

Active Process Manager

User Guide



Table of contents

Table of Contents

1	Introduction	5
2	System requirements	6
3	Concepts.....	7
3.1	Solution.....	7
3.2	Step.....	7
3.3	Plug-in.....	7
3.4	Parameter.....	9
3.5	Basic principle of operation.....	9
4	Solution examples.....	12
5	Installation	16
6	Licensing.....	16
7	User roles.....	17
8	User interface.....	17
8.1	Main window sections	19
8.1.1	Front end.....	19
8.1.2	Display and input.....	19
8.1.3	Status.....	19
8.1.4	Information.....	21
8.2	Main window's menus and buttons.....	22
8.2.1	Solution menu	22
8.2.2	Tool buttons	23

9	Configuring APM's options	24
9.1	Plug-in configuration.....	26
9.2	Management	27
9.2.1	Clear recently used solutions list.....	28
9.2.2	User management.....	28
9.2.3	Execution environment settings.....	31
10	Running APM solutions	34
10.1	Cycles	34
10.2	Solution state.....	34
11	Creating APM solutions.....	35
11.1	Flow view.....	36
11.2	Globals section	37
11.3	Adding steps and plug-ins	38
11.3.1	Adding plug-ins	38
11.3.2	Configuring plug-ins.....	39
11.3.3	Configuration window of a plug-in	39
11.3.4	Plug-in properties.....	42
11.4	Post action	45
11.5	Linking plug-ins	47
11.6	Linking parameters.....	47
11.7	Setting up solution properties.....	49
11.8	Setting up Front ends	49
12	Command line parameters.....	52
13	Calling APM from other software.....	53
14	Kiosk mode.....	54
14.1	Configuration	54

Version	Author	Date	Comment
1	JK	2024-01-18	New template, kiosk mode updated, kiosk configuration section added, and system requirements updated
2	AH	2025-12-19	Additional information on usage through command prompt (reviewed 2026-01-12 HH)
3	HH	2026-06-04	Updated the document to be up-to-date with the new APM 4 main UI.
3	JK	2026-06-25	Reviewed with minor changes

1 Introduction

Development of Active Process Manager (APM) began in 2004, with the target to make the process of personalising PKI smart cards flexible and easily configurable. Today smart card personalisation is still the main usage of APM, but its features have been extended to automate many other processes as well. APM can also be used for creating the back end and user interface for self-service kiosks.

The product was originally named to ActivePerso Manager. In 2018 it was renamed to Active Process Manager to reflect better its wider range of possible usage scenarios. The well-established APM acronym remains the same.

APM could be called a process automation engine; It controls a set of plug-ins to perform various tasks in a freely defined order. The plug-ins share a common interface, and each plug-in contains a list of parameters, which can be used to transfer data into and out of the plug-in. Plug-ins, their parameters and interaction are configured into an APM solution. The plug-ins are executed in sequential order. Each plug-in performs a specific, but usually flexibly configurable task. There are data source plug-ins to collect data from user or from for example SQL databases. Personalisation plug-ins perform for example smart card encoding and printing. Data Execute plug-ins update databases and other data stores. Plug-ins have been developed for many purposes, to control and interact with various software components, services, and hardware devices.

Few examples, what APM can be used for:

- Smart card or plain plastic card personalisation
 - contact chip, RFID and magnetic stripe encoding
 - visual printing and layout design
- Cryptographic key management
 - Perform key management tasks such as key derivation, wrapping and unwrapping on security devices like HSMs
 - Manage key creation, life cycle and renewal
- Cryptographic tasks
 - Encrypting, decrypting and signing data on security devices such as smart cards and HSMs.

- Printing forms and security envelopes
- Batch processing of data - Transferring data between databases and/or files and processing the data in various ways
- Collecting user data from input devices such as keyboard, external touch screen tablets and signature pads.
- Interacting with Active Directory – searching and updating data and transferring data to/from other systems
- Photography – taking photographs of company personnel, automatic photo processing and storing them in databases. Both operator run and self-service solutions are possible.
- Activating and deactivating IDs in access control

Because APM is very flexible many other usage scenarios are possible as well. If a plug-in for specific purpose does not exist, it can usually be developed in reasonable time.

This user guide describes the features and usage of APM in general. There is another document (APM Plug-in Guide), which contains a list of all APM plug-ins and a brief description of each of them. In addition, there are separate user guides for some of the most complex plug-ins.

2 System requirements

The supported operating systems are:

- Windows 7
- Windows 8 / 8.1
- Windows 10
- Windows 11
- Windows Server 2008 and newer

3 Concepts

The four key concepts in Active Process Manager are **solutions**, **steps**, **plug-ins** and **parameters**.

3.1 Solution

A solution represents a complete project or customer specific configuration saved to a single solution file. It can be construed as a collection of individual tasks needed in different phases to manage a larger overtaking, like e.g. the management and personalization of ID cards. A solution contains one or more **steps** that can be run separately.

3.2 Step

A solution step represents a single task that can be carried out, like “Personalise an ID card” or “Read all person data from the HR system”. Typically, each step is represented by a button in the main user interface that the user can press to launch the respective task. A step contains a number of **plug-ins** that are executed in a predefined order.

3.3 Plug-in

Plug-ins are software components that perform a specific elementary task in a step. They make the actual functionality of a step. They are executed in a predefined order, and exchange data with each other, and with the external world. There are currently more than 90 different plug-ins.

Plug-ins share a common interface enabling APM to control them in a well-defined manner. For example, Visual plug-in is used for editing visual layouts and printing them on media, such as smart cards. SQL Query plug-in is used to read data from SQL databases. CMP Client is used to request PKI certificates from a CA system using the CMP protocol. There are several categories of plug-ins. The two main categories are control plug-ins and work plug-ins.

Control plug-ins are placed at the beginning of a solution. They provide services for work plug-ins or other control plug-ins. The control plug-ins include Smart Card, RFID and Data Source.

Work plug-ins form the actual functionality of the step. There can be many instances of the same plug-in in one step. Work plug-ins are placed after each other according to the desired execution order. A special “jump” plug-in exists to control the workflow.

The control plug-ins and work plug-ins fall into several subcategories depending on their purpose. However, there are no significant restrictions on what a plug-in can do. A data source plug-in could for example retrieve data from HTTP requests, and a control plug-in could control a custom microcontroller through a serial interface.

Common properties of plug-ins are:

- Provide its own user interface for configuring the plug-in
- Define a set of input and output parameters, which can be linked to other plug-ins
- Store its own configuration as XML and parse its own configuration from an XML node.
- Verify consistency of its own configuration.
- Provide functions for APM to begin processing, process one execution cycle and end processing.

Currently, the following plug-in categories are included in APM:

Type	Category	Purpose
Control plug-ins	ActiveCMS Connector	Manage connection to an ActiveCMS system
	CA Connector	Manage connections to CA systems
	Data Source	Manage connection to a data source, or directly provide data for the solution
	Image Capture	Manage connections to cameras and other image capture devices
	PKCS#11 Connector	Manage connection to a PKCS#11 cryptoki-library
	Printer	Manage connections to (smart card) printers
	RF-ID	Manage connections to RF-ID readers
	Smart Card	Manage connections to smart card readers
	Tablet Connector	Manage connections to tablet computers
Work plug-ins	ActiveCMS	Fetch data from ActiveCMS or perform updates to ActiveCMS
	Card Mover	Move smart card between modules of a card printer
	Certificate Management	Request certificates from a CA and other certificate management tasks
	Cryptography	Cryptographic computations, PKCS#11 operations
	Data Execute	Execute updates to data sources
	Data Processing	Data processing logic, for example running scripts
	External	Execute external software
	Network	Call network services, for example REST or SOAP web services
	Personalization	Generic smart card or token personalization tasks
	RF-ID Personalization	Personalization of RF-ID/NFC cards or tokens
	Smart Card Personalization	Smart card personalization, for example managing and updating ISO 7816-15 file structure
	Solution Logic	Plug-ins that control APM solution's execution flow
	Tablet Client	Perform tasks on a tablet computer, for example capture user's written signature
	User Input	User input via keyboard or other input methods, displaying data to users
	Visual Personalization	Printing to smart cards and other medias

Table: Plug-in categories

The available plug-ins are documented in **Active Process Manager Plug-in Guide**. This document contains a brief description of each plug-in. Besides this document there is additional documentation available about the most complex plug-ins.

3.4 Parameter

Plug-ins have can have user-defined and built-in parameters. Older APM versions had three parameter types: text, hexadecimal and base64. In APM 4, the parameter system has been extended with several more types and new functionality. The new extended parameters are compatible with the older parameter types, and APM performs automatic conversions when applicable. New parameter types include, Binary, Certificate, Key, Picture and DateTime. Parameters can be mandatory or optional. The purpose of the new, more refined parameter types is to make building solutions easier, as there is better visibility regarding what kind of data is needed for a specific parameter. The parameter types also contain unified logic related to the data type of the parameter.

3.5 Basic principle of operation

The following image demonstrates a very simple solution step, which inputs a text from user in an Input Console and displays it in APM's print preview area.

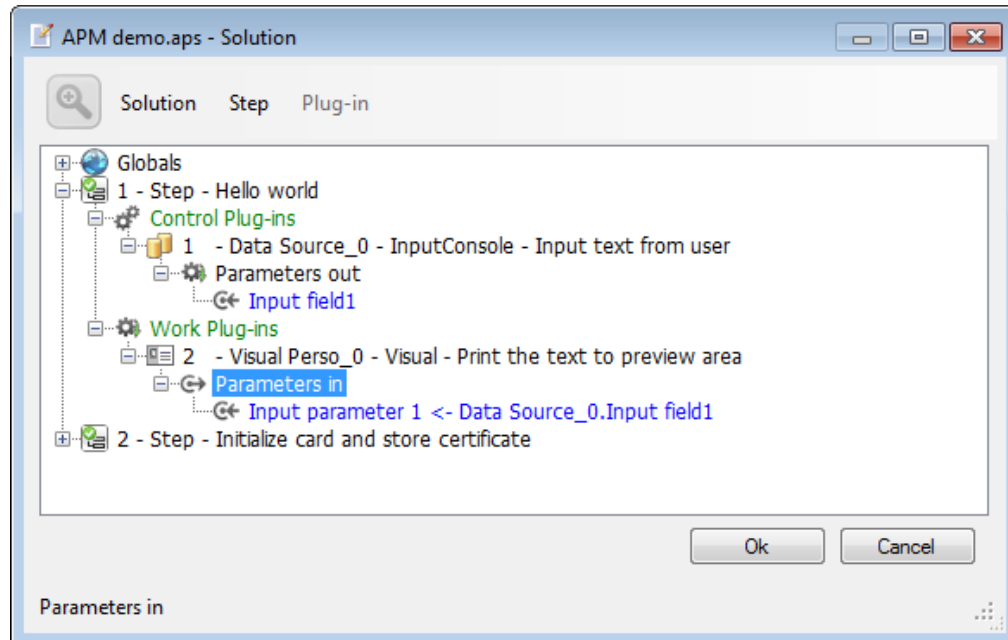


Image 1: A solution step

The first plug-in (Input Console) has a parameter named “Input field1”. It has been set up to a default value and the user can replace it with any text. The input console plug-in can also control how many times the step is executed. The default value is one cycle.

When the step is started, the following user interface is shown to the operator:

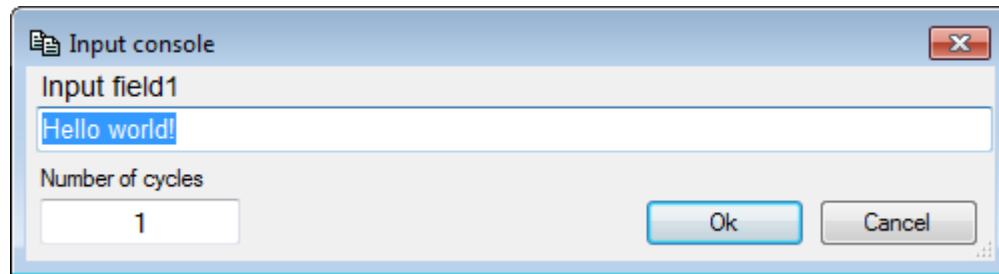


Image 2: User interface of the Input Console plug-in

When the operator presses the ok button, Input Console sets the value of its output parameter “Input field1” to “Hello world!”. The parameter is linked to parameter “Input parameter 1” of the Visual plug-in. This visual plug-in instance has just one text field that is linked to “Input parameter 1”. Visual plug-in prints the text field to the preview canvas, which is shown in APM’s main window.

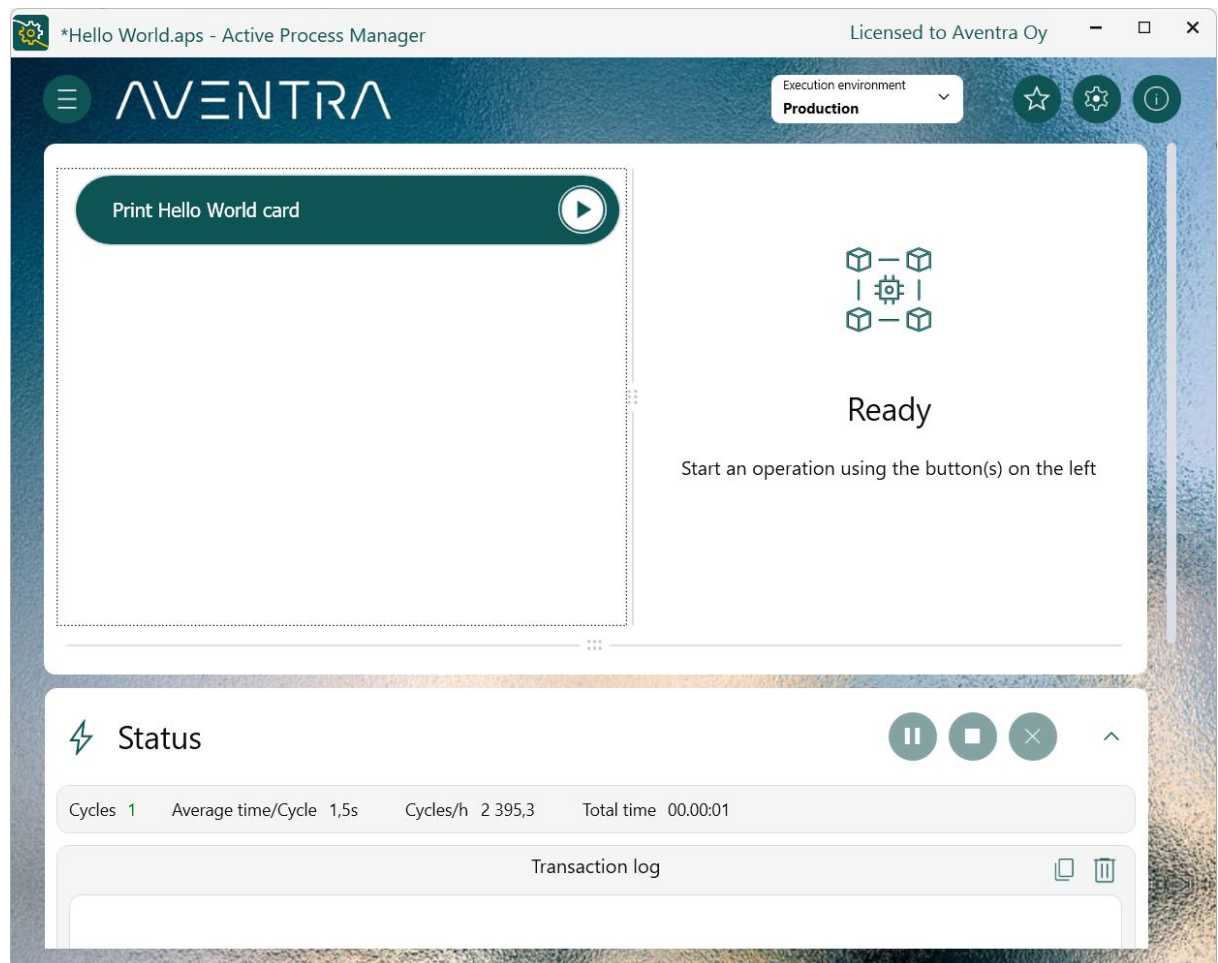


Image 3: Simple solution: preview in APM's main window

A solution step consists of a set of plug-ins executed in sequential order. There can be many instances of a specific plug-in, with different configurations. The plug-ins can pass parameters to each other.

In APM's Solution Editor the current solution is shown in a treelike view. For each cycle processed, the plug-ins in the solution tree are executed from top to bottom. Exceptions to this can be made by using plug-ins that control the execution flow. You can for example configure a Jump plug-in to skip a set of plug-ins, when specific conditions are met.

All configuration is done using the graphical solution editor. You can insert or delete plug-ins and instruct the plug-ins to exchange data by dragging an output parameter of a plug-in to an input parameter of another plug-in with the mouse. Each plug-in contains its own configuration editor.

4 Solution examples

Let's look at a few more complex solutions and functionality. The step "Initialize card and store certificate" takes a MyEID PKI card into a card printer, initializes the card, generates a 2048-bit RSA key pair, and stores a self-signed certificate into the card. The user data for the card/certificate is read from a SQL database.

In the control plug-ins section there are the plug-ins, which manage the connection to the external world, like peripherals, services, and the user. The second plug-in (Data Source) executes a SQL query after the solution has been started. In each cycle, the plug-in outputs three parameters: "Serial number", "Common name" and "PIN". In the work plug-ins section, a Card Mover plug-in commands the card printer to take the card to the printer and move it to the chip contacts. The next plug-ins do the following:

- PKCS15Initialize plug-in initializes the MyEID applet and creates the PKCS#15 file structure on the card. It also creates one PIN code and sets its value with a parameter linked from the database.
- PKCS15Update plug-in generates an RSA key on the card. The plug-in also adds the necessary metadata to the PKCS#15 Private Key Directory File.
- CertCreator plug-in creates a self-signed certificate.
- PKCS15Update plug-in stores the certificate into the card.

Finally, a Card Mover plug-in instructs the card printer to eject the card if the previous plug-ins succeeded.

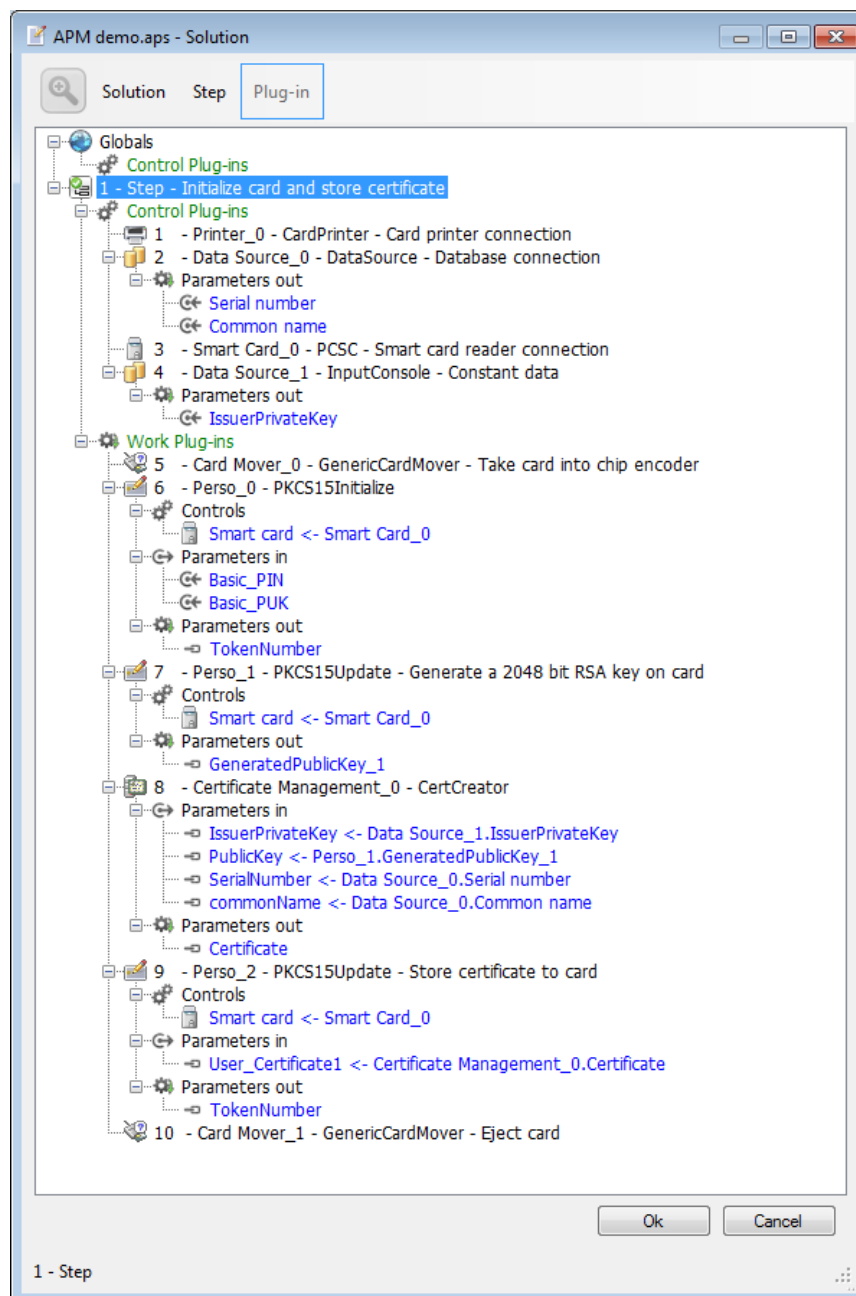


Image 4: Example of an APM solution

When the step is started, it is executed as many times as there are records matching the SQL query configured to the data source plug-in. When all the records have been processed, the data source plug-in informs Active Process Manager's main program to stop the execution.

The second solution below uses a newer plug-in called "ExecuteSteps". It can be used to execute a step from within another step. In this solution the only step that is executed from the front end is the first one (1 - Step). From that step forward the execution follows these steps:

- DataEditor plug-in asks the user for a search parameter that is used as an input for the first ExecuteSteps (1.)
- The first ExecuteSteps plug-in is used to call for the execution of the LDAP Search step (3 - Step) using the UPN given by the user as an input. (2.)
- The solution execution continues into the LDAP Search step (3 – Step) and the search parameter given by the user is used to search for the user account from active directory with DataQueryLDAP plug-in. (3.)
- The user account information is then used as an input for OutputParameters plug-in, which then comes out as ExecuteSteps output from where the plug-in was called. (4.)
- Then the execution moves back to the first ExecuteSteps plug-in from the first step. The output from the ExecuteSteps, that was originally retrieved from another step, can then be used within the first step. (5.)
- First name, last name, and personnel number, that was retrieved from active directory, is then used in Step 4 – Visual Perso, where the information is printed on an ID card.



Image 5: Example of an APM solution using ExecuteSteps

5 Installation

APM is installed with the standard Windows installer by running the installation package. To install APM you must log in with an account that has administrator privileges. When the installer has finished installation, you must start APM the first time by using the **Run as administrator** option. To do this, hold shift key and click APM icon in start menu or on desktop with right mouse button and select **Run as administrator** from the pop-up menu.

When APM starts, it requests you to select license file. Locate the license file sent by Aventura.

By default, normal users do not have rights to open or run APM solutions because of security reasons. Open Options and User Privilege Management and give at least one user the rights to use APM. Please look at the Options section of this manual for details on how to configure user privileges.

6 Licensing

APM's software licenses are controlled by using license files. A license file contains the validity period of the license and a list of plug-ins included in the license.

There are various licensing options available. Please contact Aventura to find the most suitable solution for your usage.

7 User roles

APM users can be divided into four roles: operators, developers, administrators, and customers:

Operators use preconfigured APM solutions and usually do not need to have advanced technical knowledge. Operators can be, for example, customer service personnel working at a reception desk or ID office. The user interface for operators is designed to be as simple and easy to use as possible.

APM developers create and configure APM solutions. These are many more technical tasks, although APM makes this much easier than accomplishing the same tasks by programming. APM developers are typically IT professionals.

Administrators are IT professionals, who for example install APM software and APM solutions to workstations, manage APM's settings, execution environments and user privileges.

Customers are the people served by customer service personnel or users of a self-service kiosk. A customer can be an employee coming to pick up his ID card. In an operator-run scenario, a customer is also a user of APM, when he is for example asked to enter a PIN code or sign a document with a signature pad.

8 User interface

The APM main window contains everything required to open APM solutions and launch its steps. In addition, it contains icons and menus to access the solution editor, settings, and other functionality.

The upper part of the main window contains the main menu, execution environment selection and icons to access favourite solutions, settings and the About window. The rest of the window is divided into four areas: *Front end*, *Display and input*, *Status and Information*. The *Information* area is displayed only, if there is metadata information or help to display.

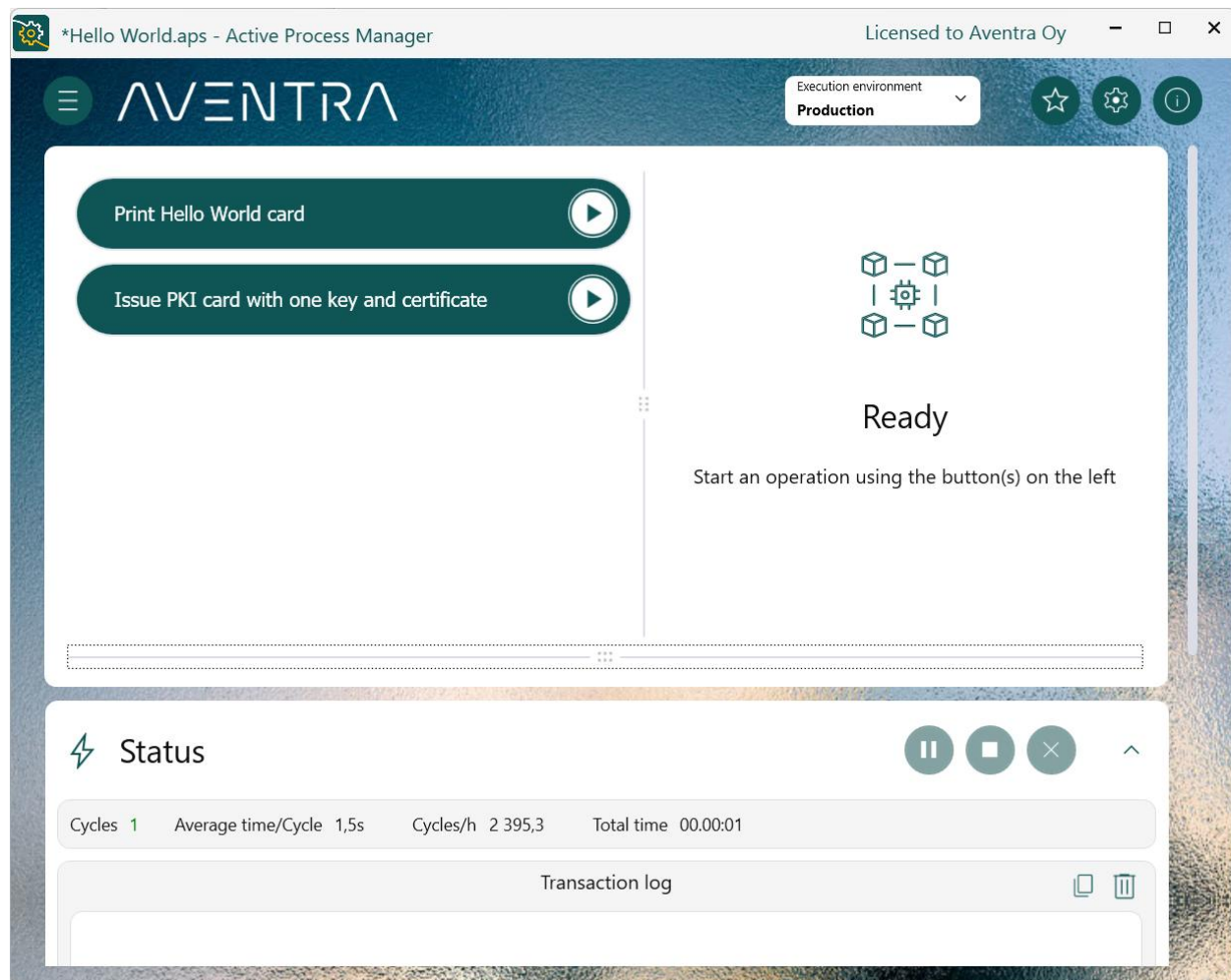


Image 6: APM's main window, Front end view

The *Front end* area is below the main menu on the left. It contains buttons for starting the operations configured in the APM solution. *Display and input* area is on the right.

8.1 Main window sections

8.1.1 Front end

The front end is the user interface that an APM operator uses to start solutions steps. In the front end area, one or more buttons are usually shown. APM automatically creates Front end buttons for steps, which are enabled for starting. APM solution developers can also create customized front end buttons and modify their appearance.

In most cases, a front end button represents a single solution step. However, a button can be configured to run several steps in a sequence.

The buttons typically contain the name of the task they perform, for example ISSUE ID CARDS. An operator simply presses the button by mouse to start the solution step(s) it represents.

8.1.2 Display and input

The main purposes of *Display and input* area are to display information to users during solution execution, and to provide embedded user interfaces when needed. For example, preview of a card being printer is displayed in this area. Plug-ins can show any necessary information, and display user interfaces, for example for selecting a smart card reader to use.

8.1.3 Status

In the *Status* area, APM displays status information of solution execution. On the right inside the area, there are buttons to pause/resume, stop (finishing the solution cycle) and abort (stop immediately) the execution. The *Status* area also contains the transaction log.

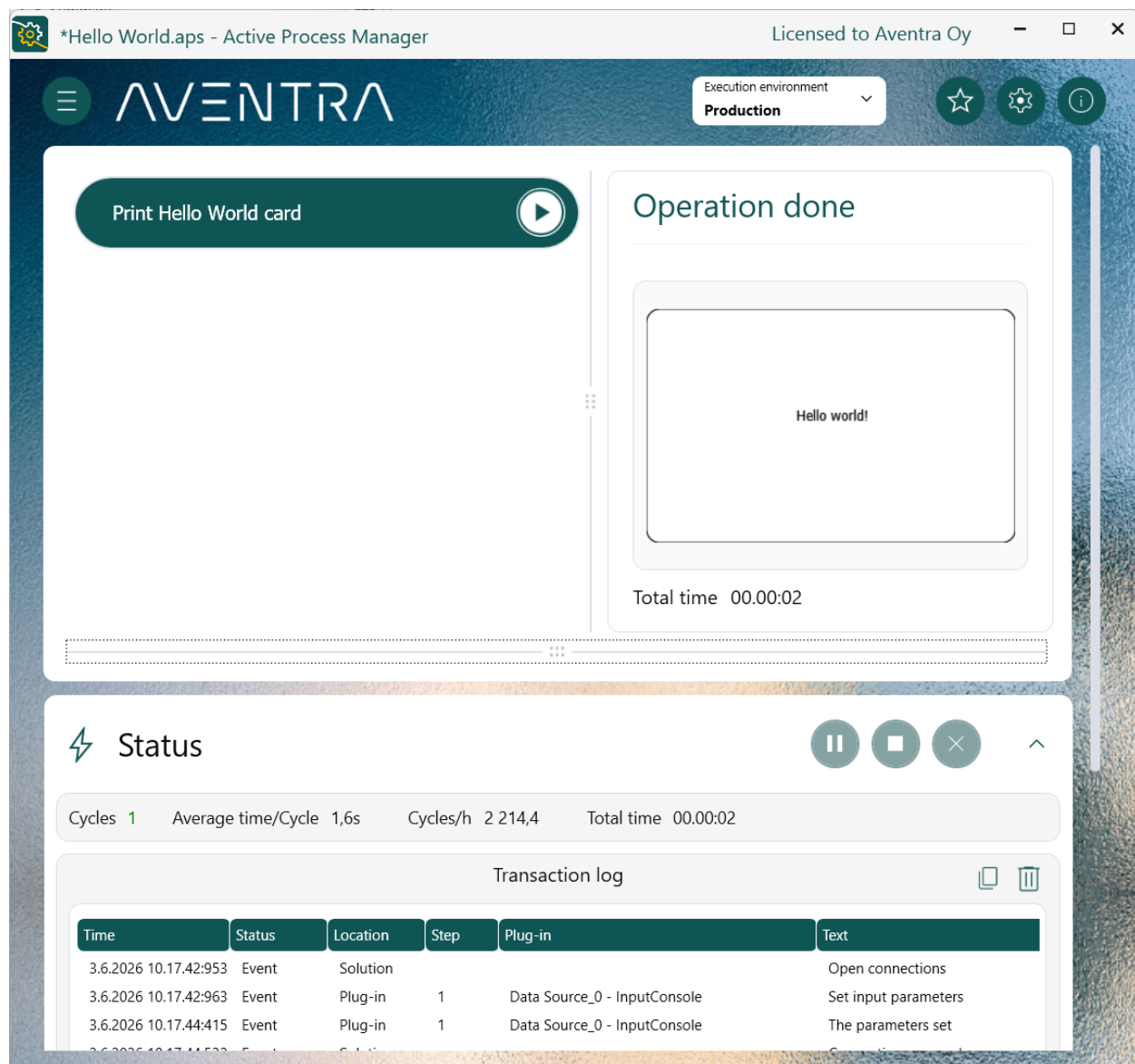
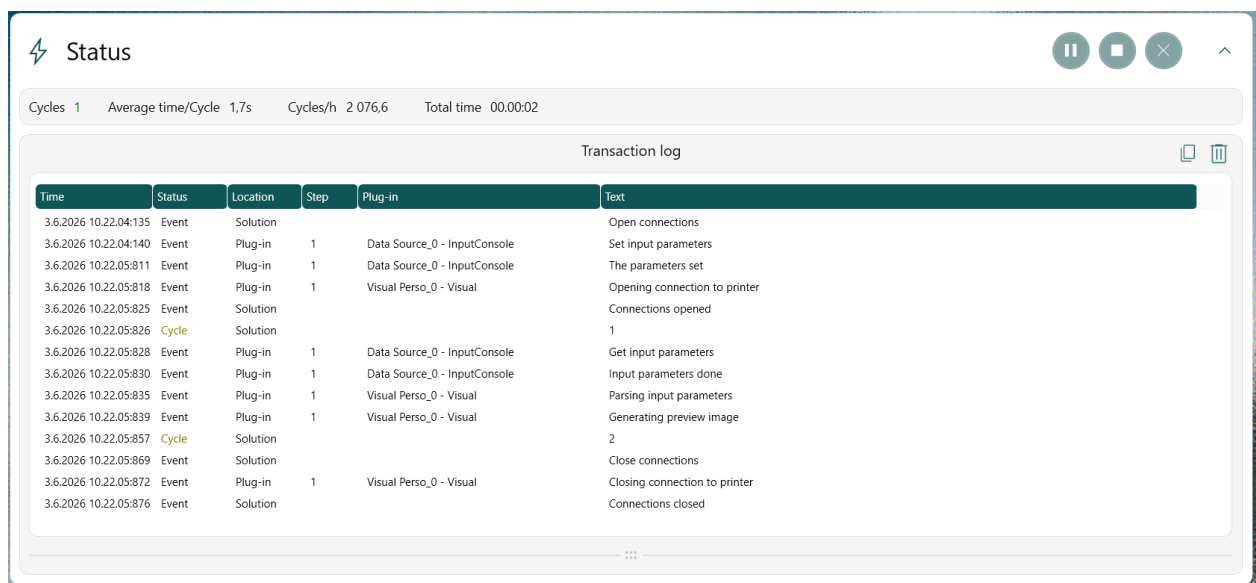


Image 7: APM's main window, Status view

8.1.3.1 Transactions

The *Transactions* subsection under *Status* displays detailed information about the running solution. APM and plug-ins write information about tasks they perform to the transaction log. By looking at the log, you can see what APM is currently doing, and which tasks the plug-ins have performed previously.

Especially in error situations, the transaction log is useful for tracking how the execution proceeded and what went wrong.



Time	Status	Location	Step	Plug-in	Text
3.6.2026 10.22.04:135	Event	Solution			Open connections
3.6.2026 10.22.04:140	Event	Plug-in	1	Data Source_0 - InputConsole	Set input parameters
3.6.2026 10.22.05:811	Event	Plug-in	1	Data Source_0 - InputConsole	The parameters set
3.6.2026 10.22.05:818	Event	Plug-in	1	Visual Perso_0 - Visual	Opening connection to printer
3.6.2026 10.22.05:825	Event	Solution			Connections opened
3.6.2026 10.22.05:826	Cycle	Solution			1
3.6.2026 10.22.05:828	Event	Plug-in	1	Data Source_0 - InputConsole	Get input parameters
3.6.2026 10.22.05:830	Event	Plug-in	1	Data Source_0 - InputConsole	Input parameters done
3.6.2026 10.22.05:835	Event	Plug-in	1	Visual Perso_0 - Visual	Parsing input parameters
3.6.2026 10.22.05:839	Event	Plug-in	1	Visual Perso_0 - Visual	Generating preview image
3.6.2026 10.22.05:857	Cycle	Solution			2
3.6.2026 10.22.05:869	Event	Solution			Close connections
3.6.2026 10.22.05:872	Event	Plug-in	1	Visual Perso_0 - Visual	Closing connection to printer
3.6.2026 10.22.05:876	Event	Solution			Connections closed

Image 8: APM's main window, Transactions log

8.1.4 Information

The purpose of the *Information* section is to display solution metadata, such as version, description and the automatic change counter, and a solution-specific help. APM users can create a HTML-formatted help, which is shown in the *Information area*, to provide usage instructions for users of the particular solution. The *Information* section is not displayed, if all metadata fields are empty and no solution help has been created.

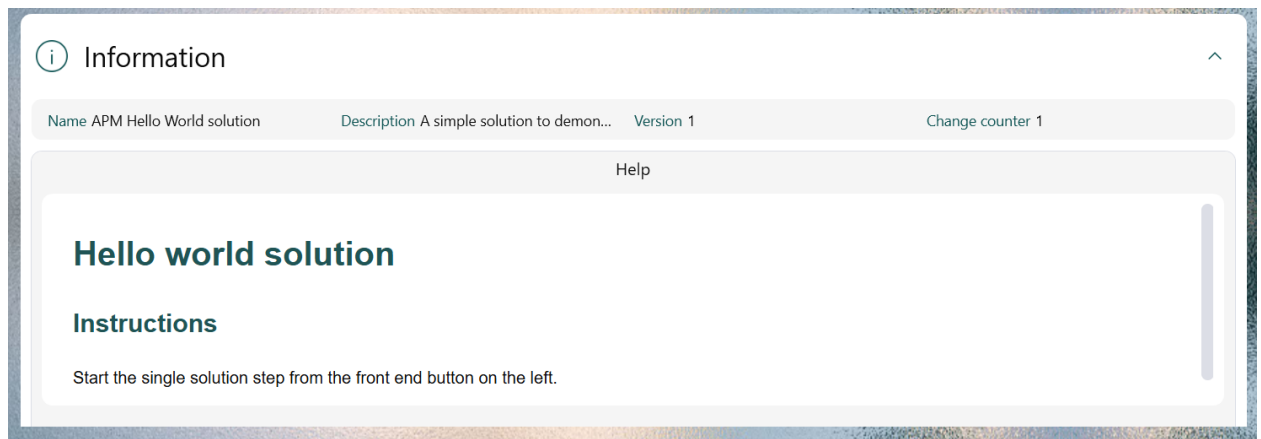


Image 9: APM's main window, Information section

8.2 Main window's menus and buttons

8.2.1 Solution Solution ▾ menu

The solution menu contains the following commands:

- **NEW SOLUTION:** Empties the currently loaded solution if present and creates a new solution.
- **OPEN SOLUTION:** Opens a previously created solution.
- **SAVE SOLUTION:** Saves the current solution as a solution file (.aps). If the solution is new, a file dialog is displayed, and you can specify the file name. If the solution has been saved previously, it is automatically saved with the same file name.
- **SAVE SOLUTION AS:** Saves the current solution, and lets you specify a new file name.
- **CLOSE SOLUTIONS:** Close the current solution.
- **EDIT SOLUTION:** Opens the solution editor.
- **RECENT SOLUTION:** Displays a list of recently used solutions, and lets you quickly select and load one.

- **EXIT:** Closes the Active Process Manager.

8.2.2 Tool buttons

Some of the tool buttons are shortcuts to the same commands that are found in the solution menu. Other buttons are:



PAUSE SOLUTION – pauses execution of the solution.



STOP SOLUTION – Completes the current cycle and stops execution of the solution.



ABORT – Stops execution of the solution instantly. If the plug-in that is running cannot stop instantly (for example because of accessing external hardware), APM stops when the plug-in returns control to APM.



OPTIONS – Opens the Options window.



FAVOURITES – Opens the list of favourite solutions.



HELP – Show About box or enables the changing of the license file.



CLEAR TRANSACTION LOG – Clears the transaction log within the *Status* section of the main window.



COPY TRANSACTION LOG – Copies contents of the transaction log to clipboard.

9 Configuring APM's options

In the Options window you can configure APM's generic settings and open user interfaces to set some specific settings. APM has alternative ways to set the generic settings. The simplest is global configuration for all users. Secondly, users can configure their own individual settings if rights are given. The third way is that the administrator configures individual user specific settings for each user and users cannot change them themselves. The choice can be made separately for each user.

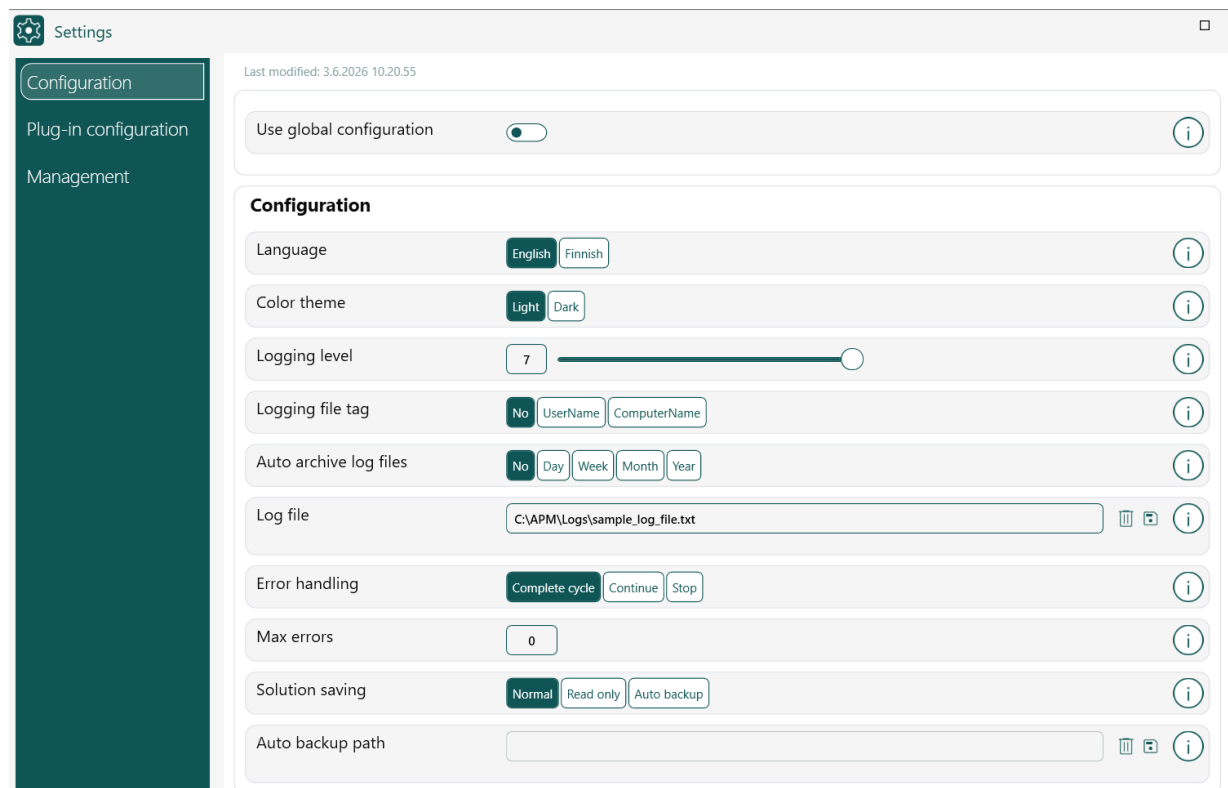


Image 10: Options window, Global configuration

The screenshot shows the 'Options' window with two main sections: 'Startup' and 'Run options'.
Startup section:
 - 'Options' dropdown: 'None', 'Open last solution' (selected), 'Open selected solution'.
 - 'Solution path' text field with a clear button and an info icon.
 - 'Single instance' toggle switch, currently turned on.
Run options section:
 - 'Auto save settings' dropdown: 'Prompt to save changes' (selected), 'Save changes', 'Do not save changes'.
 - 'Favourite path' text field containing 'C:\APM\Test solutions\favorites.apf' with clear, refresh, and info icons.
 - 'Alias path' text field containing 'C:\APM\Alias files\Test Alias File.apa' with clear, refresh, and info icons.

Image 11: Options window, Startup and Run options sections

It is possible to use Windows environment variables in the options, for example setting the Log file as **C:\APM-solutions\log\%USERNAME%-apm-log.txt** would save the log file with a prefix of the username of the logged in user.

In the Options Window, the first two tabs contain the same options. In User configuration, by default the “Use global configuration” checkbox is checked, and the global configuration is used.

Available configuration options are:

1. Configuration

Language – select your preferred language for the user interface. Currently available languages are English and Finnish.

Logging level – select the level of detail, in which the events of the application are saved to a log file.

Log file – file name and path to a file where the events are logged. If no file is specified, no logging occurs.

Error handling – configures how APM should respond to errors, when running a solution. Options are:

- **STOP:** the execution is stopped instantly if an error occurs.

- **COMPLETE CYCLE:** the current cycle is completed leaving a chance to e.g. Data Execute plug-ins to save data to data sources, and to Card Mover plug-ins to move the current card to the printer's reject tray. This is the preferred setting in the most cases.
- **CONTINUE:** allows the execution to continue until the error count reaches the number specified in the next option: **MAX ERRORS**.

2. Startup

Startup options – selects which action to take when starting APM:

- **NONE:** no solution is loaded.
- **LAST SOLUTION:** APM loads the solution that was used the last time when using APM.
- **SOLUTION:** the solution specified in the next option: *Startup path* is loaded.

3. Run options

Auto save settings – Select whether you want to save changes made to the current solution when the solution is started.

Favourite path – Path of the file containing the list of selected favourite solutions.

Alias file – Path of file that contains Aliases. An alias can be used for configuring devices and connections that are environment or workstation specific. See *Execution environment settings* for more information about aliases.

9.1 Plug-in configuration

Under the *Plug-in configuration* tab, you can find information about installed plug-ins. Number of found plug-ins in each configured folder is shown in this view. Press:

- **SHOW** button to view a detailed list of the found plug-ins.
- **SCAN** button to reload plug-in information.
- **ADD FOLDER** to add additional folder to search plug-ins from

By default, plug-ins are installed into folder named “Plug-ins” under the folder containing APM’s executables. This folder is included in Plug-in configuration when APM is installed.

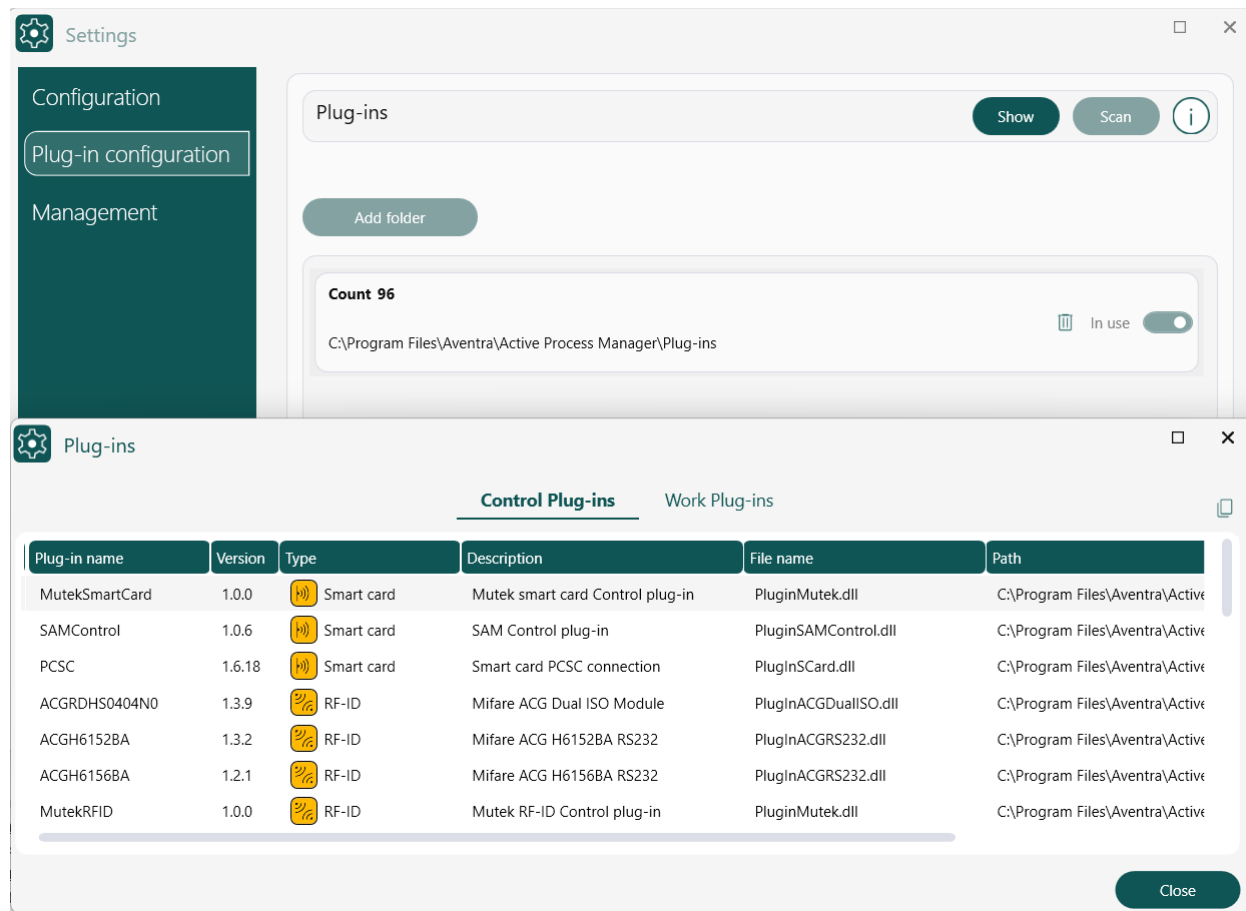


Image 12: Options window, plug-in configuration

9.2 Management

The *management* tab contains various management tools for APM operators and administrators.

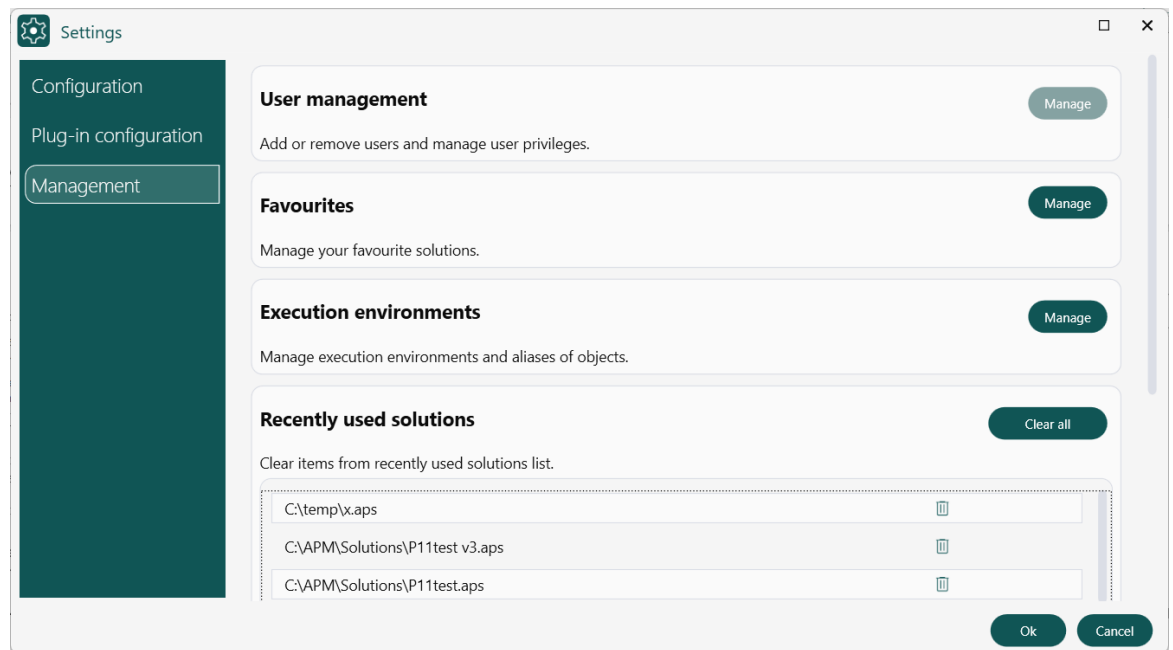


Image 13: Options window, management tab

9.2.1 Clear recently used solutions list

Clear the list of recently used solutions in the Solution menu.

9.2.2 User management

In the user management window, you can add and remove users and modify their privileges. APM's user privilege management is based on Windows or domain user accounts. APM has a list of allowed users. When APM is run, it checks the name of the account that current user is logged in and searches the user list. If the logged in account is not found, the user cannot do anything with APM. If the account is found from APM's list, APM sets the privileges, which are set for this user.

To access user management, start APM with an account that has administrator privileges using the select **Run as administrator** option.

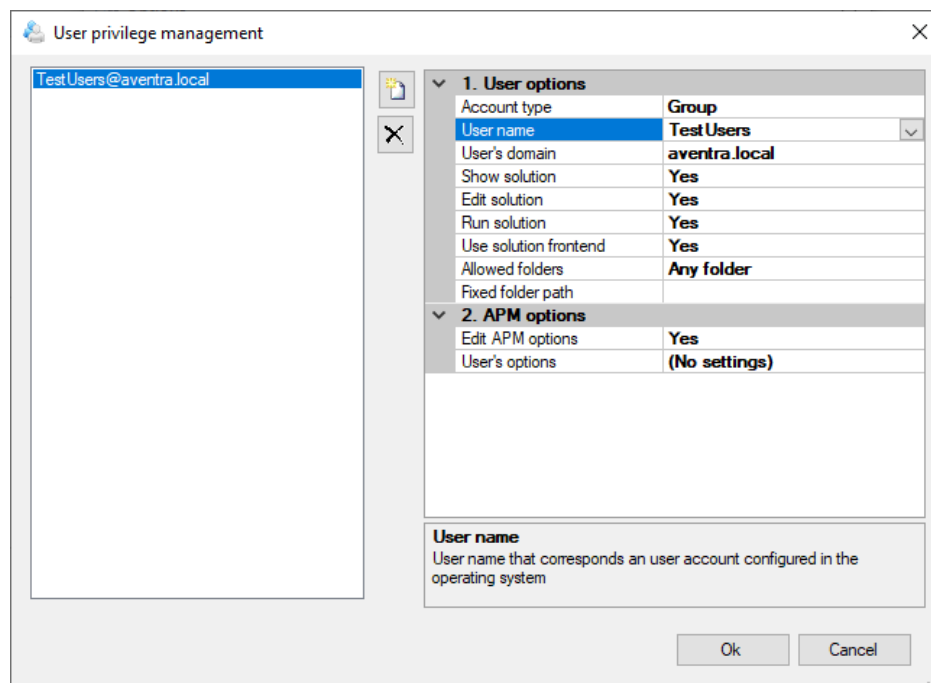


Image 14: User privilege management

List of users is shown on the left side of the window. You can add a new user or remove a user by pressing the buttons on the right side of the list. For each user, the following properties are available:

9.2.2.1 Account type

Type of the account to be given access, this can be a domain user type domain group type.

9.2.2.2 Username

Log-on name of the user. This is the user account name that is used to log on to Windows or Windows domain or a domain group. To allow any logged-on user to use APM, enter ***User*** to the user name field.

9.2.2.3 User's domain

You can optionally set user's domain. If domain is left blank, only username is checked.

9.2.2.4 Show solution

Grant or deny access to open the solution editor for viewing.

9.2.2.5 Edit solution

Grant or deny access to edit the solution.

9.2.2.6 Run solution

Grant or deny access to start a solution by pressing RUN SOLUTION button. The RUN SOLUTION button was removed in APM 4 version, but the access is needed for using front end buttons to start solution.

9.2.2.7 Use solution front end

Grant or deny access to start a solution by pressing a button on the solution's front end.

9.2.2.8 Allowed folders

By default, users are allowed to open APM solutions from any folder. With this property, you can limit the user to be able to open solutions only from a specified folder.

9.2.2.9 Fixed folder path

If allowed folders is set to *Fixed folder*, select the folder at this property.

9.2.2.10 Edit APM's options

Allow or deny editing users own APM options under *User configuration* tab.

9.2.2.11 User's options

By selecting this property and pressing the ... button, a window is opened where an administrator can configure user specific settings.

9.2.3 Execution environment settings

Plug-ins often include settings to be changed when the solution is installed to a new workstation or switched between test and production environments. To make these situations easier for APM developers and administrators, APM has a concept known as aliases. Some plug-in settings can be given an alias, which is replaced with a value specific to the execution environment.

For example, different Registration Authority service points might have different smart card readers. A solution can have many steps where the smart card reader is specified. In PC/SC plug-ins, instead of selecting a specific device, an alias (for example [+Card reader 1+]) can be selected. The same solution can then be distributed to many service points, and the alias value can be set to a specific device at each workstation.

Another example is having a different database for testing and production. You can create aliases for database name and server URL and set the values of these aliases for both environments. By switching the execution environment, the setting is changed in all Data Source plug-in instances where this alias is used.

In the Execution environments window under *Settings* tab, you can define the execution environments you need and configure separate alias values for each environment. The execution environments can be named to describe their purpose, for example “*Production*” and “*Testing*”.

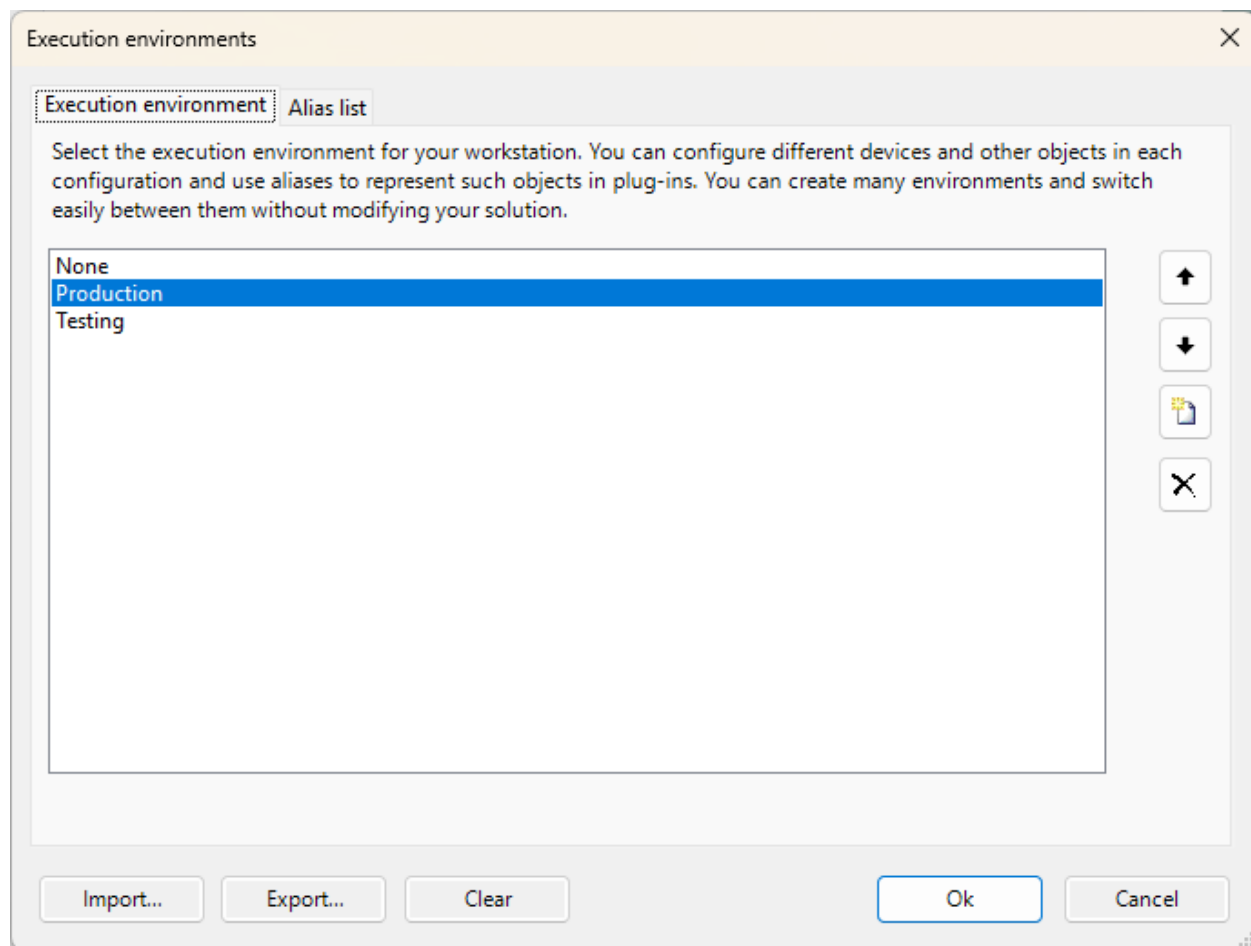


Image 15: Execution environments - Settings

In the *Alias list* tab, you can add, remove, and modify aliases.

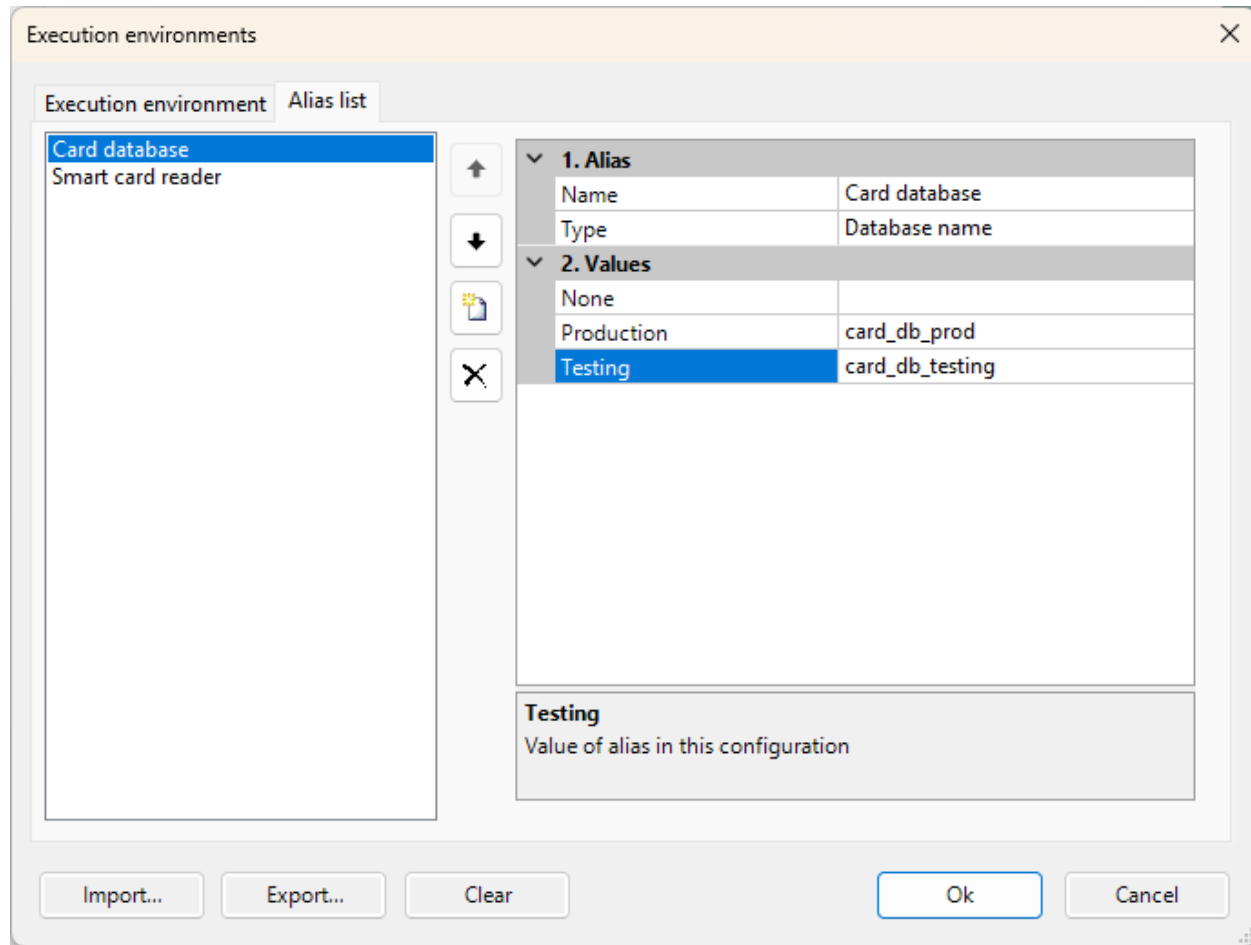


Image 16: Execution environments – alias list

Each alias has the following properties:

Name: The alias name is shown in the plug-ins, where this alias can be used. For example, when selecting the smart card reader to connect in PC/SC Smart Card plug-in, the plug-in would show, in addition to all connected smart card reader, all aliases of type PC/SC in form [+alias name+].

Type: Selects what kind of object this alias represents. This selection defines, which plug-ins show this alias as an option. For example, PC/SC plug-in show aliases of type PS/SC, Data source plug-ins show Database names, URL/Internet addresses and TCP/IP ports when configuring database connections.

Production: Value of the alias in production environment.

Testing: Value of the alias in testing environment.

10 Running APM solutions

An APM solution step is started by pressing a button in the *Front End* area. APM then starts running the (first) step associated with the button.

10.1 Cycles

Running one solution step from top to bottom makes one cycle. After a cycle is finished, APM goes back to the top of the step and starts a new cycle, unless one of the following conditions occurs:

- Maximum execution count has exceeded.
- A control plug-in signals APM to stop running the solution. Typical reasons to stop are: The last database record has been processed, user clicked on CANCEL button on an Input Console or Data Editor plug-in, or the number of cycles set in Input Console has been exceeded.
- An error has occurred in a plug-in (configuration options define whether APM continues in this situation)

10.2 Solution state

When an APM solution is running, APM is in one of the following execution states: Success, Failed or Cancelled

When a solution is started, it is initially in the Success state. If an error occurs in one of the plug-ins, the solution is switched to Failed state. Depending on the configured options, APM either stops or continues. If APM is configured to continue when a plug-in fails, the subsequent plug-ins can still perform various actions (for example updating “failed” state to a database record or reject a smart card to reject tray of a printer). In error situations APM can be configured to stop immediately, run until end of the current cycle, or continue to the next cycle until maximum number of errors is reached.

Cancelled state is used for example, when an operator cancels the operation in APM’s user interface. Input Console and Data Editor plug-ins contain cancel buttons. When a plug-in cancels APM’s execution, next plug-ins are not executed, but their

CloseConnection methods are called to let plug-ins cancel their work and close their resources cleanly.

APM solution can also be stopped or aborted from the buttons in the main window.

11 Creating APM solutions

Creating an APM solution has a lot in common with creating a computer program with a programming language. In fact, developing APM solutions could be called visual programming.

In an APM solution step plug-ins can be placed in any order and there can be unlimited instances of the same plug-in with individual configurations. Execution of a step flows normally from top to bottom, but plug-ins can be skipped, and execution can jump to another position on specific conditions. Plug-in parameters are like program variables. However, the logic is created by dragging plug-ins and parameters with the mouse and filling in values to graphical property grids.

A new solution is created by selecting NEW SOLUTION from the Main menu or pressing Ctrl + N. The solution editor opens with a blank solution. In the solution editor most of functions for editing a solution can be accessed by two ways; using the menu at the top of the editor window or by pressing the right mouse button over an item in the solution tree, which opens a pop-up menu. These menus contain functions for adding, configuring, deleting and copying plug-ins and steps. In the *editor* window, there is also a menu named *Solution*. This menu contains solution specific actions like setting up solution's properties and front ends.

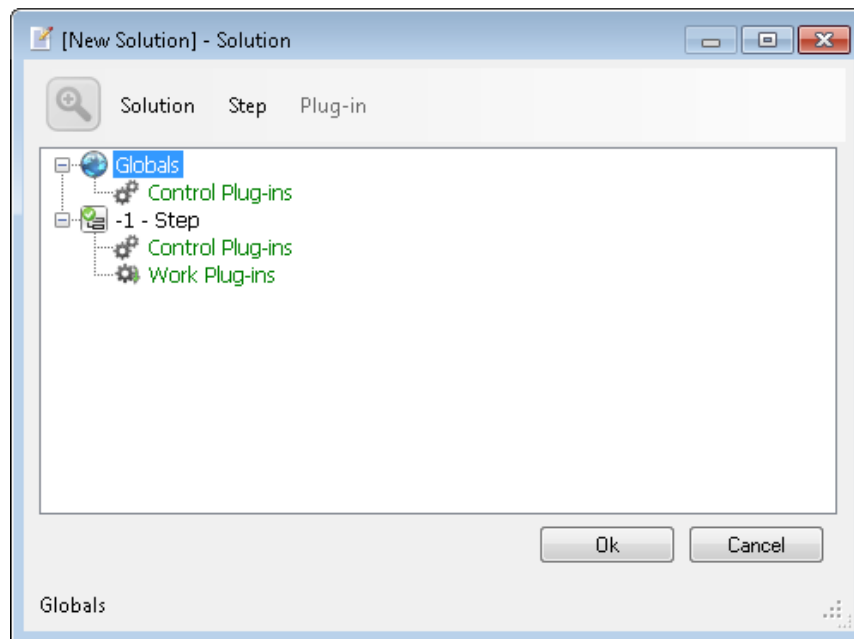


Image 17: a blank solution

11.1 Flow view

APM version 4 introduced an alternative user interface to view the solution: Flow view. The flow view can be accessed from the main menu. In the flow view, you can view and navigate the solution in a flowchart-style representation. At time of writing, the editing capabilities in the flow view are limited, but are going to be expanded in future. In the flow view, you can see the flow of the solution and branching based on preconditions such as parameter values, and you can navigate between different levels of the solution, e.g. from a Front-end function to internals of the step, and to other steps called from the step using the ExecuteSteps plug-in.

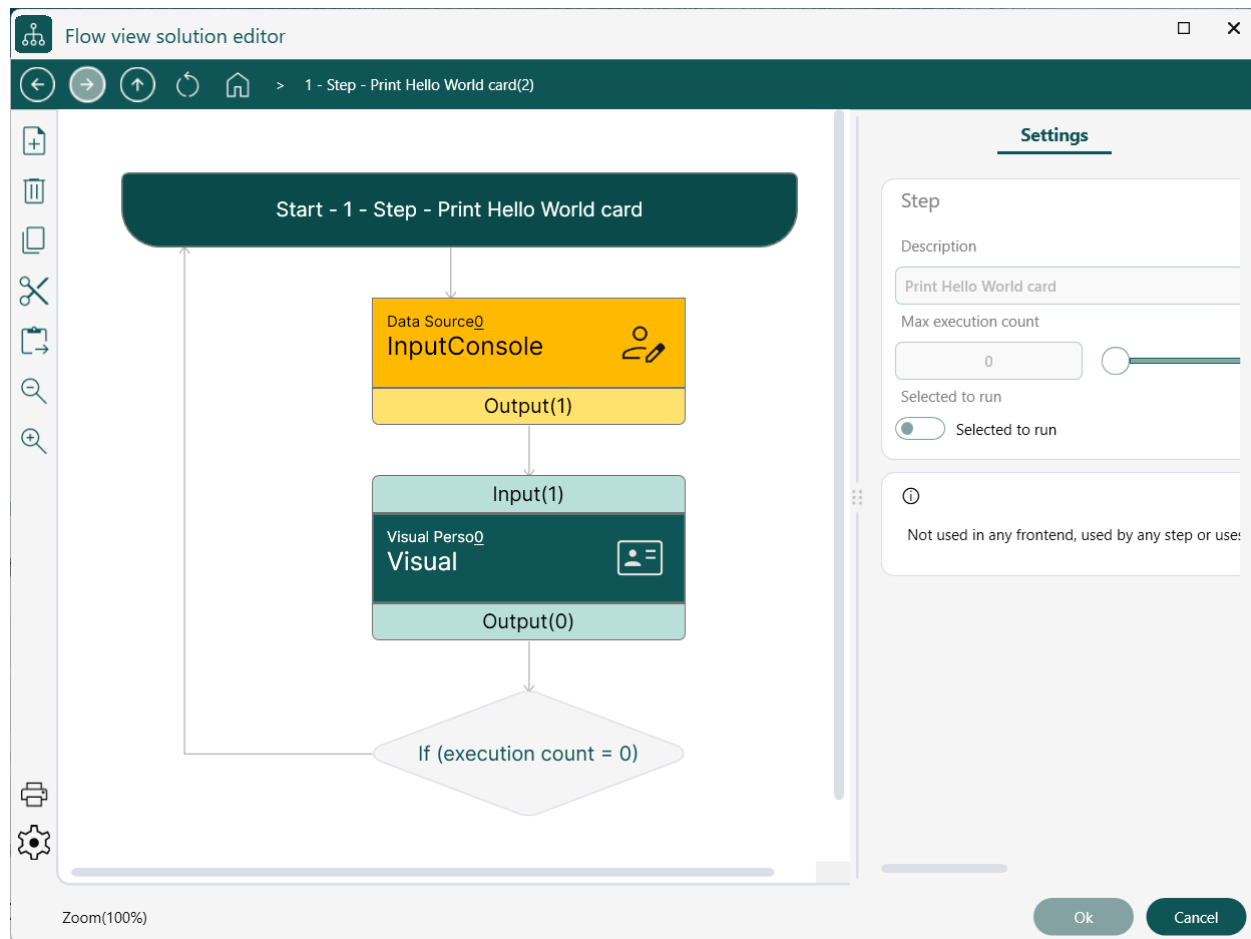


Image 18: Flow view editor

11.2 Globals section

At the top of the solution there is a section called *Globals*. In this section you can insert plug-ins, which can be used in any step of the solution. E.g. If there are many steps in a solution connected to the same database, all steps can share the same instance of a DataSource plug-in, and you need to configure the database connection only once. Global plug-ins are always control plug-ins. If you click the *Globals* section with the right mouse button and select CONFIGURE, you can select when the global plug-ins are executed. By default, they are executed always, but you can also select that executed only when running a step linked to them. You can also bypass the global plug-ins, which can be useful for testing purposes.

11.3 Adding steps and plug-ins

Below the Globals section, there is initially one step. From the right mouse button pop-up menu, you can select CONFIGURE to set up step specific properties.

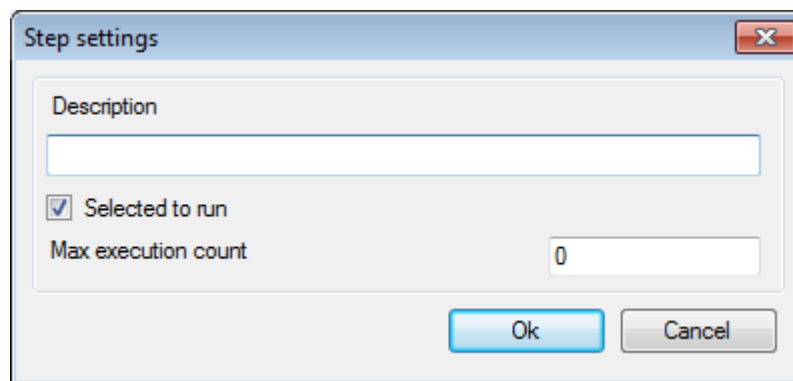


Image 19: Step settings

To description field you can type the function of this step, for example Issue ID cards. The description is displayed in solution three. If SELECTED TO RUN checkbox is checked, this step can be started from the front end view, and a front end button is created for the step automatically, unless a custom front end has been configured.

MAX EXECUTION COUNT defines how many times the step is run when started. If zero, the step is run continuously until the operator stops it in the user interface, or some control plug-in signals execution of the step should be stopped. In many cases data source or input console plug-ins control how many cycles are run. However, if the step is always used to perform a single task in one cycle or a specific number of cycles, you can set the number of cycles.

11.3.1 Adding plug-ins

You can add a plug-in to the solution by opening the pop-up menu with the right mouse button. The plug-ins are divided into subcategories according to their purpose. E.g. available control and work plug-ins are shown in their own categories.

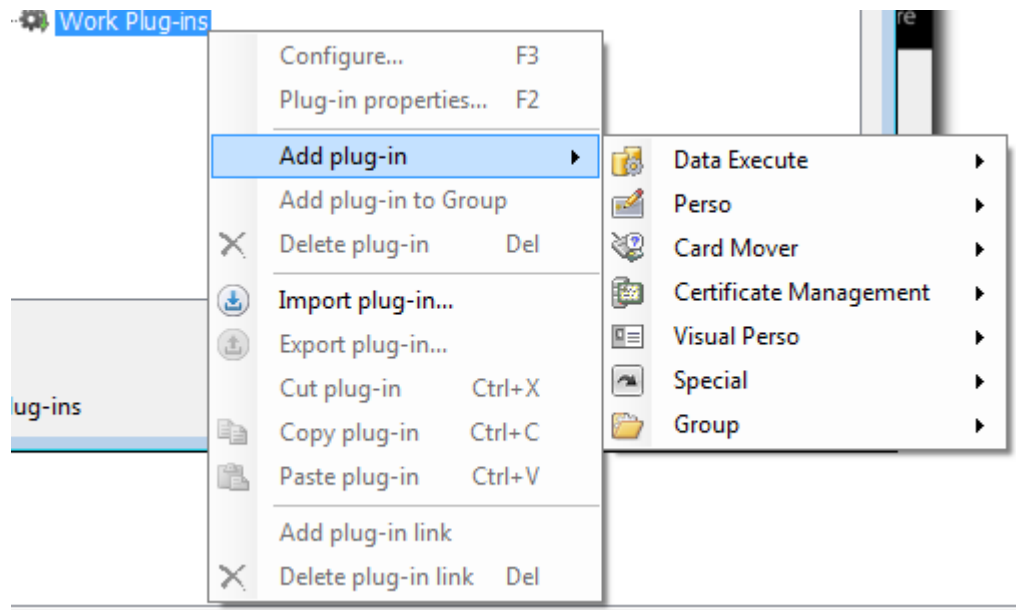


Image 20: Adding a plug-in to a solution step

11.3.2 Configuring plug-ins

After adding a plug-in you have to configure it by using own configuration window. There you can set the properties that affect how APM handles the plug-in.

In the plug-in's pop-up menu, there are the items CONFIGURE and PLUG-IN PROPERTIES. They open configuration windows, which serve different purposes.

The CONFIGURE menu item opens the plug-in's own configuration window. This configuration window is displayed by the plug-in itself – the window comes from the plug-in DLL. The plug-in can show any kind of user interface that is required to configure it. PLUG-IN PROPERTIES in turn opens a window, where you can set common properties all plug-ins like the plug-in's description.

11.3.3 Configuration window of a plug-in

With a few exceptions the configuration windows of different plug-ins have quite similar layouts. They usually contain at least two tabs, one for generic options and one for the plug-in's parameters. Options are displayed in a grid, where the option's name is on the left and the selected option on the right side. At the bottom of the view there is a help section. When an option is selected, you can find information about the meaning of the options and the possible choices in the help section.

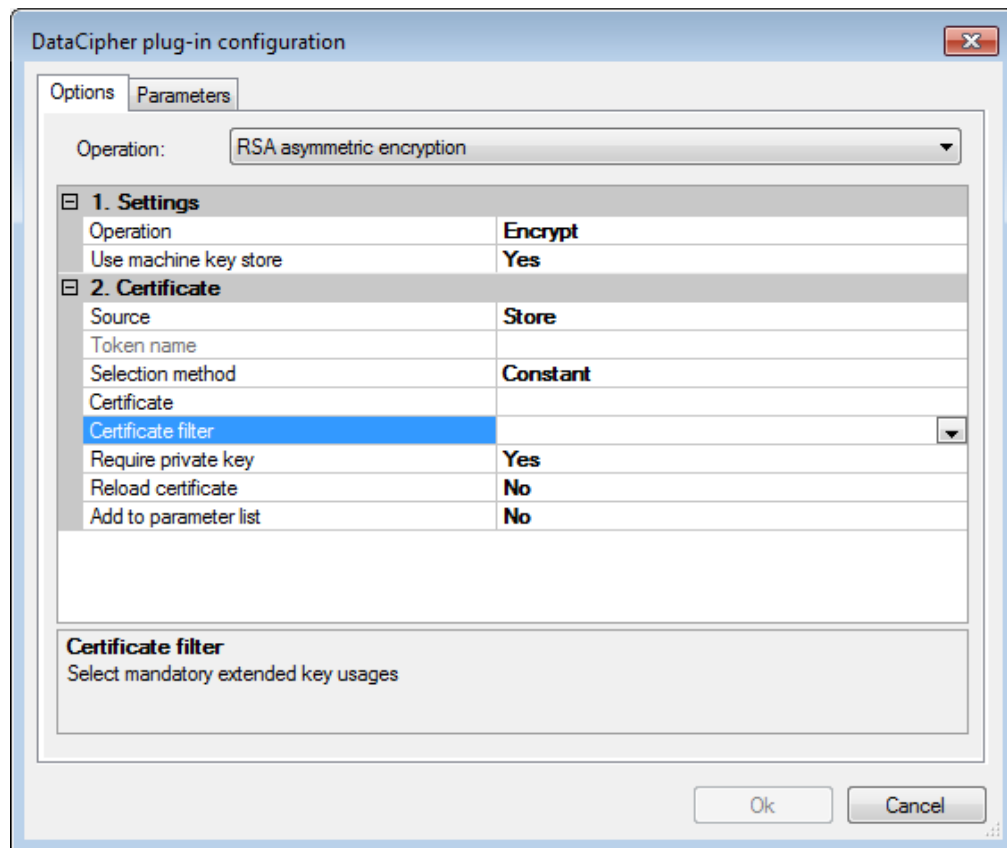


Image 18: configuration window of a plug-in

Under the *Parameters* tab is a list of parameters and a grid like in the options tab, where properties of each parameter are configured. There are also buttons to add, remove and move parameters.

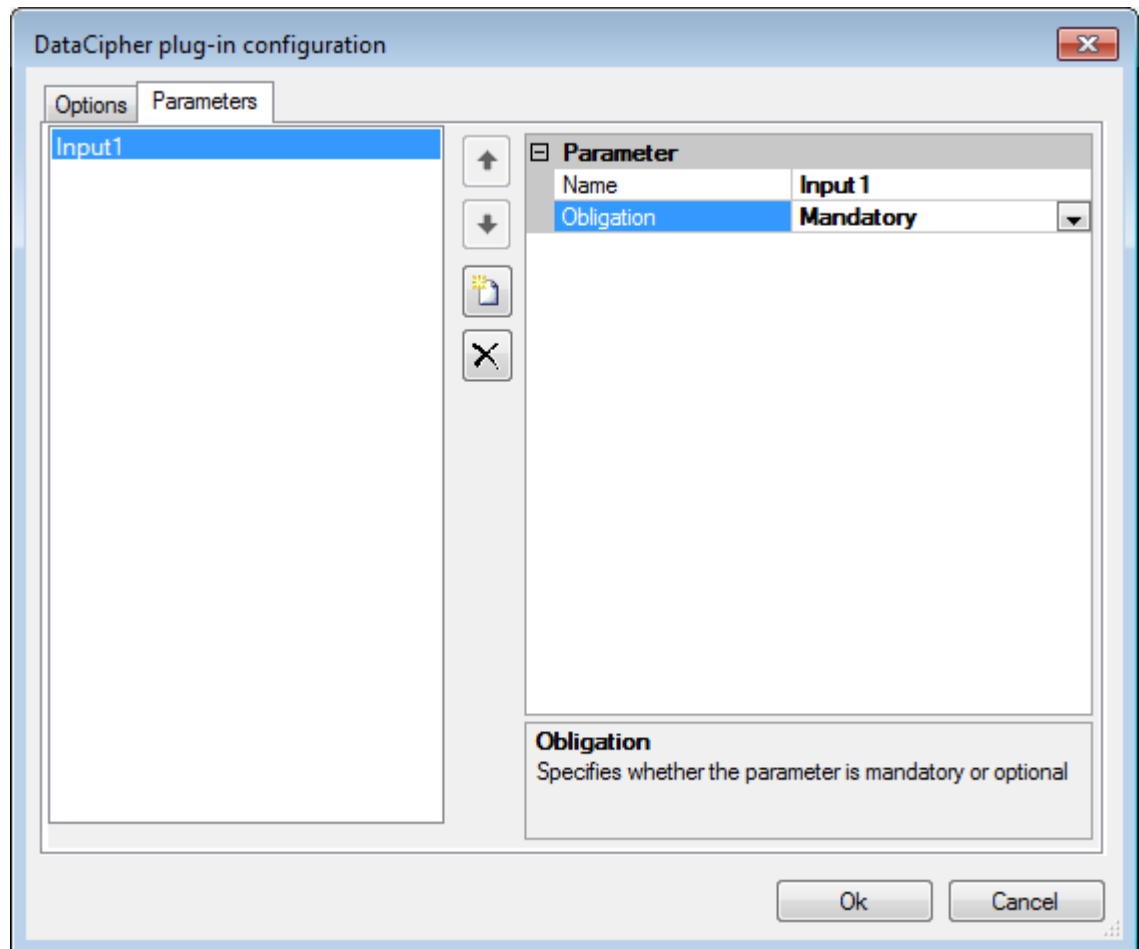


Image 21: Configuring parameters of a plug-in

In some plug-ins the standard layout is not enough for configuring it and the configuration window can be more complex. For example, Visual plug-in contains a graphical layout editor.

11.3.4 Plug-in properties

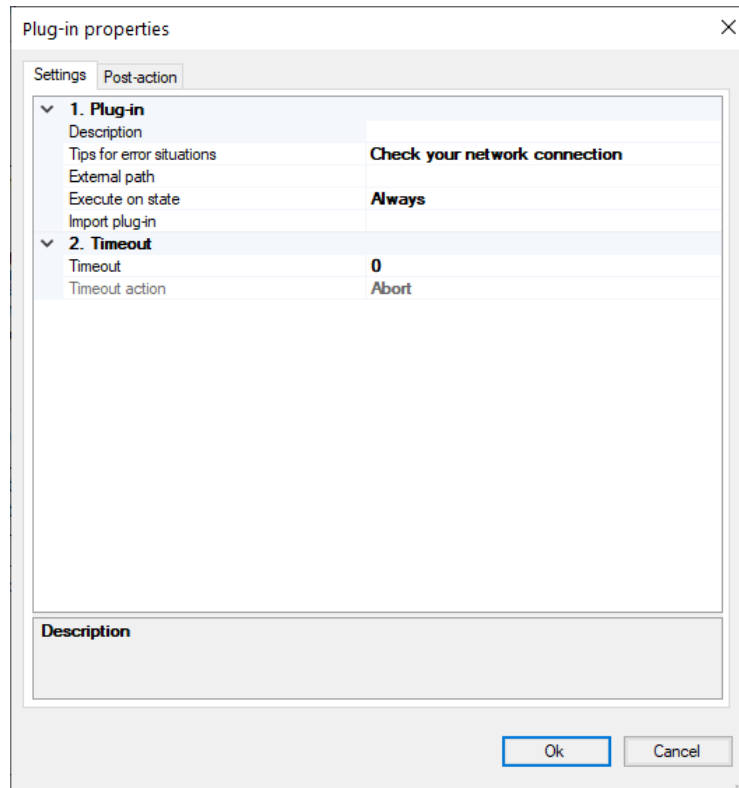


Image 22: Plug-in properties

In the *properties* window, you can fill in a set of properties, which are common to all plug-ins.

11.3.4.1 Description

It is useful to enter a description for the plug-in. Description is displayed in the solution tree next to the plug-ins type, for example “DataExecute_1 – SQLQuery – Check number of users whose IDs need renewal.” or “Perso_0 - PKCS15Update – Generate a 2048-bit RSA key on smart card.”.

When all plug-ins have descriptions, it will be easier to understand and follow the process flow of the step in the solution tree.

11.3.4.2 Tips for error situations

In error situations APM displays error messages, which usually originate from plug-ins in the solution. These error messages are often quite technical.

This property is intended for an error message that is more useful for the operator, such as a tip on how to circumvent or fix the problem that might have caused the error. The following example shows how this property could be used.

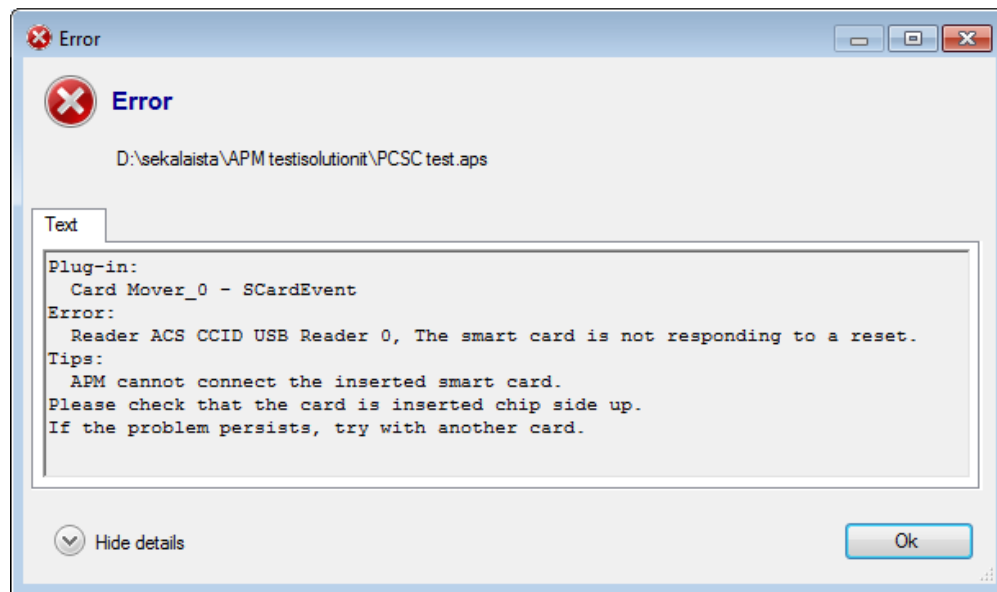


Image 23: Error message

11.3.4.3 External path

You can save a plug-in to an external file instead of the APM solution file. For example, you have a solution that is used for photographing personnel and issuing ID's. The same solution is used at many sites. Settings of the camera plug-in are different at each site due to photo shooting conditions are different. You can keep the camera settings in an external site. If the solution is updated, you can just replace the APM solution file at each site, but the site-specific Camera settings remain.

11.3.4.4 Execute on state

Select in which solution states the plug-in is run. Most commonly it is useful to set this value to "Success". This way the plug-in is skipped if a previous plug-in has failed. The other choices are also needed for many situations. E.g. a SQLExecute plug-in can have different SQL statements for Success and Failure states, and sometimes a plug-in could be used for handling only the Failure state.

11.3.4.5 Import plug-in

You can import a plug-in configuration, which has been saved to an external file. This can be used to copy a plug-in instance from another solution.

11.3.4.6 Timeout

You can set maximum time in seconds, in which APM expects the plug-in to finish its work.

11.3.4.7 Timeout action

If the timeout is exceeded, you can select one of these actions:

- Abort: stop the plug-in and the solution immediately
- Cancel: notify the plug-in to stop cleanly, the plug-in should return Cancel status
- Continue: notify the plug-in to interrupt its work and return cleanly, APM continues normally

11.4 Post action

With the post action feature it is possible to perform certain additional functions, which depend on the result of the plug-in or the selection of the operator in a dialog box. For example, if a failure occurs in a certain plug-in, it is possible to change the failure result of the plug-in to success and then jump to another plug-in. This allows easier error management and opens up new possibilities for configuring APM.

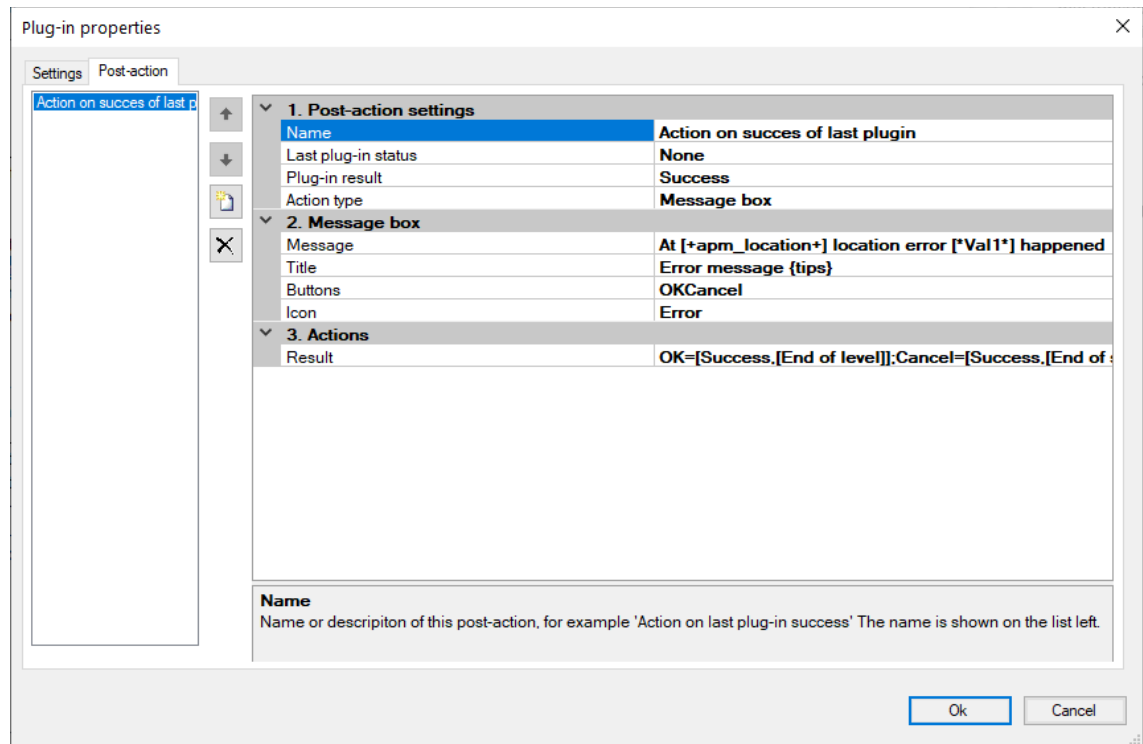


Image 24: Result action editing window

Within the result action editing window, you can configure the desired action in the following ways:

Description

It is useful to enter a description for the post-action function. This description is shown in the list on the left side.

Last plug-in status

This setting defines what the last plug-in status needs to be for post-action function to be performed.

Plug-in result

This setting defines what the plug-in result in the attached plug-in needs to be for the post-action function to be performed. You can select more than one plug-in result and configure a different action for each result.

Action type

Action type sets to what the post-action is based on. It can be either the result of the executed plug-in or the selection of the operator in the message box.

Message

This is the message shown to the operator when the message box action type is selected. It is possible to add text from aliases using `[+ and +]` around the alias name or global variables using `[* and *]` around the variable name.

Title

Title of the message box is defined here. It is possible to use the error tips set in the plug-in configuration using `{tips}`.

Buttons

Setting allows to define the different buttons shown in the message box.

Icon

This specifies the icon shown in the message box.

Result

This is used to define the result of the post-action. This can be set independently for each button in the message box or for each plug-in result. It is also possible to define different plug-ins to jump to for each button in the message box or for each plug-in result.

2. Message box	
Message	At [+apm_location+] location error [*Val1*] hap
Title	Error message {tips}
Buttons	YesNoCancel
Icon	Error

3. Actions	
Result	s=[Success];No=[Success];Cancel=[Success]...

Dialog result	Plug-in result	Jump destination
Yes	Success	Group_0
No	Success	Perso_0 - Testing 0
Cancel	Success	[End of step]

Image 25: Post action result editing window

11.5 Linking plug-ins

If a work plug-in requires control plug-ins, types of the required plug-ins are shown in red below the work plug-ins name. Drag the required control plug-in over the work plug-in type while holding the left mouse button. Alternatively, you can link to a control plug-in from the right button pop-up menu, which is useful in large steps.

11.6 Linking parameters

Parameters are linked by dragging, or in the pop-up menu like plug-ins. You can link a parameter from a control to a work plug-in or from a work plug-in to another work plug-in. Parameters must be linked from up to down in the solution tree, so that the output parameter is always above the target input parameter.

Locate the output parameter from the source plug-in and drag it over to the input parameter in the target plug-in. Alternatively, open the pop-up menu over the input parameter by clicking on the right mouse button and select ADD PARAMETER LINK. Follow the appearing submenus to locate the source output parameter.

If the data type of the input parameter is different from the type of the source output parameter, APM performs an automatic conversion. It is possible to link any output parameter to any input parameter, but many plug-ins expect specific types of data. It is each plug-in's responsibility to validate the incoming data of its input parameters during solution execution and return an error message if the data is invalid.

If you have many parameters which need to be linked between two plug-ins, you can launch the parameter linking window, where you can link the parameters more quickly. To open parameter linking window, hold ALT key and drag the source plug-in over the target plug-in holding left mouse button. The window opens, and you can drag the parameters you want to link from left to right or remove links using Delete button on keyboard.

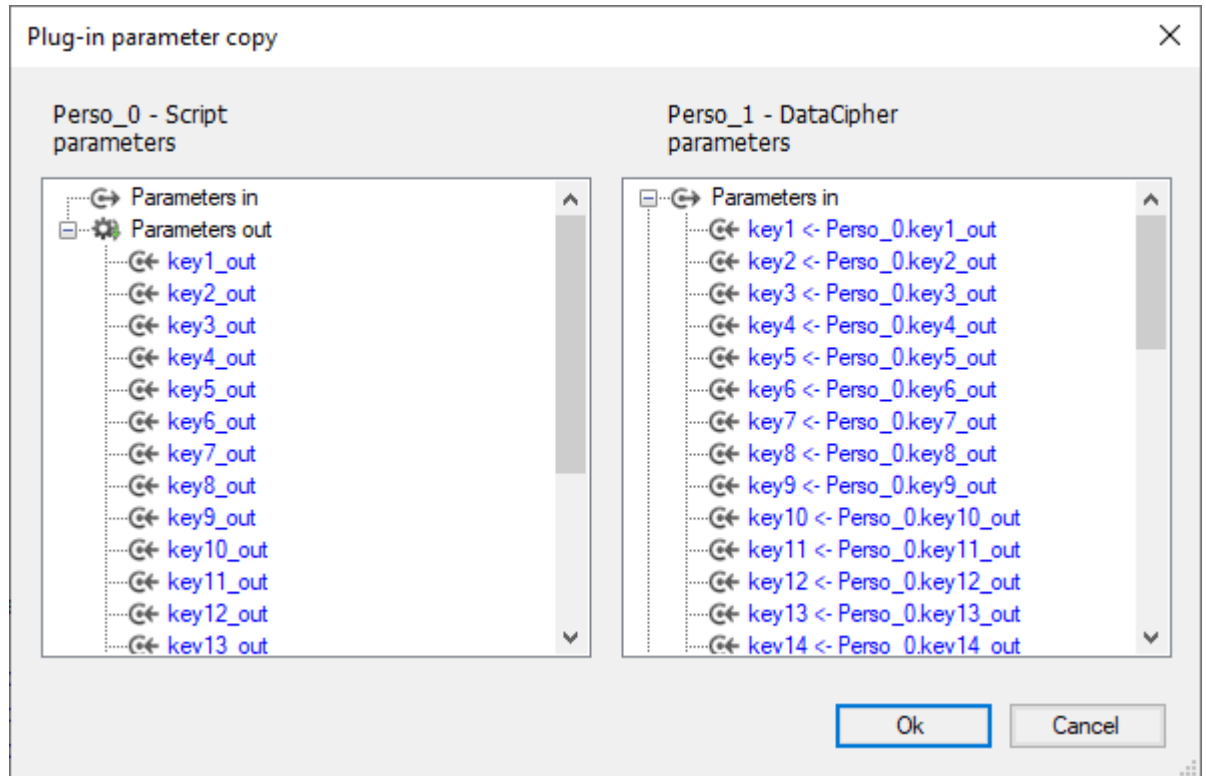
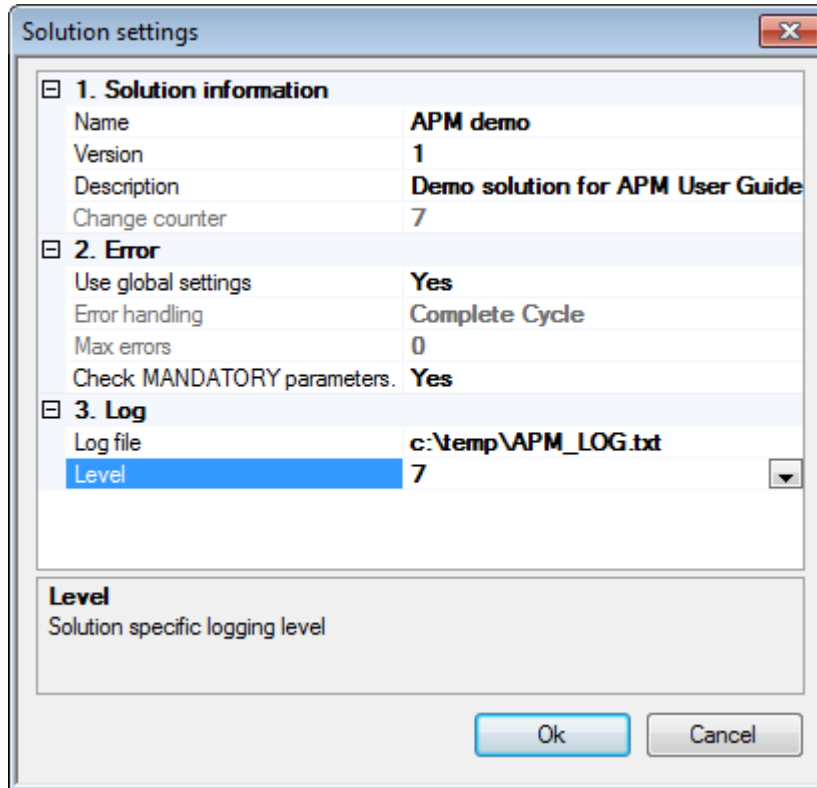


Image 26: Parameter linking window

11.7 Setting up solution properties



Solution settings	
1. Solution information	
Name	APM demo
Version	1
Description	Demo solution for APM User Guide
Change counter	7
2. Error	
Use global settings	Yes
Error handling	Complete Cycle
Max errors	0
Check MANDATORY parameters.	Yes
3. Log	
Log file	c:\temp\APM_LOG.txt
Level	7
Level Solution specific logging level	
Ok Cancel	

Image 27: Solution properties

11.8 Setting up Front ends

Open *Front End* from the Solution menu to edit it. In this view, you can add or remove front end buttons and set their properties. Each front end button represents either a single step or a set of steps, which are executed in sequence.

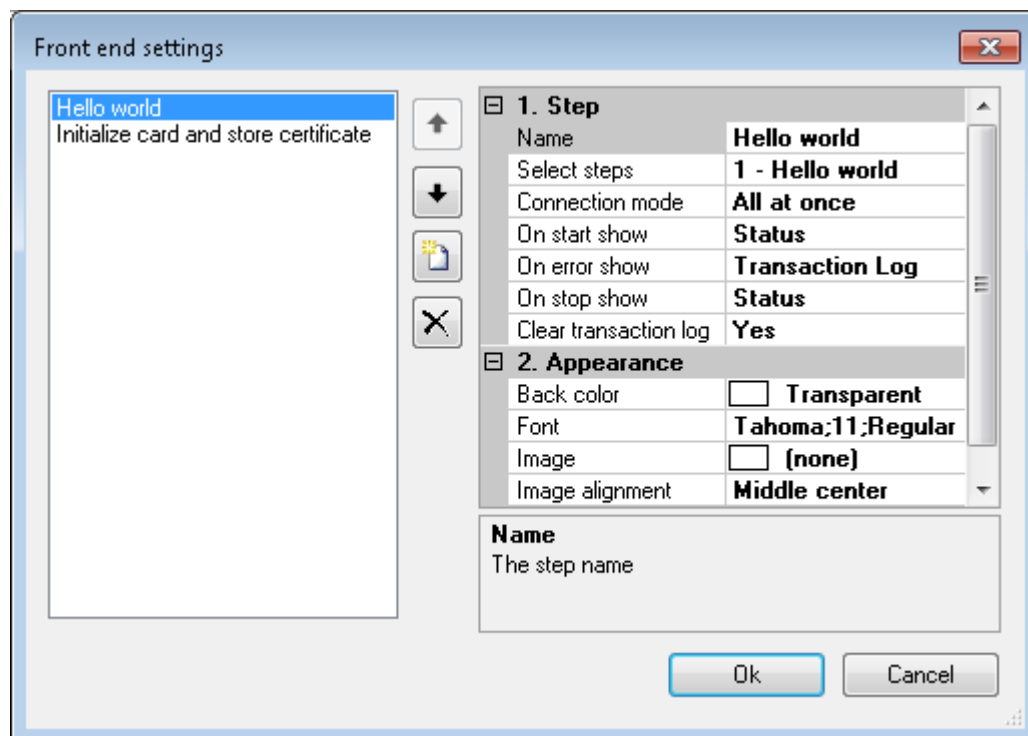


Image 28: Setting up front end

Name

Name of the Front end button, i.e. the text that is displayed in the button.

Select steps

Selects one or more steps, which are run when the button is pressed.

Connection mode

Selects how APM opens the connections to plug-ins when multiple steps are selected.

All at once: APM opens connections of all plug-ins, when the front end button is pressed and closes them after all steps have been run.

Step by step: APM runs the steps separately - Open Connection methods are called at the beginning of each step, and the connections are closed after each step.

If only one step is selected, this setting is redundant.

On start show

Selects which view is shown when execution is started.

On error show

Selects which view is shown if solution is stopped because of an error.

On stop show

Selects which view is shown, when the execution of the solution is stopped by operator, or the work is completed.

Clear transaction log

Selects whether the transaction list (under Transactions tab) is cleared when execution is started.

Back color

Background color of the front end button.

Font

Font used in the front end button.+

Image

You can select an image file to be displayed in the button.

Image alignment

With this property you can control how the image is positioned inside the button.

Text color

Text color of the front end button.

12 Command line parameters

APM can be run from the command line by navigating to the application folder and running ActiveProcessManager.exe. The following command line parameters can be set when starting the APM application. All parameters are optional. Command line parameters override the equivalent parameters configured in APM's settings.

Parameter	Description	Example
-load [solution file path]	Open the specified solution when starting APM.	-load "c:\APM-solutions\MySolution.aps"
-alias [alias file path]	Open alias file from the path provided in the command line parameter.	-alias "c:\APM\settings\MyAliasFile.apa"
-license [license file path]	Choose which APM license is being used.	-license "c:\APM\licenses\MyLicenseFile.alf"
-execute [solution file path] [step]	Load and execute a specified APM solution. The parameters define the solution to load and which step to start.	-execute "c:\APM-solutions\MySolution.aps" 1
-executeExit [solution file path] [step]	Load and execute a specified APM solution. APM shuts down when the solution execution is finished. The parameters define the solution to load and which step to start.	-executeExit "c:\APM-solutions\MySolution.aps" 1
-plugins [plugins folder path]	Set the folder where APM loads plug-ins from.	-plug-ins "c:\APM\Plug-ins"
-logFile [log file path]	Choose where the log file is written.	-logFile "c:\APM\Logs.txt"

-logLevel [1-7]	Choose how much information is written into the log file.	-logLevel 5
-help, -?	Display all the commands in the terminal.	-help
-normal, -hidden, -minimized, -maximized	Choose the window state that the application opens in.	-maximized

Example of a valid command	ActiveProcessManager -maximized –execute “c:\path\to\solution.aps” 1
----------------------------	--

13 Calling APM from other software

Sometimes it is useful to call APM from other software to do some processing in background without showing APM’s user interface. It is possible to integrate APM with other software using a DLL interface or HTTP REST API. The calling software can pass parameters to APM and receive parameters, which APM returns after completing its task.

Please contact Aventra for more information if you have need for this feature.

14 Kiosk mode

There is a special feature in APM for creating self-service user interfaces. This is called the kiosk mode. In this mode, APM is run on full screen. Kiosk mode's user interfaces are typically run on a touch screen tablet PC, which might be embedded into a self-service kiosk.

Typical usage scenarios for kiosk mode are:

- card issuance
- updating issued cards
- self-service photography
- user registration

A kiosk mode APM is run from a separate executable (ActiveProcessKiosk.exe). In production environment, this process should be set to auto-start when the operating system starts. When the kiosk process is run under an user account with no admin-privileges, the user cannot exit the kiosk mode.

This executable file starts running the APM solution defined in Configuring APM's options, in the Startup Solution option. This way end-user can only use APM to execute the selected solution. If no solution is selected to run from startup, only blank screen appears. Setting up solution to run from startup is done in Options -> Startup solution path -> and then selecting solution file (.aps)

14.1 Configuration

Kiosk mode can be configured by running ActiveProcessKiosk.exe in **Run as administrator** option. Only then user can use these keys below:

- F2=Configuration
- F3=Show transactions
- F5=Start solution
- F6=Abort solution
- F12=Exit

Pressing the F2 key opens the Configurations window, which contains three tabs: Configuration, Panel, and Controls. In **Configuration tab**, user can change the visuals of the background. In **Panel tab**, user can add new visual elements to plug-ins. Each panel can be associated with a plug-in, and the panel is shown when the plug-in's execution starts, during the plug-in's execution, or right after the plug-in's execution is finished. And In **Control tab**, user can define element's type and appearance. Configurations are saved to APM_kiosk.xml file.

© Copyright Aventura Ltd, 2026. All rights reserved. The contents of this document are subject to copyright. Any changes, modifications, additions or amendments require prior written consent from Aventura Ltd. Reproduction in any form is only permitted on the condition that the copyright notice remains on the actual document. Publication or translation in any form requires prior written consent from Aventura Ltd. Trademarks are the property of their respective owners.