

Quick Guide - How to start testing CPOS

In this document we will go through what has to be done/prepared to be able to record/play the scripts on CPOS environments.

After all of this is done we can now start to configurate everything in Executive Automats.

First, we need to create System Definition for our CPOS instance.

Go to Settings → System Definition tab and hit button +ADD.

Create system definition

* Environment:	CPOS System Definition
* Environment address:	https://URLofCPOS.com/
* Application name:	CPOS
* Environment type:	Test
Description:	Enter description
Multi-window support:	<input type="checkbox"/>
Disable resource sharing:	<input type="checkbox"/>
* System platform:	Cloud POS

Environment - name of the record,

Environment address - here paste the link that was for CPOS instance that you want to test.
NOTE: Link has to end with “/” or it can cause problems when trying to access the environment by player/recorder.

Application name - name of the application,

Environment type - tag that tells other users what kind of environment it is.

System Platform - we have to select here record **Cloud POS**.

After filling those details, click create.

Second thing that we need to create is Virtual User.

NOTE: We cannot utilize the same Virtual User as for Dynamics F&O environments.

Go to Settings → Virtual User tab and create +ADD.

Create virtual user

* Name:	VU for CPOS
Description:	Enter description

System authentication

Enable system authentication:

Manual mode:

Remember sensitive data:

Custom script:

* Platform:	Cloud POS
* Login:	AAD account login/Email
* Password:
* Cloud POS URL:	https://URLofCPOS.com/
* Device ID:	DeviceID
* Register number:	RegisterNumber
* Operator ID:	OperatorID
* Operator password:	OperatorPassword

Pair hardware station:

Re-authenticate:

Name - Name of Virtual User,

Enable System authentication - this has to be checked and it is checked automatically. Do not change it.

Manual Mode - this option also has to be checked if we create Virtual User for CPOS.

Remember sensitive data - this option can be marked if you want to save the password for that Virtual User. If this option is not checked, every time when you will try to authenticate this Virtual User will be asked to provide password manually.

Platform - select record Cloud POS. This will cause for more fields to appear which we need to fill out.

Cloud POS URL - paste the same link to CPOS that you used in System Definition.

NOTE: Link has to end with “/” or it can cause problems when trying to access the environment by player/recorder.

Device ID - provide Device ID that you want to access (it is required to active CPOS)

Register number - provide Register Number that you want to access (it is required to activate CPOS)

Operator ID - ID (login) of Operator that can access the store,

Operator password - Password of Operator that can access the store,

After filling those details hit create.

The virtual User is now created but is not yet authenticated. Select it from list of all Virtual User and click Authenticate.

You will notice that the process of authentication for CPOS Virtual User is much longer than usual Virtual User used for example to access DFO environments. This is because not only we have to log in the AAD account but additionally we also have to activate the register.

The final screen that should be seen and the last two steps should happen is screen where we should provide Operator ID and Password to access the store.

After that you should be able to start recording new scripts on CPOS environment.

Quick Guide - How to start with EA

This document will showcase how to start working with Executive Automats.

We will include in this document aspects as:

- [Configuration and settings](#)
- [Creation of first project](#)
- [Creation of first script](#)
- [Playing the first script](#)

Configuration and settings

Before we can start recording any script we have to configuration needed data in settings.

Go to settings:

The screenshot shows the 'Settings' tab selected in the top navigation bar. Under 'Settings > Sequences', there is a table with columns for Name, Format, and Next Value. A red box highlights the 'Sequences' tab in the left sidebar. The table shows one entry: 'No Data' with a small folder icon.

Here we need to take care of two things: System Definition and Virtual User.

Lets start with System Definition.

System Definition is record that contains details about the system/web based app that we want to test. To create such click Add.

The screenshot shows the 'Settings' tab selected in the top navigation bar. Under 'Settings > System Definitions', there is a table with columns for Name and Address. A red box highlights the '+ Add' button in the top toolbar and the 'System definition' tab in the left sidebar. The table shows one entry: 'No Data' with a small folder icon.

We need to fill below information:

- Name - Name of the record,

- Environment address - URL of the system that we would like to test.
Note: We highly recommend to paste here URL of main page/home page of application we want to test. The reason is that from the position of this URL all scripts will be starting when played or when recording.
- Application Name - name of application we are creating record for.
- Environment Type - this is the way to tag your record as one of the following:
 - Test
 - User Acceptance Test
 - Production
 - Development
 - Other

NOTE: This is only information for other users what kind of environment it is and does not affect in any way your system. Even if you tag your environment as Development one even though it is a test environment it will not turn it into development environment.

- System platform - this is a list of all platforms/application/system that we tested with our tool with variety of clients. Here you should find your application that you want to test.
NOTE: If you cannot find correct record here or there is no record that you can select then select “Default Platform”. It will cause our application to run standard set of rules to handle working on application you want to test.

Create system definition

* Environment:	Enter environment
* Environment address:	Enter environment address
* Application name:	Enter application name
* Environment type:	Select environment type
Description:	Enter description
Multi-window support: <input type="checkbox"/>	
Disable resource sharing: <input type="checkbox"/>	
* System platform:	Default platform

There are more fields or options that can be selected. But those noted are enough to create record.

If you want more information about them please refer to our Manual Guide [Chapter 2.1.5 System Definition](#).

NOTE: Those records are shared between users so each of user do not have to create same records.

Second thing that we have to take care of is Virtual User.

Virtual User is a way our application will access the applications that we want to test. Virtual user contains credentials to account (mostly AAD account) that can access environments that we created records for in System Definition tab.

To create one click Add.

The screenshot shows the 'Virtual Users' page under the 'Settings' tab. At the top, there's a navigation bar with links for Application, Projects, Scripts, Scheduler, Documentation, History, Reports, and Settings. Below the navigation bar is a toolbar with buttons for '+ Add', 'Remove', 'Refresh', 'Edit', and 'Show errors/warnings'. On the left, a sidebar lists various settings categories: Global settings, Global actions, Sequences, Global variables, System definition, Storages, and Virtual user. The 'Virtual user' category is highlighted with a red box. The main content area has columns for Name, System login, Server login, and Status. A message 'No Data' is displayed below the table.

Fill below information:

- Name - name of the record,
- Platform - here you select platform which will be used to log in this user. Most of times we choose Microsoft Login here.
- Login - type in email address to account that we want to use to access applications with,

Additionally you can select option Remember Sensitive Data so Password field will appear. If this option is not selected user will have to provide password every time whenever he will try to authenticate Virtual Users.

- Password - password to account that we want to use to access applications with. Password is always encrypted and hidden behind dot icons.

Create virtual user

* Name:

Description:

System authentication

Enable system authentication:

Manual mode:

Remember sensitive data:

Custom script:

* Platform:

* Login:

* Password:

Re-authenticate:

Additional authentication

* Additional authentication: None Basic NTLM

There are more options and field that we can fill but those are enough to create new Virtual User. About other options you can read more in our Manual Guide [Chapter 2.1.7 Virtual User](#).

When Virtual User is created it will be noted as *NOT_AUTHENTICATED*. It means that record has been created but the configuration is not yet over.



Select the record that you just created and click Authenticate button.

Process of authentication of selected Virtual User will begin. It causes to run special authentication script to run in background that will authenticate selected Virtual User. During that process our tool takes credentials that were provided for record and tries to login that account to site office.com. If everything goes without any issues status will change to *AUTHENTICATED* and token will be generated and saved in database of our tool. This token is responsible for **AUTHOMATICALLY** logging this Virtual User to the system, so you will not have to do it manually every time when you want to record/play the script.



NOTE: If your Virtual User uses account that has MFA enable then during the creation of such Virtual User you have to select option *Manual Mode*. Thanks to this when we run authentication process, we are able to see it on the screen. Whenever you are asked to

perform MFA related action, pause the player, do the action related to MFA (do not click anything on authentication screen) and after that hit play again so process can continue.

NOTE2: Those records are shared between users so each of user do not have to create same records.

After that we are done with configuration settings of Executive Automats. Now we have to jump to Projects tab.

Creation of first project

Projects are way to organize work in Executive Automats. Treat them like folders on your computer.

Projects contains scripts inside of them. Projects can have subprojects, and those subprojects can have more subprojects etc.

To create new project click Create and provide name of the project:

The screenshot shows the 'Details' tab of a project named 'Test Automation Project'. The 'Create' button is highlighted with a red box. The 'Edit' button is also highlighted with a red box. The 'Execution status' section shows 'No data'. The 'Last 5 results' section indicates 'Not enough data. At least 2 executions should be performed'.

NOTE: If you want to create subproject to the project then you at first have to select project and then click Create. Doing so it will create subproject for selected project.

To freshly created project we need to determinate which System Definitions and which Virtual Users we want to use in that project.

Go to systems tab in selected project and hit EDIT:

The screenshot shows the 'Systems' tab of the 'Test Automation Project'. The 'Edit' button is highlighted with a red box. The 'Systems' tab is selected and highlighted with a red box.

Select System Definition that you want to add and hit Arrow pointed to right to add System Definition to project and click Save. It can be later edited to add more projects or remove the ones that are no longer needed for this project.

The screenshot shows the 'Systems' tab of a project. At the top, there are tabs for Details, Scripts, Test Results, Documented scripts, History, User groups, Virtual users, Systems (which is highlighted with a red box), and Issue tracking. Below the tabs, there's a search bar and a 'Select category' dropdown. The main area is titled 'All systems' and lists a single item: 'Dynamics 365'. This entry includes columns for Name, Address, Env. Type, and App name. To the right of the list is a 'Selected systems' section with a table header: Name, Address, Env. Type, and App name. Below the table, it says 'No Data'. A red arrow points from the 'Save' button at the top to the 'Selected systems' table.

The same has to be done for Virtual User. Go to Virtual User tab in selected project and hit EDIT:

The screenshot shows the 'Virtual users' tab of a project. At the top, there are tabs for Details, Scripts, Test Results, Documented scripts, History, User groups, Virtual users (which is highlighted with a red box), Systems, and Issue tracking. Below the tabs, there's a search bar and a 'Select category' dropdown. The main area is titled 'All virtual users' and lists a single item: 'VU test'. This entry includes columns for Name, System login, Server login, Status, Type, and Description. To the right of the list is a 'Selected virtual users' section with a table header: Name, System login, Server login, Status, Type, and Description. Below the table, it says 'No Data'. A red arrow points from the 'Save' button at the top to the 'Selected virtual users' table.

Select Virtual User that you want to add and hit Arrow pointed to right to add Virtual User to project and click Save. It can be later edited to add more projects or remove the ones that are no longer needed for this project.

After that we are able to create first script.

Creation of first script

Go to scripts tab in selected project (you can also select Scripts module from top ribbon and create script from that position) and hit add.

The screenshot shows the 'Scripts' tab of a project. At the top, there are tabs for Details, Scripts (which is highlighted with a red box), Test Results, Documented scripts, History, User groups, Virtual users, Systems, and Issue tracking. Below the tabs, there's a search bar and a 'Select category' dropdown. The main area has a 'Create' button and an 'Add' button (highlighted with a red box). Below these are buttons for Close, Clone, Import, Refresh, Show closed, Show scripts with work items, Export to file, Export to csv, Open, and Start. The table below lists scripts with columns for Id, Name, Environment, Env. Type, Status, Created At, and Created By. A red arrow points from the 'Add' button at the top to the 'Create' button.

Fill out below information to create a script:

- Project - Select project in which Script will be stored (automatically selects project in which you are right now),
- Name - name of the script
- Environment - option to use System Definition and Virtual User or Manual URL to access environment that we want to test.
 - Manual URL - paste direct URL of application you want to access
 - Environment - choose System Definition and Virtual User that should be used (recommended)

NOTE: One of System Definition and Virtual User is automatically selected as Default, if you want to change and use different records, un-toggle buttons, open menu and select correct records.

Create new script

* Project: Test Automation Project
* Name: Name of the script
Environment: Default Dynamics 365
Use virtual user:
Virtual user: Default
Description: Enter description
Execution timeout: seconds
Take screenshots:
Show overlay hint:
Hide comments:
Click on disabled buttons:
Auto-refresh datasource:
Custom resolvers mode:
Documentation settings

This is enough to create first script so click Create. If you want to know more about other options please refer to our Manual Guide Chapter [6. How to record a script](#)

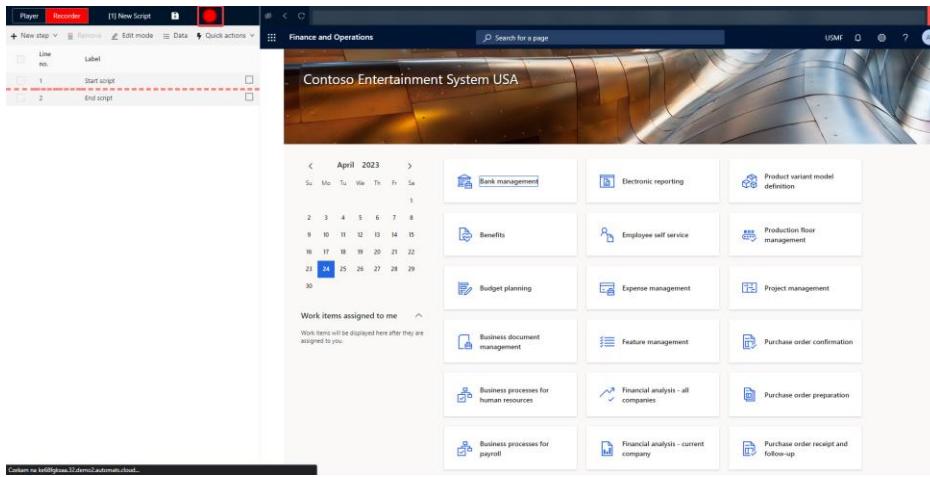
After that you will be directed to Steps page of Script. To start recording Hit Start → Record.

Scripts > 1 (New Script) > Steps [SAFAT](#)
General Variables Steps [Link in](#) Advanced conditionals History Work items Changelog Documentation
+ New step Refresh Copy Paste Record Quick actions
No step selected

After that you will be redirected to new page with Player/Recorder on the left and on the right side you should see the environment that you selected.

Note: If you correctly authenticated Virtual User before starting script you should automatically be logged into system. If you would be asked about credentials still, it means that either authentication process has not performed correctly or token has died and process has to be repeated.

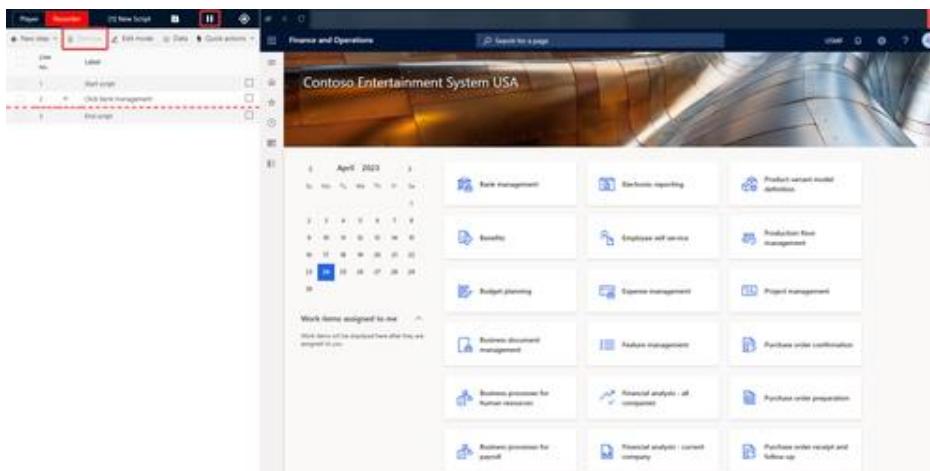
To Start recording click Big Red Circle button. Now every action, every click, every change will be recorded as a step.



Steps are saved automatically after couple of seconds.

If you want to Pause the recording you can do so by clicking Pause button. After that steps will not be created when performing actions.

If you record some steps and you want to remove them, you have to select them and click Remove.

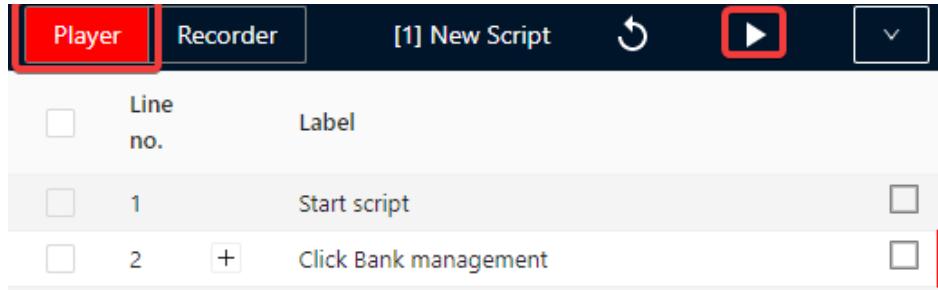


If you want more information about recording steps, modifying them etc. please refer to Manual Guide Chapter [6. How to record a script](#)

Playing the first script

After you have recorded steps for your first script you can play created script to do so click Player button on recorder. This will cause to turn recorder into player which allows to play your created script.

To start the player, click Play button.



Player will always reload environment and start from the beginning (not from where you left recorder).

Player can be also paused by pause button to stop script.

When player executes the script it marks executed steps as either:

- Success - marked with green marks
- Warning - marked with yellow triangle with ! in it
- Error - marked with red square with X inside.

Errors are always causing player to stop from executing script further. Warnings does not causes player to stop.

When executing the script our player tries to reproduce action of script as fast as it can only depending on environment performance. Means that if environment is overloaded and works slower, player will also execute actions in slower manner.

Quick Guide - How to start with JIRA

This document will showcase how to gain access to Jira portal and process of creating a request and most important information about raising a ticket via Jira.

Link to Jira - <https://xplus-product-support.atlassian.net/servicedesk/customer/portals>

Logging into Jira for the first time

You will receive a notification to the email address provided in the application for access to the portal.

The welcome e-mail contains information about the need to complete the process of adding the account, just click on the "sign-up link" and complete the process by selecting the account name to be displayed (usually name and surname) and setting a password for logging in.

From: GN <jira@xplus-product-support.atlassian.net>

Sent: Sunday, November 6, 2022 11:05:28 AM

To: pawel.baranowski@xplusglobal.com

Subject: Welcome to Client A

Welcome!

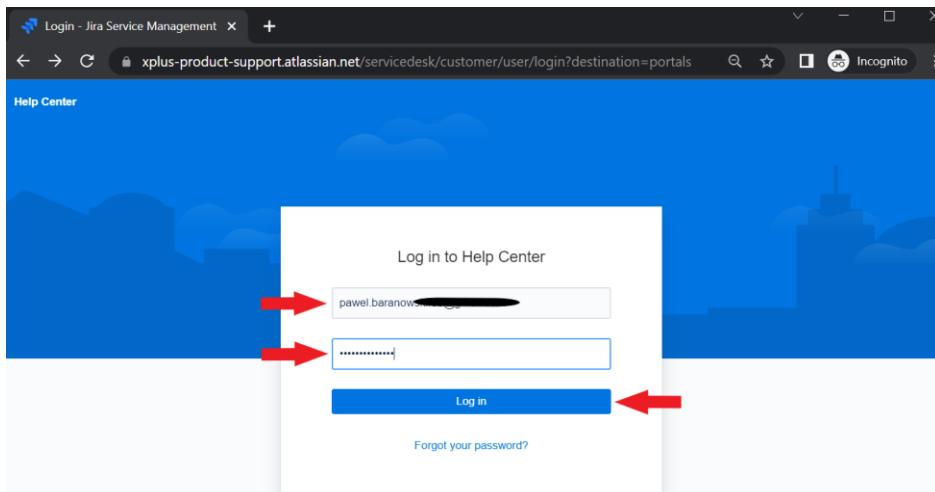
You've been invited to Client A portal. You can use this to raise requests and get help.

To finish setting up your account, simply click on your [sign-up link](#).



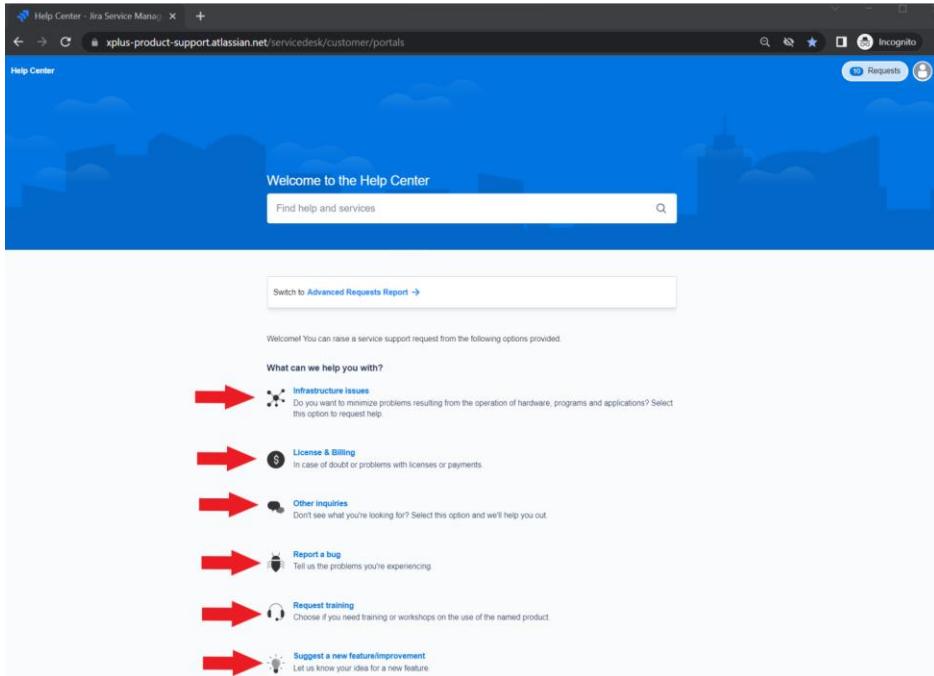
Powered by Jira Service Management

Login using your credentials:

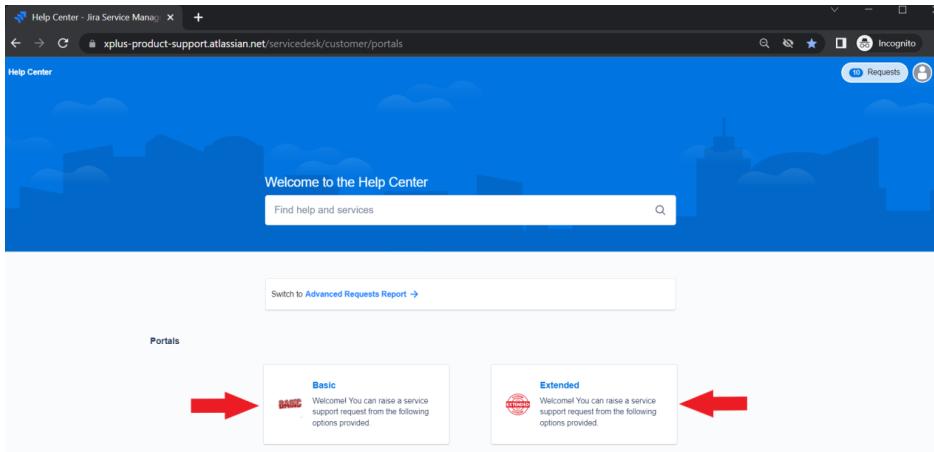


Main Page

If you have an access to one Company support profile:



If you have an access to more than one Company support profile, first step will be choose a Project, on behalf of whom the support request will be submitted:



Create a request

Depending on the category of your request, you can choose from the following 6 options:

- Infrastructure issues - issue/request related to infrastructure of EA e.g. schedule update, Foreground/Background mode does not work, problems with accessing environments from EA etc.
- License & Billing - issue/request related to License or billing e.g. change license from Basic to Extended for EA, questions about billing or invoices etc.

- Other inquiries - select this one if issue/request does not fit in any noted section,
- Report a bug - issue/request about error/potential bug that you encountered when working with EA e.g. fields are not recognize by player, cannot record step, not expected outcome etc.
- Request training - requests about organizing meeting, training session about EA or specific part of EA
- Suggest a new feature/improvement - request if you have an idea to improve our application or change something in it. We gladly hear any idea our users have;

Customer with Regular license, can create a ticket only with default priority (Medium).

Help Center / Basic

BASIC Basic

Welcome! You can raise a service support request from the following options provided.

What can we help you with?

Other inquiries
Don't see what you're looking for? Select this option and we'll help you out.

Summary *

Description *

Normal text

Attachment

Drag and drop files, paste screenshots, or browse

Share with *

Share with Team A

Send

Customer with Enterprise license, can create a ticket with a priority other than the default (Medium). Other priorities are Low, High and Critical.



Extended

Welcome! You can raise a service support request from the following options provided.

What can we help you with?



Report a bug

Tell us the problems you're experiencing.

Product*



D365 platform version*



Priority*



Summary*



Description*



Normal text ...

Attachment

Drag and drop files, paste screenshots, or browse



Share with*



If the customer has more than one product, he can choose it from the list, if he has only one, it is set as the default and the checkbox disappears automatically.

Customers with D365 environment reporting bugs or infrastructure issues will be additionally asked for a Platform version (i.e. Update50 (7.0.6354.90)) which will allow us diagnose issue faster.

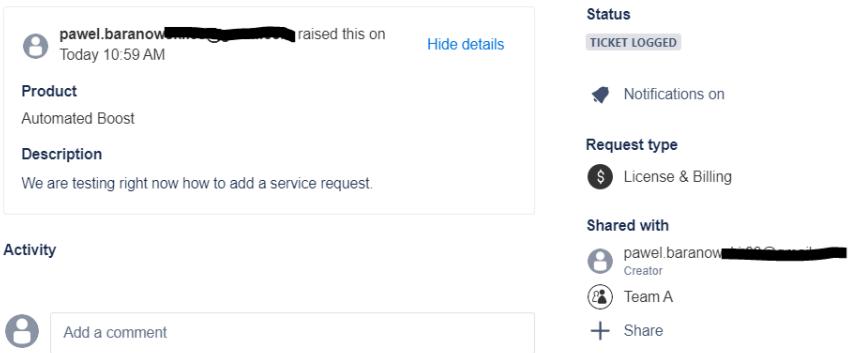
The Summary (subject of the request) and Description (broad description of the reported incident) fields must be completed. An attachment can be added, it will help us diagnose the reported problem faster, but it is not required.

If there are several departments within the client, then when adding, you can choose which department is reporting (Share with), if there is only one, it is the default. Send button create a ticket.

Newly created ticket look like bellow:

Help Center / Basic / BSC-13

Test 112



Status
TICKET LOGGED

Product
Automated Boost

Description
We are testing right now how to add a service request.

Activity

Add a comment

Request type
\$ License & Billing

Shared with

- pawel.baranowski Creator
- Team A

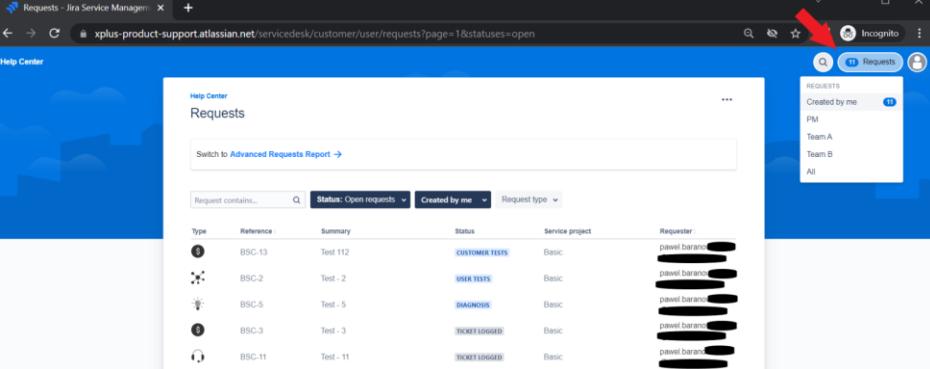
+

Share

When handling the request, we will change the statuses accordingly, so as to transparently show at what stage the request is currently at.

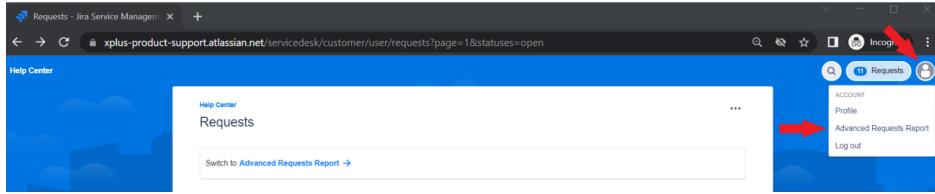
Reporting

After clicking the Requests button, you can see the requests to which you have access in a simple form.



Type	Reference	Summary	Status	Service project	Requester
BUG	BSC-13	Test 112	CUSTOMER TESTS	Basic	pawel.baranowski
BUG	BSC-2	Test - 2	USER TESTS	Basic	pawel.baranowski
BUG	BSC-5	Test - 5	DIAGNOSIS	Basic	pawel.baranowski
BUG	BSC-3	Test - 3	TICKET LOGGED	Basic	pawel.baranowski
BUG	BSC-11	Test - 11	TICKET LOGGED	Basic	pawel.baranowski

By clicking on the avatar on the right, you can choose a more advanced view and go to Advanced Requests Report



Now you see the advanced report:

Reference	Summary	Assignee	Created	Creator	Description	Issue Type	Organizations	Support Plan	Priority	Product	Request Type	Response Time	Solution	Support expiry date	Support Cost
BSG-12	Test 112	Fixed Baranowski	06/12/2023	Baranowski	We are testing right now how to add...	LICENSE & BILLING	Team A	Basic	1 Medium	Automated Boost	Report a bug	20m 30s ✅ SLA Goal 24h	On-Prem	11/11/2023	72
BSG-12	Test - 12	Fixed Baranowski	06/12/2023	Baranowski	Test baranowski	R&D	Team A	Basic	1 Low	Automated Boost	Report a bug	20m 30s ✅ SLA Goal 24h	On-Prem	11/11/2023	72
BSG-11	Test - 11	Fixed Baranowski	06/12/2023	Baranowski	Description 11	TRAINING	Team A	Basic	1 Medium	Data Protection	Request training	20m 30s ✅ SLA Goal 24h	On-Prem	11/11/2023	72
BSG-10	Test - 10	Fixed Baranowski	06/12/2023	Baranowski	Description 10	LICENSE & BILLING	Team A	Basic	1 Medium	Executive Automats	Report a bug	20m 30s ✅ SLA Goal 24h	On-Prem	11/11/2023	72
BSG-9	Test - 9	Fixed Baranowski	06/12/2023	Baranowski	Description 9	OTHER	Team A	Basic	1 Medium	Executive Automats	Report a bug	20m 30s ✅ SLA Goal 24h	On-Prem	11/11/2023	72
BSG-8	Test - 8	Fixed Baranowski	06/12/2023	Baranowski	Description 8	NOW FEATURE	Team A	Basic	1 Low	Data Protection Suite	Other inquiries	20m 30s ✅ SLA Goal 24h	On-Prem	11/11/2023	72
BSG-7	Test - 7	Fixed Baranowski	06/12/2023	Baranowski	Description 7	NOW FEATURE	Team A	Basic	1 Medium	PEP	Suggest a new feature/Improvement	20m 30s ✅ SLA Goal 24h	On-Prem	11/11/2023	72
BSG-6	Test - 6	Fixed Baranowski	06/12/2023	Baranowski	Description 6	NOW FEATURE	Team A	Basic	1 Low	Executive Automats	Suggest a new feature/Improvement	20m 30s ✅ SLA Goal 24h	On-Prem	11/11/2023	72
BSG-5	Test - 5	Fixed Baranowski	06/12/2023	Baranowski	Description 5	NOW FEATURE	Team A	Basic	1 Medium	Executive Automats	Suggest a new feature/Improvement	20m 30s ✅ SLA Goal 24h	On-Prem	11/11/2023	72
BSG-4	Test - 4	Fixed Baranowski	06/12/2023	Baranowski	Description 4	LICENSE & BILLING	Team A	Basic	1 Medium	Split Payment	Report a bug	20m 30s ✅ SLA Goal 24h	On-Prem	11/11/2023	72
BSG-3	Test - 3	Fixed Baranowski	06/12/2023	Baranowski	Description 3	LICENSE & BILLING	Team A	Basic	1 Low	Data Protection Suite	Report a bug	20m 30s ✅ SLA Goal 24h	On-Prem	11/11/2023	72
BSG-2	Test - 2	Fixed Baranowski	06/12/2023	Baranowski	Description 2	DATA ISSUES	Team A	Basic	1 Medium	Executive Automats	Infrastructure issues	20m 30s ✅ SLA Goal 24h	On-Prem	11/11/2023	72

In this view, you can sort the reports according to your needs, by department, status, persons submitting them, types of reports or by searching for any text.

You can manage the column view, add and delete it, organize it, then just save the view and it will be available next time exactly as you set it.

You can also export the list of applications to excel for your own needs.

EXECUTIVE AUTOMATS D365 USER GUIDE

EXECUTIVE AUTOMATS D365 USER GUIDE

Document Title	EXECUTIVE AUTOMATS D365 USER GUIDE
Author	XPLUS
Date	2023-03-13
EA Version	4.3

Executive Automats®® is a proprietary tool of XPLUS, which as a Microsoft Partner has been implementing Dynamics AX and Dynamics 365 For Finance and Operations solutions in Europe over the last 20 years. Testing business processes implemented in the AX system is an inseparable and very costly part of the implementation. To increase the profitability of projects and speed up the acceptance of individual modules, the Executive Automats tool was created by XPLUS. For the first few years XPLUS used test automation as a competitive advantage on the market, and from 2014 it started selling this solution to partners and end-users. In 2018 a version supporting Dynamics 365 for Finance and Operations was released, which also allows cross-platform tests, namely process tests that include integration with other external systems (interned browser based).

How can You benefit from implementation of the Executive Automats?

- Ø Intuitive and easy to use automated tests based on recorded Scripts
- Ø Functionalities allowing re-use Scripts and combine them with other Scripts. Minimizing the work required to maintain automated tests that are compatible with business processes.
- Ø Manage automated tests of all environments (UAT, Release, DEV) in one place.
- Ø Planning and autonomous running of test suites.
- Ø Ability to integrate with VSTS and support the entire manufacturing process directly from VSTS.
- Ø Cross-platform tests allowing testing of processes starting in D365FO passing through Cloud POS and CRM or Outlook.
- Ø Migrate complex data between applications without having to develop integrations.

- Ø Automate production processes (production environments, LIVE) and minimize the "mechanical" work of users.
- Ø Measuring the length of processes (time and step).
- Ø Verification of the efficiency of the environment as well as the business process itself.
- Ø Increased stability of the production environment.

Table of Contents

- [Glossary](#)
- [1. EA tool interface – description](#)
- [2. SETTINGS module](#)
 - [2.1 Application](#)
 - [2.1.1 Global Settings](#)
 - [2.1.2 Global Actions](#)
 - [2.1.3 Sequences](#)
 - [2.1.4 Global Variables](#)
 - [2.1.5 System Definition](#)
 - [2.1.6 Storages](#)
 - [2.1.7 Virtual User](#)
 - [2.1.8 API Key](#)
 - [2.1.9 User groups](#)
 - [2.1.10 Project Categories](#)
 - [2.1.11 Users](#)
 - [2.1.12 Localizations](#)
 - [2.1.13 Report templates](#)
 - [2.1.14 License](#)
 - [2.1.15. KPI counter](#)
 - [2.1.16 Code](#)
 - [2.1.16.1 Code Template Group](#)
 - [2.1.16.2 Code Templates](#)
 - [2.2 Profile](#)
 - [2.2.1 Basic Settings](#)
 - [2.2.2 Execution Settings](#)
 - [2.3 Azure DevOps Integration](#)
 - [2.3.1 Basic Parametrization](#)
 - [2.3.2 Manual creating work items](#)
 - [2.3.3 Creating rules for automatic work item creation](#)
 - [2.3.4 Executing Scripts/Schedulers/Projects using DevOps build Pipelines.](#)
 - [2.3.5 Test Plans Integration](#)
 - [2.3.6 Executing Test Plans using DevOps build Pipeline](#)
 - [2.3.7 Executing Test Plans and Test Cases from EA with results in DevOps](#)
 - [2.3.8 Opening test case and test result using History tab](#)
 - [2.3.9 Closing Test Plans and Test Cases](#)
 - [2.3.10 Configurations](#)

- [3. History module](#)
 - [3.1 Execution/play states](#)
 - [3.2 Performance counters statistics](#)
- [4. PROJECTS module](#)
 - [4.1 Creating project](#)
 - [4.2 Projects interface](#)
 - [4.2.1. Details tab](#)
 - [4.2.2. Scripts tab](#)
 - [4.2.3. Test results tab](#)
 - [4.2.4. Documented Scripts tab](#)
 - [4.2.5 History Tab](#)
 - [4.2.6. User groups tab](#)
 - [4.2.7. Virtual Users tab](#)
 - [4.2.8. Systems tab](#)
 - [4.2.9. Issue tracking tab](#)
 - [4.3 Scheduled execution of the project](#)
 - [4.3.1 Scheduled execution of the project by Script status](#)
- [5. SCRIPTS module](#)
 - [5.1 Function interface scripts](#)
 - [5.2 GENERAL tab](#)
 - [5.2.1. General Information](#)
 - [5.2.2. Data Source](#)
 - [5.3 Variables tab](#)
 - [5.3.1 Defining local variables](#)
 - [5.3.2 Defining global variables](#)
 - [5.3.3 How to use a variable on a Step](#)
 - [5.4 Steps tab](#)
 - [5.5 Linked In tab](#)
 - [5.6 Advanced Conditional tab](#)
 - [5.6.1 Defining Advanced Conditionals](#)
 - [5.7 History tab](#)
 - [5.8 Work Items tab](#)
 - [5.9 Changelog tab](#)
 - [5.10 Documentation tab](#)
 - [5.10.1 Skip documentation for linked scripts](#)
- [6. How to record a script](#)
 - [6.1 Scripts form: step by step](#)
 - [6.2 How to work with a recorded Script](#)
 - [6.2.1 Modifying the Script](#)
 - [6.2.2. Adding steps to a Script](#)
 - [6.2.3. Editing and making changes to the recorded steps](#)
 - [6.2.3.1 Change the order of steps](#)
 - [6.2.3.2 Copy/paste and delete steps.](#)
 - [6.2.3.3 Quick Actions](#)
 - [6.2.3.4 Assigning sequence numbers to steps](#)
 - [6.2.3.5 Work with an Excel file](#)
 - [6.2.3.6 Expressions and Advanced Code](#)
 - [6.2.3.7 How to use Expression](#)

- [6.2.3.8 How to use Advanced Code](#)
- [6.2.3.9 How to use Code Template](#)
- [6.2.3.10 New Step](#)
 - [6.2.3.10.1. Assign value to variable](#)
 - [6.2.3.10.2 Conditional step as a data validation](#)
 - [6.2.3.10.3 Conditional step as a tool for reading Excel data](#)
 - [6.2.3.10.4 Login step](#)
 - [6.2.3.10.5 Shortcut step](#)
 - [6.2.3.10.6 Script step, linking scripts](#)
 - [6.2.3.10.7 Auth step, Virtual user Custom authentication script](#)
 - [6.2.3.10.8 Code step](#)
- [6.2.3.11 Changing the Status of the script](#)
- [6.2.3.12. Cloning the Script](#)
- [6.2.3.13 Changing Path to the element](#)
- [6.3 How to work with scripts using Advanced Recorder](#)
 - [6.3.1. Recording a Script using Advanced Recorder](#)
 - [6.3.2. Working with a recorded script using Advanced Recorder](#)
 - [6.3.2.1 Add a new step](#)
 - [6.3.2.2 Remove selected steps](#)
 - [6.3.2.3 Edit mode](#)
 - [6.3.2.4 Data tab](#)
 - [6.3.2.5 Quick actions](#)
 - [6.3.2.6 Step details](#)
- [6.4 Playback of a recorded Script](#)
 - [6.4.1 Standard playback of the Script using basic player](#)
 - [6.4.2 Standard playback of the Script using Advanced Player](#)
 - [6.4.3 Non-standard playback of the Script](#)
- [6.5 Import/Export Scripts](#)
- [6.6 Capturing files](#)
- [7. SCEDULER module](#)
 - [7.1 How to create a Scheduler](#)
 - [7.2 How to create Gantt scheduler](#)
- [8. Documentation module](#)
- [9. Reports module](#)

Glossary

Definitions listed below will help users to understand terms and abbreviations used in Executive Automats tool and this manual.

Concept	Description
EA	Executive Automats
Environment	Environment used by the user to record business processes and saves as scripts

Projects	Projects are used to group and manage security of the scripts
Script	A sequence of actions (steps), representing execution of a business process recorded by the user
Step	Individual action taken by the user when recording a script
Recording	A functionality that allows to record all actions taken by the user in the system
Picker	A functionality that provides the user, in an easy and intuitive graphical way, all usage options concerning external data input for editable field
Variable	A functionality that allows to transfer data with a script (continuous process) or multiply merged Scripts (between business processes)
Expression	An advanced formula where the user can create logical expressions with the help of hints
Advanced Code	Advanced formulas, based on the combination of several logical values
Logical expression	A formula that, based on fixed or variable functions and dependencies between them, returns one logical value: True or False
Constant	A fixed value of a given field indicated during recording a script step (assigned to a step of a type Select row or Type)
Data source	Data source for the script. The most common: Excel file containing data columns
Logs	A launch log of the selected EA element (script, script, Scheduler)
Sequences	The sequence of numbers defined in the settings, which can then be used in selected script steps

1. EA tool interface – description

Executive Automats interface has been designed so that the user can easily find the necessary functionalities and be fully productive in the first moments of using the application.

The following figure shows the main screen of the Executive Automats application.

The top menu consists of 7 items that facilitate navigating the EA area without filtering.



- Projects** – Projects module shows all projects created in EA and allows users with assigned **Admin** role to create new projects and manage projects security.
- Scripts** – shows all Scripts that were recorded using EA
- Scheduler** – task planner that can be run by EA independently depending on the conditions that were previously defined for a specific Script. The scheduled execution can be one-time or cyclical. These tasks are done in the background, we do not need any interaction with the system. More in [Chapter 7. Scheduler Mode](#)
- Documentation** – quick access to generated documentation. More in [chapter 8. DOCUMENTATION module](#)
- History** – shows all runs of Scripts. Each Script run is registered in history, so you can check the correctness of the function in each play. If the function does not work correctly, the tool logs the fault in the form of a warning or error along with the reason for its occurrence
- Reports** – shows report created from level of Power BI. More in [chapter 9. Reports](#).
- Settings** – a tab for EA parameterization, creation of sequences and preview of basic settings

Before we start work, it is necessary to check whether all parameters necessary for the proper use of the EA tool have been parameterized. Let's start then with the **Settings** area.

2. SETTINGS module

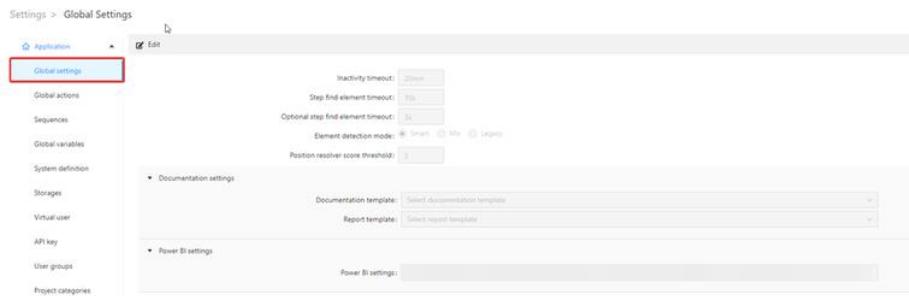
The **Settings** module contains basic areas that are subject to parameterization and ensure the proper functioning of the EA tool and work with it.

To have full access to the Settings module, the user must have an Admin role assigned to him.

2.1 Application

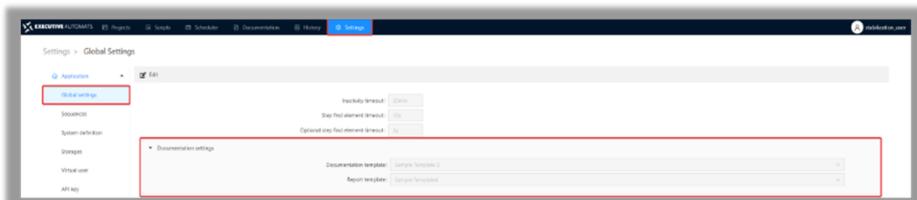
2.1.1 Global Settings

In **Global setting** area users can modify Player timeouts, such as: **Inactivity timeout**, **Step find element timeout** and **Optional step find element timeout**. NOTE: we recommend to not change those settings without consulting with our support group.



Click **Edit** to make options editable. When everything is ready. Click **Save** to save changes.

In this area in **Documentation settings** section users can select which templates should be used for generated Reports and created Documentation.



Click **Edit** to make options editable, select previously created templates (how to create a template in [2.1.13 Report templates](#)) and click **Save**.



The section contains two fields:

- **Documentation template:** the template will be used for reports automatically created in **Documentation mode** (more about documentation in chapter [5.10. DOCUMENTATION tab](#))
- **Report template:** the template will be used for report created after clicking **Generate report** in **History tab**

Note: user can select different templates in **Documentation template** and **Report template** fields.

Note 2: User can determinates different templates for certain scripts, projects.

In area Power BI settings user can leave URL to Power BI application which can be later accessed from Report tab and can generate reports depending on it. More about that in separate document: Power BI integration with EA.

2.1.2 Global Actions

Global actions is a place to delete logs screenshots, once or every two 24 hours older than the time period that user sets. Screenshots will be deleted from disk and therefore it will more space available but you will be no longer able to check screenshot from executions older that period of days determinate during the removal.

2.1.3 Sequences

Sequences is an area that allows you to define sequences of numbers, which can then be used in the selected steps.

Sequences of identifiers are numbering groups of a specific format created in the system.

When creating documents in the system, subsequent numbers from an adequate group are automatically down-loaded and assigned to newly created documents.

There is no need to create a new sequence of identifiers if the numbering sequence for the given documents already exists in the system. Then it should be found in the table using the filter function and used for the needs of the script being recorded.

To add new numbering sequence click **Settings\Application\Sequences** and click **Add**.

The **Create a new sequence** form will be opened to create a new sequence.

The screenshot shows a 'Create sequence' dialog box. At the top, there are two dropdown menus: 'Name' containing 'User##' and 'Format' containing 'User##'. The 'Format' dropdown is highlighted with a red box. At the bottom of the dialog are two buttons: 'Create' (highlighted with a red box) and 'Close'.

Complete fields:

Ø **Name** – in the Name field, enter the full name for the sequence of identifiers to be created.

Ø **Format** – the field in which we set a constant value to be always assigned to the used sequence and the variable value defined as ### - that is, numeric characters that represent numbers, increasing each time a number from this sequence is used.

For example, the format User_### allows you to create sequences User_001, User_002 or format ####_2018 allows you to create the sequence 0001_2018, 0002_2018 etc.

To save changes, click **Create**.

The system will reload the table with all created sequences, and the created sequence will appear at the top of the table

The screenshot shows the EXECUTIVE AUTOMATS application interface. The top navigation bar includes links for Projects, Scripts, Scheduler, Documentation, History, Reports, and Settings. The current view is under the Settings tab, specifically in the Sequences section. On the left, there is a sidebar with categories: Global settings, Global actions, Sequences (which is selected and highlighted in blue), Global variables, and System definition. The main area displays a table with columns: Name, Format, Next Value, Created At, and Updated At. The table contains several entries, including 'testreire' (Format: Montag####, Next Value: 581, Created At: 21.02.2022, 11:18:35, Updated At: 29.06.2022, 13:22:21), 'nopath####' (Format: ####, Next Value: 7, Created At: 14.02.2022, 12:50:26, Updated At: 27.06.2022, 09:18:06), 'ss_#' (Format: ##, Next Value: 1, Created At: 14.02.2022, 10:26:43, Updated At: 14.02.2022, 08:26:43), 'MT' (Format: ##, Next Value: 2, Created At: 11.02.2022, 15:10:12, Updated At: 24.02.2022, 08:23:12), 'NewSeq' (Format: New####, Next Value: 115, Created At: 9.02.2022, 14:44:40, Updated At: 27.06.2022, 12:34:53), 'ef##' (Format: ##, Next Value: 1, Created At: 9.02.2022, 14:42:58, Updated At: 9.02.2022, 14:42:58), and 'TestSeq' (Format: Test##, Next Value: 2, Created At: 9.02.2022, 14:22:35, Updated At: 9.02.2022, 14:24:32).

2.1.4 Global Variables

In the **global variables tab** the user is provided with the list of all **global variables** in the EA tool. In this tab global variables can be maintained i.e. added, removed, and edited. Global variables can be of two types: constant or variable. Global variables can be used in all scripts in the EA tool.

Using global variables the user can pass the values between scripts (when using global variables of type variable) or can set a value that will be the same for all scripts (when using global variables of type constant). There is no need to create a global variable if it will be used only in one script.

To create a new **global variable** click **Settings/Application/Global variables** and click **Add**.

The screenshot shows the EXECUTIVE AUTOMATS application interface. The top navigation bar includes links for Projects, Scripts, Scheduler, Documentation, History, Reports, and Settings. The current view is under the Settings tab, specifically in the Global variables section. On the left, there is a sidebar with categories: Global settings, Sequences, and Global variables (which is selected and highlighted in blue). The main area displays a table with columns: Name, Value, Type, Value type, and Description. The table contains three entries: 'myVarGlobal' (Value: test, Type: Variable, Value type: Text), 'myConstGlobal' (Value: test, Type: Constant, Value type: Text), and 'global3' (Value: test, Type: Constant, Value type: Text).

The **Create global variable** form will be opened to create a new global variable.

Create global variable

* Variable type:	Select type
* Name:	Enter name
Value type:	Text
Value:	Enter value
Description:	Enter description

Create **Cancel**

Complete fields:

Ø Variable type – this is a required field. Type constant should be chosen if the user wants this global variable to be fixed and unchangeable, or type variable should be chosen if the user wants to modify the value of this global variables in scripts.

Ø Name – this is a required field. In the Name field, enter the full and unique name for the global variable to be created.

Ø Value type – the field for choosing an option In this field is not obligatory.

Ø Value – if a value type was selected a new field appears and the value can be set.

Ø Description – the field for notes. To save changes, click **Create**.

2.1.5 System Definition

System definition is a module in which the systems operated by the EA tool are parameterized.

The system definition helps in defining the name, address, application name and type of environment, which will be the place of work with EA.

To define a new work environment, go to the **Settings** area, the **System definitions** tab and click **Add**.



On the **Create system definition** form, fill in the data necessary to describe the new environment, where the user will be able to work with the EA tool – **Environment name**, **address**, **Application name**, **type** and **Platform**.

In **Platform** field you can determinate which type of environment (from possible ones to choose) you are using,
then our tool will use correct set of rules.

Note: If Platform is set up as Dynamics 365 for Finance and Operations then new Checkboxes will appear on form :

- “Expand All tabs automatically” – this will cause all tabs inside the environment to be expanded so there will be no need to record steps in which you were expanding specific tabs.
- “Force popups to open” – this option fixes issues with lookups not opening in FO correctly. Do not check this option if it is not an issue for you.
- “Force context menu to open” – this option fixes issues with context menu not working correctly in FO. Do not check this option if it is not an issue for you.

Note: For On-premise environments type / at the end of the link (e.g. <https://office.com/> instead of <https://office.com>)

For example:

A screenshot of the 'Create system definition' dialog box. It contains the following fields:

- * Environment: Xplus Demo Environment
- * Environment address: https://login.microsoftonline.com
- * Application name: Dynamics 365 for Finance and Operations
- * Environment type: Test
- Description: Enter description
- Multi-window support:
- Disable web security:
- * System platform: Default platform

To save the settings, click **Create**.

Environment types are described in the [chapter 5.1 Functional Interface Scripts](#). For some environments enabling additional options might be useful

- **Multi-window support** – this option helps resolve issues caused when after performing some actions an application opens new tab instead of re-directing in the current one.
- **Disable web security** – this option help solve issues caused by Iframes.

2.1.6 Storages

This tab is used to create and manage cloud storages which can be used to integrate EA tool with cloud environments. (Functionality in development).

2.1.7 Virtual User

The area where we can define new **Virtual users** and make changes to the data of users already created. To add a new virtual user, click **Add**.

The screenshot shows the EA tool's Settings interface with the 'Virtual Users' tab selected. A red box highlights the 'Virtual user' link in the left sidebar. Below it, a larger red box highlights the 'Create virtual user' dialog box.

Create virtual user

* Name: XPLUS Demo User
 Description: Enter description

System authentication

Enable system authentication:
 Manual mode:
 Remember sensitive data:
 Custom script:

* Platform: Microsoft Login
 * Login:
 * Password:
 Re-authenticate:
 * Repeats every: 24 hours

Additional authentication

* Additional authentication: None Basic NTLM

Buttons: Create, Cancel

The **Create virtual user** window will appear to enter the data.

Complete fields:

Ø **Name** – Enter user name

Ø **Description** – User can add additional description for the Virtual user (this field is no-obligatory)

Ø **Enable system authentication** – Selecting this checkbox enables system authentication. For enabled authentication user have to define Platform and credentials for Virtual User

Ø **Manual mode** – Selecting this checkbox allow user to display Authentication process and then allow user to make changes to this process. For example if user needs to type Text message during login

Ø **Remember sensitive data** – Selecting this checkbox allow system to store sensitive data (e.g. password)

Ø **Custom script** – Selecting this checkbox allow user to point at recorded script which will be used during authentication process (how to create and use custom script is explained in *chapter 6.2.3.10.7. Auth step, Virtual user Custom authentication script*)

Ø **Platform** – Select the Platform to which the given user will log in (Cloud POS , Microsoft Login or Okta)

Ø **Login** – enter the login assigned to the given user

Ø **Password** – Enter the password assigned to the given user

Ø **Re-authenticate** – Selecting this checkbox enables automatic Re-authentication

o **Repeats every** – Set the number of hours after which virtual user should be re-authenticated

Ø **Additional authentication:**

o **None** – Select this checkbox if the additional authentication is not necessary

o **Basic** – Select this checkbox if the standard authentication should be performed

o **NTLM** – Select this checkbox if the additional NTLM authenticate type is necessary

Click **Create** to save changes.

The created user is now **NOT_AUTHENTICATED**.

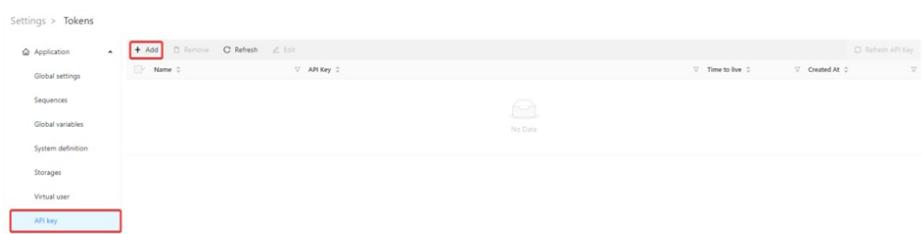


Click **Authenticate** to activate the user. Using this functionality causes the system to log on to the user in the system background.

2.1.8 API Key

This area is a repository for **API keys** generated by EA. Executive Automats can create API keys which are used to integrate EA with other instances or systems.

To create new API key click **Add** button.



On the next form fill **Name** for the API key and then select amount of time that this API key should be active.

The dialog box has the following fields:

- * Enter name: Enterprise portal
- * Type Time to Live: 90 Days

The dropdown menu for 'Time to Live' contains the following options:

- 90 Days
- 180 Days
- 1 Year

Next create save API key by using button **Create**.

The newly API key will be created. You can reactive API keys by using **Refresh API key** button.

<input type="checkbox"/>	Name	API Key	Time to live	Created At	<input type="button" value="Refresh API Key"/>
<input type="checkbox"/>	PW DA	1 Year	11/05/2020, 0:55:54 PM	
<input type="checkbox"/>	Power Automate	1 Year	11/12/2020, 0:22:52 PM	
<input checked="" type="checkbox"/>	Enterprise portal	30 Days	12/14/2020, 0:38:56 AM	
<input type="checkbox"/>	Phuz	1 Year	11/13/2020, 10:32:23 AM	

If it is necessary you can display **API key** with the **Eye** button.

To quick copy **API key** use **Copy** button.

2.1.9 User groups

This area is used to create and manage **User groups**. **User groups** are used to group users into teams that can be added to specific projects and thus gain the access to work within given project.

To create new **User group** use **Add** button:

On the next form enter name for **User group** and optionally add description.

Create user group

* Name: Enter name

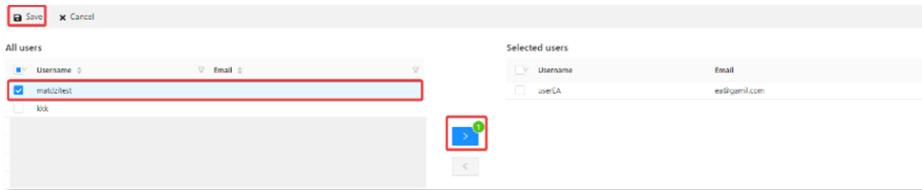
Description: Enter description

After a **User group** is created, you can assign users to the group.

Select user group and click **Open** button.



Next use **Edit** button and select specific users to this group.



After assigning users to group, use **Save** button.

2.1.10 Project Categories

Project categories tab is used to create and manage categories that are assigned to projects. Categories are used for additional project grouping and can also be used for filtering of projects.

To create new **Project category** use **Add** button.



On the next form enter **Name** of project category and optionally fill **Description** and then click **Create**.

Create project category

* Name:	Production
Description:	Enter description

2.1.11 Users

This tab is used for defining users of the EA tool. This is the place to set up access accounts. To add login details for a new user, click **Add**.

Settings > Users

The screenshot shows a list of users in a table format. The columns are: Username, Email, Group name, and Active. There are 18 users listed, each with a checkmark in the 'Active' column. The 'Group name' column shows various group names like 'no group', 'me', and 'test'. The 'Active' column shows all entries as checked.

Username	Email	Group name	Active
testuser2	testuser2@opiu.pl	no group	✓
pwuser@opiu.pl		no group	✓
allow		no group	✓
jl_user		no group	✓
pr_user		no group	✓
jl_user		no group	✓
test123		no group	✓
project		no group	✓
settings		no group	✓
meta		no group	✓
user123		me	✓
jl		no group	✓
user001		no group	✓
user002		no group	✓
Black_Mamba		test	✓
pr		no group	✓
area		no group	✓
projects_nile		test	✓

The form for creating a new user will open. Complete the necessary data:

Create user

Single sign-on:	Select single sign-on provider
* Username:	
* Email:	Enter email
* Password:	*
* Language:	Select language
Roles:	Assign roles
Issue tracking tool:	Select issue tracking tool
* Execution report extension:	Select execution report extension
Display only published documentation:	<input type="checkbox"/>
* Scripts and reports locale:	Select scripts and reports locale
Is active:	<input type="checkbox"/>

Ø **Username** – enter the username

Ø Email – enter the user's email address

Ø Password – enter password

Ø Language – Select language EN/PL

Ø Roles – Assign specific roles for newly created user:

- **Scheduler** – Allows user to access Scheduler module
- **Logs** – Allows user to access History and Scripts module
- **Projects** – Allows user to access Projects module
- **Settings** – Allows user to access Settings module
- **Admin** – Access to every module and additional advanced settings. Admins can also create new projects
- **Documentation_reader** – Allows user to read reports created during scripts background playback. Note: This role should not be assigned if user has any other roles assigned. (All other roles contains Documentation_reader role). This role is only to read documentation.
- **Documentation_creator** – Allows user to enable automatic reports creation during scripts background playback
- **Reports** – allows user access to reports module

Ø Issue tracking tool – Select Issue tracking tool if user should be allowed to create work items in Azure DevOps based on errors found in Scripts. Integration between EA and Azure DevOps is described in the chapter [2.3. Azure DevOps integration](#)

Ø Execution report extension - Select format (docx/pdf) of the report that user can generate for specific script execution

Ø Display only published documents – Select this checkbox if user with **documentation_reader** role should have access only to published reports

Ø Scripts and Reports locale – Select which translation should be used for Recorder/Player, script steps and generated documentation (more about translations in chapter [2.1.12. Localizations](#))

Ø Is active – Select this option so the user is active and have a possibility to log to the tool. If this checkbox is not selected during the creation of user, then it will be not possible to access tool with this user.

Ø **Single Sign-On** – If in this field Azure Active Directory is selected, then the user that will be created will use Azure Active Directory account to log in to Executive Automats. In username you have to type down email address associated with AAD account. During logging this user will have to use option Log in with Microsoft instead of standard procedure.

To save user, click **Create**.

2.1.12 Localizations

This tab is used to create translation for script executions reports generated, Recorder and Player elements and Script steps. To create new Localization click **Add**.

Name	Description	Created At	Updated At
en		24/05/2021, 13:55:18	03/07/2021, 11:01:36
de	German	14/02/2020, 11:04:07	03/07/2021, 11:01:36
es		14/02/2020, 10:49:01	03/07/2021, 11:01:36
fr		14/02/2020, 10:47:40	03/07/2021, 11:01:36
it		14/02/2020, 11:00:47	03/07/2021, 11:01:36

Edit documentation form will appear. Type **Name** for new localization and click **Save** at the bottom of the form.

Edit documentation

* Name: de

Description: German

New localization will be created with status **Needs translation**. Select it and click **Edit translations**.

Name	Description	Created At	Updated At
en		03/07/2021, 11:00:03 PM	12/14/2023, 0:51:18 AM
de	German	03/07/2021, 11:00:47 AM	12/14/2023, 0:51:18 AM

Translate form will appear. On the left side original fields names are displayed, on the right side type new name for each field.

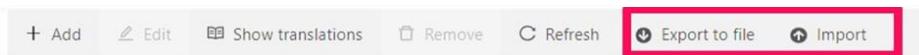
The screenshot shows a 'Translate' dialog box with a list of localization entries. Each entry consists of a left column with the original field name and a right column for translating it. The entries include:

- End script
- Link
- Start script
- Stop
- Assertion condition result is [result]
- Calculating assertion [leftConditionValue] [operator] [rightConditionValue]
- Cannot read checkbox/togglebox value. Element was clicked without checking value
- [LogicalOperator] Condition "[leftConditionHeader]" [leftCondition] > [leftConditionValue]
- Checking if datasource [value] has next row
- [leftConditionValue] [operator] [rightConditionValue] is [comparisonResult]
- Condition "[type]" value [value]
- No record found
- Number of executions: [counter] exceeded limit: [limit]. Execution will be terminated
- Number of executions: [counter] exceeded limit: [limit]. Step will be ignored
- Override status to [status]
- Script execution expired

At the bottom are 'Save' and 'Cancel' buttons.

When all fields will be translated click **Save**.

It is also possible to **Export/import** localizations.



TO export a translation, select one and click **Export to file**. New json file will be downloaded

Note: such json file can be edited (e.g. using notepad) to create translation instead of using Translate form within the tool.

To import a translation, click **Import**

The screenshot shows an 'Import localization file' dialog box. It contains a checkbox labeled 'Overwrite existing translations:' with a checked state. Below the checkbox is a file selection area showing 'de.json' and a delete icon.

Ø Import localization file form will appear. Drag & drop previously exported json file and click **Import** at the bottom of the form.

Ø Check **Override existing translations** checkbox if there is already created translation with such name and should be overridden

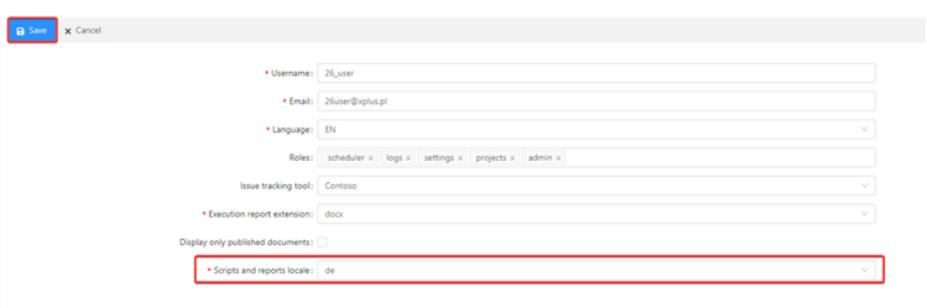
o If translation with such name exists and user imports it without **Override existing translations** checkbox, as a result sum of these two files will be created

§ In that case, before import, edit exported file (e.g. in notepad) and remove these fields which shouldn't be overridden (like e.g. all empty values) and keep only these that should be added

§ If some fields are translated in both files, translation will contain value from the imported file.

To use created translation go to **Settings -> Profile -> Basic Settings** and click **Edit**. In **Scripts and Reports locale**

field select translation which should be used. After that click **Save**.



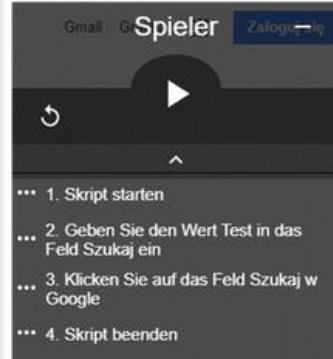
From now, all reports generated by this user will use attached translation, as well as Player, Recorder, script Steps and History tab.



Recorder — [Zaloguj się](#)

II ⏪ ⏴ ⏵ ⏹

1. Skript starten
2. Geben Sie den Wert Test in das Feld Szukaj ein
3. Klicken Sie auf das Feld Szukaj w Google
4. Skript beenden



[Gmail](#) [Spieler](#) [Zaloguj się](#)

⟳ ⏪ ⏴ ⏵ ⏹

- ... 1. Skript starten
- ... 2. Geben Sie den Wert Test in das Feld Szukaj ein
- ... 3. Klicken Sie auf das Feld Szukaj w Google
- ... 4. Skript beenden

Player	Recorder	[928] Free text invoice 3	⟳	▶	▼
<input type="checkbox"/>	Line no.	Label			
<input type="checkbox"/>	1	Skript starten	<input type="checkbox"/>		
<input type="checkbox"/>	2	+ Gehe zu Accounts receivable/invoices/All free text inv...	<input type="checkbox"/>		
<input type="checkbox"/>	3	+ Klicken Sie auf New	<input type="checkbox"/>		
<input type="checkbox"/>	4	+ Klicken Sie auf Customer account	<input type="checkbox"/>		
<input type="checkbox"/>	5	+ Zeile 000006 für Feld AccountNum auswählen	<input type="checkbox"/>		
<input type="checkbox"/>	6	Skript beenden	<input type="checkbox"/>		

The screenshot shows the Script Editor interface. At the top, there are buttons for 'New step', 'Remove', 'Refresh', 'Copy', 'Paste', 'Recorder', and 'Quick actions'. Below the toolbar, there's a header with 'LIVE' and 'OPTIONAL' tabs, and a 'LABELS' section. The main area displays a list of 13 steps:

- 1 Start script 0.7 s
- 2 Gehe zu Accounts receivable/Invoices/All free text invoices 0.7 s
- 3 Klicken Sie auf New 1.9 s
- 4 Klicken Sie auf Customer account 0.9 s
- 5 Zeile 004005 für Feld AccountNum auswählen 1.3 s
- 6 Geben Sie den Wert Test in das Feld Description ein 1 s
- 7 Klicken Sie auf Main account 1.1 s
- 8 Klicken Sie auf die Schaltfläche Dialogfeld 1.1 s
- 9 Wählen Sie den Wert 1 (töt 10) für das Segment MainAccount des Feldes Main account 1.1 s
- 10 Geben Sie den Wert 5000 in das Feld Unit price ein 0.9 s
- 11 Klicken Sie auf Totals 0.9 s
- 12 Wert abrufen 0.9 s
- 13 End script 0.5 s

Below the list, there are tabs for 'General', 'Variables', 'Steps', 'Linked in', 'Advanced conditions', 'History', 'Work items', 'Changelog', and 'Documentation'. A 'Change status' button is also present.

If user recorded a script using specific translation and wants to generate report in another one, they don't need to run script once again after changing translation. Script steps and History executions are displayed in current translation selected, so that for one execution can be generated many reports using different translation. To do it go to **Settings -> Profile -> Basic Settings** and click **Edit**. In **Scripts and Reports locale** field select translation which should be used. After that click **Save**.

Note: It concerns only reports generated for specific execution. Reports generated in Documentation mode are translated permanently. To generate report in Documentation mode with different translation, another script run has to be performed. *More about Documentation mode in chapter 5.10. DOCUMENTATION tab.*

2.1.13 Report templates

This tab is used to add templates for documentation and reports generated. To create new Template click **Add**.

Report Templates						
	Name	Description	Created At	Updated At		
Global settings	SampleTemplate - renamed	SampleTemplate.docx	2020-07-15 09:44	2020-07-15 09:47		
Variables	SampleTemplate 3	SampleTemplate.docx	2020-07-15 09:18	2020-07-15 09:19		
Global variables	SampleTemplate	SampleTemplate (1).docx	2020-07-15 09:48	2020-07-15 09:48		
System definition	None	SampleTemplate (2).docx	2020-07-15 09:51	2020-07-15 09:51		
Storage	SampleTemplate 2	SampleTemplate.docx	2020-07-15 09:49	2020-07-15 09:49		
Virtual user	None	SampleTemplate (3).docx	2020-07-15 09:58	2020-07-15 09:58		
API key	Sample001	SampleTemplate.docx	2020-07-15 09:57	2020-07-15 09:57		
User profile	None	SampleTemplate (4).docx	2020-07-15 09:59	2020-07-15 09:59		
Import categories	None	None	2020-07-15 09:59	2020-07-15 09:59		
Locations	None	None	2020-07-15 09:59	2020-07-15 09:59		
Report Templates	SampleTemplate	test test	2020-07-15 09:59	2020-07-15 09:59		

Create report template

* Name:	Sample Template
Description:	Enter description
<input style="width: 100%;" type="button" value="Sample Template2 (1) (1).docx"/> X	

Create report template form will appear. Type **Name** for new template and add previously prepared .docx file. After that click **Create** at the bottom of the form.

To use created template go to **Settings -> Global settings** and click **Edit**. In **Documentation settings** section select created Template and click **Save** (see chapter [2.1.1. Global settings](#)).

2.1.14 License

This Tab License shows us all the important information about our application such as:

Ø Current License

Ø License Expiry Date

Ø Support Plan renewal Date

Ø Maximum Concurrent Sessions

Ø The Release notes

NOTE: If you want to change/update your License, please contact with support@executiveautomats.com or contact person from our Sales Team that you had contact with.

2.1.15. KPI counter

KPI counters are used to count an event when user decide that the condition is meet. This allows user to counter if for example some sort of actions, event or condition was executed correctly then count it using KPI. KPI can be used in advanced conditionals and are always increased the counter by one. In settings user can set up KPI counters:

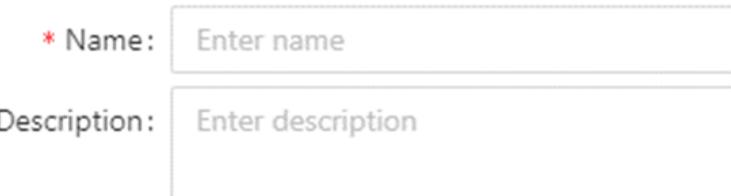


The screenshot shows the 'Kpi' section of the settings interface. On the left is a sidebar with various project management options like Global settings, Global actions, Sequences, etc. The main area has a header with 'Add', 'Remove', 'Refresh', and 'Edit' buttons. A table lists existing KPI counters with columns for 'Name' and 'Description'. The first row is expanded to show its details.

Name	Description
advancedelf	
advancedrules	
simplelf	
simplerules	
test	

Ø Add – allows to create new KPI counter:

Create kpi



The screenshot shows a 'Create kpi' form. It has two input fields: one for 'Name' with a red asterisk indicating it's required, and another for 'Description'.

* Name:

Description:

- o Name – in this field type the name of the KPI counter (it has to be unique value),
- o Description – optional field in which you can leave a description of KPI counter.
- o Create – confirms the creation of KPI and new KPI is added to the list,

- o Cancel – Closes the form “create KPI”,
- Ø Remove – removes selected KPI counters,
- Ø Refresh – refreshes the list of KPI counters,
- Ø Edit – allows to change details about selected KPI counter such as Name and Description.

2.1.16 Code

In Executive Automats you can create many different codes using JavaScript that can generate values, dates, actions etc. depending on users JavaScript knowledge. About using JavaScript code you can find later in user guide in sections related to Advanced Code or Code steps. In order to help our customers (since not everybody has knowledge of JavaScript) we have created a library of codes that can be used in the tool but also clients themselves can create such code.

2.1.16.1 Code Template Group

In this section you can create code templates group in which you later could store your Code templates.

Settings > Code Template Groups

Application	+ Add	Remove	Refresh	Edit
Global settings	<input type="checkbox"/>	Name	<input type="checkbox"/>	Description
Global actions	<input type="checkbox"/>	rererer		
Sequences	<input type="checkbox"/>	Helpers		Helper functions
Global variables	<input type="checkbox"/>	Odata		Odata examples
	<input type="checkbox"/>	Date123		Code templates related to date
	<input type="checkbox"/>	EAFunction		

- Ø Add – opens a form to create new Code Template Group:

Create code template group

* Name:	<input type="text" value="Enter name"/>
Description:	<input type="text" value="Enter description"/>

- o Name – name of the code template group (later cannot be changed!),
- o Description – optional description of the group,
- o Create – creates new Code Template group,

- o Cancel – closes the form.

This need to be setup BEFORE you set up any Code Template since it is necessary to at first create group for your Code templates.

2.1.16.2 Code Templates

Here you can create actual code templates that contains code written in JavaScript that can be executed and used in scripts.

Some of them will be implemented automatically by our development team so you can use them in scripts but is it possible to create new one.

Settings > Code Templates						
	Name	Description	Script name	Active version	Group	Created At
Global settings	00000000000000000000000000000000	test	test	1	main	16.05.2022, 15:56:11
Global actions	add			1	function	16.05.2022, 15:51:45
Sequences	test			1	function	16.05.2022, 15:51:24
Global variables	SAH002		SAH002	1	function	16.05.2022, 14:07:27
String definitions	Last day of previous month	Return last day of previous month as string or format yy-mm-dd	getLastDayOfPreviousMonth	1	Dev123	12.05.2022, 14:07:08
Iterators	Last day of next month	Return last day of next month as string in format yy-mm-dd	getLastDayOfNextMonth	1	Dev123	12.05.2022, 14:07:08
Helpers	Get random value	Returning random value from array	getRandomElement	1	Helpers	12.05.2022, 14:07:08
Virtual user	Substring	Return substring	getSubstring	1	Helpers	12.05.2022, 14:07:08
AFF key	Add week	Add days to current day and return string or format yy-mm-dd	addDaysToCurrentDate	1	Dev123	12.05.2022, 14:07:08
User profile	Negative to positive	Change negative number to positive	changeNegativeToPositive	1	Helpers	12.05.2022, 14:07:08
Project categories	Positive to negative	Change positive number to negative	changePositiveToNegative	1	Helpers	12.05.2022, 14:07:08
Users	Replace all	Change substring to new string	replaceAll	1	Helpers	12.05.2022, 14:07:08
Localizations	Customer account	Get customer account	getCustomerAccount	1	Dev	12.05.2022, 14:07:08
Report templates	Find value with number	Return find value with number	getFindValueWithNumber	1	Helpers	12.05.2022, 14:07:08
Issue tracking	Current date	Return current date as string or format yy-mm-dd	getCurrentDateAsString	1	Dev123	12.05.2022, 14:07:08
Others	Current date with hours	Return current date as string in format yy-mm-dd hh:mm:ss	addDaysToCurrentDateInHours	1	Dev123	12.05.2022, 14:07:08
	Add number of hour to custom date	Add value in minutes to date	addValueInMinutesToDate	1	Helpers	12.05.2022, 14:07:08

- Ø Add – opens a form that lets you create new Code Template:

Create a code template

* Name:

Description:

* Group:

* Function name:

Active version:

▼ Code section

Currently modified version:
+ Add function parameters

• * Code:

```
|
```

- o Name – Name of the Code Template,
- o Description – optional field to leave description of Code Template,
- o Group – group to which you want to add new Code Template,
- o Function Name – name of the function used in Code below (has to be unique),
- o Active Version – information about what version of code is activated now (you can have multiple version of same code template but only one version can be active and used in scripts),
- o Currently modified version – information about what version of code template you are currently editing/creating,
- o Add Function parameters – in code you can use parameters in your function, after clicking button fields pops out:

The screenshot shows a dialog box titled 'Code section'. At the top, it says 'Currently modified version: 1'. Below that is a button labeled '+ Add function parameters'. The main area contains two input fields: one for 'Name' with the placeholder 'Enter name' and another for 'Description' with the placeholder 'Enter description'. There is also a small red 'X' icon in the bottom right corner of the dialog.

§ Name – name of the parameter,

§ Description – description of the parameter,

- o Code – section in which you write your code using JavaScript. Code has to include Function name and parameters (if they were created),
- o Create – Code Template is created and added to list,
- o Cancel – closes the form.

Ø Remove – deletes selected code templates (if it is not used in anywhere),

Ø Refresh – refreshes the list of Code templates,

Ø Edit – open forms that enables edition of selected code template (you cannot change Function name and group during edition). You can change Name, Description, Active Version, Name of parameter and its description, Code and decide which version of the code you want to edit. After changing something in code section new options shows up:



- o Save Mode – Currently modified – means that changes in the code will be saved to currently modified version of code template,
- o Save mode – As a new version – means that changes in the code will be saved under new version of code template,
- o Save Mode – As a new active version – means that changes in the code will be saved under new version of code template and this version will be set up as active version.

Ø Export to file – download a file .eacodetemplate that can be later imported to another instance of EA,

Ø Import – in this form you can import any .eacodetemplate file to create code template,

Ø Used In – here you can check in which script selected Code template is being used.

NOTE: More about creating JavaScript code you can read at chapter [6.2.3.8. How to use Advanced Code](#)

2.2 Profile

An area that personalizes the settings assigned to the EA user.

2.2.1 Basic Settings

Basic settings are basic user data. An editable area with the possibility of making changes.

User with **Settings or Admin** role can make changes like:

- Single sign-on
- Username
- Email
- Language
- Issue tracking tool
- Execution report extension
- Display only published documents
- Scripts and reports locale
- Is active

NOTE: Only user with **Admin** role assigned to him can change **Roles, Single Sign-on and Is Active checkbox** assigned to other users and edit **Display only published documents** option for users.



2.2.2 Execution Settings

Execution Settings is a place where the user can parameterize, among others the method of play of the recorded **Script**. An editable area with the possibility of making changes.



2.3 Azure DevOps Integration

2.3.1 Basic Parametrization

Executive Automats allows connection with **Azure DevOps** projects. Integration with **Azure DevOps** allows EA to automatically create work items in **Azure DevOps** and use Pipelines to execute **Scripts, Projects and Schedulers or export published scripts as test cases to Azure DevOps Test Plans**.

First of all, we need to set up a connection between EA and Azure DevOps. Defined connection in EA is called **System Definition**.

To create **Azure DevOps** system definition go to the **Settings -> Issue tracking -> System definition**.



System definition tab contains columns:

- Ø **Name** – Name for the Azure DevOps connection.
- Ø **Host** – Azure DevOps address.
- Ø **Project** – Selected Azure DevOps project.
- Ø **Token** – Access token to the Azure DevOps.
- Ø **Created At** – Date and time of the created connection.

Click **Add** to create new connection.

On the next form, fill all necessary fields:



- Ø **Name** – Name for the Azure DevOps connection
- Ø **Host** – Azure DevOps address
- Ø **Access token** – Access token generated in Azure DevOps. Token is used to give EA access to the DevOps project. **Access token has to be generated by Azure DevOps user.**
- Ø **Fetch projects** – Use this button to test connection and to fetch projects from Azure DevOps.
- Ø In **Issue tracking** tab fill:
 - **Project** – Select project from connected Azure DevOps environment.
 - **Default iteration** – Select default iteration for work items.
 - **Default work item type** – Select default work item from Azure DevOps

- **Default priority** – Select default priority for created tasks
- **Default severity** – Select default severity for created tasks
- **Add custom fields** – Use this button to add custom fields from work item template in your Azure DevOps, which you want to fill while creating work items by EA (E.g. tags, parents or assigned user):
 - o **Name** – Select Name of the field work item template from the list. Name of the field can be displayed by moving cursor to this field and wait for few seconds.
 - o **Value** – Define value which will be typed to the added custom field.



To remove custom field use “Remove custom field“ button.

Ø If integration should be also set for Test Plans, next in **Test plans** tab fill:

- **Project** – Select project from connected Azure DevOps environment.
- **Default iteration** – Select default iteration for test cases.
- **Default priority** – Select default priority for created test cases



After all necessary fields are filled in, use **Create** button to create new connection. In the next step, we need to attach issue tracking system definition to the user.



To do this follow the path **Settings -> Profile -> Basic settings**

Next use **Edit** button to edit basic user options.

Select newly created system definition in **Issue tracking tool** field and then click **Save**.



2.3.2 Manual creating work items

After the basic **Azure DevOps** integration is parametrized, we are ready to create work items manually. Select **Script** which ended with error or warning and then go to its **History**.



Then select playback and click **Show errors/warnings** button.



Nie można wyświetlić obrazu.

Next dialog box with all errors/warnings appears.



Nie można wyświetlić obrazu.

Next users can select which error/warning they want to report to **Azure DevOps**, and then create work item by using button **Create work item**.



Nie można wyświetlić obrazu.

After work item is created you will get notification from the system and next you can use **Open work item** button to go through direct link to the work item created in **DevOps**.



Nie można wyświetlić obrazu.

The created work item contains all of the necessary information about **Script** along with error/warning message.

If the playback of the **Script** was done in background mode along with the Take screenshots function on, the screen- shot created while occurring bug will also be attached to the work item.

To create work item use **Create work item** button like in previous example.



Nie można wyświetlić obrazu.

The work item will be created with screenshot attached.



Nie można wyświetlić obrazu.

All created work items for **Script** are contained below **Work Items** tab.



Nie można wyświetlić obrazu.

From this tab there is also ability to use direct link to the **Azure DevOps** work item.

All of the **Scripts** that contains work items can be filtered by using **Show scripts with work items** button on the main **Script** tab.



Nie można wyświetlić obrazu.

To reverse filtering just use **Show all scripts** button.



2.3.3 Creating rules for automatic work item creation

There is functionality to define rules to create bugs/task/issues automatically when error occur during **Script**

playback. To define **Issue tracking rule**, go to **Settings->Issue tracking->Rules**.



Use **Add** button to add new tracking rule. Next rule creation form opens.



Fill up all the necessary fields:

Ø **Name** – Name for the Issue tracking rule

Ø **Issue tracking tool:** – Select issue tracking system definition

Ø **Triggers** – Select events that should trigger issue tracking rule to action:

- **Error** – Error occurred by Executive Automats
- **Warning** – Warning occurred by Executive Automats
- **Info** – Information generated by Executive Automats
- **Platform error** – Error generated by tested platform
- **Platform warning** – Warning generated by tested platform

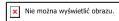
Note: You can selected more than one trigger.

Ø **Iteration** – Select default iteration for automatically generated work items.

Ø **Work item type** – Select default work item from Azure DevOps

Ø **Priority** – Select default priority for automatically created tasks

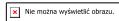
Ø **Severity** – Select default severity for automatically generated work items.

 Nie można wyświetlić obrazu.

After all necessary fields are complete, you can save rule by clicking **Create** button.

Then after the issue tracking rule is created, the rule needs to be attached to the project in which we want to use this rule.

To do this we need got to the **Projects** tab and select project we want to use. Next we select Issue tracking tab and click **Edit** button.

 Nie można wyświetlić obrazu.

Select created issue tracking rules and click Arrow button to attach rule to the project.

 Nie można wyświetlić obrazu.

When the rules are selected, click **Save** button.

When the Issue tracking rules are set up. Whenever **Script** from particular project generates error/warning, the rule will automatically create work items in **Azure DevOps**.

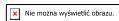
The work items created are available in the same way as manually created, in the work items tab.

 Nie można wyświetlić obrazu.

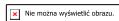
2.3.4 Executing Scripts/Schedulers/Projects using DevOps build Pipelines.

EA integration with DevOps allow also to execute projects/schedulers or separate scripts by using build pipelines form Azure DevOps.

To do this open your project in **Azure DevOps** and then move to the **Pipelines** tab.

 Nie można wyświetlić obrazu.

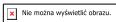
Then click **New Pipeline**.

 Nie można wyświetlić obrazu.

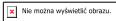
Use the **classic editor**.

 Nie można wyświetlić obrazu.

Next use any source and click continue. In our case we use **Azure Repos Git**.

 Nie można wyświetlić obrazu.

Then select **Empty pipeline** and click **Apply**.

 Nie można wyświetlić obrazu.

On the next form you can change name of new **Build Pipeline**. It has to be unique otherwise it will not run.

 Nie można wyświetlić obrazu.

Now we need select **Executive Automats agent**. To do so click on + button.

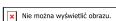
 Nie można wyświetlić obrazu.

When we use **EA-DevOps** integration for the first time, we need to search for **Executive Automats Integration Custom build** service from Microsoft Marketplace and then install it. Our plug-in is public in Marketplace and it is available for free:

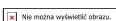
 Nie można wyświetlić obrazu.

Note: If after running Pipeline it will fail with an error like e.g. failed, reason: unable to verify the first certificate, select **EA integration Custom Build** plug-in.

If you have EA integration already installed, then type **ea** in search bar and select it.

 Nie można wyświetlić obrazu.

Then for the first use we need to define **Azure API service**, to do this click **New** button.

 Nie można wyświetlić obrazu.

 Nie można wyświetlić obrazu.

On the next form, we need to fill following fields:

Ø **Connection URL** – URL address to Azure DevOps

Ø **Personal Access Token** – Access token generated in Azure DevOps. Token is used to give AB access to the DevOps project.

Ø **Service Connection name** – Name of connection between EA and DevOps

When all fields are filled, click **Verify and save**.



Now click **New for EA API service**.

On the next form, we need to fill following fields:

Ø **Server URL** – URL address to EA environment

Ø **Type of entity to run in EA** – You can select one of following entities from EA:

- **Project** – You can select particular project(scripts included in this project) which will be executed by pipeline build
- **Scheduler job** – You can select Scheduler's job to be executed by build pipeline
- **Script** – select single Script which should be executed by build pipeline

Ø **API Token** – This is token generated by Executive automats in Settings/API Key. How to create API Keys is described in chapter [2.1.8. API key](#)

Ø **Service Connection name** – Name of connection between EA and DevOps

After the fields are completed click **Save** button.

Next in **Select test using** select Manual and regarding to **Type of entity** selected at previous window, select desired **Project/Scheduler/Script** in **EA executable to run** field.



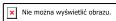
After everything is set up correctly then we can **Save and Queue** pipeline.



On the next window, you can add optional comment and then confirm pipeline by clicking **Save and Run** button. Then Agent will start working.

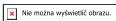


And particular entity will be executed in the Executive Automats. One the screen example **Script** entity is executed.

 Nie można wyświetlić obrazu.

Note: **API Key** column is hidden by default. To show it click on table columns settings in bottom left corner (cog icon).

After **Script** or other entity is executed then we are getting information about executions result. **Script** form screen example ended with an error.

 Nie można wyświetlić obrazu.

2.3.5 Test Plans Integration

Executive automats is also integrated with **Azure DevOps Test Plans**. It allows user to export scripts to **Test Plans** as **test cases**. It can be done in two ways:

Ø Export single script while Publishing (more about changing script status in *chapter 6.2.3.11. Changing the Status of the Script*):

 Nie można wyświetlić obrazu.

- Open script which should be exported as test case and Click Change status button
- In Status field select **Published**

Note: Only Published scripts can be exported to Test Plans!

- Check **Export to test case checkbox**
- Select previously created **Issue tracking tool definition**
- **Project, Iteration and Priority** fields will be automatically filled with default values, but they can also be changed
- When everything is ready, click **Ok**
- Script status will be changed to **Published**, which means it is not editable now
- In script **General tab** will appear:

 Nie można wyświetlić obrazu.

Attached test case field, where user can see which test case is attached to this script. It means that if the attached test case will be run by Pipelines, this script will be also run in EA

Open attached test case button. Clicking it opens created test case in **Azure DevOps Test Plans**



- Created **Test Case** contains exactly the same steps as the script in EA
- The **Test Plan** for this test case will be created based on **main Project** for the exported script. If between the script and the Project sub-projects are created, they will be exported as **Test suites**

Note: If Project and sub-projects have been exported to Test Plans previously, the new test case will be created in existing Test suite/Test Plan

Export whole Project with all **Published** scripts

- In **Projects tab** click **Add-ons, Export to Test Plans**
- In **Export to Test Plan** form select:
 - § Previously created **Issue tracking tool** definition

Project, Iteration and Priority fields should be automatically filled with default values, but they can also be changed

§ Projects/sub-projects which should be exported

Note: If only Project is selected, none of its subprojects will be exported. To export subprojects they also have to be selected

§ When everything is ready click **Export**



After that an information about success export will appear. In exported Project, in **Details** tab will appear:

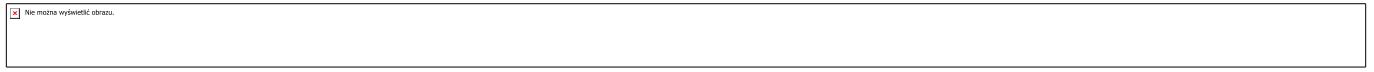
- **Attached test plan** field, where user can see which test plan is attached to the Project. For sub-projects it will be **Attached test suites**
- **Open attached test plan** button. Clicking it opens created test plan in **Azure DevOps Test Plans**



2.3.6 Executing Test Plans using DevOps build Pipeline

Similar as with scripts, schedulers and projects execution, exported test cases and attached scripts can be run by using build pipelines from Azure DevOps

Setting up Pipeline process is almost the same as this one explained in previous chapter – once **Azure API service** and **EA API service** are added, next in **Select test using** select **Test plan**, in **Test plan** and **Test suite** select which test suites from which test plan will be executed.



After everything is set up correctly then we can **Save and Queue** pipeline.



On the next window, you can add optional comment and then confirm pipeline by clicking **Save and Run** button.

Then Agent will start working and particular entity will be executed in the Executive automats.

At **API Key** column there is visible which **API Key** was used to execute particular entity.

After that, run test cases will have updated status and it will be the same as the attached scripts.



2.3.7 Executing Test Plans and Test Cases from EA with results in DevOps

The second way to run scripts and have outcome in Azure DevOps Test Plans is to do it in EA. Open previously exported script and expand the Playback options.

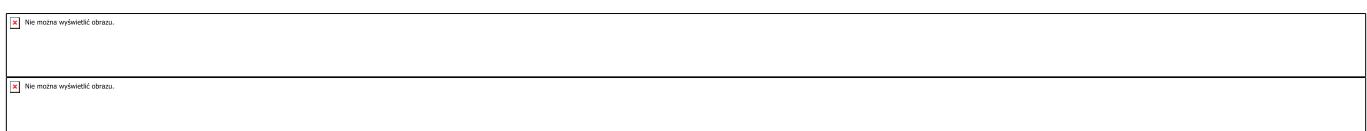
Next check **Results in DevOps** checkbox (this option is available in both **Foreground** and **Background** executions) and click **Run**.



The outcome of the test case in DevOps will change to **In Progress**.



When script execution will be finished, Outcome will be the same as the execution result.



If script has been played in **Documentation mode** (more about it in *chapter [5.10. DOCUMENTATION tab](#)*), or during the execution any files have been captured (how to capture files is explained in *chapter [6.6. Capturing files](#)*) and with **Results in DevOps** option enabled, the test case run will contain generated document and captured files.



Just like single script, exported Projects/subproject also have **Results in DevOps** option.

In **Projects** tab select exported Project. Then in **Details** tab click **Edit** and check **Results in DevOps** checkbox. After that click **Save**.



Next click **Actions** and select **Schedule execution** (more about Scheduled execution in *chapter [4.3. Scheduled execution of the project](#)*).

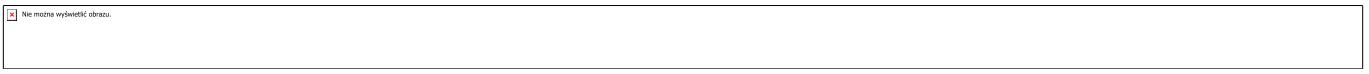
After that all exported scripts from the Project will be executed and similar as for single scripts, test cases outcomes first will change to **In Progress** and then to appropriate status to attached scripts results. After checking option Result in DevOps new field will appear called Configuration. Here you can determinate with which configuration new test suite should be created in DevOps. For more about Configurations please refer to *chapter [2.3.10 Configurations](#)*.



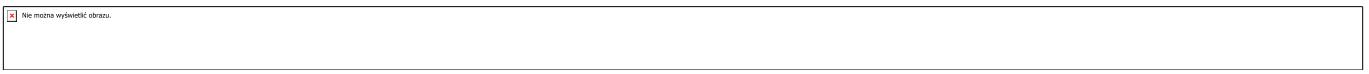
2.3.8 Opening test case and test result using History tab

When execution with **Results in DevOps** will finish, users can easily go to the run details in Azure DevOps Test Plans.

Go to the **History tab** (no matter if main History or exported script/Project), select desired execution, click **Azure DevOps -> Open test result**.



In new page test case run will open.



To make finding those runs easier, click **Azure DevOps -> Show test run executions**. Then the table will show only executions with **Results in DevOps enabled**.



Using **Open test case** option users can open exported test case (similar as using **Open attached test case** button in **General tab**).

2.3.9 Closing Test Plans and Test Cases

If users decide to close exported test case or remove test plan, they can do it while closing script or removing project.

To close test case in Azure DevOps Test Plan, open attached script and click **Change status**.



Change status form should appear. In **Status** field select **Closed** (more about changing script status in chapter [6.2.3.11. Changing the Status of the Script](#)), check **Update test case status** checkbox and click **Ok**.

Script status will change to **Closed**. Go to **General tab** and click **Open attached test case** to see that its status has also changed to **Closed**.



To reopen the test case again change status of the script with **Update test case status** enabled, this time in **Status** field select **Draft**.



To remove whole test plan in Azure DevOps Test Plans go to the **Projects tab**, select desired Project and click

Remove.

In displayed dialog box check **Remove connected Test Plan / Suite?** checkbox and click **Save**.



Note: Only Project in which all scripts are closed can be removed. After that both Project and attached Test Plan will be removed.

2.3.10 Configurations

Configurations allows us to create or fetch configuration setup from Azure DevOps. A test configuration is a combination of configuration variable values. Your configuration variables could be, for example, operating system, browser, CPU type, database. A configuration might be "Windows 8 + 32-bit CPU" or "Windows 10 + 64-bit CPU." This allows us to ran tests scripts with different configurations without need of creating copies same test scripts. Configurations setup work in both ways. If new configuration is created in Executive Automats it will be also created in Azure DevOps. If you create new configuration in Azure DevOps then by using option Fetch it will be exported to Executive Automats.



Configurations contains below functions:

Ø **Fetch** – allows to export new configurations created in Azure DevOps and imports them into Executive Automats,

Ø **Add** – allows to create new empty configuration in Executive Automats. New configurations (if Azure DevOps connections work correctly) will be automatically exported to Azure DevOps,



- o Project – determinate the Project from ADO in which new configuration should be created
- o Name – in this field, insert Name of your configuration,
- o Description – in this field, insert additional description,
- o Select Active – in this field determinate if the created configuration is activated or not,

Ø **Remove** – deletes selected configurations from Executive Automats,

Ø **Refresh** – refreshes page,

Ø **Edit** – allow to change details about selected configuration

NOTE: For this option to work correctly, user have to have configurate connection between Executive Automats and Azure DevOps correctly (for that please refer to section: [2.3.1 Basic Parametrization](#)).

3. History module

In the **History** module, each **Script** run is registered, so that you can check the functionality of the function after each play. If the execution does not work correctly, the tool logs the fault in the form of a warning or error along with the reason for its occurrence.



History module header contains following functions:

1. **Refresh** – refreshing the list.
2. **Statistics** – shows statistics about all of **Scripts** playbacks.
3. **Generate report** – allows user to create **MS Word** or **pdf** file with listed steps and screenshots taken during **Script** playback. This type of report can be used as instruction of the process recorded. User can select report format in **Settings/Profile/Basic Settings/Execution report extension**.
4. **Terminate** – allows to stop playback of the **Script**, with State “Running”.
5. **Screenshot gallery** – contains screenshots taken during **Script** recording/playing.
6. **Show errors/warnings** – displays all execution errors and warnings with detailed information.
7. **Export to file** – used to export full playback history to MS Excel file.
8. **Captured files** – displays files which have been captured during that session.
9. **Azure DevOps** – integration with Test Plans (see chapter [2.3.7 Executing Test Plans and Test Cases from EA with results in DevOps](#)).

a. **Open test case** – if executed script has been exported to Azure DevOps Test Plans, clicking this button will open created test case.

b. **Open test result** – if previously exported script has been executed with **Results in DevOps** option enabled, clicking this button will open run in Azure DevOps Test Plans.

c. **Show test run executions** – clicking this button will show only executions with **Results in DevOps** option enabled.

10. **Performance counters** – this feature allows you to check statistics in relation to used KPI counters in selected script. More about statistics you can refer to chapter [3.2. Performance counters statistics](#).
11. **Show active sessions** – This feature allows you to check active sessions or logs for sessions that have ended.
12. **Go to** – allows to move to the given **Script**.
13. **Open** – used to display detailed information about **Script** playback.

3.1 Execution/play states

The execution/play states of individual Scripts can have the following values:

- Ø **Succes** – Script correct, play completed successfully
- Ø **Warning** – warning; The Script contains errors, however, it does not cause interruption of Script play
- Ø **Error** – error in Script play; interrupts further play
- Ø **Running** – Script is running
- Ø **Terminated** – Script play has been interrupted by the User

3.2 Performance counters statistics

After selecting script from the history (that used in some way KPI counter in it) you can check statistics about performance counters. The graph shows how often KPI counter was increased during the execution of the script. If multiple KPI counters were used in the script all of them will be listed.

- Ø **KPI Name** – name of used KPI counter,
- Ø **Total occurrences** – how often did KPI was increased during the execution,
- Ø **Occurrences per minute** – how often KPI was increased per minute of execution,
- Ø **Occurrences per hours** - how often KPI was increased per hour of execution.



4. PROJECTS module

The **Projects** module is used to create and manage projects made in the Executive Automats. **Scripts** are assigned to relevant projects and subprojects, which allows them to be easily grouped. This module also allows you to manage the level of access of users, to specific projects, as well as to limit virtual users and systems that can be used to record and play **Scripts** contained in a given project. Selecting project is required during the **Script** creation. Projects module can be reviewed only by users with **Projects** role assigned to them.



4.1 Creating project

Projects can only be created by users that have Admin role assigned to them. User roles are described in *chapter 2.1.9. Users*.

To create new **Project** click **Create** button.



Then enter the name of the **Project** and use checkmark button to save the project.



After creating project there is also possibility to create subprojects. To do that select **Project(parent)** and then click again **Create** button to create **Subproject(child)**.



Projects and **Subprojects** can also be deleted. In this case select unwanted **Project** and click **Remove** button.

Note: Projects can be removed only by users with **Admin** role assigned to them.

Note: You cannot delete **Project** if it contains any **Scripts**.

4.2 Projects interface

4.2.1. Details tab

Details tab contains main information about **Project**. Users with access to given project and with **Projects** role assigned to them can add and edit all necessary information to the project.

Information that can be managed in this tab:

- Ø **Name** – name of the project
- Ø **Category** – project category assigned to project
- Ø **Version** – version of the project
- Ø **Start date** – project start date
- Ø **Due date** – project end date
- Ø **Description** – project description
- Ø **Project owners** – owners of the project
- Ø **Run mode** – mode of the playback of scripts for scheduled execution functionality
- Ø **Status to run** – only scripts with selected statuses will be run in scheduled execution
- Ø **Documentation** – By checking this checkbox, all scripts run during Project Scheduled execution (more about it in chapter [4.3. Scheduled execution of the project](#)) and finished with status **Success** will have generated documentation. When this checkbox will be marked two additional fields will appear:
 - o **Mode** – here select what type of documentation should be created. Options are Document, video or both. About them read more in chapter Documentation)
 - o **Reports locale** – selected localization in this field overrides user settings
 - o **Doc template** – selecting template in this field overrides selection in **Global settings** (see

chapter [2.2.1 Basic Settings](#)

- Ø **Report template** – selecting template in this field overrides selection in **Global settings** (see *chapter*
[2.1.1. Global settings](#))

Ø **Attached test plan** – This function allows you to check and access the test plan of the DevOps to which the project is connected.

Ø **Result in DevOps** – function allows the results to be passed to DevOps.

- o Configuration – allows you to select the configuration in DevOps to which the result will be saved.

4.2.2. Scripts tab

All **Scripts** created within the project are displayed in **Project Script** tab. There is possibility to manage scripts from this area for example creating, playing or deleting scripts.



4.2.3. Test results tab

Test results tab will show **Progression chart**, if **Project Scheduled execution** has been run at least two times. More about this tab in *chapter 4.3. Scheduled execution of the project*.

4.2.4. Documented Scripts tab

In **Documented scripts** tab are displayed scripts, for which reports have been created during Background Playback (more about creating documentation in *chapter 5.10. Documentation tab*).

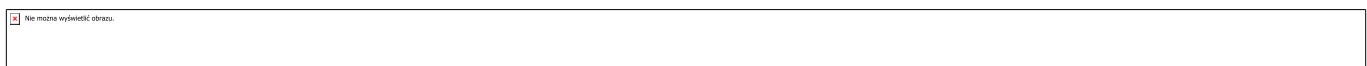
From this tab it is also possible to download such report. Select a script with created documentation and click **Download documentation**.



4.2.5 History Tab

Project History tab contains information of every playback of scripts that are assigned to the given project. All functionalities described in *chapter 3 History module* are also available from this place.

Additionally **Project History** tab allows to Aggregate log for specific Scheduled execution. Click **Aggregated logs** button, each Scheduled execution will be displayed with details such as **ID**, **Start & End Time**, **State**, **Status**, **Duration**, numbers how many scripts were finished with specific status and if the log is **Excluded**. Each log can be opened to see history details for each specific execution.



To change the view back, click **Non-aggregated logs** tab.



4.2.6. User groups tab

In this area users with **Admin** role assigned to them, can select **User groups** (user groups described in *chapter 2.1.7. User groups*) which can access to particular projects or subprojects.

To assign user group to the project select specific **Project/Subproject** and on the **User groups** tab click **Edit** button.

 Nie można wyświetlić obrazu.

Then select which **User group** should have access to given **Project/Subproject**.

 Nie można wyświetlić obrazu.

Next after all **User groups** are assigned to the project click **Save** button.

 Nie można wyświetlić obrazu.

Adding **User groups** to project will give access to this particular project to each user that is assigned to those groups.

There is also possibility to remove user group from project. To do that just use opposite arrow and then **Save** the changes.

 Nie można wyświetlić obrazu.

All User groups added/removed from the upper Project (parent) will be automatically also assigned/removed from Subprojects (children) below.

In case we want **Subprojects** to not inherit **User groups** from parent **Project** above then select given **Subproject** and mark **Override** checkbox.

 Nie można wyświetlić obrazu.

Next **Edit** button will be available to use and we can override **User groups** assigned to this **Subproject** and all other **Subprojects** (children) below it.

 Nie można wyświetlić obrazu.

4.2.7. Virtual Users tab

Project **Virtual users** tab is used to manage virtual users within project. In this area we can specify which from **Virtual Users** existing in the system could be used within **Project/Subproject**.

Adding or removing virtual users from the Project can be done only by users with Admin role assigned to them. To assign **Virtual user**, select specific project and then click **Edit**.

 Nie można wyświetlić obrazu.

On the next form select **Virtual Users** which you want to assign to the given project.

 Nie można wyświetlić obrazu.

Next if **Virtual Users** are assigned click **Save** button.

From this form, there is also possibility to **Remove Virtual User** from project. You can do this by selecting a **Virtual user** on the right side and by removing it from the project. After a **Virtual User** is removed from the project **Save** changes.

 Nie można wyświetlić obrazu.

Virtual Users can only be deleted by users with **Admin** role assigned to them.

All **Virtual Users** added/removed from the upper **Project (parent)** will be automatically also assigned/removed from **Subprojects (children)** below.

In case that we want **Subprojects** to not inherit **Virtual Users** from parent **Project** above then select given **Subproject** and mark **Override** checkbox.

 Nie można wyświetlić obrazu.

Next **Edit** button will be available to use and we can override **Virtual Users** assigned to this **Subproject** and all other **Subprojects** (children) below it.

 Nie można wyświetlić obrazu.

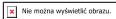
On the **Virtual Users** tab you can also set up **Default Virtual User** for project. To set up default **Virtual User**, select user and then click **Set as Default** button.

 Nie można wyświetlić obrazu.

4.2.8. Systems tab

This tab is used to manage **System definitions** within projects. In this area you can assign **System definitions** which can be used within particular project. Users when creating **Scripts** for the given **Project** can use only those **System definitions** which are assigned to this **Project**.

Assigning **System definitions** to the project can be done only by users with **Admin** role assigned to them.



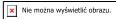
To add **System Definition** to particular project click **Edit** button.

Next select **System definition** which you want to assign to the **Project**.



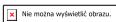
After **System definition** has been added to the **Project**, click **Save** button.

While editing **System Definitions**, you can also remove **System Definition** form the project by using opposite arrow button. Next **Save** changes.

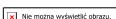


All System definitions added/removed from the upper **Project (parent)** will be automatically also assigned/re- moved from Subprojects (children) below.

In case that we want **Subprojects** to not inherit **System definitions** from parent **Project** above then select given **Subproject** and mark **Override** checkbox.



Next **Edit** button will be available to use and we can override **System definitions** assigned to this **Subproject** and all other **Subprojects** (children) below it.



On the **System definitions** tab you can also set up **Default** system for project.

To set up default **System definition**, select system for the list and then click **Set as Default** button.



4.2.9. Issue tracking tab

In this area user can add previously set rules to automatic work items creation (More about rules in *chapter*

[2.3.3. Creating rules for automatic work item creation\).](#)

4.3 Scheduled execution of the project

Schedule Execution functionality is used to test projects in Executive Automats. The **Schedule Execution** functionality runs all of the scripts included to the project. There is also possibility to run only the Scripts with selected script Status. Results of the test are displayed on diagram in project details tab and the detailed results can be investigated in Test results tab. **Scripts** can be ran in two playback modes.



Ø **Sequential** – scripts are run one by one, next Script will start after the previous one will end

Ø **Parallel** – parallel execution of scripts means that many scripts from the project will be run simultaneously. The number of scripts that can be run at the same time depends on the server resources. If Parallel option is set up then new field appears **Max parallel sessions**. It allows us to determinate how much of session should ran at same time.

To test project by using scheduled execution click on the project you want to test, next click **Actions** and then **Schedule** execution button.



After execution is started, on the right side there will be displayed diagram which shows status of the execution.



During execution diagram change and display information about statuses of the **Scripts** executed.

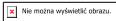
Below the diagram chart with last 5 executions results is displayed (Note: chart will be visible after at least 2 executions performed!)

After execution is over, we can easily compare results with previous executions on the chart in **Test Results** tab.



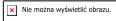
The chart shows Project Scheduled execution Job ID and Script count for each execution. By moving cursor on specific execution user can see number of scripts for each status.

If users want not to include specific log in **Progression chart**, go to **History tab**, in **Aggregated logs** view select specific execution and click **Exclude results**.



Nie można wyświetlić obrazu.

Such log will be marked in **Excluded** tab. Also it won't be included in **Progression chart** in **Test results** tab.



Nie można wyświetlić obrazu.



Nie można wyświetlić obrazu.

To include excluded log, select it in **History** tab and click **Exclude results** once again.

4.3.1 Scheduled execution of the project by Script status

Instead of executing all of the **Scripts** from the particular project, there is possibility to let user choose which scripts should be executed. You can limit executed Scripts from the project by selecting **Scripts** statuses.

On the **Project details** tab, we can specify which **Scripts** with which **Statuses** should be executed with **Scheduled Execution**.

To run **Scheduled Execution** based on **Script** statuses go to **Projects**.

Then select desired **Project** and click **Edit**.

In the field **Status to run**, you can select one or multiple script statuses which should be executed.



Nie można wyświetlić obrazu.

Then you can use **Scheduled Execution** of the project in the standard way and only **Scripts** with selected statuses would be executed.



Nie można wyświetlić obrazu.

5. SCRIPTS module

Scripts in the understanding of EA are nothing but sequences of actions performed by the user. These activities are saved in the form of a **Script** that allows you to replicate user actions in the system.

5.1 Function interface scripts

The **Scripts** module includes the following buttons:



1. **Add** – allows to add a new Script (specify its name, environment and add a description).
More details in [chapter 6. How to record a Script](#)
2. **Close** – allows to delete a selected record, Closed script is considered as deleted
3. **Clone** – allows to create duplicate of the script
4. **Export to file** – allows to export selected scripts to .ea file
5. **Import** – allows to import .ea file with scripts. More details about Import and Export in [chapter 6.5 Import/Export scripts](#)
6. **Refresh** – refreshes the list of Scripts after the changes
7. **Show/hide closed** – shows / hides closed Scripts. Note: you can not re-work with the closed Script. It is not possible to apply the functionality of adding steps (referred to in [chapter 6.2.2 Adding steps to a Script](#)).
8. **Show scripts with work items** – shows only those Scripts which have created work items
9. **Export to csv** – exports data from the table to .csv file. This button contains two additional functions:

Ø **Export to CSV** – export data like Id, Name, Project Name, Environment, Env.Type, Status, Created At, Created By.

Ø **Export with additional fields** – This function allows more parameters to be downloaded into the CSV file, such as:

- o Take screenshots
 - o Screenshots only on interruption
 - o Click on disabled buttons
 - o Use default environment
 - o Top project name
 - o Virtual user name
10. **Open** – opens the form with the steps of the selected Script
 11. **Start** – starts play of the selected Script *

*More about custom options for the play of the selected Script in [chapter 6.4.3 Non-standard playback of the Script](#).

The table with the recorded **Scripts** contains the following headings:

- **Id** – Script number
- **Name** – Script name
- **Project name** – Name of the project in which Script is created
- **Environment** – Name of the environment on which the Script was recorded
- **Environment Type** – Type of environment on which the Script was recorded
 - Ø **Development** – Development/programming environment
 - Ø **Test** – Test environment (tests carried out by EA developers)
 - Ø **Production** – Production environment that is usually a simulation of the actual production environment (tests performed after accepting the functionality by the Customer)
 - Ø **User Acceptance Tests** – Environment that checks the system for compliance with customer requirements (test cases are performed by the client with the participation of project representatives)
 - Ø **Other** – Other environment created in case of other additional needs
- **Status** – Status of the Script
 - Ø **Draft** – the Script has been recorded but not yet published
 - Ø **Published** – the Script was published, its model version was selected
 - Ø **In Progress** – the Script has been recorded, but edition hasn't finished yet
 - Ø **Ready for Review** – the Script is ready to be checked
 - Ø **Ready to Test** – the Script is ready to be tested
- **Created At** – the date and time the Script was created. The ability to filter out after indicating a specific date
- **Created By** – the person recording the script; user defined earlier in settings (Settings)

The **Script** form allows to register new **Scripts**, as well as view and modify all previously registered **Scripts**.

To go from the list of all recorded **Scripts** to edit the selected **Script**, double-click it. The system will open a form in which the tabs will be visible:

Ø **General**

Ø **Variables**

Ø **Steps**

Ø **Linked in**

Ø **Advanced Conditionals**

Ø **History**

Ø **Work items**

Ø **Changelog**

Ø **Documentation**

5.2 GENERAL tab

The **General** tab contains all the information about the recorded script that was entered during the creation of the **Script**.

5.2.1. General Information

General Information – data defined during the creation of the **Script**. Use **Edit** button to make changes.



The **General Info** section contains following fields and controls:

Ø **Project** – allows to change project for the Script

Ø **Name** – allows to change name of the Script

Ø **Manual URL/Environment** – allows to change environment for the Script or use of manually typed URL address

Ø **Use project default environment** – if the checkbox is selected the script will be run on the environment (system definition) that is default for the project

Ø **Use Virtual User** – if this toggle box is on it allows to use project default virtual user or different one. If this is off then no virtual user will be used.

Ø **Use project default virtual user** – if the toggle box ‘Use virtual user’ is on and the checkbox ‘Use project default virtual user’ is marked the script will log in to the environment using default Virtual User for the project

Ø **Description** – description of the Script

Ø **Execution timeout** – allows to set up maximum time of the playback after which Script will be stopped

Ø **Take Screenshots** – by selecting this parameter Executive Automats creates screenshots after every step during playback of the script. This functionality can be used only while background playback of the script

Ø **Screenshots only on interruption** – by selecting this parameter Executive Automats creates screenshots only on interruption. This functionality can be used only while background playback of the script

Ø **Show overlay hint** – by selecting this parameter every element of the system used by Executive Automats during playback of the script will be marked with overlay hint

Ø **Hide comments** – select this option to hide comments added on the steps of the Script

Ø **Click on disabled buttons** – if the checkbox is selected the script will be able to click on disabled buttons during playback without errors occurring

Ø **Disable start step** – If the checkbox is selected, it will cause the Start script step to be skipped when this script is executed. Start script step is responsible to reload environment and start from position of the URL typed in System definition tab.

Ø **Auto-refresh datasource** – if the checkbox is selected data in attached Excel file will be refreshed before script execution (can be used e.g. when spreadsheet contains some formulas)

Ø **Custom resolver mode** – if the checkbox is selected you can determine which step detection mechanism will be used for this script, Smart, mix or Legacy.

Ø **Documentation settings** – this section allows users to override settings for templates (more about it in

[*chapter 2.1.1. Global settings*](#)

5.2.2. Data Source

Data Source – a place where users can upload an Excel file with data that will be used within the script steps. Additional option is to generate data source template and then fill all fields and upload an Excel file with filled data.



How to put an Excel file in **Data Source** and how to use it in working with the recorded **Script** is described in

[*chapter 6.2.3.5 Working with an Excel file.*](#)

5.3 Variables tab

Variables are created for later use in selected **Script** steps (among others when using **Inspect** and recording the **Get value** step). The created variable can be used in a situation where for several steps we want to use the same value.

5.3.1 Defining local variables

A defined local variable can be used only in the script in which it is defined. To define a local variable for a given **Script**, go to local and click **Edit**.



Then click **Add** to add the variable.



Enter the **Variable name** and specify the data type:

- Ø **Text** – variable in text format
- Ø **Number** – variable in numerical value
- Ø **Date** – variable in date format



Click **Save** to save the defined variable.



5.3.2 Defining global variables

A global variable can be used if the user wants to easily pass the value between scripts or when a user wants to use a constant value for many scripts (for example the same account number for many scripts). A global variable created in a script will be visible on the list of all global variables in **Settings/Application/Global variables**.

In the script the user can create and assign a brand new global variable or use an existing one. To define a new global variable for a given Script go to global and click **Create and assign**.



The remaining steps of creating a global variable is described in [chapter 2.1.4. Global Variables](#) in this User's Manual.

To assign an already existing global variable to a Script go to global and click **Assign**. Desired global variables must be marked. Click **Add** to assign marked global variables to this Script.



Global variables can be unassigned from a Script. To perform this operation mark desired global variables and click **Unassign**. Unassigned global variable will be removed from the list of assigned global variables In the Script.



5.3.3 How to use a variable on a Step

Both local and global variables can be used on Steps in a Script.

To assign a local variable to a step, mark the step, click Edit, mark Picker and select a variable from Script.



To assign a global variable to a step, mark the step, click Edit, mark Picker and select a variable from Global type Constant or Global type Variable.

NOTE: Global variables will not work when linking scripts. Please use Local ones and override variable option to make it work correctly.



5.4 Steps tab

The functionality of this tab is described in detail in [chapter 6. How to record a Script](#).

5.5 Linked In tab

This tab shows work items created in Azure DevOps (more about it in [chapter 2.3.2. Manual creating work items](#))



5.6 Advanced Conditional tab

Advanced Conditionals are used to create some kind of **Guard** which monitor playback of the **Scripts**. During creation of the **Advanced Conditional** we can select **step or steps** which should be monitored and then we can set up automatic response to the events which can occur during playback. This kind of functionality allows to create complicated conditions in the test cases e.g. **Negative tests**.

5.6.1 Defining Advanced Conditionals

To define **Advanced Conditional** go to the Script and then open **Advanced Conditionals** tab and click **Add** button.



The **Advanced Conditional** creation form opens.



First enter the name of the **Advanced Conditional**. Next select which **Step or Steps** should be guarded.



Then choose which type of the event should trigger **Guard's** reaction. We can select two types of events:

Ø **Success** – Guard's reaction would be triggered when one of the selected steps ends up with success

result

Ø **Interruption** – Guard's reaction would be triggered when one of the selected steps ends up with specific type of interruption.

After selecting event type **Success** we are able to select type of **Guard's Activity**. We can select two types of **Activity**:

Ø **Simple** – Allows to select one simple reaction to the specified event. The options which can be

selected:

- **Continue** – Continue the script playback.
- **Log and continue** – Allow to continue script playback, but also to Add notification which would be visible on the screen and also saved in the Script history.
- **Warning** – Allow to continue script playback, but also causes Warning message to be visible on the screen and also saved in the Script history. Step status is set to Warning. Warning message can be specified.
- **Stop and show the error message** – Stop Script playback with an error. Error message can be specified.
- **Take a next row from data source and jump to a step** – Allow to take another row from the Excel file and then jump to another step.
- **Jump to a step** – Allow to jump to other steps from the script.
- **Increase KPI** – allows to increase selected KPI meter by 1 which can help user count if some actions were performed couple of times. About KPI you can learn more at [Chapter 2.1.15 KPI Counter](#).
- **Repeat** – Repeat the step

Ø **Advanced** – Allows to select more complicated reaction to the particular event. Advanced Activity type form is the same form as form of the **Conditional step** which is described at chapter [6.2.3.10.2. Conditional step as data validation](#).

After selecting **Interruption**, select specified type of interruption which **Guard** should react on.

Interruption events which can trigger **Guard**:

- System error – During playback of the Script, the error occurred in the system which interrupts further playback
- System warning – During playback of the Script, the warning occurred in the system, but it doesn't cause interruption of Script play
- Element not found – Element of step was not found in the tested system during playback of the Script.
- Warning – the script contains warning from Executive Automats
- Error – the script contains error from Executive Automats that stops it,



To add more than one reaction to interruption event click **Add reaction**:



Similar to the previous **Success** type on the **In- tERRUPTION** event time we can select one of two **Activity types**:

Ø **Simple**

Ø Advanced

In both of event types and despite of **Guard's** reaction there is possibility to check **Override status checkbox**.

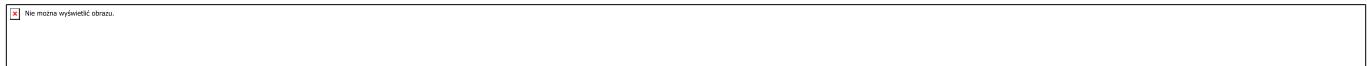


After checking this it is possible to change status of the **Guarded step** to one of the other **Statuses**.

This kind of using **Advanced Conditional** can be used in **Negative testing**, for example if **Guarded step** ends up with an error (and it is desired output of the test) we can override it's status to **Success**.

5.7 History tab

Each script run is registered in the **History** tab, so you can check the correctness of the function in each play. If the function does not work correctly, the tool logs the fault in the form of a warning or error along with the reason for its occurrence.



1. **Session id** – the number of the session during which the Script was executed
2. **Project** – the name of project in which Script exists
3. **Environment** – name of defined system definition used during execution start
4. **Virtual user** – name of the Virtual user used during execution start
5. **Start time** – the date and time of the start of the session
6. **End time** – the date and time of the end of the session
7. **Duration** – the time in which the task in a given session was performed
8. **State** – the state of the task performed
9. **Status** – the status of the task performed in a given session*

*particular statuses are described in the module [3. History module](#)

10. **Username** – name of the user who ran the script
11. **API Key** – the access key used to integrate with other application (e.g. Azure DevOps)
12. **Configuration** – configuration used for the script execution (related to Azure DevOps)

Note: **Session Id** and **API Key**, **Configuration** columns are hidden by default. To show it click on table columns settings (cog icon) in bottom left corner.

It is possible to generate statistics of all executions/plays of a given **Script**. If you want to view them, click **Statistics**.

 Nie można wyświetlić obrazu.

The system will generate information in the form of a pie chart containing data on the number of executions and statuses of tasks.

 Nie można wyświetlić obrazu.

5.8 Work Items tab

This tab shows work items created in Azure DevOps (more about it in [chapter 2.3.2. Manual creating work items](#)).

5.9 Changelog tab

The **Changelog tab** contains information of all status changes of the **Script**. Statuses and it's changes are described in [chapter 6.2.3.11. Changing the Status of the Script](#).

 Nie można wyświetlić obrazu.

Changelog tab contains following columns:

1. Previous status - Information about previous Status (before change)
2. New status – Information about New Status (after the change)
3. User – name of the user which made a change to the Script's Status
4. Updated at – The date of the Status change
5. Comment – Comment added during Status change

In Changelog tab there is also possibility to see steps which Script contained in each Status. To do it click “+” next to desired status.

 Nie można wyświetlić obrazu.

User also can Compare Script steps between selected statuses. Mark two statuses and click Compare.

 Nie można wyświetlić obrazu.

After that the Compare box will appear with such information like: status change and steps list for both statuses.

 Nie można wyświetlić obrazu.

After that the Compare box will appear with such information like: status change and steps list for both statuses.

5.10 Documentation tab

The **Documentation** tab contains all reports created during script Background Playback.



To create such documentation expand Playback option (more about playing a script in [chapter 6.4. Playback of a recorded Script](#))

Then select **Background**, check **Documentation** checkbox and click **Run**.



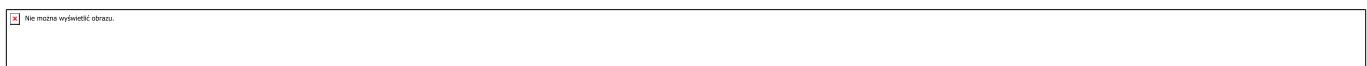
The system will display information about the action being started:



When the Playback will finish, in **Documentation** tab new report with status **Draft** will appear. It will use translation selected in user **Basic settings** (more about translations in [chapter 2.1.12. Localizations](#)) and template selected in **General settings** (more about templates in [chapter 2.1.13. Report templates](#)) unless it has been overridden for the Script or whole Project

Note: It can be only one report with status draft. If a report has been already created, it will be overridden by the new one.

If a report shouldn't be overridden, select it and click **Edit**.



In **Edit documentation** form, in **Status** field, change status to **Published** and click **Save**.



From now, this report won't be overridden, and after the new Background Playback with document creation,

new report with status **Draft** will be generated.

Note: One script can have many published documents



Nie można wyświetlić obrazu.

Documentation can be also created in form of Video or both Video and Documentation in same time from same Executions. After selecting option Documentation new field will pop out called Mode.



Here you can determinate if documentation should be created in text form or video.



If Video option is chosen then new fields appears. You can determinate if player should be seen on the video in option Player view.

You can also slow the execution in field Slow Execution. You can also decide if video should be create depending on outcome of the execution for example: save video only if scripts ends in status Success.

In documentation tab new record with type of Video will be created. After that select it and click Download documentation. MP4 file will be downloaded which you can open and watch execution of the script.

5.10.1 Skip documentation for linked scripts

In version 4.3 we added option to Skip documentation for linked scripts. This feature causes that when script is executed with documentation mode enabled, the document will not contain information about linked scripts steps, there will be indicator that linked script was executed. This feature was created so documentation will not be as large when we linked multiple of scripts inside the main script.

To enable this option open script -> go to general tab of script -> expand documentation section -> checkmark the box next to “Skip linked scripts when generating documentation”



NOTE: This option when enabled will causes ALL of linked scripts related to that main script to NOT have documentation about it

6. How to record a script

To create and record new **Script** go to the **Script module** and click **Add** button.

The form will be opened to define the basic parameters for recording a new **Script**.



Ø Project – select specific project. You can select only those projects which you have access to and newly created script will be displayed only for those users with access to this project.

Ø Name – enter Script name

Ø Manual URL/Environment – select the destination where you will be recording the Script (URL or Environment).

Ø Description – Optionally you can add description to script

Ø Execution timeout – Optionally you can choose maximum amount of time after that Script execution should be stopped

Ø Take screenshots – Select this option if you want system to take screenshot during playback of the script.

Ø Screenshots only on interruption – (only visible after selecting option Take Screenshots) Select this option if you want to limit amount of screenshots taken to only for steps with outcome as Warning or Error

Ø Show overlay hint – System will mark overlay of elements during Script execution

Ø Hide comments – Select this option if u want to hide comments added to the steps

Ø Click on disabled buttons – if the checkbox is selected the script will be able to click on disabled buttons during playback without errors occurring

Ø Disable start step - If the checkbox is selected, it will cause to Start script step to be skipped when this script is executed. Start script step is responsible to reload environment and start from position of the URL typed in System definition tab.

Ø Auto-refresh datasource – if the checkbox is selected data in attached Excel file will be refreshed before script execution (can be used e.g. when spreadsheet contains some formulas)

Ø Custom resolver mode – if the checkbox is selected you can determinate which step detection

Ø Documentation settings – this section allows users to override settings for templates (more about it in

chapter 2.1.1. Global settings

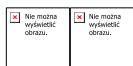
After entering the data, click **Create**.

If you choose **Manual URL**, enter URL address. Recording of the **Scripts** will then begin at the location indicated in the URL.

Note: For Manual URL Virtual users don't work

If you choose **Environment**, indicate the environment on which the **Script** will be recorded.

After selecting environment option, you can decide to whether use default **System definition** for the project or select any other **System definition** attached to selected project.



*Environment is defined in **System definitions (Settings/Application/System definitions)**. How to define new Environment is described in [chapter 2.1.5. System definition](#)

It is possible to define a different environment indicated by the Client. The current EA tool can be adapted to support business processes on the another system (e.g. **ERP, CRM, portal** etc).

Select **Virtual user** on whose credentials the **Script** will be recorded.

Toogle box **Use Virtual User** allows to decide if you want to use a Virtual User in a script. If this toogle is set off then no Virtual user will be used. If this Toogle box is set on then you will be able to determinate which Virtual User you would like to use. You can decide whether to use a default **Virtual user** for the project or select any other **Virtual user** attached to the selected project.



*Virtual user is defined in **Virtual user (Settings/Application/Virtual users)**. How to define new Virtual user is described in [chapter 2.1.7. Virtual users](#).

Click **Create** to save the settings.

After clicking **Create** a form will open to create a new Script. Click **Start/record** to start recording the Script.



If the environment where you will record the steps of the **Script** is indicated by URL address, the window of the defined browser will open and **Recorder** will appear, thanks to which you will be able to start recording the **Script**.

If the environment where you will record the steps of the **Script** is **Dynamics For Operations** or any other system, the new tab will open, so you can record your business process.



Click **Recorder**. The system will start recording.



New buttons will appear on the Recorder:

Ø **Pause** – pause recording

Ø **Stop** – stop / end recording

Inspect - a shortcut button that allows you to register value from the field in the system and then register it as a variable which can be later used on the Script steps.

After using Inspect fill the necessary fields:

Step name - Name of the newly created inspect step

Variable name - Type name for newly created variable



Start recording the business process. When recording, you can:

Ø Change the position of the Recorder

Ø Delete registered steps (1). If any of the steps recorded in the RECORDER window are unnecessary, you can delete them during recording.

Ø Collapse Recorder (2)

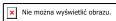
Ø Collapse a list of Recorded steps (3)

Click **Stop** to stop / end recording.

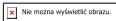
Click **Yes** if you want to save the steps. The system will display information:

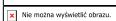


The recorded steps have been saved. Return to the Script form, the system will display recorded steps:

 Nie można wyświetlić obrazu.

During the creation of variable we can in addition create it with condition. If option Create with Condition is checked then couple of field will appear to create condition. Condition is ALWAYS compared to the value seen on screen and cannot be changed from this view, but after creating this step it is possible to edit it from list of steps. About conditionals please refer back to [chapter: 6.2.3.9.2 Conditionals as data validation](#).

 Nie można wyświetlić obrazu.

 Nie można wyświetlić obrazu.

Description of work with the recorded **Script** in *chapter 6.2. How to work with a recorded Script*.

6.1 Scripts form: step by step

From the **Scripts** module, go to the recorded **Script** by double-clicking it. The system will open the form with the steps of the recorded **Script**.

 Nie można wyświetlić obrazu.

The table with the **Script steps** contains the following headers:

1. Select if you want to make changes in the selected step / steps
2. **Line** – step number,
3. **Optional** – selecting this step will cause the step to be executed as an option during Script playback,
4. **Label** – steps list,
5. **Average time** – average duration of the execution of the particular step

Click **Edit** if you want to make changes.

1. Label – displays the label of the current step

- a. **Manual label** – only for some kind of steps. Allows to edit current step label

2. Delay – allows you to set the delay for the next step. Used when the current step takes longer than the standard time interval between stages (e.g., Generating reports, workflow scripts). The delay is expressed in seconds

3. Optional – select if during the script play- back the step is to be performed optionally

4. Disabled – select if during the script play- back the step is to be excluded from the given Script

5. One time – select if the specific step should be executed single time

6. Enable custom timeout – allows users to change step execution timeout

7. Take screenshots – allows you to choose if in background mode for this step screenshot should/not be taken or if this option should inherit from script setting

8. Comment – optionally you can add comment to the step

9. Termination type – optionally you can specify termination of the step:

Ø **Timeout** – if step is not executed after specified time the script will be stopped

Ø **Number of executions** – maximum number of step executions after which the script will be stopped

After we select type of termination we can choose one of two actions:

- o **Error** – Script will be terminated and error will occur
- o **Ignore step** – step will be ignored and the Script will not be interrupted

10. Value – the value assigned to a given step; lets you choose a value defined as Data Source, Sequence or Variable, which will then be used when the script is played back *

a) **Constant** – a constant value is given when the value in a given field can be repeated. Specifying a fixed value in a given field will cause the automatically completed value to be the same each time the Script is played back.

b) **Picker** – Graphic interface for handling variables, values from Excel or sequence numbers. A user not familiar with JavaScript programming is able to intuitively indicate what value is to be entered into the indicated field with a variable value, which is used when the value in the field cannot be constant. Defining variables is described in [chapter 6.2.3.5 Work with an Excel file](#) or in [chapter 6.2.3.4 Assigning sequence numbers to steps](#).

c) **Expression** – a value in the form of a logical expression used to determine the values of variables that must be calculated (created) using the function. It is possible to combine parameters; the formula, based on functions and their relations, returns a single logical value.

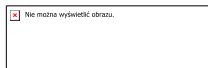
d) **Advanced code** – advanced formulas, for programmers who want to extend the functionality of Executive Automats using their own functions and formulas.

e) **Code template** – allows to use Code templates that are stored in Settings. About Code templates you can read more at chapter [2.1.16.2 Code Templates](#).

11. Advanced fields – section where user can change the Path to the specific element (more about it in *chapter 6.2.3.13. Changing Path to the element*)

- a) **Registration type, Rule ID, Rule version** – for developers
- b) **Path type** – type XPath/Selector, depending on Path typed
- c) **Path** – allow user to change Path to the element which will be used during the step

The changes made should be saved by clicking **Save**:



6.2 How to work with a recorded Script

6.2.1 Modifying the Script

If it is necessary, the recorded **Script** can be modified. It is possible to:

- Ø add steps to a **Script**
- Ø edit and make changes to recorded steps
- Ø delete recorded steps
- Ø assign step with a values of a specific type (Constant/Picker/Expression/Advance Code)
- Ø add existing **Script** as a step to another **Script** and link them together

6.2.2. Adding steps to a Script

If it turns out that it is necessary to add missing steps to the script, go to the **Steps** tab in the selected Script.

The **Rework** function allows you to add missing steps in the finished Script. This functionality can be used both when you forget about a step during recording or when the course of the function/process changes.

Click on the step in Script, after which steps are to be added. Click **Start/Record new steps**.



The login page in system will open. On the right side of the Recorder, a dashed line will appear at the point indicated in advance.

Start **recording additional steps** at the right time, proceeding in the same way as the **Script** recording method described in *chapter 6. How to record a script*.

During the recording of additional steps, all the functionalities are available on the **Recorder**.

6.2.3. Editing and making changes to the recorded steps

In order to make changes in individual Script steps, click the **Edit** button on the selected step, which allows users to edit/modify the previously recorded Script step.

6.2.3.1 Change the order of steps

Reorder functionality – allows users to change the sequence (position) of steps



To change the position of a given step, select the desired step in the Script and click **Reorder**. Move the step to the right place.

After applying this functionality, the steps are automatically renumbered.



After the action has been completed, click **Stop reordering**.

6.2.3.2 Copy/paste and delete steps.

While working with the recorded **Script**, it is possible to copy, paste and delete selected steps. To delete the selected step in the **Script**, click **Remove**.



When the dialog box appears, click **Delete** to Remove selected steps. The step will be deleted and the remaining steps will be renumbered.



To copy the selected step, click **Copy**.



The system will display information that the step has been copied.



Select the step after which the copied step is to be pasted and click **Paste**. The system will display information that the step has been pasted.



All of the steps will be automatically renumbered.



6.2.3.3 Quick Actions

It is possible to use the option of selecting several steps at the same time as **Disable/Optional**.

Quick actions – functionality that allows for the simultaneous application of selected changes (**Enable/Disable step(s)** and **Set/Unset step(s) optional**) in all indicated steps.

Ø **Enable step(s)** – allows the marked steps to assign the enabled functionality if they are disabled.

Ø **Disable step(s)** - allows the marked steps to assign the disabled functionality if they are enabled.

Ø **Unset step(s) optional** – allows the marked steps to unassign the Optional functionality.

Ø **Set step(s) optional** – allows the marked steps to assign the Optional functionality.



Select the steps for which you want to apply **Quick actions**.

Click **Quick actions**.

1. **Disable step(s)** – if you have selected Disabled step(s) for the steps indicated, these steps will be grayed out in the steps list and will not be played

In addition, the **Disable** option will be marked in the details of each step visible on the right .

To cancel off this functionality for the steps indicated, click again **Enable Step(s)**.



2. Set step(s) optional – if you have selected Set step(s) optional for the steps indicated, these steps will be marked in the list of steps as **Optional** and will be executed during the **Script** playback optionally. The functionality is used most often when some of dialog windows appear in the system.



In addition, the field **Optional** will be marked in the details of each step visible on the right.



To cancel off this functionality for the steps indicated, click **Unset step(s) optional**.

6.2.3.4 Assigning sequence numbers to steps

How to create a sequence is described in the [chapter 2.1.3 Sequences](#)

To use in the **Script** a defined in settings the sequence number, it is necessary to decide which step in the Script is to be operated by this sequence.

1. Go to **Scripts** module.
2. Select **Script**.
3. Select the **Script** step in which you want to use the sequence of numbers.
4. Click **Edit**.



Click **Picker**, and then click in the **Value** field



Select the pre-defined numbering sequence from **Sequence**.



Nie można wyświetlić obrazu.

To save the settings, click **Save**.

The sequence of numbers has been assigned to the selected step.

Now, during each subsequent playback of the script, in the field to which the numbering sequence has been inserted, the next number defined in **Settings\Sequences** will be valid.



Nie można wyświetlić obrazu.

6.2.3.5 Work with an Excel file

While working with the recorded **Script**, there is a possibility of using import of data from an Excel file. The column headers in the prepared file will be (after placing the file in the **Data Source**) visible during their hooking to the picker on the **Script** step.

Sample Excel file:



Nie można wyświetlić obrazu.

To import data from an Excel file, place this file in **Data Source**.

Go to the **General** tab on the Script.

Click in the **Data Source** field, then point to the location of the data file and click **Open**



Nie można wyświetlić obrazu.

or drag the Excel file to the **Data Source** field

The system has displayed information that the file has been added to **Data Source**.



Nie można wyświetlić obrazu.

To reupload file, for example after making changes in Excel sheets use button with refresh sign on it.

To delete a loaded file, click the trash bin icon.

Note: if the data from the Excel file has been used in the Script steps, it is not possible to delete the Excel file from Data Source. You will need firstly detach Excel columns from steps.

Removal of the file will only be possible if it is not used in any of the **Script** steps.

It is also possible to generate a data source template, which can be used to customize the data to be included in the script.



Analyze the list of steps (Steps tab) and think about which fields you want to complete the data using the data import function from the file.



Each selected Script step should be mapped to the column header from the Excel file. Choosing the right column will then make EA to take data from the file.

For the system to be able to read data from a file, go to the selected **Script step** and click **Edit**.

Click Picker. Click Value.



A window will appear with variables, from which you can choose the one you want to assign to the step.

In **Data Source**, all column names from the Excel file are now visible.



Select the **DataSource** and the name of this column from the file from which the data should be read by EA for that step.



To save changes, click **Save**.

Repeat the action for all steps for which you want to retrieve data from an Excel file.

To read all data by the system, it is necessary to define the basic parameters for the assertion running the Script. Assertion enables, among other things, repeated playback of the Script, until all data from the file is used

More on defining the assertions in [*chapter 6.2.3.10.3. Conditional step a tool for reading Excel data*](#)

6.2.3.6 Expressions and Advanced Code

Advanced user of Executive Automats can extend functionality of the EA with forming own logical expressions and formulas.

On the step Edition form we can select two types of formulas:



Expression – define an advanced formula in which you can create logical expressions using hints; a value in the form of a logical expression used to determine the values of variables that must be calculated (created) by means of a function. It is possible to combine parameters; the formula, based on functions and their relations, returns a single logical value.

Advanced Code – define an advanced formula based on the combination of several logical values; are advanced formulas, for programmers who want to extend the functionality of Executive Automats using custom functions or formulas.

Users familiar with JavaScript can create their own functions and formulas to extend Script functionality or use one of the available templates already defined in the EA. The EA Functions can be used both on Expressions and Advanced Code.

Examples of Available **EAFunctions**:

EA Functions	Use
EAFunctions.addDays E.g.: EAFunctions.addDays(7) or EAFunctions.addDays(1, new Date("10/05/2019")) or EAFunctions.addDays(30, Script.Varia- blename)	Allows to use date or variable which contain date and add to it particular amount of days. In this function we can use todays date, type down our own date or use date from variable

<p>EAFunctions.changeSign</p> <p>E.g. EAFunctions.changeSign(6)</p> <p>or EAFunctions.changeSign(Script.Varia-blename)</p>	Allows to change sing of a number or variable
<p>EAFunctions.getTextWith- Number</p> <p>E.g. EAFunctions.getTextWith- Number("Dynamics 365 FO")</p>	Allows to reextract part of text which contains also numbers
<p>EAFunctions.getRandomDate</p> <p>E.g. EAFunctions.getRandomDate("01/01/2020", "05/20/2020")</p> <p>Or EAFunctions.getRandomDate(new Date("01.01.2020"), new Date("05.20.2020"))</p>	Allows to generate random date. User can pass two parameters to function getRandomDate (start and end) and function will return date between those. If user won't pass parameters it will check current date and set startDate as current - 20 years and endDate as current + 20 years. User can't pass only one parameter and end date can't be smaller than start date. Date can be string or data object
<p>EAFunctions.getRandomIdentifi- fier</p> <p>E.g. EAFunctions.getRandomIdentifier()</p>	Allows to generate random number
<p>EAFunctions.getRandomLast- Name</p> <p>E.g. EAFunctions.getRandomLastName()</p>	Allows to generate random LastName
<p>EAFunctions.getRandomName</p> <p>E.g. EAFunctions.getRandomName()</p>	Allows to generate random Name

6.2.3.7 How to use Expression

Expressions can be used on the **Selector steps**, **Type Steps** or **Get Value steps** to change variable value. To use **Expression** go to the **Script** and Select desired **step**.



Then click **Edit** step and select **Expression**.

Then there is space to type JavaScript formula. Users familiar with JavaScript(Template strings) can create their own formulas. Also user can use one of already defined **EAFunctions**.

The **Expression** formulas should be typed with this format \${**BodyOfFormula**}. User can click **CTRL+SPACE** to view tooltips:



The example usage of the **Expression** and **EA Function**



After saving **Expression** the step label will get changed.



Then if the **Script** is played back, system will use function or formula form the **Expression** to type value.

6.2.3.8 How to use Advanced Code

Using **Advanced Code** is quite similar to using **Expressions**, but instead of one line commands in **Advanced Code** field we have a lot more space to create our own **JavaScript** code. At **Advanced Code** section advanced users can create their own formulas and functions by suing **Java Script**, but also there is possibility to use predefined **EA Functions**.

To use **Advanced Code** formulas go to the **Script** and select desired step.

The **Edit** step and select **Advanced Code**.



Then there is a space where user can input the Java Script code.

Advanced Code give plenty of opportunities to extend Scripts. Click **CTRL+SPACE** to view the tooltips.



To use any **EA Functions**, before typing body function we have to use function return.

Example of usage of **Advance Code** and **EA Function**. After saving **Advanced Code** the label of the step will got changed.



Then if the **Script** is played back, system will use function or formula from the **Advanced Code** to type value.

6.2.3.9 How to use Code Template

Using Code Templatized is really similar to using Advanced Code but instead of typing code directly at the step we created it from level of settings. To use Code Template to step you have at first select step that you want change then hit Edit button. During Edition of the step select Code Template you choose from which Code Template Group you would like to use Code Template. Some of them we will deliver but users can extend this list:



After you select code template for step it is updated with grey icon on step list that indicates that code template is being used for that step:



After you run the script it will execute code in this code template to type value.

6.2.3.10 New Step

While working with the recorded Script, there is also the possibility of adding steps so that the user can, among other things, assign a defined value for the variable (**Assign value to variable**) or verify whether a given condition has been met (**Conditional**). This tab also allows to record new steps (**Record new steps**) or log in to another system by using **Login** type step, add a **Shortcut** to the script, call JavaScript function without assigning it (**Code**) or to get tokens in our custom Virtual User authentication script (**Auth**). There is also functionality to merge several **Scripts** into larger business process.



To use these functionalities, go to the window with a specific **Script** and click on **New Step**.

6.2.3.10.1. Assign value to variable

The **Assign value to variable** step is performed in order to use the variable in the selected Script step. You can use a variable that has already been defined (how to create a variable described in [chapter 5.3. VARIA-BLES tab](#)) or define a new one.

Click **New Step/Assign value to variable**.

The Create ‘Assign’ Step window will appear.

Complete the required fields:

Ø **Label** – the name of the variable/step that will be visible in the list

Ø **Create after step** – indication of the step after which it is to create a step with variable assignment

Ø Define the source from which the generated variable should be derived:

o **Pick or create variable** – choose or create a variable. Click on the Pick or create variable field. If a variable has been defined before, its name will appear in this field. If you want to create a new variable, enter its name. After entering the name, the message "Variable will be created" will appear

a. **Constant** – select if the assigned variable is to have a constant value .



b. **Picker** – select if the assigned variable is to come from the pre-defined sequence of numbers, Variables or Data Source



c. **Expression** – define an advanced formula in which you can create logical expressions using hints; a value in the form of a logical expression used to determine the values of variables that must be calculated (created) by means of a function. It is possible to combine parameters; the formula, based on functions and their relations, returns a single logical value;

d. **Advance Code** – define an advanced formula based on the combination of several logical values; are advanced formulas, for programmers who want to extend the functionality of Executive Automats using custom functions or formulas,

e. **Code Template** – define Code Template that will executed code to generate a value.

To save your settings, click **OK**.

In the **Script**, a new step appeared in the list of steps while the steps were automatically renumbered.



Go to editing the step for which you want to use the previously created variable and click **Picker**. In the Value field, indicate the defined variable. Save your changes by clicking **Save**.



When playing the script in the step for which the variable was assigned, the system will retrieve values from the defined variable. If there is such a need, the created variable can be assigned to several steps.

6.2.3.10.2 Conditional step as a data validation

Conditional step allows checking the correctness of data placed in the selected field. The purpose is to define a condition that should be met during **Script** playback and ensure the correctness of the business function.

Checking is used in situations where we want to verify:

- whether the values from several fields counted correctly
- whether the fields that determine the execution of the function are completed
- whether the values in the fields that determine the execution of the function are correct.

Go to the steps of the previously recorded **Script**. Click **New Step/Conditional**.



On the next form complete the required fields:

- Ø **Label** – the name of the step that will be visible in the list
- Ø **Create after step** – an indication after which step the assertion will be added
- Ø **(IF) Condition type** – assertion type
 - o **Check value:** specifying whether the assertion is to verify the logical expression
- Ø **Condition** – checking condition
 - o **Select type** – allows to choose the source value whose validity will be verified. This value can be taken from: Constant, From Step Value, From Variable. Values can be text, number or date.
 - Note: To work properly, when option From Step Value is selected, the value has to be visible for the script during condition check and it should be used only with type value steps. For other steps use inspect and then select option From Variable

Depending on the type of value chosen, it should be defined, that is, fill in the field below, indicating the value for Constant, selecting the step (Select step) or variable (Select variable) or global variable or global Constant



Select operator – allows you to define a condition that should be met for the verification to be successful. The following operators are available:



Select type – allows to choose the value that will be used to define the correctness of the assertion. This value can be a constant (from Step Value) or from a variable (from Variable) or global variable or global Constant.

With the conditional step we can create more than one condition which should be met with the conditional step. To add more conditions click + icon.



Then we can select parameter **And/Or**:

Ø And – We can add another condition to the previous created condition and the system will check if all conditions linked by And parameter are met.

Ø Or – This parameter allows user to create opposite conditions. System will check if at least one of conditions are met.



Note: There is no limitation to the amount of the conditions within conditional step.

Ø (THEN) Select action – if the condition has been met, the defined action will be performed

Available options for the fulfilled condition :

- o **Continue** – continue
- o **Log and continue** – display a message and continue
- o **Warning** – warning message shows

- o **Stop and show an error message** – stops and shows an error
- o **Take a next row from the data source and jump to a step** – available for Excel; take the next value from the Excel file and go to step execution
- o **Jump to a step** – go to step
- o **Repeat** – repeat the step



o **Select step** – option only available for Jump to the step or Excel. After fulfilling the THEN condition the system will go to the step indicated in this field.

Ø **(ELSE) Select action** – if the condition is not met, an alternative action will be performed



If the correctness of the value in the defined field is critical for the execution of the script, then the execution of the Script should be stopped.

Let's try to follow this functionality on the example of a specific business function.

Let us assume that an order for the product from the customer is entered to Dynamics For Operation (Purchase order). We can define the assertion so that the EA checks whether, for example, the quantity of the ordered product is more than 5pcs.

- Record the **Script** with the purchase order steps (Sales order).
- Go to Script steps.
- Click **New Steps/Conditional.**
- Complete the required fields:

For our business example, a properly defined conditional step should take the following values:



Click Ok

In the list of steps, a new condition step has just appeared. The steps were automatically renumbered.



Nie można wyświetlić obrazu.

During subsequent **Script** plays, the system will run the assertion and check the correctness of the entered data. If the data does not meet the conditions defined in the check (assertion), the system will return an error.

6.2.3.10.3 Conditional step as a tool for reading Excel data

The conditional step can be used to supply the system with data from an **Excel** file

After mapping the Excel file with the selected **Script** steps (described in the section [6.2.3.5 Work with an Excel file](#)), it is necessary to set the assertion, which will allow you to define how the system should behave in case we have multiple rows in Excel file.

On the Script that is to handle the data import from the file, click **New Step/Conditional**.



The form **Create "Conditional" Step** will open.

Complete required fields and select **Condition type** as **Data source**:

- Ø **Label** – the name of the step that will be visible in the list
- Ø **Create after step** – an indication of which step the assertion step will be added to
- Ø **(IF) Conditional type** – The type of assertion
 - o **Data source**: when the assertion is to be based on an external data source, e.g. collecting successive lines from Excel according to the condition
- Ø **Condition** – checking condition for Data Source option we got condition the Data Source has a next row
- Ø **(THEN) Select action** – if the condition has been met, the defined action will be performed
- Ø **Select step** – an option only available for Excel. After fulfilling the THEN condition the system will go to the step indicated in this field.
- Ø **(ELSE) Select action** – if the condition is not met, an alternative action will be performed



Click **Ok**.

In the Script steps list, a new step has been saved (in the place defined during the creation of the conditional step).

Start the **Script** to start reading data from the file.

The script will be processed as long as it has not exhausted all the data from the attached data source (in the Excel file)



6.2.3.10.4 Login step

Login is special type of the step which allows user to log to another Dynamics 365 instance, other system or log in to the system by using different **Virtual User**.

To create **Login** step click **New step**, then select **Login**.



On the next form fill necessary fields:

- Ø **Label** – the name of the step that will be visible in the list
- Ø **Delay** – specify if the step should be executed with delay
- Ø **Create after step** – an indication of which step the login step will be added to
- Ø **Environment** – Select environment which should be opened within Login step. You can use default or any environment from Project in which Script exists.
- Ø **Virtual User** – Select Virtual user which should be used with the Login step. You can select default or other Virtual User from the project within which Script exists

After all required fields are completed click **Ok** button.



Then **Login** step is displayed on the step list.



6.2.3.10.5 Shortcut step

Shortcut step allows to add a shortcut to a script. To create **Shortcut** step click **New step**, then select **Shortcut**.



On the next form fill necessary fields:

- Ø **Label** – the name of the step that will be visible in the list
- Ø **Delay** – specify if the step should be executed with delay
- Ø **Create after step** – an indication of which step shortcut step will be added to
- Ø **Shortcut** – select shortcut that should be used



After all required fields are completed click **Ok** button.

Then **Shortcut** step is displayed on the step list.



6.2.3.10.6 Script step, linking scripts

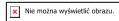
This functionality allows users to combine **Scripts** into larger scenarios. You can choose already existing **Script** in the system and then add it as a step into the another **Script** and link them together. This functionality is used to support complex business processes.



To add already existing **Script** as a step to another **Script** click button **New step** in the step tab and then choose **Script**.

Then list with existing **Scripts** will open.

Select the **Script** which u want to add as a step in the currently edited **Script**, then click button **Next**.

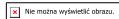
 Nie można wyświetlić obrazu.

Then fill all required fields:

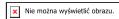
Ø **Label** – Type label for linked step

Ø **Link name** – Enter the name of **Script linking**

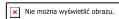
Then click **Ok** button at the bottom of the form.

 Nie można wyświetlić obrazu.

After adding new **Script step** it will be displayed at the step list and all of existing steps will be renumbered.

 Nie można wyświetlić obrazu.

This feature allows to link multiple **Scripts** together and merge them into complex scenarios to meet requirements of any business process.

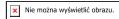
 Nie można wyświetlić obrazu.

To support creating complete business processes, we can pass value between linked scripts. To do that, we are using variables. In simple words we can match variables between linked scripts.

To **match** variables between scripts, we have to create variables on both sides.

For example in the linked script we are creating variable **ProductNumber** with inspect feature, then we want to pass this variable to the main script. To do this in main script we have to manually create variable which will be linked to variable **ProductNumber** from the linked script. The example manually created variable should be easy to recognize. Then we can match these two.

NOTE: Disable linked start step is a feature that will cause to start script of that linked script to NOT be executed when it is played as a part of main script. If this script would be run separately NOT as a linked script the start script option will be executed.

 Nie można wyświetlić obrazu.

After we have variables in both script go to steps and the select desired **Link step** and click **Edit**. Then in Override variables section click **Add**.

Then select variable from linked script.

Then to match it with the variable manually created in main script, select picker and then **Script** and proper variable name.



Then click **Save**.

Now both variables are linked and the value from the linked script is transferred to the main script and this variable can be used within steps in main script.

6.2.3.10.7 Auth step, Virtual user Custom authentication script

In Executive automats, except standard virtual user authentication script, allows also to record own custom authentication script (E.g. when user needs to login to some additional portal).

- Record new script with login process
- In **Variables** tab create two variables: for login and password (more about creating variables in *chapter*)

5.3. VARIABLES tab



- Assign both variables to steps typing login and password (just like in *chapter 6.2.3.9.1. Assign value to variable*)



- Click **New step** and select **Auth**



- **Create Auth step** form will appear. Now type name for new step and add it after the login process



- Click **Ok** at the bottom of the form. New **Auth** step will be added to the script



Our Custom script is ready, now we can go **Settings -> Virtual user** to create virtual user using the script

- In **Virtual user** tab click **Add**



- In **Create virtual user** form type **Name** for new virtual user, check **Custom script** checkbox and type ID of the script that should be used during authentication process.
 - o If you also want to use re-authentication option, check **Remember sensitive data** checkbox
- In **Variables** section add variables for login and password
 - o If you want to use re-authentication option, type value for login and password and mark them as **Is sensitive**
- When everything is ready click **Create**



The Virtual user is ready and can be authenticated.

6.2.3.10.8 Code step

Code step allows to call a JavaScript Expression/Advanced code/Code template function without assigning it to the variable.



On the next form fill necessary fields:

- Ø **Label** – the name of the step that will be visible in the list
- Ø **Delay** – specify if the step should be executed with delay
- Ø **Create after step** – an indication of which step shortcut step will be added to
 - Ø **Expression/Advanced code** – type of the code that will be executed

In field below type the code



After all required fields are completed click **Ok** button.

Then **Code** step is displayed on the step list.



6.2.3.11 Changing the Status of the script

In the **Executive Automats** we can define multiple **Statuses of the Scripts** which refers to the particular design stages of the test case. Every newly created **Script** by default has the **Draft status**, which can be changed by user according to the next stages pf the Script creation.

Available **Statuses of the Scripts**:

- Ø **Draft** – Newly created Script which is still in the early phase of the test design
- Ø **In progress** – Script is already in the finale phase of test design or Script still need some fixes.
- Ø **Ready for review** – Test case is recorded and modified and then can be Reviewed by other user
- Ø **Ready to test** – Script got Reviewed and is ready to use in the first test and then published
- Ø **Published** – The script that has been verified, configured in accordance with the assumptions of the process and is ready to be run in the Scheduler, can be published.

Note: The published Scripts cannot be edited.

We can change statuses multiple times, so if we decide to move **Script** to previous status, then it also can be done.

To change Status of the selected **Script**, open selected script and click **Change Status**.

Changing status is an operation similar with script versioning. By changing status of the **Scripts**, the user can track **Script changes**.



The form **Change Status** will open. Select one of the possible statuses.



Optionally you can add comment and then click **Ok**.

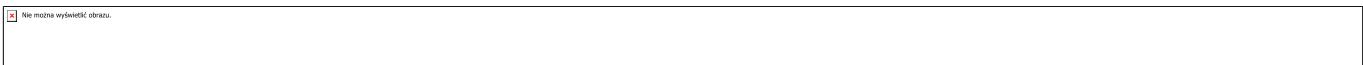
 Nie można wyświetlić obrazu.

After saving changes, the **Script** status will change and will be visible both in **Script** form and on the **Script** list.

 Nie można wyświetlić obrazu.

 Nie można wyświetlić obrazu.

All of the changes made to **Script Statuses** can be viewed on Script form under the **Changelog** tab.

 Nie można wyświetlić obrazu.

Statuses can be used not only to mark **Script** creation phases, but also to select which **Scripts** should be executed by **Schedulers Execution** on project. **Scheduler execution** functionality is described in *chapter 4.3.1. Scheduled execution of the project by Script status*.

6.2.3.12. Cloning the Script

This feature allows users to create duplicate scripts without the need for creating new **Script**. To clone **Script** open up existing **Script** and then use **Clone** button.

 Nie można wyświetlić obrazu.

The **Clone Script** form opens. Enter the name of the **Script** which will be created and optionally add a description. After everything is correct click button **Ok**.

 Nie można wyświetlić obrazu.

The newly created **Script** is displayed at script list.

 Nie można wyświetlić obrazu.

6.2.3.13 Changing Path to the element

Steps edition allows user to change Path to recorded element manually by typing XPath or Selector from the application code.

Select a step for which the path should be changed and Click **Edit**. In **Advanced fields** section, in **Path** field type Path to the element, which should be used during Script Playback, then in **Path type** field type **XPATH/SELECTOR**, depending on which type has been used. After all changes click **Save**.

Good practice is to change **Path** and **Path type** in all fields. It can be also a good workaround for some elements that the Recorder doesn't support.

It is also possible to add new field and then type Path to it, by clicking **Add**.



6.3 How to work with scripts using Advanced Recorder

6.3.1. Recording a Script using Advanced Recorder

In order to record a script in Advanced recorder Click start > more options



A new window opens where you can select recording mode. Mark Advanced and choose where the recorder should open.



Recorder open mode:

Ø **Current window** - recorder will open in the same tab replacing the EA application,

Ø **New window** - recorder will open in a new tab.

Recorder position:

Ø **Left side** - the recorder will be located on the left side on the screen,

Ø **Right side** - the recorder will be located on the right side on the screen. After selecting **Current window** and **Left side** position, the Recorder appears.



Using Advanced Recorder you can simply switch from Recorder to Player, so make sure that recorder is selected, and click Start recording.



Now, the application will start recording user's actions.

After clicking start recording button the recorder's interface changes and the user can perform new actions.



Ø **Synchronize** – Advanced Recorder synchronizes and save its steps continuously but the user can always perform this action manually.

Ø **Pause** – the recording is paused.

Ø **Inspect** - a shortcut button that allows you to register value from the field in the system and then register it as a variable which can be later used on the Script steps.

6.3.2. Working with a recorded script using Advanced Recorder

Using Advanced Recorder the user can now perform many actions on the recorded steps with no need of coming back to EA application.

6.3.2.1 Add a new step

To add a new step simply click 'New step' button and choose action. More on that can be read in chapter

[6.2.3.10. New step.](#)



6.3.2.2 Remove selected steps

To remove steps the user has to mark them and click 'Remove' button.



6.3.2.3 Edit mode

In the edit mode the user can:

- Comment
- Change value of the step
- Change xpath

These actions are described in chapter [6.1 Scripts form:step by step](#) under click Edit information in points 8, 10, 11.

To enter the edit mode the user has to expand the step by clicking button and next clicking edit mode button.



After all changes are done the user has to save them by clicking Save button.

6.3.2.4 Data tab

In the data tab the user can preview:

Ø Local variables – a list of local variables existing in the script, the user can create new local variables in this view,



Ø Live variables – a preview of current values of local variables, global variables and datasource entries,



Ø Datasource – a preview of the data source that is assigned to the script. The data source can be up-loaded, reloaded and deleted in this view,



Ø Sequences – a list of all sequences in the EA application,



Ø Global – a list of global variables and global constants assigned to the script, the user can also create and modify them in this view.



To exit this data preview click Close button in the right upper corner.

6.3.2.5 Quick actions

In the Quick actions tab the user can:

Ø Enable step(s) – previously disabled steps will be enabled,

- Ø **Disable step(s)** – steps will be disabled and will not be played,
- Ø **Unset step(s) optional** – steps previously set as optional will no longer be optional,
- Ø **Set step(s) optional** – steps will be executed optionally in the script,
- Ø **Set recording position** – selects new position for the recorder,
- Ø **Reorder** – allows to reorder steps using drag and drop gesture,
- Ø **Mark step element** – during the playback an overlay hint will be shown.



6.3.2.6 Step details

After expanding a recorded step using + button the user can go further and expand step details. In this preview the user can see details of the step and modify them.



Advanced Recorder can be set as default for a user. More on that in chapter [2.2.2. Execution Settings](#).

6.4 Playback of a recorded Script

6.4.1 Standard playback of the Script using basic player

In order to verify the correctness of the operation of the recorded **Script**, it should be **Played**.

To **Play** the recorded script, go to the **Scripts module**. Select the **Script** to be played and click **Start/Play**



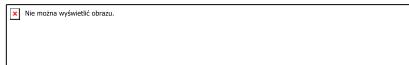
The **Player** will be displayed before the script starts playing. While playing the **Script**:



1. It is possible to use Pause

2. All recorded steps are visible along with an indication of whether they have been carried out correctly
3. It is possible to collapse the list of displayed steps
4. Collapse a Player window
5. Repeat the Script playback

To start playing the recorded Script, click **Play** on the **Player** window.



The system will start playback of the **Script**, indicating progress in its execution:

Green checkmark - the step was done correctly

Red square with X - the step did not work correctly and therefore it is not possible to execute the next steps of the **Script**

Yellow triangle with ! -the step has taken place, but the system has generated a warning about the incompatibility of the data with business requirements of Dynamics 365.

6.4.2 Standard playback of the Script using Advanced Player

Using Advanced Recorder the user no longer has to open new session with the player. Now the switching is easy and fast. To open Advanced Player simply click Player button to switch from Advanced Recorder.

The icons are very similar to the ones in Basic Player. In order to play the script click the play button. If you want to replay click the refresh button and next the play button. Using Advanced Player the user can now start the playback from a marked step.



Both Advanced Player and Advanced Recorder can be collapsed using ‘Hide steps pane’ button.

6.4.3 Non-standard playback of the Script

Before the **Script** play, the user can define how the Script should be played. **Non-standard playback** options allow you to **automatically start** playback, **slow down** the playback of steps and perform **background playback**.

To select one of the non-standard Script playback methods, go to the selected **Script** and click **Start\...**



A window will open to select the play mode :

Play mode – selection of the play mode



Ø Foreground – the script will start playing; the user will have the option of tracking the execution step by step

- **Auto start** – Script playback will start automatically without clicking Play on the Recorder
- **Slow mode** – Script playback at a slower pace, taking into account the larger time intervals between each step
- **Player Type** – Changes between Basic Player and Advanced Player
- **Player open mode** – The script will be executed in this window or a new one.
- **Player position** – The panel will be on the left or right.

Ø Background – Script playback in the background. The system will display information about the action being started :



The system will not display the course of the **Script** being played. On the **History tab**, however, the implementation status will be visible :



Select the play mode and click **Run**.

The system will recreate the Script in a mode consistent with the user's choice .

6.5 Import/Export Scripts

Executive Automats allows user to **Export** selected scripts and **Import** them to other EA instance.

In order to **Export** scripts, go to Scripts tab, select wanted scripts and click **Export to file**.



The Script.ea scripts package will be downloaded

In order to **Import** scripts, go to Scripts tab and click **Import**.



Script import form will appear.

Select Project, to which scripts should be imported and add previously exported Script.ea scripts package by pointing to the location of the file and click **Open** or drag the Script.ea file to the field.



If you want to import another scripts package, click trash bin button add other package. When everything is ready, click **Save**.

After that, all scripts from the package will be imported and each script will appear on the Scripts tab. Imported scripts contain:

- Steps
- Variables
- Number Sequences used
- Attached Excel file
- Linked scripts

Imported scripts doesn't contain:

- History
- Changelog
- Links to created work items
- Generated documentation



6.6 Capturing files

Executive allows users to create processes which will download/upload files. The only things user has to do is:

Ø Record file download and play the script

- o After finished Playback, the downloaded file will be displayed in **History** tab after clicking **Captured files**



Ø Record file upload

- o Uploading file (e.g. uploading attachment in Dynamics) will be recorded as **Upload a file** and the file will be stored on the server.



To change the file that should be uploaded, select the step and click **Edit**.

Click on the **Upload file** icon to select new file and click **Save**.



To download the file from the step, click on its name.

7. SCHEDULER module

Scheduler allows you to define tasks/plays of selected scripts with an indication of, among others, the start time, end date and frequency of repetitions. The functionality can be used to ensure proper functioning of this Dynamics 365 For Finance and Operations system element for which the given **Script** applies. This mode is used for **Regression tests** and **Performance tests**.

For example, let's assume that we are creating a modification in the system and we want to check whether the changes made do not disturb the proper functioning of the system. To do this, we create a **Scheduler** that will verify that the system is still working properly. Tasks defined in **Scheduler** are performed in the background without affecting the user's work. They can be one-time or recurring.

Below you can see the **Scheduler** form and the description of the headers it contains.



1. **Job Id** – number of the defined task
2. **Name** – name of the defined task
3. **Description** – description of the defined task
4. **First occurrence** – the date and time of the first task execution
5. **Ending date** – date and time of completing the task
6. **Trigger type** – select trigger for scheduler:
 - a. **Time** – select date and time which should trigger Scheduler
 - b. **Storage** – you can select Azure Storage as a trigger, when MS Excel file shows up in cloud storage it will trigger Scheduler
7. **Recurrence** – frequency with which the task is to be performed
8. **Created At** – date and time of creation
9. **Status** – status of the task

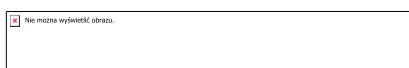
7.1 How to create a Scheduler

Go to module **Scheduler**.



Click **Add**, to create a new record.

The **Create a scheduled job** window will open.



Complete the required fields:

1. Name - name for the defined job
2. Mode - Basic or Gantt
3. Run mode - specify the mode of the playback

Ø **Sequential** – Scripts are run one by one, next Script will start after the previous one will end

Ø **Parallel** – parallel execution of scripts means that many scripts from the project will be run simultaneously. The number of scripts that can be run at the same time depends on the server resources.

4. Description - description of defined job

5. Start time – select date and time when the job should start

6. Recurring – if the task is to be done cyclically, move the slider to the right and fill in the fields below:

7. Max parallel sessions (only in Scheduler Basic Parallel) – allows you to limit amount of scripts at once in multiple sessions

8. Repeat every – specify the frequency with which tasks are to be performed (for example: every 3 hours, every 2 days, etc.)

9. Notify by email – allows to notify user by email about finished job

10. Result in DevOps – results about finished job is automatically exported to DevOps instance. Results will be transferred only to scripts that are already exported to DevOps. Also you can set Configuration in Scheduler.

11. Configuration – determinates which configuration should be used for this scheduler (appears only when Result in DevOps option is checked),

12. Override – function to set the given conditions to the scheduler.

Ø **Environment** – allows to change the Environment to the scheduler for which the scripts will run.

Ø **Virtual user** – allows to change the Virtual user to the scheduler for which the scripts will run.

13. Ends – specify the end date of the job

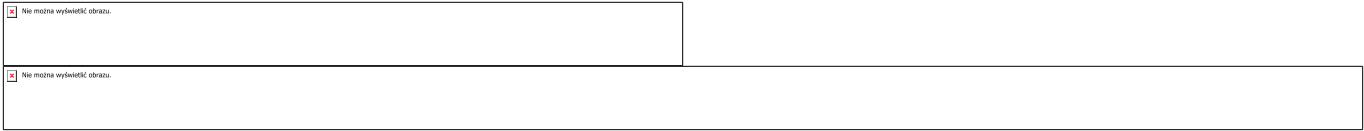
Ø **Never** – if the tasks are to be carried out continuously, without the final date.



Ø **After occurrences** – if the tasks are to be performed a specified number of times in the indicated time interval

Ø **On date** – if we want to set a limit date for the job

Click **Add** to save settings. The details of the created work will be complemented.



Status of the newly created **Scheduler job** is now **Inactive**. Go to the details of the created task by double-clicking it.



A form with three tabs opens : **General, Flow and History:**

1. **General** – settings that were saved when the task was created. After clicking **Edit** you can change the

settings regarding the frequency and ending date of the task (after changes have been made click **Save**).

2. **Flow** – after clicking Edit a system will display a list of all recorded Scripts
3. **History** – a list of all jobs executed so far along with it statuses

Go to **Flow** tab. Click **Add scripts**.



Now below a list of all recorded scripts will be opened.

Select the **script/scripts** that you want to add to the task being performed.



Then click **Save**. The scripts have been added to the scheduled task/run.



If you have selected more than one script, you can change their order by clicking **Edit -> Reorder**. Drag and drop **Scripts** to change their location.

Note: Order of the task matters if the run mode was selected as **Sequential**.



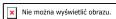
After reorder is finished use **Stop reordering** button.



You can also delete the previously selected script/scripts from the **Scheduler flow**. For this purpose :

Ø Select a particular **Scripts**,

Ø Then click Remove.



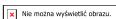
Next you can choose the number of playbacks of the **Scripts** during single batch run. Click in the field **Repeat count** and type the number of playbacks.

*Example: If you create **Scheduler job** with there occurring parameter as 3 and you select just one **Script** in **Scheduler flow** and then select **Repeat count** for this **Script** as 2 . It means that in single run of the **Scheduler job**, **Script** will be executed 2 times and there will be 3 **Scheduler runs** so the total number of **Script executions** will equal 6.*



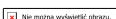
You can also override which Virtual User or Environment or both should be used for that specific script. This allows user to quickly check if script run on different environment or Virtual User will still work correctly. To be able to change those click Edit and then checkmark the box **Override** for script. That allows Overridden Virtual User and Environment fields to be editable. Then user is able to select desired records.

NOTE: If we override Environment or Virtual User on both General tab and in Flow Tab, the environment and Virtual user will be taken from Flow tab.

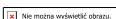


Click **Save**. The Scheduler job flow form opens, where you can see the task just created. The task has a status

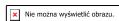
INACTIVE.



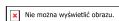
By using the **Activate** button in the upper right corner you can activate the task execution.



The status of this task will change to **Active**. You can change the status of the task at any time by clicking **Deactivate**.

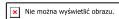
 Nie można wyświetlić obrazu.

When the **Scheduler job** is over go to the **History** tab to see the results, also the status of the **Scheduler job** will change to **Completed**.

 Nie można wyświetlić obrazu.

If you selected **Notify by email** option, you will also get email notification with the results fo the **Scheduler job**. Go to **History** tab. All tasks performed so far are visible here.

Similar as it is explained in *chapter 4.2.5. HISTORY tab*, Scheduler History also can be aggregated for each run.

 Nie można wyświetlić obrazu.

Each specific log can be opened and each script execution details will be displayed

1. **Project** – project name
 2. **Session Id** – session number
 3. **Script Id** – Id of the particular Script
 4. **Name** – Script name
 5. **URL** – Environment's URL address
 6. **Environment** – name of defined system definition used during execution start
 7. **Virtual user** – name of the Virtual user used during execution start
 8. **Start time** – the date and time of the start of the session
 9. **End time** – the date and time of the end of the session
10. **Duration** – the time in which the task in a given session was performed
11. **State** – the state of the task performed
12. **Status** – the status of the task performed in a given session*
- *particular statuses are described in the *chapter 3. History module*
13. **Username** – name of the user who ran the Scheduler
 14. **Last issues** – listed last error and warning that happened during execution of specific script,
 15. **Job id** – Id of the job

If you want to view the performance statistics for opened log, click the **Statistics**.

 Nie można wyświetlić obrazu.

The system will generate information in the form of a progression chart containing data on the number of executions and statuses of tasks.

 Nie można wyświetlić obrazu.

7.2 How to create Gantt scheduler

Gantt mode allows users to plan how scripts in the Scheduler should be run. Go to module Scheduler.

 Nie można wyświetlić obrazu.

Click **Add**, to create a new record.

The Create a scheduled job window will open.



Type **Name** for new scheduler and select **Gantt mode**. Additional options when creating a Gantt Scheduler are to set Description, Notify by e-mail, Result in DevOps using the Configuration for DevOps setting and set Environment, Virtual User in which the scripts added to the scheduler will be run. Then click **Add** at the bottom of the form.

Click **Add** to save settings.

 Nie można wyświetlić obrazu.

Status of the newly created Scheduler job is now **Inactive**. Go to the details of the created task by double-clicking it. A form with three tabs opens: General, Gantt and History.



1. **General** – settings that were saved when the task was created. After clicking Edit you can change the settings regarding the mode of Scheduler and the basic information.
2. **Gantt** – here we can add and set up our scripts to execute in a desired way.
3. **History** – a list of all jobs executed so far along with its statuses (more in chapter [4.2.5 History tab](#)).

To create a test scenario, go to the **Gantt** tab.



Nie można wyświetlić obrazu.

An empty diagram will appear where users can manipulate to adjust scripts to their needs. To add new scripts to the Gantt scheduler click **Add**. Now the list of all recorded scripts will be opened. Select the scripts to add to the test scenario and click **Next**.



Nie można wyświetlić obrazu.

On the next form users can change **Label** for each script, select **Run mode** and **Repeat count** (if **Run mode** has been selected) and add selected scripts as **Group**. When everything is ready click **Ok**.



Nie można wyświetlić obrazu.

All scripts will be added to **Gantt** tab. Click **Edit** to manage scheduler run.



Nie można wyświetlić obrazu.

Every script with its own Run mode and Repeat count is represented by rectangle with different color. Users can change the start time by dragging the script to left or right. Information of the Start time can be found on the left diagram part. It is also possible set the time frame by changing the **Zoom** option above the diagram. To perform changes in the Scheduler click **Edit**. Now users are able to:



Nie można wyświetlić obrazu.

1. **Edit** each particular script **Name**, **Description**, **Start date**, **Run mode**, **Repeat count** by selecting it (panel on the right side will appear)
2. **Create a group** – When more than one script is selected (to do it hold CTRL and select script) users can add them to the same group – then instead of setting each script start time, run mode and repeat count users set it all for the group. Name the group in the field **Group name** and click **Ok**. The group with marked scripts will be created.
3. **Move to group** – If there is existing group, scripts can be added to it using this option
4. **Remove** – Removes selected scripts from Scheduler
5. **Functions:**
 - a. **Change gantt start time** – select when Scheduler should start
 - b. **Change scripts start time** – select at least two scripts and change their start time (Changing start time for single script can be done in panel on the right side when script is selected)

To make tests executions more advanced, users can set up links, which can define the logic of test scenarios. To do this, using drag and drop connect white circle at the end of 1st script to the

beginning of the 2nd and click **Save**. As it shown on an example below, group of scripts called „GroupTwo” will start only when the FreetextInvoice run is done, no matter if it takes 3 minutes or an hour.



To remove a link, in edition mode mark desired link click **Remove links** and click **Save**.

After everything is ready, click **Activate** button. Now Gantt Scheduler is ready to be executed.



When the Gantt is active, it can't be edited. However, if users want to perform any changes click **Deactivate** to make Scheduler editable.



After the execution is done, Scheduler status will change to **Completed**.



8. Documentation module

Documentation mode tree gives users quick access to documents generated during scripts **background + documentation** mode playback (how to perform it is explained in [chapter 5.10. DOCUMENTATION tab](#)).



On the left side all created **Projects** are displayed. Selecting any of them, in **Documentations** tab users can **Remove**, **Edit**, **Download documentation** (the same as in [chapter 5.10. DOCUMENTATION tab](#)) or **Play** documented scripts.



9. Reports module

In reports module you can see reports created thanks to connection with Power BI application. To set up such connection please refer back to chapter [2.1.1 Global settings](#). Content of this page

can be different depending on the Power BI app connected and if user has access to this app.
Example of how it can look can be see below:

Nie można wyświetlić obrazu.

For information about how to set up connection to Power BI to utilize data from EA please refer to document: [Power BI and EA - integration - Executive Automats - Confluence \(atlassian.net\)](#)

Nie można wyświetlić obrazu.

Dynamics Automated

T r a d e m a r k o f X P L U S

ul. Puławska 435A 02-801 Warsaw Poland	Austin Friars 27 London EC2N 2QP United Kingdom
---	--

info@dynamicsautomated.com | www.dynamicsautomated.com | [@DynamicsAutomat](#)
www.executiveautomats.com

tel.: +48 660 631 061

Quick Guide - How to connect EA with Azure DevOps

In this document we will sum up what is required and how to create connection between Executive Automats and Azure DevOps. This document focus solely on creating the connection. Benefits from such connection are summed up in this document: XXX

At first to connection to work properly we need to be sure that our Azure Dev Ops has in section Organization Settings → Billings, license as below:

The screenshot shows the 'Organization Settings' page for 'contoso-xpl'. The left sidebar lists 'General' settings like Overview, Projects, Users, and Billing. The 'Billing' section is selected and highlighted with a red box. It displays usage limits for various services. The 'Basic users' row is specifically highlighted with a red box.

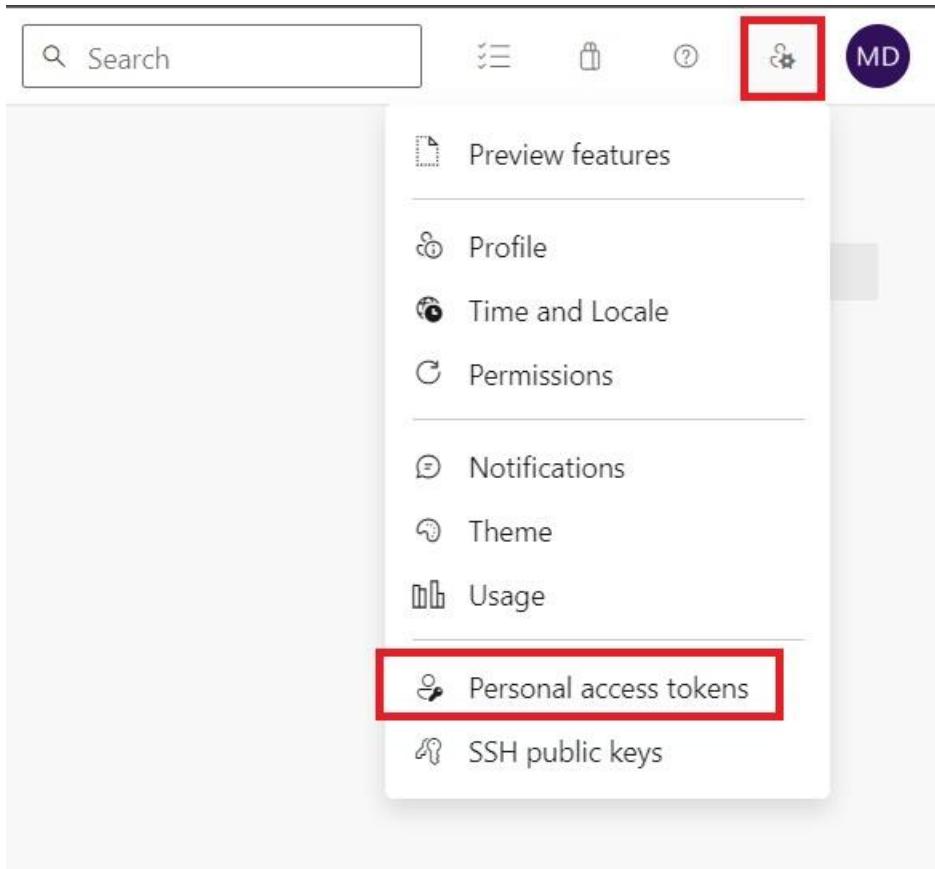
Service	Limit
Pipelines for private projects	Free
MS Hosted CI/CD	1800 minutes
Self-Hosted CI/CD	0
Boards, Repos and Test Plans	Free
Basic users	5
Basic + Test Plans	

This is required at least to connection work properly.

Second thing that we need to take from Azure DevOps is Personal Access Token.

Go to your Azure DevOps instance and log in there with your user.

Go to User settings → Personal Access Token.



There you need to create new Personal access token that is required to connect EA with Azure DevOps.

Recommended is to give it “full access”. After creating token remember to copy it and save it somewhere since the token is not saved anywhere in Azure DevOps and if you forgot to do so, then you would have to create new token.

Now lets jump to Executive Automats → Settings → Issue tracking → System Definition and hit +Add.

Create issue tracking tool definition

* Name:	Name of the record
* Host:	URLofAzureDevops Main Page
* Access token:

Name - name of created issue tracking record,

Host - URL of the Azure DevOps you want to connect. It has to be URL of the main page where you can see all projects.

Access token - Personal Access Token that you generated earlier. Paste it here.

After giving all those details we need to click button “**Fetch projects**” to test the connection.

If everything was set up correctly the fields below will be editable.

The screenshot shows the 'Issue tracking' configuration page. It has two tabs at the top: 'Issue tracking' (which is selected) and 'Test plans'. Below the tabs, there are six dropdown fields with asterisks indicating they are required:

- * Project: Contoso
- * Default iteration: Enter default iteration
- Default area path: Select default area path
- * Default work item type: Select default work item type
- * Default priority: Select default priority
- * Default severity: Select default severity

At the bottom right of the form, there is a button labeled '+ Add custom field'.

Next step is to set up default values for Issue tracking tab and Test Plans. It is required since those details will be needed to properly export data from Executive Automats to Azure DevOps.

NOTE: If your Azure DevOps requires some additional fields to be added when exporting items to Azure DevOps then option Add custom field allows to fill out more details that may be required.

When you filled out all data in every field, click Create.

Last step is to attached connection with Azure DevOps to our user in Executive Automats.

Go to user settings by clicking your **User Name** (in right upper corner) → **Settings**.

In field Issue tracking tool please select record that you just created and click Save to save the changes.

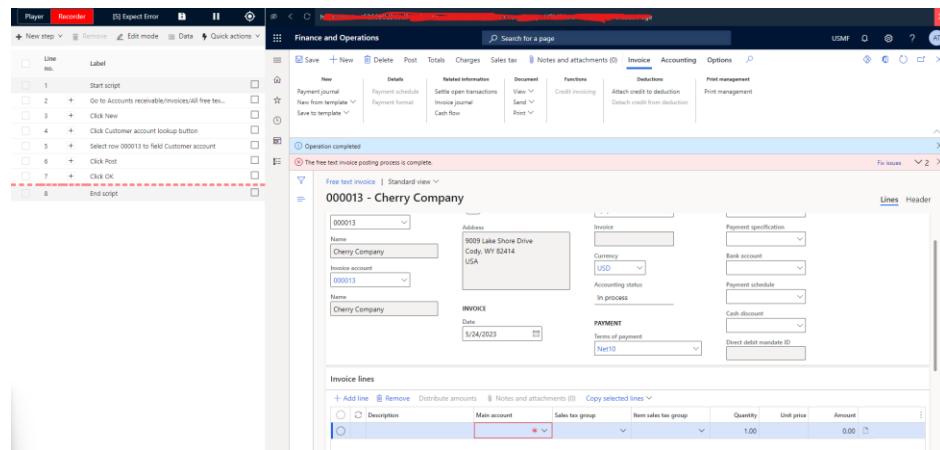
Now the connection between Executive Automats and Azure DevOps is created.

Quick Guide - How to create Negative Test

In this guide, we will focus on how you can create a Negative Test in Executive Automats.

Negative test is a test that we expect to end with error. In this case since we expect the error to happen, the error is an success.

In below screenshot, I have created a script that creates new Free Text Invoice but it does not fill any information in invoice lines section. As you see after clicking OK the error in DFO appears saying: "The free text invoice posting process is complete. Invoice account 000013 There are no lines in free text invoice"



This is the outcome positive for me since the test will confirm that if we do not fill any information in Invoice Lines section we cannot post Free text invoice.

However, if now I play that script the result of the script will be Error:

Two screenshots of the Executive Automats application. The top screenshot shows the 'History' tab for a script named '5 (Expect Error)'. It lists seven steps: Start script, Go to Accounts receivable/invoices/all free text invoices, Click New, Click Customer account lookup button, Select row 000013 to field Customer account, Click Post, and Click OK. All steps are marked as 'SUCCESS'. The bottom screenshot shows the 'Logs' tab for the same script. It displays the same seven steps along with their start and end times, durations, and status. The first six steps are marked as 'SUCCESS', while the final step 'Click OK' is marked as 'ERROR' with a red status indicator. The log also includes a note: 'The free text invoice posting process is complete. Invoice account 000013 There are no lines on the free-text invoice.'

That at first can be confusing since Error status tells us that something is wrong with the script but in this case that is fine.

To resolve this we can utilize **Advanced Conditionals** (you can read more about Advanced Conditionals in our Manual Guide: [Executive Automats User Guide - Advanced Conditionals](#)).

Advanced Conditional can be used in many various situations and Negative Testing is one of the most common one.

We need to create Advanced Conditional that will guard the step “Click OK” since this is the step which encounters the error.

Create advanced conditional

* Enter name: Error is expected

* Guard steps: 7. Click OK X

* Type: Success Interruption

+ Add reaction

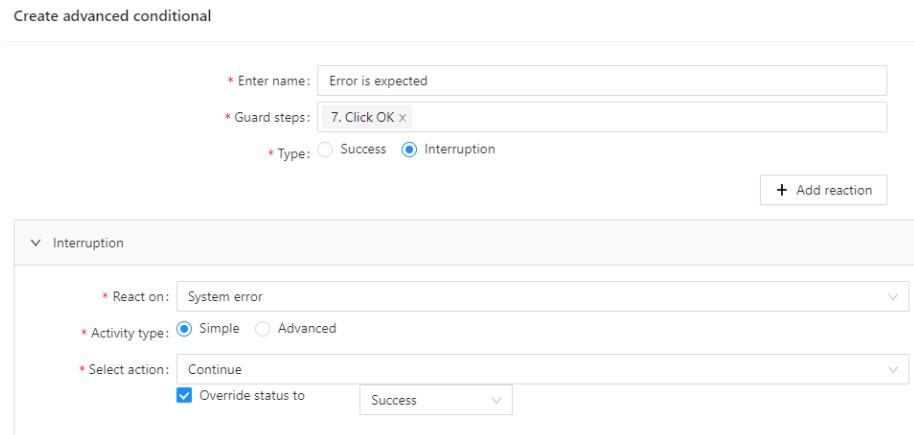
▼ Interruption

* React on: System error

* Activity type: Simple Advanced

* Select action: Continue

Override status to Success



Explanation of above Advanced Conditional that I created:

Enter Name - here you determinate the name of the Advanced Conditional so it is up to you how you name it,

Guard Steps - here you determinate the steps that should be guarded - when the steps is executed after that the advanced conditional action will take place,

Type - Here we need to tell what is expected outcome of the step. Success is for steps that ends up with status Success and Interruption is for steps that ends up with status Warning or Error. So in our case we select Interruption.

React on - Here is the specific action that we need to react on. In our case it is System error since the error comes from the system (DFO) that we are testing.

Activity Type - Here we can decide if we need to react simply or more advanced reaction is needed. In our case we choose Simple,

Select Action - Those are similar actions like in Conditionals steps. I selected Continue since i want to continue with my script but you can change it to meet your needs.

Override Status to - this option if checked will change the outcome of the step to one selected in field next to it. In my case I want to turn this **Error** to **Success**.

NOTE: on the steps list you will notice that guarded step is marked with blue icon. That tells the user that this step is checked against some condition from Advanced Conditional.

If right now I will play that script the outcome will be as below:

The screenshot shows the Executive Automats interface with the History tab selected. The timeline of events is as follows:

Line no.	Label	Start Time	End Time	Duration	Status
1	Start script	24.05.2023, 15:29:01	24.05.2023, 15:29:04	3.03 s	Error
2	Go to Accounts receivable/invoices/All free test invoices	24.05.2023, 15:29:06	24.05.2023, 15:29:13	6.16 s	Success
3	Click new	24.05.2023, 15:29:13	24.05.2023, 15:29:18	4.75 s	Success
4	Click Customer account lookup button	24.05.2023, 15:29:19	24.05.2023, 15:29:23	4.03 s	Success
5	Select no 000013 to field Customer account	24.05.2023, 15:29:24	24.05.2023, 15:29:28	4.07 s	Success
6	Click Post	24.05.2023, 15:29:29	24.05.2023, 15:29:33	3.56 s	Success
7	Click OK	24.05.2023, 15:29:33	24.05.2023, 15:29:41	8.23 s	Success

Bottom status bar: Success

The status of the step has changed to Success and even though the Error happened it did not stopped my script and instead it continued.

As you see thanks to Advanced Conditional I was able to create Negative Test.