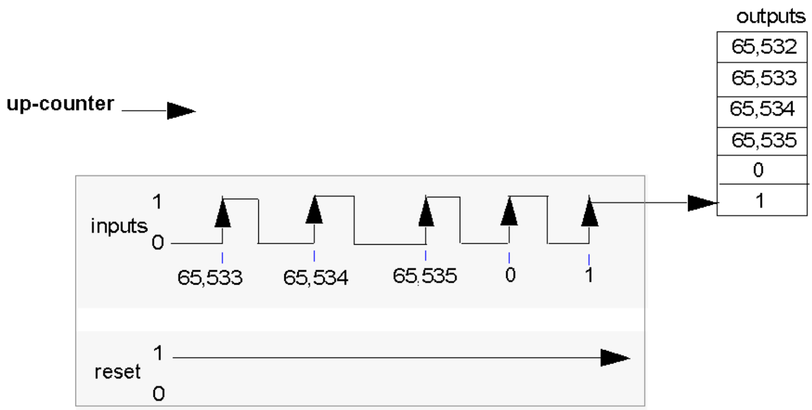
For example, say you have an up-counter with a preset value of 10. The counter will start its counting sequence at 10 and will increment by 1 each time the count input rises from 0 to 1. Suppose that the reset input drops to 0 after 3 counts:



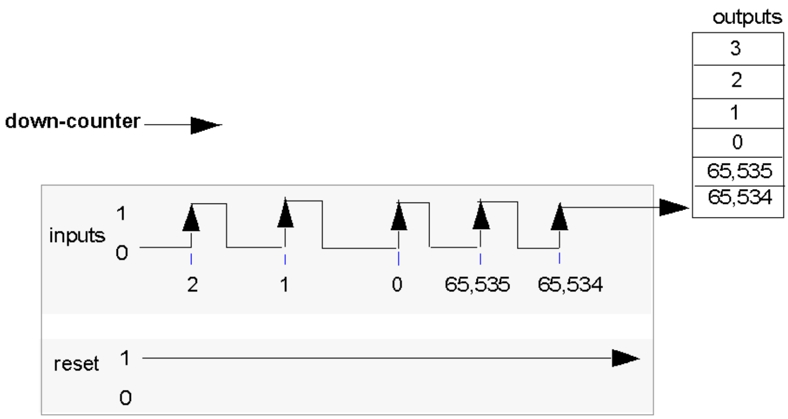
The reset input causes the counter to return to the preset value (10) and to start the up-counting process over again.

Wrap-Arounds

If an up-counter increments up to 65,535 and does not receive a reset input, it will wrap to 0 and continue to increment from there until it is reset. At reset, the counter will return to the preset value and start a new incremental up-count:



If a down-counter decrements down to 0 and does not receive a reset input, it will wrap to 65,535 and continue to decrement from there until it is reset. At reset, the counter will return to the preset value and start a new incremental down-count:



Physical Output

The output of a rising-edge counter block is a word that holds an unsigned integer value in the range 0...65,535. The physical output *(see page 325)* needs to be mapped to a digital action module.

NOTE: A counter is always the first block in a nested reflex action. The action module has to always be a digital output module, which will be configured to perform a compare action (preferably an unsigned compare, since values greater than 32,767 would be misinterpreted by a standard integer compare action). The Reflex Editor does not allow you to map the counter's output to an analog module.

You need to specify the channel to which the counter output will be mapped as **None** so that the output will be stored temporarily in an internal reflex buffer, and then used as the count input of the compare block.

Power-Up and Fallback

Upon power-up, the counter's output data is set to the preset value (if the enable in put is on).

If a detected error causes the counter block to go to its fallback state, the output freezes in its last active state. Upon removal of the detected error condition, the counter starts counting again at the point where the output was frozen.

Section 6.6
Timer Reflex Blocks

Introduction

This section describes 2 types of timer blocks, delay timers and edge timers.

Delay timer blocks start timing when a timer trigger is set, count timing intervals for some specified number of counts, and then hold the terminal count value until the trigger launches another timing operation.

Edge time blocks start timing when a timer trigger is set, count timing intervals for some specified number of counts, and then return to their start state until the trigger launches a new timing operation.

What Is in This Section?

This section contains the following topics:

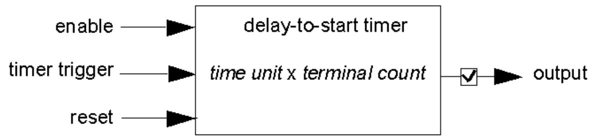
Delay-to-Start Timer Block

Summary

A delay-to-start timer block starts a timing operation when its *trigger* rises from 0 to 1. The timer needs to be preset to accumulate a user-specified *time unit* for a specified number of counts (the *terminal count*). The output from a delay-to-start timer block is a boolean value that rises to 1 when the terminal count is reached and stays at 1 as long as the terminal count is held. You may invert the value of the output.

Structure of a Delay-to-Start Timer Block

A block diagram for a delay-to-start timer is shown below:



The block has 3 inputs:

Input	Description
Enable Input	allows or stops the output from being updated
Timer Trigger	timer start command
Reset	a boolean value that stops the timer operation when it is set to 0

The block also has 2 preset values *(see page 324)*:

Preset Value	Description
Terminal Count	a user-defined number of time units
Time Unit	a number of ms in which the timer counts

When a timing operation starts, it will accumulate time units from 0 up to the terminal count (as long as the reset value is 1). When the timer reaches the terminal count, the output turns on and stays on until the timer resets. When the timer resets, it turns off.

The output is a boolean value. The standard output is 1 while the block holds the terminal count and 0 when it resets. The output may be inverted.

Time Units and Terminal Count

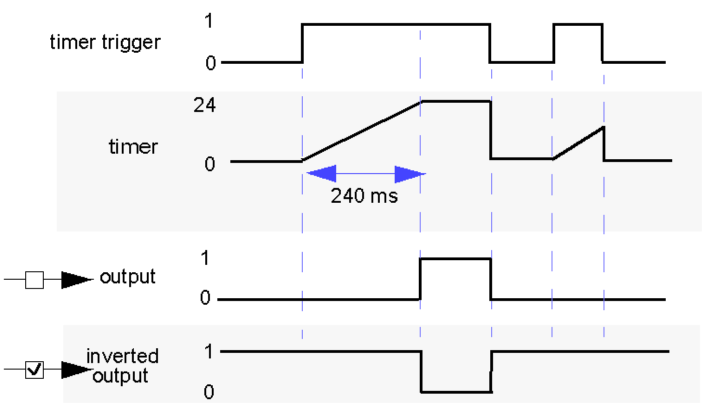
You need to preset the timer block to accumulate in 1 of the following time units:

- 1 ms
- 10 ms
- 100 ms
- 1,000 ms
- 10,000 ms

When the timer is enabled and the trigger starts the accumulation, the block will count a specified number of time units. The maximum number of unit counts allowed is called the *terminal count*. The terminal count is a user-specified integer value in the range 1...32,767.

When the timer reaches the terminal count, the accumulator stops counting time units and the output turns on (1 if the output is standard, 0 if the output is inverted). The output remains on as long as the timer accumulator holds the terminal count.

For example, suppose you specify a time unit of 10 ms and a terminal count of 24. When the timer trigger input rises from 0 to 1, the timer accumulates to 240 ms, and then stops and holds its terminal count until the trigger drops to 0:



In the example above, the output behavior is as follows:

If the timer ...	Then the output ...	Then the inverted output ...
reaches the terminal count	is set 1.	is set 0.
holds the terminal count	remains 1.	remains 0.
resets	is set back to 0.	is set back to 1.

Enable Input

A delay-to-start timer block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. The timer finishes a timing cycle if it has already started it, but it does not change the output. The output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Timer Trigger Input

The trigger input is a set of boolean 1 s and 0 s. The rising edge of the trigger input starts a timing operation, and the falling edge of the trigger input causes the timer accumulator to drop to 0.

The timer trigger input may be produced by the following:
- a constant
- a digital input from a module on the Island
- a digital output from the virtual module *(see page 326)*
- an output on the action module *(see page 330)* written to by the fieldbus master
- the output of the first reflex block if the timer is the second block in a nested reflex action *(see page 333)*
  In this case, its trigger input may be configured as the output of the first reflex block.

The value of the trigger input is important for the output of the block. If the trigger drops to 0 before the timer reaches the terminal count, the timer stops accumulating and drops to 0. When this happens, the output never turns on. If the trigger remains at 1 after the terminal count has been reached, the timer accumulator holds the terminal count value and the output rises to 1.

NOTE: At start-up, check that the trigger input provides a 0 to the timer block, see the table below.

The following applies when the timer block is enabled:

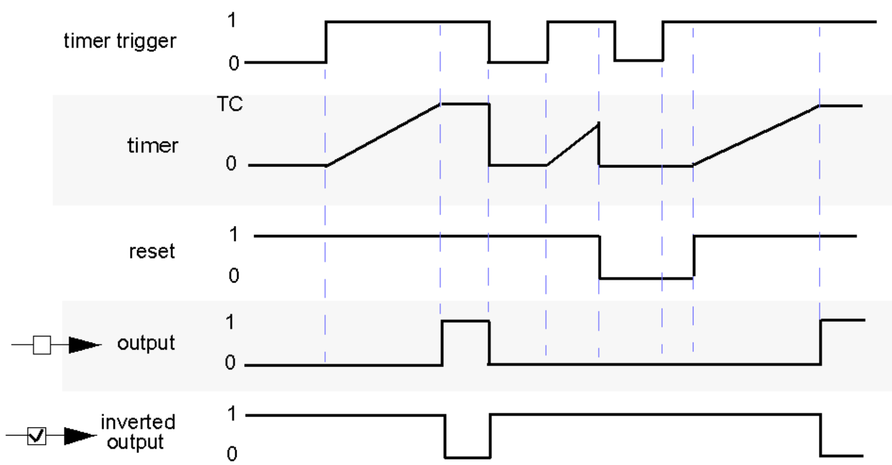
If the trigger input is ...	Then ...
1	the timer assumes that a rising-edge transition has just taken place and starts accumulating counts immediately.
0	the timer waits for the next rising-edge transition before it starts accumulating counts.

Timer Reset Input

The reset input is essentially a timer override mechanism. It may be a boolean 1 or 0. The timer is operational when the reset value is 1; it does not operate when the reset value is 0.

The reset input may be produced by the following:
- constant
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

The following timing diagram shows how the value of the reset input effects the output from the timer block:



At the beginning of the timing sequence, when the reset input is 1, the standard output is 0 (the inverted output is 1) while the timer is accumulating. The standard output rises to 1 (the inverted output drops to 0) when the terminal count (TC) is reached. When the trigger drops to 0, the timer and the standard output drop to 0 (or the inverted output rises to 1).

The second time the trigger input rises to 1, the timer begins to accumulate again. But before TC is reached the second time, the reset input drops to 0, thereby resetting the timer. The standard output remains at 0 (or the inverted output remains at 1) during this second timing sequence.

When the reset input rises back to 1, the timer begins to accumulate again starting at 0. The reason that the reset input is able to restart the timer is because the timer trigger input is 1 when the reset rises to 1. Once TC has been reached, the standard output rises to 1 again and stays there (or the inverted output drops to 0 and stays there) as long as both the trigger input and the reset input hold the terminal count.

Physical Output

The output from a delay-to-start timer block is a boolean 1 or 0 and the following applies:

If the output is ...	Then the output goes to ...
not inverted	1 when the block reaches its specified terminal count and stays at 1 as long as the timer accumulator holds the terminal count. The output falls to 0 when the block resets.
inverted	0 when the block reaches its specified terminal count and stays at 0 as long as the timer accumulator holds the terminal count. The output is 1 when the block resets.

The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the timer is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

When the output from a reflex block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write data to this bit address in the NIM, and the reflex action editor lets you use these data from the fieldbus master as an input to the block.

Power-Up and Fallback

Upon power-up, the timer’s output data is reset to 0.

If a detected error causes the timer block to go to its fallback state, the output freezes in its last active state. Upon removal of the detected error condition, the timer is reset to 0.

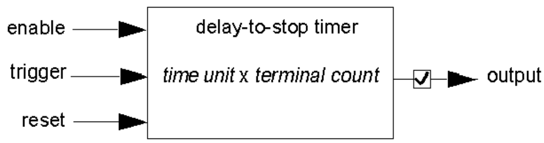
Delay-to-Stop Timer Block

Summary

The delay-to-stop timer block starts a timing operation when its *trigger* falls from 1 to 0. The timer needs to be preset to accumulate a user-specified *time unit* for a specified number of counts (the *terminal count*). The output of a delay-to-stop timer block is a boolean that goes to 0 as soon as the terminal count is reached and stays at 0 as long as the terminal count is held. Optionally, you may invert the value of the output.

Structure of a Delay-to-Stop Timer Block

A block diagram for a delay-to-stop timer is shown below:



The block has 3 inputs:

Input	Description
Enable Input	allows or stops the output from being updated
Timer Trigger	timer start command
Reset	a boolean value that stops the timer operation when it is set to 0

The block also has 2 preset values *(see page 324)*:

Preset Value	Description
Terminal Count	a user-defined number of time units
Time Unit	a number of ms in which the timer counts

When a timing operation starts, it will accumulate time units from 0 up to the terminal count. When the timer reaches the terminal count, the output turns off until the timer resets. When the timer resets, it turns on.

The output is a boolean value. The standard output is 0 while the timer holds the terminal count and 1 when the timer resets. The output may be inverted.

Time Units and Terminal Count

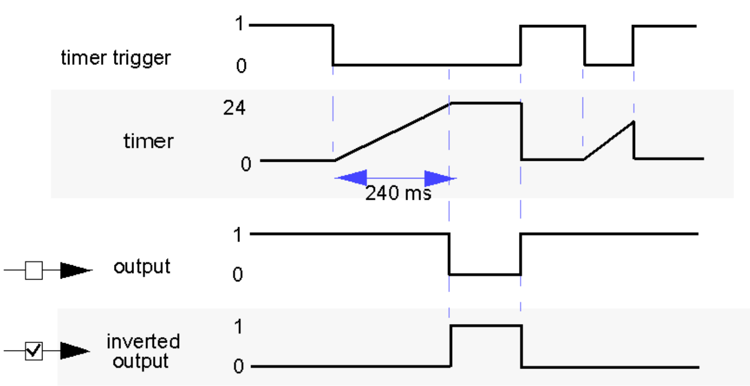
You need to preset the timer block to accumulate in 1 of the following time units:

- 1 ms
- 10 ms
- 100 ms
- 1,000 ms
- 10,000 ms

When the timer block is enabled and the trigger starts the accumulation, the block will count a specified number of time units. The maximum number of unit counts allowed is called the *terminal count*. The terminal count is a user-specified integer value in the range 1...32,767.

When the block reaches the terminal count, the accumulator stops counting time units and the output of the action turns off (0 if the output is standard, 1 if the output is inverted). The output remains off as long as the timer accumulator holds the terminal count.

For example, suppose you specify a time unit of 10 ms and a terminal count of 24. When the timer trigger input drops from 1 to 0, the timer accumulates to 240 ms, and then stops and holds its terminal count as long as the trigger remains at 0:



In the example above, the output behavior is as follows:

If the timer ...	Then the output ...	Then the inverted output ...
reaches the terminal count	is set 0.	is set 1.
holds the terminal count	remains 0.	remains 1.
resets	is set to 1.	is set to 0.

Enable Input

A delay-to-stop timer block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. The timer finishes a timing cycle if it has already started it, but it does not change the output. The output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Timer Trigger Input

The falling edge of the trigger input starts a timing operation, and the rising edge of the trigger input causes the timer accumulator to drop to 0. The trigger input may be a boolean 1 or 0.

For a delay-to-stop timer block, the value of the trigger input is important for the output from the block. If the trigger rises to 1 before the timer reaches the terminal count, the timer stops accumulating and drops to 0. When this happens, the output never turns off. If the trigger remains at 0 after the terminal count has been reached, the timer accumulator holds the terminal count value and the output turns off.

The timer trigger value may be produced by the following:
- a constant
- a digital input from a module on the Island
- a digital output from the virtual module *(see page 326)*
- an output on the action module *(see page 330)* written to by the fieldbus master
- the output of the first reflex action if the timer is the second part of a nested reflex action *(see page 333)*
  In this case, its trigger input may be configured as the output of the first reflex action.

NOTE: At start-up, check that the trigger input provides a 1 to the timer block, see the table below.

The following applies when the timer block is enabled:

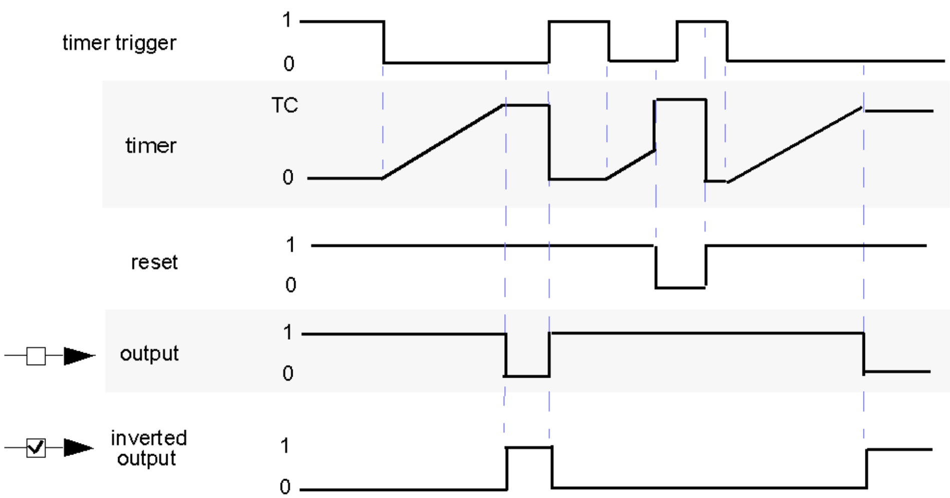
If the trigger input is ...	Then ...
0	the timer assumes that a faling-edge transition has just taken place and starts accumulating counts immediately.
1	the timer waits for the next falling-edge transition before it starts accumulating counts.

Timer Reset Input

The reset input is essentially a timer override mechanism. It may be a boolean 1 or 0. The block is operational when the reset value is 1; it does not operate when the reset value is 0.

The reset input may be produced by the following:
- constant
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

The following timing diagram shows how the value of the reset input effects the inverted output from the timer block:



At the beginning of the timing sequence, while the reset input is 1, the standard output is 1 (the inverted output is 0) before and while the block is accumulating. The standard output goes to 0 (the inverted output goes to 1) when TC is reached. When the trigger rises to 1, the timer drops to 0, and the standard output rises to 1 (the inverted output falls to 0).

The second time the trigger input falls to 0, the timer begins to accumulate again. But before the accumulation completes the second time, the reset input drops to 0, thereby sending the timer to 0. The standard output remains at 1 (the inverted output remains at 0).

When the timer trigger value drops to 0 for the third time, the timer begins to accumulate again. Once TC has been reached, the output drops to 0 again and stays at 0 as long as the trigger input is 0 and the reset input is 1.

Physical Output

The output from a delay-to-stop timer block is a boolean 1 or 0 and the following applies:

If the output is ...	Then the output goes to ...
not inverted	0 when the block has reached its specified terminal count and it will stay at 0 as long as the timer accumulator holds the terminal count. The output falls to 1 when the block resets.
inverted	1 when the block has reached its specified terminal count and will stay at 1 as long as the timer accumulator holds the terminal count. The output is 0 when the block resets.

The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the timer is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

When the output from a reflex block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write data to this bit address in the NIM, and the reflex action editor lets you use these data from the fieldbus master as an input to the block.

Power-Up and Fallback

Upon power-up, the timer’s output data is reset to the terminal count.

If a detected error causes the timer block to go to its fallback state, the output freezes in its last active state. Upon removal of the detected error condition, the timer is reset to the terminal count.

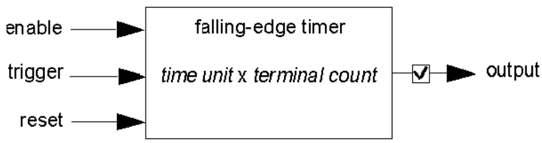
Falling-Edge Timer Block

Summary

A falling-edge timer block starts a timing operation when its *trigger* falls from 1 to 0. The timer needs to be preset to accumulate at a user-specified *time unit* for a specified number of counts (the *terminal count*). The output from a falling-edge timer block is a boolean that goes to 1 while the timer is accumulating and 0 when the timer is not accumulating time units (when the accumulator is at the terminal count). Optionally, you may invert the value of the output.

Structure of a Falling-Edge Timer Block

A block diagram for a standard falling-edge timer is shown below:



The block has 3 inputs:

Input	Description
Enable Input	allows or stops the output from being updated
Timer Trigger	timer start command
Reset	a boolean value that stops the timer operation when it is set to 0

The block also has 2 preset values *(see page 324)*:

Preset Value	Description
Terminal Count	a user-defined number of time units
Time Unit	a number of ms in which the timer counts

When a timing operation starts, it will accumulate time units from 0 up to the terminal count. While the timer is counting, the output turns on. As soon as the timer reaches the terminal count, the output turns off and remains off until the trigger starts a new counting sequence.

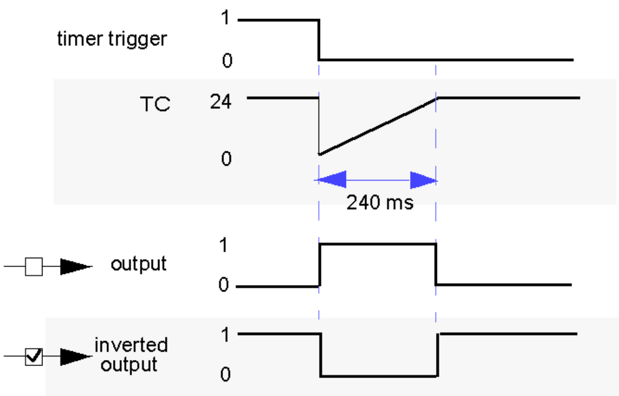
The output value is a boolean. The standard output is 1 while the timer is accumulating counts and 0 when it is not counting. The output may be inverted.

Time Units and Terminal Count

You need to preset the timer block to accumulate in 1 of the following time units:
- 1 ms
- 10 ms
- 100 ms
- 1,000 ms
- 10,000 ms

When the timer is enabled and the trigger starts the accumulation, the block will count a specified number of time units. This number is called the *terminal count*. The terminal count is a user-specified integer value in the range 0...32,767.

For example, suppose you specify a time unit of 10 ms and a terminal count of 24. When the timer trigger input drops from 1 to 0, the timer begins accumulating from 0 in 10 ms time units. It accumulates 24 time units (240 ms), and then stops accumulating and holds its terminal count:



In the example above, the output behavior is as follows:

If the timer ...	Then the output ...	Then the inverted output ...
starts accumulating	is set 1.	is set 0.
reaches the terminal count	is set to 0.	is set to 1.
holds the terminal count	remains 0.	remains 1.

Enable Input

A falling-edge timer block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. The timer finishes a timing cycle if it has already started it, but it does not change the output. The output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Timer Trigger Input

The timer trigger is essentially a timer start command. It may be a boolean 1 or 0. The block starts accumulating time units when the timer trigger drops from 1 to 0.

The timer trigger value may be produced by the following:
- a constant
- a digital input from a module on the Island
- a digital output from the virtual module *(see page 326)*
- an output on the action module *(see page 330)* written to by the fieldbus master
- the output of the first reflex block if the timer is the second block in a nested reflex action *(see page 333)*
  In this case, its trigger input may be configured as the output of the first reflex block.

NOTE: At start-up, check that the trigger input provides a 1 value to the timer block, see the table below.

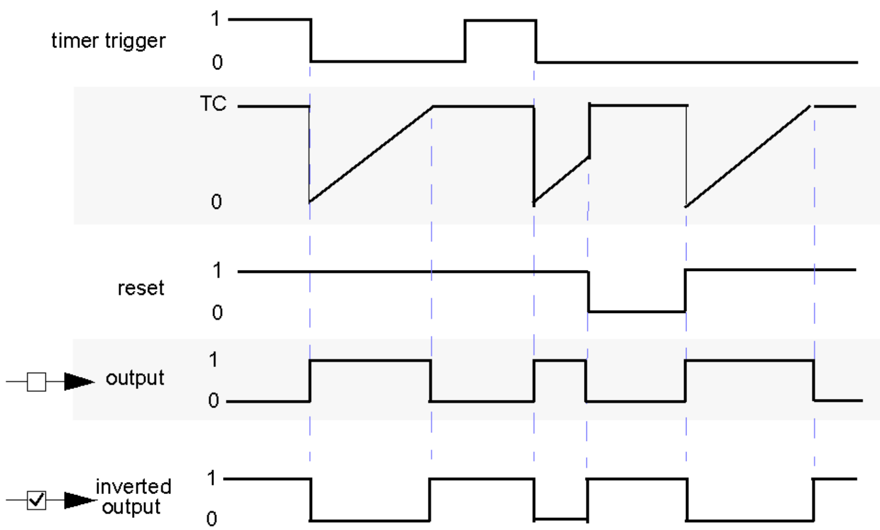
The following applies when the timer block is enabled:

If the trigger input is ...	Then ...
0	the timer assumes that a falling-edge transition has just taken place and starts accumulating counts immediately.
1	the timer waits for the next falling-edge transition before it starts accumulating counts.

Timer Reset Input

The reset input is essentially a timer override mechanism. It may be a boolean 1 or 0. The timer is operational when the reset value is 1; it does not operate when the reset value is 0.

The following timing diagram shows how the value of the reset input affects the block's output:



At the beginning of the timing sequence, while the reset input is high, the standard output rises to 1 (the inverted output drops to 0) while the block is accumulating. The standard output drops to 0 (the inverted output rises to 1) when the terminal count (TC) is reached.

The second time that the trigger drops to 0, the block begins to accumulate and the standard output rises to 1 (the inverted output drops to 0). But before TC is reached the second time, the reset input drops to 0, thereby stopping the timer and sending the standard output back to 0 (the inverted output back to 1.

When the reset input rises back to 1, the block begins to accumulate again starting at 0, and the standard output rises again to 1 (the inverted output drops to 0). The reset input is able to restart the timer because the state of the timer trigger input is 0 when the reset rises to 1.

NOTE: If the timer trigger is low when the reset goes high, the timer will start. It will not start if the trigger is high.

Physical Output

The output from a falling-edge timer block is a boolean 1 or 0 and the following applies:

If the output is ...	Then the output goes ...
not inverted	to 1 when the block is accumulating time units and to 0 when the timer is at 0 or at the terminal count.
inverted	to 0 when the timer is accumulating time units and to 1 when the timer is at 0 or at the terminal count.

The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the timer is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

When the output from a reflex block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write data to this bit address in the NIM, and the reflex action editor lets you use these data from the fieldbus master as an input to the block.

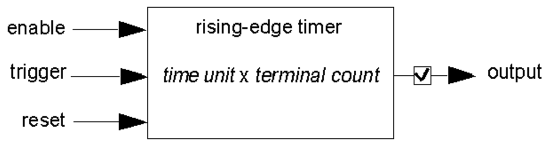
Rising-Edge Timer Block

Summary

A rising-edge timer block starts a timing operation when its *trigger* rises from 0 to 1. The block needs to be preset to accumulate at a user-specified *time unit* for a specified number of counts (the *terminal count*). The output from a rising-edge timer block is a boolean that rises to 1 while the timer is accumulating and drops to 0 when the accumulator is at the terminal count. You may invert the value of the output.

Structure of a Rising-Edge Timer Block

A block diagram for a standard rising-edge timer is shown below:



The block has 3 inputs:

Input	Description
Enable Input	allows or stops the output from being updated
Timer Trigger	timer start command
Reset	a boolean value that stops the timer operation when it is set to 0

The block also has 2 preset values *(see page 324)*:

Preset Value	Description
Terminal Count	a user-defined number of time units
Time Unit	a number of ms in which the timer counts

When a timing operation starts, it will accumulate time units from 0 up to the terminal count. While the block is counting, the output turns on. As soon as the timer reaches the terminal count, the output turns off and remains off until the trigger starts a new counting sequence.

The output is a boolean value. The standard output is 1 while the block is accumulating counts and 0 when it is not counting. The output may be inverted.

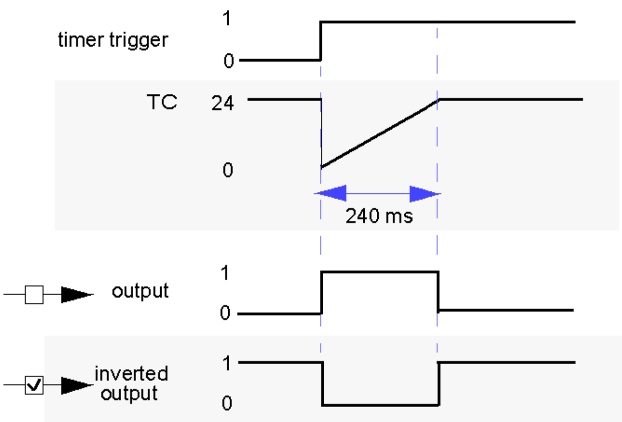
Time Units and Terminal Count

You need to preset the timer block to accumulate in 1 of the following time units:

- 1 ms
- 10 ms
- 100 ms
- 1,000 ms
- 10,000 ms

When the timer is enabled and the trigger starts the accumulation, the block will count a specified number of time units. This number is called the *terminal count*. The terminal count is a user-specified integer value in the range 0...32,767.

For example, suppose you specify a time unit of 10 ms and a terminal count of 24. When the timer trigger input rises from 0 to 1, the timer begins accumulating from 0 in 10 ms time units. It accumulates 24 time units (240 ms), and then stops accumulating and holds its terminal count:



As the timing diagram above shows, a standard output rises to 1 while the timer is accumulating and drops to 0 whenever the timer is not accumulating. An inverted output drops to 0 while the timer is accumulating and rises to 1 whenever the timer is not accumulating.

In the example above, the output behavior is as follows:

If the timer ...	Then the output ...	Then the inverted output ...
starts accumulating	is set 1.	is set 0.
reaches the terminal count	is set to 0.	is set to 1.
holds the terminal count	remains 0.	remains 1.

Enable Input

A rising-edge timer block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. The timer finishes a timing cycle if it has already started it, but it does not change the output. The output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Timer Trigger Input

The timer trigger is essentially a timer start command. It may be a boolean 1 or 0. The timer starts accumulating time units when the timer trigger rises from 0 to 1.

The timer trigger value may be produced by the following:
- a constant
- a digital input from a module on the Island
- a digital output from the virtual module *(see page 326)*
- an output on the action module *(see page 330)* written to by the fieldbus master
- the output of the first reflex block if the timer is the second block in a nested reflex action *(see page 333)*
  In this case, its trigger input may be configured as the output of the first reflex block.

NOTE: At start-up, check that the trigger input provides a 0 to the timer block, see the table below.

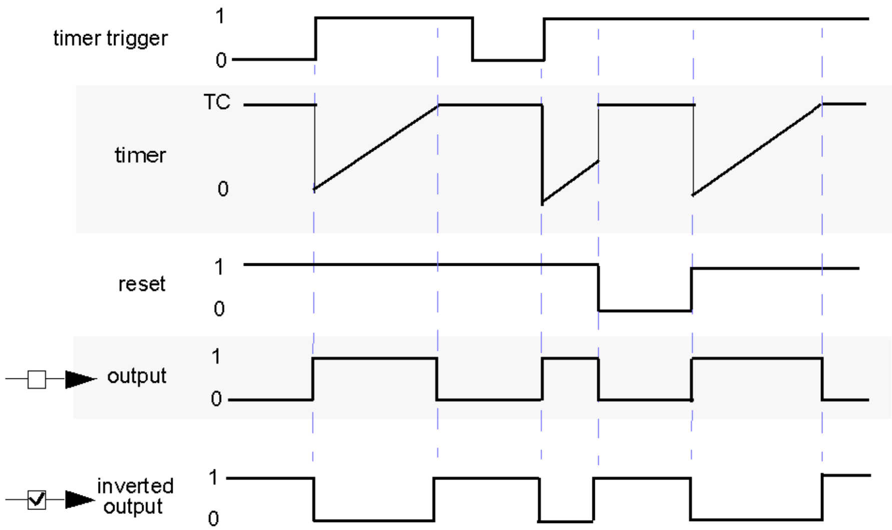
The following applies when the timer block is enabled:

If the trigger input is ...	Then ...
1	the timer assumes that a rising-edge transition has just taken place and starts accumulating counts immediately.
0	the timer waits for the next rising-edge transition before it starts accumulating counts.

Timer Reset Input

The reset input is essentially a timer override mechanism. It may be a boolean 1 or 0. The block is operational when the reset value is 1; it does not operate when the reset value is 0.

The following timing diagram shows how the value of the reset input effects the output from the block:



At the beginning of the timing sequence, while the reset input is 1, the standard output rises to 1 (the inverted output drops to 0) while the block is accumulating. The standard output drops to 0 (the inverted output rises to 1) when the terminal count (TC) is reached.

The second time that the trigger drops to 0, the timer begins to accumulate and the standard output rises to 1 again (the inverted output drops to 0 again). But before TC is reached the second time, the reset input drops to 0, thereby stopping the block and sending the standard output to 0 (the inverted output to 1).

When the reset input rises back to 1, the timer begins to accumulate again starting at 0, and the standard output rises again to 1 (the inverted output falls to 0). The reset input restarts the timer because the state of the timer trigger input is high when the reset goes high.

Physical Output

The output from a rising-edge timer block is a boolean 1 or 0 and the following applies:

If the output is ...	Then the output goes ...
not inverted	to 1 when the block is accumulating time units and to 0 when the timer is at 0 or at the terminal count.
inverted	to 0 when the timer is accumulating time units and to 1 when the timer is at 0 or at the terminal count.

The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the compare is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

When the output from a reflex block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write data to this bit address in the NIM, and the reflex action editor lets you use these data from the fieldbus master as an input to the block.

Section 6.7
Analog Latch Reflex Blocks

Introduction

In this section, 2 types of analog latch blocks are described, edge latches and level latches.

Edge latches latch an analog value on either the rising edge or falling edge of the block’s trigger. The output from the block remains latched until the trigger causes another input value to be latched. The output is always a latched value.

Level latches produce an output that is latched when the trigger is at 1 level (1 or 0) and unlatched when the trigger is not at that level.

What Is in This Section?

This section contains the following topics:

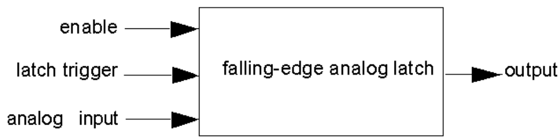
Falling-Edge Analog Latch Block

Summary

A falling-edge analog latch block produces an output that latches the value of an analog input when the *trigger* drops from 1 to 0. The output remains latched while the trigger is at 0 and while it transitions back to 1. If the latch trigger transitions from 1 to 0 again, the block latches the output to the value of the analog input at the time of the second transition. The output is always a latched analog value in the form of a 16-bit word.

Structure of a Falling-Edge Analog Latch Block

A block diagram for a falling-edge analog latch is shown below:



The block has 3 inputs:

Input	Description
Enable Input	turns the block on or off
Latch Trigger	causes the block to latch onto the value of the analog input
Analog Input	an integer value that is latched when the trigger fires

The analog input may be an unsigned integer value in the range 0...65,535 or a signed integer value in the range -32,768...+32,767.

The output of the block is the latched value. It may be an unsigned integer value in the range 0...65,535 or a signed integer value in the range -32,768...+32,767.

Enable Input

A falling-edge analog latch can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Latch Trigger Input

The latch trigger may be a boolean 1 or 0. When the value of the trigger falls from 1 to 0, the block latches the value of the analog input and the latched value becomes the block's output. The latched output value remains set until the trigger falls again from 1 to 0, producing a new latched output.

The latch trigger value may be produced by the following:

- a constant
- a digital input from a module on the Island
- a digital output from the virtual module *(see page 326)*
- the output from the first reflex block if the latch is the second block in a nested reflex action *(see page 333)*
  In this case, the latch trigger input may be the output from the first reflex block.

NOTE: At start-up, check that the trigger input provides a 1 to the latch block, see the table below.

The following applies when the latch block is enabled:

If the trigger input is ...	Then ...
0	the latch assumes that a falling-edge transition has just taken place and immediately latches the value.
1	the latch waits for the next falling-edge transition before it latches a value.

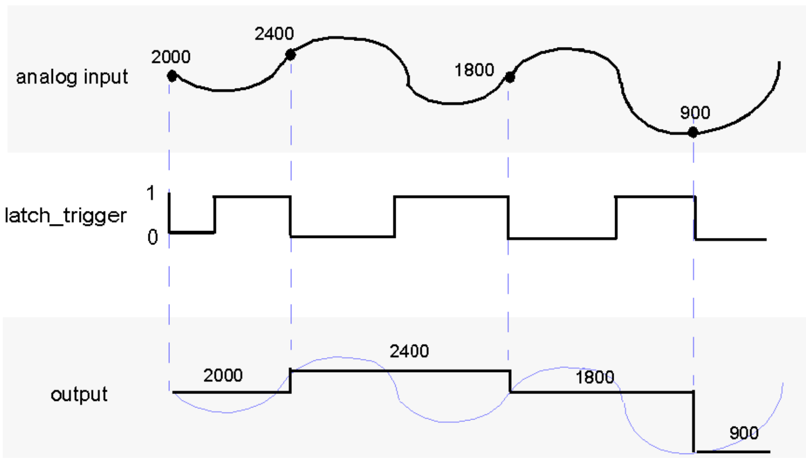
Analog Input

The analog input may be an unsigned integer value in the range 0...65,535 or a signed integer value in the range -32,768...+32,767. The value of the input will be latched by the trigger when it falls from 1 to 0.

The input value may be produced by the following:

- analog input from a module on the Island
- analog output from the virtual module *(see page 326)*

The following timing diagram shows how the value of the trigger effects the output of the latch block:



The output behavior of the block is as follows:

If the trigger ...	Then the output ...
falls from 1 to 0 the first time	is latched to the current value of the operational input (2,000).
remains 0	holds the last value (2,000).
is set from 0 to 1	holds the last value (2,000).
remains 1	holds the last value (2,000).
falls from 1 to 0 the second time	is latched to the current value of the operational input (2,400).

Physical Output

The output from a falling-edge analog latch block is a 16-bit word. It may be an unsigned integer in the range 0...65,535 or a signed integer in the range -32,768...+32,767. The output value is the value of the analog input at the moment of the last falling edge of the latch trigger.

NOTE: The type of output value from this block matches the type of input value, for instance, if you input an unsigned integer value, the output will be an unsigned integer value. The block itself does not discriminate between an unsigned value of 65,535 and a signed value of -32,768. You need to check that the block output is being sent to an output module that can handle the output value correctly.

The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is an analog output module on the Island bus	specify 1 of the analog output channels as the destination for the reflex output.
the latch is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

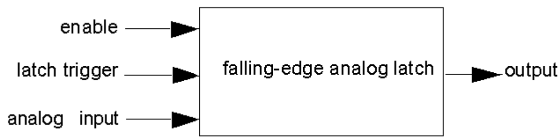
Rising-Edge Analog Latch Block

Summary

A rising-edge analog latch block produces an output that latches the value of an analog input when the block's *trigger* rises from 0 to 1. The output remains latched to this value while the trigger is at 1 and while it transitions back to 0. If the latch trigger transitions from 0 to 1 again, the block latches the output to the value of the analog input at the time of the second transition. The output of a rising-edge analog latch action is a latched analog value in the form of a 16-bit word.

Structure of a Rising-Edge Analog Latch Block

A block diagram for a standard rising-edge analog latch is shown below:



The block has 3 inputs:

Input	Description
Enable Input	turns the block on or off
Latch Trigger	causes the block to latch onto the value of the analog input
Analog Input	an integer value that is latched when the trigger fires

The analog input may be an unsigned integer value in the range 0...65,535 or a signed integer value in the range -32,768...+32,767.

The output of the block is the latched value. It may be an unsigned integer value in the range 0...65,535 or a signed integer value in the range -32,768...+32,767.

Enable Input

A rising-edge analog latch can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled.That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Latch Trigger Input

The latch trigger may be a boolean 1 or 0. When the value of the trigger rises from 0 to 1, the block latches the value of the analog input and that latched value becomes the block's output. The latched output value remains set until the trigger rises again from 0 to 1, producing a new latched output.

The latch trigger value may be produced by the following:
- a constant
- a digital input from a module on the Island
- a digital output from the virtual module *(see page 326)*
- the output from the first reflex block if the latch is the second block in a nested reflex action *(see page 333)*
  In this case, the latch trigger input may be the output from the first reflex block.

NOTE: At start-up, check that the trigger input provides a 0 to the latch block, see the table below.

The following applies when the latch block is enabled:

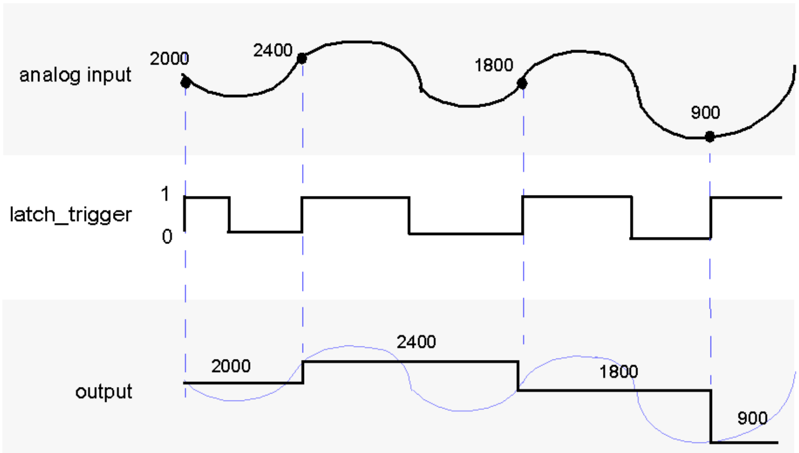
If the trigger input is ...	Then ...
1	the latch assumes that a rising-edge transition has just taken place and immediately latches the value.
0	the latch waits for the next rising-edge transition before it latches a value.

Analog Input

The analog input may be an unsigned integer value in the range 0...65,535 or a signed integer value in the range -32,768...+32,767. The value of the input is latched by the trigger value when it rises from 0 to 1.

The input value may be produced by the following:
- analog input from a module on the Island
- analog output from the virtual module *(see page 326)*

The following timing diagram shows how the value of the trigger effects the output from the block:



The output behavior of the block is as follows:

If the trigger ...	Then the output ...
rises from 0 to 1 the first time	is latched to the current value of the operational input (2,000).
remains 1	holds the last value (2,000).
is set from 1 to 0	holds the last value (2,000).
remains 0	holds the last value (2,000).
rises from 0 to 1 the second time	is latched to the current value of the operational input (2,400).

Physical and Logical Output

The output of a rising-edge analog latch block is a 16-bit word. It may be an unsigned integer in the range 0...65,535 or a signed integer in the range -32,768...+32,767. The output is the value of the analog input at the moment of the last falling edge of the latch trigger.

NOTE: The type of output value from this block matches the type of input value, for instance, if you input an unsigned integer value, the output will be an unsigned integer value. The block itself does not discriminate between an unsigned value of 65,535 and a signed value of -32,768. You need to check that the block output is being sent to an output module that can handle the output value correctly.

The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is an analog output module on the Island bus	specify 1 of the analog output channels as the destination for the reflex output.
the latch is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

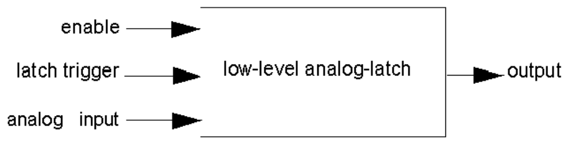
Low-Level Analog Latch Block

Summary

A low-level analog latch block produces a latched output when the its *trigger* is 0 and an unlatched output when the trigger is 1. When the action is unlatched, the value of the output is identical to the value of the analog input. When the action is latched, the value of the output is latched to the value of the analog input at the moment when the latch trigger has fallen from 1 to 0. The output of a low-level analog latch action is a latched or unlatched analog value in the form of a 16-bit word.

Structure of a Low-Level Analog Latch Block

A block diagram for a low-level analog latch is shown below:



The block has 3 inputs:

Input	Description
Enable Input	turns the block on or off
Latch Trigger	causes the block to latch onto the value of the analog input
Analog Input	an integer value that is latched when the trigger fires

The analog value may be an unsigned integer value in the range from 0...65,535 or a signed integer value in the range -32,768...+32,767.

The output is the value of the block, either latched or unlatched. It may be an unsigned integer value in the range 0...65,535 or a signed integer value in the range -32,768...+32,767.

Enable Input

A low-level analog latch can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- a digital input from a module on the Island
- a digital output from the virtual module *(see page 326)*

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled.That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Latch Trigger Input

The latch trigger may be a boolean 1 or 0. When the value is 0, the block latches the value of the analog input, and that latched value becomes the output from the block. When the trigger value is 1, the output is unlatched and equal to the value of the analog input.

The latch trigger value may be produced by the following:

- a constant
- a digital input from a module on the Island
- a digital output from the virtual module
- the output from the first reflex block if the latch is the second block in a nested reflex action *(see page 333)*
  In this case, the latch trigger input may be the output from the first reflex block.

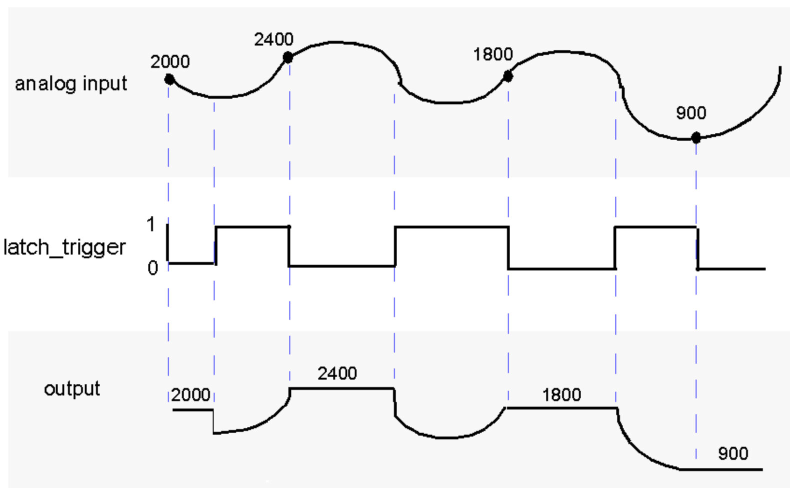
Analog Input

The analog input may be an unsigned integer value in the range 0...65,535 or a signed integer value in the range -32,768...+32,767. The value of the input is latched when the value of the latch trigger is 0 and unlatched when the value of the latch trigger is 1.

The input value may be produced by the following:

- analog input from a module on the Island
- analog output from the virtual module *(see page 326)*

The following timing diagram shows how the value of the trigger effects the block’s output:



The output behavior of the block is as follows:

If the trigger ...	Then the output ...
falls from 1 to 0 the first time	is latched to the current value of the operational input (2,000).
remains 0	holds the last value (2,000).
is set from 0 to 1	is unlatched and set to current value of the operational input.
remains 1	echoes the current value of the operational input.
falls from 1 to 0 the second time	is latched to the current value of the operational input (2,400).

Physical Output

The output of a low-level analog latch block is a 16-bit word. It may be an unsigned integer in the range 0...65,535 or a signed integer in the range -32,768...+32,767. The output is latched to the value of the analog input when the value of the latch trigger is 0, and it is unlatched when the value of the trigger is 1.

NOTE: The type of output value from this block matches the type of input value, for instance, if you input an unsigned integer value, the output will be an unsigned integer value. The block itself does not discriminate between an unsigned value of 65,535 and a signed value of -32,768. You need to check that the block output is being sent to an output module that can handle the output value correctly.

The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is an analog output module on the Island bus	specify 1 of the analog output channels as the destination for the reflex output.
the latch is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

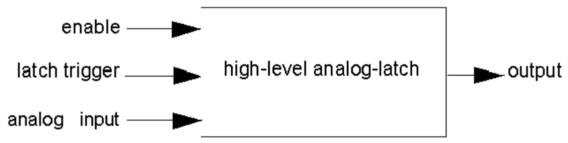
High-Level Analog Latch Block

Summary

A high-level analog latch block produces a latched output when the block's *trigger* is 1 and an unlatched output when the trigger is 0. When the block is unlatched, the value of the output is identical to the value of the analog input. When the block is latched, the value of the output is latched to the value of the analog input when the latch trigger rises from 0 to 1. The output is an analog value in the form of a 16-bit word.

Structure of a High-Level Analog Latch Block

A block diagram for a high-level analog latch is shown below:



The block has 3 inputs:

Input	Description
Enable Input	turns the block on or off
Latch Trigger	causes the block to latch onto the value of the analog input
Analog Input	an integer value that is latched when the trigger fires

The analog input may be an unsigned integer value in the range 0...65,535 or a signed integer value in the range -32,768...+32,767.

The output is the value of the block, either latched or unlatched. It may be an unsigned integer value in the range 0...65,535 or a signed integer value in the range -32,768...+32,767.

Enable Input

A high-level analog latch block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Latch Trigger Input

The latch trigger may be a boolean 1 or 0. When the trigger is 1, the value of the analog input is latched, and that value becomes the output from the block as long as the trigger is 1. When the trigger value is 0, the block latches the value of the analog input and that latched value becomes the block's output. When the trigger value is 1, the output is unlatched and equal to the value of the analog input.

The latch trigger value may be produced by the following:

- a constant
- a digital input from a module on the Island
- a digital output from the virtual module
- the output from the first reflex block if the latch is the second block in a nested reflex action *(see page 333)*
  In this case, the latch trigger input may be the output from the first reflex block.

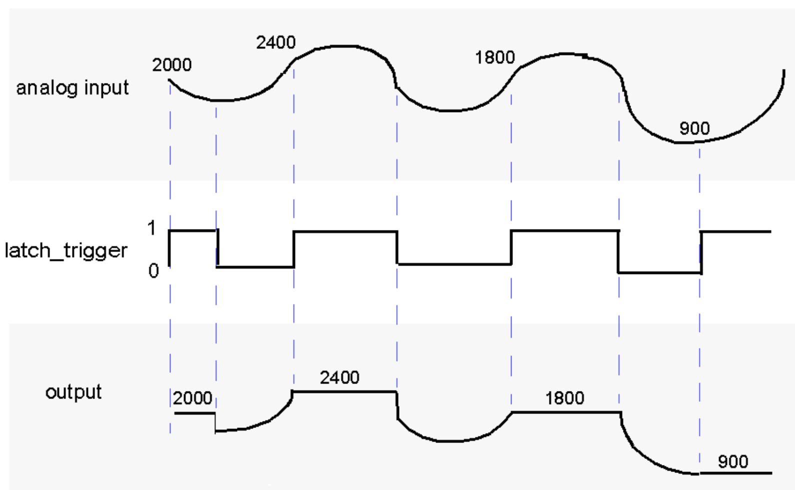
Analog Input

The analog input may be an unsigned integer value in the range 0...65,535 or a signed integer value in the range -32,768...+32,767. The value of the input is latched when the latch trigger is 1 and unlatched when the latch trigger is 0.

The input value may be produced by the following:

- analog input from a module on the Island
- analog output from the virtual module *(see page 326)*

The following timing diagram shows how the value of the trigger effects the output from the block:



The output behavior of the block is as follows:

If the trigger ...	Then the output ...
rises from 0 to 1 the first time	is latched to the current value of the operational input (2,000).
remains 1	holds the last value (2,000).
is set from 1 to 0	is unlatched and set to current value of the operational input.
remains 0	echoes the current value of the operational input.
rises from 0 to 1 the second time	is latched to the current value of the operational input (2,400).

Physical Output

The output of a high-level analog latch block is a 16-bit word. It may be an unsigned integer in the range 0...65,535 or a signed integer in the range -32,768...+32,767. The output is latched to the value of the analog input when the latch trigger is 1 and unlatched when the trigger is 0.

NOTE: The type of output value from this block matches the type of input value, for instance, if you input an unsigned integer value, the output will be an unsigned integer value. The block itself does not discriminate between an unsigned value of 65,535 and a signed value of -32,768. You need to check that the block output is being sent to an output module that can handle the output value correctly.

The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is an analog output module on the Island bus	specify 1 of the analog output channels as the destination for the reflex output.
the latch is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

Section 6.8
Digital Latch Reflex Blocks

Introduction

In this section, 2 types of digital latch blocks are described, edge latches and level latches.

Edge latch blocks latch a digital value on the rising edge or falling edge of the block's trigger. The output remains latched until the trigger causes another input value to be latched; the output is always a latched value.

Level latches produce an output that is latched when the trigger value is at 1 level (1 or 0) and unlatched when the trigger value is not at that level.

What Is in This Section?

This section contains the following topics:

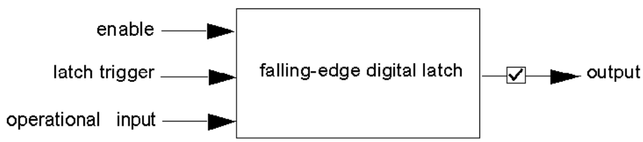
Falling-Edge Digital Latch Block

Summary

A falling-edge digital latch block produces an output that latches the value of an operational input when the *trigger* falls from 1 to 0. The output remains latched to that digital value. The output from the block is a boolean 1 or 0. You may invert the value of the output by marking the check box on the output line of the block.

Structure of a Falling-Edge Digital Latch Block

A block diagram for a falling-edge digital latch is shown below:



The block has 3 inputs:

Input	Description
Enable Input	turns the block on or off
Latch Trigger	causes the block to latch onto the value of the operational input
Operational Input	a stream of boolean values on which the latch operates

Enable Input

A falling-edge digital latch block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Latch Trigger Input

The latch trigger may be a boolean 1 or 0. When it transitions from 1 to 0, the value of the operational input is latched. If the output is standard, the value of the operational input becomes the output from the block; if the output is inverted, the inverted value of the operational input becomes the output from the block.

The latch trigger value may be produced by the following:

- a constant
- a digital input from a module on the Island
- a digital output from the virtual module *(see page 326)*
- an output on the action module *(see page 330)* written to by the fieldbus master
- the output from the first reflex block if the latch is the second block in a nested reflex action *(see page 333)*
  In this case, the latch trigger input may be the output from the first reflex block.

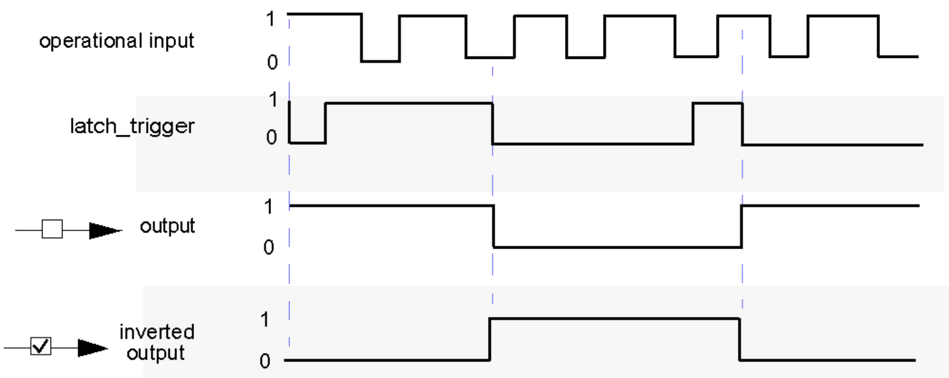
Operational Input

The operational input is a stream of boolean 1 s and 0 s that is latched by the falling edge of the latch trigger.

It may be produced by the following:

- a constant
- a digital input from a module on the Island
- a digital output from the virtual module *(see page 326)*
- an output on the action module *(see page 330)* written to by the fieldbus master
- the output from the first reflex block if the latch is the second block in a nested reflex action *(see page 333)*
  In this case, the latch trigger input may be the output from the first reflex block.

The following timing diagram shows how the value of the trigger effects the output of the latch action:



In the example illustrated above, the output behavior of the block is as follows:

If the trigger ...	Then the output ...	Then the inverted output ...
falls from 1 to 0 the first time	is latched to the current value of the operational input (1).	is latched to the inverted value of the current operational input (0).
remains 0	holds the last value (1).	holds the last value (0).
is set from 0 to 1	holds the last value (1).	holds the last value (0).
remains 1	holds the last value (1).	holds the last value (0).
falls from 1 to 0 the second time	is latched to the current value of the operational input (0).	is latched to the inverted value of the current operational input (1).

Physical Output

The output from a falling-edge digital latch block is a boolean 1 or 0. The output is always a latched value, determined by the value of the operational input when the trigger transitions from 1 to 0:

If the output is ...	Then the output latches ...
not inverted	the value of the operational input when the latch trigger falls from 1 to 0.
inverted	the inverse of the value of the operational input when the latch trigger falls from 1 to 0.

The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the latch is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

When the output from a block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write data to this bit address in the NIM, and the reflex action editor lets you use this data from the fieldbus master as an input to the block.

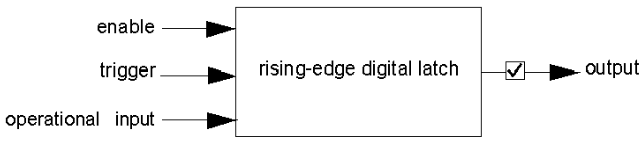
Rising-Edge Digital Latch Block

Summary

A rising-edge digital latch block produces an output that latches the value of an operational input when the block's *trigger* rises from 0 to 1. The output remains latched when the trigger falls to 0 and until the trigger rises to 1 again. The output is a boolean 1 or 0. You may invert the value of the output by marking the check box on the output line of the block.

Structure of a Rising-Edge Digital Latch Block

A block diagram for a rising-edge digital latch is shown below:



The block has 3 inputs:

Input	Description
Enable Input	turns the block on or off
Latch Trigger	causes the block to latch onto the value of the operational input
Operational Input	a stream of boolean values on which the latch operates

Enable Input

A rising-edge digital latch block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:
- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Latch Trigger Input

The latch trigger may be a boolean 1 or 0. When it rises from 0 to 1, the value of the operational input is latched. If the output is standard, the value of the operational input becomes the output of the action; if the output is inverted, the inverted value of the operational input becomes the output of the action.

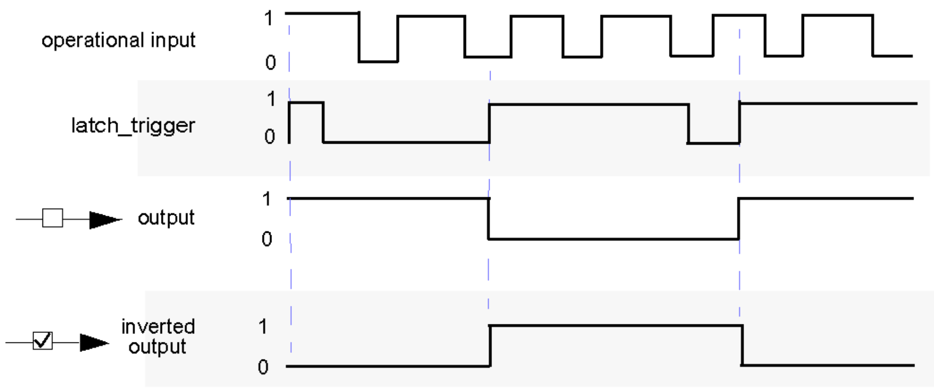
The latch trigger value may be produced by the following:
- a constant
- a digital input from a module on the Island
- a digital output from the virtual module
- an output on the action module written to by the fieldbus master
- the output from the first reflex block if the latch is the second block in a nested reflex action *(see page 333)*
  In this case, the latch trigger input may be the output from the first reflex block.

Operational Input

The operational input is a pulse train of boolean 1 s and 0 s that will be latched at any time by the rising edge of the latch trigger.

It may be produced by the following:
- a digital input or output from a module on the Island
- a digital output from the virtual module
- an input data bit from a channel on the action module
- the output from the first reflex block if the latch is the second block in a nested reflex action *(see page 333)*
  In this case, the latch trigger input may be the output from the first reflex block.

The following timing diagram shows how the value of the trigger effects the output from the block:



At the beginning of the timing sequence, the operational input is high when the trigger rises from 0 to 1. The standard output is latched at 1 (the inverted output is latched at 0).

The output value remains latched until the latch trigger rises from 0 to 1 a second time. At that moment, the operational input is low. The standard output is latched at 0 (the inverted output is latched at 1).

When the latch trigger rises from 0 to 1 the third time, the operational input is high. The standard output is latched again at 1 (the inverted output is latched at 0).

In the example illustrated above, the output behavior of the block is as follows:

If the trigger ...	Then the output ...	Then the inverted output ...
rises from 0 to 1 the first time	is latched to the current value of the operational input (1).	is latched to the inverted value of the current operational input (0).
remains 1	holds the last value (1).	holds the last value (0).
is set from 1 to 0	holds the last value (1).	holds the last value (0).
remains 0	holds the last value (1).	holds the last value (0).
rises from 0 to 1 the second time	is latched to the current value of the operational input (0).	is latched to the inverted value of the current operational input (1).

Physical Output

The output from a rising-edge digital latch block is a boolean 1 or 0. The output is always a latched value, determined by the value of the operational input when the trigger transitions from 0 to 1:

If the output is ...	Then the output latches ...
not inverted	the value of the operational input when the latch trigger rises from 0 to 1.
inverted	the inverse of the value of the operational input when the latch trigger rises from 0 to 1.

The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the latch is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

When the output from a block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write data to this bit address in the NIM, and the reflex action editor lets you use this data from the fieldbus master as an input to the block.

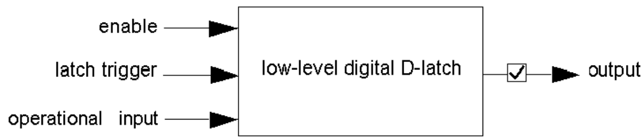
Low-Level Digital D-Latch Block

Summary

A low-level digital D-latch block produces a latched output when the block's *trigger* is 0 and an unlatched output when the trigger is 1. The output is a boolean 1 or 0. You may invert the value of the output by marking the check box on the output line of the block.

Structure of a Low-Level Digital D-Latch Block

A block diagram for a standard low-level digital D-latch is shown below:



The block has 3 inputs:

Input	Description
Enable Input	turns the block on or off
Latch Trigger	causes the block to latch onto or unlatch from the value of the operational input
Operational Input	a stream of boolean values on which the latch operates

Enable Input

A low-level digital D-latch can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:

- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Latch Trigger Input

The latch trigger may be a boolean 1 or 0. When it is 0, the value of the operational input is latched. When the trigger value is 1, the output is unlatched.

The latch trigger value may be produced by the following:

- a constant
- a digital input from a module on the Island
- a digital output from the virtual module
- an output on the action module written to by the fieldbus master
- the output from the first reflex block if the latch is the second block in a nested reflex action *(see page 333)*
  In this case, the latch trigger input may be the output from the first reflex block.

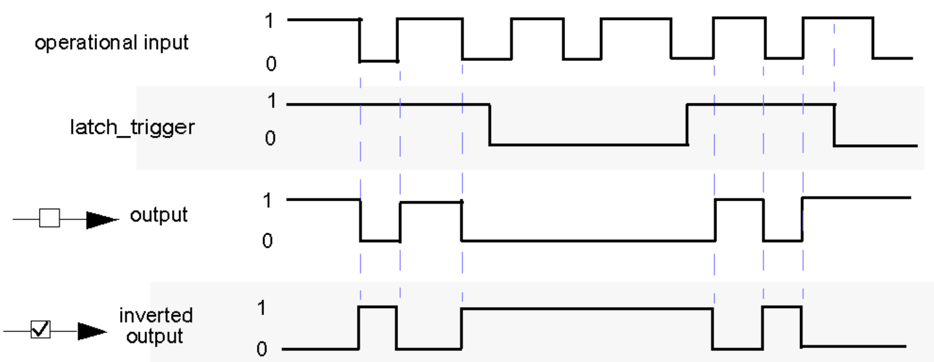
Operational Input

The operational input is a stream of boolean 1 s and 0 s that can be latched and unlatched by the latch trigger.

It may be produced by the following:

- a constant
- a digital input from a module on the Island
- a digital output from the virtual module
- an output on the action module written to by the fieldbus master
- the output from the first reflex block if the latch is the second block in a nested reflex action *(see page 333)*
  In this case, the latch trigger input may be the output from the first reflex block.

The following timing diagram shows how the value of the trigger effects the output from the block:



At the beginning of the timing sequence, the standard output echoes the value of the operational input (the inverted output echoes the inverse of the operational input) as long as the trigger is high. Then, the output behavior is as follows:

If the trigger ...	Then the output ...	Then the inverted output ...
is set from 1 to 0 the first time	is latched to the current value of the operational input (0).	is latched to the inverted value of the current operational input (1).
remains 0	holds the last value (0).	holds the last value (1).
is set from 0 to 1	is unlatched and set to the current value of the operational input.	is unlatched and set to the inverted value of the current operational input.
remains 1	echoes the current value of the operational input.	echoes the inverted value of the current operational input.
is set from 1 to 0 the second time	is latched to the current value of the operational input (1).	is latched to the inverted value of the current operational input (0).

Physical Output

The output from a low-level digital D-latch block is a boolean 1 or 0. The output is latched when the trigger value is 0 and unlatched when the trigger value is 1:

If the output is ...	Then the output echoes ...
not inverted	the current operational input when the latch trigger is high, and it latches the value of the operational input at the moment that the trigger drops from 1 to 0.
inverted	the inverse of the current operational input when the latch trigger is high, and it latches a value that is the inverse of the value of the operational input at the moment that the trigger drops from 1 to 0.

The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the latch is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

When the output from a block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write data to this bit address in the NIM, and the reflex action editor lets you use this data from the fieldbus master as an input to the block.

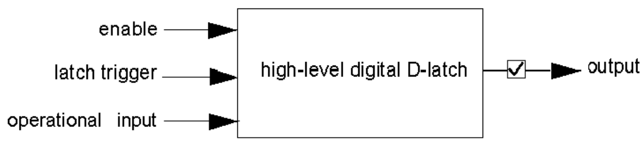
High-Level Digital D-Latch Block

Summary

A high-level digital D-latch block produces a latched output when the block's *trigger* is 1 and an unlatched output when the trigger is 0. The output is a boolean 1 or 0. You may invert the value of the output by marking the check box on the output line of the block.

Structure of a High-Level Digital D-Latch Block

A block diagram for a standard high-level digital D-latch is shown below:



The block has 3 inputs:

Input	Description
Enable Input	turns the block on or off
Latch Trigger	causes the block to latch onto or unlatch from the value of the operational input
Operational Input	a stream of boolean values on which the latch operates

Enable Input

A high-level digital D-latch block can be enabled either by a boolean 1 or an Always Enabled constant. It can be disabled by a boolean 0 or an Always Disabled constant.

If the enable input is a boolean, it may be produced by the following:

- digital input from a module on the Island
- digital output from the virtual module *(see page 326)*
- output on the action module *(see page 330)* written to by the fieldbus master

When the enable input is a boolean 0 or an Always Disabled constant, the block is disabled. That means that the action does not execute and the output is frozen in the state it was in when the block became disabled. The block continues to process inputs but does not act on them. If the block becomes enabled, it immediately begins acting on the latest set of inputs received.

Latch Trigger Input

The latch trigger may be a boolean 1 or 0. When it is 1, the value of the operational input is latched. When the trigger value is 0, the output is unlatched.

The latch trigger value may be produced by the following:

- a constant
- a digital input from a module on the Island
- a digital output from the virtual module *(see page 326)*
- an output on the action module *(see page 330)* written to by the fieldbus master
- the output from the first reflex block if the latch is the second block in a nested reflex action *(see page 333)*
  In this case, the latch trigger input may be the output from the first reflex block.

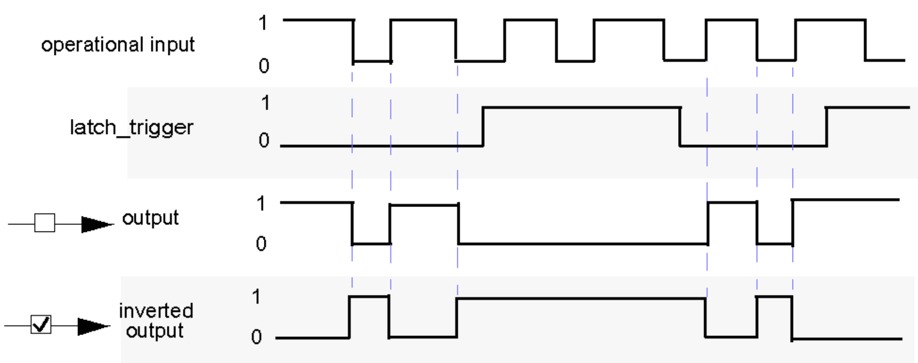
Operational Input

The operational input is a stream of boolean 1 s and 0 s that will be latched and unlatched by the trigger value.

It may be produced by the following:

- a constant
- a digital input from a module on the Island
- a digital output from the virtual module *(see page 326)*
- an output on the action module *(see page 330)* written to by the fieldbus master
- the output from the first reflex block if the latch is the second block in a nested reflex action *(see page 333)*
  In this case, the latch trigger input may be the output from the first reflex block.

The following timing diagram shows how the value of the trigger effects the output from the block:



At the beginning of the timing sequence, the standard output echoes the value of the operational input (the inverted output echoes the inverse of the operational input) as long as the trigger is low. Then, the output behavior is as follows:

If the trigger ...	Then the output ...	Then the inverted output ...
is set from 0 to 1 the first time	is latched to the current value of the operational input (0).	is latched to the inverted value of the current operational input (1).
remains 1	holds the last value (0).	holds the last value (1).
is set from 1 to 0	is unlatched and set to the current value of the operational input.	is unlatched and set to the inverted value of the current operational input.
remains 0	echoes the current value of the operational input.	echoes the inverted value of the current operational input.
is set from 0 to 1 the second time	is latched to the current value of the operational input (1).	is latched to the inverted value of the current operational input (0).

Physical Output

The output from a high-level digital D-latch block is a boolean 1 or 0. The output is latched when the trigger value is 1 and unlatched when the trigger value is 0:

If the output is ...	Then the output echoes ...
not inverted	the current operational input when the latch trigger is low, and it latches the value of the operational input at the moment that the trigger rises from 0 to 1.
inverted	the inverse of the current operational input when the latch trigger is low, and it latches a value that is the inverse of the value of the operational input at the moment that the trigger rises from 0 to 1.

The physical output *(see page 325)* needs to be mapped to an action module:

If ...	Then ...
the action module is a digital output module on the Island bus	specify 1 of the digital output channels as the destination for the reflex output.
the latch is the first block in a nested reflex action *(see page 333)*	specify the channel as **None** because the action module needs to be the same as the one specified for the second reflex block.

When the output from a block is mapped to a channel on a digital output module, that channel becomes dedicated to the reflex action and can no longer use data from the fieldbus master to update its field device. The fieldbus master still has the ability to write data to this bit address in the NIM, and the reflex action editor lets you use this data from the fieldbus master as an input to the block.

Glossary

!

100Base-T

An adaptation of the IEEE 802.3u (Ethernet) standard, the 100Base-T standard uses twisted-pair wiring with a maximum segment length of 100 m (328 ft) and terminates with an RJ-45 connector. A 100Base-T network is a baseband network capable of transmitting data at a maximum speed of 100 Mbit/s. "Fast Ethernet" is another name for 100Base-T, because it is ten times faster than 10Base-T.

10Base-T

An adaptation of the IEEE 802.3 (Ethernet) standard, the 10Base-T standard uses twisted-pair wiring with a maximum segment length of 100 m (328 ft) and terminates with an RJ-45 connector. A 10Base-T network is a baseband network capable of transmitting data at a maximum speed of 10 Mbit/s.

802.3 frame

A frame format, specified in the IEEE 802.3 (Ethernet) standard, in which the header specifies the data packet length.

A

agent

1. SNMP – the SNMP application that runs on a network device.
2. Fipio – a slave device on a network.

analog input

A module that contains circuits that convert analog DC input signals to digital values that can be manipulated by the processor. By implication, these analog inputs are direct. That means a data table value directly reflects the analog signal value.

analog output

A module that contains circuits that transmit an analog DC signal proportional to a digital value input to the module from the processor. By implication, these analog outputs are direct. That means a data table value directly controls the analog signal value.

application object

In CAN-based networks, application objects represent device-specific functionality, such as the state of input or output data.

ARP

The ARP (address resolution protocol) is the IP network layer protocol, which uses ARP to map an IP address to a MAC (hardware) address.

auto baud

The automatic assignment and detection of a common baud rate as well as the ability of a device on a network to adapt to that rate.

auto-addressing

The assignment of an address to each Island bus I/O module and preferred device.

auto-configuration

The ability of Island modules to operate with predefined default parameters. A configuration of the Island bus based completely on the actual assembly of I/O modules.

B

basic I/O

Low-cost Advantys STB input/output modules that use a fixed set of operating parameters. A basic I/O module cannot be reconfigured with the Advantys Configuration Software and cannot be used in reflex actions.

basic network interface

A low-cost Advantys STB network interface module that supports up to 12 Advantys STB I/O modules. A basic NIM does not support the Advantys Configuration Software, reflex actions, nor the use of an HMI panel.

basic power distribution module

A low-cost Advantys STB PDM that distributes sensor power and actuator power over a single field power bus on the Island. The bus provides a maximum of 4 A total power. A basic PDM includes a 5 A fuse.

BootP

BootP (bootstrap protocol) is an UDP/IP protocol that allows an internet node to obtain its IP parameters based on its MAC address.

BOS

BOS stands for beginning of segment. When more than 1 segment of I/O modules is used in an Island, an STB XBE 1200 or an STB XBE 1300 BOS module is installed in the first position in each extension segment. Its job is to carry Island bus communications to and generate logic power for the modules in the extension segment. Which BOS module has to be selected depends on the module types that shall follow.

bus arbitrator

A master on a Fipio network.

C

CAN

The CAN (controller area network) protocol (ISO 11898) for serial bus networks is designed for the interconnection of smart devices (from multiple manufacturers) in smart systems for real-time industrial applications. CAN multi-master systems provide high data integrity through the implementation of broadcast messaging and advanced diagnostic mechanisms. Originally developed for use in automobiles, CAN is now used in a variety of industrial automation control environments.

CANopen protocol

An open industry standard protocol used on the internal communication bus. The protocol allows the connection of any enhanced CANopen device to the Island bus.

CI

This abbreviation stands for command interface.

CiA

CiA (CAN in Automation) is a non-profit group of manufacturers and users dedicated to developing and supporting CAN-based higher layer protocols.

CIP

Common Industrial Protocol. Networks that include CIP in the application layer can communicate seamlessly with other CIP-based networks. For example, the implementation of CIP in the application layer of an Ethernet TCP/IP network creates an EtherNet/IP environment. Similarly, CIP in the application layer of a CAN network creates a DeviceNet environment. Devices on an EtherNet/IP network can therefore communicate with devices on a DeviceNet network via CIP bridges or routers.

COB

A COB (communication object) is a unit of transportation (a message) in a CAN-based network. Communication objects indicate a particular functionality in a device. They are specified in the CANopen communication profile.

configuration

The arrangement and interconnection of hardware components within a system and the hardware and software selections that determine the operating characteristics of the system.

CRC

cyclic redundancy check. Messages that implement this detected error mechanism have a CRC field that is calculated by the transmitter according to the message’s content. Receiving nodes recalculate the field. Disagreement in the two codes indicates a difference between the transmitted message and the one received.

CSMA/CS

carrier sense multiple access/collision detection. CSMA/CS is a MAC protocol that networks use to manage transmissions. The absence of a carrier (transmission signal) indicates that a network channel is idle. Multiple nodes may try to simultaneously transmit on the channel, which creates a collision of signals. Each node detects the collision and immediately terminates transmission. Messages from each node are retransmitted at random intervals until the frames are successfully transmitted.

D

DDXML

Device Description eXtensible Markup Language

device name

A customer-driven, unique logical personal identifier for an Ethernet NIM. A device name (or *role name*) is created when you combine the numeric rotary switch setting with the NIM (for example, STBNIP2212_010).

After the NIM is configured with a valid device name, the DHCP server uses it to identify the island at power up.

DeviceNet protocol

DeviceNet is a low-level, connection-based network that is based on CAN, a serial bus system without a defined application layer. DeviceNet, therefore, defines a layer for the industrial application of CAN.

DHCP

dynamic host configuration protocol. A TCP/IP protocol that allows a server to assign an IP address based on a device name (host name) to a network node.

differential input

A type of input design where two wires (+ and -) are run from each signal source to the data acquisition interface. The voltage between the input and the interface ground are measured by two high-impedance amplifiers, and the outputs from the two amplifiers are subtracted by a third amplifier to yield the difference between the + and - inputs. Voltage common to both wires is thereby removed. When ground differences exist, use differential signalling instead of single ended signalling to help reduce cross channel noise.

digital I/O

An input or output that has an individual circuit connection at the module corresponding directly to a data table bit or word that stores the value of the signal at that I/O circuit. It allows the control logic to have discrete access to the I/O values.

DIN

Deutsche industrial norms. A German agency that sets engineering and dimensional standards and now has worldwide recognition.

Drivecom Profile

The Drivecom profile is part of CiA DSP 402 (profile), which defines the behavior of drives and motion control devices on CANopen networks.

E

economy segment

A special type of STB I/O segment created when an STB NCO 1113 economy CANopen NIM is used in the first location. In this implementation, the NIM acts as a simple gateway between the I/O modules in the segment and a CANopen master. Each I/O module in an economy segment acts as a independent node on the CANopen network. An economy segment cannot be extended to other STB I/O segments, preferred modules or enhanced CANopen devices.

EDS

electronic data sheet. The EDS is a standardized ASCII file that contains information about a network device’s communications functionality and the contents of its object dictionary. The EDS also defines device-specific and manufacturer-specific objects.

EIA

Electronic Industries Association. An organization that establishes electrical/electronic and data communication standards.

EMC

electromagnetic compatibility. Devices that meet EMC requirements can operate within a system’s expected electromagnetic limits without interruption.

EMI

electromagnetic interference. EMI can cause an interruption or disturbance in the performance of electronic equipment. It occurs when a source electronically transmits a signal that interferes with other equipment.

EOS

This abbreviation stands for end of segment. When more than 1 segment of I/O modules is used in an Island, an STB XBE 1000 or an STB XBE 1100 EOS module is installed in the last position in every segment that has an extension following it. The EOS module extends Island bus communications to the next segment. Which EOS module has to be selected depends on the module types that shall follow.

Ethernet

A LAN cabling and signaling specification used to connect devices within a defined area, e.g., a building. Ethernet uses a bus or a star topology to connect different nodes on a network.

Ethernet II

A frame format in which the header specifies the packet type, Ethernet II is the default frame format for NIM communications.

EtherNet/IP

EtherNet/IP (the Ethernet Industrial Protocol) is especially suited to factory applications in which there is a need to control, configure, and monitor events within an industrial system. The ODVA-specified protocol runs CIP (the Common Industrial Protocol) on top of standard Internet protocols, like TCP/IP and UDP. It is an open local (communications) network that enables the interconnectivity of all levels of manufacturing operations from the plant's office to the sensors and actuators on its floor.

F

fallback state

A known state to which an Advantys STB I/O module can return in the event that its communication connection is not open.

fallback value

The value that a device assumes during fallback. Typically, the fallback value is either configurable or the last stored value for the device.

FED_P

Fipio extended device profile. On a Fipio network, the standard device profile type for agents whose data length is more than 8 words and equal to or less than 32 words.

Fipio

Fieldbus Interface Protocol (FIP). An open fieldbus standard and protocol that conforms to the FIP/World FIP standard. Fipio is designed to provide low-level configuration, parameterization, data exchange, and diagnostic services.

Flash memory

Flash memory is nonvolatile memory that can be overwritten. It is stored on a special EEPROM that can be erased and reprogrammed.

FRD_P

Fipio reduced device profile. On a Fipio network, the standard device profile type for agents whose data length is two words or less.

FSD_P

Fipio standard device profile. On a Fipio network, the standard device profile type for agents whose data length is more than two words and equal to or less than 8 words.

full scale

The maximum level in a specific range—e.g., in an analog input circuit the maximum allowable voltage or current level is at full scale when any increase beyond that level is over-range.

function block

A function block performs a specific automation function, such as speed control. A function block comprises configuration data and a set of operating parameters.

function code

A function code is an instruction set commanding 1 or more slave devices at a specified address(es) to perform a type of action, e.g., read a set of data registers and respond with the content.

G

gateway

A program or hardware that passes data between networks.

global_ID

global_identifier. A 16-bit integer that uniquely identifies a device's location on a network. A global_ID is a symbolic address that is universally recognized by all other devices on the network.

GSD

generic slave data (file). A device description file, supplied by the device's manufacturer, that defines a device's functionality on a Profibus DP network.

H

HMI

human-machine interface. An operator interface, graphical, for industrial equipment.

hot swapping

Replacing a component with a like component while the system remains operational. When the replacement component is installed, it begins to function automatically.

HTTP

hypertext transfer protocol. The protocol that a web server and a client browser use to communicate with one another.

I

I/O base

A mounting device, designed to seat an Advantys STB I/O module, connect it on a DIN rail, and connect it to the Island bus. It provides the connection point where the module can receive either 24 VDC or 115/230 VAC from the input or output power bus distributed by a PDM.

I/O module

In a programmable controller system, an I/O module interfaces directly to the sensors and actuators of the machine/process. This module is the component that mounts in an I/O base and provides electrical connections between the controller and the field devices. Normal I/O module capacities are offered in a variety of signal levels and capacities.

I/O scanning
The continuous polling of the Advantys STB I/O modules performed by the COMS to collect data bits, status, nd diagnostics information.

IEC
International Electrotechnical Commission Carrier. Founded in 1884 to focus on advancing the theory and practice of electrical, electronics, and computer engineering, and computer science. EN 61131-2 is the specification that deals with industrial automation equipment.

IEC type 1 input
Type 1 digital inputs support sensor signals from mechanical switching devices such as relay contacts and push buttons operating in normal environmental conditions.

IEC type 2 input
Type 2 digital inputs support sensor signals from solid state devices or mechanical contact switching devices such as relay contacts, push buttons (in normal or harsh environmental conditions), and 2- or 3-wire proximity switches.

IEC type 3 input
Type 3 digital inputs support sensor signals from mechanical switching devices such as relay contacts, push buttons (in normal-to-moderate environmental conditions), 3-wire proximity switches and 2-wire proximity switches that have:
- a voltage drop of no more than 8 V
- a minimum operating current capability less than or equal to 2.5 mA
- a maximum off-state current less than or equal to 1.5 mA

IEEE
Institute of Electrical and Electronics Engineers, Inc. The international standards and conformity assessment body for all fields of electrotechnology, including electricity and electronics.

IGMP
(Internet group management protocol). This Internet standard for multicasting allows a host to subscribe to a particular multicast group.

industrial I/O
An Advantys STB I/O module designed at a moderate cost for typical continuous, high-duty-cycle applications. Modules of this type often feature standard IEC threshold ratings, providing user-configurable parameter options, on-board protection, good resolution, and field wiring options. They are designed to operate in moderate-to-high temperature ranges.

input filtering
The amount of time that a sensor has to hold its signal on or off before the input module detects the change of state.

input polarity
An input channel’s polarity determines when the input module sends a 1 and when it sends a 0 to the master controller. If the polarity is *normal*, an input channel sends a 1 to the controller when its field sensor turns on. If the polarity is *reverse*, an input channel sends a 0 to the controller when its field sensor turns on.

input response time

The time it takes for an input channel to receive a signal from the field sensor and put it on the Island bus.

INTERBUS protocol

The INTERBUS fieldbus protocol observes a master/slave network model with an active ring topology, having all devices integrated in a closed transmission path.

IOC object

Island operation control object. A special object that appears in the CANopen object dictionary when the remote virtual placeholder option is enabled in a CANopen NIM. It is a 16-bit word that provides the fieldbus master with a mechanism for issuing reconfiguration and start requests.

IOS object

Island operation status object. A special object that appears in the CANopen object dictionary when the remote virtual placeholder option is enabled in a CANopen NIM. It is a 16-bit word that reports the success of reconfiguration and start requests or records diagnostic information in the event that a request is not completed.

IP

internet protocol. That part of the TCP/IP protocol family that tracks the internet addresses of nodes, routes outgoing messages, and recognizes incoming messages.

IP Rating

Ingress Protection rating according to IEC 60529. Each IP rating requires the following standards to be met with respect to a rated device:
- IP20 modules are protected against ingress and contact of objects larger than 12.5 mm. The module is not protected against harmful ingress of water.
- IP67 modules are completely protected against ingress of dust and contact. Ingress of water in harmful quantity is not possible when the enclosure is immersed in water up to 1 m.

L

LAN

local area network. A short-distance data communications network.

light industrial I/O

An Advantys STB I/O module designed at a low cost for less rigorous (e.g., intermittent, low-duty-cycle) operating environments. Modules of this type operate in lower temperature ranges with lower qualification and agency requirements and limited on-board protection; they have limited or no user-configuration options.

linearity

A measure of how closely a characteristic follows a straight-line function.

LSB

least significant bit, least significant byte. The part of a number, address, or field that is written as the rightmost single value in conventional hexadecimal or binary notation.

M

MAC address

media access control address. A 48-bit number, unique on a network, that is programmed into each network card or device when it is manufactured.

mandatory module

When an Advantys STB I/O module is configured to be mandatory, it should be present and healthy in the Island configuration for the Island to be operational. If a mandatory module is inoperable or is removed from its location on the Island bus, the Island goes to a pre-operational state. By default, all I/O modules are not mandatory. You should use the Advantys Configuration Software to set this parameter.

master/slave model

The direction of control in a network that implements the master/slave model is from the master to the slave devices.

Modbus

Modbus is an application layer messaging protocol. Modbus provides client and server communications between devices connected on different types of buses or networks. Modbus offers many services specified by function codes.

MOV

metal oxide varistor. A 2-electrode semiconductor device with a voltage-dependant nonlinear resistance that drops markedly as the applied voltage is increased. It is used to suppress transient voltage surges.

MSB

most significant bit, most significant byte. The part of a number, address, or field that is written as the leftmost single value in conventional hexadecimal or binary notation.

N

N.C. contact

normally closed contact. A relay contact pair that is closed when the relay coil is de-energized and open when the coil is energized.

N.O. contact

normally open contact. A relay contact pair that is open when the relay coil is de-energized and closed when the coil is energized.

NEMA

National Electrical Manufacturers Association

network cycle time

The time that a master requires to complete a single scan of the configured I/O modules on a network device; typically expressed in microseconds.

NIM
network interface module. This module is the interface between an Island bus and the fieldbus network of which the Island is a part. A NIM enables all the I/O on the Island to be treated as a single node on the fieldbus. The NIM also provides 5 V of logic power to the Advantys STB I/O modules in the same segment as the NIM.

NMT
network management. NMT protocols provide services for network initialization, diagnostic control, and device status control.

O

object dictionary
Part of the CANopen device model that provides a map to the internal structure of CANopen devices (according to CANopen profile DS-401). A device's object dictionary (also called the *object directory*) is a lookup table that describes the data types, communications objects, and application objects the device uses. By accessing a particular device's object dictionary through the CANopen fieldbus, you can predict its network behavior and build a distributed application.

ODVA
Open Devicenet Vendors Association. The ODVA supports the family of network technologies that are built on the Common Industrial Protocol (EtherNet/IP, DeviceNet, and CompoNet).

open industrial communication network
A distributed communication network for industrial environments based on open standards (EN 50235, EN50254, and EN50170, and others) that allows the exchange of data between devices from different manufacturers.

output filtering
The amount that it takes an output channel to send change-of-state information to an actuator after the output module has received updated data from the NIM.

output polarity
An output channel's polarity determines when the output module turns its field actuator on and when it turns the actuator off. If the polarity is *normal*, an output channel turns its actuator on when the master controller sends it a 1. If the polarity is *reverse*, an output channel turns its actuator on when the master controller sends it a 0.

output response time
The time it takes for an output module to take an output signal from the Island bus and send it to its field actuator.

P

parameterize
To supply the required value for an attribute of a device at run-time.

PDM

power distribution module. A module that distributes either AC or DC field power to a cluster of I/O modules directly to its right on the Island bus. A PDM delivers field power to the input modules and the output modules. It is important that all the I/O installed directly to the right of a PDM be in the same voltage group—either 24 VDC, 115 VAC, or 230 VAC.

PDO

process data object. In CAN-based networks, PDOs are transmitted as unconfirmed broadcast messages or sent from a producer device to a consumer device. The transmit PDO from the producer device has a specific identifier that corresponds to the receive PDO of the consumer devices.

PE

protective ground. A return line across the bus to keep improper currents generated at a sensor or actuator device out of the control system.

peer-to-peer communications

In peer-to-peer communications, there is no master/slave or client/server relationship. Messages are exchanged between entities of comparable or equivalent levels of functionality, without having to go through a third party (like a master device).

PLC

programmable logic controller. The PLC is the brain of an industrial manufacturing process. It automates a process as opposed to relay control systems. PLCs are computers suited to survive the harsh conditions of the industrial environment.

PowerSuite Software

PowerSuite Software is a tool for configuring and monitoring control devices for electric motors, including ATV31x, ATV71, and TeSys U.

preferred module

An I/O module that functions as an auto-addressable device on an Advantys STB Island but is not in the same form factor as a standard Advantys STB I/O module and therefore does not fit in an I/O base. A preferred device connects to the Island bus via an EOS module and a length of a preferred module extension cable. It can be extended to another preferred module or back into a BOS module. If it is the last device on the Island, it should be terminated with a 120 Ω terminator.

premium network interface

A premium NIM has advanced features over a standard or basic NIM.

prioritization

An optional feature on a standard NIM that allows you to selectively identify digital input modules to be scanned more frequently during a the NIM’s logic scan.

process I/O

An Advantys STB I/O module designed for operation at extended temperature ranges in conformance with IEC type 2 thresholds. Modules of this type often feature high levels of on-board diagnostics, high resolution, user-configurable parameter options, and higher levels of agency approval.

process image

A part of the NIM firmware that serves as a real-time data area for the data exchange process. The process image includes an input buffer that contains current data and status information from the Island bus and an output buffer that contains the current outputs for the Island bus, from the fieldbus master.

producer/consumer model

In networks that observe the producer/consumer model, data packets are identified according to their data content rather than by their node address. All nodes *listen* on the network and consume those data packets that have appropriate identifiers.

Profibus DP

Profibus Decentralized Peripheral. An open bus system that uses an electrical network based on a shielded 2-wire line or an optical network based on a fiber-optic cable. DP transmission allows for high-speed, cyclic exchange of data between the controller CPU and the distributed I/O devices.

Q

QoS

(quality of service). The practice of assigning different priorities to traffic types for the purpose of regulating data flow on the network. In an Industrial network, QoS can help provide a predictable level of network performance.

R

reflex action

A simple, logical command function configured locally on an Island bus I/O module. Reflex actions are executed by Island bus modules on data from various Island locations, like input and output modules or the NIM. Examples of reflex actions include compare and copy operations.

repeater

An interconnection device that extends the permissible length of a bus.

reverse polarity protection

Use of a diode in a circuit to help protect against damage and unintended operation in the event that the polarity of the applied power is accidentally reversed.

rms

root mean square. The effective value of an alternating current, corresponding to the DC value that produces the same heating effect. The rms value is computed as the square root of the average of the squares of the instantaneous amplitude for 1 complete cycle. For a sine wave, the rms value is 0.707 times the peak value.

role name

A customer-driven, unique logical personal identifier for an Ethernet NIM. A role name (or *device name*) is created when you:

- combine the numeric rotary switch setting with the NIM (for example, STBNIP2212_010), or . . .
- edit the **Device Name** setting in the NIM's embedded web server pages

After the NIM is configured with a valid role name, the DHCP server uses it to identify the island at power up.

RSTP

(rapid spanning tree protocol). Allows a network design to include spare (redundant) links that provide automatic backup paths when an active link becomes inoperable, without loops or manual enabling/disabling of backup links. Loops should be avoided because they result in flooding the network.

RTD

resistive temperature detect. An RTD device is a temperature transducer composed of conductive wire elements typically made of platinum, nickel, copper, or nickel-iron. An RTD device provides a variable resistance across a specified temperature range.

RTP

run-time parameters. RTP lets you monitor and modify selected I/O parameters and Island bus status registers of the NIM while the Advantys STB Island is running. The RTP feature uses 5 reserved output words in the NIM’s process image (the RTP request block) to send requests, and 4 reserved input words in the NIM’s process image (the RTP response block) to receive responses. Available only in standard NIMs running firmware version 2.0 or higher.

Rx

reception. For example, in a CAN-based network, a PDO is described as an RxPDO of the device that receives it.

S

SAP

service access point. The point at which the services of 1 communications layer, as defined by the ISO OSI reference model, is made available to the next layer.

SCADA

supervisory control and data acquisition. Typically accomplished in industrial settings by means of microcomputers.

SDO

service data object. In CAN-based networks, SDO messages are used by the fieldbus master to access (read/write) the object directories of network nodes.

segment

A group of interconnected I/O and power modules on an Island bus. An Island should have at least 1 segment and, depending on the type of NIM used, may have as many as 7 segments. The first (leftmost) module in a segment needs to provide logic power and Island bus communications to the I/O modules on its right. In the primary or basic segment, that function is filled by a NIM. In an extension segment, that function is filled by an STB XBE 1200 or an STB XBE 1300 BOS module.

SELV

safety extra low voltage. A secondary circuit designed so that the voltage between any 2 accessible parts (or between 1 accessible part and the PE terminal for Class 1 equipment) does not exceed a specified value under normal conditions or under single-fault conditions.

SIM

subscriber identification module. Originally intended for authenticating users of mobile communications, SIMs now have multiple applications. In Advantys STB, configuration data created or modified with the Advantys Configuration Software can be stored on a SIM (referred to as the “removable memory card”) and then written to the NIM’s Flash memory.

single-ended inputs

An analog input design technique whereby a wire from each signal source is connected to the data acquisition interface, and the difference between the signal and ground is measured. For the success of this design technique, 2 conditions are imperative: the signal source should be grounded, and the signal ground and data acquisition interface ground (the PDM lead) should have the same potential.

sink load

An output that, when turned on, receives DC current from its load.

size 1 base

A mounting device, designed to seat an STB module, install it on a DIN rail, and connect it to the Island bus. It is 13.9 mm (0.55 in.) wide and 128.25 mm (5.05 in.) high.

size 2 base

A mounting device, designed to seat an STB module, install it on a DIN rail, and connect it to the Island bus. It is 18.4 mm (0.73 in.) wide and 128.25 mm (5.05 in.) high.

size 3 base

A mounting device, designed to seat an STB module, install it on a DIN rail, and connect it to the Island bus. It is 28.1 mm (1.11 in.) wide and 128.25 mm (5.05 in.) high.

slice I/O

An I/O module design that combines a small number of channels (between 2 and 6) in a small package. The idea is to allow a system developer to purchase just the right amount of I/O and to be able to distribute it around the machine in an efficient, mechatronics way.

SM_MPS

state management_message periodic services. The applications and network management services used for process control, data exchange, diagnostic message reporting, and device status notification on a Fipio network.

SNMP

simple network management protocol. The UDP/IP standard protocol used to manage nodes on an IP network.

snubber

A circuit generally used to suppress inductive loads—it consists of a resistor in series with a capacitor (in the case of an RC snubber) and/or a metal-oxide varistor placed across the AC load.

source load

A load with a current directed into its input; has to be driven by a current source.

standard I/O

Any of a subset of Advantys STB input/output modules designed at a moderate cost to operate with user-configurable parameters. A standard I/O module may be reconfigured with the Advantys Configuration Software and, in most cases, may be used in reflex actions.

standard network interface

An Advantys STB network interface module designed at moderate cost to support the configuration capabilities, multi-segment design and throughput capacity suitable for most standard applications on the Island bus. An Island run by a standard NIM can support up to 32 addressable Advantys STB and/or preferred I/O modules, up to 12 of which may be standard CANopen devices.

standard power distribution module

An Advantys STB module that distributes sensor power to the input modules and actuator power to the output modules over two separate power buses on the Island. The bus provides a maximum of 4 A to the input modules and 8 A to the output modules. A standard PDM requires a 5 A fuse for the input modules and an 8 A fuse for the outputs.

STD_P

standard profile. On a Fipio network, a standard profile is a fixed set of configuration and operating parameters for an agent device, based on the number of modules that the device contains and the device's total data length. There are 3 types of standard profiles: Fipio reduced device profile (FRD_P), Fipio standard device profile (FSD_P), and the Fipio extended device profile (FED_P).

stepper motor

A specialized DC motor that allows discrete positioning without feedback.

subnet

A part of a network that shares a network address with the other parts of a network. A subnet may be physically and/or logically independent of the rest of the network. A part of an internet address called a subnet number, which is ignored in IP routing, distinguishes the subnet.

surge suppression

The process of absorbing and clipping voltage transients on an incoming AC line or control circuit. Metal-oxide varistors and specially designed RC networks are frequently used as surge suppression mechanisms.

T

TC

thermocouple. A TC device is a bimetallic temperature transducer that provides a temperature value by measuring the voltage differential caused by joining together two different metals at different temperatures.

TCP

transmission control protocol. A connection-oriented transport layer protocol that provides full-duplex data transmission. TCP is part of the TCP/IP suite of protocols.

telegram

A data packet used in serial communication.

TFE

transparent factory Ethernet. Schneider Electric's open automation framework based on TCP/IP.

Tx

transmission. For example, in a CAN-based network, a PDO is described as a TxPDO of the device that transmits it.

U

UDP

user datagram protocol. A connectionless mode protocol in which messages are delivered in a datagram to a destination computer. The UDP protocol is typically bundled with the Internet Protocol (UPD/IP).

V

varistor

A 2-electrode semiconductor device with a voltage-dependant nonlinear resistance that drops markedly as the applied voltage is increased. It is used to suppress transient voltage surges.

voltage group

A grouping of Advantys STB I/O modules, all with the same voltage requirement, installed directly to the right of the appropriate power distribution module (PDM) and separated from modules with different voltage requirements. Install modules with different voltage requirements in different voltage groups.

VPCR object

virtual placeholder configuration read object. A special object that appears in the CANopen object dictionary when the remote virtual placeholder option is enabled in a CANopen NIM. It provides a 32-bit subindex that represents the actual module configuration used in a physical Island.

VPCW object

virtual placeholder configuration write object. A special object that appears in the CANopen object dictionary when the remote virtual placeholder option is enabled in a CANopen NIM. It provides a 32-bit subindex where the fieldbus master can write a module reconfiguration. After the fieldbus writes to the VPCW subindex, it can issue a reconfiguration request to the NIM that begins the remote virtual placeholder operation.

W

watchdog timer

A timer that monitors a cyclical process and is cleared at the conclusion of each cycle. If the watchdog runs past its programmed time period, it reports a time-out.

Index

F

G

H

I

O

P

R

S