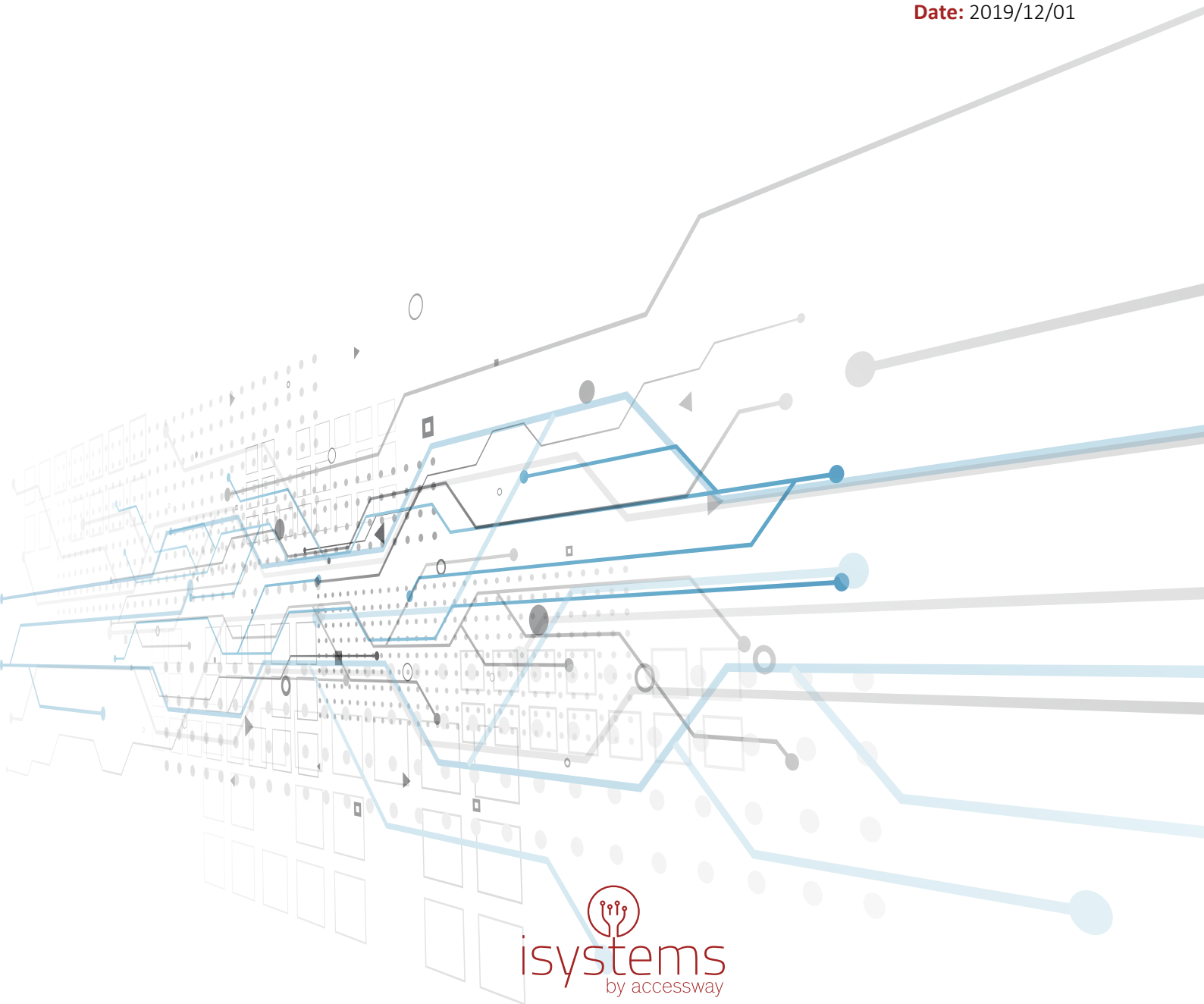
Integration Plugin

XProtect® Milestone Clients
SALTO Systems® Access Control

Installation and operation guide

Version: v1.2.007
Date: 2019/12/01



isystems
by accessway

Contents

Acronym and Abbreviations

MIP: Milestone Integration Platform

AC: Access Control

VMS: Video Management Software

SDK: Software Development Kit

API: Application Programming Interface

WCF: Windows Communication Foundation

IIS: Internet Information Services (web server)

Introduction

Salto is an AC system that provides a wide range of features to manage activity operation and localizations integration with regard to access control infrastructure.

As for Milestone XProtect, it is a high-performance monitoring system for video surveillance systems. This management platform includes the ability to integrate AC systems and consequent operation through its user interfaces (XProtect Management Client and XProtect Smart Client), through its MIP SDK, allowing in practice operators in the market with security systems the integration of their AC systems with the XProtect platform.

In this context, the plugin referred to in this document was developed with the purpose of providing access to the information present in the SALTO access control system by way of its integration or reading by other systems external to SALTO, in particular the Milestone XProtect platform. The main objective is to facilitate and make more efficient the operation of these systems, allowing its execution through a single operator interface, as opposed to the need to operate several interfaces in organizations with several systems, being one of them SALTO.

SALTO provides an API for external communication, which is the only way to communicate with it, otherwise it is a closed system.

In terms of macro blocks, the system as a whole can be described by the following picture:

There are essentially 3 workflows to consider:

a) Obtaining the initial configuration, done the way AC -> VMS in terms of passing information, but with requests for information executed by the plugin, which implies communication in the other direction. In this workflow, the transmitted information includes doors identification, types of events and cardholders;
b) Obtaining events (historic and real-time), carried out the way AC -> VMS, being that requests for events batches are performed by the plugin. In this workflow, the transmitted information includes events occurrence information, such as doors openings, intrusions, and so on;
c) Sending commands, carried out in the VMS -> AC direction. In this workflow, the plugin sends commands to the AC for, for example, opening doors.

The communication and information passing between the VMS (plugin) and the AC is never done directly, always passes through the communication service, which mediates all the interaction between both systems.

It should also be noted that this plugin has been developed viewing its use in systems of various dimensions, from large systems, with VMS distributed by several machines (eg, federated hierarchy) and managing hundreds of cameras (see Fig.1), to smaller systems with the VMS on a single machine managing a small number of cameras. In the latter case, it will even be possible to consider a single machine with both the AC and the VMS systems, although the desirable is a distributed installation, that is, each system on its machine. For the purpose of this manual, the latter configuration will be considered.

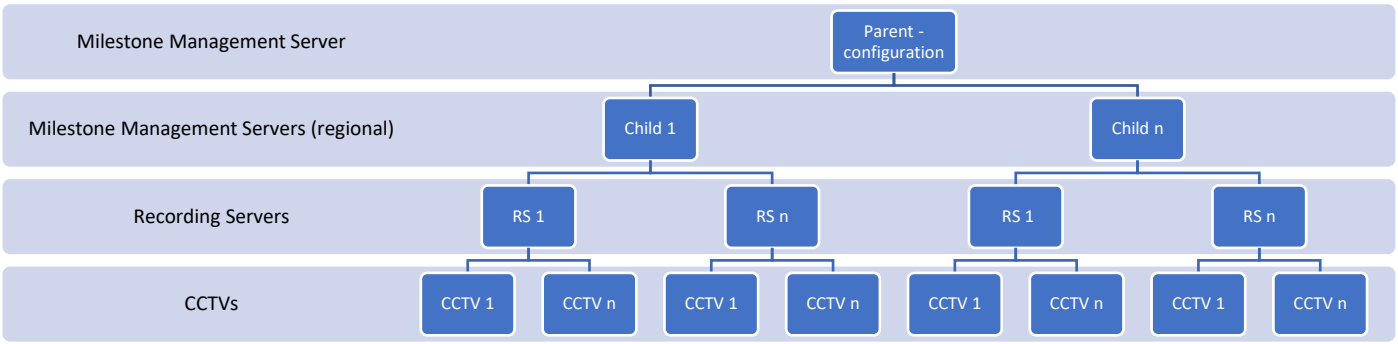


Fig.1: network infrastructure scheme of a VMS with a hierarchical structure of Milestone servers with cameras attached

Requirements

This integration system is composed of 3 components that communicate with each other: the AC Salto system, the Milestone VMS system and a WCF technology web service that mediates communication between both. As previously noted, for the purpose of this guide, each will be considered installed on its machine.

Typically, the Salto AC system and the Milestone VMS will be installed on different machines, located in different areas of an organization's network, so it will be in this case a distributed installation. It does not invalidate that the same principles cannot be applied to a local installation or to fewer machines. Figure 2 represents a distributed installation, which will serve as a reference for the information that will follow.

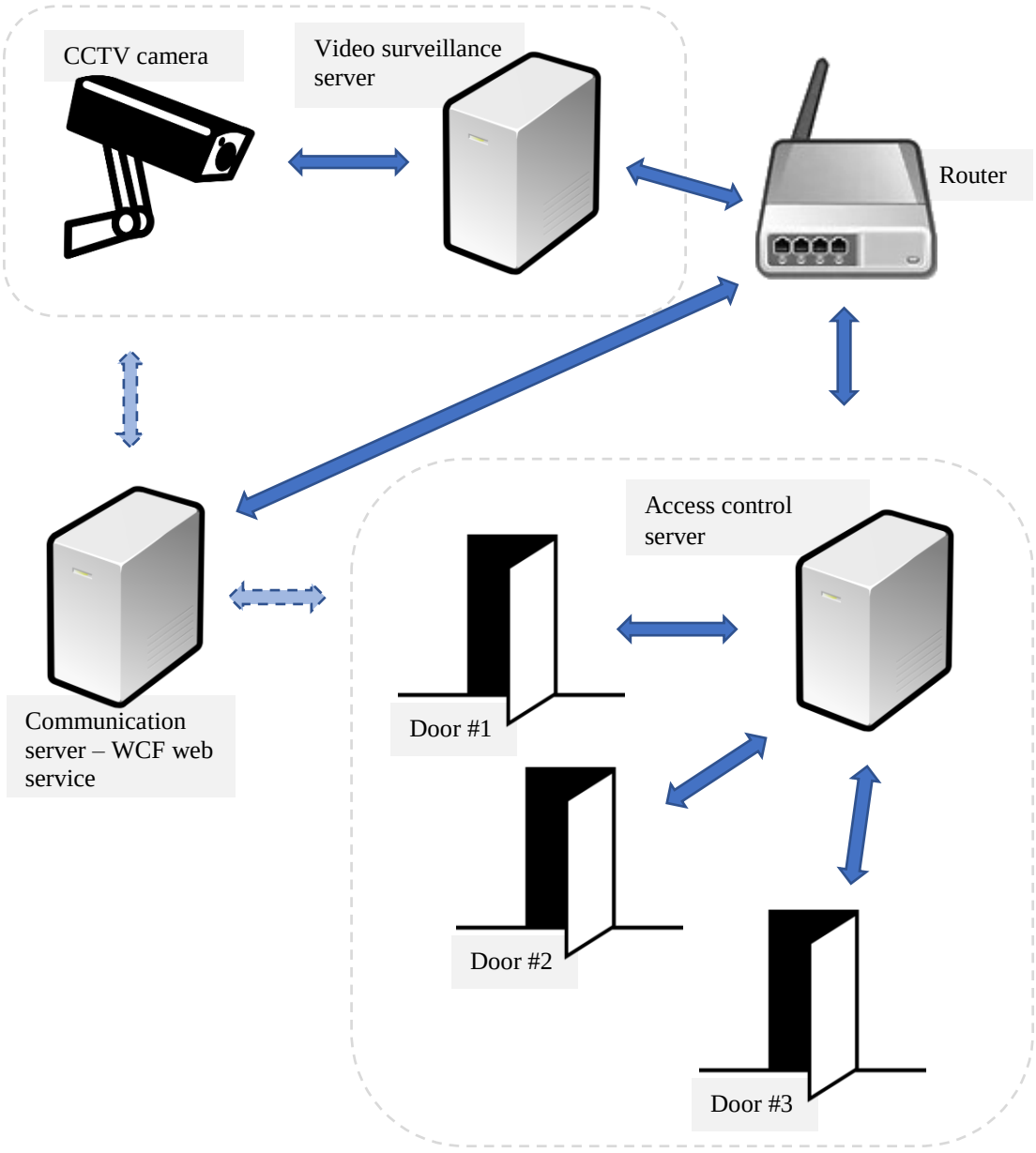


Fig.2: infrastructure example scheme with distributed installation of video surveillance system and access control system, assuming also the existence of a communication server for the plugin web service

The following is a list of the minimum requirements for integration of a distributed Salto – Milestone system:

- Win 7, 8+ | Win Server 2010 – 2012+ (64bit): on any of the machines
- Framework .NET 4.0.0 / 4.5.0: on any of the machines
- SQL Server 2012 (can be express version): on the WCF web service machine
    - SQL Server must be installed in “Mixed Mode”, so it allows the authentication of users created in SQL Server itself
- IIS 10.0: on the WCF web service machine
- Salto 3.1.4.0 | SHIP 1.26c: on the AC system machine
- Milestone XProtect Corporate 2014: on the VMS machine (http://www.milestonesys.com/systemrequirements)
- Milestone XProtect Smart Client 2016 R3: on the VMS machine
- MIPSDK 2016 R2: on the VMS machine

The machines are identified in the network through IPs (xxx.xxx.xxx.xxx, set of 4 integers with a maximum of 3 digits separated by ‘.’) and ports (also an integer). This is how they “see” each other and thus can communicate with each other. Considering the already described infrastructure distributed by 3 machines, the person in charge of the installation should also bear in mind as an installation requirement the following information:

- Milestone XProtect machine network IP address and port
- SALTO machine network IP address and port
- WCF web service communication machine network IP address and port

The plugin itself is to be installed on the VMS system machine. The plugin was implemented using Milestone’s MIP-AC technology, a specific SDK component for integration of AC systems. The plugin receives the configuration from the AC (doors, cardholders and event types), as well as events (past and realtime), and can send action commands on doors to Salto by way of SHIP.

The web service built in WCF architecture has, as indicated, the objective of serving as a Milestone – Salto communication interface, acting as a command queue, as well as storing the events locally in its own SQL database and synchronizing all the generated events by Salto / SHIP with all licensed and authorized points.

Finally, the Salto server, with its own database, closed behind the SHIP API / interface, accepts action commands on doors and sends, at the plugins request, the AC configuration information and the events stream.

It is also very important as a requirement that the clocks of the involved machines are in sync with each other, that is, all with the same time (by the second if possible).

Delivery Package Content

The contents of the installation package, with MIPSaltoPuresecurity base directory, are listed below:

- MIPSaltoPuresecurity
  - \MIPSDK2016R2
    - MIPSDK_Installer_2016R2.msi
  - \Plugin
    - acplugin.def
    - clsJSONZip.dll
    - MIPSALTOPuresecurity.dll
    - MIPSALTOPuresecurity.dll.config
    - Newtonsoft.Json.dll
  - \WebserviceWCF
    - isystemsSaltoShipWS.isystemsSaltoSHIP.svc
    - Web.config
  - \WebserviceWCF\bin
    - BLayer.dll
    - BLayer.pdb
    - isystemsSaltoShipWS.dll
    - isystemsSaltoShipWS.pdb
    - Newtonsoft.Json.dll
  - \WebserviceWCF\DB
    - 2017-05-22-isystems_Salto_MIP.bak
  - milestone_salto_plugin_guide_byPuresecurity.pdf

Plugin files description

acplugin.def: plugin general configuration file, required for the plugin recognition by the XProtect system

clsJSONZip.dll: JSON messages compression and decompression functions library, technology used in communication messages between the plugin, the communication service and the Salto AC (via SHIP)

MIPSALTOPuresecurity.dll: logic for the entire plugin specific operation

MIPSALTOPuresecurity.dll.config: plugin configuration xml file

Newtonsoft.Json.dll: public library of functions related to operations on JSON technology, such as “from” and “to” JSON conversions

WCF web service files description

isystemsSaltoShipWS.isystemsSaltoSHIP.svc: web service access file

Web.config: web service configuration file

bin\BLayer.dll e BLayer.pdb: communication service business rules

bin\isystemsSaltoShipWS.dll e isystemsSaltoShipWS.pdb: logic regarding the remaining web service specific operation, namely methods of access

bin\Newtonsoft.Json.dll: (see above)

DB\2017-05-22-isystems_Salto_MIP.bak: backup file to restore SQL Server database used for logging information by the web service

Plugin installation procedure

In this section of this guide, the necessary steps for installing the plugin and the communication WCF web service will be indicated. Installation is assumed on a global system in which the VMS and the AC are already installed.

For the following installation and configuration to take place, the support of the company's network administrator may be helpful.

WCF web service installation

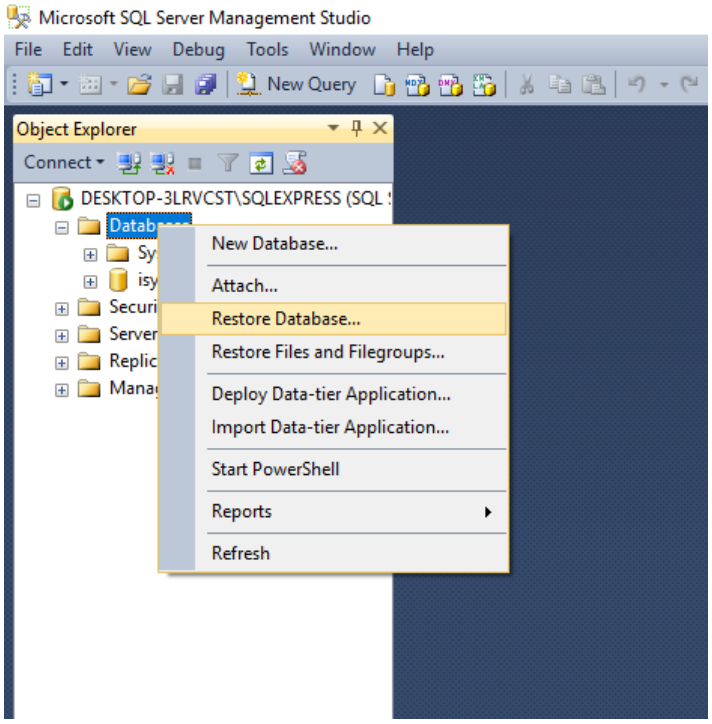
It is considered in the first phase the installation of the WCF web service. This is installed using the Microsoft SQL Server tools for database placement and IIS web server for web service instantiation with reference to the files in \WebserviceWCF and \WebserviceWCF\bin in the installation package.

The web service is installed on the machine destined for the communication web service and has as prerequisites, therefore, SQL Server and IIS.

Step 1 Dabase file restore

Restore 2017-05-22-isystems_Salto_MIP.bak file referring the database. To do this, open Microsoft SQL Server Management Studio database management tool.

1.1 Click on "databases" with the right mouse button -> "Restore Database…":



1.2 The dialog window for restoring a database will appear. In this, the “Device” option under “Source” must be selected, followed by click on the “…” button in the alignment.

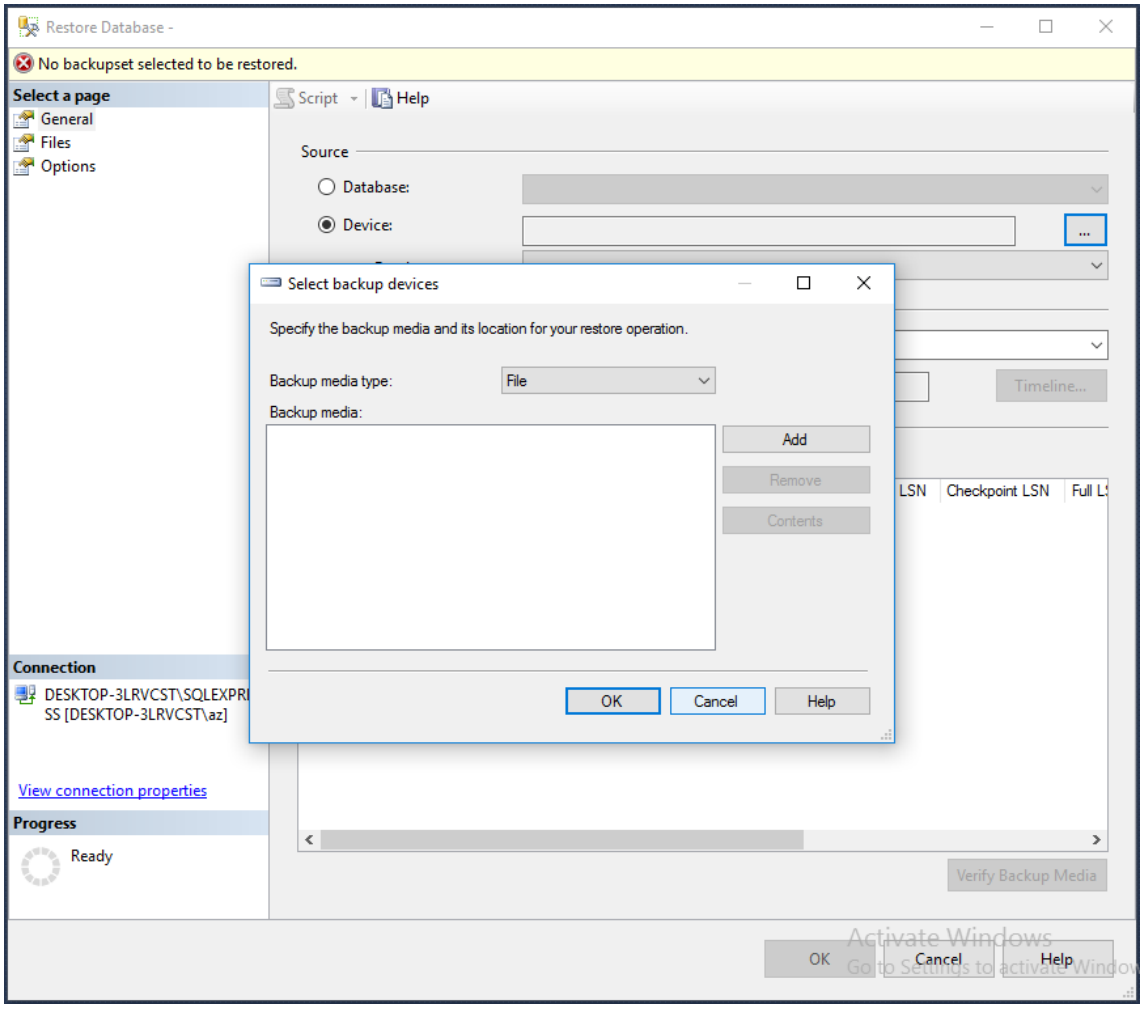
Another dialog window will appear that will be superimposed on the first one (“Select backup devices”) in which the following choices and actions must be made:

1.2.1 In “Backup media type” -> “File”

1.2.2 Click in “Add”

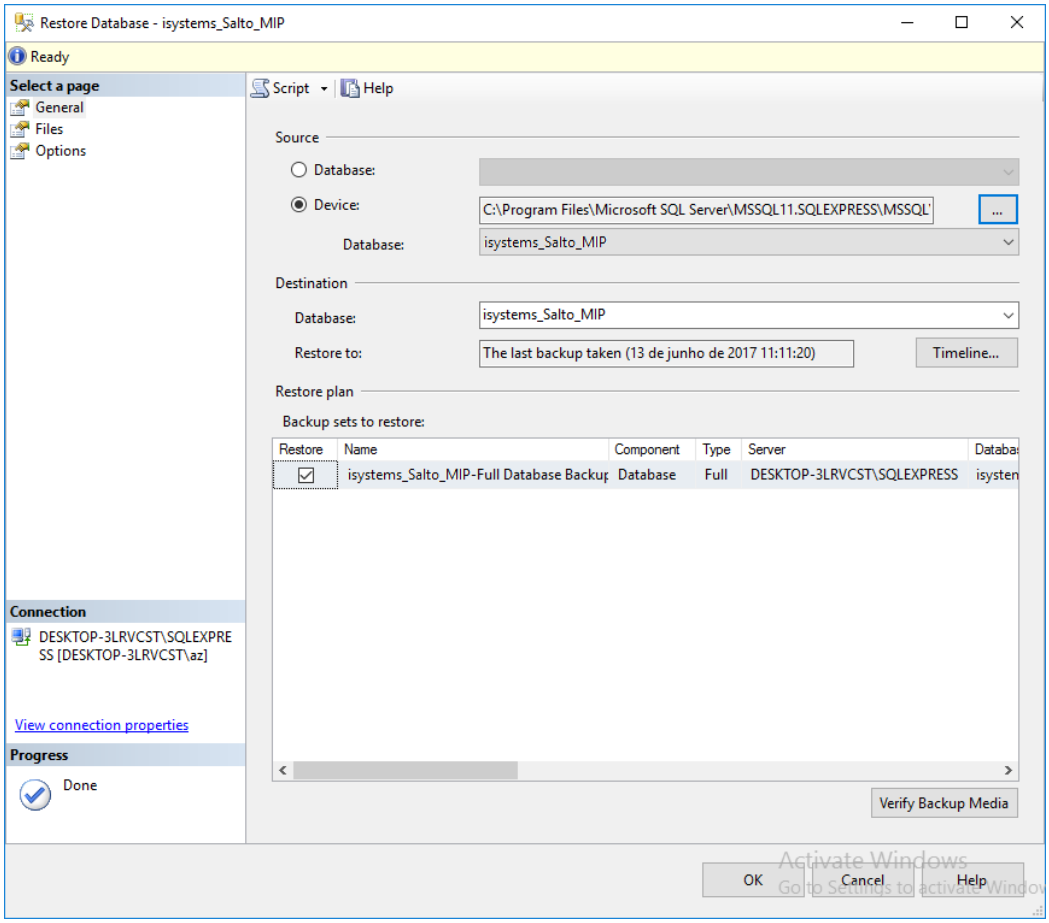
1.2.3 Select the .bak file provided in the installation files in \WebserviceWCF\DB (your choice will be displayed in the “Backup media” window)

1.2.4 Click in “OK”:

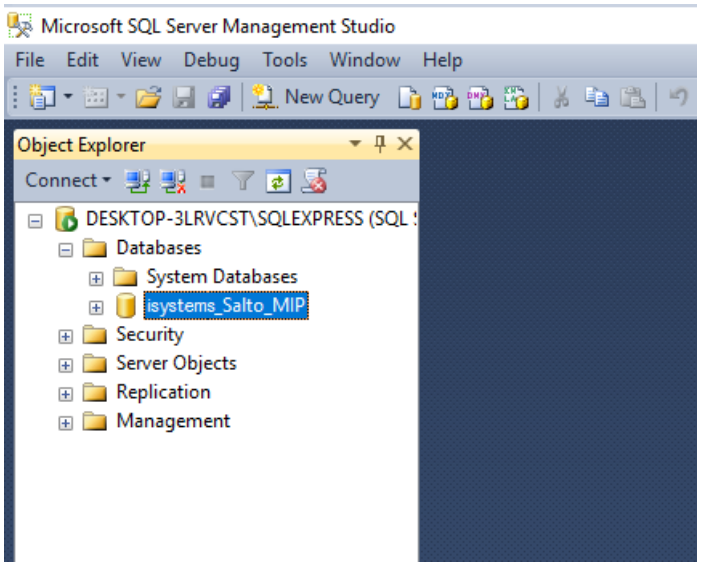


The result summary of the made choices will appear on the first dialog window in “Destination” e “Restore plan”, referring to the database file that will be restored.

1.2.5 Click in “OK” again:



The restored database will be placed in SQL Server Management Studio, under “Databases”:

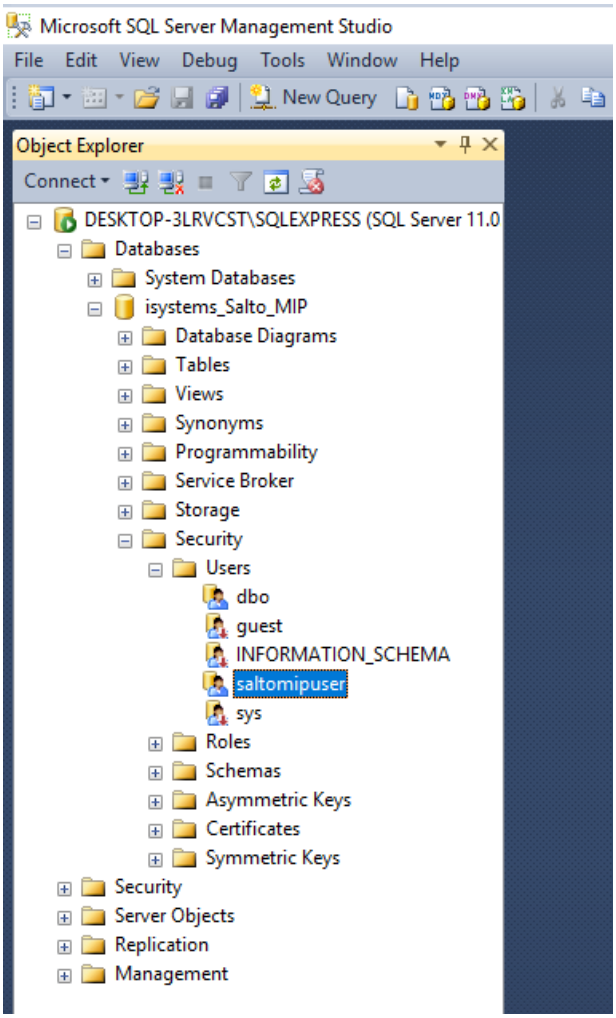


Step 2 'default' user removal

Remove user "saltomipuser" from restored database.

2.1 Check its existence by expanding the created database (default name: isystems_Salto_MIP) and click in "Security" -> "Users".

2.2 Remove the user by clicking in "saltomipuser" with the right mouse button, followed by "Delete".

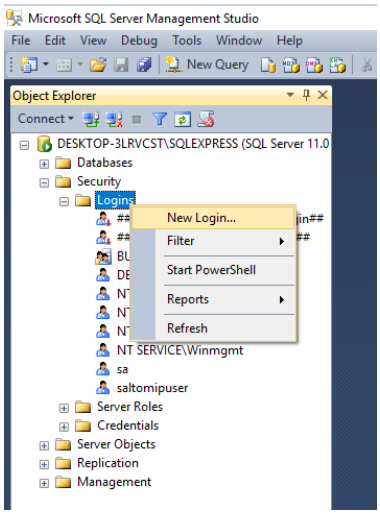


Step 3 Creation of a new user

Create user for database access.

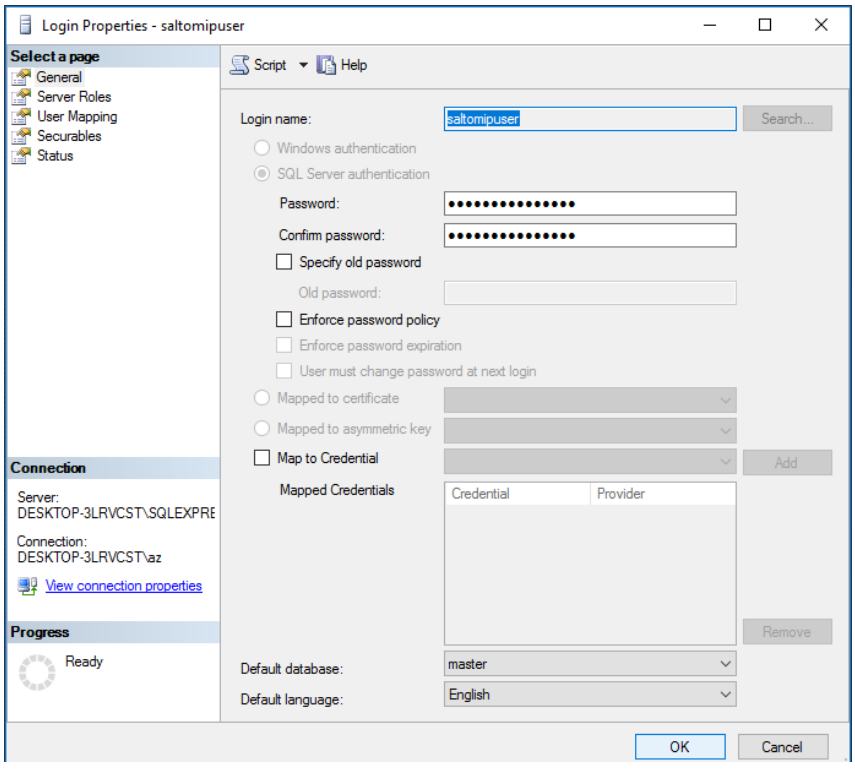
3.1 Expand the local SQL server in SQL Server Management Studio and click “Security” -> “Logins”.

Then, right click on “Logins” -> “New Login…”

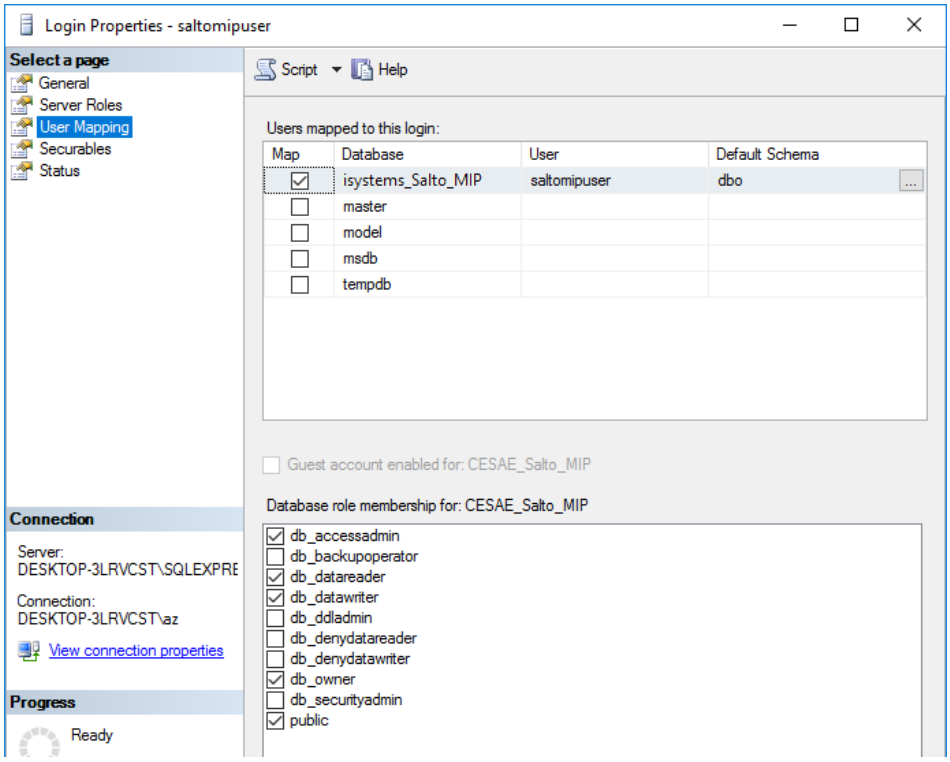


3.2 Fill in the required fields for user setup.

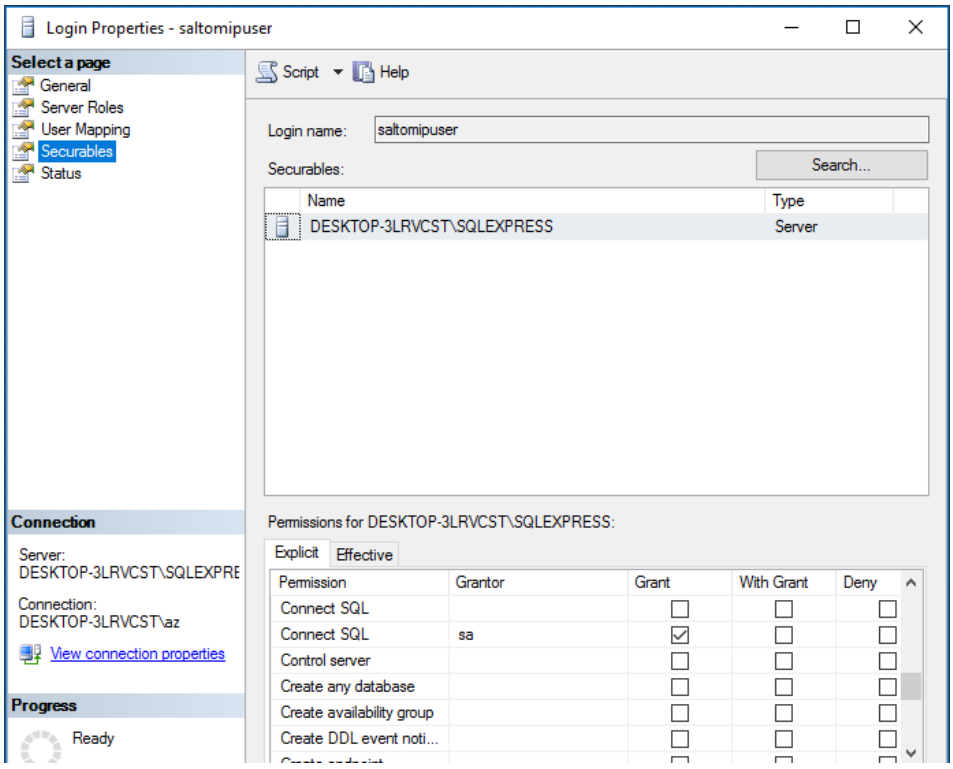
3.2.1 On the “General” tab, fill in “Login name” with "saltomipuser" and "Password"/"Confirm password" with the desired password:



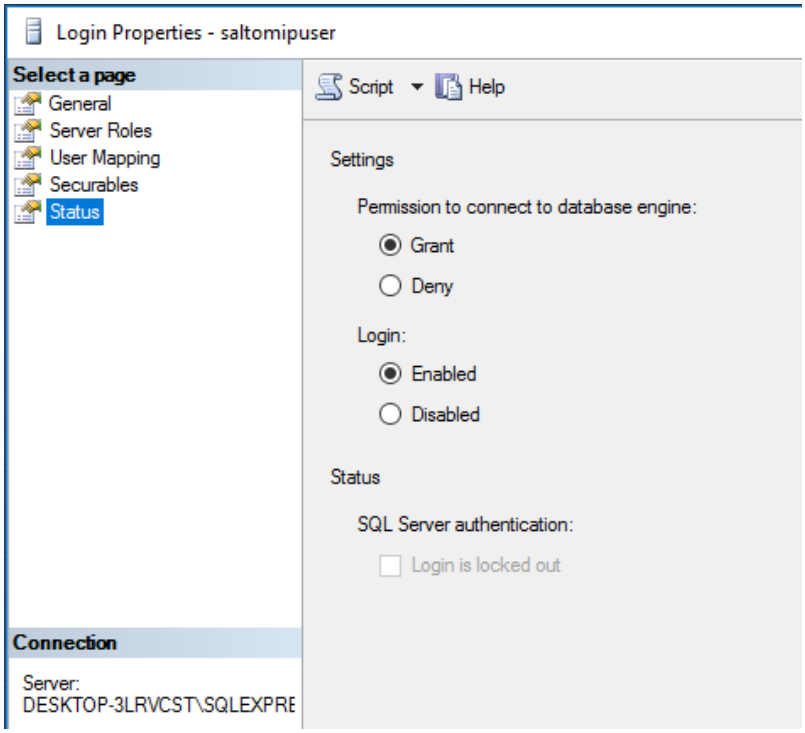
3.2.2 On the "User Mapping" tab, select the restored database in “Users mapped to this login” and the permissions marked on the image in “Database role membership”:



3.2.3 On the "Securables" tab, ensure that the "Grant" permission is selected in the “Connect SQL" option under “Permissions”:



3.2.4 On the "Status" tab, ensure the selection of "Grant" and "Enabled" as per the image, followed by “OK”:



As indicated previously in the requirements section, SQL Server must allow authentication of users created in SQL Server itself, as is the case of the user “saltomipuser”.

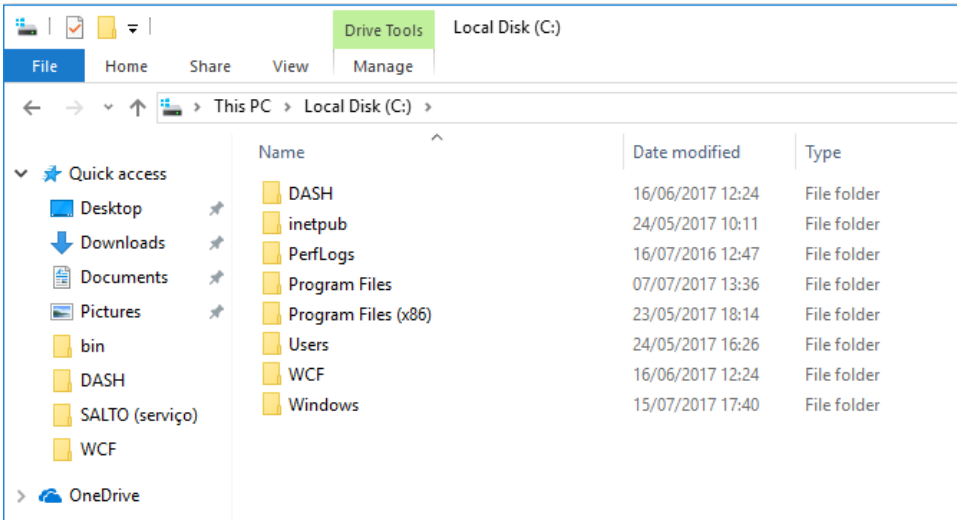
The way to test if this user, created to allow web service access to its database, really has access to this database, is to test its login through SQL Server Management Studio application. As such, after opening the application, click on ‘File’ -> ‘Connect Object Explorer’ and, in the following window, choose the name of the correct SQL Server, where the database is located, and ‘SQL Server Authentication’, and fill the ‘Login’ and ‘Password’ fields with “saltomipuser” and its password, respectively. After clicking ‘Connect’, there shouldn’t be any connection error and on the left side there must be an ‘Object Explorer’ window which allows access to the databases from the chosen SQL Server, including “isystems_Salto_MIP” database.

If some connection error occurs in this operation, this most likely means that the SQL Server is not allowing access to sql users and that is must have been wrongly installed in ‘Windows Authentication’ mode. In this case, this SQL Server configuration must be corrected before proceeding. This must be done, again, by means of the SQL Server Management Studio application. In the ‘Object Explorer’ window, normally located on the left side, right-click the intended SQL Server name, followed by click on ‘Properties’. In the following window, click the ‘Security’ tab and select ‘SQL Server and Windows Authentication mode’ under ‘Server Authentication’, followed by click in OK, and click in yet another OK button to confirm the change. Finally, right-click again the intended SQL Server name, followed by click in ‘Restart’. Restarting the SQL Server is needed so that the requested change becomes effective.

Step 4 Files installation

Install the WCF web service files.

4.1 Create folder in C:/ directory of the machine chosen to host the service. It may be in the root or in any subfolder that is convenient according to the folder structure organization there. As an example, in the following image it was given the name “WCF”.

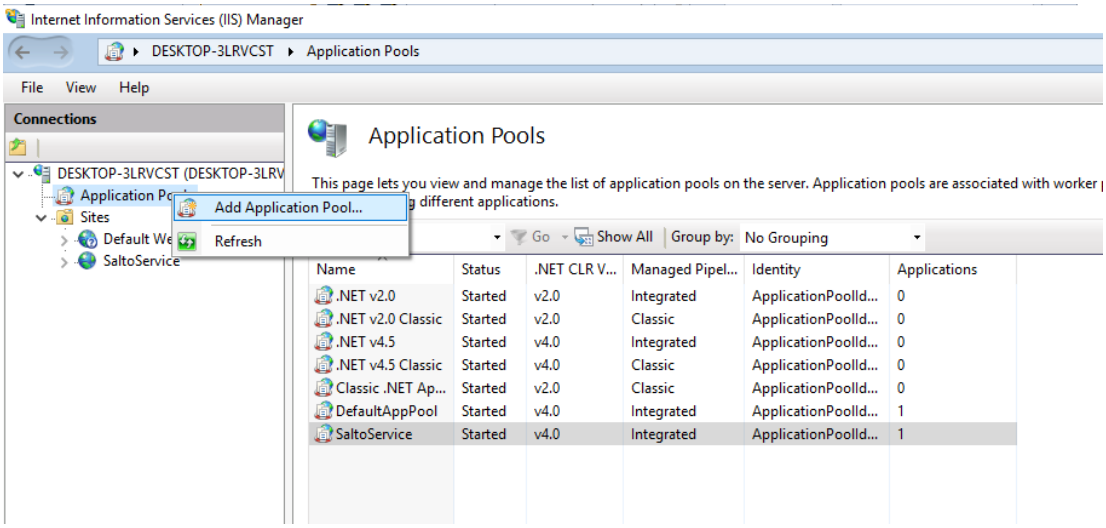


4.2 Copy WCF service files present in the installation bundle (those in the root of the \WebserviceWCF folder, including the \WebserviceWCF\bin subfolder and their files) to the folder created specifically for the purpose in the previous point.

4.3 Create the website for the web service in IIS, associated with the web service files placed in the “WCF” folder.

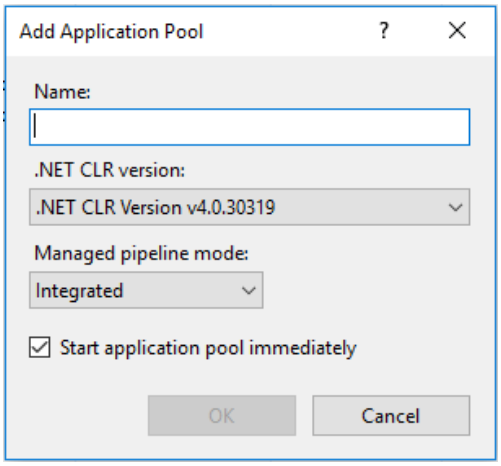
4.3.1 Open Windows IIS application through “Windows Administrative Tools” -> “Internet Information Services”

4.3.2 As a preparatory task, create an “application pool” for the website, that will later be created, through “<web server name>” (in the example below “DESKTOP-3LRVCST”) -> “Application Pools” -> “Add Application Pool”:

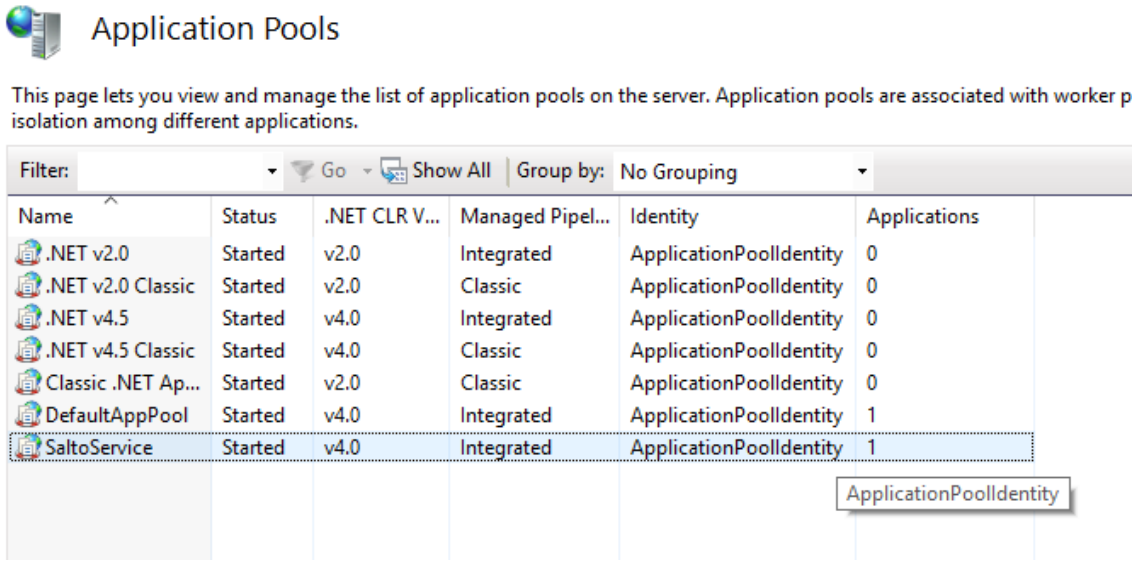


4.3.3 Fill in the fields necessary to create the “pool”, “Name” with “SaltoService” (this field will be on the criteria of the installer, it is not mandatory to be the provided example), “.NET CLR Version” with “.NET CLR Version v4.0.30319” and “Managed pipeline mode” with “Integrated”.

Click in “OK”.

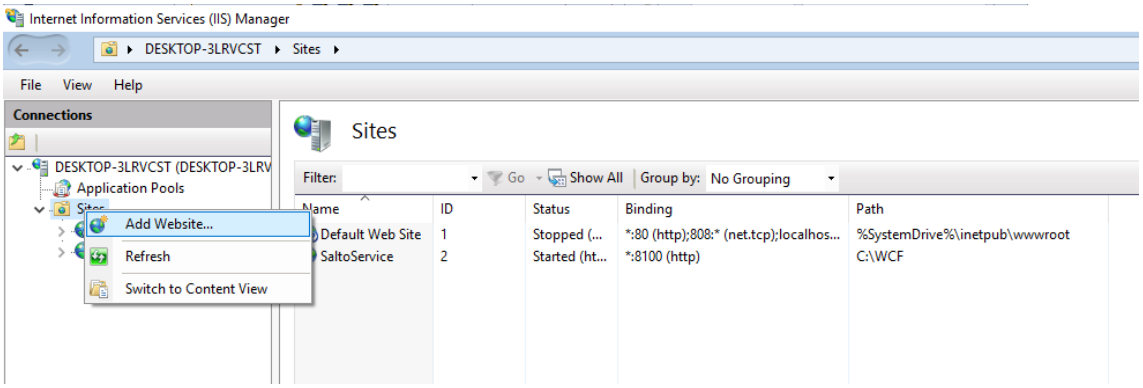


4.3.4 Verify that the "Identity" field of the created pool is "ApplicationPoolIdentity":



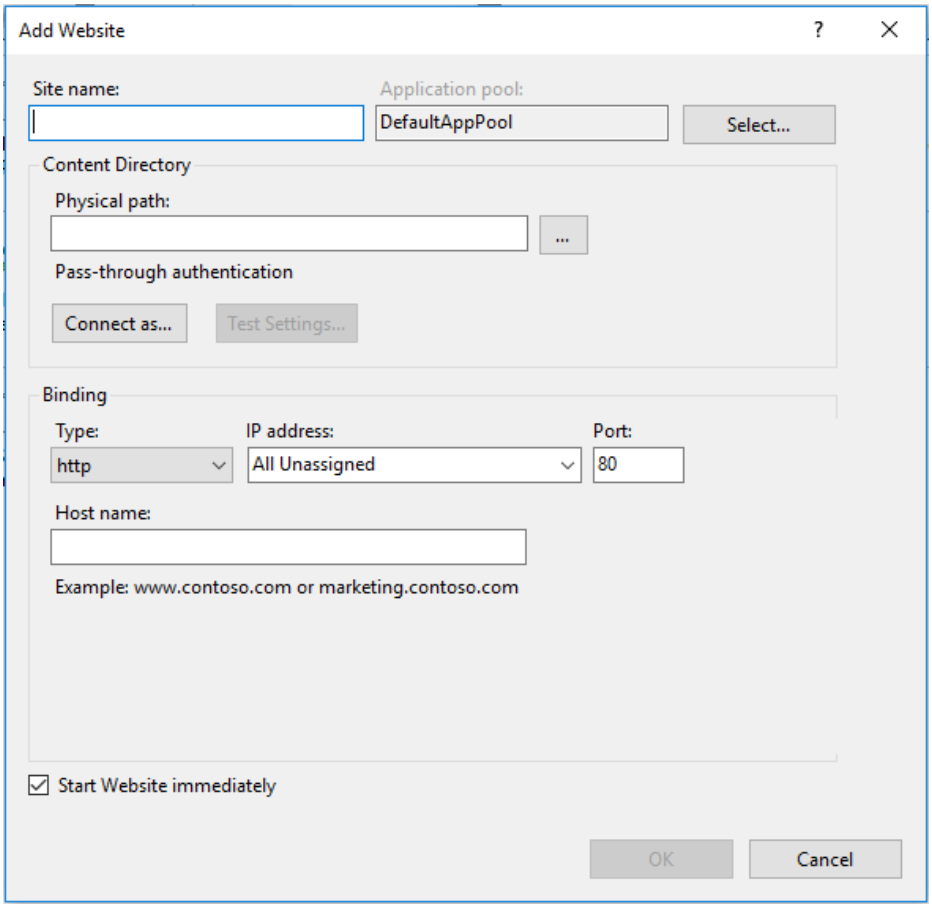
If this is not the case, then the advanced settings of the pool should be accessed by clicking the right mouse button on it, and adjusting the “Identity” field in the “Process Model” section by clicking “…”, followed by the appropriate configuration in “Built-in account”.

4.3.5 Create the website itself for the web service, through “<web server name>” -> “Sites” -> “Add Website”:



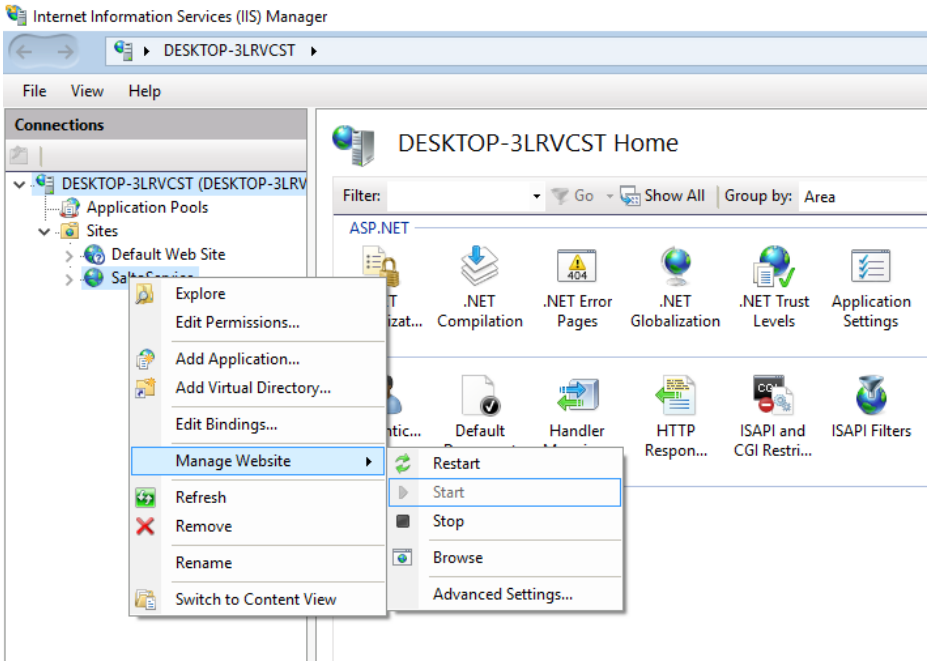
4.3.6 In the form that will subsequently open, fill in the fields "Site name" (for example, with “SaltoService”, the same name that was given to the pool created in 4.3.3), "Application pool" which should be filled automatically with the previously created pool (otherwise, this should be selected using the “Select… ” button by its side), "Physical path" with the path to the web service folder, selecting it via the “…” button (eg, “C:\WCF”), and "Host name" with "*:<web service port number>".

<web service port number> is, as already indicated in the requirements, an integer representative of the port from which the web service will respond on the machine in which it is placed (eg, 8100, 9000, ...). All systems that try to communicate with it, will have to have in their configuration this port identified (eg, the plugin).



The “Start Website immediately" checkbox can be left selected, because this way the web service will start immediately after clicking “OK”.

Otherwise, you will need to give start order later by clicking on “Sites” -> SaltoService" (right click) -> "Manage Website" -> "Start":



Step 5 Web service configuration

In order for the web service to be functional, a final configuration step is required. This is done configuring the xml file “Web.Config”, located in the web service files folder placed in the root of the machine (“C:\WCF” or another location in the machines folder structure according to the preference in the installation).

The “Web.Config” file can be opened with a simple text editor like "notepad". It is a simple xml/text file.

5.1 Database connection configuration

After opened, the following entry should be located on one of its lines: key="ADDRESSDB"

In this line the web service connection to its own database must be configured and, for this, the ‘value’ atribute must be updated.

Example: <add key="ADDRESSDB" value="server=192.168.254.40; database=isystems_SALTO_MIP; uid=saltomipuser; pwd=saltomipuser" />

This configuration line represents the web service connection to the previously configured database (step 1). The value attribute has in this case 4 sub-attributes (server, database, uid and pwd).

The database name (database=isystems_SALTO_MIP), having been followed the restore instructions provided in step 1, will be the same, so it doesn’t need updating, as well as the user name (uid=saltomipuser).

The database server IP will have to be configured (server=192.168.254.40), replacing the IP “192.168.254.40" (indicated by way of example) by the IP of the machine in which the web service database is placed, ideally, the same as the web service itself.

The password of the user used to access the database, previously created in this procedure (step 3), must also be configured, replacing the password "saltomipuser" with the password created when creating the user.

5.2 Configuration of communication with Salto AC system

Also remember that the web service serves as communication interface between the plugin (in VMS Milestone XProtect) and the Salto AC system, through its own SHIP interface. Therefore, the web service must have configured the connection to SHIP.

As in the database connection configuration seen above, the following entries should be located in the “Web.Config” configuration file:

key="ComandosIP" and key="ComandosPort"

Also in these cases the respective ‘value’ attributes must be updated.

Example:

<add key="ComandosIP" value="192.168.254.50" />

<add key="ComandosPort" value="9100" />

These two configuration lines represent the IP of the machine in which the Salto AC is located, namely the SHIP API, and the port of this machine that allows the entry of communication requests with SHIP.

Therefore, the IP of the machine where SHIP is located should be configured (value="192.168.254.50"), replacing the IP "192.168.254.50" (indicated by way of example) with the correct IP.

The access port to the machine where SHIP is located must also be set up (value="9100"), replacing the value "9100" (indicated by way of example) with the correct value.

5.3 Other configurations

One of the factors to consider when installing this integration is communications performance. In a distributed installation you can have the VMS Milestone system on one machine, the communication web service on another machine, and the AC Salto system on yet another machine. The exchange of information between these several elements generates traffic on the network over other traffic that may already exist. This can lead to network performance problems, which in turn affects the performance and smooth operation of communications between the integration components.

There are two lines in this web service configuration file relating to the configuration of the regularity with which the web service performs event synchronization and queries the command queue, which can be identified, respectively, by the entries: key="syncEventTimeIntervalSecs" e key="syncCmdQueueTimeIntervalSecs".

Example: <add key="syncEventTimeIntervalSecs" value="10" />

The ‘value’ attribute must be used to configure the timing of event synchronization. The lower the value, the greater the regularity of the event synchronization, that is, the events will appear faster and closer to real-time in the Milestone client.

Example: <add key="syncCmdQueueTimeIntervalSecs" value="10" />

The ‘value’ attribute must be used to configure the regularity of the command queue queries. The lower the value, the greater the regularity of the queue queries, that is, the commands over doors will run faster.

The following guidelines must be taken into account in this configuration:

- The configuration of these values should be done on a case-by-case basis and according to the performance verified after installation of the integration in any given network. If there is a good performance of the system in the network, the values can be reduced;

- The values must not be set to zero, a balance must be found between the functional performance by decreasing the values and the overall performance of the system in the network.

Whenever any configuration in this configuration file (web.config) is changed, the web service must be restarted by executing the command “Sites” -> SaltoService" (right mouse button) -> "Manage Website" -> "Restart" on the IIS web server.

Plugin installation

After the WCF web service installation, in the 2nd phase it is necessary to install the plugin itself.

This will be installed on the machine where Milestone XProtect VMS is located, since it was implemented using Milestone technology, precisely for the purpose of its incorporation in this type of systems.

The plugin installation requires 2 steps, placement of the plugin files (present in the “\Plugin” folder in the installation bundle) in a certain location of the Milestone XProtect folder structure, and its configuration in XProtect.

Step 1 Files installation

Placing the plugin files in the correct location.

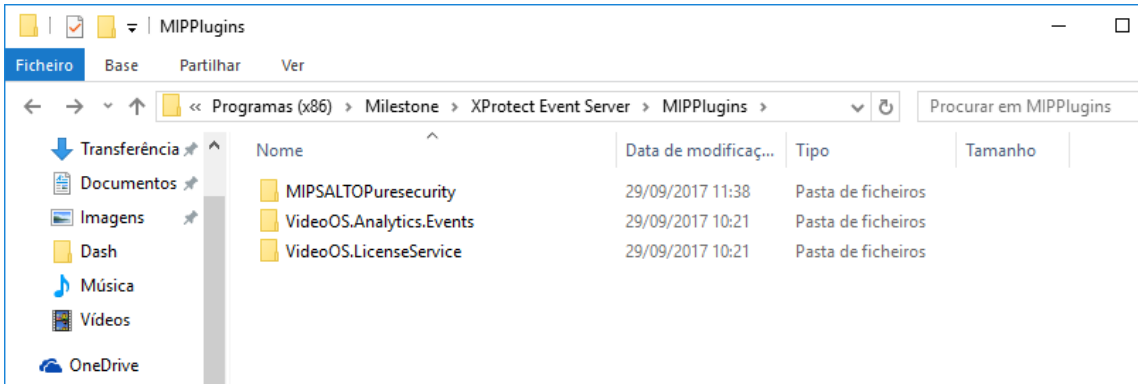
Regarding this step, one aspect that must be mentioned is that the plugin is strongly interconnected with the Milestone XProtect “event server”. This means that one of the, if not the main purpose of the plugin is to receive events in real-time from the AC system and register them on the VMS. Thus, it is natural that it be incorporated in XProtect in relation to the “event server”. Another aspect to mention is that the location of the plugin in XProtect may vary depending on the version of this software, for the purpose of this manual, it is the 2014 version.

1.1 Copy the plugin folder to the Milestone Event Server plugins folder.

To do this, the Event Server plugin folder must be located first, using Windows Explorer. As mentioned, in the 2014 version of XProtect software, this folder will be located in “C:\Program Files (x86)\Milestone\XProtect Event Server\MIPPlugins”.

After locating this folder in a Windows Explorer window, simply perform a copy/paste operation of the “\Plugin” folder located in the installation package into the open window located at the destination location.

The result will be what can be observed in the following figure. The “\Plugin” folder should be renamed to a meaningful name (eg, MIPSalto, PluginMIPSalto, and so on). As an example, in the figure the “\Plugin” folder has been renamed to “\MIPSALTOPuresecurity”.



1.2 Prepare the plugin configuration file.

Inside the plugin folder previously copied to the Milestone XProtect folder structure you will find a configuration file, “MIPSALTOPuresecurity.dll.config”. This file should be opened for editing and for this you can use Notepad, since it is a simple XML text file.

After opening, the lines marked in the following figure should be located:

These lines refer to communication web service “endpoint”, that is, in practice, contain the information that the plugin needs to invoke the web service. The endpoint “address” attribute should be updated as follows:

- Replace the “192.168.254.40” IP, given as an example, by the network IP address of the WCF web service on the communication machine;
- Replace the “8100” port, given as an example, by the WCF web service port on the communication machine.

Also in the same file, locate the lines marked in the following figure:

```
            </endpoint>
        </client>
    </system.serviceModel>
    <system.web>
      <identity impersonate="true" userName="DESKTOP-3LRVCST\edp1" password="edpmilestone"/>
    </system.web>
</configuration>
```

They are located at the end of the file, just below the “endpoint” definition.

This line contains information about the user used to login to the web service machine, in the “identity” definition. In this, the attributes “userName” and “password” must be updated for the name of the user, and respective password, to be used to access the web service machine.

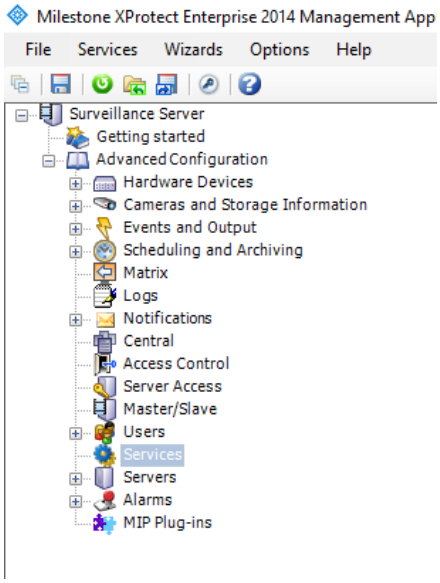
Step 2 Plugin configuration

Plugin configuration on the Milestone XProtect software.

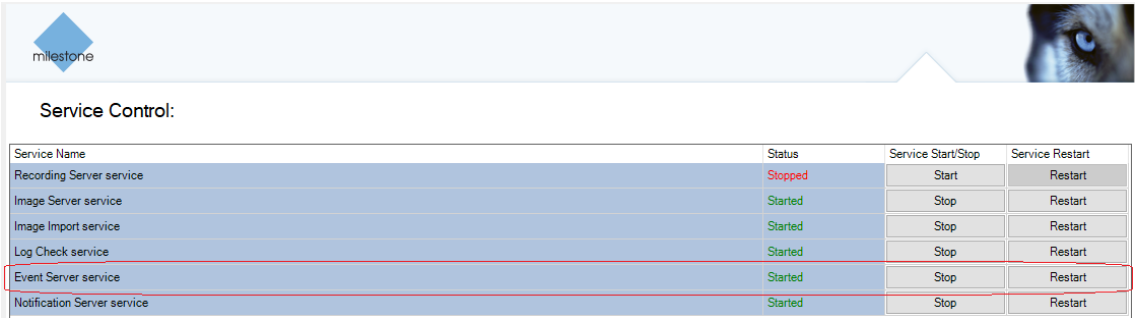
2.1 Register the new plugin in XProtect.

After placing the plugin files in the correct location, XProtect will need to recognize the plugin. To do this, you must restart the XProtect Event Server service.

Open the XProtect “management client”, this software management application, typically placed in the Windows applications structure under “Milestone” -> “XProtect Management Application”. After this application is opened, in the options tree on the left, the “Advanced Configuration” option should be extended, followed by click in “Services”.



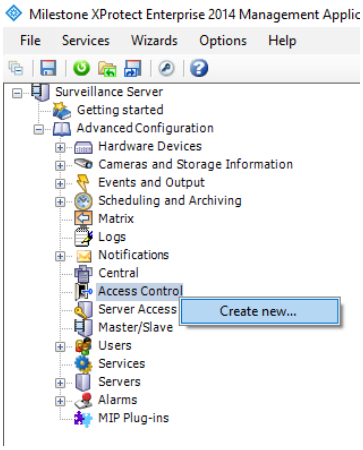
On the right-hand side will appear the management actions regarding XProtect services. The service to be restarted is the “Event Server service”, as shown in the following figure. The restart action can be performed by clicking the “Restart” button or by clicking “Stop” followed by “Start” (after the status of the service is ‘stopped’).



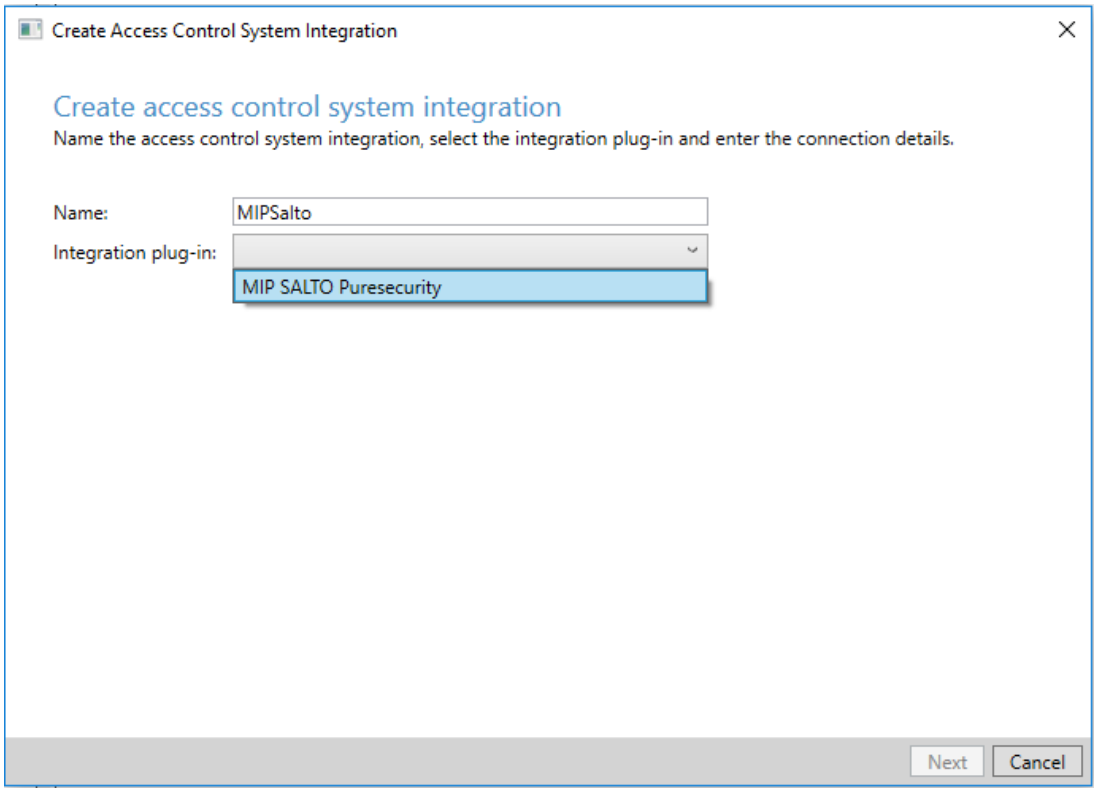
This step will allow the following steps to be performed. Otherwise it wouldn't be possible.

2.2 Create an instance of the plugin.

2.2.1 After the plugin registration by the VMS system, the next step is to make it visible in the XProtect interface, thus, create an instance. To do this, right click the "Access Control" option in the options tree and then click "Create new…".



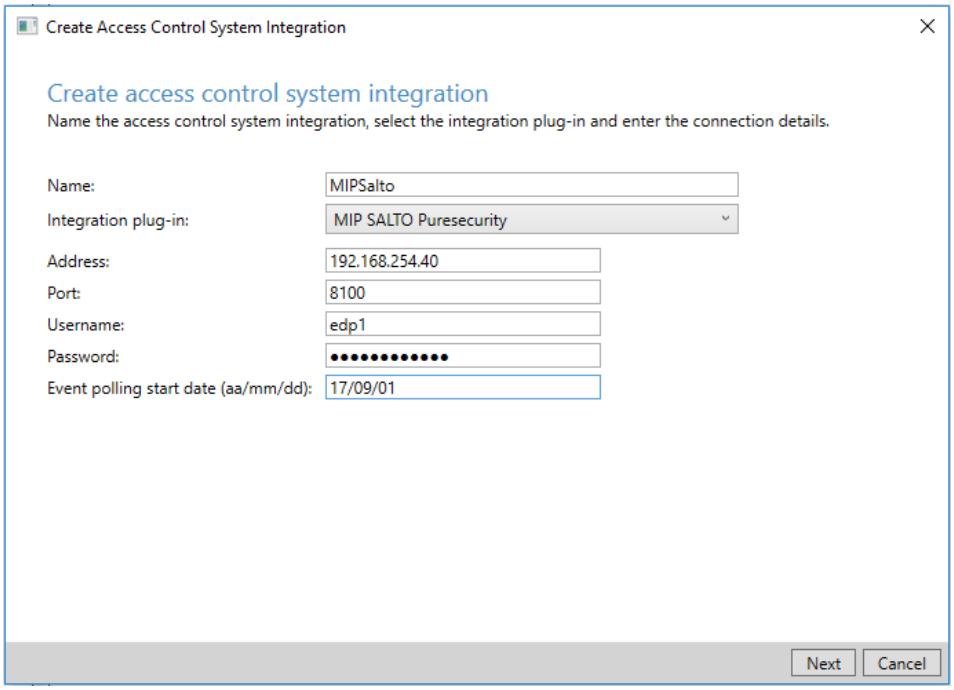
2.2.2 Fill in the fields in the first window of the plugin's creation wizard which, meanwhile, will have appeared on the screen, "Name" with some meaningful designation for the plugin (in the figure exemple, "MIPSalto"), and choose from the list of plugins "Integration plug-in" the one you want to instantiate, in this case, "MIP SALTO Puresecurity", followed by clicking "Next" button:

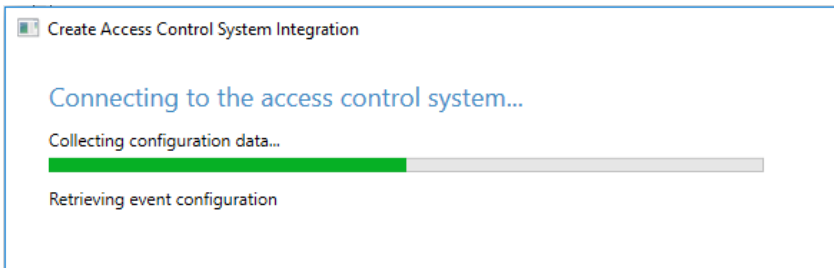


2.2.3 The second part of the plugin creation wizard will appear, in which all the additional fields should be filled, “Address” with the IP address of the WCF web service machine (in the figure example, 192.168.254.40), “Port” with the port from which the web service in the respective machine responds (in the figure example, 8100), “Username” with the user name for authentication in the web service machine (in the figure example, edp1) and its password, and, finally, “Event polling start date”.

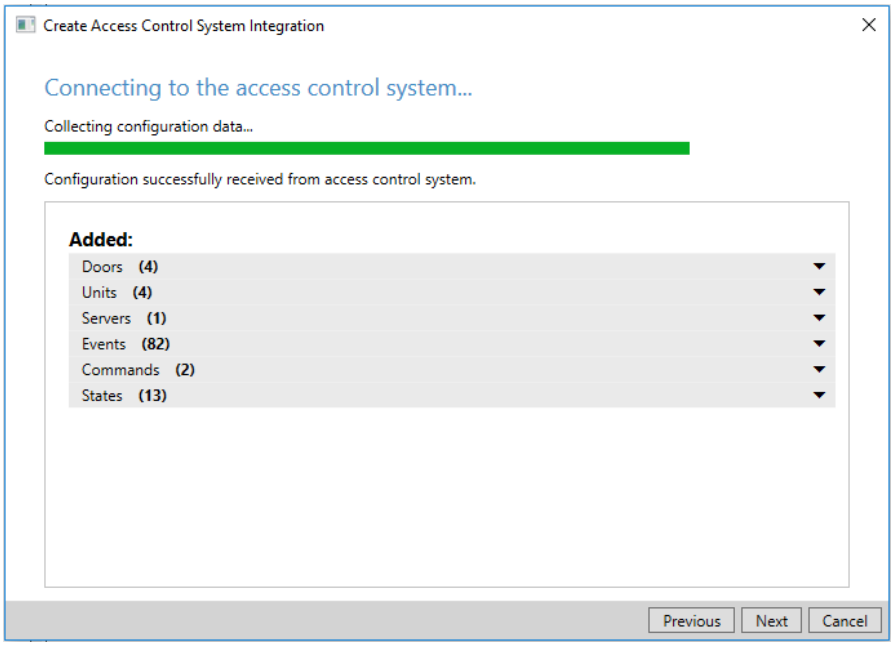
As for the later, a separate paragraph to explain: as indicate earlier, the plugin has an essential component for obtaining AC events in real-time. However, the moment from which to start getting events can be configured at this point, so that the plugin has the option to get not only the events in the immediate moment, but rather “go to the past” to get events in the AC historic from the defined moment to the current moment, from which it will engage in real-time fetching. This last wizard field is used to indicate the “backwards” date from which to obtain this event historic, in the format “aa/mm/dd” (2 digits for year, followed by 2 digits for month and 2 digits for day, all these pairs of digits separated by slash). In the example of the figure that follows, it is indicated the pretension of obtaining all events from September 1 2017 on.

This step will continue with click in the “Next” button, after which the automatic phase of the wizard will be entered, signaled by the “Connecting to the access control system…” title and a green progress bar (as can be observed below in the picture after the next one).





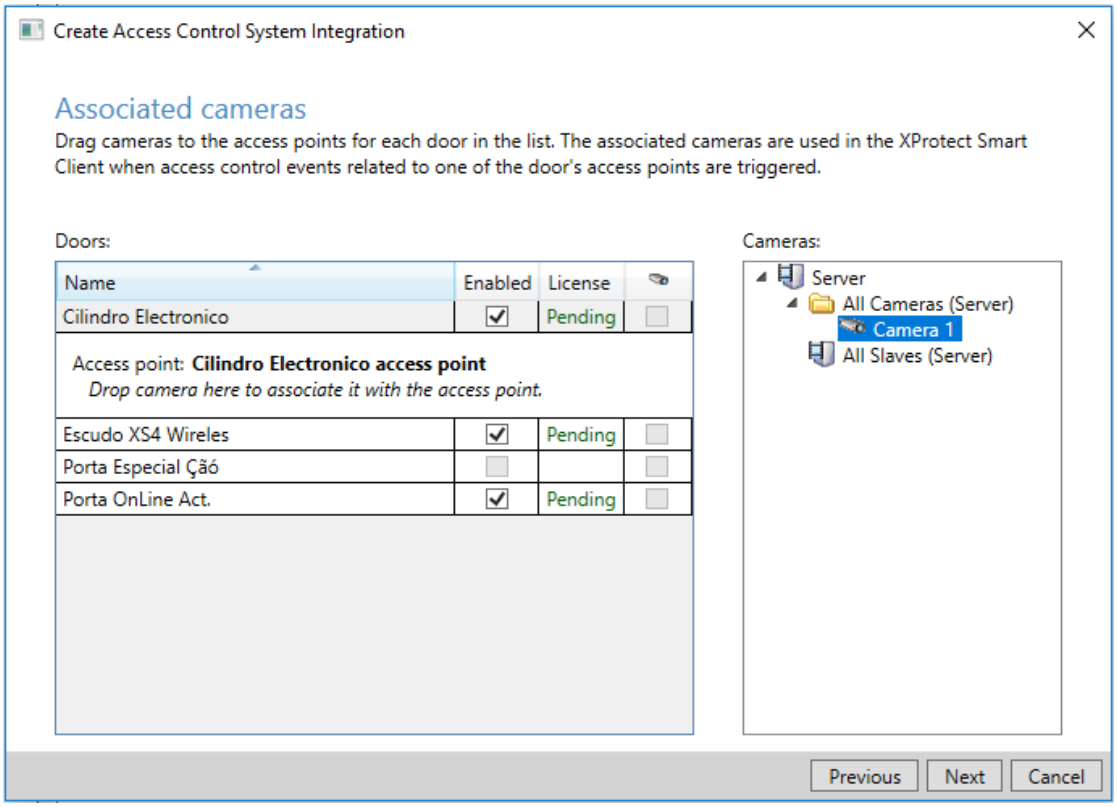
If the previous procedure is executed correctly, the result should be a screen similar to the following figure, in which you can find a summary of the information obtained by the plugin when integrating with the AC. Of greater relevance, information regarding AC doors ("Doors"), event types ("Events"), allowed command types ("Commands") and door states ("States").



The previous screen can serve immediately for the operator/administrator to evaluate the information of the integrated AC received by the plugin.

To proceed, click on the “Next” button.

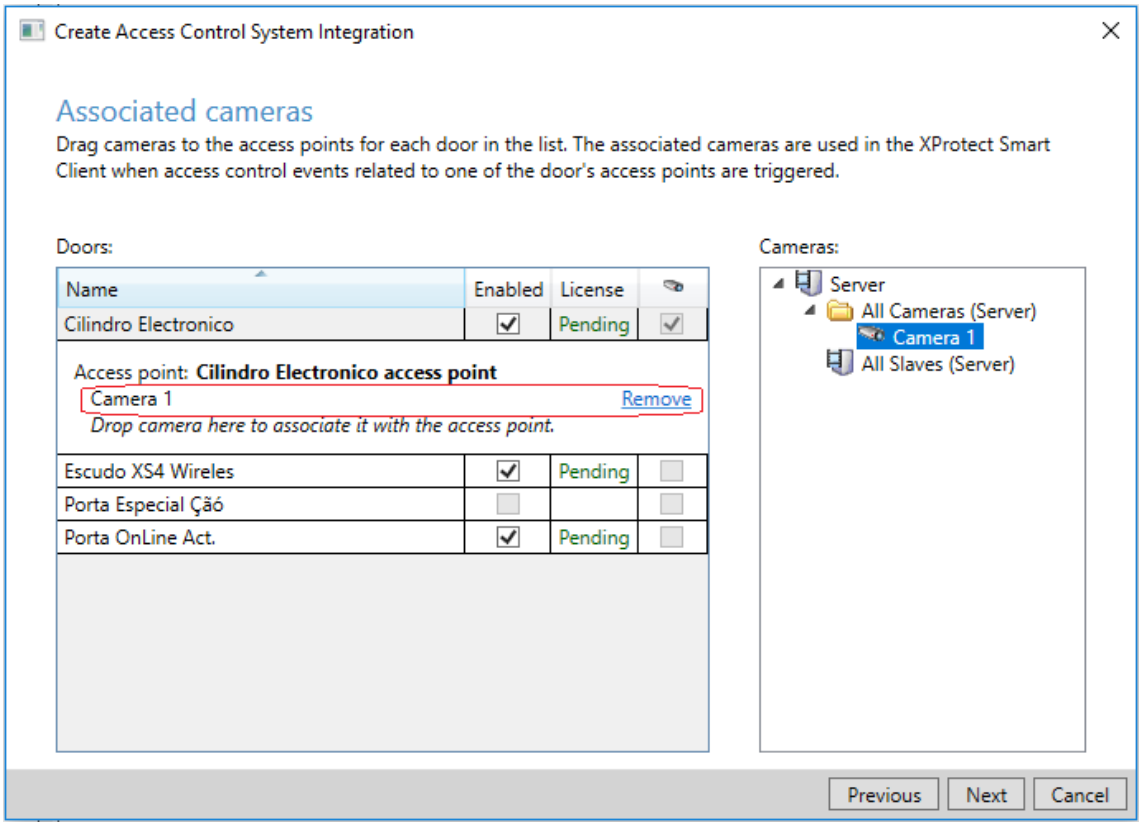
2.2.4 The wizard will continue with a screen identical to the following figure, labeled “Associated cameras”. In this phase of the plugin configuration, you can find a list of the obtained doors from the AC on the left side and on the right side a list of the cameras already registered by the VMS (Milestone XProtect).



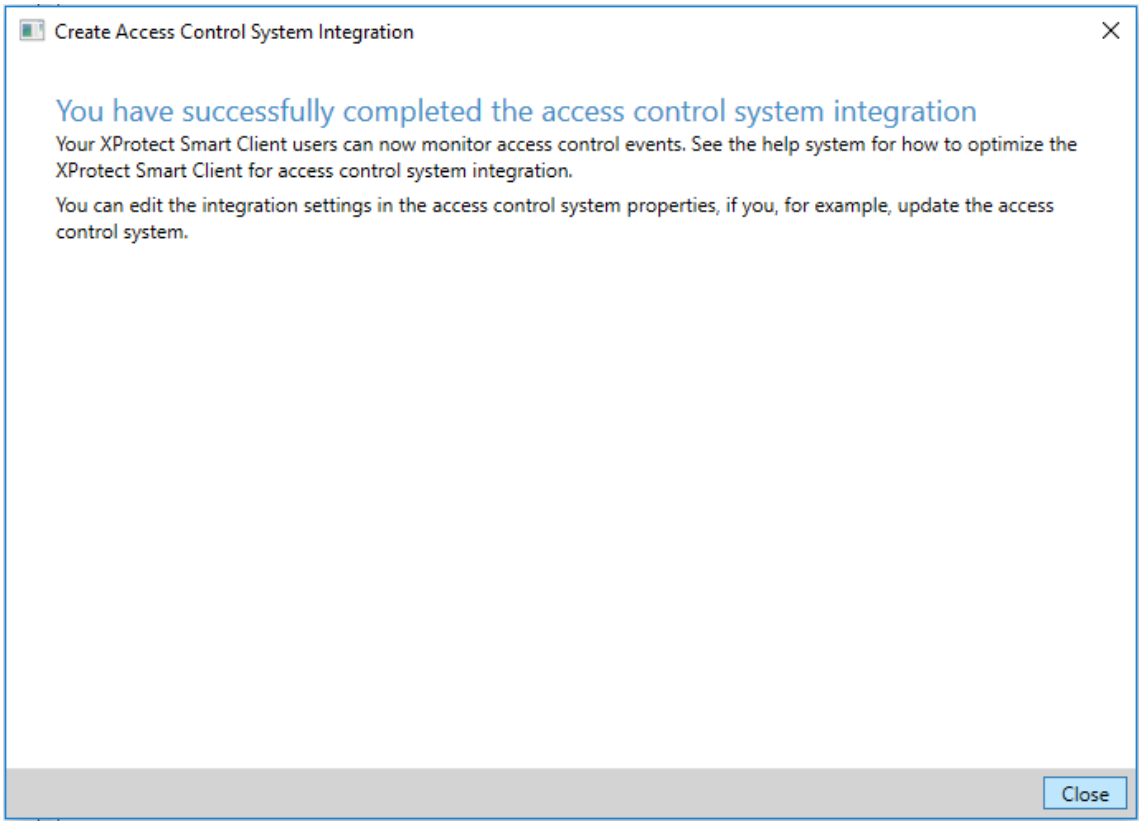
One of the purposes of this VMS Milestone <-> AC Salto integration is also to allow to associate these cameras with doors, in order to later create automatisms of, for example, vídeo recording through the cameras in the occurrence of door events.

So, at this point in the plugin configuration, it is already possible to start this association of cameras to doors, through drag-and-drop operations, taking the cameras on the right side and dragging them to left side. Depending on the doors you want to associate the cameras with, these must be dragged to the zones marked with “Drop camera here…” of the respective doors. In the following figure, comparing to the previous, the subtle difference resulting from the association of “Camera 1” with the “Cilindro Electronico” door can be observed, in which above the expression “Drop camera here…” is the designation of the camera associated with the door, with an aligned “Remove” button, used to remove the association, if so desired.

To proceed, click on the “Next” button.



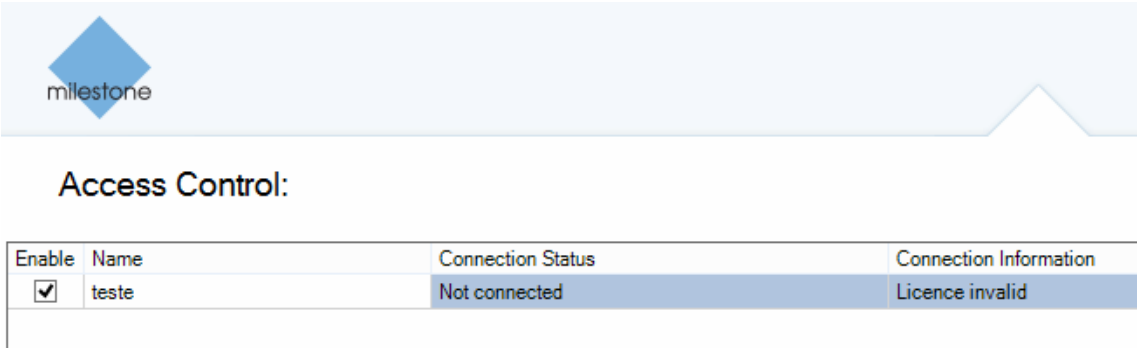
2.2.5 After finishing the previous steps, the plugin configuration wizard finalization screen will appear.



Updating the plugin with a new version (new files) always requires repeating step 1 of the "Plugin" section of this guide, with a nuance: given that it is an update, before copying the new files over the old ones, you must stop ("Stop") the "Event Server service", as exemplified in step 2.1, to unlock the plugin files registered in Milestone XProtect. Then, you can copy the new version files over the files of the version in use, never forgetting the need to, at the end of the update procedure, start ("Start") the "Event Server service" again.

Licensing

If there is no licensing defined for the plugin, it will display a "Licence invalid" message in its connection information:



To license the plugin, you need to obtain a licensing file ("licenca.kws") and place it in a specific location on the WCF web service machine, according to parametrization. This parametrization is found in the web service xml configuration file, "Web.Config". In this, the following line should be located:

<add key="licencePath" value="c:\wcf\" />, where key is the identification of the parameter and value its value, being the second the one that must be adjusted according to the desired location for the licensing file.

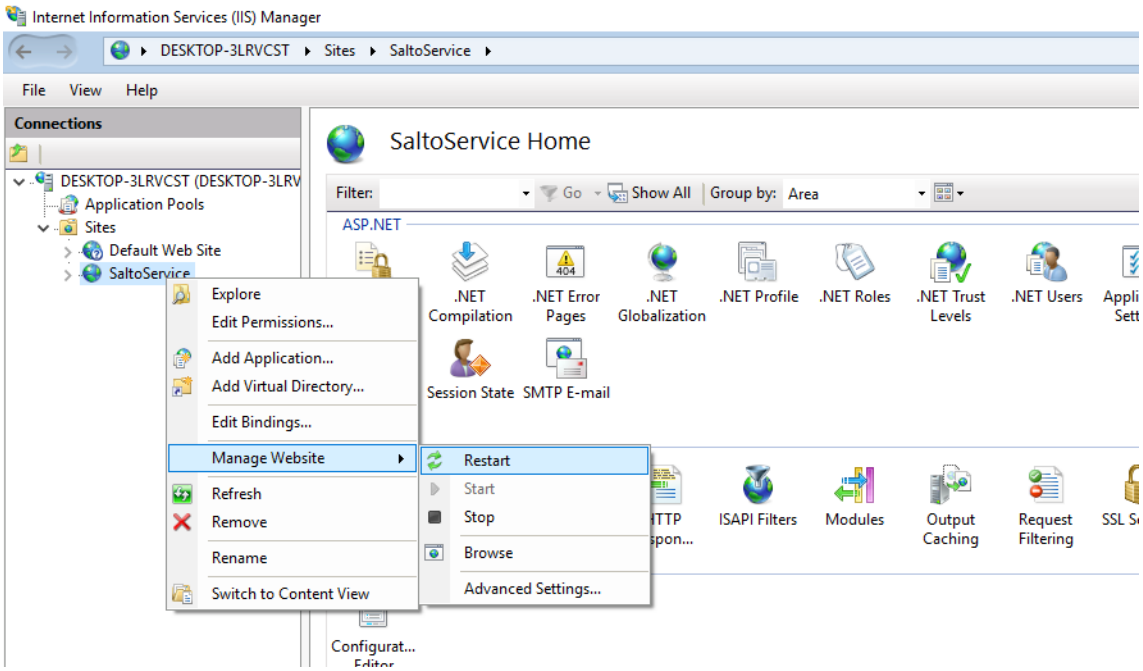
This parameter is, therefore, the indication of the location of the licensing file, so that the web service and the plugin know where it is, verify it and act according to the validity of the license.

In the example above, the web service folder itself is indicated for the placement of the license file, folder where the web service files were placed as indicated in step 4 of the web service installation section of this guide. However, you can choose any other location on the same machine. Even so, for the sake of organization and efficiency, it is advisable to place the licensing file together with the web service files.

As a final and essential step to activate the license, it is necessary to restart the web service in IIS (Windows web server), so you will have to perform the following steps:

➔ Open the Windows IIS application through "Windows Administrative Tools" -> "Internet Information Services"

➔ Next, order the web service to restart by clicking on “Sites” -> SaltoService" (right mouse button) -> "Manage Website" -> "Restart" (as shown below)



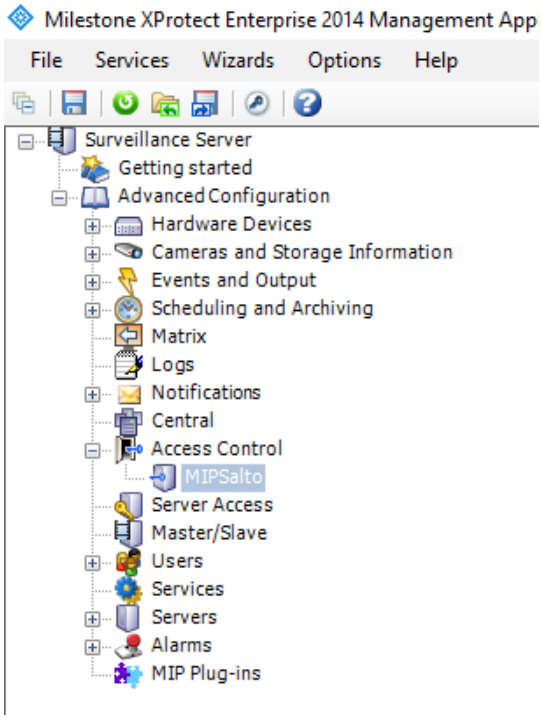
When it comes to a license renewal, simply copy the new updated license file over the previous one, in the same location, and perform again the two steps in order to restart the web service.

Licensing has two aspects to it: limiting the maximum number of doors to be managed by the plugin and expiration date, so its expiration or loss of validity may be related to the introduction of new doors in the Salto AC system, which may, after being imported by the plugin, cause it to exceed the maximum limit of doors allowed by the license or, simply, may be related with exceeding the expiration date.

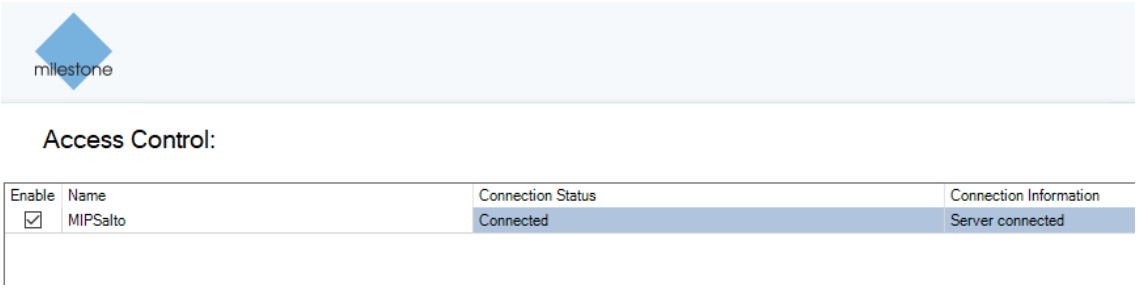
The licensing functionality of the plugin automatically generates on the Milestone machine itself, where the plugin is installed, an encrypted configuration file (“configuration.txt”), in the Windows operating system public folder “C:\Users\Public\Documents\MIPSaltoiSystems”. This file is only useful for the internal operation of the plugin, so it is not necessary to open, edit or move it.

Plugin operation

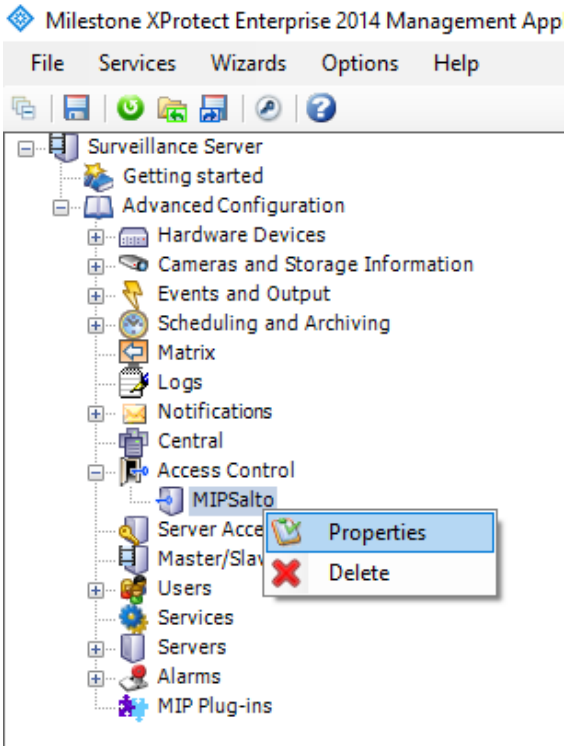
After installation, the installed plugin (“MIPSalto” in the figure) will appear in the list of plugins related to integrations with AC systems (“Access Control” tab in the “XProtect Management Application” management application interface).



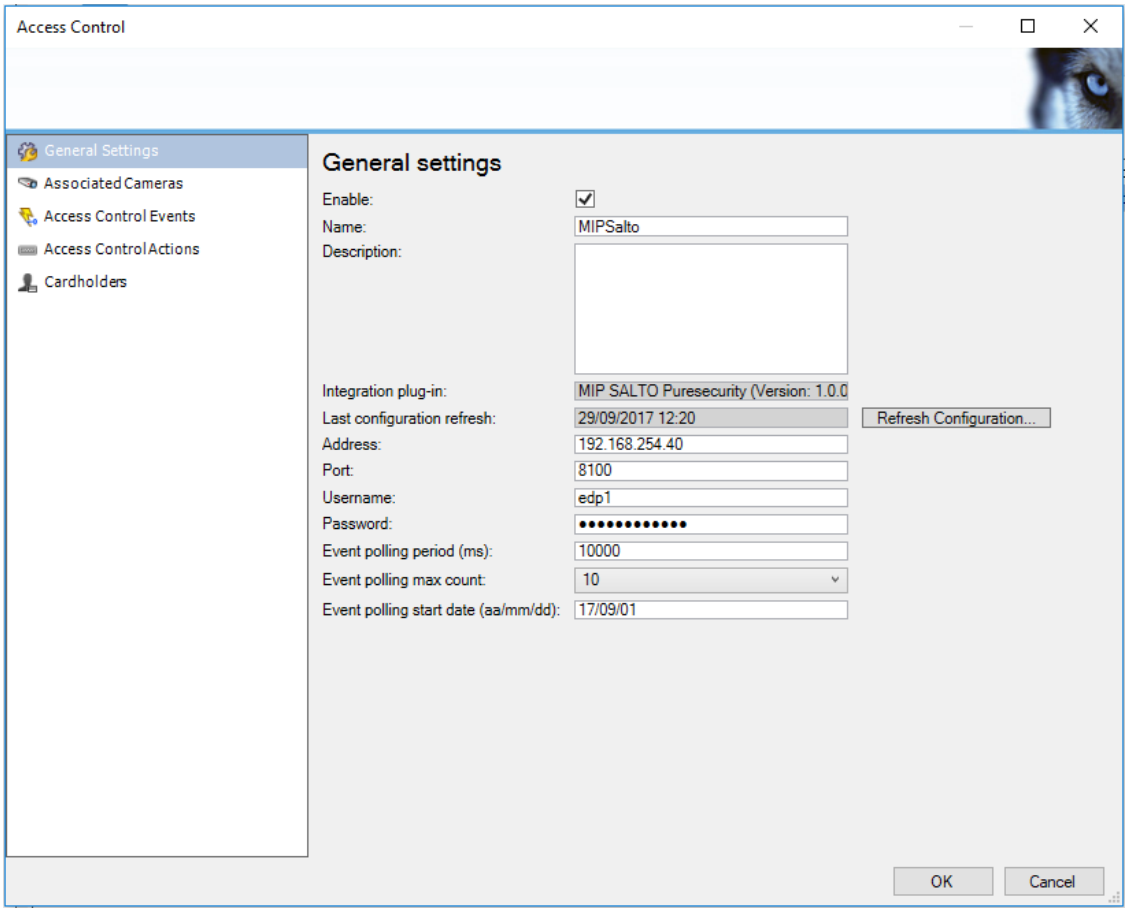
Selecting the plugin in this tab, information regarding the state of the plugin will appear on the right side. When everything is working correctly, the status must be “Connected” and “Server connected”.



To access the backoffice operation interface of the plugin, you should right click on the name of the plugin, then click on “Properties”:



Subsequently the window that allows backoffice management operation of the plugin will appear.



Backoffice plugin management

1 General Settings

In the figure above, you can see the contents of the plugin management tab for general settings. This is also the tab that opens initially when you click on “Properties” in the plugin.

In this tab, in addition to the initial configuration information used in the installation wizard (“Name”, “Integration plug-in”, “Address”, “Port”, “Username”, “Password” e “Event polling start date”), you can find the following plugin configuration fields:

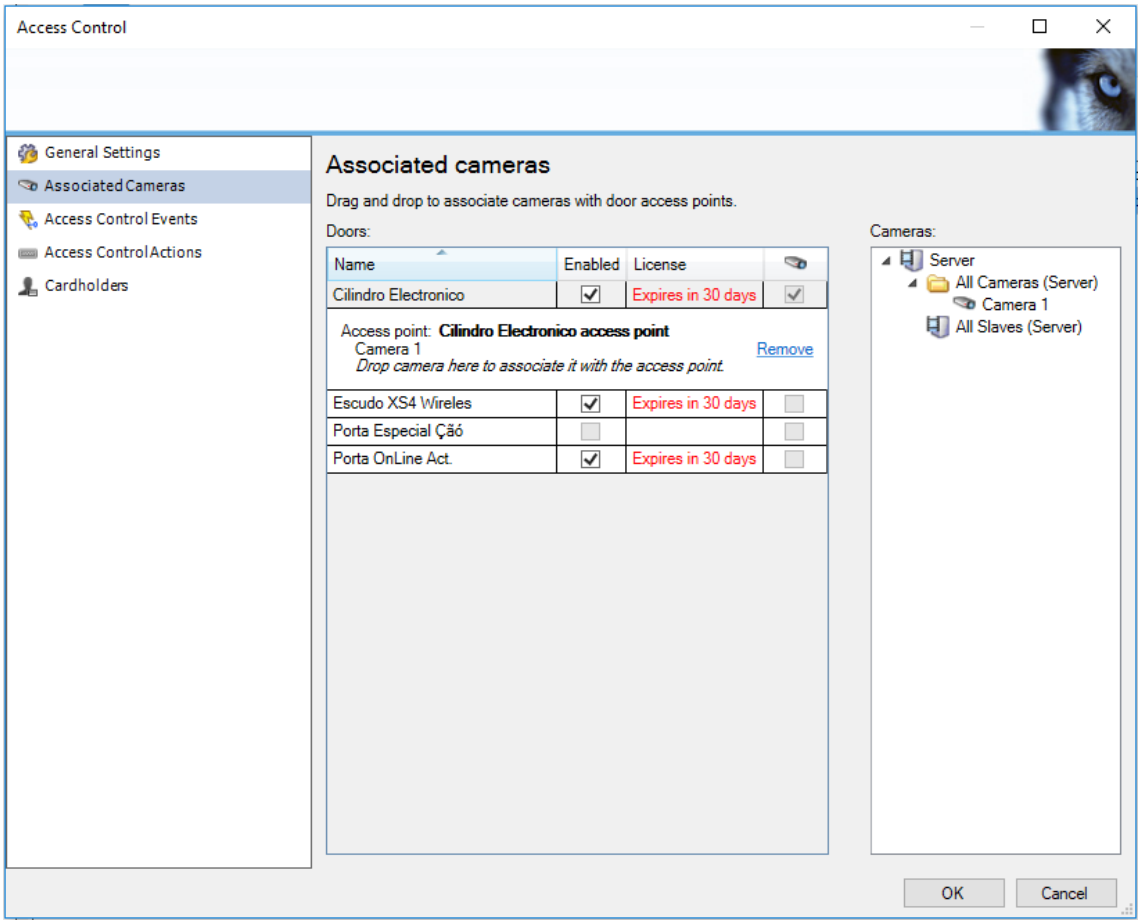
- “Enable”, to activate/deactivate the plugin;
- “Description”, in which you can register a short description of the plugin, or some notes related to it that you may consider relevant and necessary;
- “Last configuration refresh”, which indicates the date of the last information synchronization made with the AC system (Salto). This field has a button associated with it to perform new manual synchronization whenever desirable;
- “Event polling period” and “Event polling max count”. In relation to these two configuration fields, they are paired because they are both associated with obtaining events, the first one being “the time interval between events requests to the web service” (in milliseconds) and the second, “how many events are pulled at a time”. Thus, in the example above, 10 events will be pulled every 10 seconds (10000 milliseconds). Note also regarding these configuration parameters, the importance of their tuning according to the conditions of the network in which the plugin is installed. Each time that events are pulled, traffic is generated on the network.

At any time, you can use this tab to edit and adjust all these plugin configuration fields.

2 Associated Cameras

This tab refers to the association of doors with cameras, in exactly the same way as it appeared a given time in the installation wizard. Cameras can be attached to doors by dragging from the right (cameras) to the left (doors). Cameras can be disassociated from doors by clicking the respective “Remove” button. All associations made during the installation are reflected here and can be changed.

In the figure below the contents of this tab can be observed:

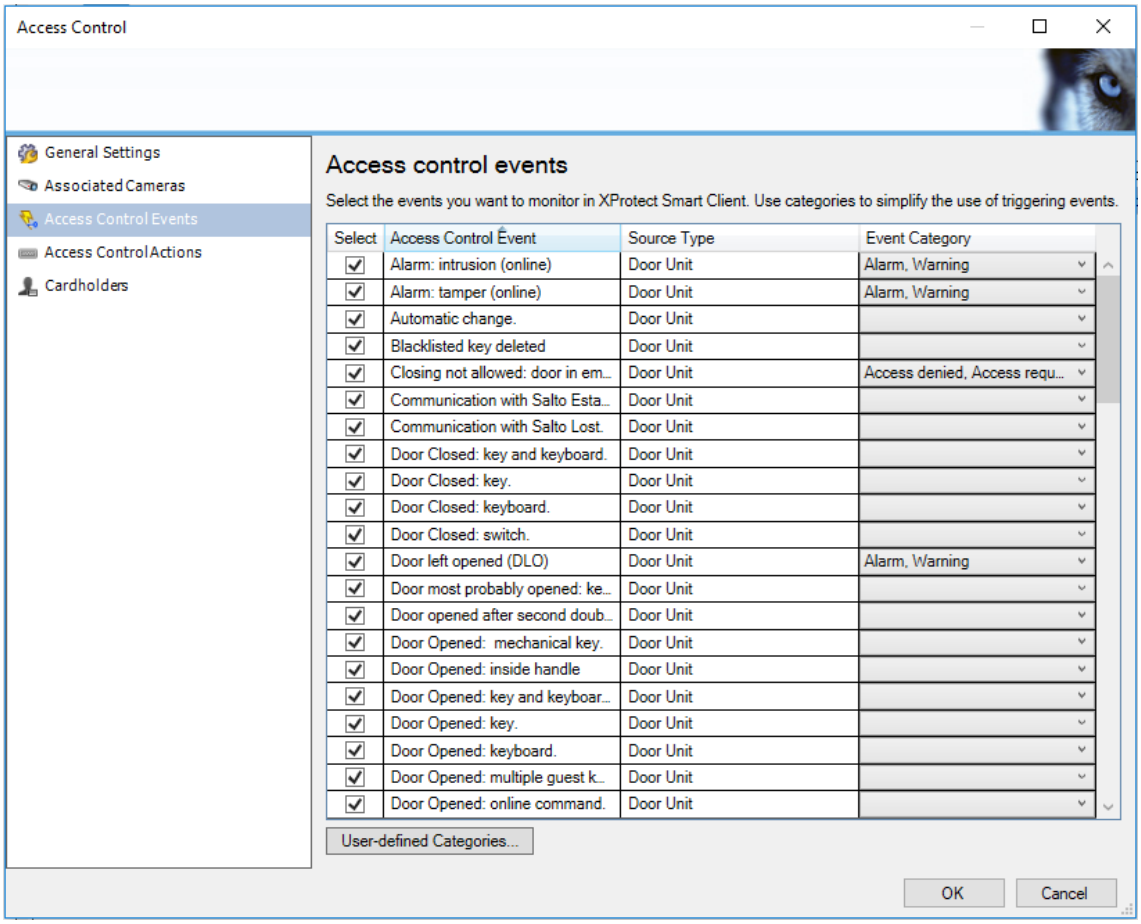


3 Access Control Events

In this tab resides the information regarding the types of events in the AC that were imported to the VMS during the plugin installation. These types of events essentially serve to classify events and eventually group similar events by categories, since several types of events can be grouped (column “Access Control Event”) in a single event category (column “Event Category”), simplifying the observation and analysis of the received event types, since they can be grouped by more significant categories and of simpler immediate understanding.

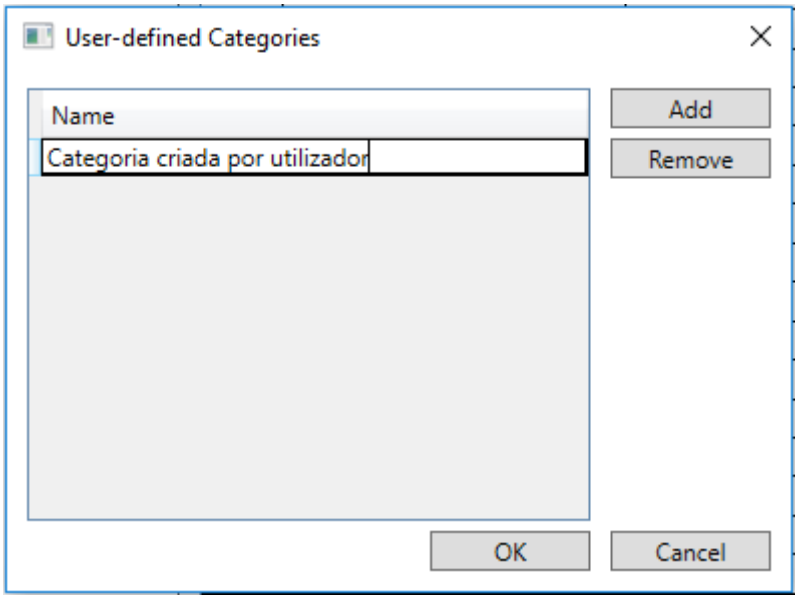
In addition, the “Select” column allows you to select which types of events to monitor and, in contrast, those that you do not want to monitor. It may happen that in certain systems many events of a certain type will occur in the AC but you are not interested at all to monitor them on the VMS side. In this case, you can unselect this type of event in the “Select” column, allowing for events of this type not showing in the events lists.

The functionality this tab refers to can be seen in the figure below:



Finally, note the existence of the “User-defined Categories…” button. This button server precisely to create additional categories, adjusted to the needs and context of the company and the system in which the plugin was installed. These “extra” categories can then be used to group event types by its choice, after creation, in the “Event Category” column.

The new category creation window can be checked in the following figure example:

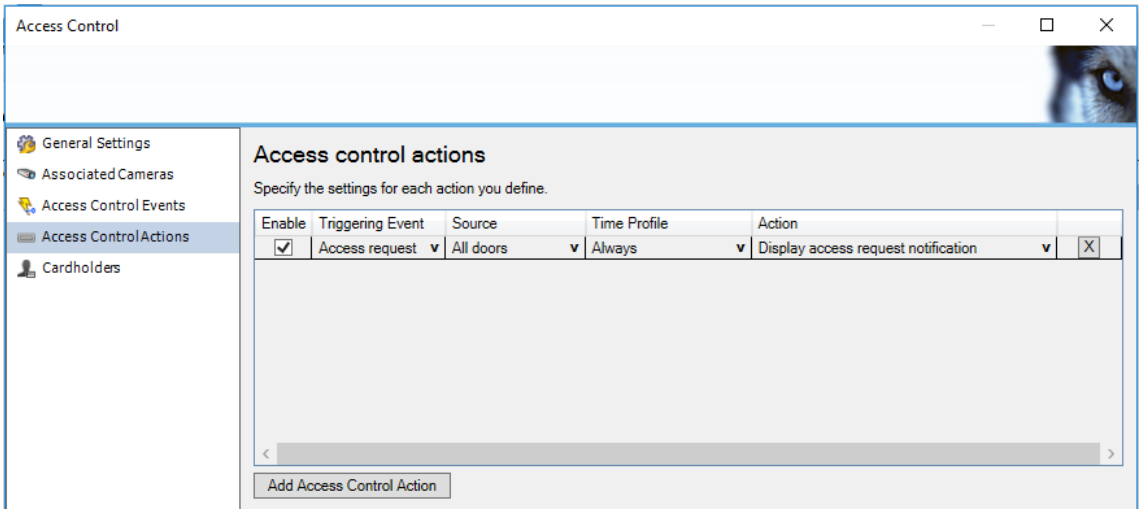


4 Access Control Actions

In this plugin management tab, actions of various types can be created according to several parameters, that is, it allows to configure the occurrence of actions (like notifications) depending on the occurrence of events of a certain type or in a certain way. In the example below, an action ("Action") is created that allow for a notification to be launched ("Display (…) notification") when events of access request nature occur ("Triggering Event") at any door type ("Source") and at any moment ("Time Profile").

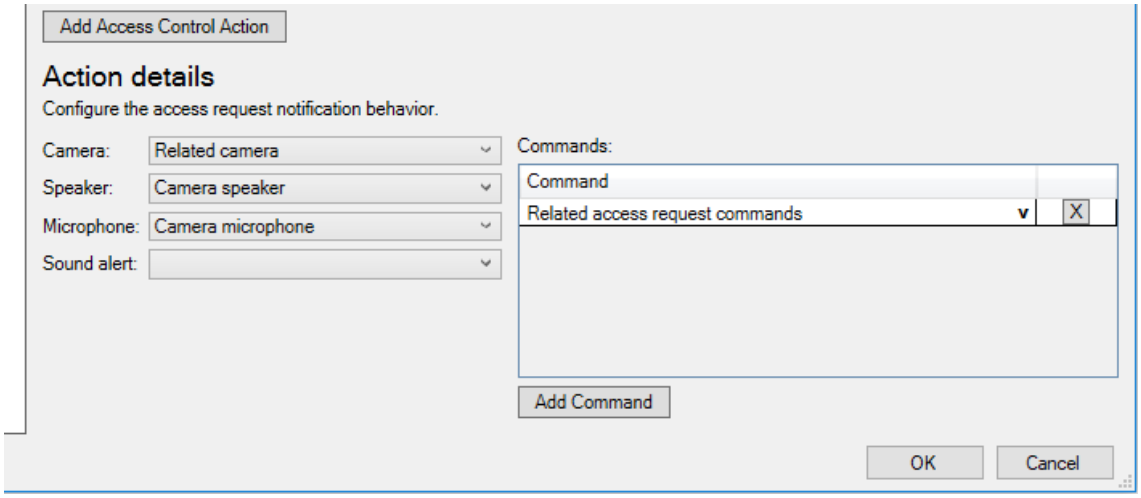
Next is a description of the various functional aspects of this tab:

- "Add Access Control Action" button, serves to create a new action and, in practice, adds a new line to the list of actions above the button;
- In the new action creation line:
    - "Enable" column, is to activate/deactivate the action after it has been created (it may no longer be useful at any given moment or during a given period);
    - "Triggering Event" column, serves to choose the type of event that will trigger the action;
    - "Source" column, serves to choose the source in which the event that will trigger the action may occur, in this case, a door (or doors);
    - "Time Profile" column, serves to choose the time or period during which the action may occur;
    - "X" column, is used to completely erase the action from the system.



In addition, some actions may have some kind of finer detail to define. This will be the case with notifications and, therefore, the example in the figure above. In this case, by selecting the action in the list, the finer data whose definition is needed to complete the action configuration will appear in the zone below the action creation button (unless the intended is the 'default').

In the given example, this detail can be observed in the following figure, corresponding to the detail originated by the action chosen in the previous figure.

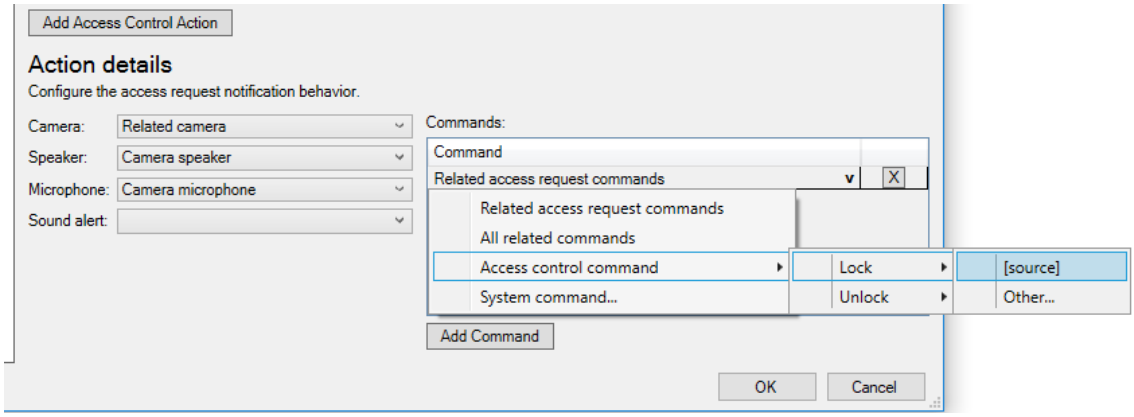


The detail for this action, of the notification type, includes, on the left side, the definition of the camera that can be triggered by the action, and other aspects such as the loudspeaker, the microphone and the sound alert.

On the right side, is located the definition of the commands allowed for this action of the notification type, in this case, the 'default' commands ("lock" and "unlock") are defined, according to the characteristics of the action in question.

A new command can be added by clicking the "Add Command" button or the existing command can be changed by clicking on it. Imagine that it is intended that in the generated notification only one of the commands "Lock" or "Unlock" is allowed: in this case, select the created command line and then, since not all the available commands are wanted, but only one of them, select "Access control command" -> "Lock" or "Unlock" (according to the command that is to be made available in the notification window) -> "[source]" (meaning the source of the notification, in the case, probably a door for which access was requested by a cardholder).

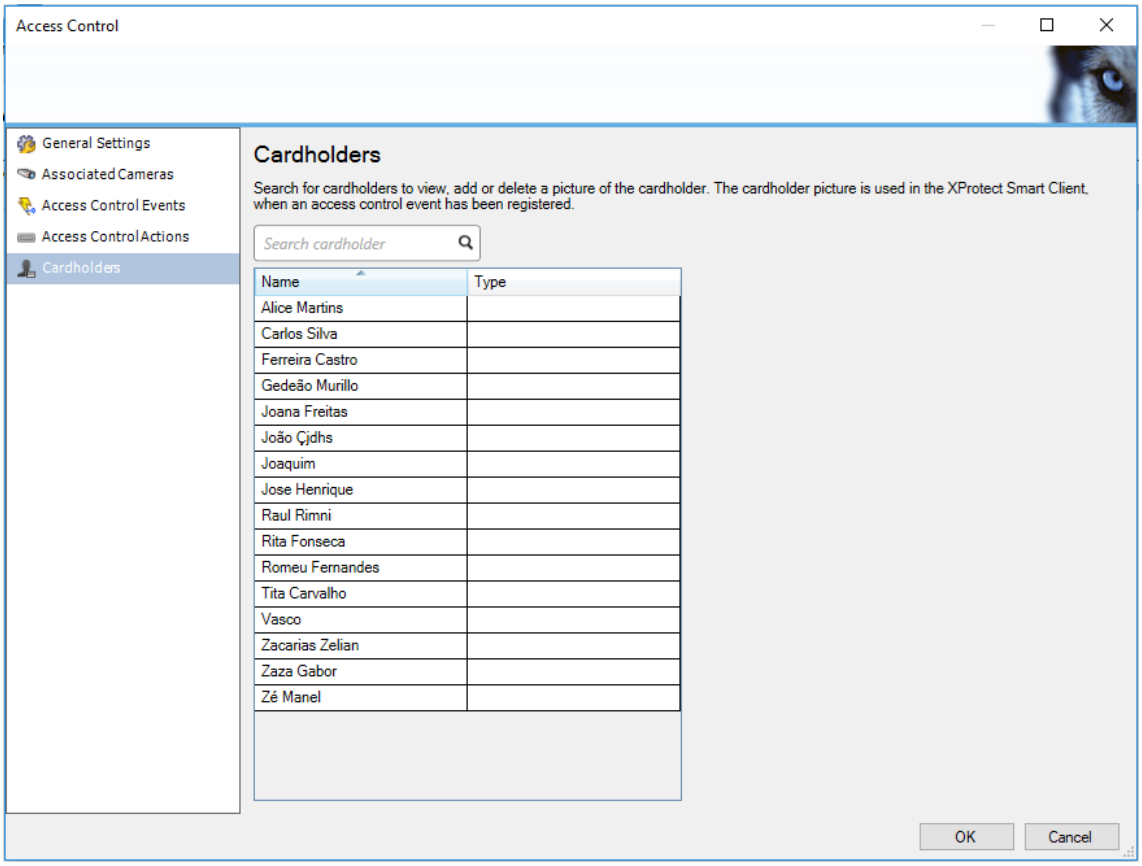
This sequence of options can be confirmed in the following figure:



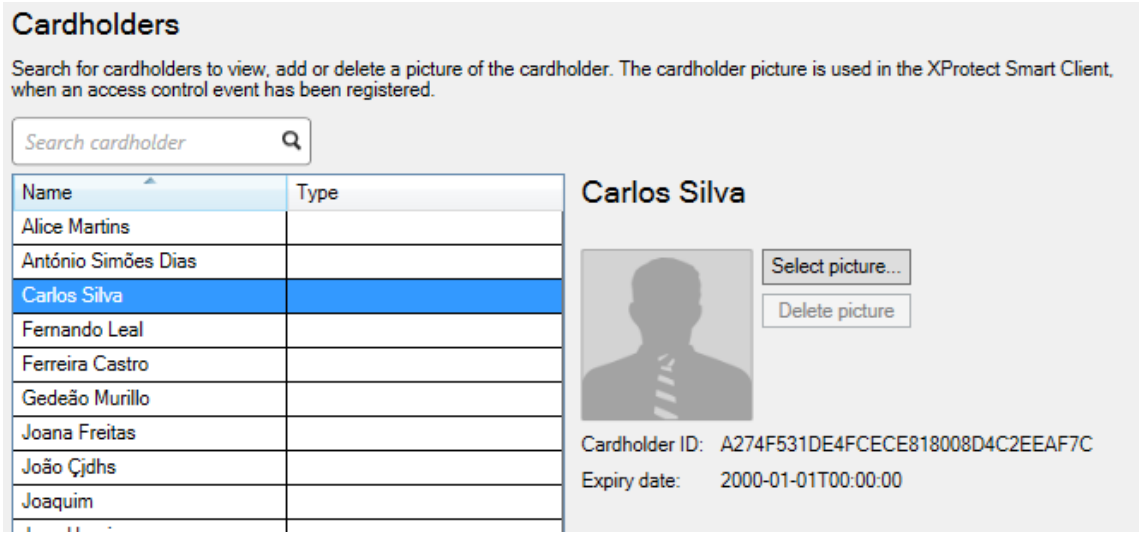
5 Cardholders

This last tab of the plugin's management interface includes information about the cardholders (card users, meaning, people registered in the system with access to the installations/doors).

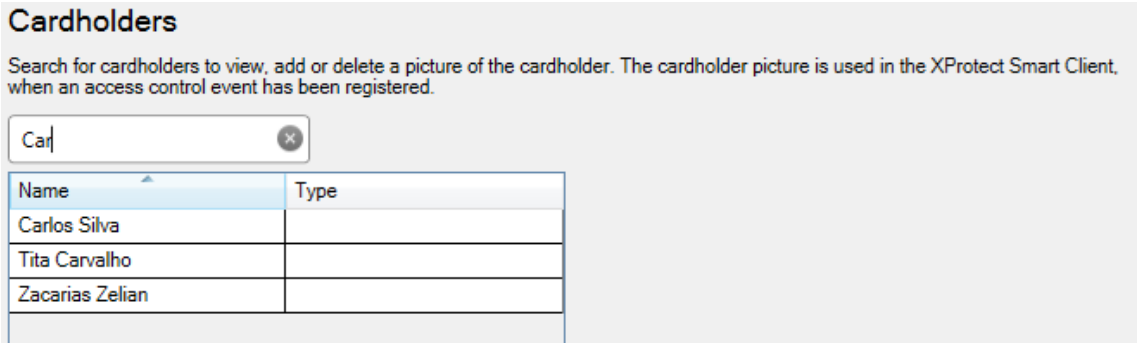
This is a simple list of cardholders, imported from the AC to the VMS.



Clicking on each cardholder's name gives access to more detailed information that will appear in the space to the right. In the following figure example, the detail of the fictitious cardholder "Carlos Silva" can be observed:



In large systems, this list of cardholders can reach a very large size, so the plugin offers a cardholders search functionality, simply by placing the text string that is to be searched in the text box with the magnifying glass, being the search result immediately reflected on the list, as can be observed in the following figure:



5.1 Cardholders photos

It's possible to associate photos to their respective cardholders.

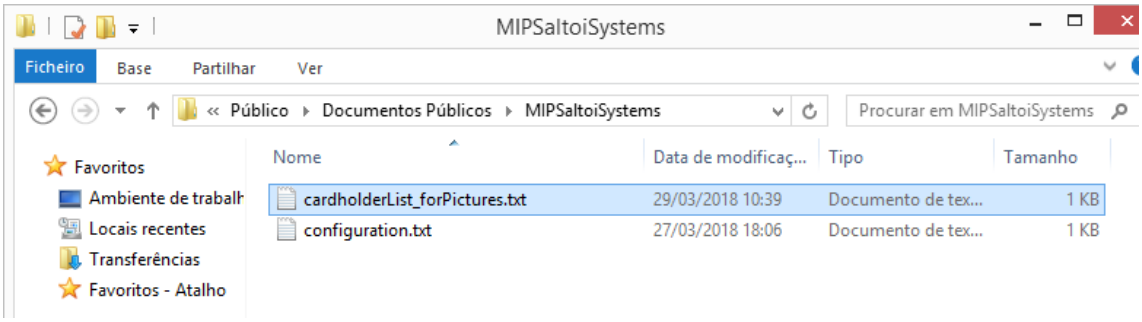
After the initial installation and consequent initial obtaining of the list of cardholders, a text file with information of the cardholders (name and ID) is generated in the system. Whenever there is a change in the list of cardholders in the system, and it is consulted through the management backoffice of the plugin, this file will be updated.

The generated file is no more than a support file whose purpose is to facilitate the operation of assigning photos to the cardholders and will be described below.

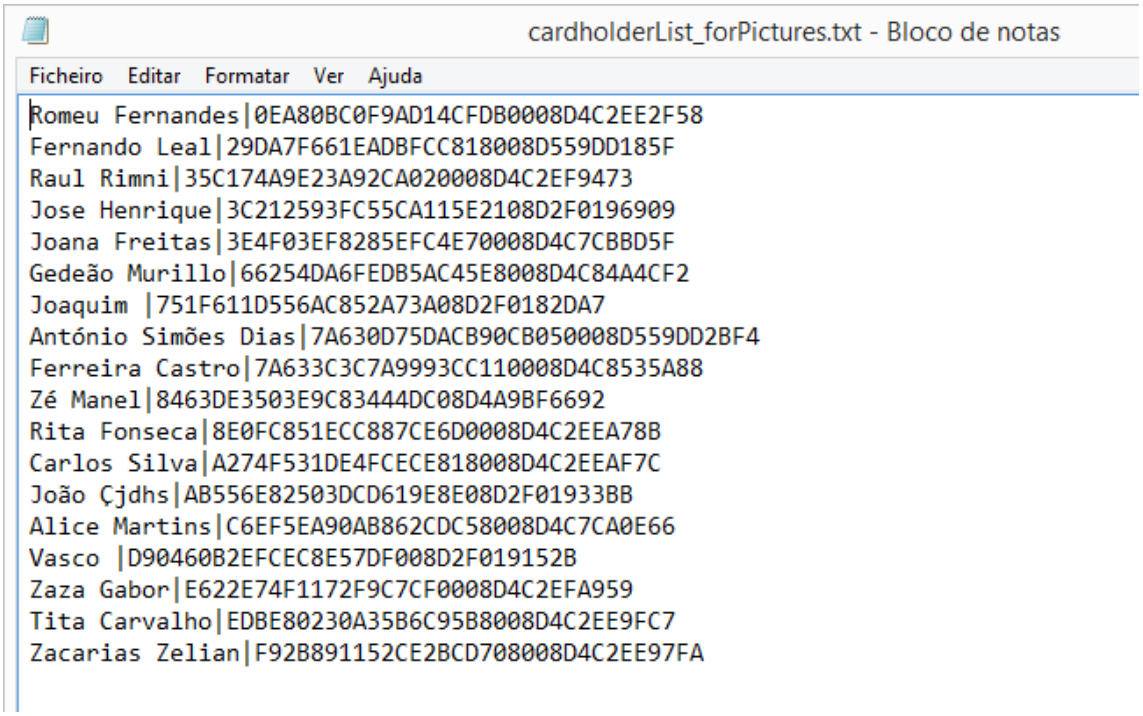
This file, as indicated, is generated automatically and will be located in a public folder of the Windows operating system on the Milestone machine, where the plugin was installed:

"C:\Users\Public\Documents\MIPSaltoiSystems"

The "MIPSaltoiSystems" folder (if it does not already exist) is also automatically generated in the public folder structure already in the operating system. In this folder you can then find all the files that the plugin generates automatically, as a support for its operation. One of them is the file "cardholderList_forPictures.txt", used to support the operation of assigning photos to cardholders:



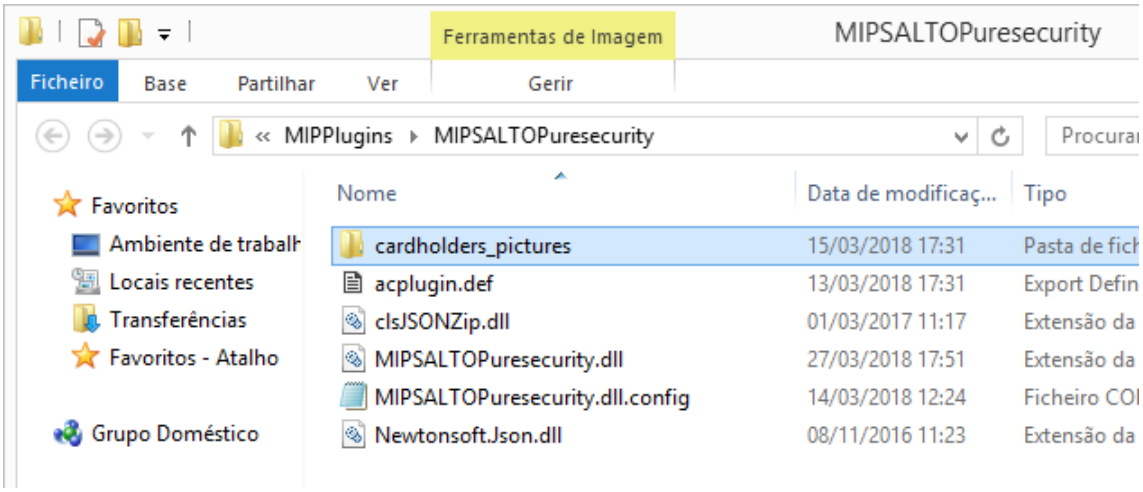
Being a simple text file, it can be opened with a simple text editor, such as the Windows notepad accessory. In the following figure can be seen, by way of example, the contents of this file, referring to a list of fictional cardholders:



The content is no more than the name and ID information of the cardholder for each line. In this way, one has easy access to this information, rather than doing it through the management backoffice and clicking each cardholder one-by-one.

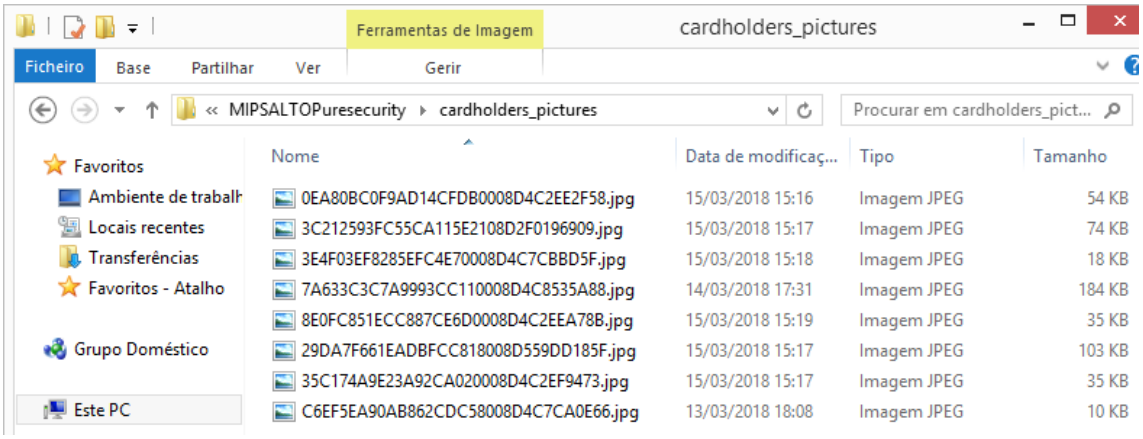
The next step for assigning photos to cardholders is to create the “cardholders_pictures” folder in the folder where the plugin was installed, ie, in the folder in which the plugin files were placed, on the Milestone machine, according to the procedure described in the section “Plugin installation – Step 1”:

"C:\Program Files (x86)\Milestone\XProtect Event Server\MIPPlugins\MIPSALTOPuresecurity"



Within this folder ("cardholders_pictures") should be placed the photos of the cardholders, in the form of JPG image files, and with file sizes ranging from a few KB (eg, ~10KB), to a few dozen KB, but avoiding to exceed 150-200KB, for efficiency reasons.

In the following step, the support file “cardholderList_forPictures.txt” mentioned above will become useful. The name to assign to each cardholder photo file must be the ID of its respective cardholder. This information will be readily available in such a support file. For a given cardholder, simply find its name in the file contents and, on the same line, after the “|” character can be found its ID, which should be selected, copied and pasted in the rename of the respective JPG image file. At the end of this operation, all cardholder photo files should have their cardholder ID as their file name, as shown in the following figure:



The recognition of the new photo files by the plugin requires the final step of the “Restart” of the “Event Server service” in Milestone XProtect management backoffice, as exemplified in step 2.1, in the plugin installation section of this guide.

This will allow the plugin to automatically associate to each cardholder its photo by way of the ID key field. The effect will be the appearance of the cardholder’s photograph in its detail, as opposed to the appearance of a generic image: