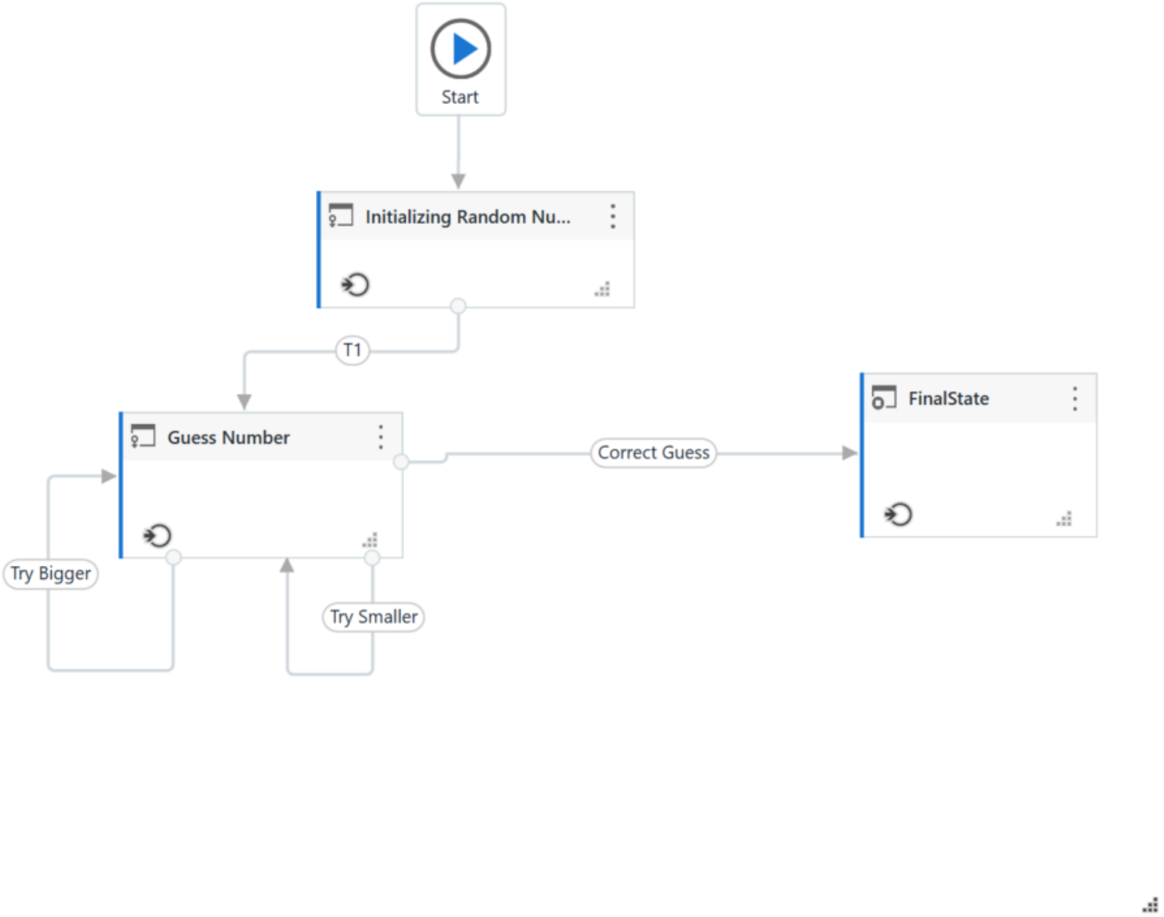
we are going to build the guessing game we did in the previous chapter, the only difference being that we will try to guess a number between 1 and 300.

1. Create a blank process and, on the **Design** tab, in the **File** group, select **New > State Machine**. The **New State Machine** window is displayed.

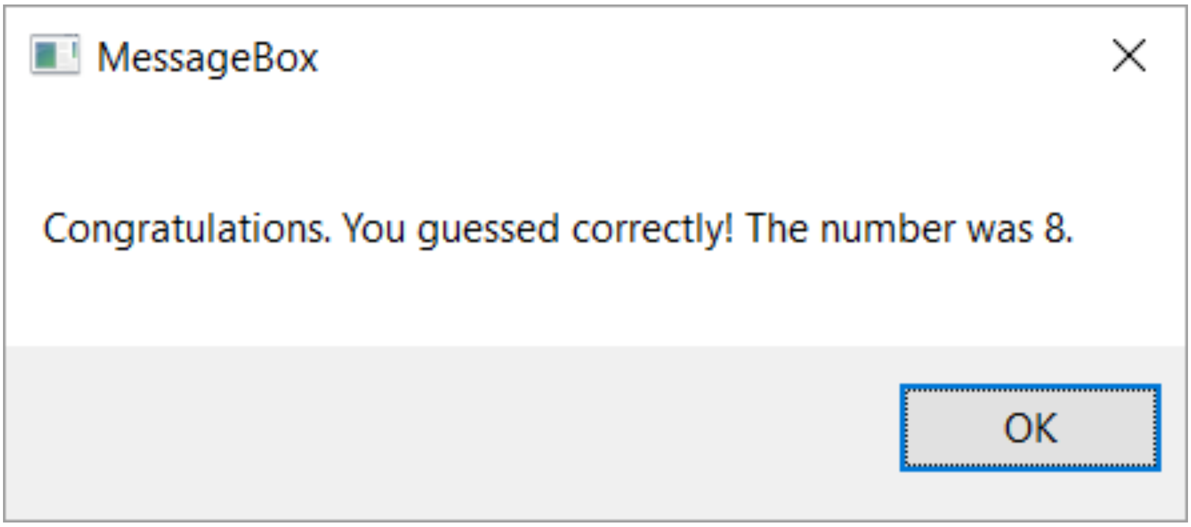
- Note: You can also add a State Machine activity to the **Designer** panel to create a new state machine automation.
- In the **Name** field type a name for the automation, such as "First State Machine", and leave the default project location or add a subfolder. Click **Create**. The **Designer** panel is updated accordingly.
- Create two integer variables, `InitialGuess` and `RandomNumber`. The first variable stores your guess, while the second stores the random number.
- Add a **State** activity to the **Designer** panel and connect it to the **Start** node. This is the initial state, and it is used to generate a random number.
- Double-click the activity. This **State** activity is displayed expanded in the **Designer** panel.
- In the **Properties** panel, in the **DisplayName** field, type Initializing Random Number. This enables you to easily tell states apart.
- In the **Entry** section, add an Assign activity.
- In the **To** field, add the `RandomNumber` variable.
- In the **Value** field, type `new Random().Next(1,300)`. This expression generates a random number.
- Return to the main project view and add a new **State** activity.
- Connect it to the previously added activity.
- Double-click the last added **State** activity. This activity is displayed expanded in the **Designer** panel.
- In the **Properties** panel, in the **DisplayName** field, type Guess Number. This state is used to prompt the user to guess a number.
- In the **Entry** section, add an Input Dialog activity.
- Select the **Input Dialog**, and in the **Properties** panel, add an appropriate **Label** and **Title** to prompt the user to guess a number between 1 and 300.
- In the **Result** field, add the `InitialGuess` variable. This variable stores the user's guess.
- Return to the main project view and create a transition that points from the Guess Number state to itself.
- Double-click the transition. The transition is displayed expanded in the **Designer** panel.
- In the **Properties** panel, in the **DisplayName** field, type Try Smaller. This message is displayed on the arrow, enabling you to run through your automation easier.
- In the **Condition** section, type `InitialGuess` > `RandomNumber`. This verifies if the user's guess is bigger than the random number.
- In the **Action** section, add a Message Box activity.
- In the **Text** field, type something similar to "Your guess is too big. Try a smaller number." This message is displayed when the user's guess is bigger than the random number.
- Return to the main project view and create a new transition that points from the **Guess Number** state to itself.
- Double-click the transition. The transition is displayed expanded in the **Designer** panel.
- In the **Properties** panel, in the **DisplayName** field, type "Try Bigger". This message is displayed on the arrow, enabling you to run through your automation easier.

- In the **Condition** section, type `InitialGuess` < `RandomNumber`. This verifies if the guess is smaller than the random number.
- In the **Action** section, add a **Message Box** activity.
- In the **Text** field, type something similar to "Your guess is too small. Try a bigger number." This message is displayed when the users guess is smaller than the random number.
- Return to main project view and add a **Final State** activity to the **Designer** panel.
- Connect a transition from the **Guess Number** activity to the **Final State**.
- In the **Properties** panel, in the **DisplayName** field, type "Correct Guess".
- In the **Condition** field, type `InitialGuess` = `RandomNumber`. This is the condition on which this automation steps to the final state and end.
- Double-click the **Final State** activity. It is displayed expanded in the **Designer** panel.
- In the **Entry** section, add a **Message Box** activity.
- In the **Text** field, type something similar to "Congratulations. You guessed correctly! The number was " + `RandomNumber.ToString` + "." This is the final message that is to be displayed, when the user correctly guesses the number.

The final project should look as in the following screenshot.

- 

- Press F5. The automation is executed correctly.

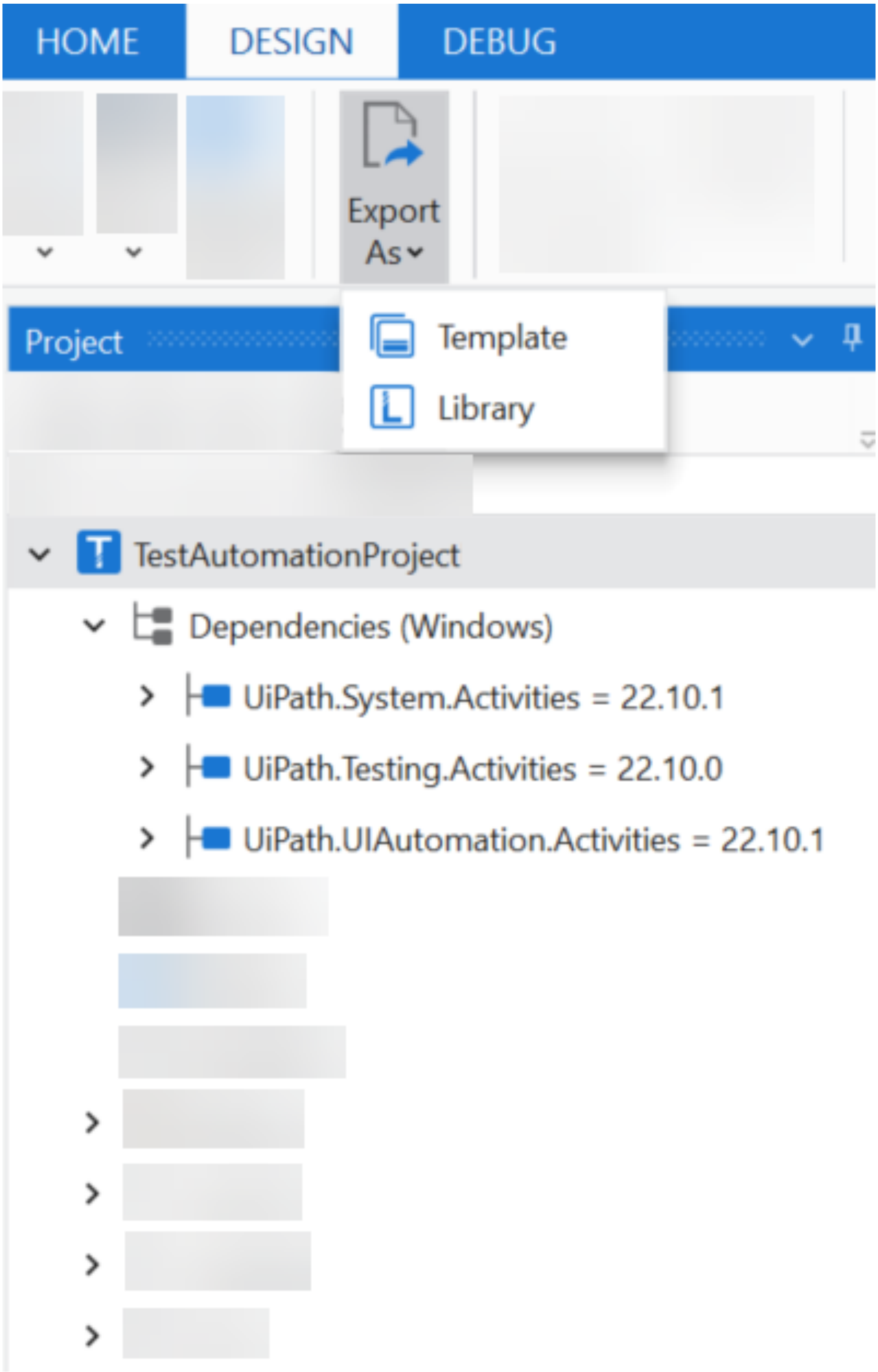
36. 

MessageBox

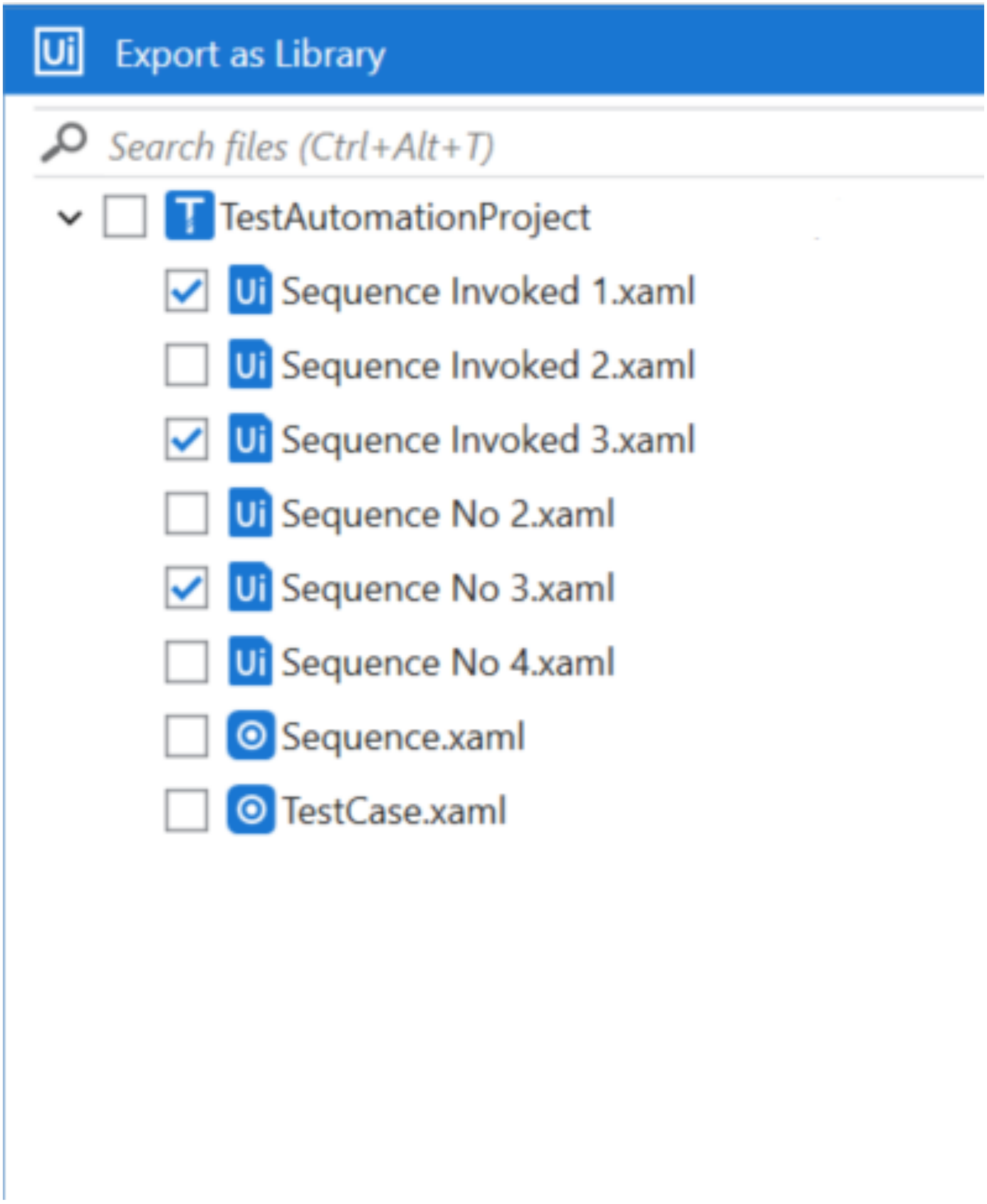
Congratulations. You guessed correctly! The number was 8.

Load Project as a library:

1. In the **Design** ribbon, select **Export As** and then **Library**.


HOME
DESIGN
DEBUG
Export As
Template
Library
Project
TestAutomationProject
Dependencies (Windows)
UiPath.System.Activities = 22.10.1
UiPath.Testing.Activities = 22.10.0
UiPath.UIAutomation.Activities = 22.10.1

1. Select the workflows that you want to export as a library. By default, the entire project is deselected, along with the entry points (**Main** workflow, and **Test Cases**).



Export as Library

Search files (Ctrl+Alt+T)

- □ TestAutomationProject
  - ☑ Sequence Invoked 1.xaml
  - □ Sequence Invoked 2.xaml
  - ☑ Sequence Invoked 3.xaml
  - □ Sequence No 2.xaml
  - ☑ Sequence No 3.xaml
  - □ Sequence No 4.xaml
  - □ Sequence.xaml
  - □ TestCase.xaml

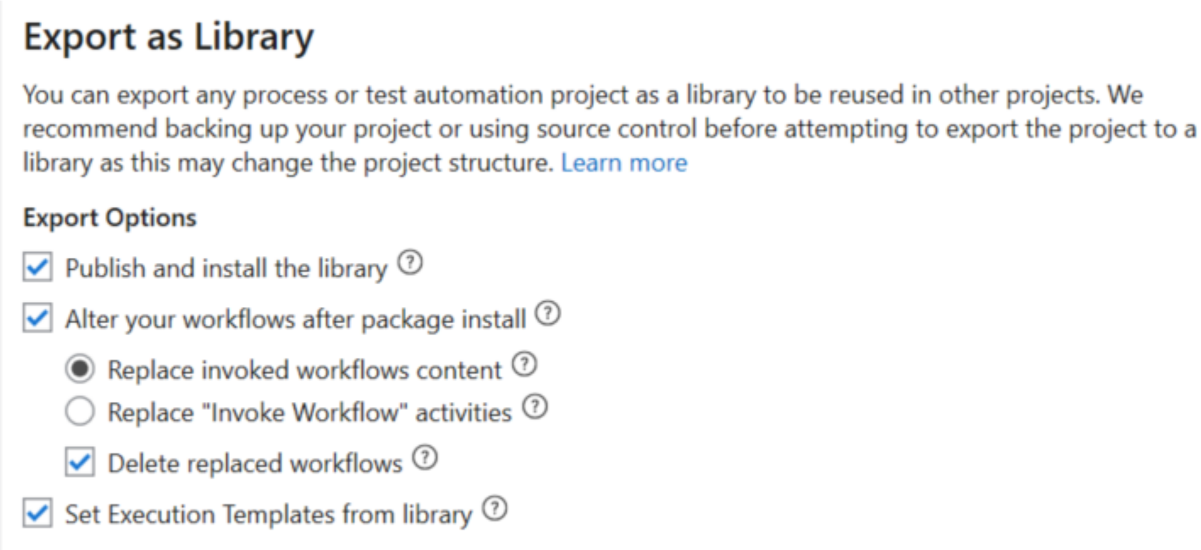
2. Configure the following **Extract Options**:
    - **Include Test Cases**: By default, this option is enabled for a Process project and disabled for a Test Automation project. You should enable this option if you want to include test cases as part of the extracted library.
    - **Publish and install the library**: Automatically selected to publish the package to a shared feed and install the library as a project dependency. You can disable this option if you want to create the library without publishing it. If disabled, the remainder options are not available for configuration.
    - **Alter your workflows after the package install**: Choose to modify the workflows with activities that have been compiled from the library.
    - **Replacing Mode**: Choose the workflow replacing method.

    - Select **Replace invoked workflows content** to change the workflows invoked from entry point, or test cases with corresponding activities from the extracted library.
    - Use this option if you use the Isolated and Target Sessions properties for Invoke Workflow activities to run in a separate Windows process, and start in a different session, respectively.
    - Select **Replace "Invoke Workflow" activities** to change the Invoke Workflow activities with activities from the extracted library.

      WARNING: Do not select this option if you are using the Isolated and Target Sessions properties for Invoke Workflow activities.

  - **Delete replaced workflows**: Delete the workflows that have been replaced by the extracted library activities.
  - **Set Execution Templates from library**: Add execution templates to the library.

3. Click **Export** to confirm the library options.



Export as Library

You can export any process or test automation project as a library to be reused in other projects. We recommend backing up your project or using source control before attempting to export the project to a library as this may change the project structure. Learn more

Export Options

☑ Publish and install the library ⓘ

☑ Alter your workflows after package install ⓘ

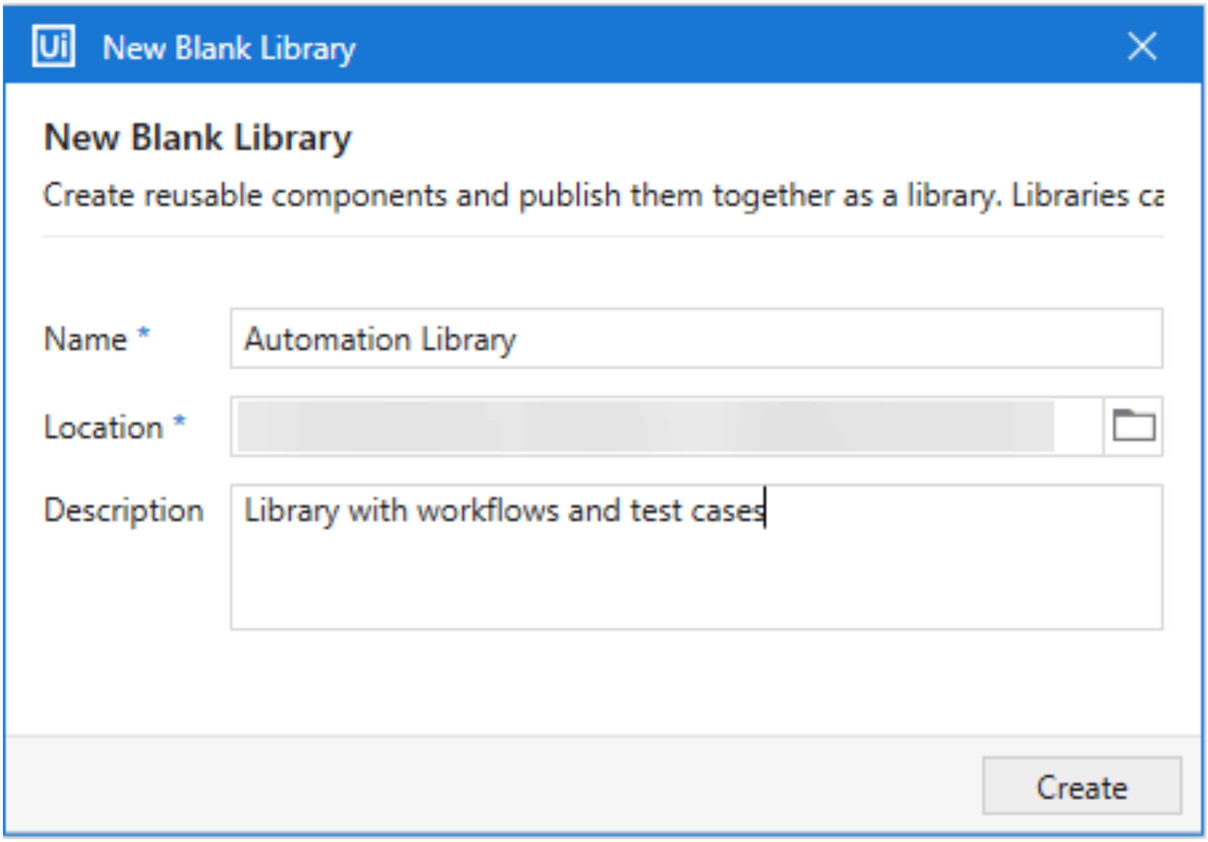
   ● Replace invoked workflows content ⓘ

   ○ Replace "Invoke Workflow" activities ⓘ

   ☑ Delete replaced workflows ⓘ

☑ Set Execution Templates from library ⓘ

4. Click **Create** to finish library creation.



New Blank Library

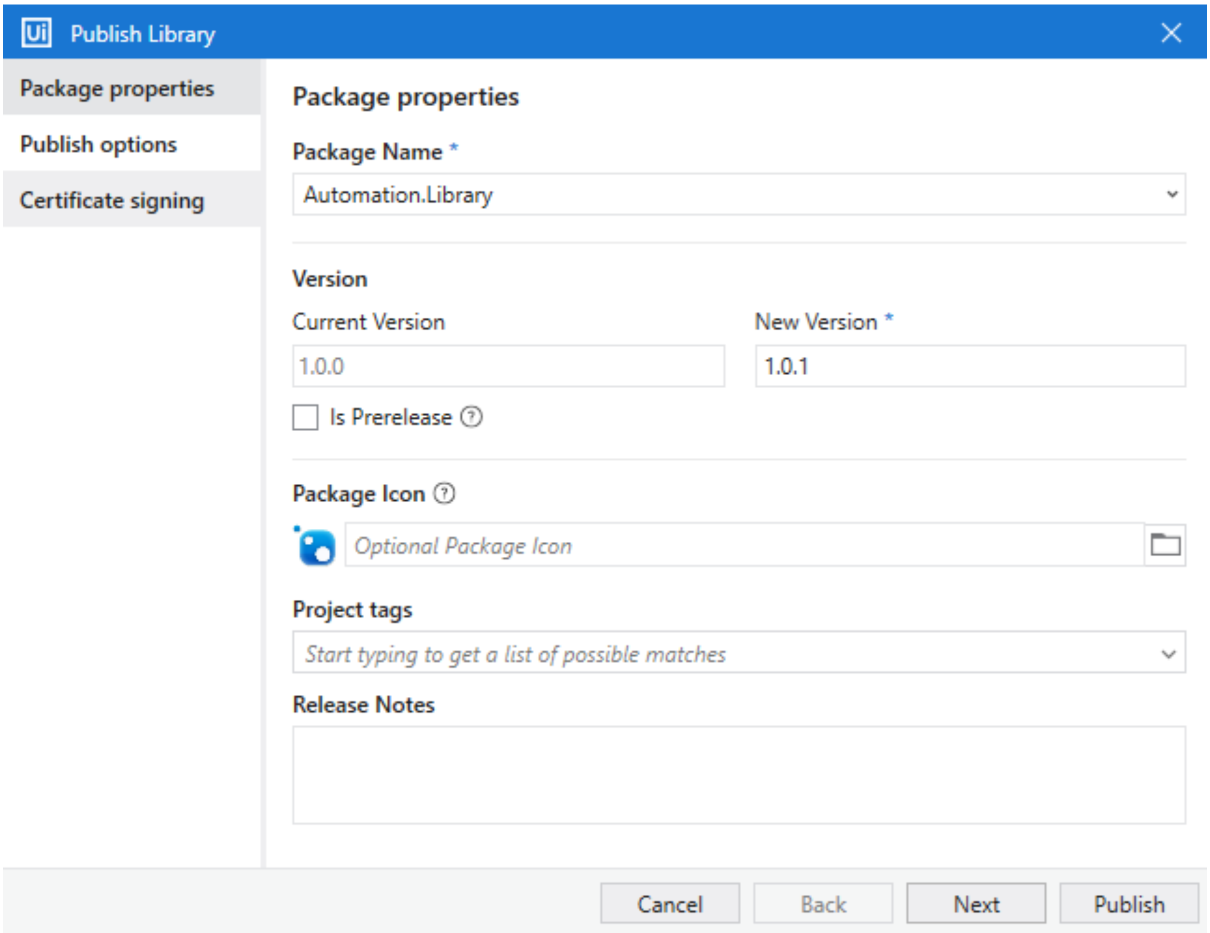
Create reusable components and publish them together as a library. Libraries ca

Name * Automation Library

Location *

Description Library with workflows and test cases

5. (Optional) Publish the library if you've selected **Publish and install the library** in step 3, and then click **Publish**.



Publish Library

Package properties

Package Name *

Automation.Library

Version

Current Version: 1.0.0

New Version *: 1.0.1

□ Is Prerelease

Package Icon

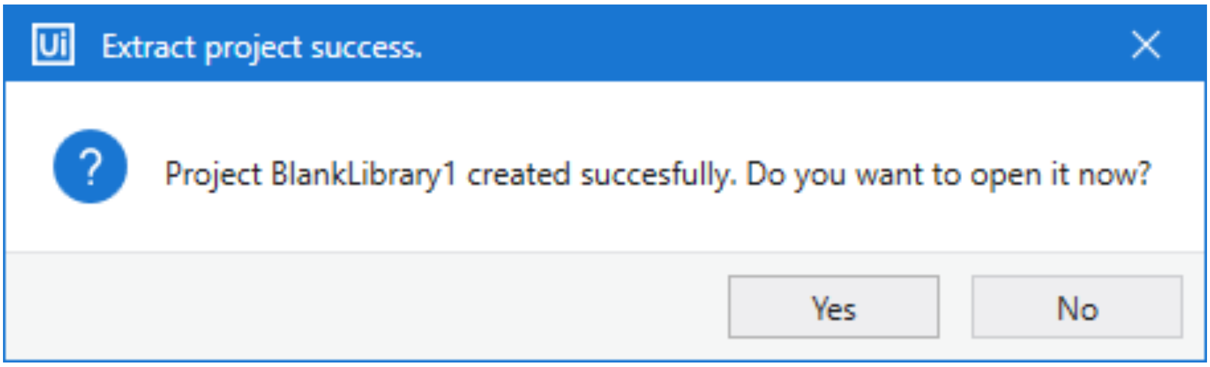
Optional Package Icon

Project tags

Start typing to get a list of possible matches

Release Notes

NOTE: In case you didn't enable the **Publish and install the library** option, you'll be prompted to open the library or continue with the current project.



Extract project success.

Project BlankLibrary1 created succesfully. Do you want to open it now?

6. Create a new process with this library from the left side menu