ApolloSDK Integration Module Settings Guide

ACFA PSIM 1.1

Last update 09/13/2023

Table of Contents

1 List of Terms used in the ApolloSDK Integration Module Settings Guide

Server—a computer configured as an *Axxon PSIM* Software System **Server**.

Access control system (ACS)—a system of hardware and software designed to monitor and control access.

Controller—an electronic device designed to monitor and control access points.

Access point—a place where access is controlled.

AIM Interface Module—an interface module for connecting readers or keypads to an AAN controller.

AIO alarm panel—a microprocessor-based alarm panel that monitors the state of alarm inputs and manages relay outputs.

Alarm input—an input to switch alarm sensors (magnetic-contact, impact-contact) or the output circuits of IR sensors and alarm panels.

Readers—electronic devices designed for entering a memorized code using a keypad or reading encoded data from system keys (identifiers).

Access card—a physical access key accepted by a reader.

Access time—the time allotted to pass through an access point. When the allotted time has passed the access point locks automatically.

Impulse—a signal used to close a relay.

Time schedule—a set of any number of time intervals during a day (24 hours) defined for several days (1 to 366), and the time intervals during specific dates. Time schedule defines a schedule of access to the secured object.

2 ApolloSDK Integration Module Settings Guide. Introduction

On the page:

2.1 Purpose of the document

This *ApolloSDK Integration Module Settings Guide* is a reference manual designed for *ApolloSDK* integration module configuration technicians and operators. This module functions as part of *ACFA PSIM*.

This Guide presents the following materials:

1. General information about the *ApolloSDK* integration module.
2. Configuration of the *ApolloSDK* integration module.
3. Working with the *ApolloSDK* integration module.

2.2 General information about the *ApolloSDK* integration module

The *ApolloSDK* integration module is part of *ACFA PSIM*. It is designed to configure and control *ApolloSDK* hardware.

The *ApolloSDK* integration module allows working with *ApolloSDK* system built using *AAN* central controllers and without them.

The following hardware is integrated with *ACFA PSIM*:

1. *AAN* central controllers (an *ACS* component).
2. *AIM* interface modules (an *ACS* component).
3. *AIO* alarm panels (an *SFA* component).

Note

It is possible to work with *AIO* alarm panels only using the *AAN* central controller.

AAN central controllers are connected via COM-port or Ethernet-connection.

Connection via COM-port can be established in case of using one interface module without *AAN* central controller. Ethernet-connection with converter is used in case of several *AIM* interface modules.

Note

Detailed information about the *ApolloSDK* system can be found in the official documentation (manufacturer AAM Systems).

Attention!

The *ApolloSDK* integration module requires *ApolloSDK* system software installed on the Server. *ACFA PSIM* supports *Apollo SDK v.4.1.1* with the ability to work via RDP.

Before configuring the *ApolloSDK* integration module, the following actions must be performed:

1. Install the required hardware on the site.
2. Install the *ApolloSDK* software on the Server (is located in the <Directory of the *Axxon PSIM* software installation>\Modules\ApolloSDK).
3. Copy the license.bin file (<Directory of the *Axxon PSIM* software installation>\Modules\ApolloSDK\ApolloSDK v.2.1 (01.15.2015) Installer\License) to directory of the *ApolloSDK* software installation.

3 Supported hardware and licensing of the Apollo SDK integration module

Manufacturer	AAM Systems Office address: Kraznokazarmennaya str., 13-402 Moscow Post address: Kraznokazarmennaya str., 14 Moscow, 111250 Tel: +7 (495) 924-2227 Fax: +7 (495) 362-7262 Email: aam@aamsystems.ru www.aamsystems.ru
Integration type	SDK
Hardware connection	RS-232, IP

Supported hardware

Hardware	Function
AAN-100	Network controller
AAN-32S	Network controller
AAN-32N	Network controller
AIM-2SL	Access controller
AIM-4SL	Access controller
AIO-168	Security panel
ASA-72	Status panel

Licensing

There are four positions in the price-list for one module:

Integration with *Apollo* (one server)
Integration with *Apollo* (one reader)
Integration with *Apollo* AIO-168
Integration with *Apollo* (one ASA-72)

Integration with Apollo (one server) is an electronic guardant key protecting *Apollo SDK* from the manufacturer site and storing serial keys of all devices. There is at least one key for system.

If hardware is connected to several servers with *Axxon PSIM* coreб then additional purchase of electronic protect keys is required for each second and next servers (Integration with *Apollo* (one reader). Own configuration for each electronic key—in accordance with settings in the objects tree of *Axxon PSIM*.

It is required to present all serial numbers of hardware at the time of module order.

Licensing doesn't depend on number of connected senior panels (AAN-100 and/or AAN-32). Licensing depends only from number of connected terminals (readers, sensor/relay, ASA-72). It is connected with the specifics of module sublicensing in company-manufacturer of module.

Apollo SDK software module works with AIM-*SL controllers directly. Also there is possibility to work with several AIM-*SL controllers using network converter ENI-110.

4 Configuration of the ApolloSDK integration module

4.1 Configuration procedure for the ApolloSDK integration module

The *ApolloSDK* integration module with an AAN controller is configured as follows:

1. Configure the connection of AAN controller.
2. Configure an AAN controller.
3. Configure connection port of AIM and AIO interface modules.
4. Configure AIM and AIO modules.
5. Configure readers of AIM module.
6. Configure executive devices of AIO module.
7. Configuring visitor groups and group lists.
8. Configuring APB Area.
9. Configuring internal variables in ApolloSDK.
10. Configure supporting of access card formats.
11. Configuring antipassback.

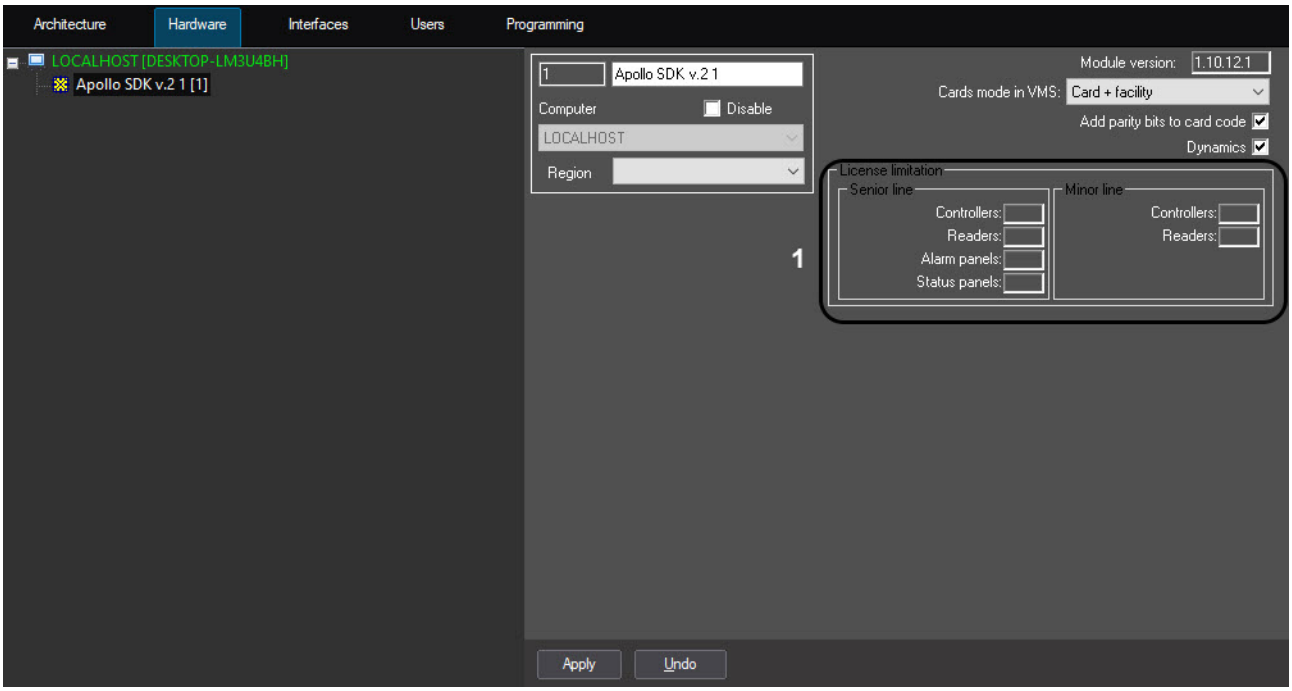
The *ApolloSDK* integration module without an AAN controller is configured as follows:

1. Configure connection of AIM SC controller.
2. Configure AIM SC controller.
3. Configure AIM SC reader.
4. Configure supporting of access card formats.

It's required to send configuration to hardware if the system configuration was changed.

4.2 Activation of the ApolloSDK integration module

To activate the *ApolloSDK* integration module, create the **Apollo SDK v.2** object on the basis of the **Computer** object.

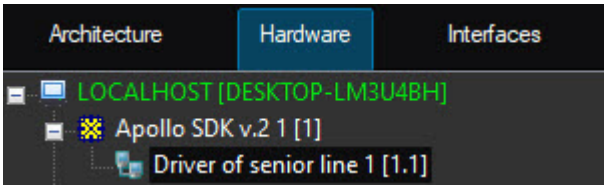


License limitation for number of devices is specified on the settings panel of the **Apollo SDK v.2** object (**1**).

4.3 Configure system with AAN central controller

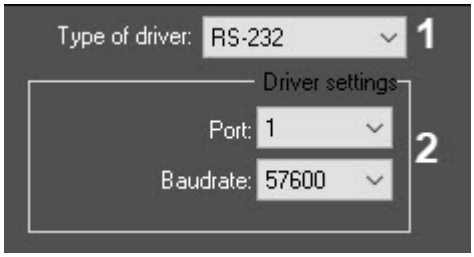
4.3.1 Configure connection of AAN controller

Connection of the *AAN* controller is configured on the settings panel of the **Driver of senior line** object created on the basis of the **Apollo SDK v.2** object.

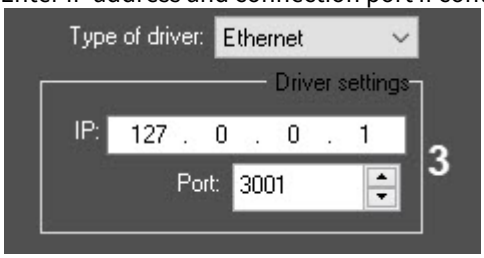


To connect the *AAN* controller, do the following:

1. Select the interface of controller connection—COM port or Ethernet (**1**).



2. Select the port number and its baudrate if controller is connected via COM port (**2**).
3. Enter IP address and connection port if controller is connected via Ethernet (**3**).



4. Click the **Apply** button.

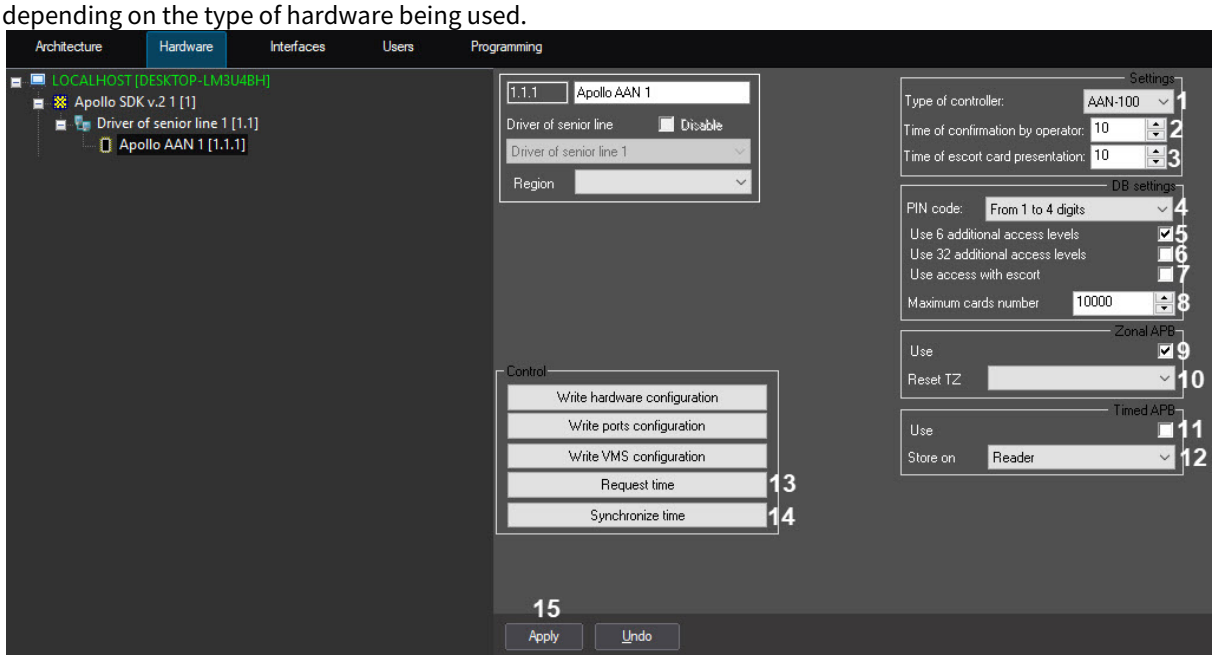
AAN controller is now connected.

4.3.2 Configure AAN controller

The *AAN* controller is configured on the settings panel of the **Apollo AAN** object created on the basis of the **Driver of senior line** object.

The *AAN* controller is configured as follows:

1. From the **Type of controller** drop-down list (**1**), select the type of *AAN* controller (*AAN-32* or *AAN-100*) depending on the type of hardware being used.



2. In the **Time of confirmation by operator** field (**2**), enter the time period in seconds for operator to make a decision to grant or deny access.
3. In the **Time of escort card presentation** field (**3**), enter the time period in seconds between the presentation of the first and second access cards which, if exceeded, will result in access not being granted.
4. From the **PIN code** drop-down list (**4**), select the length of PIN code being used. Select the **Do not use** value if it's not required to use PIN code.
5. Set the corresponding checkboxes if it's required to use additional 6 (**5**) or 32 (**6**) access levels.

> **Attention!**
>
> The maximum number of users that can be stored in controller memory decreases if the additional access levels are used.

6. Set the **Use access with escort** checkbox (**7**) if it is necessary to confirm the access to the zone with the escort card.
7. In the **Maximum cards number** field (**8**), enter the maximum number of access cards that will be stored in the controller's memory.

> **Note**
>
> The maximum number of access cards that can be stored in the controller's memory depends on the number of memory cards installed on it.

8. Configure the antipassback control:
   a. If it is necessary to activate the zonal antipassback control, set the **Use** checkbox (**9**) in the **Zonal APB** group (for details, see Configuring antipassback).
   b. If necessary, from the **Reset TZ** drop-down list (**10**), select the time zone by which the antipassback control flags will be reset for all users.
   c. If it is necessary to activate the timed antipassback control, set the **Use** checkbox (**11**) in the **Timed APB** group (for details, see Configuring antipassback).
   d. From the **Store on** drop-down list (**12**), select the location for storing the antipassback control flags: **Reader** or **APB zone**.

> **Note**
>
> - It is possible to activate either zonal or timed antipassback control. Simultaneous operation of these antipassback control modes is not allowed.
> - To enable the timed APB by zones, set the **Use** checkbox in the **Timed APB** group (step 8c) and select the **APB zone** as the location for storing the flags (step 8d). The **Use** checkbox in the **Zonal APB** group must be clear.

9. Click the **Request time** button (**13**) to request the *AAN* controller time.
10. Click the **Synchronize time** button (**14**) to synchronize the *AAN* controller time with the computer time.
11. Click the **Apply** button (**15**).

The *AAN* controller is now configured.

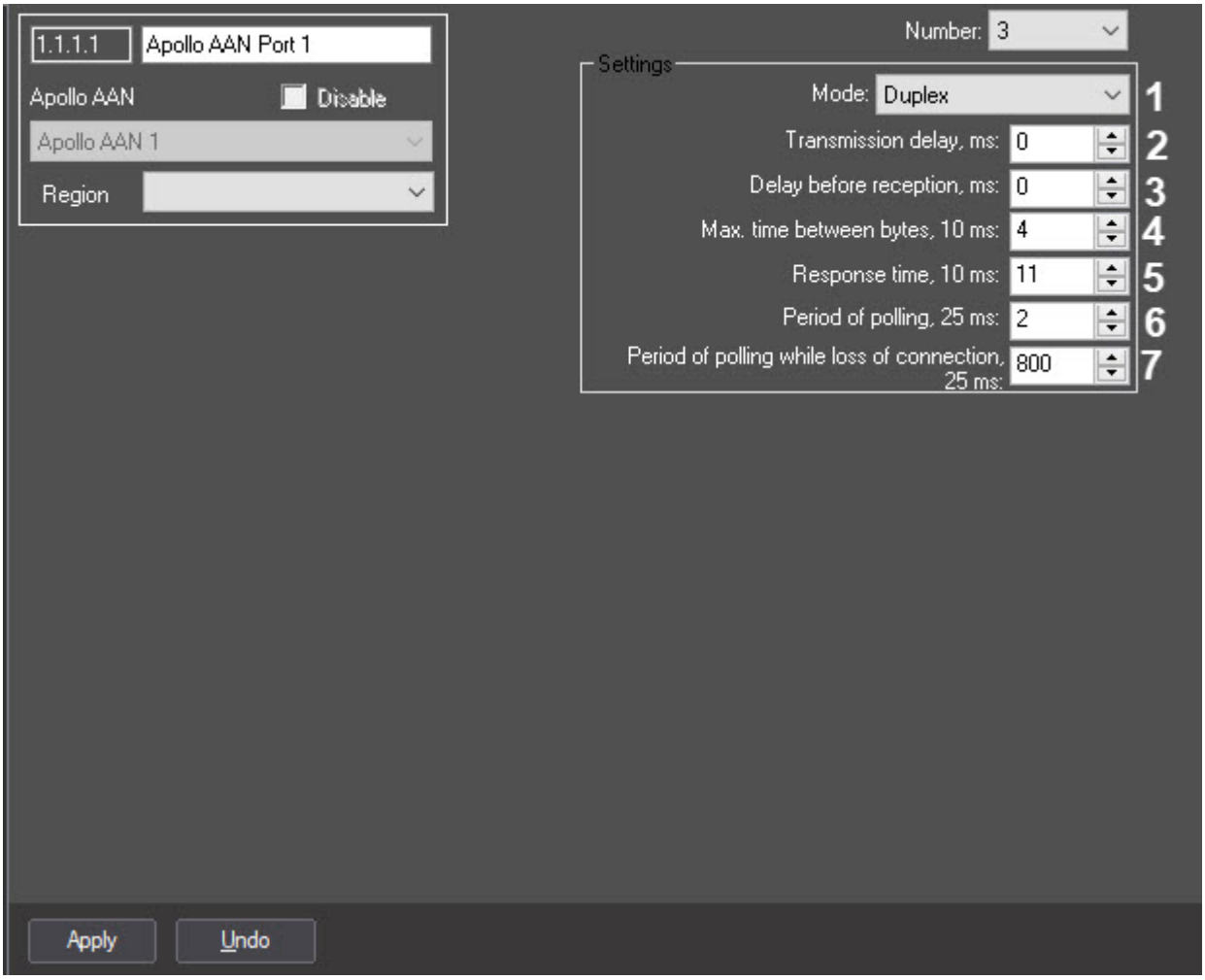
4.3.3 Configure connection ports of AIM and AIO interface modules

AIM and *AIO* modules are connected to the *AAN* controller via 4 ports (RS-485 or Ethernet).

Port is configured on the settings panel of the **Apollo AAN Port** object created on the basis of the **Apollo AAN** object.

To configure port of the *AAN* controller, do the following:

> **Attention!**
>
> It is not recommended to change default port parameters.



1. From the **Mode** drop-down list (**1**), select the mode of data exchange: duplex or half duplex. Duplex mode receives and sends data simultaneously. Receiving and sending data in half duplex mode are performed in interval.
2. Specify delay of data exchange via port in milliseconds (**2**).
3. Specify delay before data reception via port in milliseconds (**3**).
4. Specify maximum time of waiting for next byte in milliseconds (**4**).

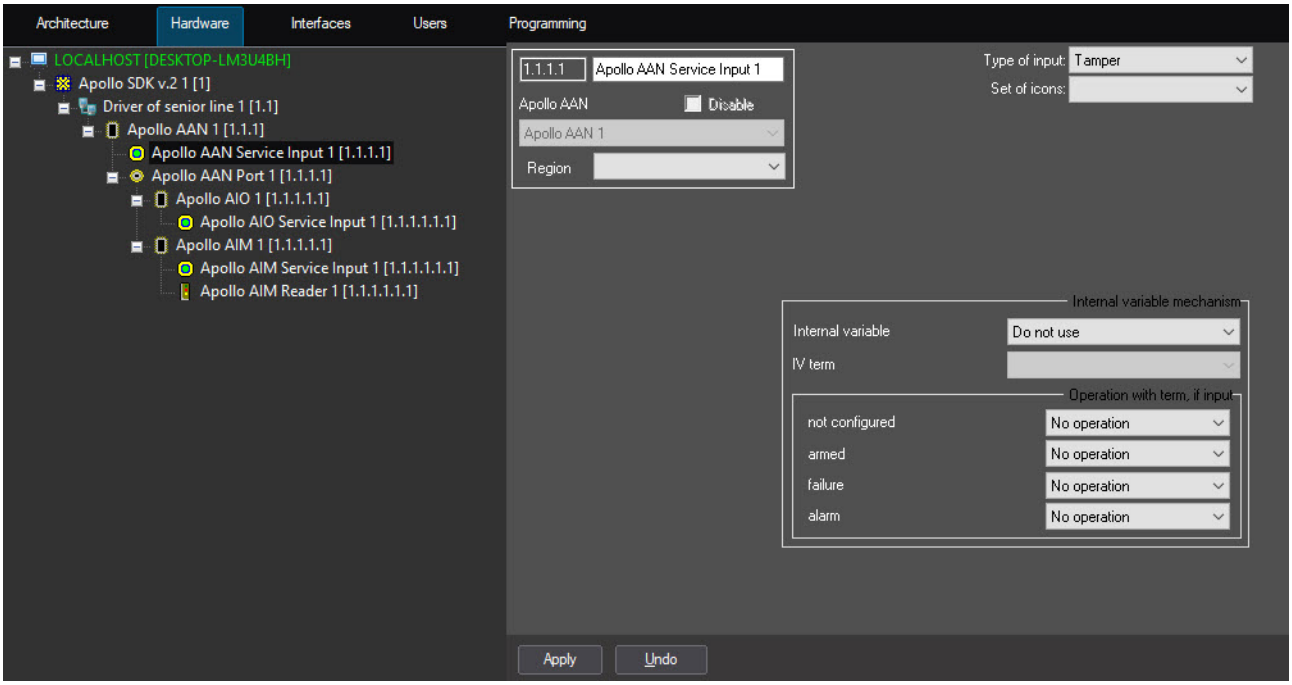
   > **Note**
   >
   > Names of this and next parameter contain multiplicity of value which is to be specified. So, if it's required to specify value of the Max. time between bytes, 10ms equal to 4, then maximum time of waiting for the next byte will be 40 ms.

5. Specify maximum time of response from devices connected via this port (**5**) considering the specified multiplicity. If device does not response during the specified time, connection with it will be lost.
6. Specify period of polling connected devices with established connection with them in miliseconds, considering the specified multiplicity (**6**).
7. Specify period of polling connected devices without connection with them in milliseconds, considering the specified multiplicity (**7**).
8. Click the **Apply** button.

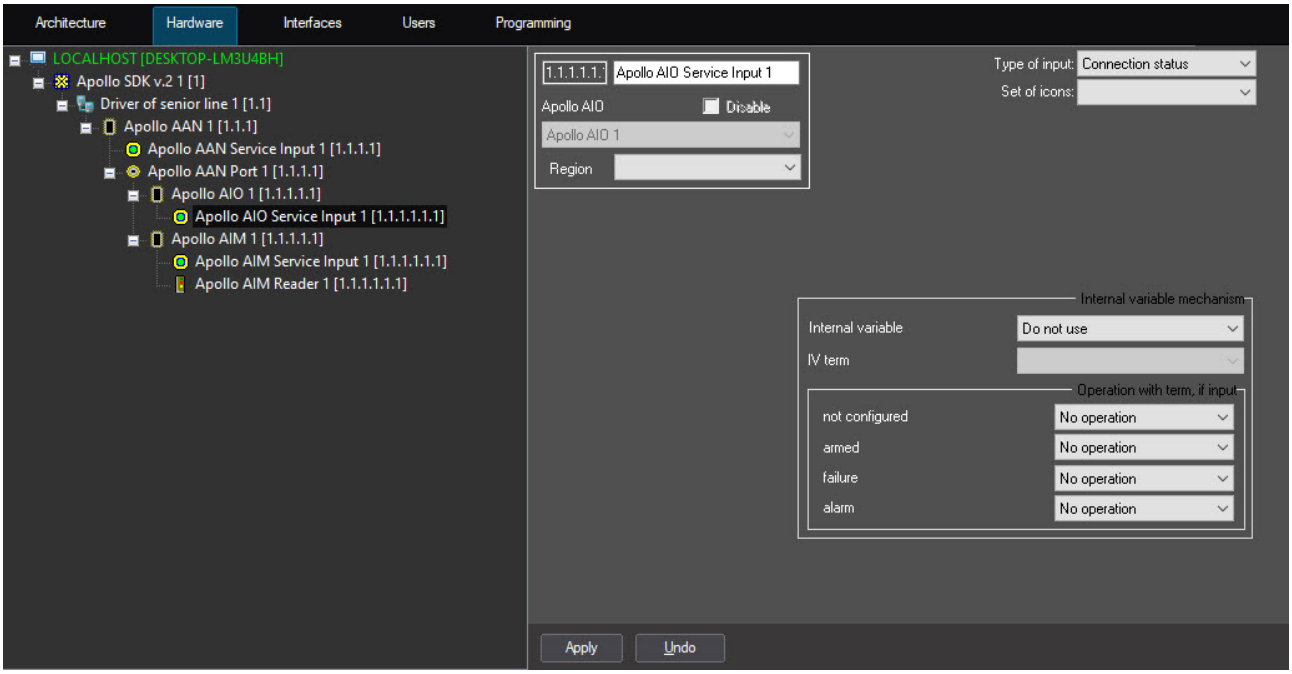
Port of the *AAN* controller is now configured.

4.3.4 Configure AAN, AIO or AIM Service input

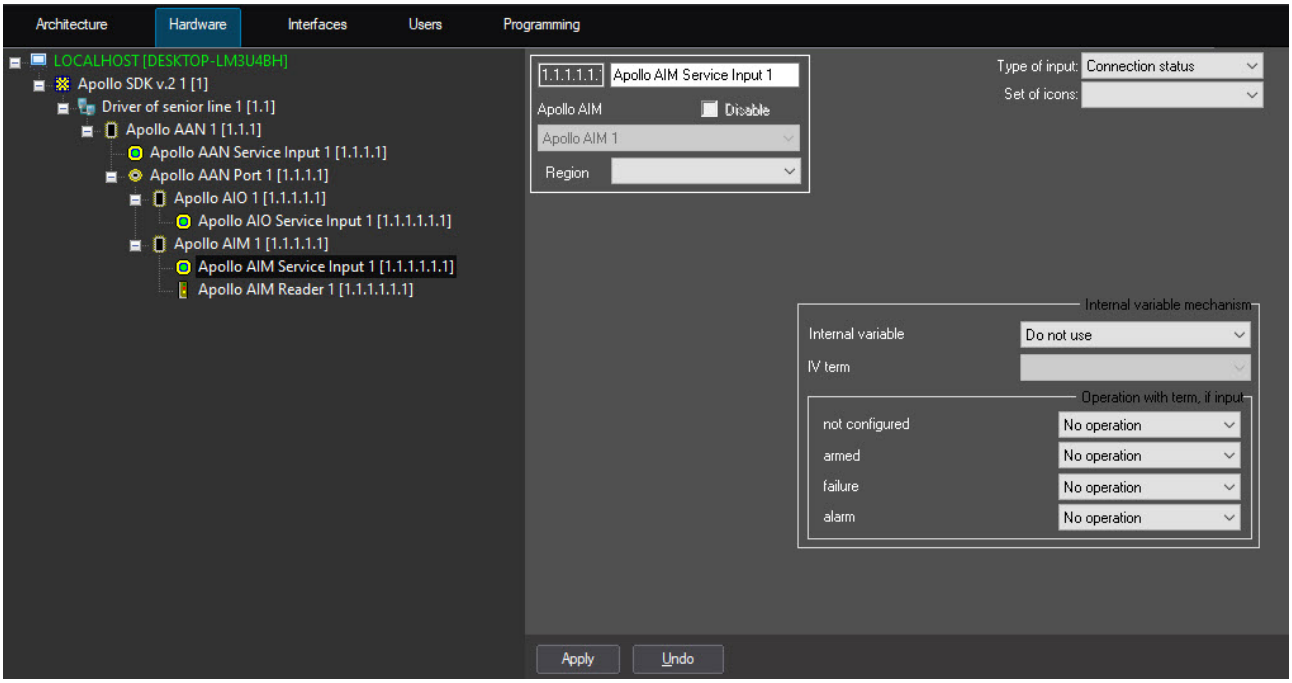
The *AAN* Service input is configured on the settings panel of the **Apollo AAN Service Input** object created on the basis of the **Apollo AAN** object on the **Hardware** tab of the **System settings** dialog box.



The *AIO* Service input is configured on the settings panel of the **Apollo AIO Service Input** object created on the basis of the **Apollo AIO** object on the **Hardware** tab of the **System settings** dialog box.



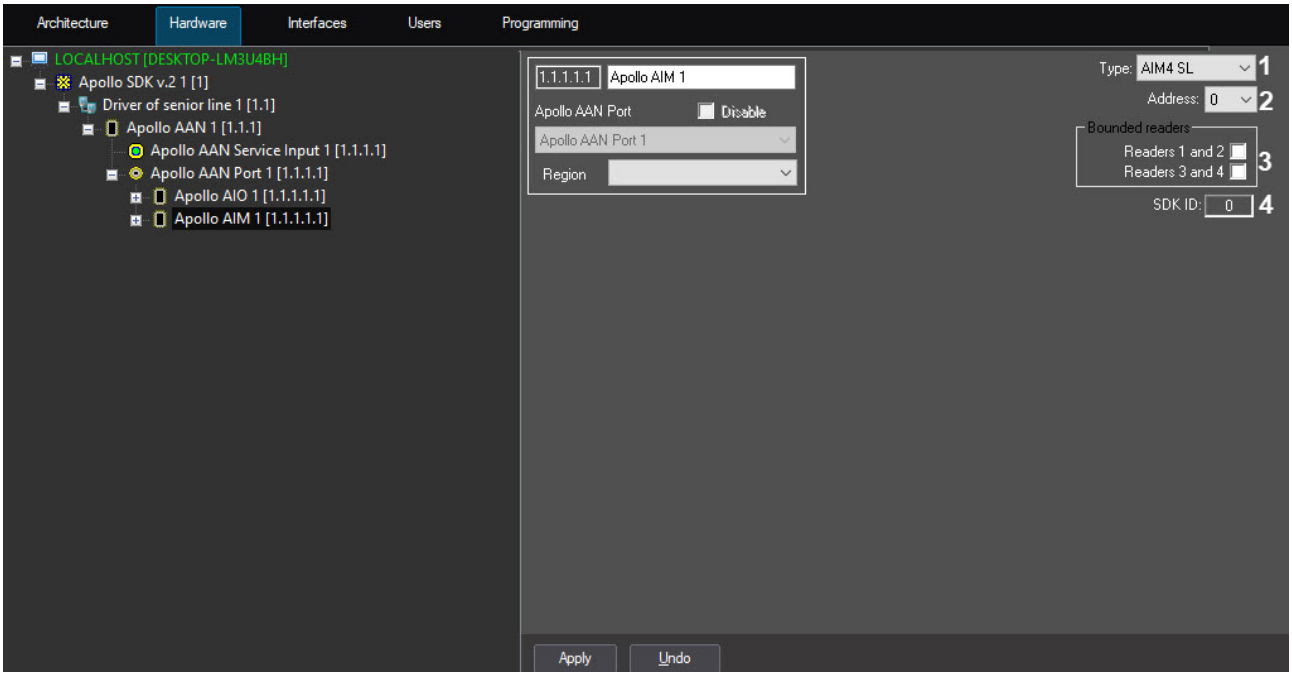
The *AIM* Service input is configured on the settings panel of the **Apollo AIM Service Input** object created on the basis of the **Apollo AIM** object on the **Hardware** tab of the **System settings** dialog box.



Please refer to Associating AAN, AIO or AIM Service Input with terms for details on configuration process for these inputs.

4.3.5 Configure the AIM interface module

The *AIM* interface module is configured on the settings panel of the **Apollo AIM** object created on the basis of the **Apollo AAN Port** object.



To configure the *AIM* interface module, do the following:

1. Select the type of the AIM module from the corresponding list (**1**).

2. From the **Address** drop-down list (**2**), select address of module in internal network.
3. Set the corresponding checkboxes, if bounded readers are used (**3**).
   Bounded readers are used from both sides of door. Single readers are used from one side of door, access from another side is performed by button.

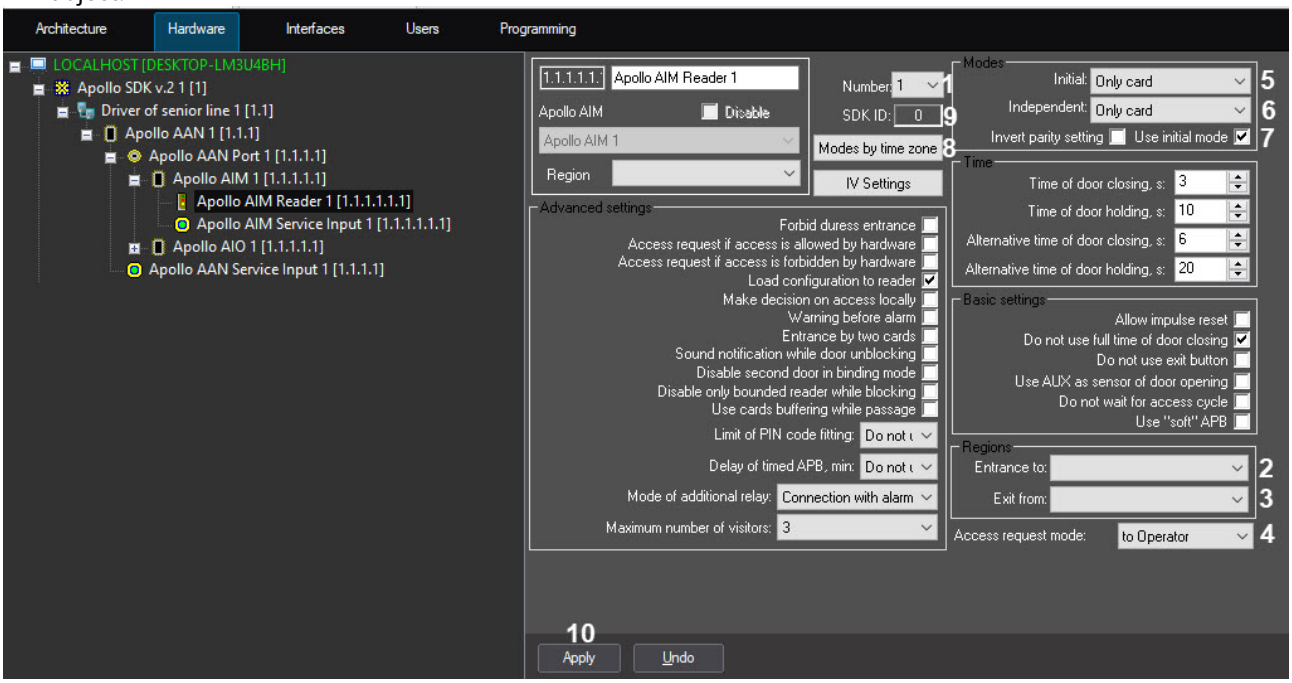
   > **Note**
   >
   > The **SDK ID** field (**4**) is filled in automatically when object is created, and it contains different values for objects of the same type. It's not recommended to create objects by template (see The Save function) for the correct operation of module because of equal **SDK ID** values.

4. Click the **Apply** button.

The *AIM* interface module is now configured.

4.3.6 Configure AIM Readers

AIM reader is configured on the settings panel of the **Apollo AIM Reader** object created on the basis of the **Apollo AIM** object.



To configure *AIM* module, do the following:

1. From the **Number** drop-down list, select the reader address (**1**).
2. From the **Entrance to** drop-down list, select the **Region** object corresponding to the area on the side of exit through this reader (**2**).
3. From the **Exit from** drop-down list, select the **Region** object corresponding to the area on the side of entrance through this reader (**3**).
4. From the **Access request mode** drop-down list, select responsible part for decision of access: the *Axxon PSIM* Server (automatic decision based on user access level and its card) or Operator (**4**).

   > **Note**

For processing request by operator it's required to create the *Event Manager* interface object and configure it for the Operator request (Access granted) event. For detailed information about this object and its functionality see Event Manager Module Settings and Operation Guide.

5. Configure operation modes of reader:
   a. From the **Initial** drop-down list, select the mode of reader operation while connection establishing (**5**).

Operation mode	Description
Closed	Access is closed for all
Card only	Access granted by access card
PIN or card	Access granted by access card or PIN code
Card and PIN	Access granted by access card and PIN code
Opened	Access is opened for all
Facility code	Access is granted by facility code

   b. From the **Independent** drop-down list, select the mode of reader operation while losing of connection (**6**).
   c. Set the **Use initial mode** checkbox, if it's required to switch to initial mode after writing of configuration (**7**).
   d. Click the **Modes by time zones** button and configure changing of reader operation mode depending on time zone (**8**). Two operation modes are selected for each time zone: at the start of time zone and after end of time zone.

Note

The **SDK ID** (**9**) field is filled in automatically while object creating and it contains different values for objects of the same type. It's not recommended to create objects by template (see The Save function) for correct working of module because of equal **SDK ID** values.

6. Specify other parameters of reader.

Parameter	Parameter setting method	Description
Time of door closing, s	Enter the value in the field	Time of opening in seconds
Time of door holding, s	Enter the value in the field	**Time period** in seconds during which door is to be closed. Otherwise, the **Door holding** message will be triggered
Alternative time of door closing, s	Enter the value in the field	Alternative time of opening in seconds. It is in use by special command (from card, using macro or script)

Parameter	Parameter setting method	Description
Alternative time of door holding, s	Enter the value in the field	**Alternative time period** in seconds during which door is to be closed. Otherwise, the **Door holding** message will be triggered. It is in use by special command (from card, using macro or script)
Allow impulse reset	Set the checkbox	Yes—impulse reset enabled
Do not use full time of door closing	Set the checkbox	Yes—lock after door closing
Do not use exit button	Set the checkbox	Yes—disable exit button
Use AUX as sensor of door opening	Set the checkbox	Yes—use additional *AUX* sensor of *AIM* module as sensor for door opening
Do not wait for access cycle	Set the checkbox	Yes—passage is performed after making an access decision No—passage is performed after triggering a sensor of door opening
Use "soft" APB	Set the checkbox	Yes—antipassback is available, but the corresponding mistake will be specified in message
Forbid duress entrance	Set the checkbox	Yes—door is locked while entering "duress" PIN-code No—door is opened and alarm message is triggered while entering "duress" PIN-code
Access request if access is allowed by hardware	Set the checkbox	Yes—send access request to operator if *AIM* or *AAN* allow access
Access request if access is forbidden by hardware	Set the checkbox	Yes—send access request to operator if *AIM* or *AAN* forbid access
Load configuration to reader	Set the checkbox	Yes—duplicate users and their access levels to the memory of *AIM* module while writing of configuration to the *AAN* controller
Make decision on access locally	Set the checkbox	Yes—the *AIM* module makes an access decision without the *AAN* central controller
Warning before alarm	Set the checkbox	Yes—trigger warning before alarm

Parameter	Parameter setting method	Description
Entrance by two cards	Set the checkbox	Yes—passage through the reader is performed by two cards
Sound notification while door unblocking	Set the checkbox	Yes—sound signal of reader while unblocking the door
Disable second door in binding mode	Set the checkbox	Yes—block the reader working for exit from secured areas
Disable only bounded reader while blocking	Set the checkbox	Yes—only bounded reader are blocked while locking signal No—all readers are blocked
Use cards buffering while passage	Set the checkbox	Yes—write access cards to the *AIM* while passage
Limit of PIN code fitting	Select the value from the list	Lock is blocked while exceeding of invalid PIN-code entrances
Delay of timed APB, min	Select the value from the list	Time interval in minutes during which antipassback is forbidden
Mode of additional relay	Select the value from the list	**Connection with alarms**—additional relay is triggered while alarm **Manual control**—additional relay is activated manually
IV Settings	Click the button	See Associating AIM reader commands with terms

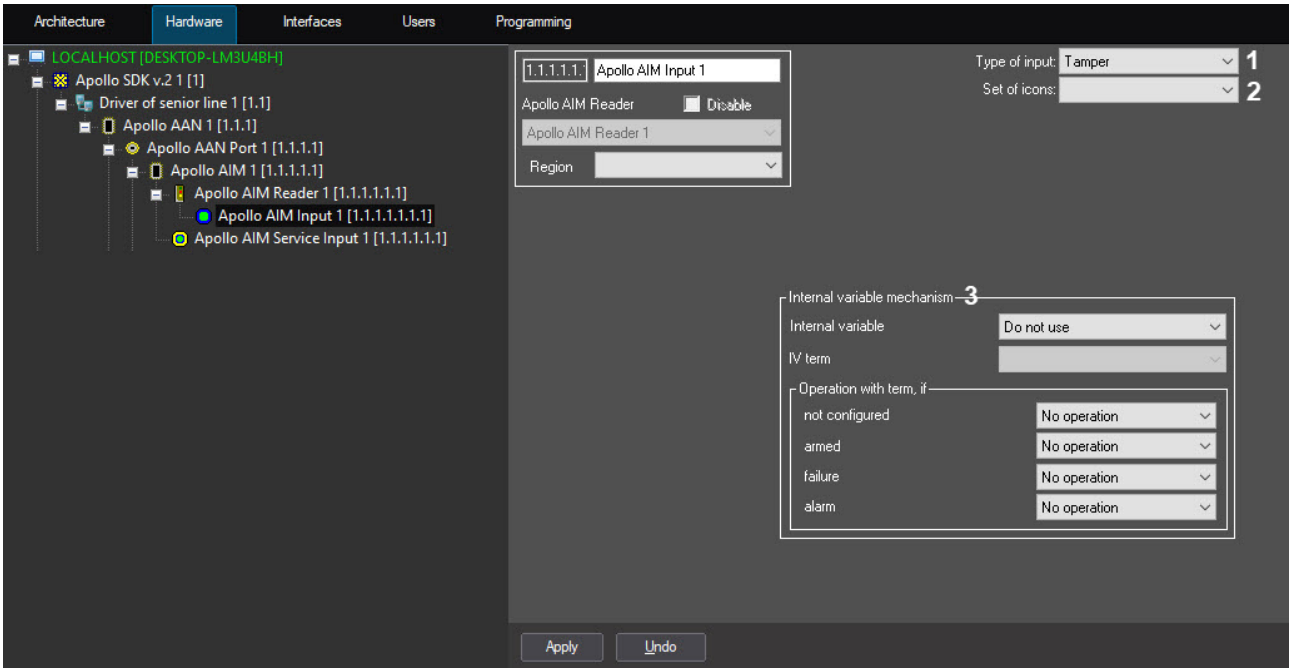
7. Click the **Apply** (**10**) button.

The *AIM* reader is now configured.

Configure input of AIM Reader

It's possible to create and configure inputs of *AIM* readers. The inputs track the defined states of reader and trigger messages on which different reactions can be configured using scripts, macros and internal variables.

Input of *AIM* Reader us configured on the settings panel of the **Apollo AIM Input** object created on the basis of the **Apollo AIM Reader** object.



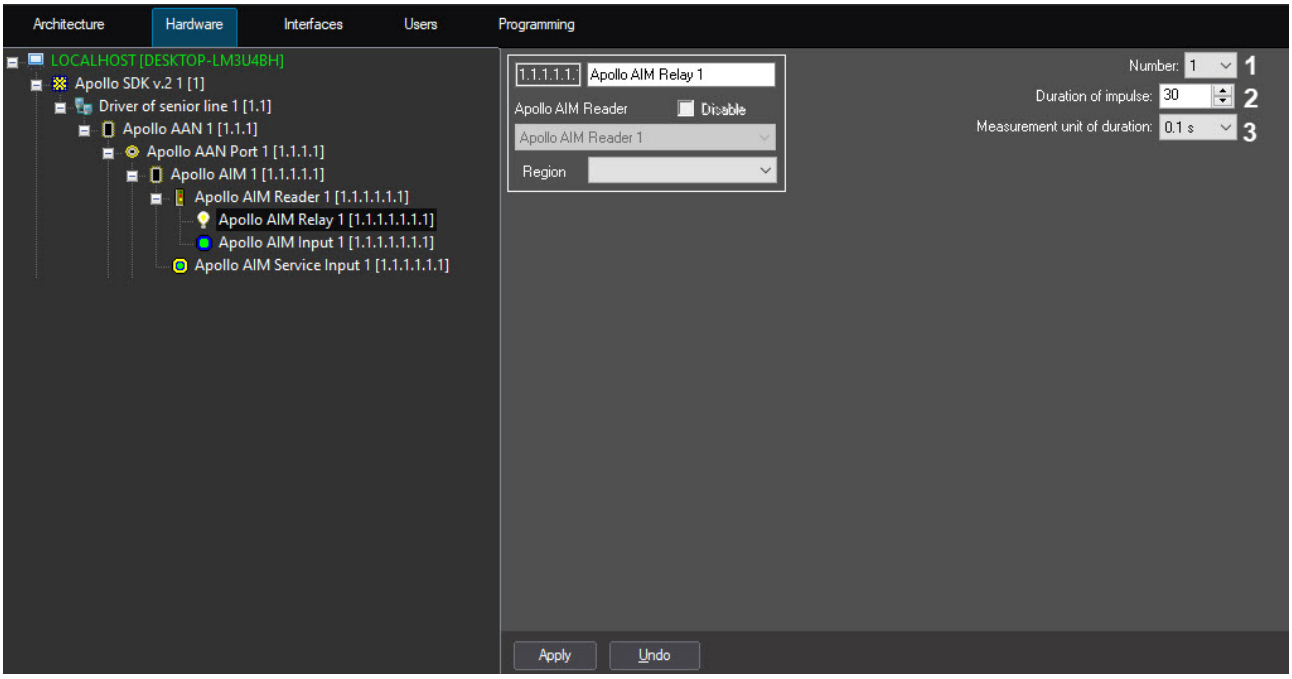
Virtual inputs are configured as follows:

1. Select an object corresponding to the event from the **Type of input** drop-down list (**1**).
2. Select the set of icons for virtual input on the **Map** from the **Set of icons** drop-down list (**2**).
3. If necessary, associate the input with a term of internal variable (**3**, see Associating AIM input with terms).
4. Click the **Apply** button.

Virtual input of *AIM* Reader is now configured.

Configure relay of AIM Reader

Relay of *AIM* Reader is configured on the settings panel of the **Apollo AIM Relay** object created on the basis of the **Apollo AIM Reader** object.



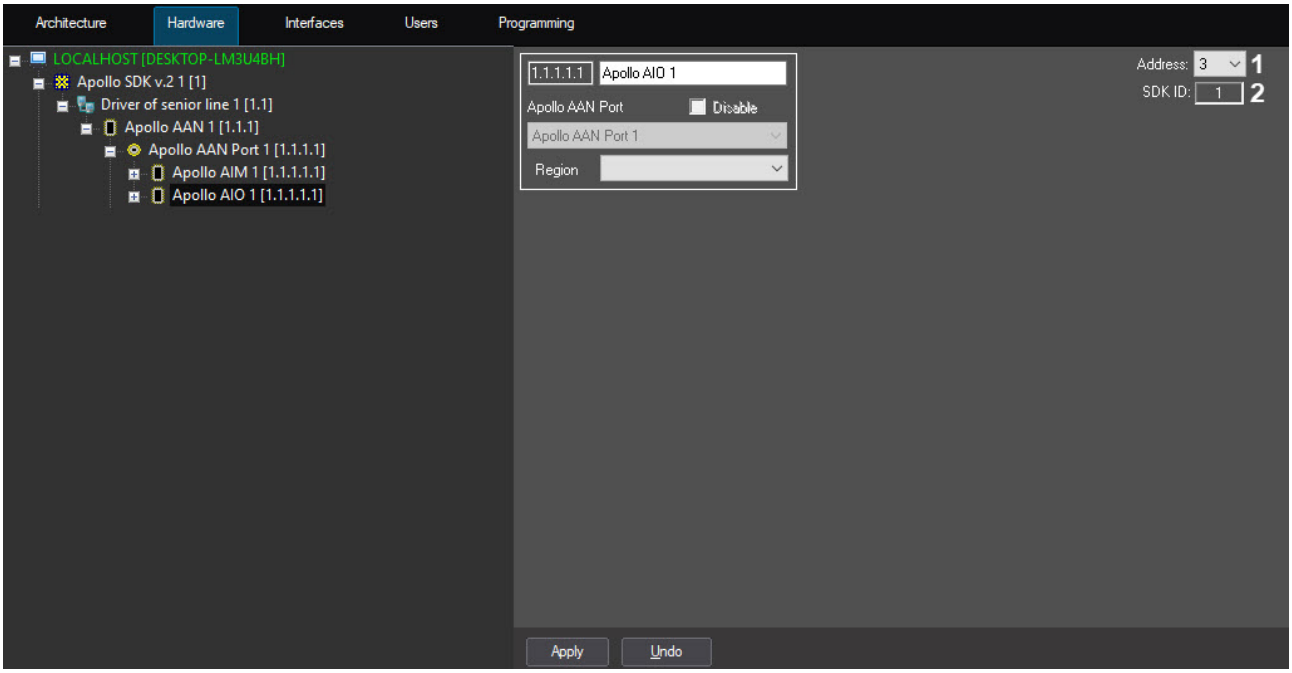
Relays are configured as follows:

1. From the **Number** drop-down list, select the relay address (**1**).
2. Specify duration of relay impulse:
    a. In the **Duration of impulse** field, set the value in selected units defining the impulse duration (**2**).
    b. From the **Measurement unit of duration** drop-down list, select the measurement unit of relay impulse (**3**).
3. Click the **Apply** button.

Relay of *AIM* Reader is now configured.

4.3.7 Configure the AIO interface module

The *AIO* interface module is configured on the settings panel of the **Apollo AIO** object created on the basis of the **Apollo AAN Port** object.



To configure the *AIO* interface module, do the following:

1. From the **Address** drop-down list, select address of module in internal network (**1**).

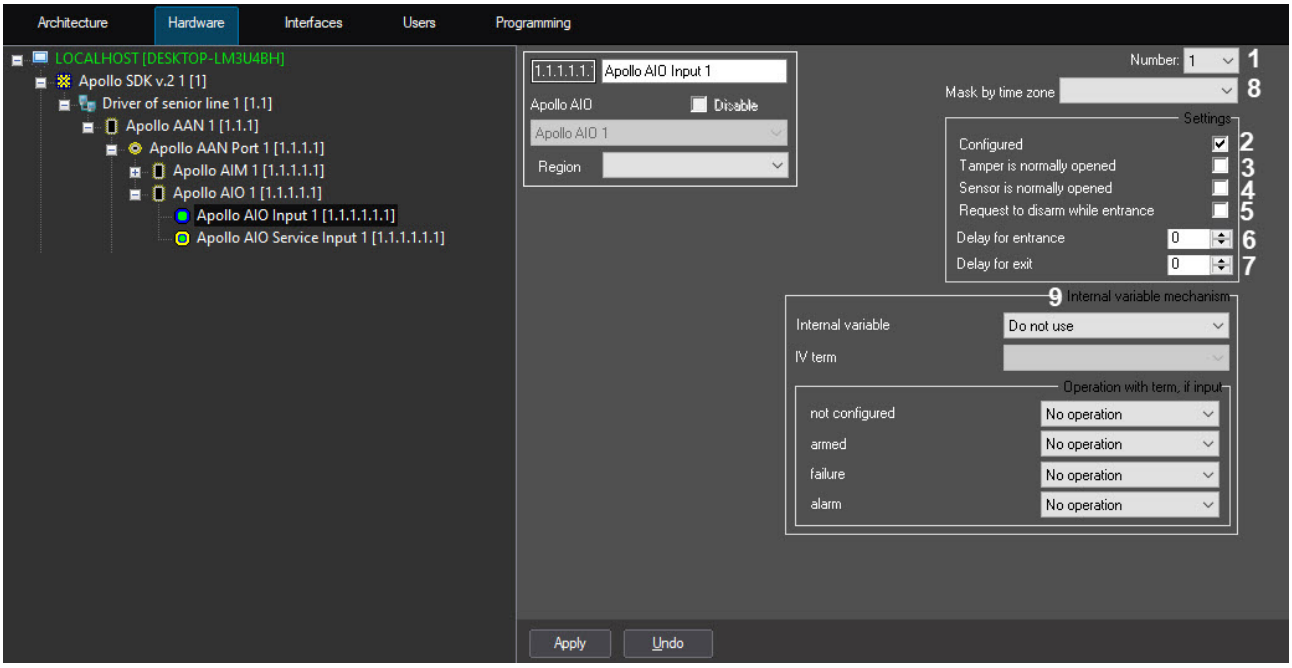
   > **Note**
   >
   > The **SDK ID** (**2**) field is filled in automatically when object is created, and it contains different values for objects of the same type. It's not recommended to create objects by template (see The Save function) for correct working of module because of equal **SDK ID** values.

2. Click the **Apply** button.

The *AIO* interface module is now configured.

Configure AIO inputs

The *AIO* input is configured on the settings panel of the **Apollo AIO Input** object, which is created on the basis of the **Apollo AIO** object.



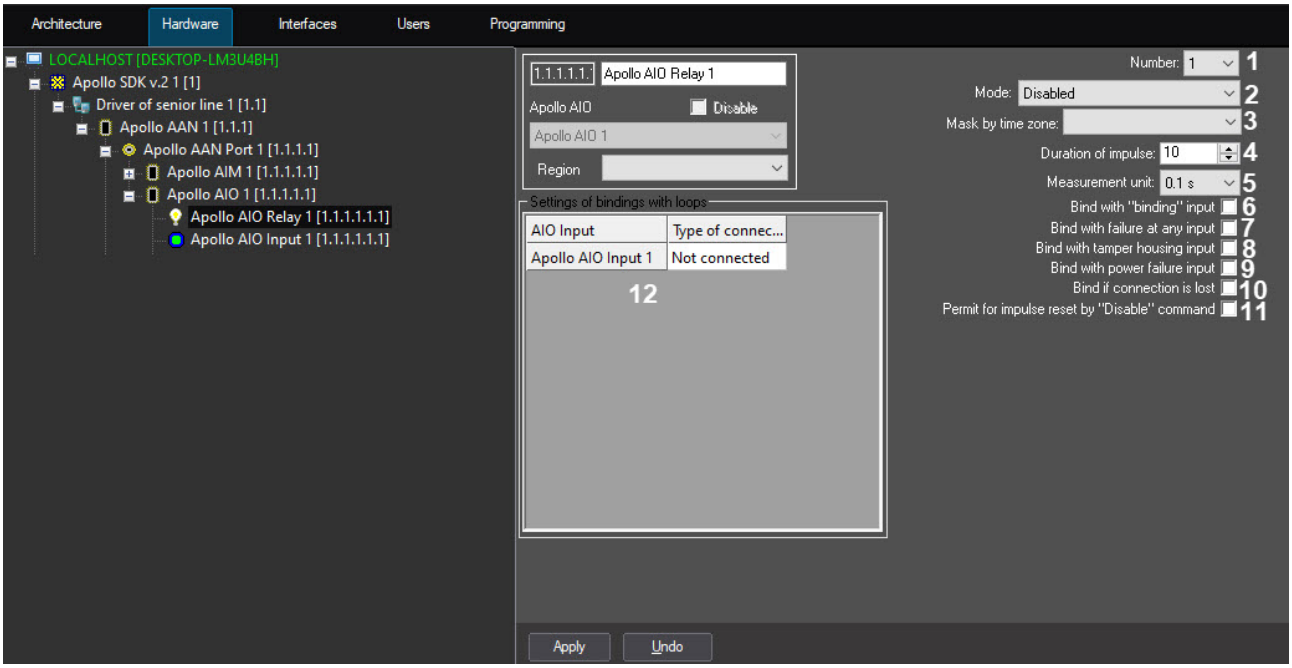
The *AIO* input is configured as follows:

1. Select the input address from the **Number** drop-down list (**1**).
2. Set the **Configured** checkbox if the input is in working state (**2**).
3. Set the **Tamper is normally opened** checkbox if it's required to receive alarm events while breaking or opening the sensor housing (**3**).
4. Set the **Sensor is normally opened** checkbox if it's required that input is to be in normal state (not alarm) while opened contacts (**4**).
5. To require disarming of input, set the corresponding checkbox (**5**).
6. In the **Delay for entrance** field, enter the value in seconds defining time for input disarming (**6**).
7. In the **Delay for exit** field, enter the value in seconds defining time period during which the object can exit after the input arming (**7**).
8. From the **Mask by time zone** drop-down list, select the time zone during which events won't be generated for the input (**8**).
9. If necessary, assign the input with a term of an internal variable (**9**, see Associating AIO input with a term).
10. Click the **Apply** button.

The *AIO* input is now configured.

Configure the AIO relay

The *AIO* relay is configured on the settings panel of the **Apollo AIO Relay** object created on the basis of the **Apollo AIO** object.



The *AIO* relay is configured as follows:

1. From the **Number** drop-down list, select the number of relay (**1**).
2. From the **Mode** drop-down list, select the mode of relay working (**2**).

Mode of relay working	Description
Disabled	Relay opened
Enabled	Relay closed
Connected with inputs locally	State of relay depends on state of loops and inputs

3. From the **Mask by time zone** drop-down list, select the time zone during which events won't be generated for the relay (**3**).
4. Specify duration of relay impulse:
   a. From the **Measurement unit** drop-down list, select the measurement unit of relay impulse (**5**).
   b. In the **Duration of impulse** field, set the value in selected units defining the impulse duration (**4**).
5. Set the **Bind with "binding" input** checkbox if it's required that relay reacts to the **Binding** state of input (**6**).
6. Set the **Bind with failure at any input** checkbox if it's required that relay reacts to failure of any input (**7**).
7. Set the **Bind with tamper housing input** checkbox if it's required that relay reacts to state of security panel state (**8**).
8. Set the **Bind with power failure input** checkbox if it's required that relay reacts to state of security panel power (**9**).
9. Set the **Bind if connection is lost** checkbox if it's required that relay reacts to loss of connection (**10**).
10. If it's required to enable possibility of impulse reset when changing the operation mode to **Disabled**, set the **Permit for impulse reset by "Disable" command** checkbox (**11**).
11. Configure interaction of relay and inputs (**12**). In the **Type of connection** column select state of input at which relay will be closed.

State	Description
Not connected	Relay is always opened at any state of input
Alarm and tamper	Relay is triggered in one of the following ways: a. Alarm message from input was received. b. Message from tamper was received
Alarm/masked and tamper	Relay is triggered in one of the following ways: a. Alarm message from input was received. b. Alarm message from disarmed input was received. c. Message from tamper was received
Alarm/masked, tamper and failure	Relay is triggered in one of the following ways: a. Alarm message from input was received. b. Alarm message from disarmed input was received. c. Message from tamper was received. d. Message about failure was received

12. Click the **Apply** button.

The *AIO* relay is now configured.

4.3.8 Configuring visitor groups and group lists

The *ApolloSDK* module allows configuring visitor groups and visitor group lists. These mechanisms are intended to deny access through Apollo readers for visitors without escort.

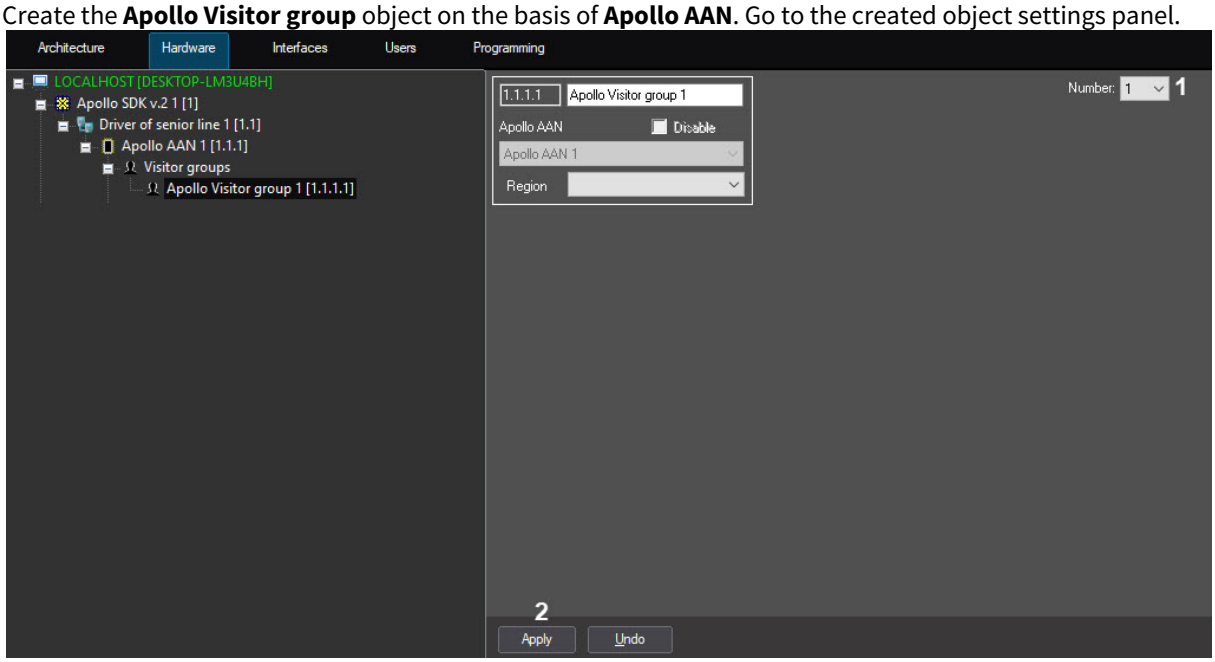
Configuration of visitor groups and visitor group lists is performed on the settings panel of corresponding objects created based on the **Apollo AAN** object on the **Hardware** tab of the **System settings** dialog box (see Configuring visitor groups and Configuring visitor group lists).

The enrollment of visitors to the group and assigning escorts for group lists is carried out using the *Access Manager* module (see Additional user settings in Apollo SDK).

Configuring visitor groups

Visitor groups are configured as follows:

1. Create the **Apollo Visitor group** object on the basis of **Apollo AAN**. Go to the created object settings panel.

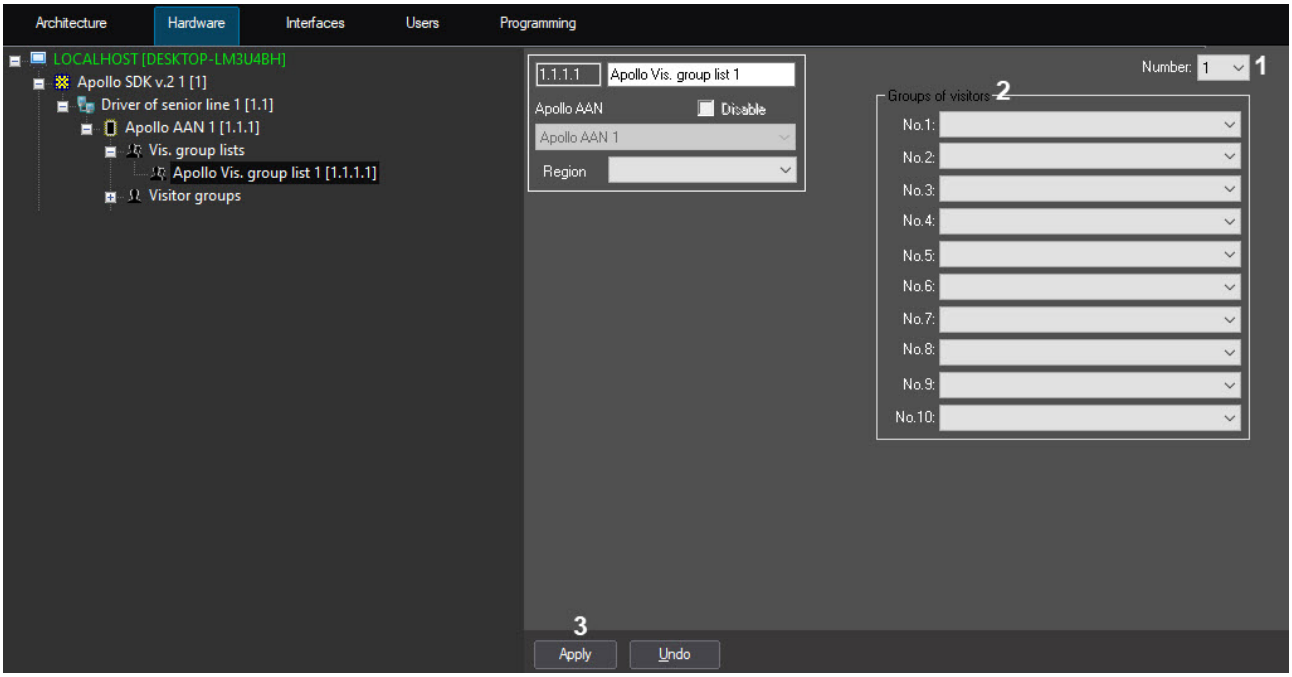


2. In the **Number** drop-down list, select the visitor group number in the controller (**1**).
3. Click the **Apply** button (**2**).

Configuring visitor groups is completed.

Configuring visitor group lists

Configuring visitor group lists is performed on the **Apollo Vis. group list** object settings panel. This object is created on the basis of **Apollo AAN** on the **Hardware** tab of the **System settings** dialog box.



Visitor group lists are configured as follows:

1. Go to the **Apollo Vis. group list** object settings panel.
2. In the **Number** drop-down list, select the visitor group list number in the controller (**1**).
3. In the drop-down lists in the **Groups of visitors** group, select **Apollo Visitor group** objects corresponding to groups that are to be included in this group list (**2**).
4. Click the **Apply** button (**3**).

Configuring visitor groups is completed.

4.3.9 Configuring APB Area

The APB Areas allow controlling the number of users in the area and denying access when maximal visitors number is reached. The APB Areas can also be used for antipassback (see Configuring antipassback).

> **Attention!**
>
> At least two APB Areas corresponding to entrance and exit areas of the AIM controller are to be configured in the system (see also Configure AIM Readers). If just one APB area is configured, the system will lock it when it is filled and it will be impossible to reset the counter. See also Example of APB Areas configuration.

The APB Area is configured on the **Apollo APB Area** object settings panel, that is created on the basis of the **Apollo AAN** object on the **Hardware** tab of the **System settings** dialog box.

APB area is configured as follows:

1. Go to the **Apollo APB Area** object settings panel.
2. In the **Number** drop-down list, select the APB area number in the controller (**1**).
3. In the **Connection with Axxon PSIM region** drop-down list, select the **Region** object corresponding to this APB Area (**2**). This Region object is to be set as entrance or exit region of an Apollo AIM reader (see also Configure AIM Readers).
4. In the **Initial mode** drop-down list, select the initial state of the APB area that will be set when writing configuration to controller: **Closed** or **Opened** (**3**).
5. In the **Mode when zone is closed** drop-down list, select access rule for closed APB area, i.e. when maximum number of users in it is exceeded (**4**).
6. In the **Reader to get into zone when it has "closed" status** drop-down list, select corresponding AIM reader (**5**).
7. Specify maximum number of people in the area (**6**).
8. If two cards are required to enter the zone, set the corresponding checkbox (**7**).
9. If users are to be allowed to enter the closed zone if commands are enabled, set the corresponding checkbox (**8**).
10. If zone status is to be changed at access if commands are enabled, set the corresponding checkbox (**9**).
11. Click **Apply** (**10**).
12. To save the changes to the controller, write configuration to hardware.

> **Attention!**
>
> User counter in zone is reset when configuration is written.

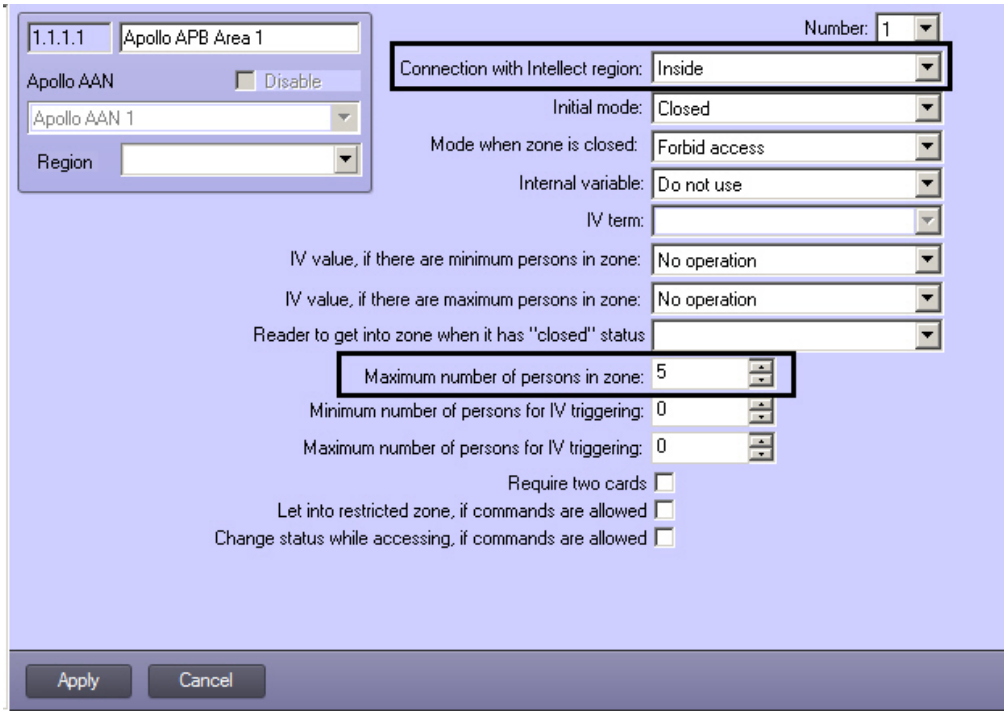
Example of APB Areas configuration

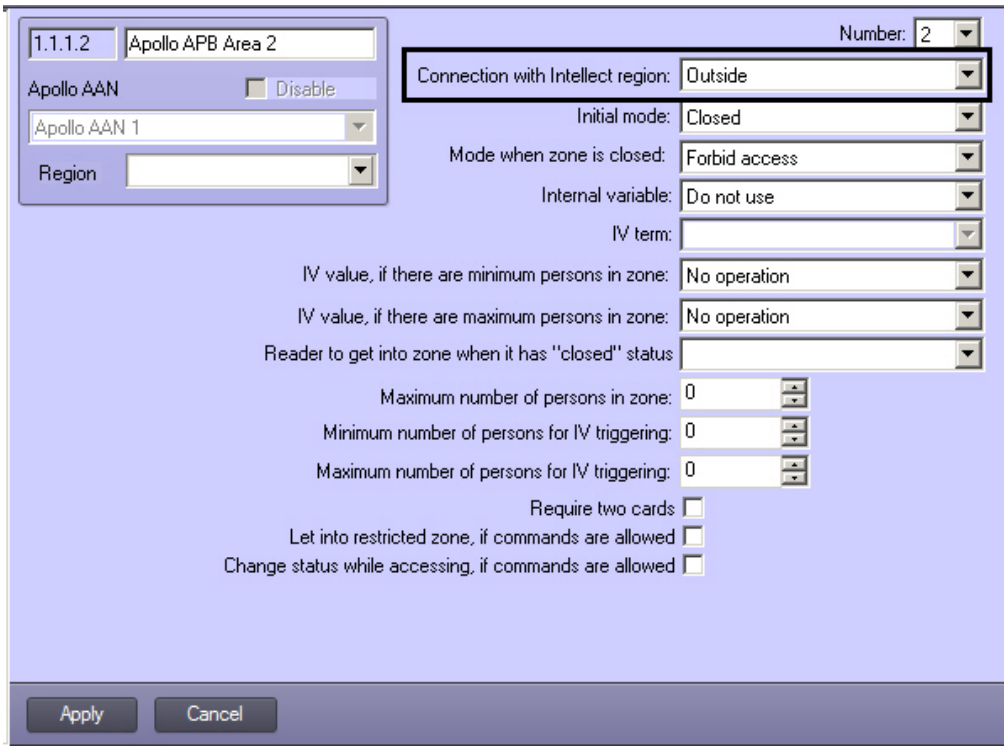
At least 2 APB areas are to be configured in the system corresponding to entrance and exit regions of a reader. The first APB zone is intended to control people count, and limits are set in it for number of users in region. The second

APB area does not necessarily have limitations on user number but entrance to this zone reduces number of users in the first zone according to readers configuration.

Create 2 regions in *ACFA PSIM*: Inside and Outside. These regions are set as entrance or exit areas for readers 1 and 2:

The APB Zone 1 is associated with the Inside region and has limited user count of 5. The APB Zone 2 is associated with the Outside region and has no limitations on user count.





When 5 persons has entered the APB Area 1 through Reader 1, the area is locked and it is impossible ti enter in it. But if 1 person leaves APB Area 1 through Reader 2 (i.e. enters APB Area 2), then APB Area 1 unlocks as so as the number of persons in it becomes 4.

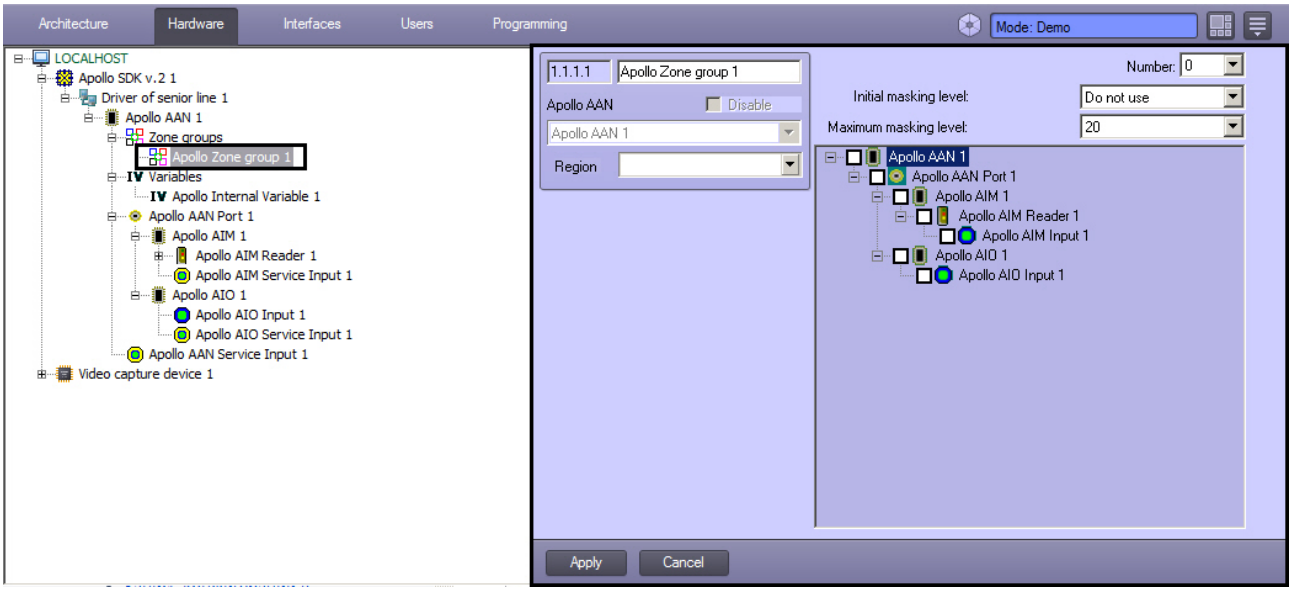
So, not more than 5 persons in zone is maintained.

> **Note.**
>
> There can be more than 2 readers for entrance and exit APB area.

4.3.10 Configure zone group

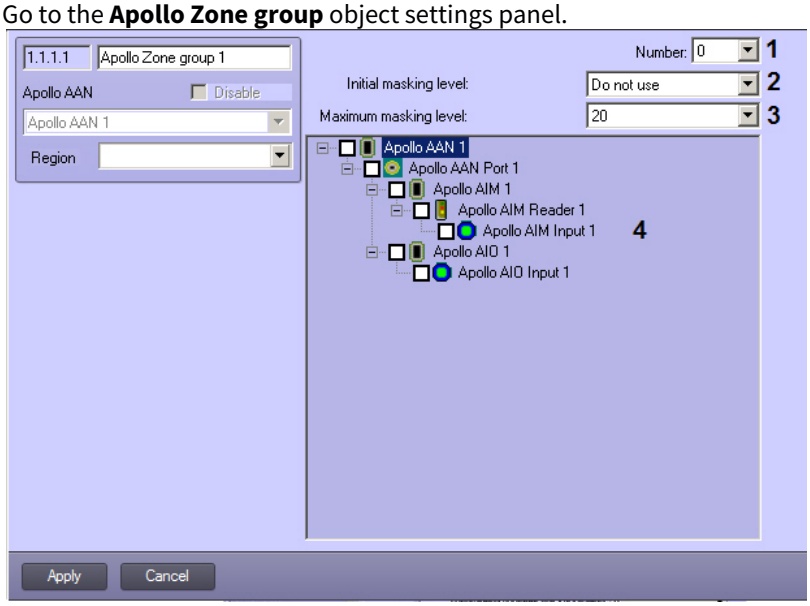
Zone groups allow you to combine zones (AIO and/or AIM inputs and/or status inputs), mask and unmask the entire group.

Zone group configuration is performed on the **Apollo Zone group** object settings panel. The object is created under **Apollo AAN** object on the **Hardware** tab of the **System Settings** dialog box.



Zone group configuration is performed as follows:

1. Go to the **Apollo Zone group** object settings panel.



2. Select zone group ID in the controller from the **Number** drop-down list (1).
3. From the **Initial masking level** drop-down list (2), select the initial masking level. If the masking level is not zero, then the group is considered to be masked, if zero – the group is unmasked.
4. From the **Maximum masking level** drop-down list (3), select the maximum masking level.
5. Set checkboxes next to objects to be included in the group (4).
6. Click **Apply**.
7. To save the changes to the controller, write configuration to hardware.

Zone group configuration is completed.

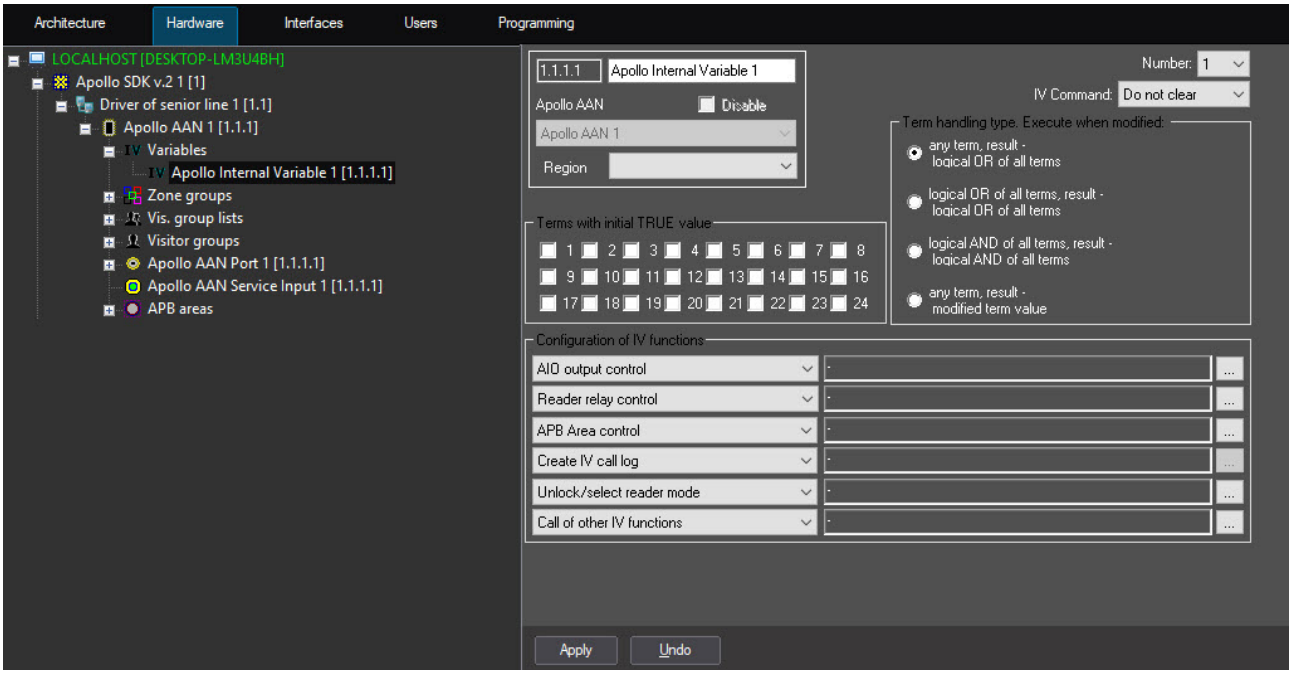
4.3.11 Configuring internal variables in ApolloSDK

General description of the internal variables mechanism

The internal variables in *ApolloSDK* allow configuring complex internal associations and reactions in the *Apollo* equipment. Internal variables can include up to 24 terms—logical elements, the value of which affects the execution of a command specified by an internal variable (one variable can contain up to 6 commands). The terms are changed by corresponding devices. In particular, you can configure the connection of terms with the states of the following devices: the AIM reader, the AIO input, the APB area. The association of an object with a term of the internal variable is configured in the settings panel of the corresponding object.

> **Note**
>
> In addition to automatic execution of internal variable functions, it is possible to manually launch them from the Map—see Control the Apollo internal variable.

Configuration of actions performed when corresponding combination of terms is carried out on the **Apollo Internal Variable** object settings panel, which is created on the basis of the **Apollo AAN** object on the **Hardware** tab of the **System settings** dialog box.



> **Attention!**
>
> Since this mechanism is hardware-based, after setting up the relevant objects, send the configuration to *Apollo* hardware—see Write configuration to hardware.

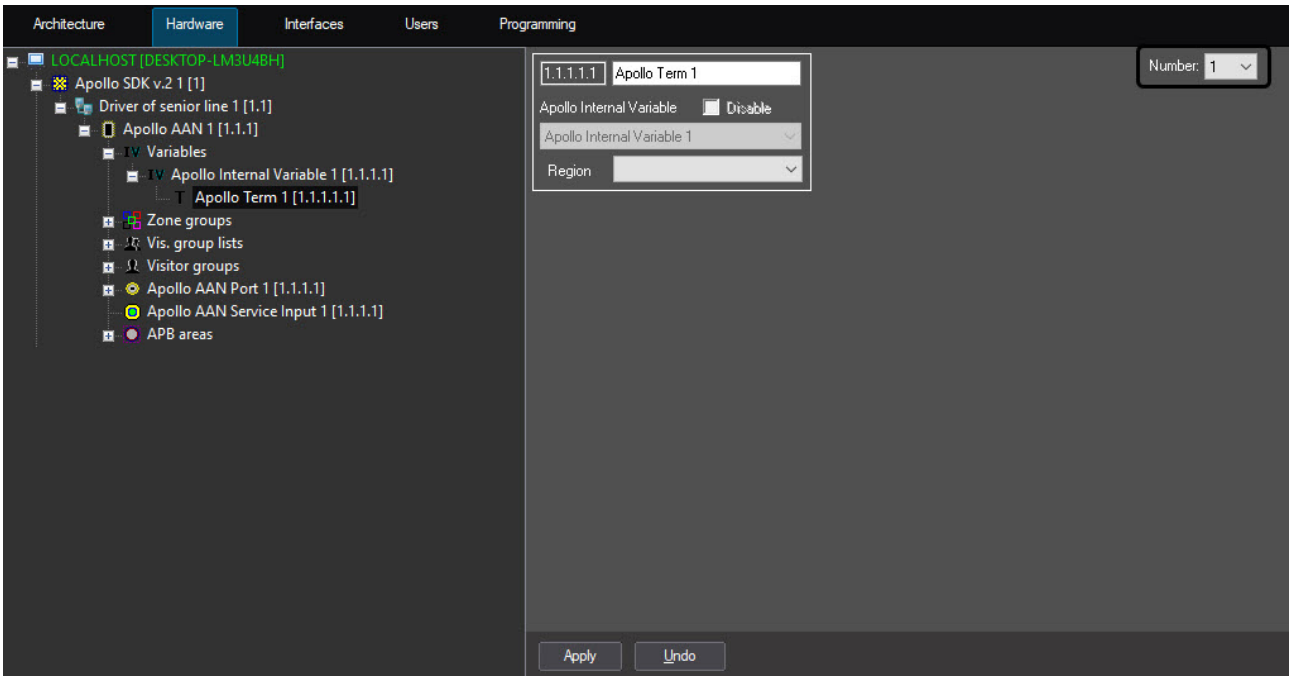
The configuration of *Apollo* internal variables is done in the following order:

1. Configuring internal variables objects in ApolloSDK
2. Associating objects with terms

Configuring internal variables objects in ApolloSDK

Creating and configuring terms

The **Apollo Term** objects are created on the basis of the corresponding **Apollo Internal Variable** object on the **Hardware** tab of the **System settings** dialog box.



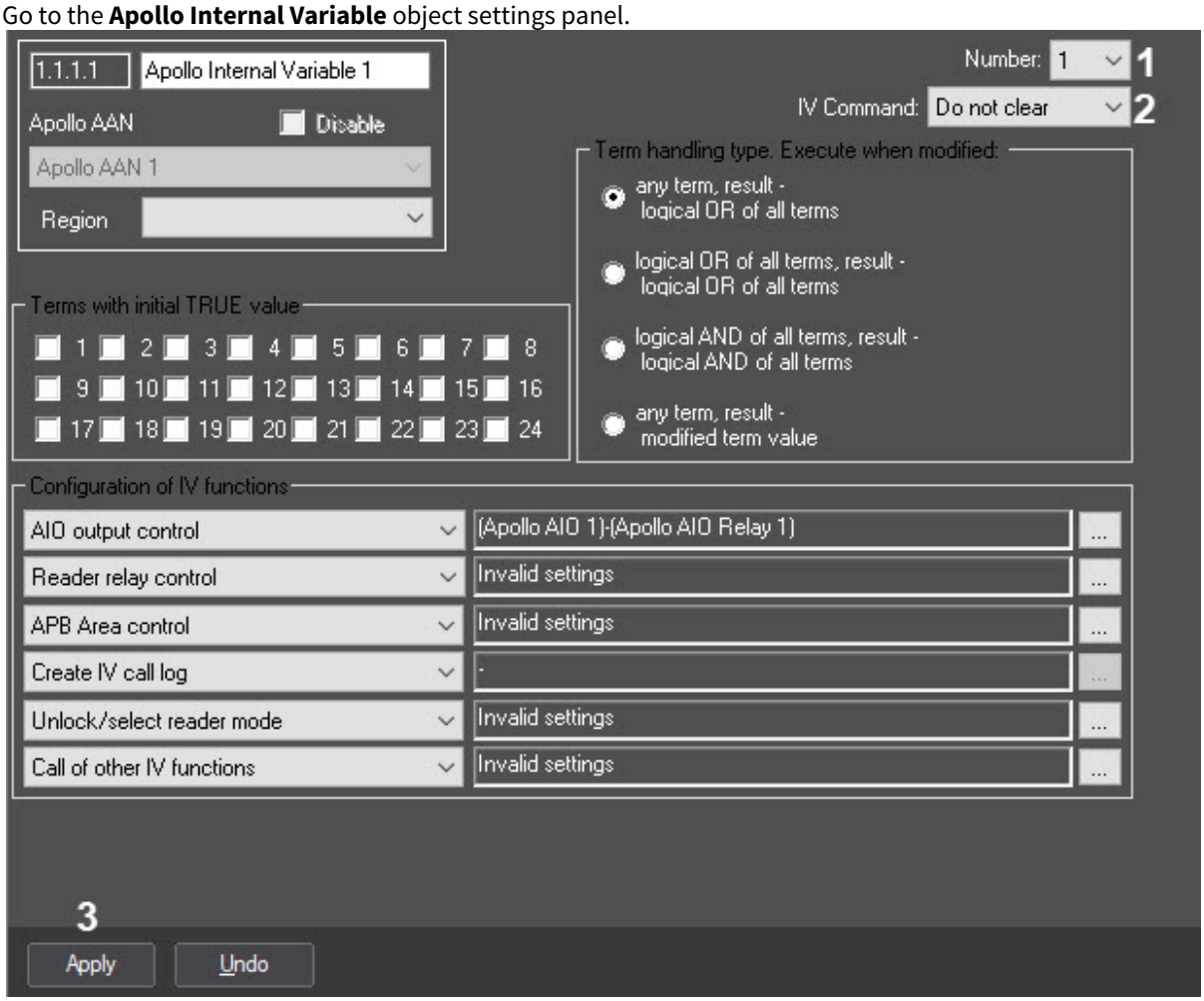
To set up a term, select its number in the corresponding drop-down list.

After all required terms are created, write configuration to hardware.

General settings of internal variable

The general parameters of the internal variable are set in the following order:

1. Go to the **Apollo Internal Variable** object settings panel.



2. In the **Number** drop-down list, select the internal variable number in the controller (**1**).
3. In the **IV Command** drop-down list, select command for internal variable: **Clear** or **Do not clear** (**2**).
4. Click the **Apply** button (**3**).
5. To save the changes to the controller, write configuration to hardware.

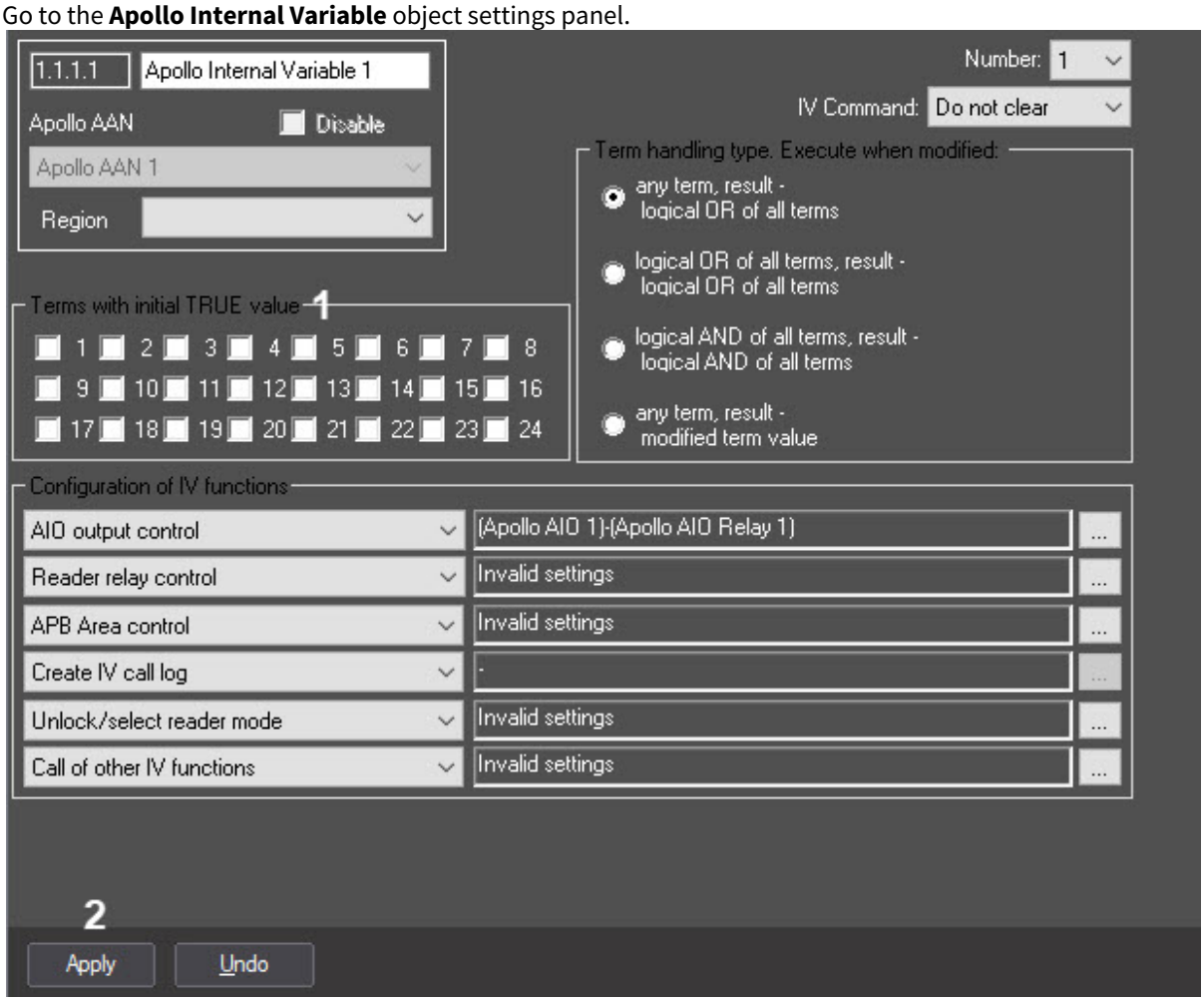
The general parameters of the internal variable are set.

Setting initial values of terms

The values of all the 24 terms in the internal variable must be initialized no matter how many terms will be used in this internal variable. By default, when you create an **Apollo Internal Variable** object, the value of all terms is set to FALSE. Initial values can subsequently be changed in accordance with the rules set for the corresponding Apollo equipment.

The initial values of the terms are set as follows:

1. Go to the **Apollo Internal Variable** object settings panel.



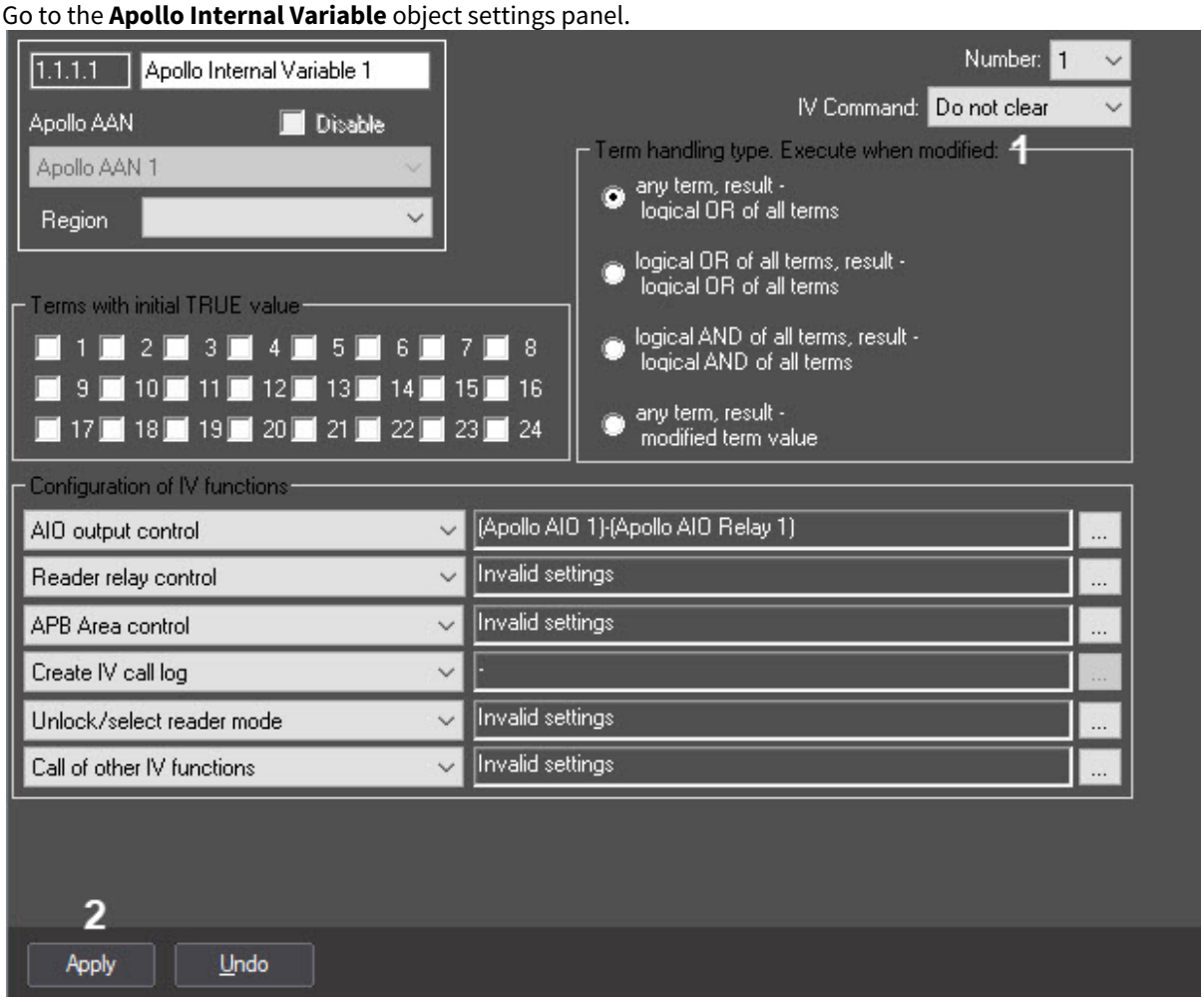
2. Set the checkboxes next to the numbers of those terms that should be set to TRUE (**1**) when writing the internal variable to the controller.
3. Click the **Apply** button (**2**).
4. To save the changes to the controller, write configuration to hardware.

Setting initial values of terms is completed.

Configuring logical condition for terms processing

Configuration of the logical condition for terms processing is carried out in the following order:

1. Go to the **Apollo Internal Variable** object settings panel.



2. Set the **Term handling type. Execute when modified** switch to one of the positions described in the table below (**1**).

Switch position	Description
any term, result—logical OR of all terms	Actions specified in the internal variable will be executed when any term is changed, provided that the logical OR of all terms is TRUE
logical OR of all terms, result—logical OR of all terms	Actions specified in the internal variable are performed if at least one term is TRUE
logical AND of all terms, result—logical AND of all terms	Actions specified in the internal variable are performed if all terms are TRUE
any term, result—modified term value	Actions specified in the internal variable will be performed when any term is changed

3. Click the **Apply** button (**2**).
4. To save the changes to the controller, write configuration to hardware.

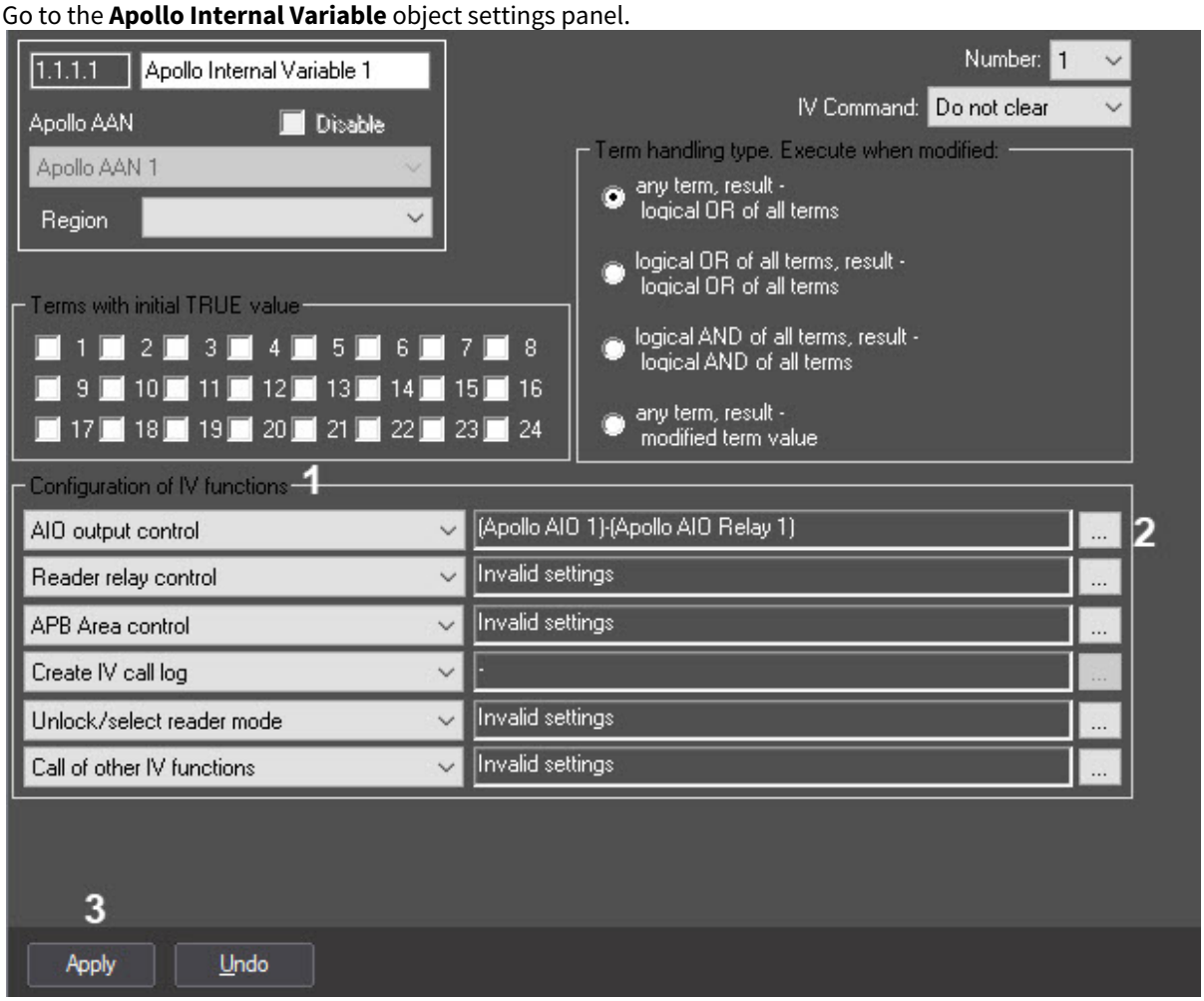
Configuring logical condition for terms processing is completed.

Configuring internal variable functions (actions)

The functions of the internal variable determine the actions to be taken when the corresponding logical condition is fulfilled. Up to 6 IV functions can be set, while an IV function can start other IV function.

The IV functions are configured as follows:

1. Go to the **Apollo Internal Variable** object settings panel.



2. Select the required function in the corresponding drop-down list (**1**, see table).
3. Click **...** and configure the function parameters (**2**, see table).

Function	Description	Parameters
AIO output control	Switch relay on AIO security panel	Security panel Panel relay
Reader relay control	Switch relay of a reader connected to an AIM controller	AIM controller Reader Relay
APB Area control	Change APB area state	APB Area
Create IV call log	Logging a message about fulfillment of conditions of the internal variable	-

Unlock/select reader mode	Setting operation mode of a reader connected to an AIM controller	AIM controller Reader type Mode
Call of other IV functions	Performing functions of another internal variable, regardless of the meaning of its terms	Variable
Shift of masking level of zone group	Increment masking level by 1 for each IV's term, if the term is TRUE	Zone group
Check zone activity in zone group	The term is set to FALSE (0) if there are active zones in the group	Zone group
Check masking level of zone group	The term is set to 1, if masking level = 0. The term is set to 0 if masking level != 0	Zone group
Switch masking level of zone group	Change masking level from 0 to 1 or from 1 to 0, i.e. trigger mode.	Zone group

4. Click the **Apply** button (**3**).
5. To save the changes to the controller, Write configuration to hardware.

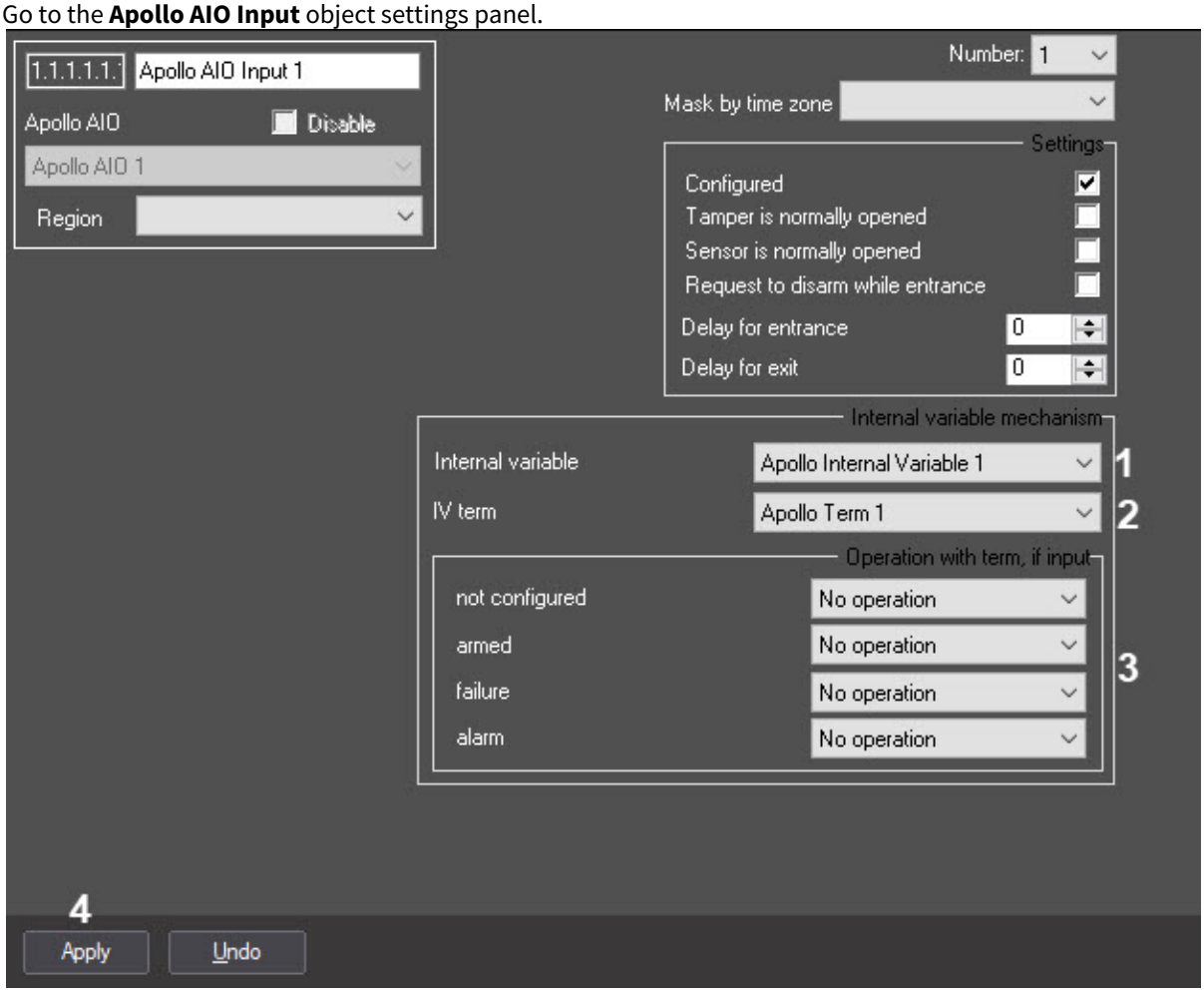
Configuring IV functions is completed.

Associating objects with terms

Associating AIO input with a term

Associating AIO input with a term is performed as follows:

1. Go to the **Apollo AIO Input** object settings panel.



2. In the **Internal variable** drop-down list, select the required **Apollo Internal Variable** object (**1**).
3. In the **IV term** drop-down list, select the **Apollo Term** object corresponding to a term to be controlled by the state of this **Apollo AIO Input** object (**2**).
4. From the drop-down lists, in the **Operation with term, if input** group, select which values the term should have if the AIO input is not configured, armed, in a fault state or in an alarm (**3**). The following operations are available:

Operation	**Description**
No operation	The value of the corresponding term does not change
Clear term	The term is set to FALSE
Set term	The term is set to TRUE
Send impulse	The value of the term changes to the opposite for a short preset period

5. Click the **Apply** button (**4**).
6. To save the changes to the controller, write configuration to hardware.

Associating AIO input with a term is completed.

Associating APB area with a term

Configuring the association of APB areas with terms of internal variables is not mandatory.

Configure the association of APB areas with terms of internal variables is carried out as follows:

1. Go to the **Apollo APB Area** object settings panel.
2. In the **Internal variable** drop-down list select the **Apollo Internal Variable** object corresponding to the required internal variable (1).
3. In the **IV term** drop-down list select the required **Apollo Term** object (2).
4. In the **IV value if there are minimum persons in zone** drop-down list select an operation to be performed on term when the condition is fulfilled (3, see the table).
5. In the **IV value if there are maximum persons in zone** drop-down list select an operation to be performed on term when the condition is fulfilled (4, see the table).

Operation	Description
No operation	The value of the corresponding term does not change
Clear term	The term is set to FALSE
Set term	The term is set to TRUE
Send impulse	The value of the term changes to the opposite for a short preset period

6. Set the required value in the **Minimum number of persons for IV triggering** field (5).
7. Set the required value in the **Maximum number of persons for IV triggering** field (6).
8. Click **Apply** (7).
9. To save the changes to the controller, write configuration to hardware.

Configure the association of APB areas with terms of internal variables is completed.

Associating AIM reader commands with terms

Associating AIM reader commands with terms allows changing term value on corresponding command from reader.

> **Attention!**
>
> The commands are executed in the **Card and PIN** reader mode only. In other modes, commands are not executed (see **1** in the figure below).
> Most of the reader commands are only supported for AP-500 reader. The usual reader with a keyboard support PIN+10 and PIN+20 commands (2 and 3 commands, correspondingly).

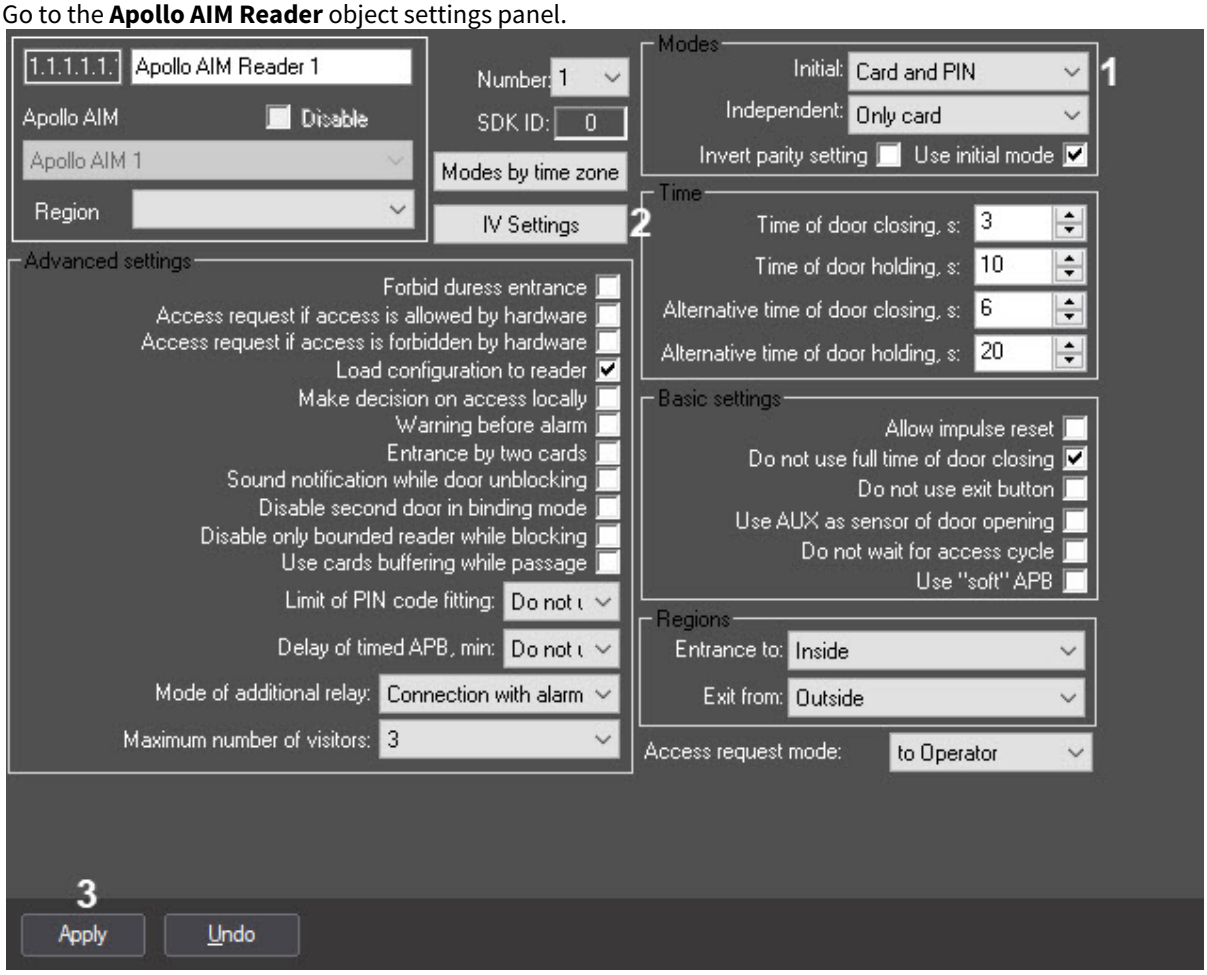
To enter a command using AP-500 reader, present a card, then press Cmd, enter a command number and press <Enter>. The even command sets corresponding term to TRUE, the odd command sets it to FALSE.

Correspondingly, when PIN+10 is entered, the term assotiated with 2 and 3 commands will be set to TRUE, when PIN+20 is entered, it will be set to FALSE.

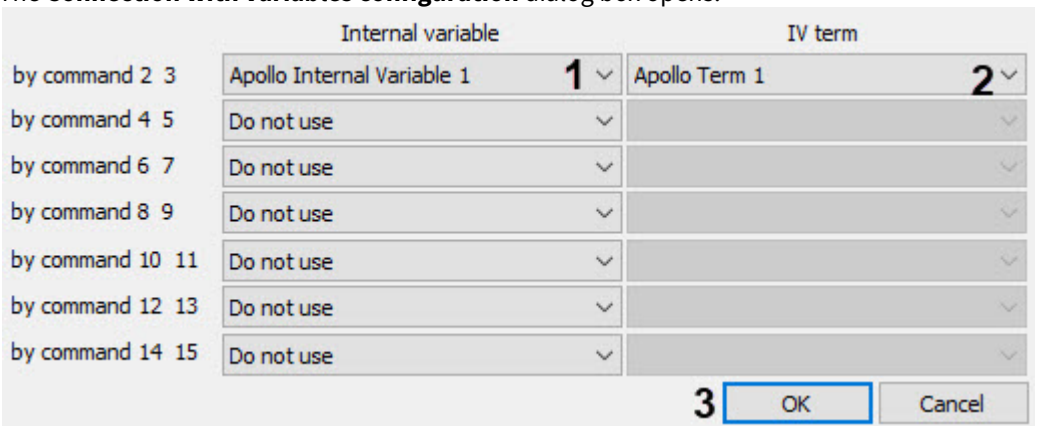
> **Note**
>
> PIN+10 and PIN+20 means that 10 or 20 is added to the PIN number, correspondingly. For example, if a user has PIN code 8080, then PIN+10 = 8090. The user will have to enter 8090 to set corresponding term to TRUE.

Assigning AIM reader commands with terms is performed as follows:

1. Go to the **Apollo AIM Reader** object settings panel.



2. Click the **IV Settings** button (**2**).
3. The **Connection with variables configuration** dialog box opens.



4. In the drop-down list in the **Internal variable** column, select the **Apollo Internal Variable** object corresponding to the variable the term of which the command from the reader should change (**1**).
5. In the **IV term** drop-down list, select the **Apollo Term** object corresponding to the required internal variable (**2**).
6. Click the **OK** button (**3**).
7. Click the **Apply** button (**3**).
8. To save the changes to the controller, write configuration to hardware.

Assigning AIM reader commands with terms is completed.

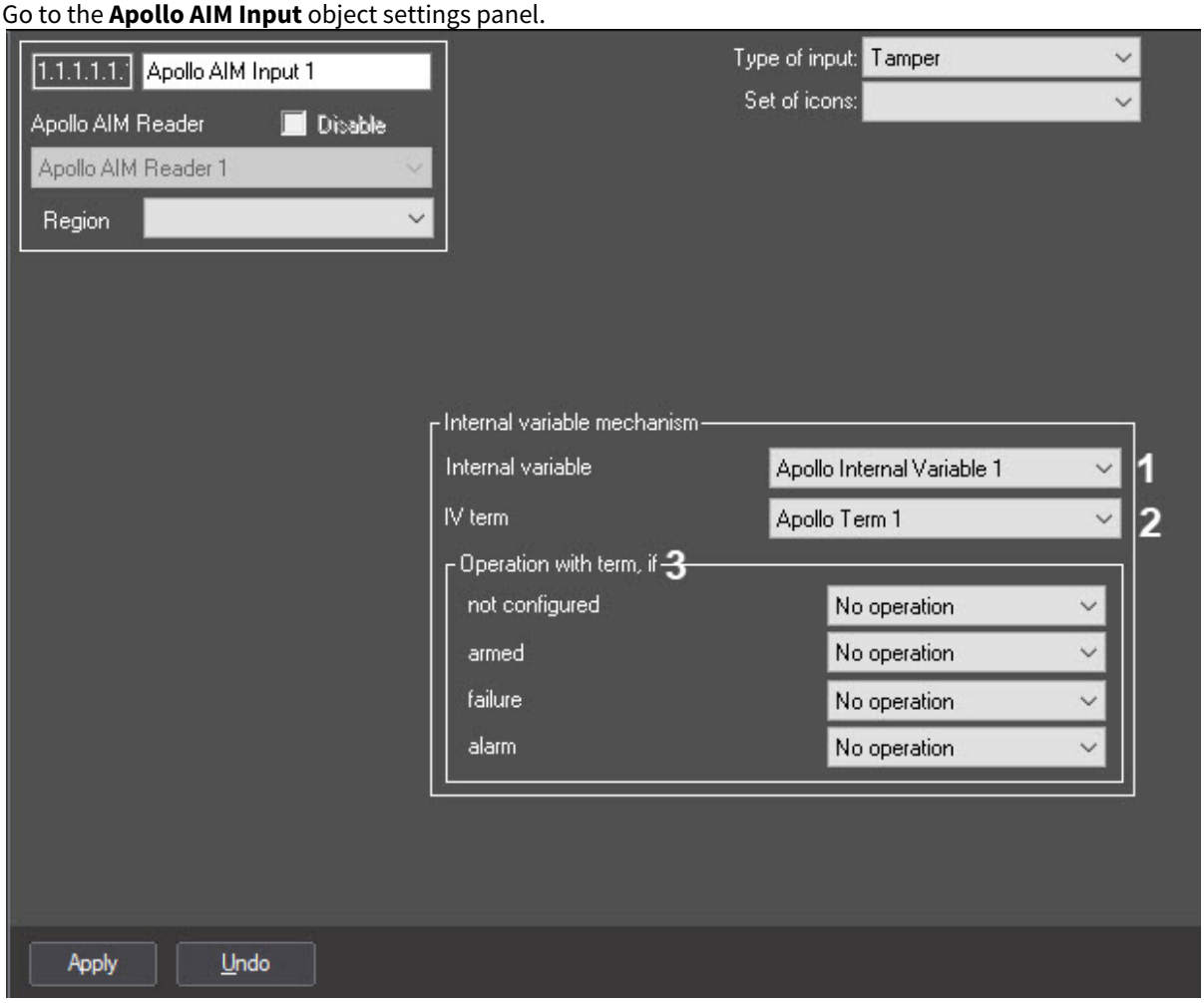
Associating AIM input with terms

Associating AIM input with terms allows changing internal variable's term value at certain events.

Associating AIM input with terms is performed on the **Apollo AIM Input** object settings panel. The object is created under **Apollo AIM Reader**.

Associating AIM input with a term is performed as follows:

1. Go to the **Apollo AIM Input** object settings panel.



2. In the **Internal variable** drop-down list, select the required **Apollo Internal Variable** object (**1**).
3. In the **IV term** drop-down list, select the **Apollo Term** object corresponding to a term to be controlled by the state of this **Apollo AIM Input** object (**2**).
4. From the drop-down lists in the **Operation with term, if** group, select which values the term should have if the AIM input is not configured, armed, in a fault state or in an alarm (**3**). The following operations are available:

Operation	**Description**
No operation	The value of the corresponding term does not change
Clear term	The term is set to FALSE

Set term	The term is set to TRUE
Send impulse	The value of the term changes to the opposite for a short preset period

5. Click the **Apply** button.
6. To save the changes to the controller, write configuration to hardware.

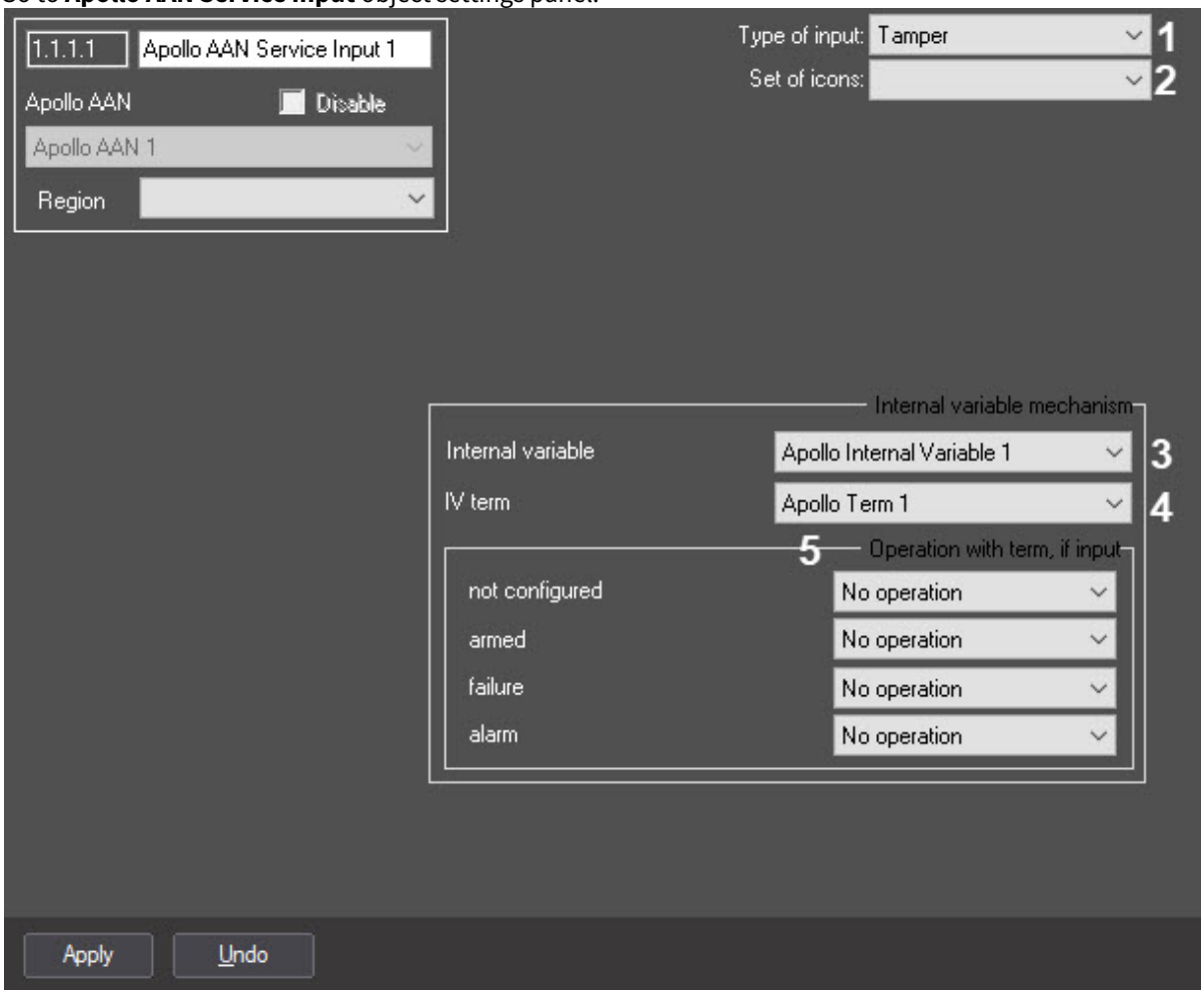
Associating AIM input with a term is completed.

Associating AAN, AIO or AIM Service Input with terms

Associating *AAN, AIO* or *AIM* Service Input with terms allows changing internal variable's term value at certain events.

The configuration steps below are shown for **Apollo AAN Service Input**. Associating *AIO* or *AIM* service inputs with terms is performed in the same way (with one exception, see the bottom of the section).

1. Go to **Apollo AAN Service Input** object settings panel.



2. Select an object corresponding to the event from the **Type of input** drop-down list (**1**, see table below).

Object	Description
Host connection	Initiates an internal variable if the AAN controller's status of communication with the computer changes
Power failure	Initiates an internal variable on power failure
Tamper	Initiates an internal variable on housing intrusion

3. Select the set of icons for input on the **Map** from the **Set of icons** drop-down list (**2**).
4. From the **Internal variable** drop-down list, select the required **Apollo Internal Variable** object (**3**).
5. From the **IV term** drop-down list, select the **Apollo Term** object corresponding to a term to be controlled by the state of this **Apollo AAN Service Input** object (**4**).
6. From the drop-down lists in the **Operation with term, if input** group, select which values the term should have if the AIM input is not configured, armed, in a fault state or in an alarm (**5**). The following operations are available:

Operation	Description
No operation	The value of the corresponding term does not change
Clear term	The term is set to FALSE
Set term	The term is set to TRUE
Send impulse	The value of the term changes to the opposite for a short preset period

7. Click the **Apply** button.
8. To save the changes to the controller, write configuration to hardware.

Associating Apollo AAN Service Input with term is completed.

Associating *AIO* (**Apollo AIO Service Input** object) or *AIM* (**Apollo AIM Service Input** object) service inputs with terms is performed in the same way with one exception: in the **Type of input** drop-down list, the following values are available:

Object	Description
Connection status	Initiates an internal variable if the AAN controller's status of communication with the AIO or AIM panel changes
Tamper	Initiates an internal variable on housing intrusion
Power failure	Initiates an internal variable on power failure

4.3.12 Configuring antipassback

To configure the antipassback, do the following:

1. Create the required regions in *Axxon PSIM* (see Subdivision of the protected facility into areas and regions).
2. On the settings panel of the **ApolloAAN** object, enable the zonal or temporary antipassback control (see Configure AAN controller).
3. Specify the created regions for the AIM readers in the **Entrance to** and **Exit from** fields (see Configure AIM Readers).
4. Create an **Apollo APB Area** object for each AIM reader. Specify the created regions in the **Connection with Axxon PSIM region** field of the APB Area settings panel (see Configuring APB Area).

   > **Note**
   >
   > It is recommended to leave the remaining settings of the **Apollo APB Area** object to default.

5. Write the current configuration to the AAN controller (see Write configuration to hardware).
6. In the *Access Manager* module, set the **Antipassback** parameter value to **Yes** for the required employee (see Setting user parameters).

Configuring antipassback is complete.

4.4 Configure system without AAN central controller

Configure the system without central controller in the following order:

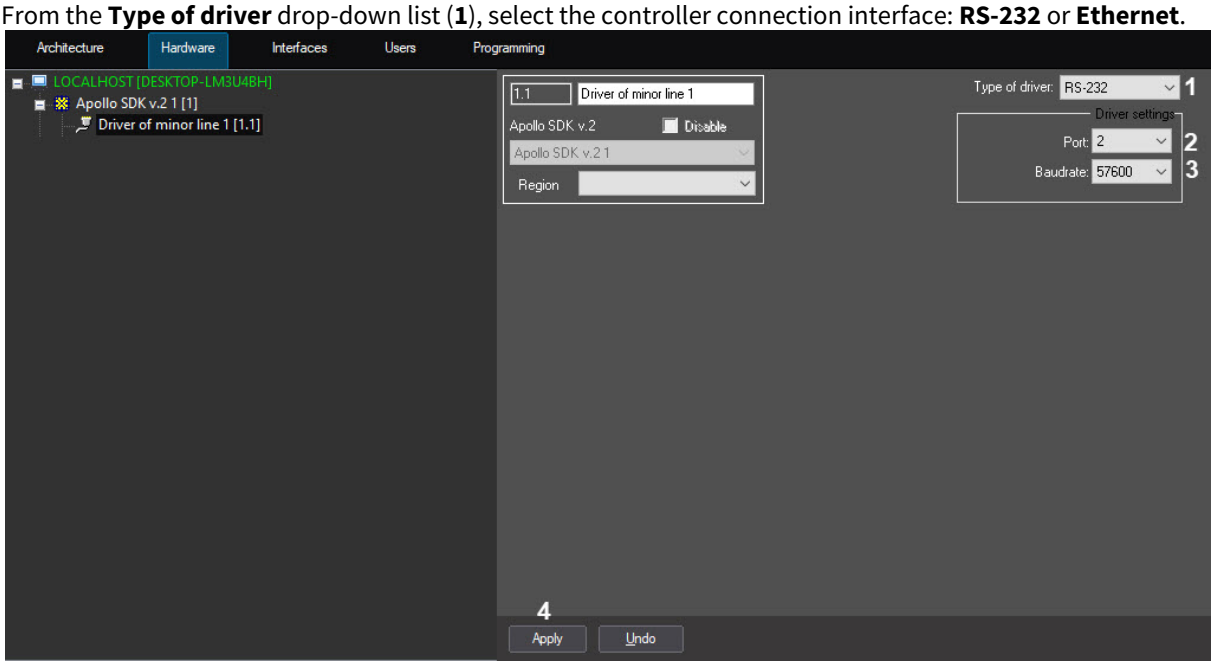
1. Create the **Driver of minor line** object and specify parameters of the *AIM SC* modules connection (see Configure connection of AIM SC controller).
2. Create the **Apollo AIM SC** object on the basis of this object and configure the *AIM* module (see Configure AIM SC controller).
3. Create the **Apollo AIM SC Reader** objects and objects of virtual inputs and relay (see Configure AIM SC reader).

4.4.1 Configure connection of AIM SC controller

The *AIM SC* controller connection is configured on the settings panel of the **Driver of minor line** object. This object is created on the basis of the **Apollo SDK v.2** object.

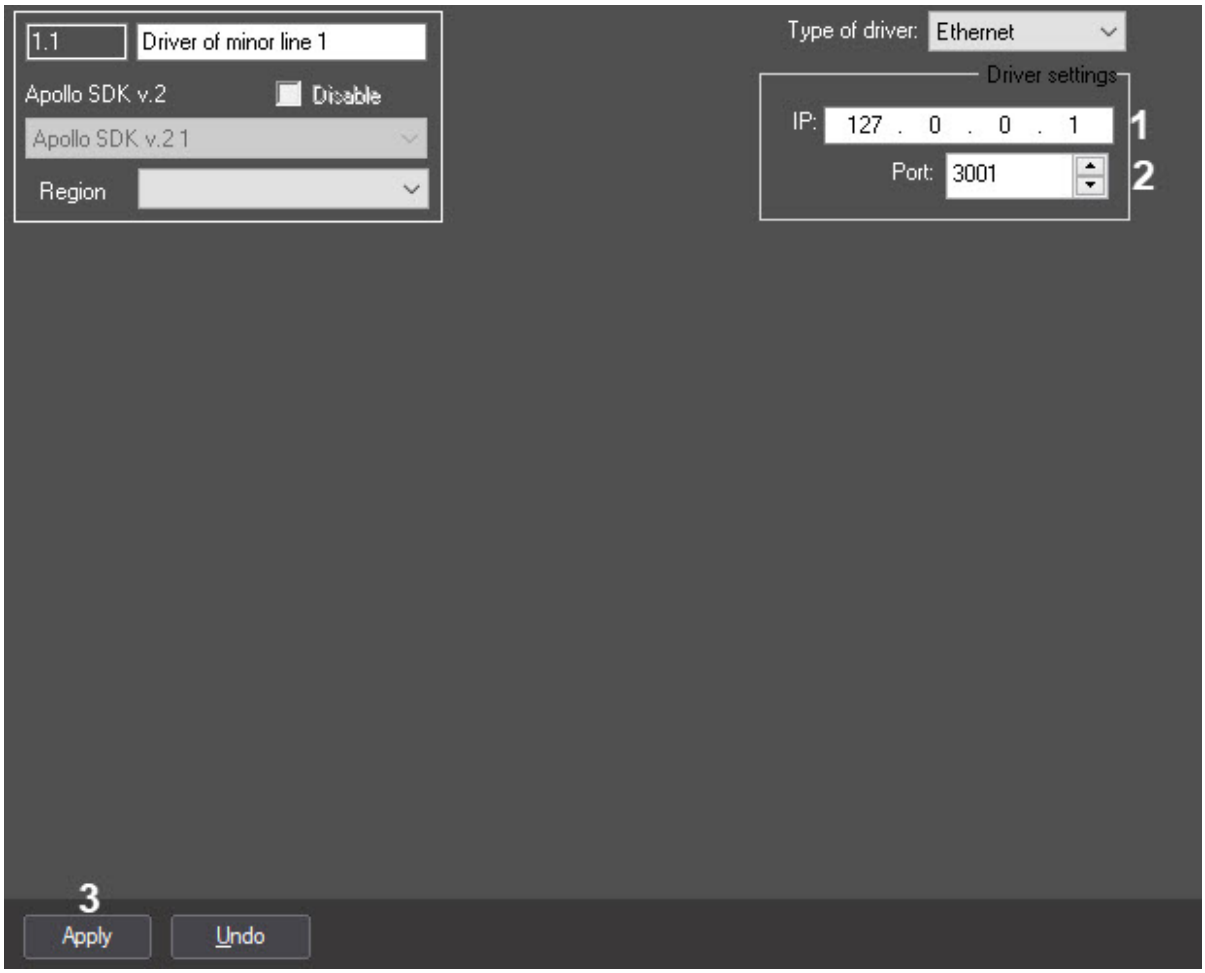
To connect the *AIM SC* controller, do the following:

1. From the **Type of driver** drop-down list (**1**), select the controller connection interface: **RS-232** or **Ethernet**.



2. If the controller is connected via a COM port:
    a. From the **Port** drop-down list (**2**), select the number of the port to which the controller is connected.
    b. From the **Baudrate** drop-down list (**3**), select the baudrate.
    c. Click the **Apply** button (**4**).

3. If the controller is connected via Ethernet:



   a. In the **IP** field (**1**), specify the controller's IP address.
   b. In the **Port** field (**2**), specify the controller port number.
   c. Click the **Apply** button (**3**).

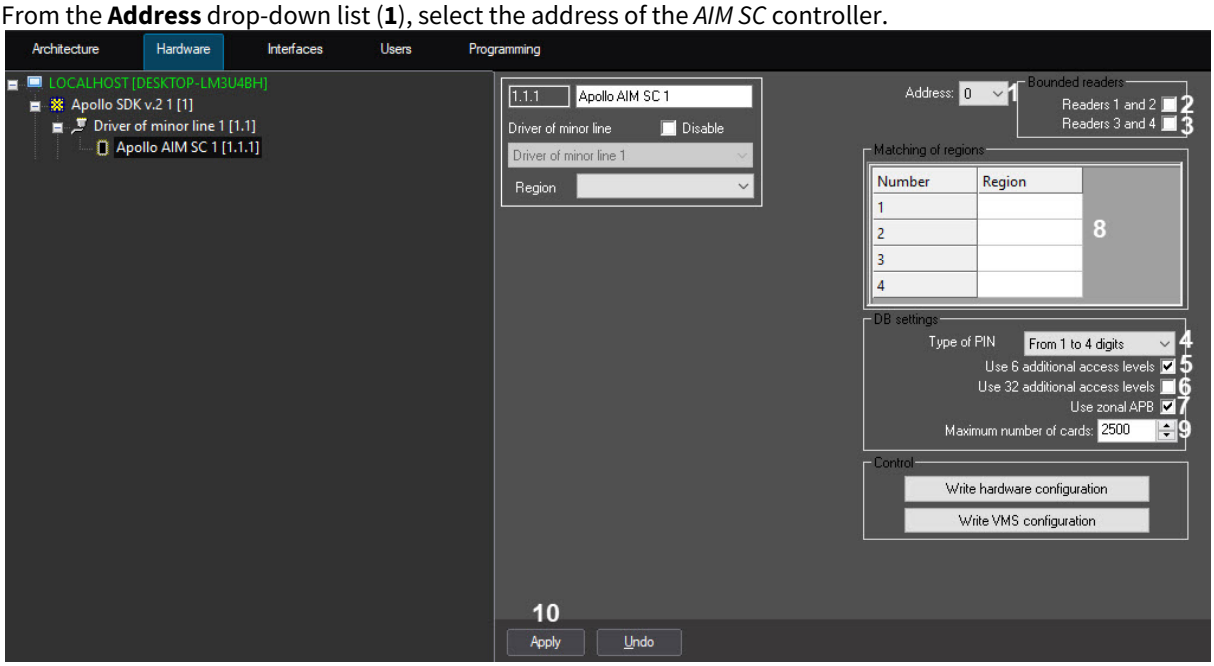
The *AIM SC* controller connection is complete.

4.4.2 Configure AIM SC controller

The *AIM SC* controller is configured on the settings panel of the **Apollo AIM SC** object. This object is created on the basis of the **Driver of minor line** object.

To configure the *AIM SC* controller, do the following:

1. From the **Address** drop-down list (**1**), select the address of the *AIM SC* controller.



2. Set the appropriate checkboxes (**2**) and (**3**) if the dual readers are used.

> **Note**
>
> Dual readers are used on different sides of the same door. If the readers are single, then they are used on one side of the door, and you can pass from the other side of the door using a button.

3. In the **Type of PIN** field (**4**), select the length of the used PIN code. Select the **Do not use** value if it's not required to use PIN code.
4. Set the corresponding checkboxes if it's required to use additional 6 (**5**) or 32 (**6**) access levels.

> **Attention!**
>
> The maximum number of users that can be stored in controller memory decreases if the additional access levels are used.

5. Configure the antipassback control. To do this, set the **Use zonal APB** checkbox (**7**). To configure global antipassback control regardless of the number of *AIM SC* controllers, the following is also required:
   a. Create regions in *Axxon PSIM.*
   b. Assign regions to readers.
   c. Associate each region in *Axxon PSIM* with a region number in the *AIM SC* controller (**8**).
6. In the **Maximum number of cards** field (**9**), enter the maximum number of access cards that will be stored in the controller's memory.

> **Note**
>
> The maximum number of access cards that can be stored in the controller's memory depends on the number of memory cards installed on it.

7. Click the **Apply** button (**10**).

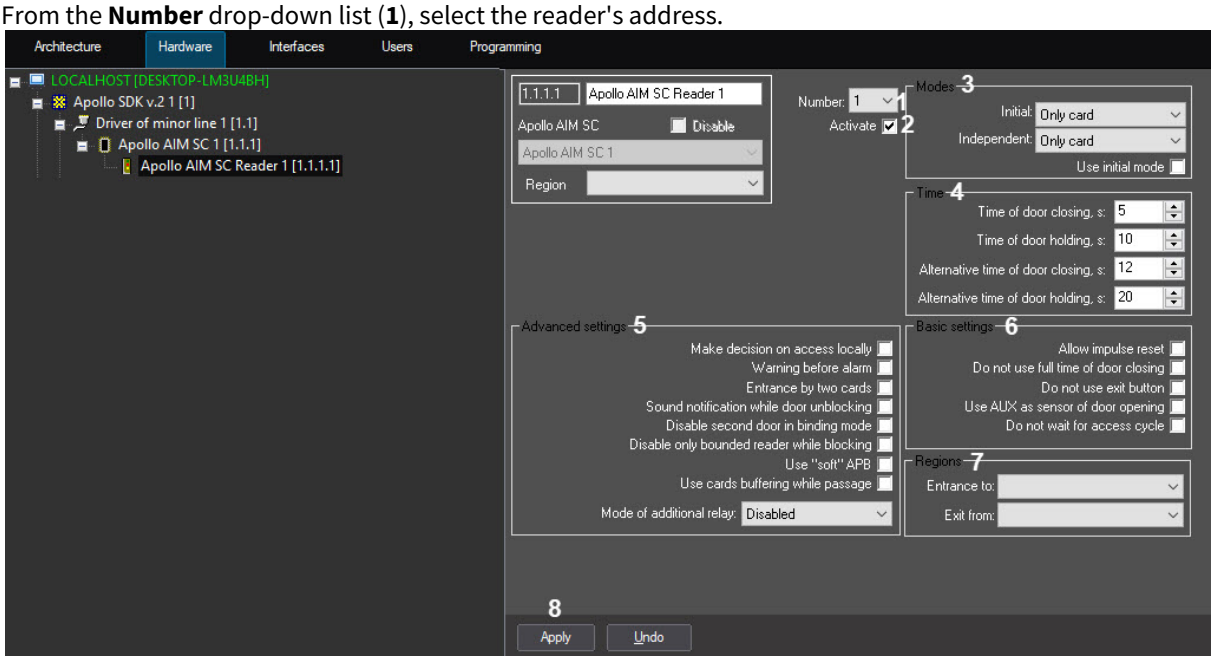
The *AIM SC* controller is now configured.

4.4.3 Configure AIM SC reader

The *AIM SC* reader is configured on the settings panel of the **Apollo AIM SC Reader** object. This object is created on the basis of the **Apollo AIM SC** object.

To configure the *AIM SC* reader, do the following:

1. From the **Number** drop-down list (**1**), select the reader's address.



2. Set the **Activate** checkbox (**2**) to activate the *AIM SC* reader.
3. In the **Modes** group (**3**), configure the reader operation modes:
   a. From the **Initial** drop-down list select the mode of reader operation while connection establishing.

Operation mode	Description
Closed	Access is closed for all
Only card	Access granted by access card
PIN or card	Access granted by access card or PIN code
Card and PIN	Access granted by access card and PIN code
Opened	Access is opened for all
Facility code	Access is granted by facility code

   b. From the **Independent** drop-down, list select the mode of reader operation while losing the connection.
   c. Set the **Use initial mode** checkbox if it's required to switch to initial mode after writing of configuration.
4. In the **Time** group (**4**), set the time parameters of the reader:

Parameter	Parameter setting method	Description
Time of door closing, s	Enter the value in the field	Time of opening in seconds

Parameter	Parameter setting method	Description
Time of door holding, s	Enter the value in the field	**Time period** in seconds during which door is to be closed. Otherwise, the **Door holding** message will be triggered
Alternative time of door closing, s	Enter the value in the field	Alternative time of opening in seconds. It is in use by special command (from card, using macro or script)
Alternative time of door holding, s	Enter the value in the field	**Alternative time period** in seconds during which door is to be closed. Otherwise, the **Door holding** message will be triggered. It is in use by special command (from card, using macro or script)

5. In the **Advanced settings** group (**5**), set the additional reader parameters:

Parameter	Parameter setting method	Description
Make decision on access locally	Set the checkbox	Yes—the *AIM* module makes an access decision without the *AAN* central controller
Warning before alarm	Set the checkbox	Yes—trigger warning before alarm
Entrance by two cards	Set the checkbox	Yes—passage through the reader is performed by two cards
Sound notification while door unblocking	Set the checkbox	Yes—sound signal of reader while unblocking the door
Disable second door in binding mode	Set the checkbox	Yes—block the reader working for exit from secured areas
Disable only bounded reader while blocking	Set the checkbox	Yes—only bounded reader are blocked while locking signal No—all readers are blocked
Use "soft" APB	Set the checkbox	Yes—antipassback is available, but the corresponding mistake will be specified in message
Use cards buffering while passage	Set the checkbox	Yes—write access cards to the *AIM* while passage
Mode of additional relay	Select the value from the list	Disabled—mode of additional relay is disabled Enabled—mode of additional relay is enabled

6. In the **Basic settings** group (**6**), set the basic parameters of the reader:

Parameter	Parameter setting method	Description
Allow impulse reset	Set the checkbox	Yes—impulse reset enabled
Do not use full time of door closing	Set the checkbox	Yes—lock after door closing
Do not use exit button	Set the checkbox	Yes—disable exit button
Use AUX as sensor of door opening	Set the checkbox	Yes—use additional *AUX* sensor of *AIM* module as sensor for door opening
Do not wait for access cycle	Set the checkbox	Yes—passage is performed after making an access decision No—passage is performed after triggering a sensor of door opening

7. In the **Regions** group (**7**):
   a. From the **Entrance to** drop-down list, select the **Region** object corresponding to the area on the side of exit through this reader.
   b. From the **Exit from** drop-down list, select the **Region** object corresponding to the area on the side of entrance through this reader.

   > **Note**
   >
   > In order to be able to select partitions, it is necessary to bind these partitions in the **Matching of regions** group on the settings panel of the **Apollo AIM SC** object (see Configure AIM SC controller).

8. Click the **Apply** (**8**) button.

The *AIM SC* reader is now configured.

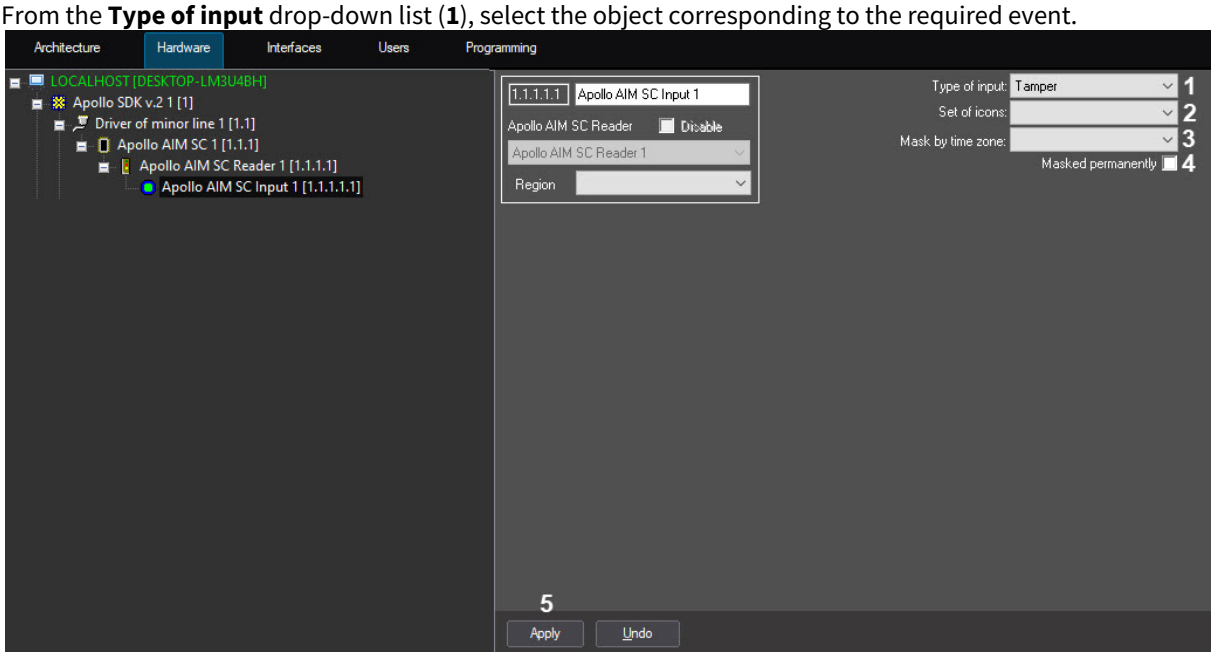
Configure input of AIM SC reader

It's possible to create and configure inputs of *AIM* readers. The inputs track the defined states of reader and trigger messages on which different reactions can be configured using scripts and macros.

Input of *AIM SC* reader is configured on the settings panel of the **Apollo AIM SC Input** object created on the basis of the **Apollo AIM SC Reader** object.

To configure the *AIM SC* reader input, do the following:

1. From the **Type of input** drop-down list (**1**), select the object corresponding to the required event.



2. From the **Set of icons** drop-down list (**2**), select the corresponding icon that will be displayed on the **Map**.
3. From the **Mask by time zone** drop-down list (**3**), select the time zone within which no events will be generated for the input.
4. Set the **Masked permanently** checkbox (**4**) to never generate the events for the input.

> **Note**
>
> During the masking period, the input cannot be in an alarm state.

5. Click the **Apply** button (**5**).

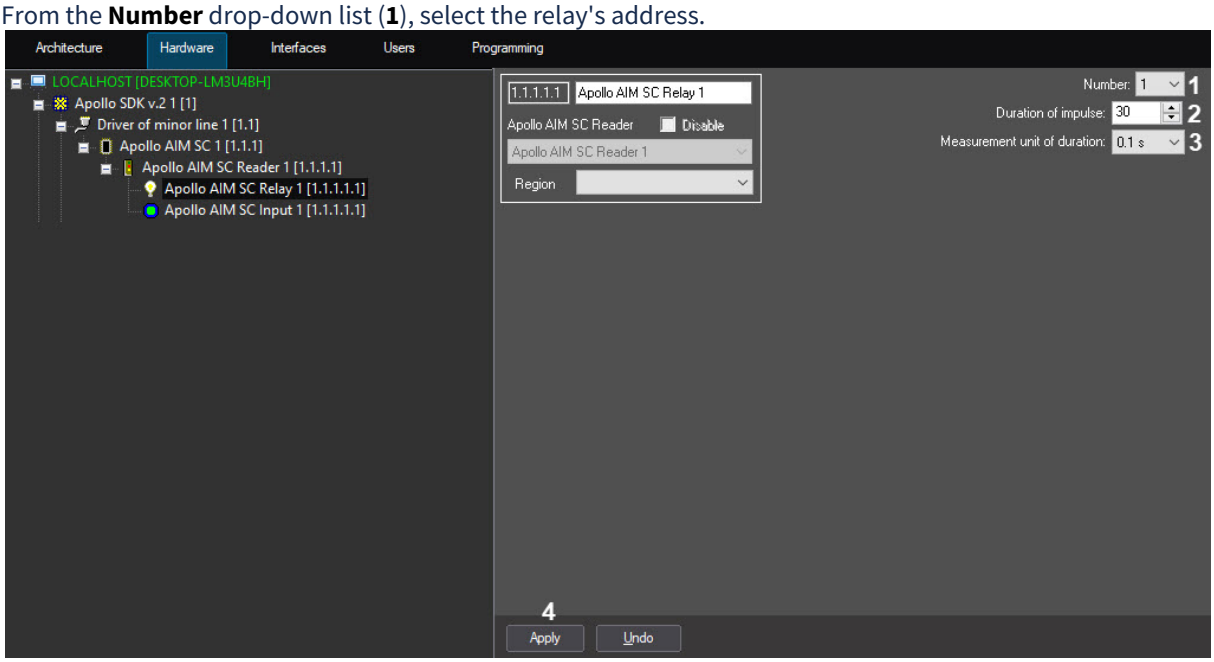
The *AIM SC* reader input is now configured.

Configure relay of AIM SC reader

Relay of *AIM SC* reader is configured on the settings panel of the **Apollo AIM SC Relay** object created on the basis of the **Apollo AIM SC Reader** object.

To configure the *AIM SC* reader relay, do the following:

1. From the **Number** drop-down list (**1**), select the relay's address.



2. In the **Duration of impulse** field (**2**), set the relay impulse duration in the unit of measurement selected in the **Measurement unit of duration** drop-down list (**3**).
3. Click the **Apply** button (**4**).

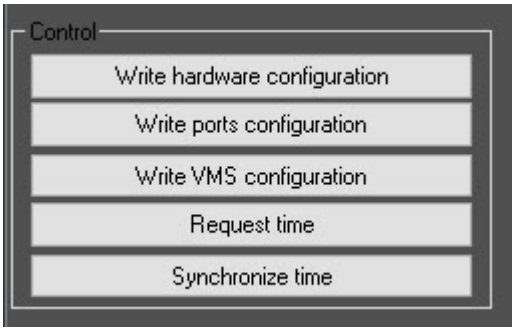
The *AIM SC* reader relay is now configured.

4.5 Write configuration to hardware

For the system with a central controller, the configuration is written to the *AAN* controller, for the system without a central controller—to the *AIM* module.

To write configuration to hardware, select the **Apollo AAN** or **Apollo AIM SC** object depending on the schema of the security system.

To write configuration to hardware, click the **Write hardware configuration** button.

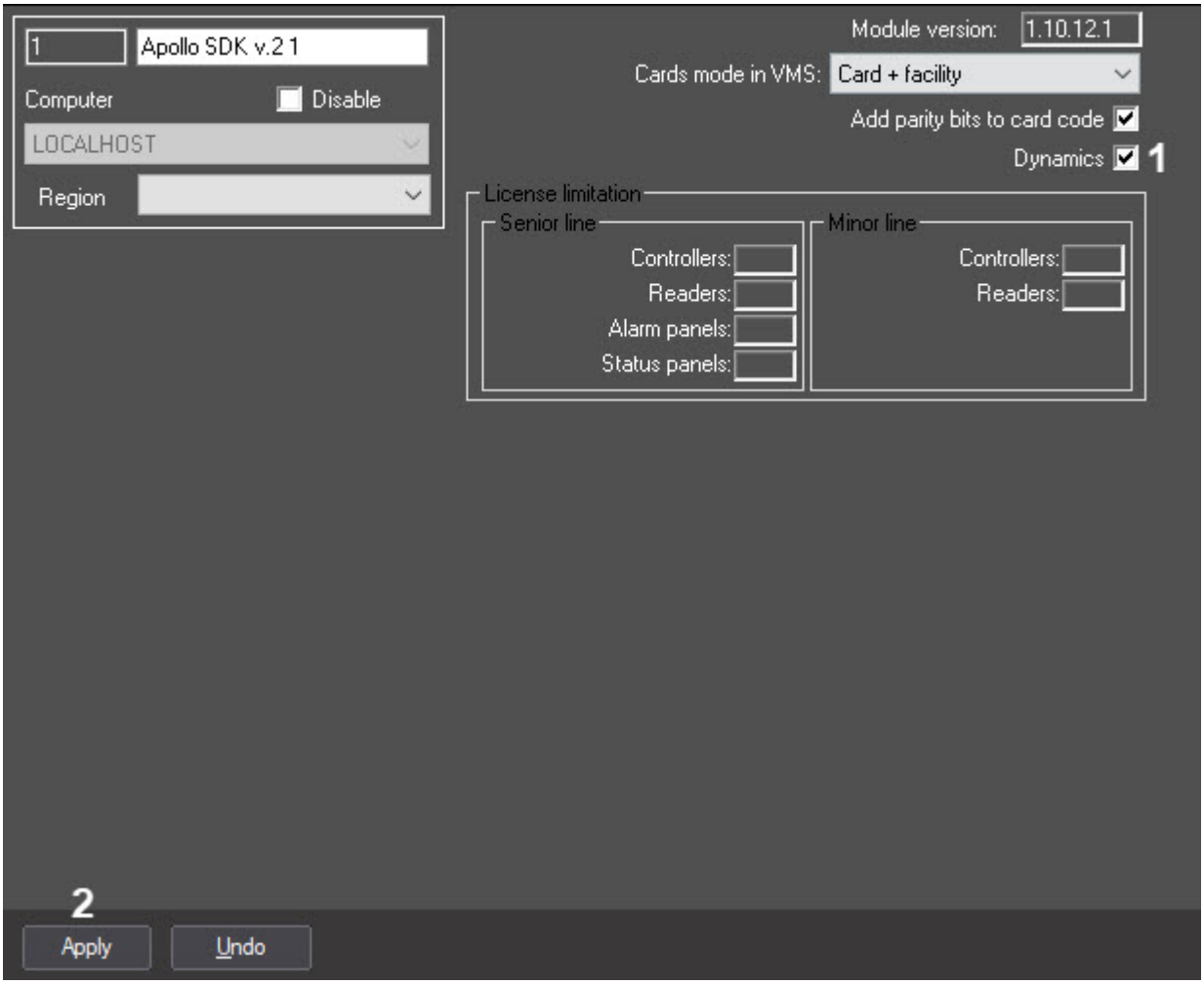


To write configuration of connection ports (only for the system with a central controller) for the *AIM* and *AIO* modules, click the **Write ports configuration** button.

To read the current time from the hardware (only for the system with a central controller), click the **Request time** button.

To write the current time of the *Axxon PSIM* Server to the hardware (only for the system with a central controller), click the **Synchronize time** button.

To write users, their cards and access levels to hardware, click the **Write VMS configuration** button. To write these data to hardware automatically, set the **Dynamics** checkbox (**1**) on the settings panel of the **Apollo SDK v.2** object and click the **Apply** button (**2**).



4.6 Configure supporting of access card formats

In *ACFA PSIM,* it is possible to configure the support for the formats of the required access cards.

For this, open the FormatsCard.xml file located in the <Directory of the *Axxon PSIM* software installation>\Modules, and specify the corresponding format parameters for the access card being added:

- BitsOnCard—number of bits on card;
- BitsForEven—number of bits for even check;
- BitsForOdd—number of bits for odd check;
- NumBitsInFC—number of bits in facility code;
- IndexFCBegin—index of facility code's begin;
- NumBitsInCardNumber—number of bits in card number;
- IndexCardNumberBegin—index of card number's begin.

Example of configuring the support for the **Wiegand26** and **Wiegand38** card formats:

```
<?xml version="1.0" encoding="UTF-8"?>
<FormatsCard>
  <Formats>
    <Format>
      <W_BitsOnCard>26</W_BitsOnCard>
      <W_BitsForEven>0</W_BitsForEven>
      <W_BitsForOdd>0</W_BitsForOdd>
      <W_NumBitsInFC>0</W_NumBitsInFC>
      <W_IndexFCBegin>0</W_IndexFCBegin>
      <W_NumBitsInCardNumber>26</W_NumBitsInCardNumber>
      <W_IndexCardNumberBegin>0</W_IndexCardNumberBegin>
    </Format>
    <Format>
      <W_BitsOnCard>38</W_BitsOnCard>
      <W_BitsForEven>19</W_BitsForEven>
      <W_BitsForOdd>19</W_BitsForOdd>
      <W_NumBitsInFC>0</W_NumBitsInFC>
      <W_IndexFCBegin>0</W_IndexFCBegin>
      <W_NumBitsInCardNumber>36</W_NumBitsInCardNumber>
      <W_IndexCardNumberBegin>1</W_IndexCardNumberBegin>
    </Format>
  </Formats>
</FormatsCard>
```

Note

The **Wiegand 26** and **Wiegand 38** card formats are supported by default.

Attention!

Maximum number of card formats being added is six. If more than six card formats have been added, the first six formats will be supported and other formats will be ignored.

4.7 Configure controlling the ApolloSDK reader by event in the system

Unlike other *ACFA PSIM* integration modules, it is not possible to control the *ApolloSDK* reader using macros. To control the reader in automatic mode, it is necessary to use scripts (for details, see Programming Guide (JScript)).

Below is an example of a script that, upon receiving an ACCESS_REQUEST event from an **AUTO access point** with id 2.1, sends a command for access granting to the **Apollo AIM reader** with id 1.1.1.1.1.1:

```
if (Event.SourceType == "VIRTUAL_AUTO_AP" && Event.Action == "ACCESS_REQUEST" &&
Event.SourceId == "2.1") {
    var id = Event.GetParam("param1");
```

```
    NotifyEventStr(Event.SourceType, Event.SourceId, "OP_GRANTED", "person_id<"+id+">
");
    DoReactStr("APL2_AIM_READER","1.1.1.1.1.1","EMULATE_ACCESS","person_id<"+id+">");
}
```

5 Working with the ApolloSDK integration module

5.1 General information about working with the ApolloSDK integration module

The following interface objects are used to work with the *ApolloSDK* integration module:

1. **Map**.
2. **Event Viewer**.
3. **Access Manager**.
4. **Event Manager**.

The information on how to configure these interface objects can be found in the following *Axxon PSIM* documents: Administrator's Guide, Access Manager Module Settings and Operation Guide and Event Manager Module Settings and Operation Guide.

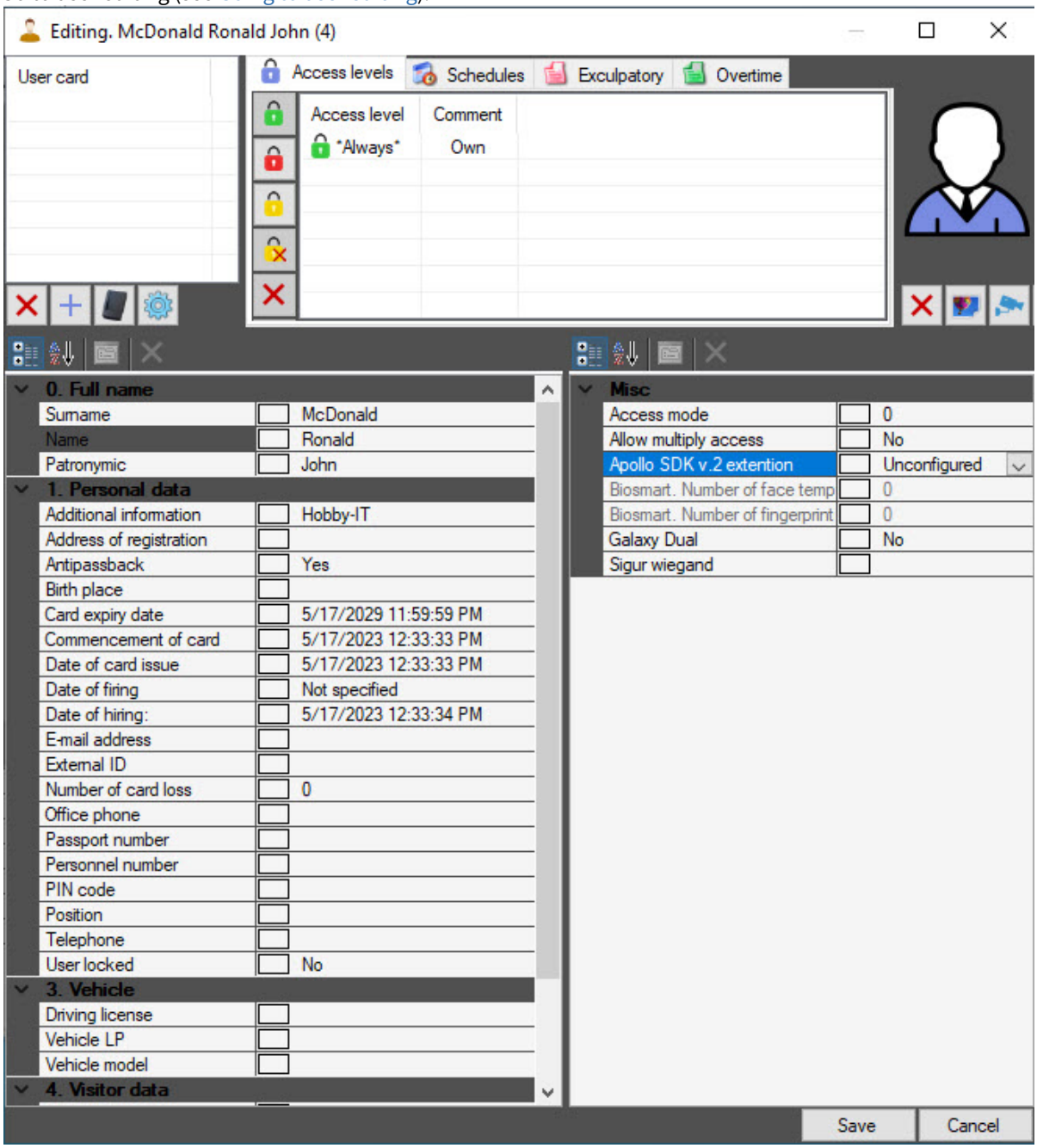
For the information on how to work with interface objects, see *Axxon PSIM:* Operator's Guide.

5.2 Additional user settings in Apollo SDK

The *Apollo SDK* integration module provides the ability to specify individual additional user settings in the *Access Manager* window (for more information about this module, see the Access Manager Module Settings and Operation Guide).

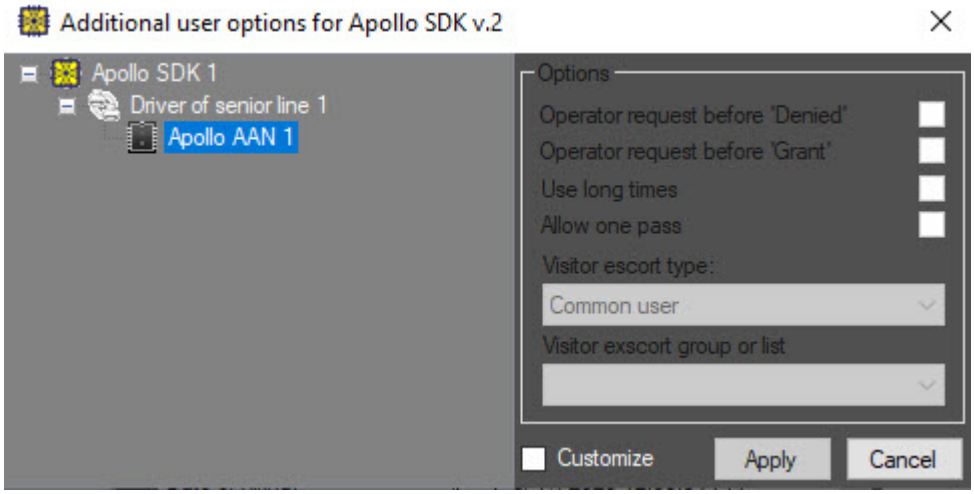
Additional user parameters are configured as follows:

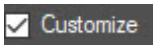
1. Go to user editing (see Going to user editing).

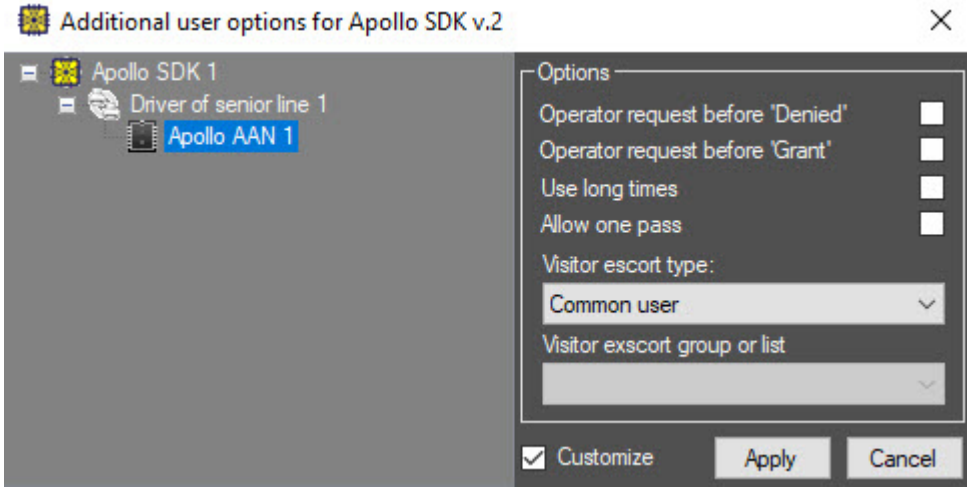


2. Click the button in the **Apollo SDK v.2 extension** field. The **Additional user options for Apollo SDK v.2** dialog box opens.

3. Select the AAN controller in the tree.



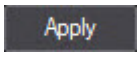
4. Set the **Customize**  checkbox. After that, the parameters of the **Apollo AAN** object can be set.



5. If operator request should be send before user access denial, set the **Operator request before "Denied"** checkbox. If operator request should be send before user access granting, set the **Operator request before "Grant"** checkbox.
6. If prolonged time interval for lock opening for this user is required, set the **Use long times** checkbox.
7. If user is allowed to pass first, set the **Allow one pass** checkbox.
8. In the **Visitor escort type** drop-down list select:

Escort type	Comment
Common user	A user who can access through the Apollo readers in accordance with assigned access levels
Visitor	A visitor who can access through the Apollo readers in accordance with assigned access levels without escort
Escorter	A user who have to escort a group of visitors assigned to a corresponding group of visitors

Escorted	A visitor who can access through the Apollo readers in accordance with assigned access levels but must access together with an escorter. Events about the passage of visitors with escort come to the system after the passage of their escort

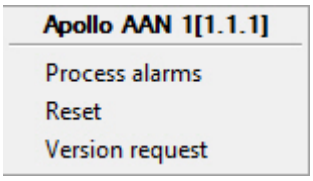
9. In the **Visitor escort group or list** drop-down list, select the group of visitors to which the user belongs, or the list of groups of visitors for which he is an escort (see also Configuring visitor groups and group lists).
10. Click the **Apply** button 

.

Setting up advanced user options in *Apollo SDK* is complete.

5.3 Control the AAN controller

Control the *AAN* controller is carried out in the **Map** interface window using the corresponding object's menu.

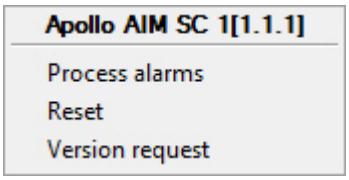


Description of the **Apollo AAN** object's menu commands is given in the table.

Command	Function
Process alarms	Processes alarm states
Reset	Reset of controller
Version request	Request for version of controller

5.4 Control the AIM SC controller

Control the *AIM SC* controller is carried out in the **Map** interface window using the corresponding object's menu.

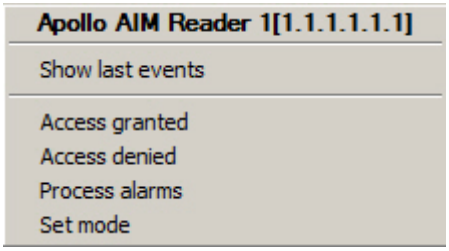


Description of the **Apollo AIM SC** object's menu commands is given in the table.

Command	Function
Process alarms	Processes alarm states
Reset	Reset of controller
Version request	Request for version of controller

5.5 Control the AIM interface module's readers

The *ApolloSDK* integration module's readers are managed in the interactive **Map** window using the **Apollo AIM Reader** object's menu.

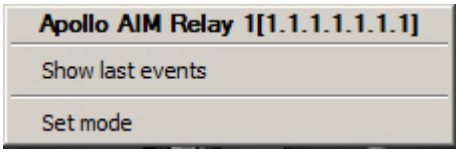


Description of the **Apollo AIM Reader** object's menu commands is given in the table.

Command	Function
Show last events	Standard *Axxon PSIM* command. Shows last 10 events (default) from the reader
Access granted	Grants access
Access denied	Denies access
Process alarms	Processes alarm states
Set mode	Selects the working mode of reader

5.6 Control relay of the AIM security panel

Control relay of the *AIM* security panel is carried out in the **Map** interface window using the function menu of the **Apollo AIM Relay** object.



To select the working mode of relay, select the **Set mode** command in the **Apollo AIM Relay** object.

The following modes are available:

Disabled	Relay is disabled
Enabled	Relay is enabled

Impulse	Short-term mode change to the opposite of the current one

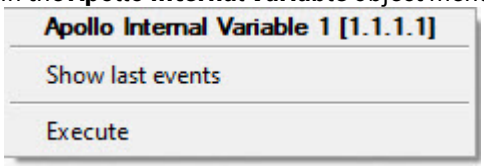
5.7 Control the Apollo internal variable

It is possible to run functions of an internal variable and set term values from the Map.

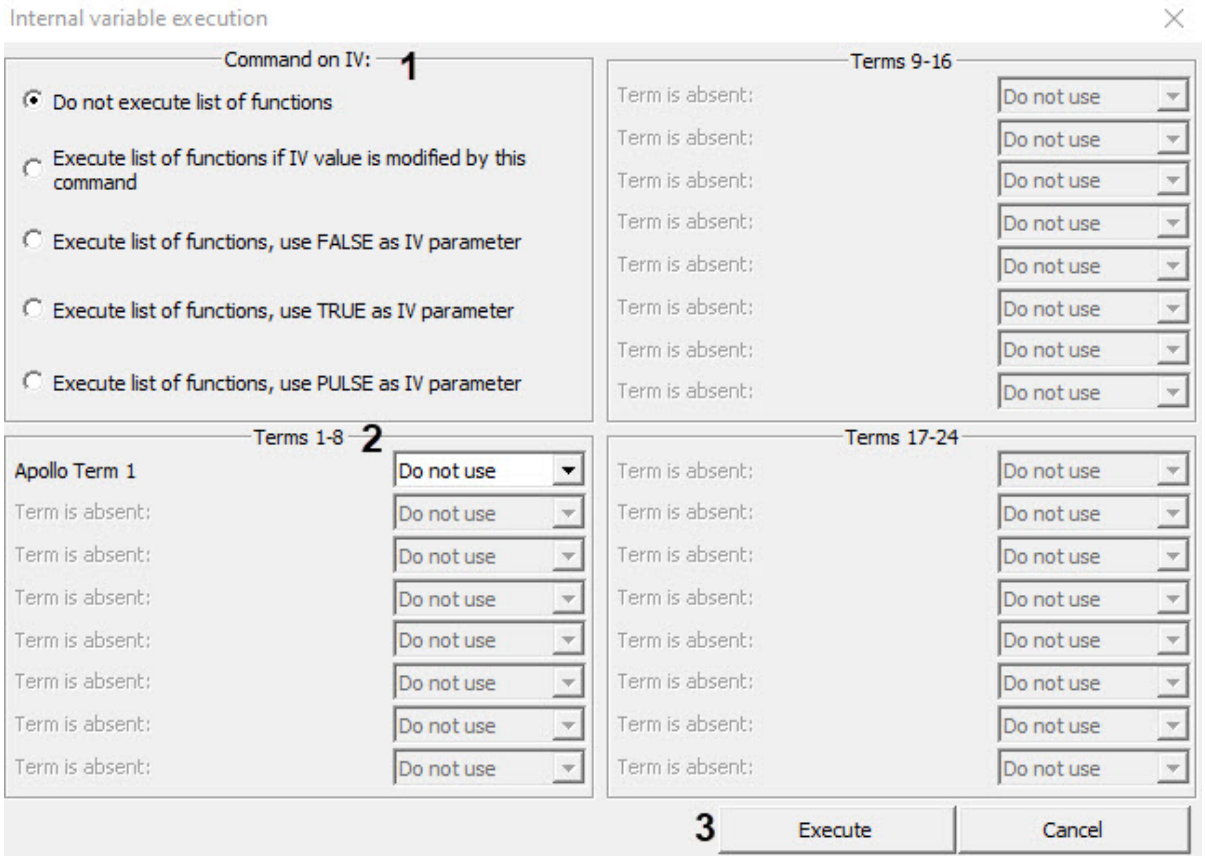
> **Note**
>
> See Configuring internal variables in ApolloSDK for more details on internal variables.

Functions of the internal variable are launched from the Map in the following way:

1. In the **Apollo Internal Variable** object menu, select **Execute**.



2. The **Internal variable execution** dialog box opens.



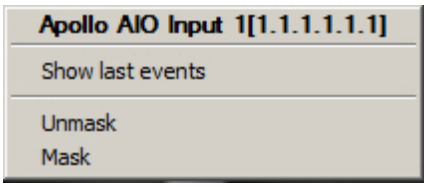
3. Set the switch to the position corresponding to the action that must be performed (**1**). If you only want to set the term values without performing the IV functions, set the switch to the **Do not execute list of functions** position.
4. From the drop-down lists corresponding to the terms, select the values that must be set to the terms of the internal variable (**2**).

> **Note**
>
> Only drop-down lists corresponding to terms created in the *ACFA PSIM* hardware tree are enabled (see Creating and configuring terms).
> If terms of an internal variable (addresses or count) were changed after *ACFA PSIM* was started, these changes will not be displayed in this dialog box even if configuration was send to the controller. Restart *ACFA PSIM* to use changed terms.

5. Click the **Execute** button (**3**).

5.8 Control the Apollo AIO Input

Control the input of the *AIO-168* security panel in the **Map** interface window using the function menu of the **Apollo AIO Input** object.



The commands are given in the table:

Command	Function
Show last events	Standard Axxon PSIM command. Shows last 10 events (default) from the input
Unmask	Cancel masking
Mask	Apply masking to input (so that events from the input do not show in Event Viewer

If the input is unmasked, its state is shown on Map as follows:

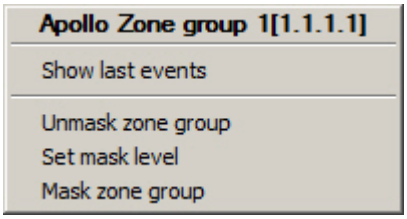
Green	Not active
Red	Active

If the input is masked, its state is shown on Map as follows:

Gray	Not active
Blue	Active

5.9 Control the zone group

The *ApolloSDK* integration module's zone groups are managed in the interactive **Map** window using the **Apollo Zone group** object's menu.

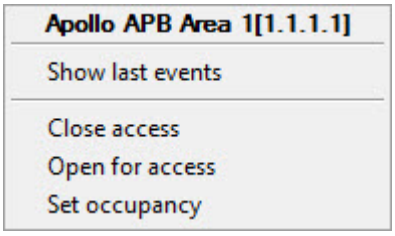


The commands are given in the table:

Command	Function
Show last events	Standard *Axxon PSIM* command. Shows last 10 events
Unmask zone group	Cancel masking
Set mask level	Allows operator to set masking level manually
Mask zone group	Applying mask to the zone group (so that events from the group are not shown in Event Viewer)

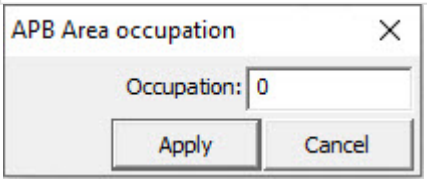
5.10 Control the APB Area

The APB Area control of the *ApolloSDK* integration module is performed in the **Map** interactive window using the **Apollo APB Area** object function menu.



The description of the Apollo AIM reader object's functional menu items is given in the table:

Menu item	Performed action
Close access	Decline the access
Open for access	Grant the access
Set occupancy	Set the maximum number of users in the APB Area



Note

The APB Area occupation value can not exceed the maximum number of persons in the zone, specified during the APB Area configuration (see Configuring APB Area). If you try to set a greater value, the error message will be displayed in the event log. In addition, the correct operation of this functionality is guaranteed only on controllers with the firmware version 3 and higher.