Process Design Document (PDD)

Unicorn Name Generator

Document History

Date	Version	Role	Name	Organization (Dept.)	Function	Comments
14.07.2021	1.0	Author	Kacper Groblewski	n/a	*Dev*	Created document v 1.0

Document Approval Flow

Version	Flow	Role	Name	Organization (Dept.)	Signature and Date:
1.0	**Document prepared by**	*Dev*	*Kacper Groblewski*	*n/a*	*Signed electronically*

Table of Contents

1. Introduction

1.1. Purpose of the Document

The Process Definition Document outlines the business process chosen for automation using UiPath Robotic Process Automation (RPA) technology.

The document describes the sequence of steps performed as part of the business process, the conditions and rules of the process prior to automation and how they are envisioned to work after automating it, partly or entirely. This specifications document serves as a base for developers, providing them the details required for applying robotic automation to the selected business process.

1.2. Objectives

Process done in purpose of assigning a new Unicorn Name for children born in Poland in 2018. Robot selects most popular names given for children in 2018, based on an article published by the Ministry of Digital Affairs in February 2019. After retrieving specified amount of names for both genders, robot selects random month for each name, and generates a new Unicorn Name, using the RPASamples.com portal.

The business objectives and benefits expected by the Business Process Owner after automation of the selected business process are:

- *Reduce processing time per item by 80 %*
- *Better Monitoring of the overall activity by using the logs provided by the robots*

1.3. Key Contacts

The specifications document includes concise and complete requirements of the business process and it is built based on the inputs provided by the process **Subject Matter Expert (SME)/ Process Owner.**

The **Process Owner** is expected **to review it and provide signoff for accuracy** and completion of the steps, context, impact and complete set of process exceptions. The names have to be included in the table below.

Role	Name	Contact details (email, phone number)	Notes
Process SME	Kacper Groblewski	groblewski.kacper@gmail.com	Point of contact for questions related to process details & exceptions
Process Reviewer	Kacper Groblewski	groblewski.kacper@gmail.com	Point of contact for questions related to process details & exceptions
Process Owner/ Approver for production	Kacper Groblewski	groblewski.kacper@gmail.com	

1.4. Minimum Prerequisites for Automation

1. Filled in Process Design Document
2. Test Data to support development
3. Dependencies with other projects on the same environment

2. Process Description

2.1. Process Overview

General information about the process selected for RPA prior to automation.

#	Item	Description
1	**Process full name**	UnicornNameGenerator
2	**Process Area**	Recruitment
3	**Department**	Recruitment
4	**Process short description** (operation, activity, outcome)	Retrieving most popular names given for children in 2018, including its frequency represented by an integer. Each name is inserted into Unicorn Name Generator, and the retrieved name is inserted into an Excel file.
5	**Role(s) required for performing the process**	n/a
6	**Process schedule and frequency**	On demand
7	**# of items processes /reference period**	~20/run
8	**Average handling time per item**	2.5 sec
9	**Peak period (s)**	n/a
10	**Transaction Volume During Peak period**	20
11	**Total # of FTEs supporting this activity**	1
12	**Expected increase of volume in the next reference period**	Volumes will not be increased.
13	**Level of exception rate**	No expected exceptions
14	**Input data**	n/a
15	**Output data**	Excel spreadsheet containing a table with results.

2.2. Applications Used in the Process

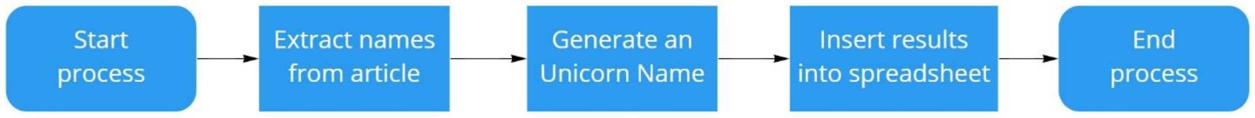
The table includes a comprehensive list all the applications that are used as part of the process automated, at various steps in the flow.

#	Application name & version	System Language	Thin/Thick Client	Environment/ Access method	Comments
1	Internet Explorer 11	PL	Thick Client	Web Browser	Retrieving source data/processing.
2	Excel	PL	Thick Client	Windows Application	Saving the report.

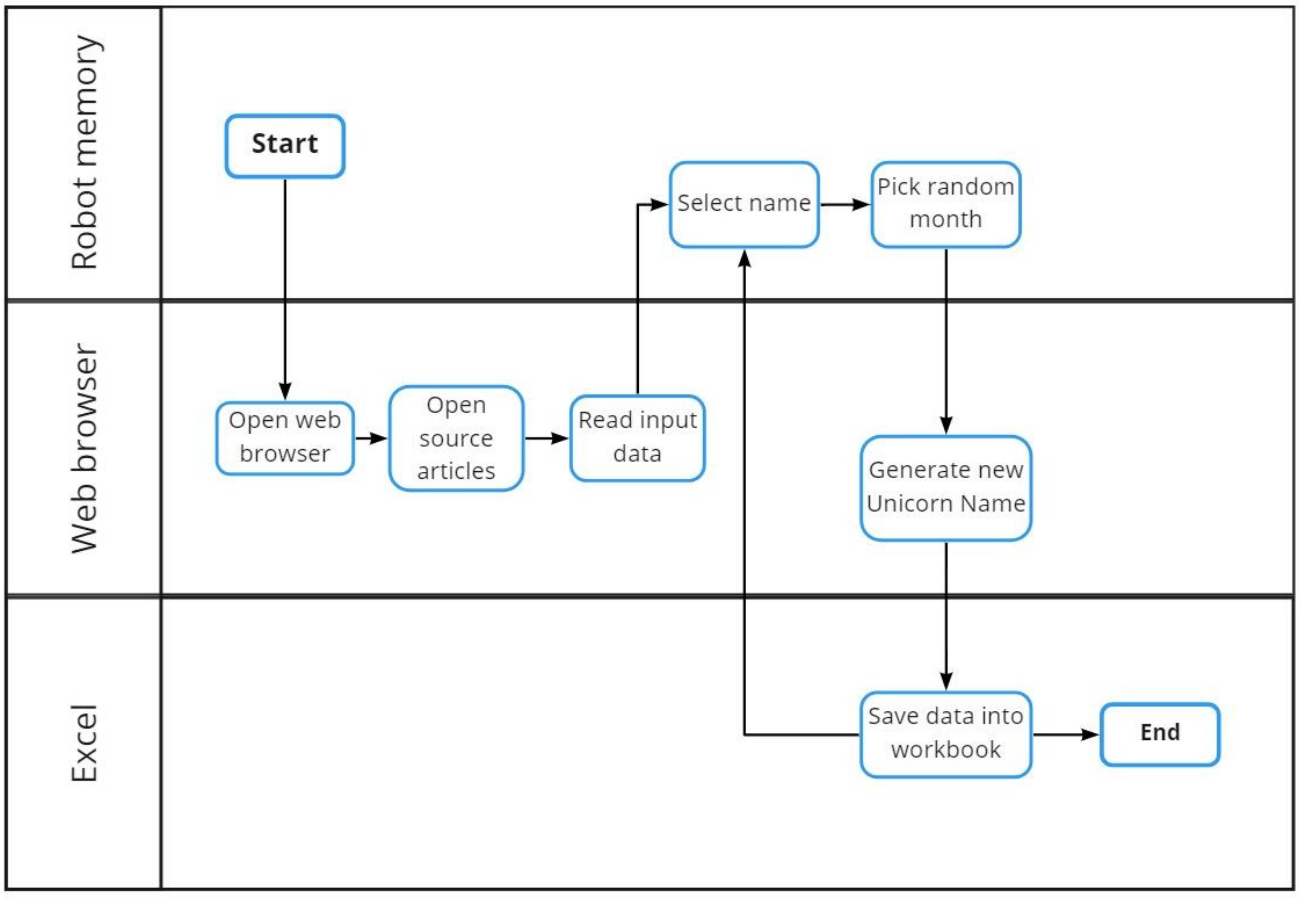
2.3. Process Map

High Level Process Map:

This chapter depicts the business process at a High Level to enable developers to have a high-level understanding of the current process.



Detailed Process Map:



2.4. Input Data Description

Sample	Input type	Location	Inputs are standard? (Yes/ NO)	Inputs are structured?	Data to be used from
Article URL Article URL	Web article	Internet, URL stored in Config	YES	YES	• Children name • Number of children

** Inputs are **standard** if the content is positioned in the same place even if the input types are different.*
E.g. a process that uses at each transaction the same template, so fields to be extracted are always fixed.
*Inputs are **structured** if it is machine readable and digital. Scanned PDF Images/ Free flow texts in Emails are unstructured inputs*

3. Design Specification

This chapter highlights the actual design of the automation.

Solution is based on ReFramework template. Detailed documentation for ReFramework can be found in the Documentation folder within the solution, or on the official UiPath website.

3.1. Project folders structure

The following folder structure is used in the project:

- *.screenshots*
- *.settings*
  - *Debug*
  - *Release*
- *Browser*
- *Data*
  - *Input*
  - *Output*
- *Documentation*
- *Exceptions_Screenshots*
- *Framework*

3.2. Configuration files

Following files are used as configuration sources for the project:

3.2.1. Project.json

This file contains the project metadata:

Variable name	Description
name	The title of the automation project.
description	The description of the project, visible in UiPath Studio.
main	The entry point of the automation project, containing an .xaml file.
dependencies	The list of NuGet packages used to create the automation project, including a version of each package.
schemaVersion	The version of the project.json file.
studioVersion	Describes version of UiPath Studio used to create solution. The parameter is updated with the Studio version in which the latest changes were made.
projectVersion	Current version of the project. Represents the version of the latest published package.
runtimeOptions	Describes variables for solution runtime.
designOptions	Describes options for solution design.
expressionLanguage	The coding language set for the process.
entryPoints	Contains the information for each file marked as an entry point to the process.
isTemplate	Describes whether the project is a template.
templateProjectData	Contains the details for template projects, if applicable.

3.2.2. Config.xlsx

Source of configuration data, specific for the process. The purpose of each configuration variable is described inside this file.
Spreadsheet is divided into 3 separate tabs:

- Settings – contains configuration variables for the process.
- Constants – constants used around the solution.
- Assets – contains mappings between configuration key names and assets names on the orchestrator. Assets are downloaded from the Orchestrator, and added to the Dictionary<string, object> Config object within the solution. Assets will override the configuration variables from Settings, if the name will be the same.

3.3. Input data

Input data is retrieved from article posted by the Polish Ministry of Digital Affairs in February 2019. Article contains a list of names given for children in Poland, with a number corresponding to its frequency. Robot reads the article, and retrieves values from a list using regular expressions. URLs to each article are stored in Config, in tab "Settings". By default, the main articles are linked – however, there are available lists for each voivodership – and those articles could be used as well, by switching the URL of the article for a gender in Config.xlsx file. It is possible to use different voivoderships for girls and boys.

Variables used for storing URLs to each article:

- BoysSourceURL – stores an URL to article with boys names
- GirlsSourceURL – stores an URL to article with girls names.

Robot selects most popular names of each article, with amount based on "NamesAmount" variable stored in settings. By default, this value is set to 10 – but user could change this value, to select a different amount of most popular names. If user picks a greater number than the names available in the article, robot would pick all the names possible and continue processing.

3.4. Output of the process

The only output of the process is an excel spreadsheet, containing the complete results of the process. The workbook consists of only one sheet, named "Unicorn names", and only one table "UnicornTable". The table consists of 5 columns, each representing a different variable:

Column name	Description
Imię	Contains children name.
Płeć	Contains a gender corresponding to the name. Only 2 values are possible – "K", representing girls, and "M", representing boys.
Liczba dzieci	Integer representing the amount of children with given name.
Miesiąc	Contains a month picked randomly during the processing.
Unicorn Name	Contains a generated Unicorn Name.

Template is stored within the solution, in a directory described within a *Config.xlsx* file, in Constants tab. Name of the template is also stored there, as well as the target path for the results file. Finished report is named "*Report HH-MM.xlsx*", where HH represents 2 digits of the current hour, and MM represents 2 digits of the current minute.

Constants used:

- InputPath - Path for input files. Could be relative or full path.
- TemplateFileName - Filename of template excel spreadsheet.
- OutputPath - Path for output files. Could be relative or full path.

Each variable is described within Config.xlsx file as well.

3.5. Project workflows

The following workflows are used in this project:

Workflow file name	Description
Browser directory	**Workflows performing actions on IE browser.**
OpenUnicornBrowser.xaml	Starts a private instance of IE browser, and opens the URL provided in Config.xlsx file, setting: *UnicornNameURL.* Action is retried 3 times before throwing an exception.
RetrieveArticleText.xaml	Starts a new, private and hidden instance of IE browser, and reads the article containing the children names.
RetrieveTransactionData.xaml	Contains 2 separate sequences – for boys and girls, in order of retrieving the children names and a number representing the amount of children with given corresponding name. Gathered data are stored in *TransactionData* collection of DataTable type.
Framework directory	Workflows contained in ReFramework.
CloseAllApplications.xaml	**Part of ReFramework**. Closes all working applications.
GetTransactionData.xaml	**Part of ReFramework.** Gets a single *TransactionItem* from *TransactionData* collection.
InitAllApplications.xaml	**Part of ReFramework.** Opens and initializes applications as needed.
InitAllSeettings.xaml	**Part of ReFramework.** Outputs a settings Dictionary with key/value pairs to be used in the project. Settings and Constants are read from local Config.xlsx file, and then assets are fetched from the Orchestrator.
KillAllProcesses.xaml	**Part of ReFramework.** Kills the working processes used in the automation. Processes killed by the workflow in this solution: *iexplore.exe*
RetryCurrentTransaction.xaml	**Part of ReFramework.** Handles the transaction increment mechanism for System Exceptions – it could either retry current transaction (if maximum retries number was not exceeded), or continue with the next transaction.
SaveResultsToSpreadsheet.xaml	Writes and saves the process results into an excel workbook.
SetTransactionStatus.xaml	**Part of ReFramework.** Sets the TransactionStatus and logs that status with details. Workflow branches into three possible statuses – Success, Business Rule Exceptions, and System Exceptions.
TakeScreenshot.xaml	**Part of ReFramework.** Captures a screenshot, used for System Exceptions.
Main project directory	
Main.xaml	Entry point of the automation. This workflow contains state machine responsible for the control flow of the solution.
Process.xaml	**Part of ReFramework.** This workflow contains activities performed to process a single *TransactionItem*.

Details of ReFramework workflows could be found within ReFramework documentation.

3.6. Retrieving TransactionData

TransactionData is retrieved at the beginning of the process, in Init State. Every time process reaches this state, robot checks if TransactionData was retrieved – if it wasn't (which is True only for the first time robot reaches this state), sequence is performed.

In this sequence, a new instance of TransactionData is created, using Build Data Table activity. TransactionData is a DataTable, which structure is presented in a table below:

Column name	Type	Description
Name	System.String	Contains children name.
Gender	System.String	Contains a gender corresponding to the name. Only 2 values are possible – "K", representing girls, and "M", representing boys.
Number	System.Int32	Integer representing the amount of children with given name.
Month	System.String	Contains a month picked randomly during the processing.
UnicornName	System.String	Contains a generated Unicorn Name.

After building new DataTable, a workflow *RetrieveTransactionData.xaml* is invoked.

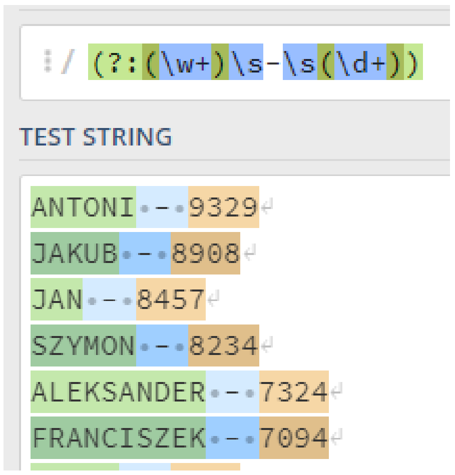
3.6.1. RetrieveTransactionData.xaml workflow

This workflow contains 2 identical sequences – one created for retrieving boys names, and the second for girls names. Each workflow starts with a log explaining which names will be retrieved. Then, another workflow is invoked – *RetrieveArticleText.xaml*, which retrieves an article text to be processed next.

Robot uses regular expressions to retrieve children names and number of children with that name. Pattern used for this expression is:

```
(?:(\w+)\s-\s(\d+))
```

And the expected results looks like this:



```
/ (?:(\w+)\s-\s(\d+))
```

TEST STRING

```
ANTONI - 9329
JAKUB - 8908
JAN - 8457
SZYMON - 8234
ALEKSANDER - 7324
FRANCISZEK - 7094
```

Data for the process is stored within groups inside a single match. *For Each* loop is used to fetch the data from groups, and inserted into new rows within the *TransactionData* DataTable. A new row is created within that loop, using the *NewRow* method on DataTable object.

Retrieved values are inserted into newly created row using *Assign* activity. However, article contains names written in uppercase. To comply with the requirements, names are converted into titlecase, using static method `CultureInfo.CurrentCulture.TextInfo.ToTitleCase` within `System.Globalization` class.

Number of children with that name is converted to System.Int32 type, to comply with the DataTable column type.

After adding new row to the collection, robot checks if enough names were added. User can select how many names of each gender should be processed, by adjusting "NamesAmount" setting within Config.xlsx file. By default, the value is set to '10'. If this amount is reached, loop is broken and process continues. If variable "NamesAmount" is greater than possible names to be retrieved, robot would process all names available.

3.6.2. RetrieveArticleText.xaml workflow

Purpose of this workflow is to retrieve the article text, to be processed within RetrieveTransactionData.xaml workflow. Workflow requires passing an article URL as an argument.

In this workflow, a new instance of IE Browser is started, in Private Mode. Robot reads the article, and immediately closes this tab.

Robot retries 3 times before throwing an exception.

3.7. Process.xaml details

Process.xaml is a main workflow used for processing each transaction. Every action is performed within Attach Browser activity, which uses the browser opened in InitAllApplications.xaml workflow.

Arguments used in workflow:

Argument name	Type	Description
in_Config	System.Collections.Generic.Dictionary<String, Object>	Dictionary structure to store configuration data of the process (settings, constants and assets).
in_MonthsArray	System.Text.RegularExpressions.Match[]	Array containing matches of selectable months in Unicorn Name Generator, to be used as attribute.
in_Random	System.Random	A Random instance, used for selecting a random month.

in_TransactionItem	System.Data.DataRow	Transaction item to be processed.
in_UnicornBrowser	UiPath.Core.Browser	Main browser instance used for processing.

At the beginning of workflow, the browser is refreshed – to make sure that there are no results pending from previous transaction, and robot will get a completely new Unicorn Name. It is confirmed by a FindElement activity, wrapped in a TryCatch activity. After confirming, a random month is selected, using a method *Next*, and picking one string from *in_MonthsArray*. Value is inserted into in_TransactionItem, into column "Month".

When month is assigned, robot enters the values into the fields in the browser. Values are taken from in_TransactionItem, columns "Name" and "Month", and submitted by pressing a button "Get Name".

Name is generated, and outputted on the right side of input fields. Robot confirms with reading the name – if it matches with previously inserted name. If it matches, the Unicorn Name is read, and inserted into in_TransactionItem, into column "UnicornName".

3.8. SaveResultsToSpreadsheet.xaml

This workflow is invoked at the end of the process, after all transactions were performed. First, created a path and filename for a new report file, using a setting *OutputPath* from Config.xlsx file, and current time to create a filename. Robot always uses 2 digits to represent an hour and a minute.

After assigning a path, robot copies the template, based on the path and filename retrieved from Config.xlsx file. If a file with identical name already exists, it always will be overwritten.

Then, robot opens newly copied file, and inserts all retrieved values into the table, expanding it if necessary. Then, file is saved and closed.

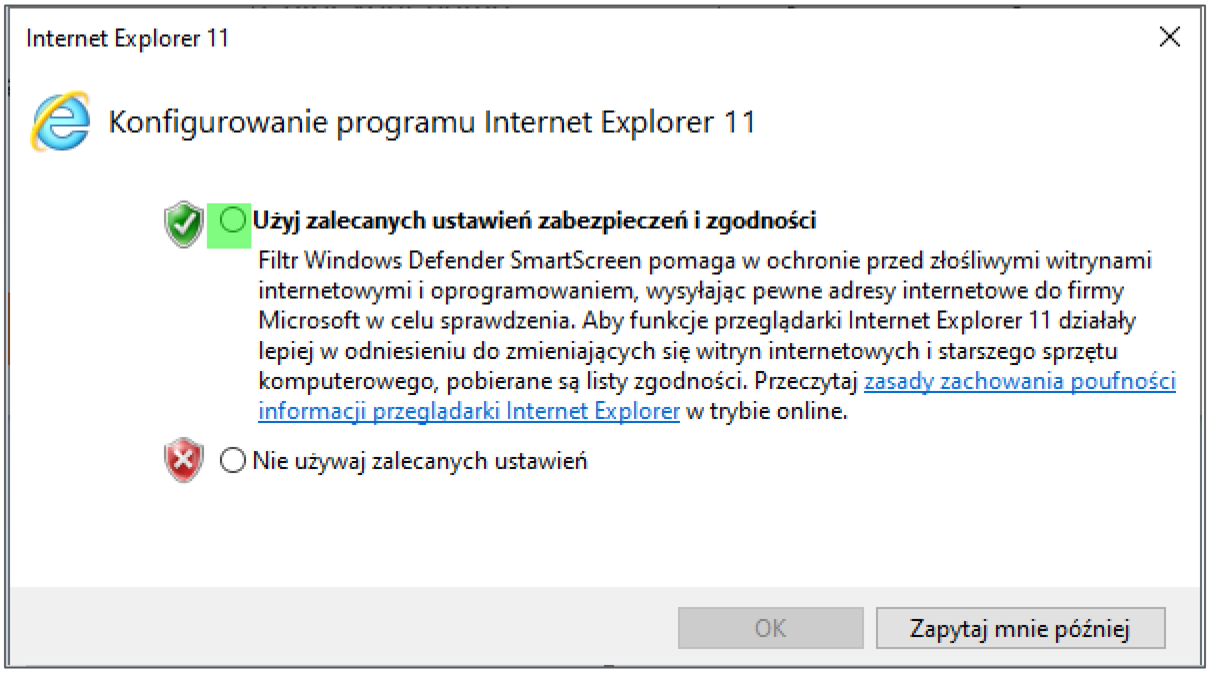
4. System Configuration

4.1. Installation

Prerequisites:

- UiPath Robot is installed.
- Computer has access to Internet connection.
- OS data format changed to Polish (pl-PL) in Region and Language Settings.
- Microsoft Excel is installed and activated.
- Internet Explorer is installed.

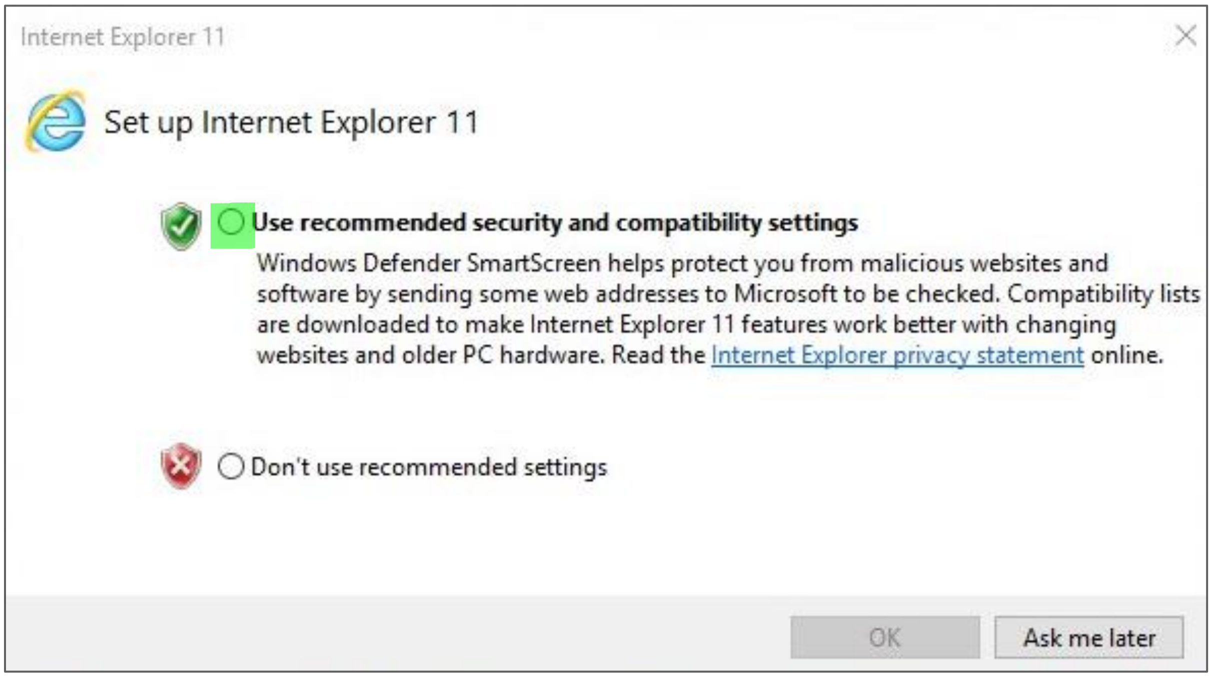
If user will be using Internet Explorer for the first time, there might be an additional popup appearing, which has to be confirmed. User has to select one of two options, or click “Zapytaj mnie później” or “Ask me later” for English version – however, if user selects the latter approach, popup might reappear during the next run. Recommended solution is to select the first option, “Użyj zalecanych ustawień zabezpieczeń I zgodności”, or “Use recommended security and compatibility settings” for English version. Please refer to screenshots below.


Internet Explorer 11

Konfigurowanie programu Internet Explorer 11

Użyj zalecanych ustawień zabezpieczeń i zgodności
Filtr Windows Defender SmartScreen pomaga w ochronie przed złośliwymi witrynami internetowymi i oprogramowaniem, wysyłając pewne adresy internetowe do firmy Microsoft w celu sprawdzenia. Aby funkcje przeglądarki Internet Explorer 11 działały lepiej w odniesieniu do zmieniających się witryn internetowych i starszego sprzętu komputerowego, pobierane są listy zgodności. Przeczytaj zasady zachowania poufności informacji przegladarki Internet Explorer w trybie online.

Nie używaj zalecanych ustawień


Internet Explorer 11

Set up Internet Explorer 11

Use recommended security and compatibility settings
Windows Defender SmartScreen helps protect you from malicious websites and software by sending some web addresses to Microsoft to be checked. Compatibility lists are downloaded to make Internet Explorer 11 features work better with changing websites and older PC hardware. Read the Internet Explorer privacy statement online.

Don't use recommended settings