RPA Design and Development

v4.0

Lesson 10 | Object Repository

Object Repository – Exam Topics

1. Create, publish, and consume a UI Library with static and dynamic descriptors.

What is Object Repository?

The object repository is a **centralized storage system** that allows developers and automation teams to **store, manage, and reuse UI elements** by capturing them as objects in a DOM-like repository, **sharable across projects.**

The Object Repository allows for creating and reusing UI taxonomies inside and across automation projects.

With Object Repository you can build a UI API for your application and share it with your team within minutes.

Object Repository's Key Features

1. Centralized Storage
2. Reusability
3. Version Control
4. Collaboration
5. Metadata and Documentation
6. Search and Discovery
7. Dependencies Management
8. Security and Access Control

Object Repository's Key Features (1 of 2)

Feature	Description
1. Centralized Storage	Object repositories allow you to store and manage reusable components, activities, and workflows so they are easy to access, update, and maintain.
2. Reusability	A repository enables the creation of reusable components, activities, and workflows that can be easily shared and reused across multiple automation projects. This saves time and effort by eliminating the need to recreate the same functionality across multiple projects.
3. Version Control	Object repositories support version control, so developers can manage different versions of components or activities. When updating or making changes to existing assets, this feature ensures that older versions can still be referenced if necessary.
4. Collaboration	It facilitates collaboration between team members by providing a shared place to store and share automation assets. Multiple developers are able to collaborate, contribute to, and benefit from each other's work.

Object Repository's Key Features (2 of 2)

Feature	Description
5. Metadata and Documentation	Developers can add metadata and documentation to describe the purpose, usage, and dependencies of each asset, which improves understanding and promotes consistency when using the assets across projects.
6. Search and Discovery	Search capabilities provide quick access to specific components or activities based on keywords, descriptions, or tags, reducing duplication.
7. Dependencies Management	The object repository enables developers to define relationships and dependencies between different components and activities so that changes in one asset will be reflected in all dependent assets.
8. Security and Access Control	It provides security and access control features, which enable administrators to define user roles and permissions, ensuring that only authorized individuals are permitted access, modification, and deletion of repository assets.

Benefits of using the Object Repository

Object Repository's Key Benefits

- UI elements across the project are managed, updated, and modified from a centralized place
- View a list of all your UI activities inside your process by using the UI Activities tab inside Object Repository panel
- Quickly capture elements you need in your automation with the Capture Elements wizard
- Increased selector reliability with the help of the Capture Elements recorder that captures elements, together with their anchors
- Drag-and-drop elements from the Object Repository panel
- Objects are reusable in local project or across projects when packaged as libraries
- Upgrade application and process UI elements in one go with UI libraries

About Object Repository

About Object Repository

https://docs.uipath.com/studio/standalone/2022.10/user-guide/about-object-repository

Object Repository Key Concepts

Object Repository provides a structured way for organizing and maintaining the objects, enabling **easy reuse and maintenance**.

The object repository has a **tree structure** where each node is an object representing screens or elements, all hierarchical under the application.

The structure is as follows:

1. **Application**: It can be one of **2 types: mobile or desktop/web**, depending on what technology is used for UI Automation.
2. **Version**: Applications can have multiple versions
3. **Screen**: Top-level window of an application version that can only be created under an app version.
4. **UI Element:** An object on the screen with a descriptor and metadata. It can be of multiple types.

Object Repository Key Components

The key components related to the Object Repository are

1. UI Descriptors
2. UI Applications
3. Screens
4. UI Elements
5. UI Libraries
6. UI Activities

Object Repository : UI Descriptor

A UI Descriptor is a superset of selectors.

It holds information for uniquely identifying elements on the screen.

UI Descriptors are extracted from activities in the workflow and added to a structured schema that groups them by

Applications, Application Versions, Screens, and UI Elements.

Out of this taxonomy structure, **only Screens and Elements hold descriptor information**.
The rest are used for grouping and their role is to ensure upgrades between versions of an application.

UI Descriptors can be part of:

- One project for wide reuse
- Snippets repositories for testing purposes
- UI Libraries for global cross-project sharing

Object Repository : UI Application

A UI Application is a targeted application that **can have multiple versions** and **each version can have multiple screens**.

Applications can be of multiple types:

- Desktop/ Web Application
- Mobile Application

Note: For defining mobile applications, you need to use **UiPath.MobileAutomation.Activities** package.

Object Repository : Screens

Screens are UI Scopes that are either

- extracted from activities inside the workflow or are
- generated at element capture time

A screen **groups together multiple elements belonging to the same screen.**

Object Repository : Ui Elements

UI Elements contain

- Full or Partial element selectors
- Anchor selectors
- Screen and element image capture context
- Other metadata that describes the element on the screen

Object Repository : Ui Libraries

A UI Library is an encapsulation of **elements grouped by applications, application versions, and screens**.

The elements you define can be extracted as a UI Library, and after publishing, can be installed in other projects as a dependency.

A UI Library may contain several applications but can contain only one version of a certain application.

This mechanism ensures that when you upgrade a dependency, you also upgrade the application version you were using inside your projects.

Object Repository : Ui Activities

UI Activities allows you to view a list of all your UI activities inside your process.

Object Repository : UI Library Structure

The structure of UI libraries created with the Object Browser has the following hierarchy:
Application > Version > Screen > UI Element.

Let's see all the elements depicted below in the process of capturing elements using the Capture Elements recorder in the Object Repository.

Object Repository : UI Library Structure

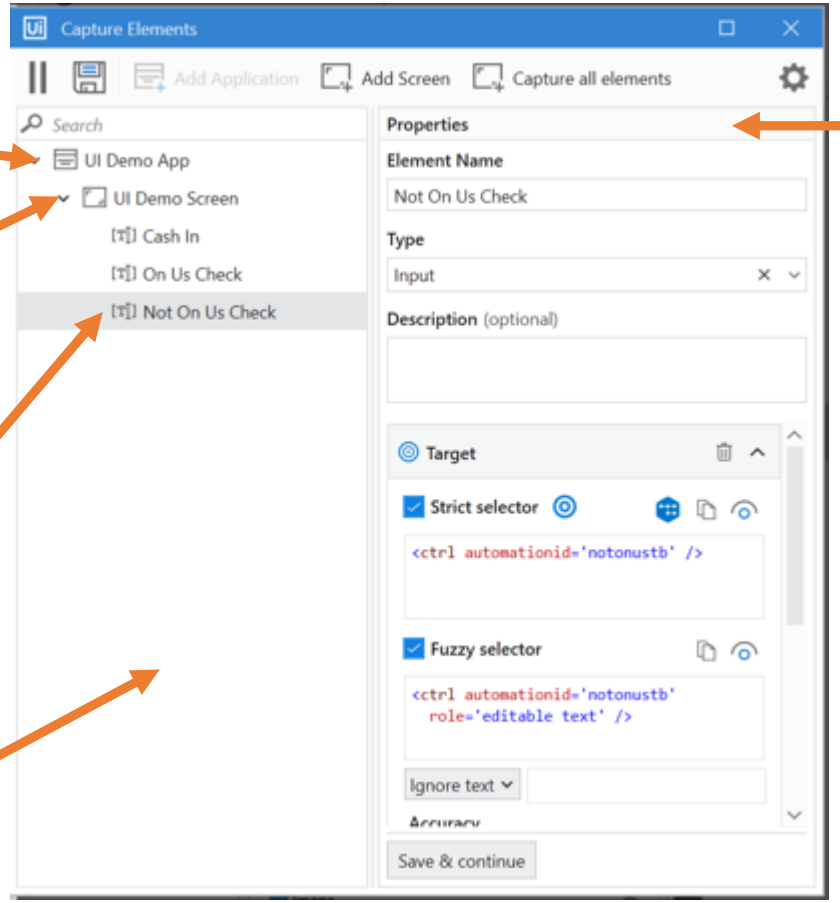
Let's see all the elements depicted below in the process of capturing elements using the Capture Elements recorder in the Object Repository.

The Application

A Screen

Ui Element
contains identification & Descriptor Information

Element Hierarchy
Visible on the left-hand side panel



UI Element Properties

The Properties include the Element Name, Type, Description and the Descriptor. The Element Name, Type and Description help developers identify the object. The Descriptor helps the Robot identify the element on the application screen. In this example, the Descriptor includes several targeting methods used to identify the Not On Us Check field :

- Strict Selector
- Fuzzy Selector
- Image
- Anchor

Object Repository : Creation Options

An object repository is a collection of elements grouped by applications, versions, and screens.
It can come **from a local project**, i.e. the local repository, **or from a library**, i.e. UI Library dependencies.

Creation Method	Description
Creating a Local Repository	You can create Local Repository by Capture Elements recorder to capture UI descriptors. Also , you can manually create applications, screens, and elements directly in the Object Repository panel.
Creating a UI Library	You can extract the local repository from any project into a library project and publishing it as a NuGet package. You can start from a new library, build the repository there, and publish it as a NuGet package. You can then use the UI Library NuGet packages and add them as dependencies to your processes.

Note: The Capture Elements recorder only records objects for reuse in projects,
the recorder does not generate a workflow in the Designer panel.

Object Repository Panel

Object Repository panel provides several features and functionalities that can be leveraged to create and reuse UI taxonomies inside and across projects.

- You can create a repository using Capture Elements button in the Object Repository. With the Capture All Elements feature, you can capture all the elements of an application using Computer Vision and add them to your Object Repository
- You can create a new library that can be installed in other projects as a dependency. Also, you can update an existing library if you exit the recorder before capturing all the desired
- You can add descriptors to the Snippets panel to reuse them or extract them as UI libraries
- You can access objects from UI libraries installed as dependencies to the current project

Object Repository Panel Options

Expand All	Collapse All	Refresh	Add a new Element under this screen	Capture Elements
Expands all of the categories in the Descriptor tab.	Collapses all of the categories in the Descriptor tab.	Refreshes the content of Object Repository.	Add a new Element under the selected screen	Use to capture elements

Manual Creation of a Repository

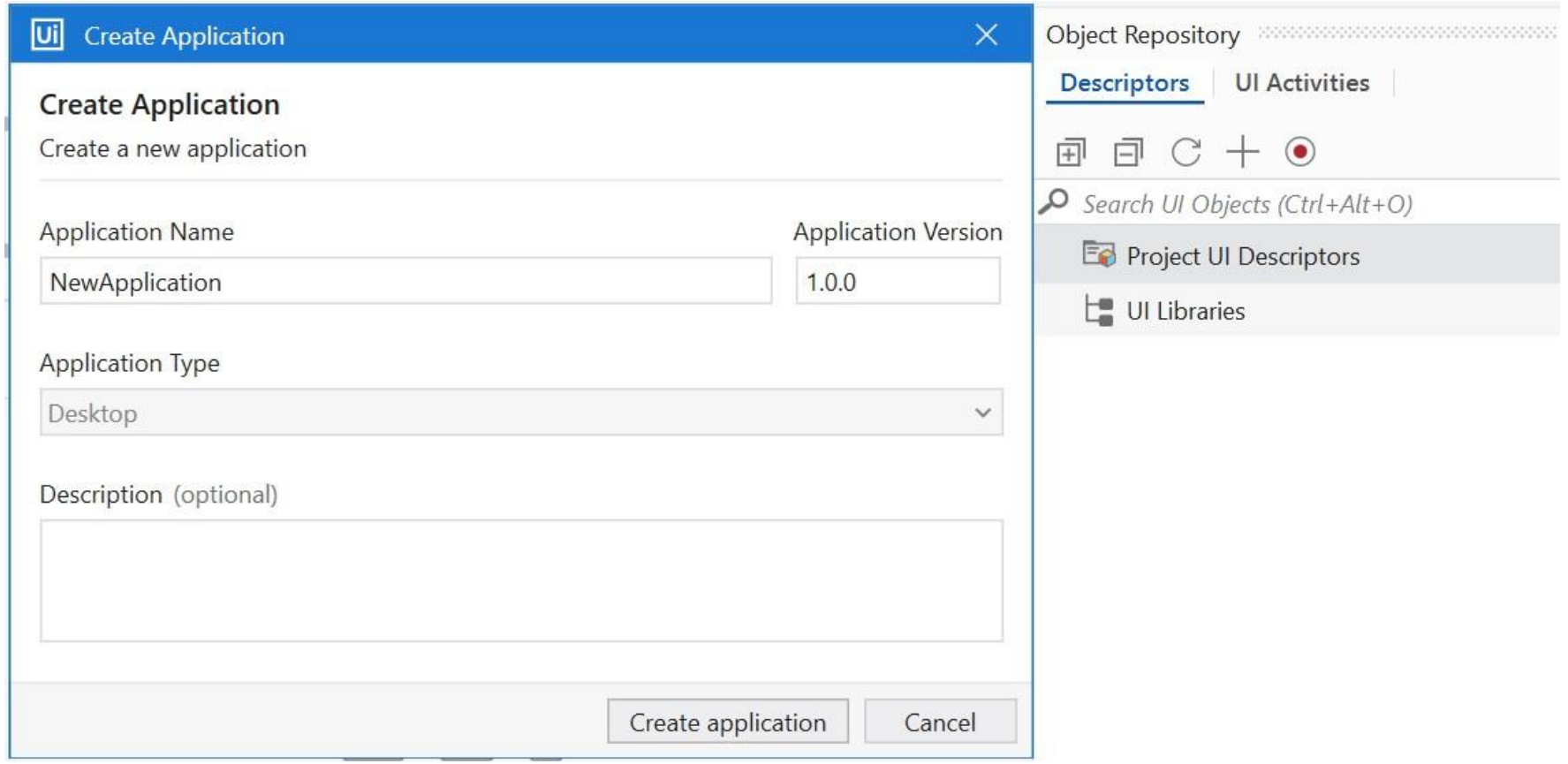
You can create your own repo from within the Object Repository panel by defining **the application, each screen, and element manually.**

1. Create Ui Application

You can create a UI application by **clicking on the plus sign** in the Descriptors tab or **right-click Project Descriptors** in the same tab and select Create Application.

Add a unique application name, version, and description.

Click Create application.
Your application is visible in the Descriptors tab, under Project UI Descriptors.

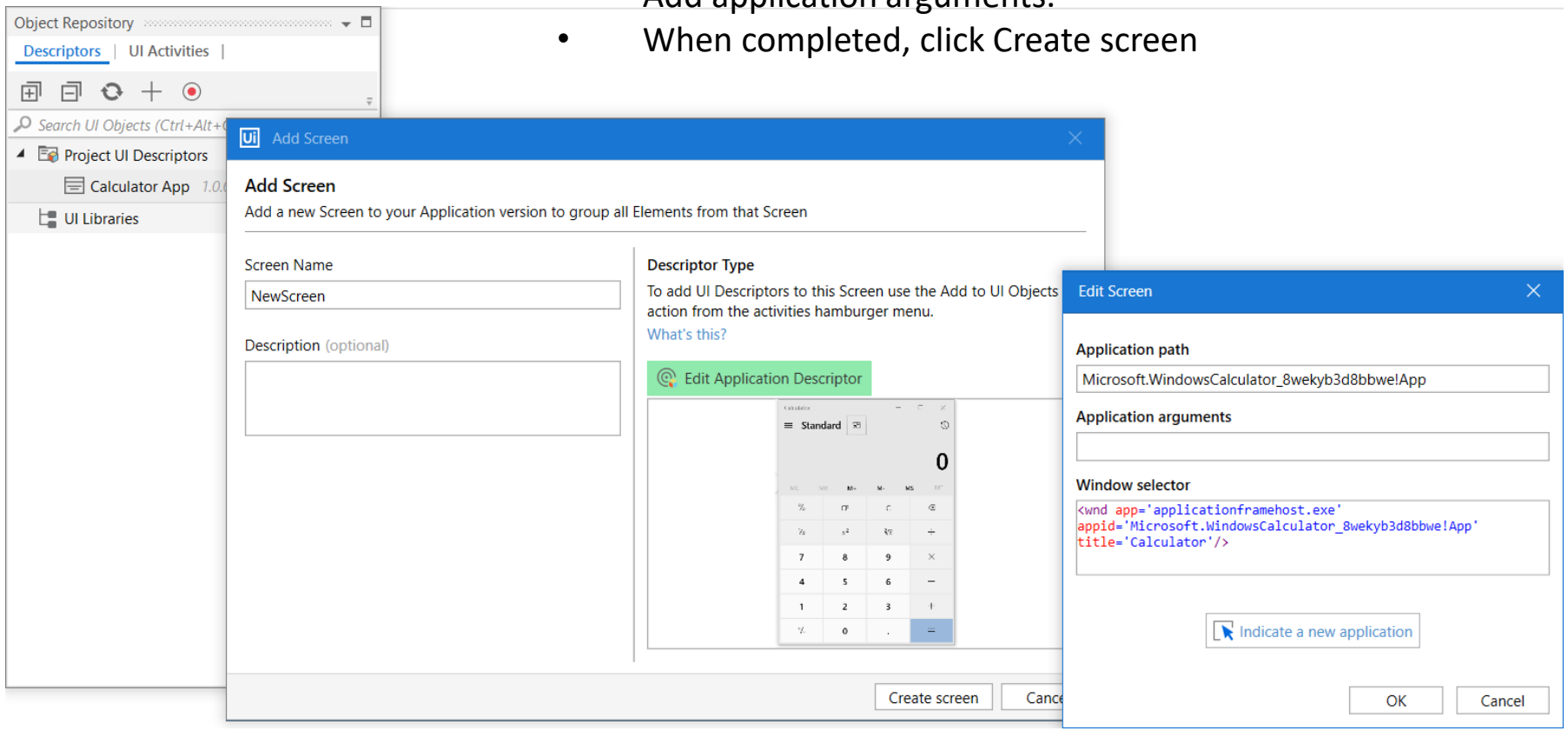


Manual Creation of a Repository

2. Create a Screen

By selecting an app for example here Calculator app, click the plus sign or right-click your application and select Create Screen. The Add Screen window opens, add a unique name for the screen, and a description. Indicate the screen that you would like to automate, and edit the captured element with Unified Target:

- Change the application path if needed
- Add application arguments.
- When completed, click Create screen



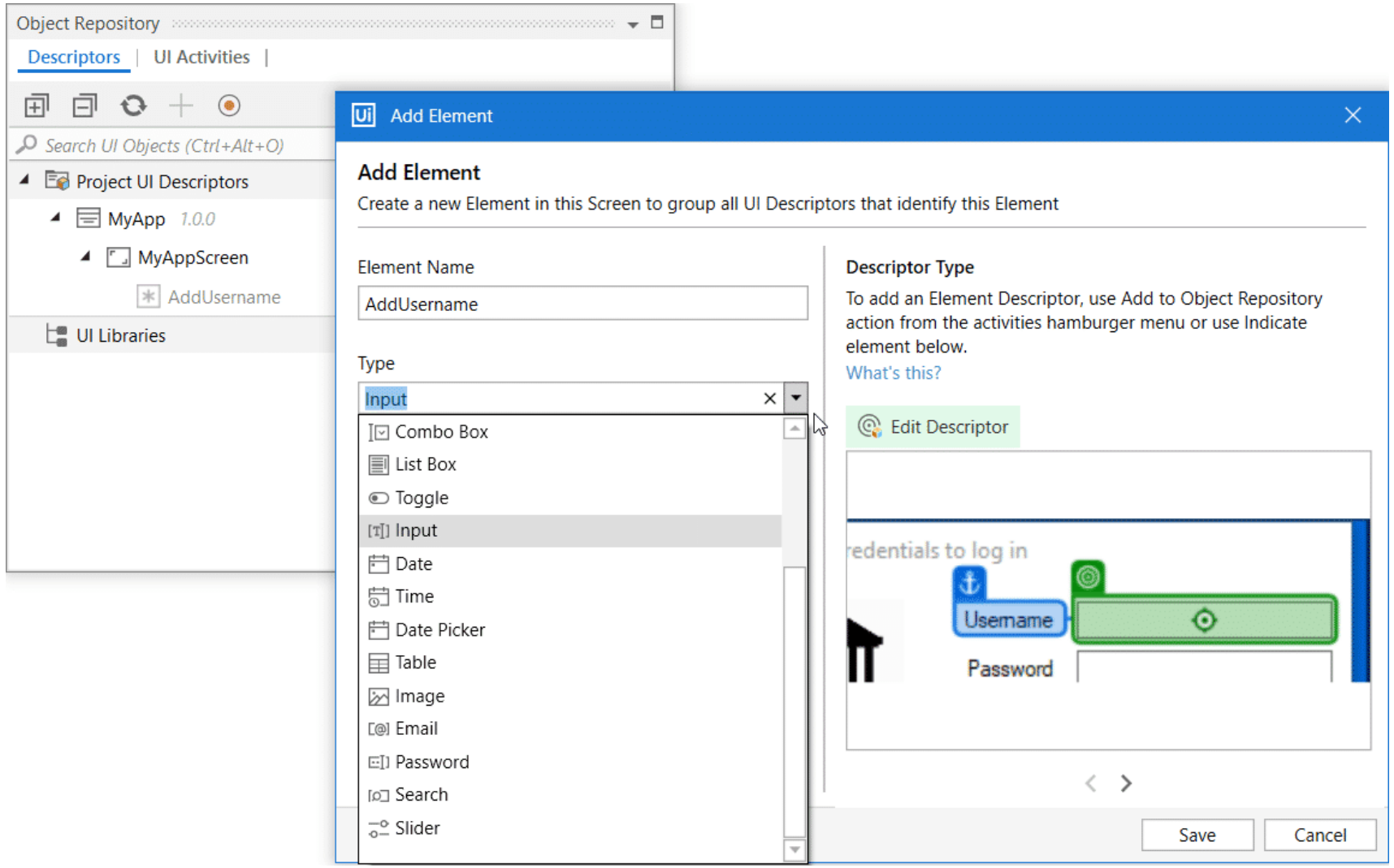
Manual Creation of a Repository

3. Create Ui Element

You can create a UI Element by selecting the app screen, click the plus sign or right-click the screen and select Create Element. The Add Element window opens:

From the Descriptor Type section click Indicate element to use Unified Target to indicate and capture the element from the screen.

Add an intuitive name in the Element Name field and select the type of the element from the drop-down list under Type.
Click Save when done.



Manual Creation of a Repository

NOTE:

- To edit an element from the Object Repository, right-click it, and then select Edit Element, or double-click the element in the tree.
- To edit the descriptor of an element, use the Edit Descriptor option from the Edit Element window, or right-click the element in the tree, and then select Edit Descriptor.

Creating and Publishing a UI Library

Capture all relevant :

- UI objects identifying the app
- Screens
- UI elements in the target application

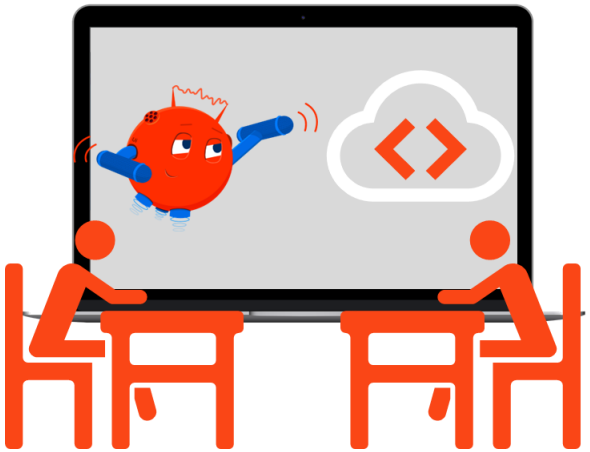
There are two ways of doing this:

1. You can start from an empty Library project and capture the objects
2. You can export them from a pre-existing Process type project which uses Modern UI Activities

After you've extracted the needed elements :

- You can fine-tune them for increased reliability and
- Then publish the UI Library locally or to Orchestrator

Exercise - Create Object Repository Library



Create a new Library type project and provide a descriptive name
Use the Capture Elements recorder to capture elements in Microsoft Edge.

Capture the following elements: (https://acme-test.uipath.com)

The ACME Website in Microsoft Edge(Application)

The ACME Login Screen (Screen)

The Email Field (UI Element)
The Password Field (UI Element)
The Login Button (UI Element)

The ACME Dashboard (Screen)

All the Buttons (UI Element)

The Employee Screen (Screen)

Close Button (UI Element)
Select an employee (UI Element) & set "Brad M Roberts" as the default value to a newly created variable in a dynamic descriptor
The Log Out Button (UI Element)

Once you have captured the elements
Edit the properties and descriptors
Publish the Library locally

UiPath Academy

Chapter 15:

Object Repository in Studio

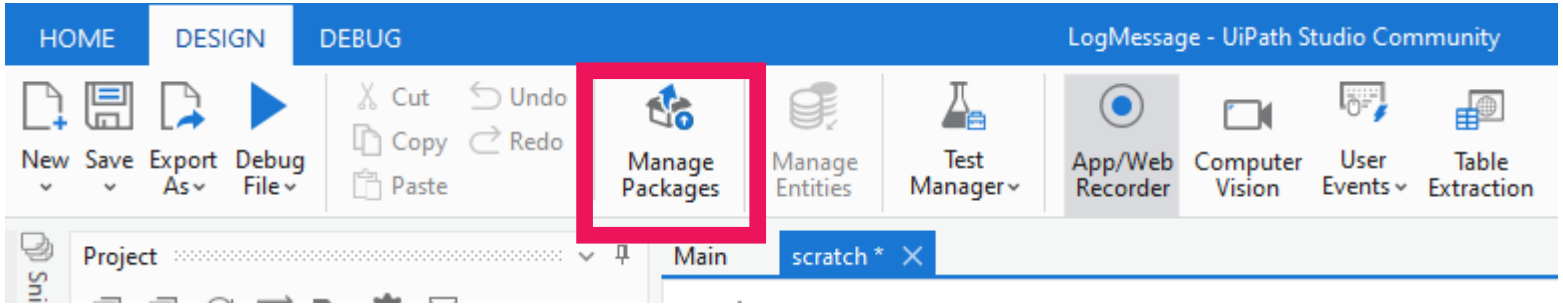
Creating and Publishing a UI Library

Resources

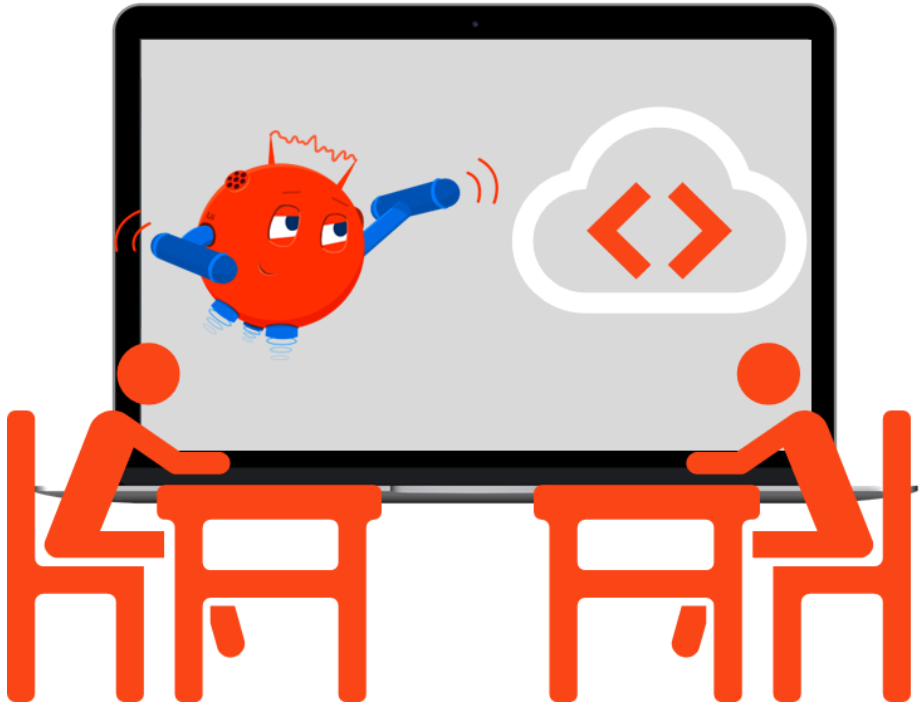
Topic	Link
Creating an Object Repository	https://docs.uipath.com/studio/standalone/2022.10/user-guide/creating-an-object-repository#create-a-ui-element
About Libraries	https://docs.uipath.com/studio/standalone/2022.10/user-guide/about-libraries
Creating a Basic Library	https://docs.uipath.com/studio/standalone/2022.10/user-guide/creating-basic-library
About Publishing Automation Projects	https://docs.uipath.com/studio/standalone/2022.10/user-guide/about-publishing-automation-projects

Import a Locally Published UI Library into a New Process

Import an existing UI library using the **Manage Packages window** and use the descriptors in the library to develop UI automation workflows



Exercise - Import and use a UI Library



Please note that some of the UiElements required to complete this task currently do not exist in the ACME Ui Library

Create a new process called **ACME Employee Details**

(https://acme-test.uipath.com)

1. Import the locally published ACME UI Library (from the previous exercise) as a dependency into your project

2. Using the UI Library for the descriptors build three workflows

a) Login.xaml - logs into ACME webpage
b) EmployeeDetails.xaml - Retrieves suggested employee details
c) Logout.xaml – Logs out of ACME

3. Update the ACME UI Library with the new UiElelments required (from the Employee Details screen) and re-publish it

UiPath Academy

Chapter 15:

Object Repository in Studio

Importing and Using a UI Library

Resources

Reusing Objects and UI Libraries

https://docs.uipath.com/studio/standalone/2022.10/user-guide/reusing-objects-ui-libraries

Updating an UI Library

With an existing Ui Library

- Modify Element Descriptors/Names or add/remove screens/elements etc.
- Save
- Publish the library locally updating release notes

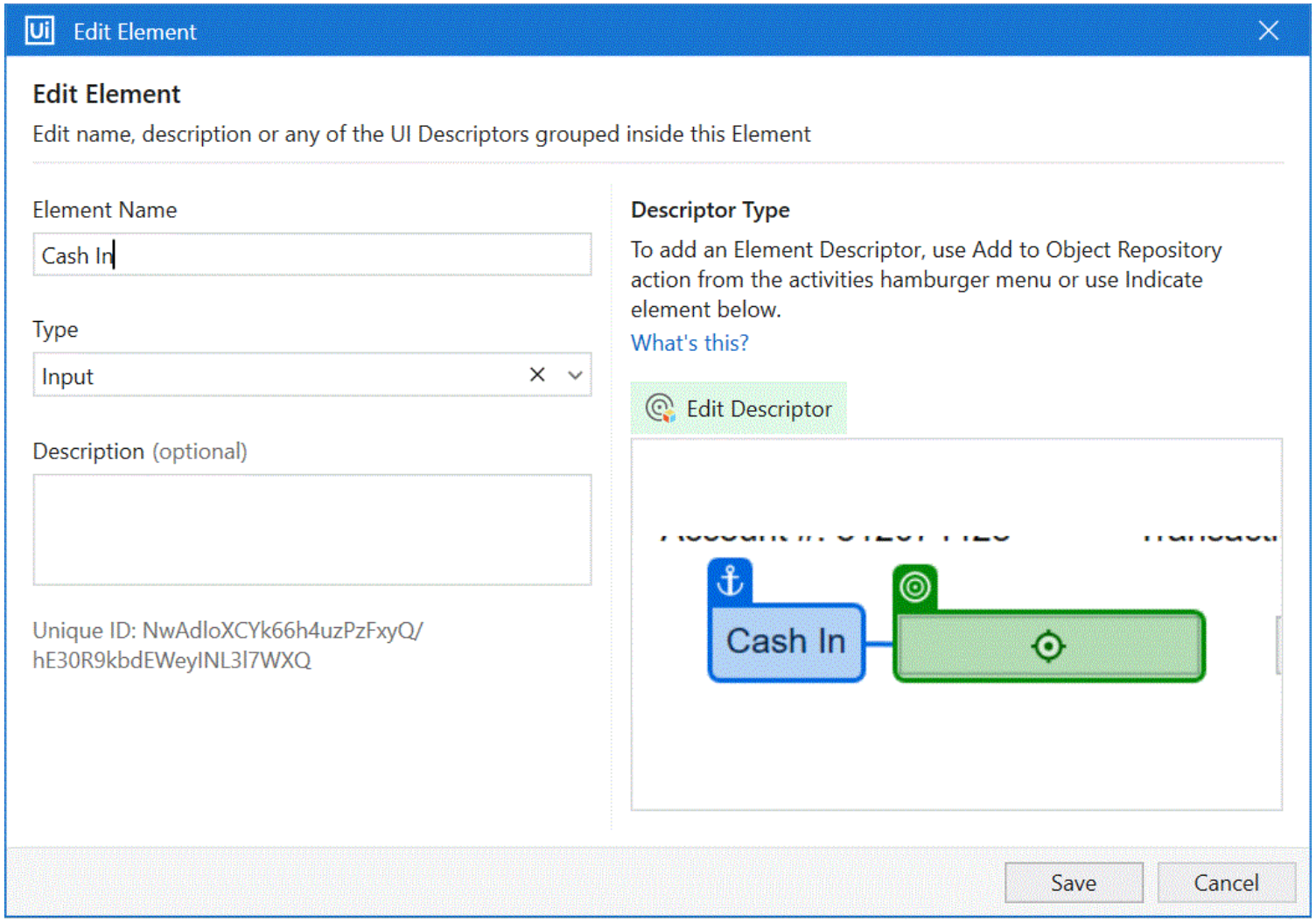
UiPath Academy

Chapter 15:

Object Repository in Studio

Updating a UI Library

Editing Descriptors in Object Repository



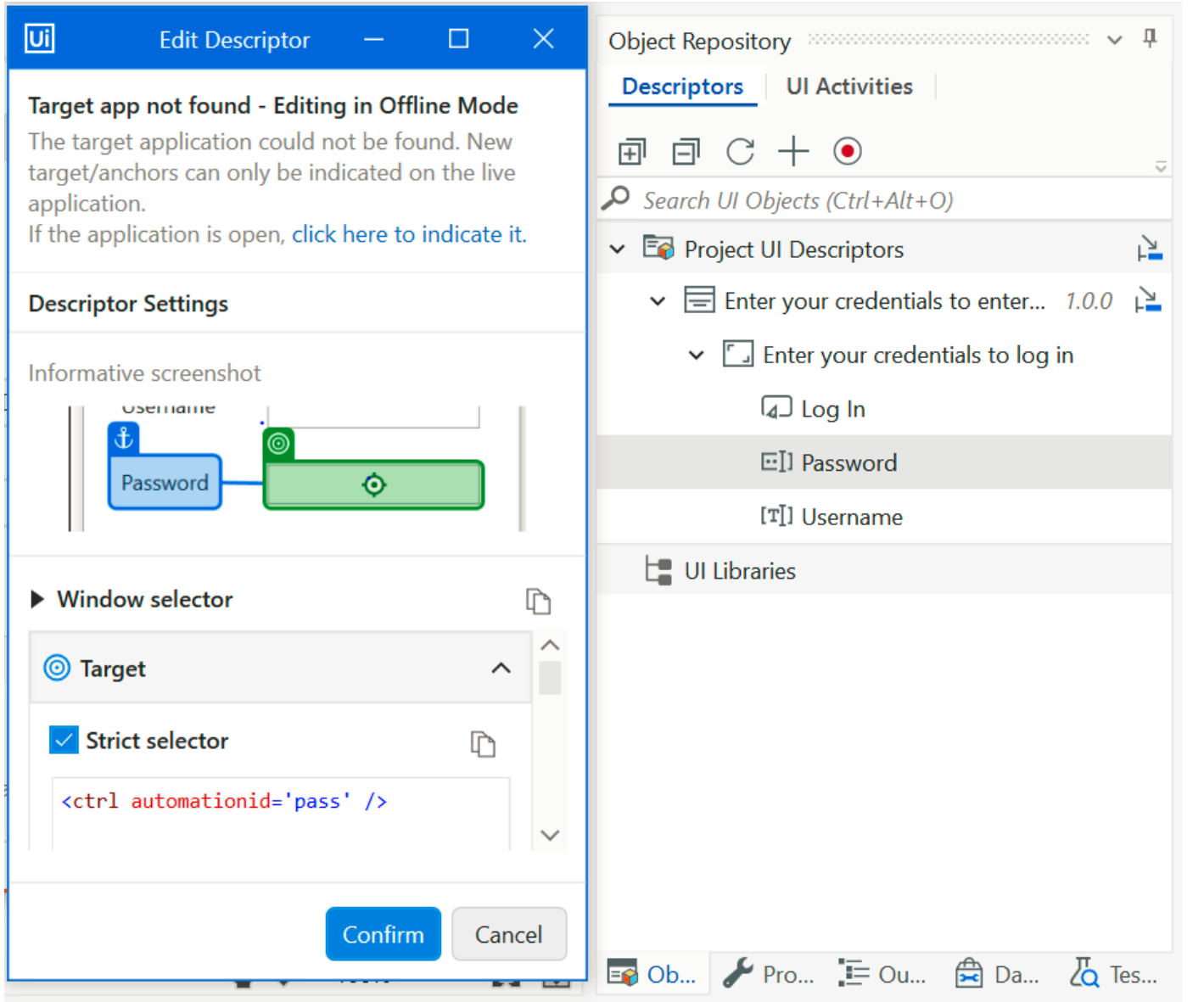
Descriptors can be edited from the Object Repository in two scenarios: Online Editing & Offline Editing

1. Online Editing

In the Descriptors tab, right-click an element and select Edit Descriptor.

The selection screen opens allowing you to capture a different element, edit selectors, and anchors.

Editing Descriptors in Object Repository

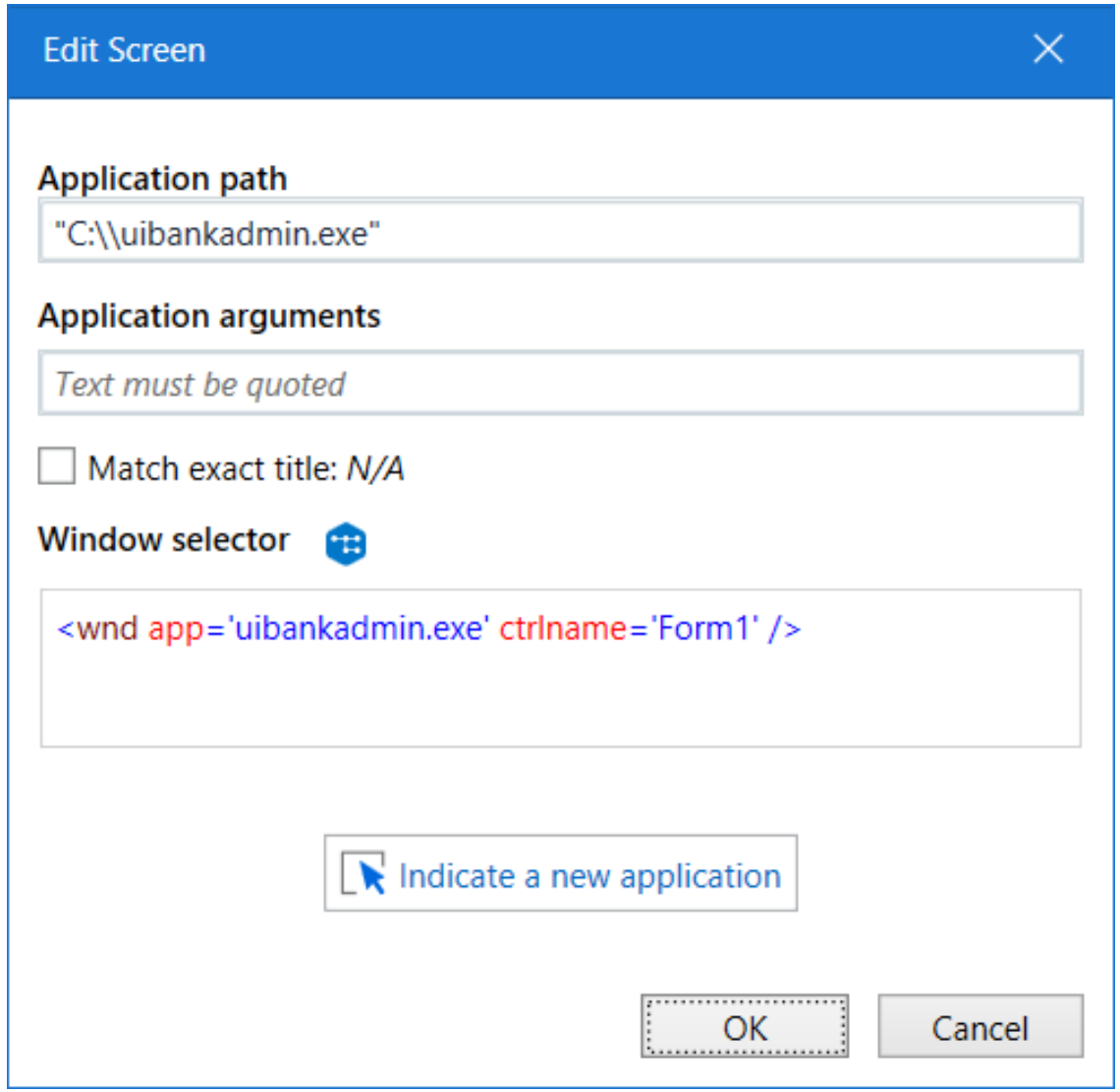


2. Offline Editing

The application or browser can not be accessed on the current machine; either due to the app not being installed, or the machine is offline.

The element's selectors are made visible allowing changes to be made.

Exploring Screen Descriptors



You can also edit a screen descriptor, in the descriptor tab.

You can edit the application path or browser URL, application arguments, window selector, and select whether only applications with a title that is an exact match can be used in the automation.

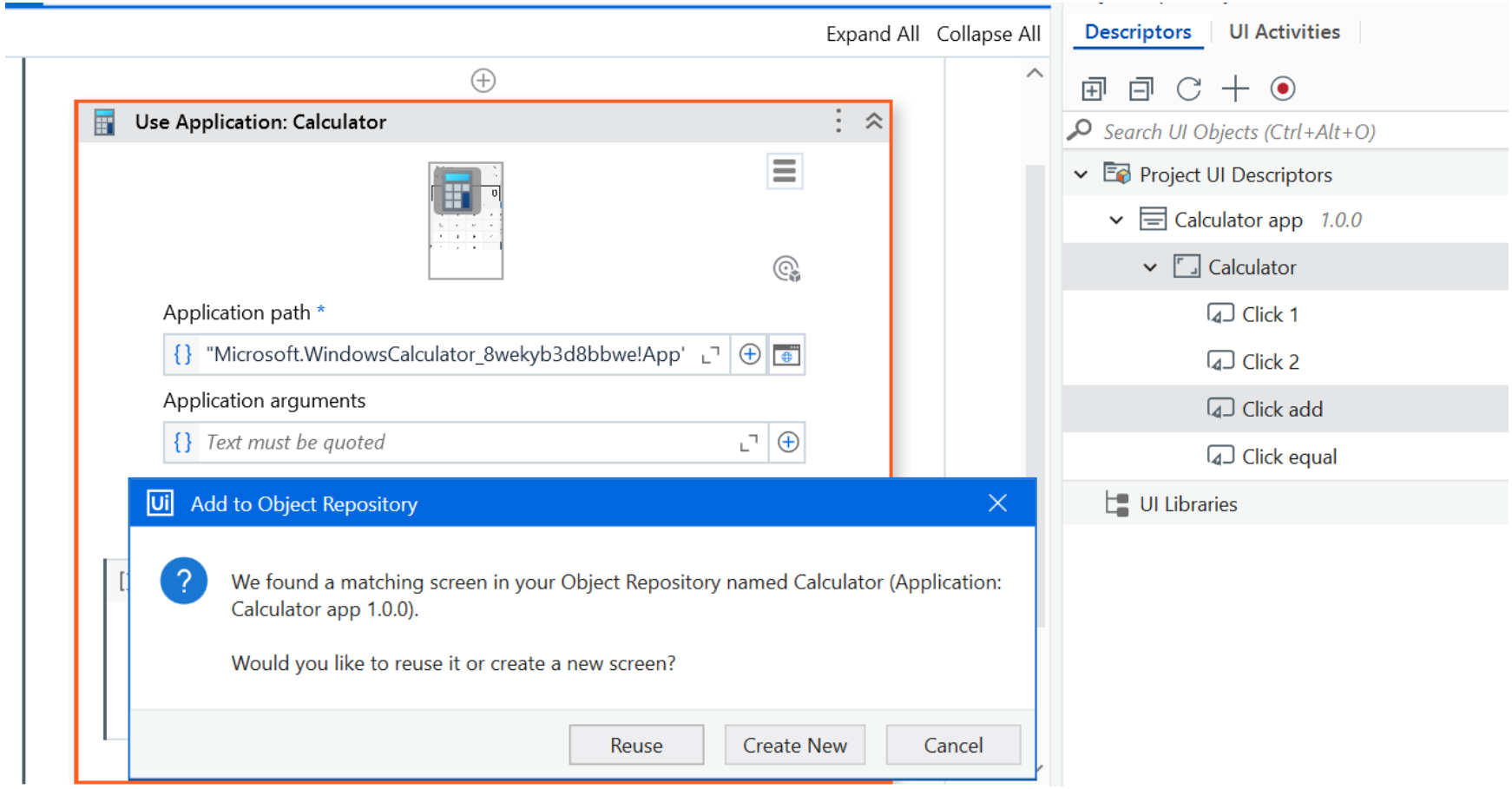
These fields support expressions, variables, and arguments.

To do this**, right-click the screen -> select Edit Descriptor**

The Edit Screen window is displayed.

Resolving UI Descriptors

Whenever you indicate a screen or UI Element which is already referenced in the Object Repository either under Project UI Descriptors or UI Libraries, the match is automatically detected and the following window is prompted:
Reuse, Create New , Cancel



Reuse
It adds the matching descriptor from the Object Repository to the current activity

Create New
It opens the options window for creating a new screen/element or updating an existing one in the Object Repository

Cancel
It keeps the currently indicated selector inside the activity, without making any changes to objects in the repository

Resolving UI Descriptors

Indicating a screen searches for a corresponding UI descriptor, while indicating a target element searches for a matching UI descriptor.

When using the Recorder, corresponding matches for screen and elements are searched within the local UI library from Project UI Descriptors or the imported dependencies from UI Libraries.

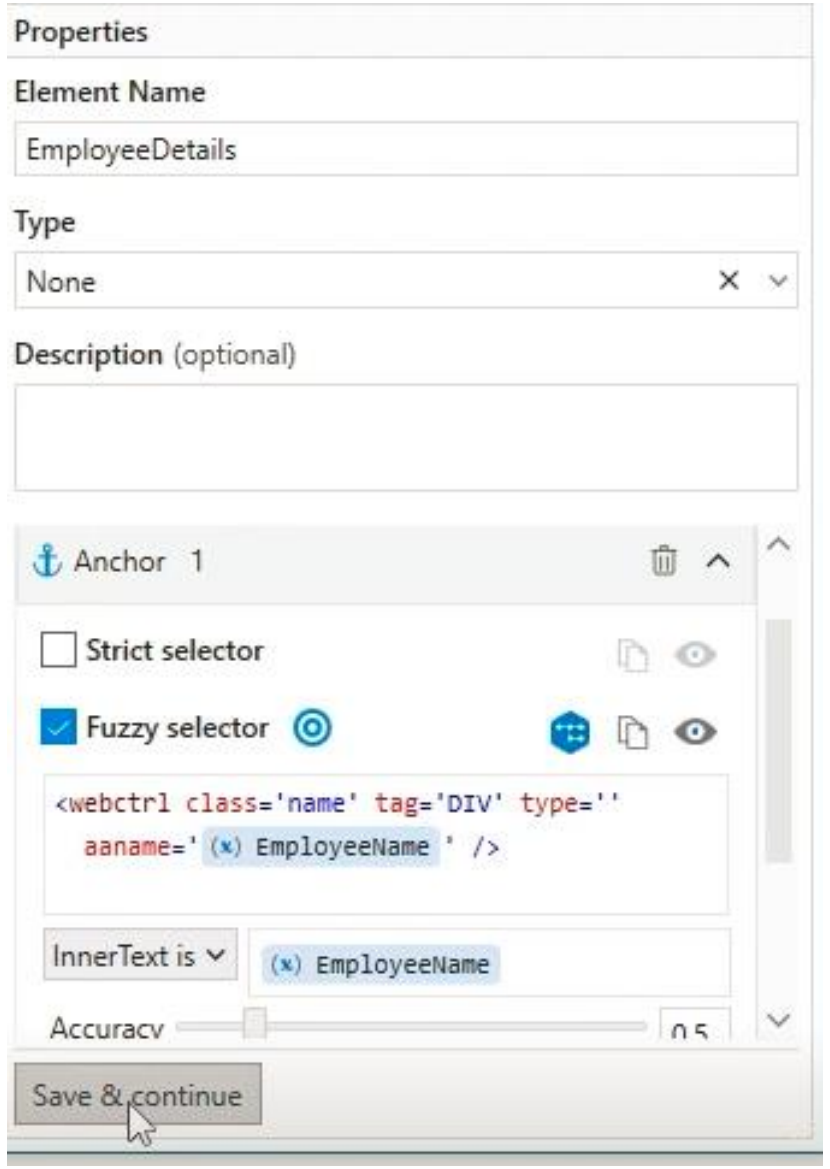
NOTE: At runtime, the UI Descriptors are resolved from the Objects Browser panel, if the activities reference such UI Descriptors.

Incorporating Variables in Descriptors

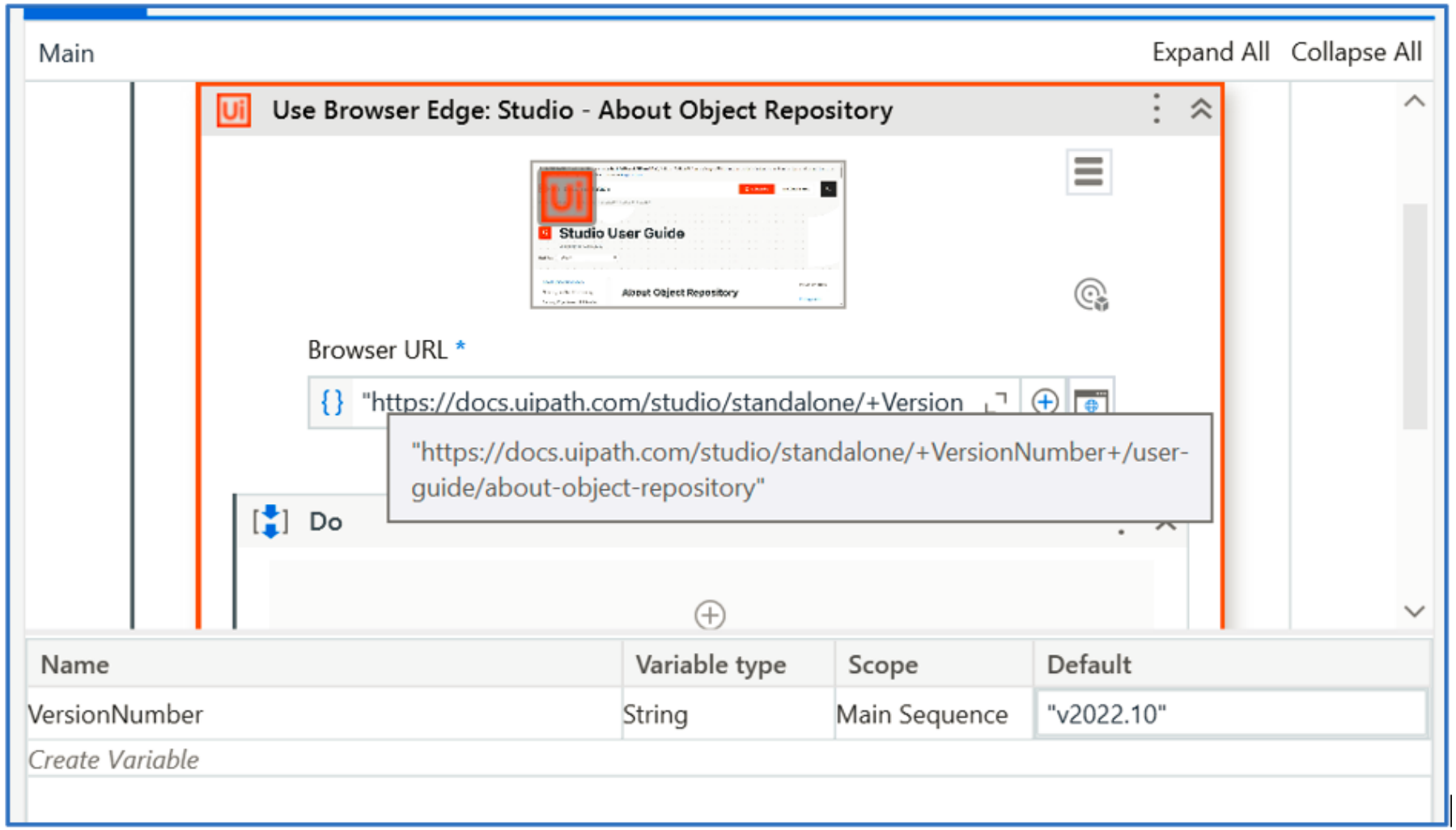
Descriptors use variables to **increase the degree of reusability**.

You can edit a target, add variables to it, and then add it to the Object Repository,
or you can add variables to selectors in existing screens or elements from the Object Repository.

Whenever you add a descriptor that contains a variable to a process, make sure to create the variable in the process, otherwise the expression cannot be read



Incorporating Variables in Descriptors



Variables can also be used inside the Browser URL field of the Use Application/Browser activity and then added the target as screen objects to the Object Repository.

As a result, you will ensure that the descriptor accounts for changing URLs.

Resources

Topic	Link
Edit Descriptor	https://docs.uipath.com/studio/standalone/2022.10/user-guide/creating-an-object-repository
Resolve UI Descriptors	https://docs.uipath.com/studio/standalone/2022.10/user-guide/creating-an-object-repository
Variables in Descriptors	https://docs.uipath.com/studio/standalone/2022.10/user-guide/creating-an-object-repository

Using Object Repository with Recorders

What Recorder is

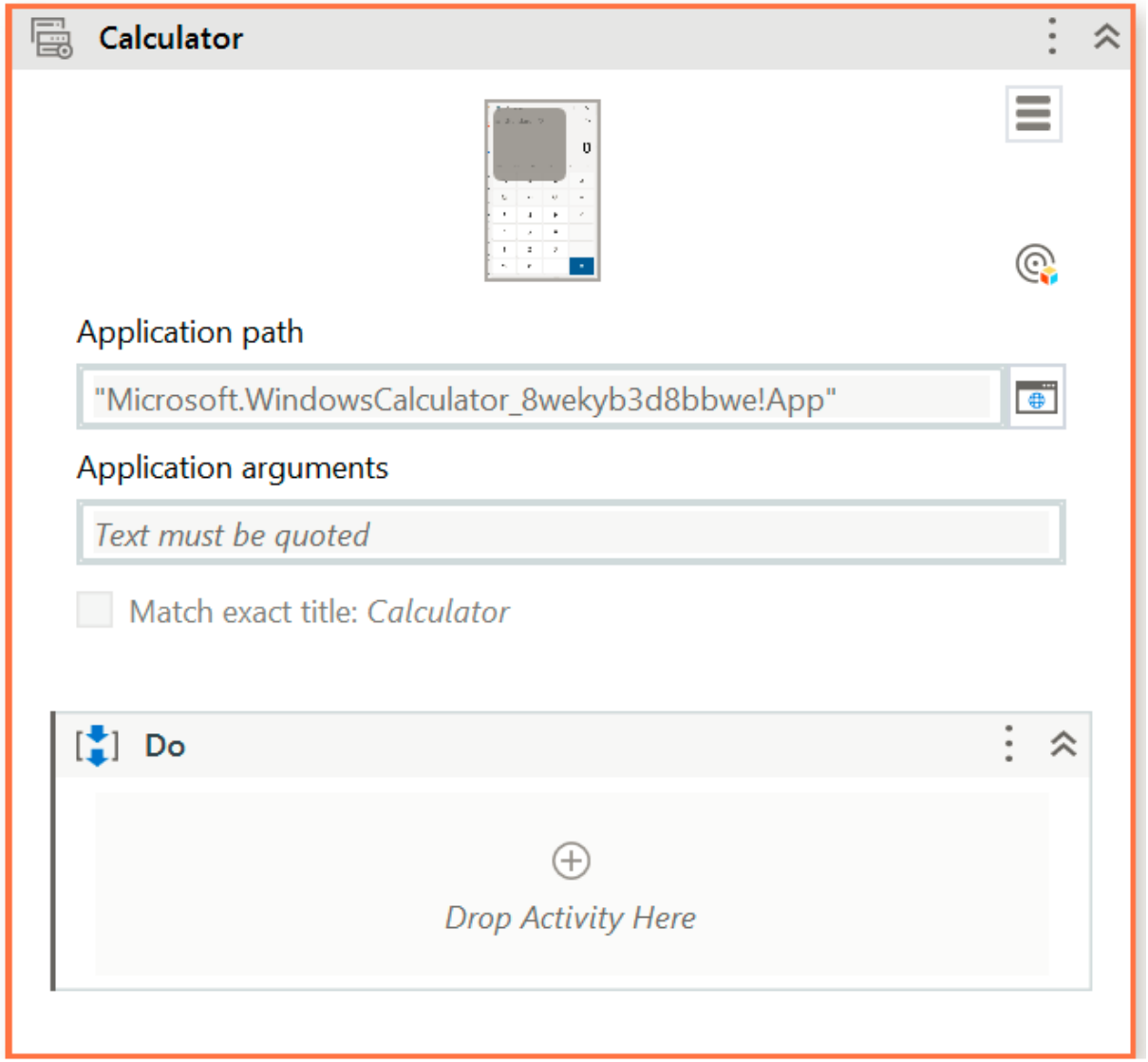
Recorder allows you to identify and record UI elements' properties and attributes that need to be included in your repository.

Creating a Repository by Capturing Elements

You can click on the Capture Elements button in the Object Repository to open the Capture Elements recorder.
For capturing elements, click Start recording.

With the Capture Elements recorder, you can capture targets, anchors, and images with a degree of accuracy using the Unified Target technology found in the **UIAutomation.Activities pack** version 20.10 or higher.

Using Object Repository with Recorders



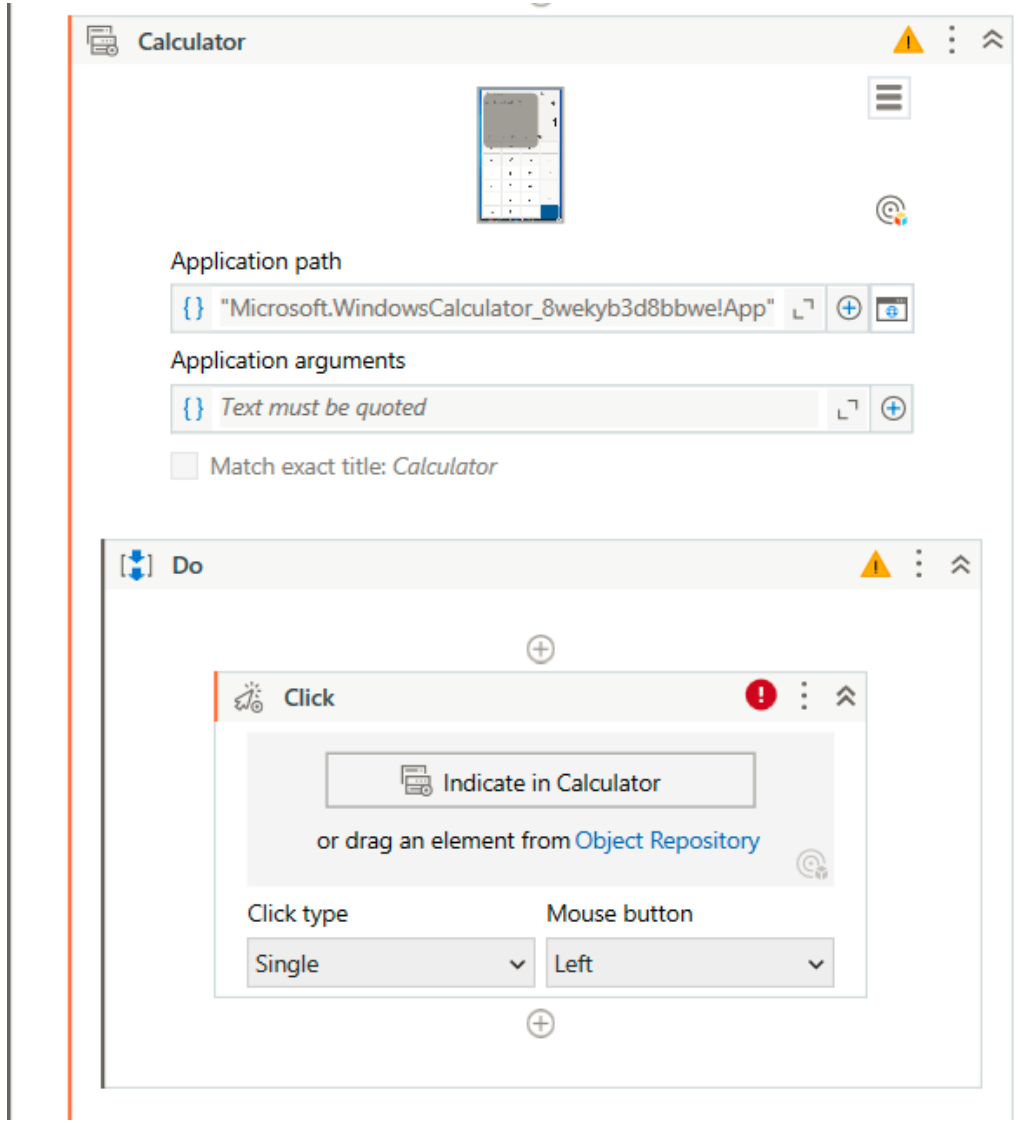
Click on the **Capture Elements button** in the Object Repository to open the Capture Elements recorder.

For capturing elements, **click Start recording**.

An informative tooltip informs you to add descriptors to the Object Repository or use existing ones.

You can add or reuse descriptors by clicking on the icon inside the activity, or you can **select Options > Add to Object Repository** to add the element.

Using Object Repository with Recorders



How to add elements From Activities to a Repository

You can add more UI elements directly from the Designer panel that supports selectors, like Click or Type Into.

Click the Object Repository icon within the activity body to open the Selection Options window. Capture the element and select Confirm.

Once the Add Element to Object Repository window opens, give the element a name and click Finish.

Using Object Repository with Recorders

Use Objects in Current Project

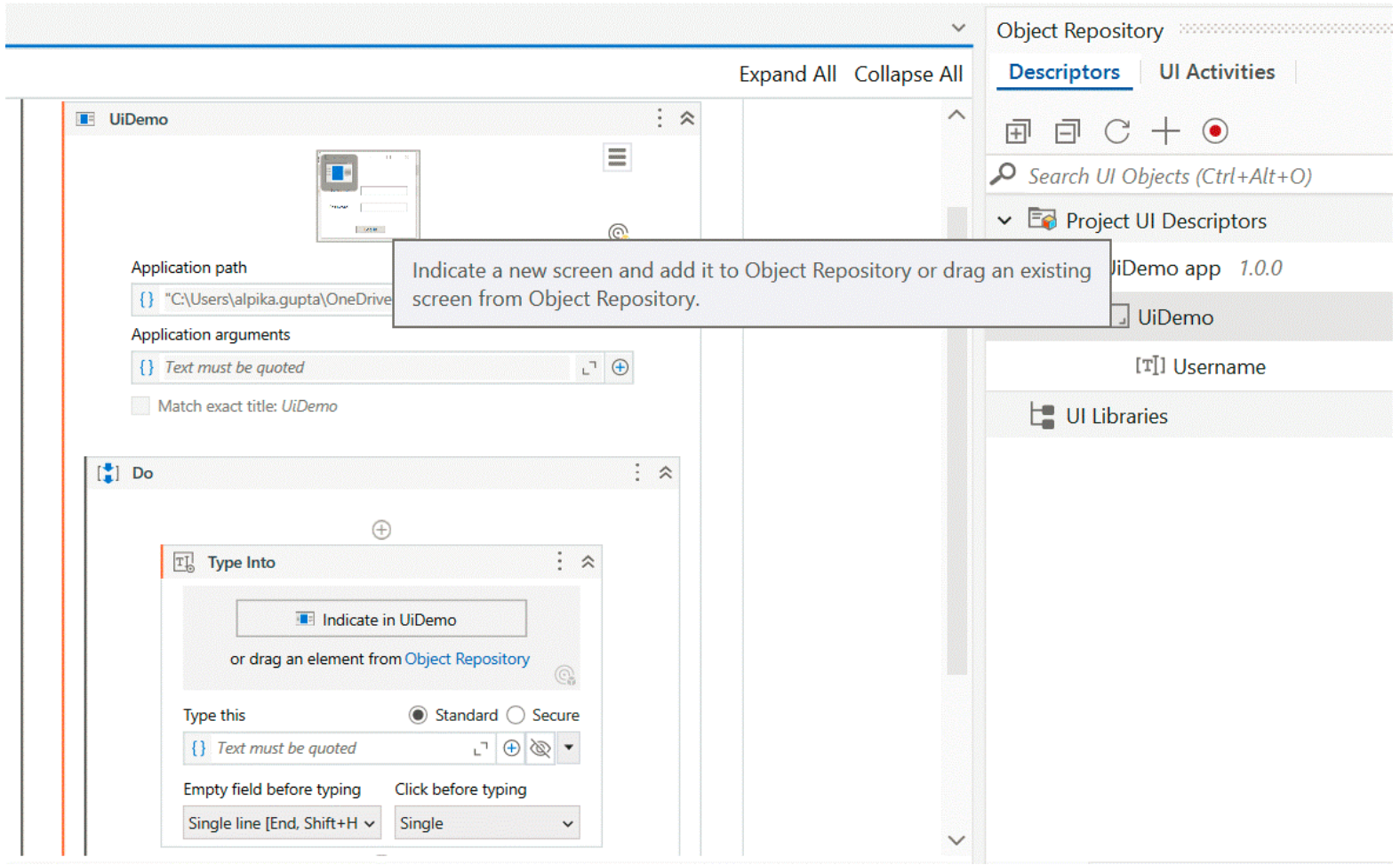
You can **drag and drop a screen or element on top of an activity in your workflow** using Object Repository.

The object's image, arguments, and other details are automatically added to the activity.

When an activity uses a screen or element from the Object Repository, clicking the Object Repository icon in the activity highlights the descriptor it uses in the Object Repository tree.

In this example, UIDemo screen is dragged on top of the Use Application/Browser activity.

The application path is automatically added to the activity.



Resources

Object Repository with Recorders

https://docs.uipath.com/studio/standalone/2022.10/user-guide/creating-an-object-repository

Extracting a UI Library

Extracting a UI Library can be very useful when you get a requirement to built multiple process for the same application and you already have descriptors you need captured in a project.

1. From a Process type project which contains targets defined in Modern UI activities, but are not in the Object Repository

2. Access the UI Activities tab in the Object Repository where we see all the targets defined in Modern UI activities in the project

3. Add elements to the Object Repository in the correct hierarchical order: Application, Screen, UI Element. To add the elements to the Object Repository, us the Add Target to Object Repository button on the UI Automation activities

4. Click **Extract as UI Library Project** in the **Object Repository Descriptors tab**

5. Publish the Library, added it as a dependency to the initial project and replace the project Descriptors with the UI Library Descriptors.

UiPath Academy

Chapter 15:

Object Repository in Studio

Extracting a UI Library

Resources

Add Elements from Activities to a Repository

https://docs.uipath.com/studio/standalone/2023.4/user-guide/creating-an-object-repository#add-elements-from-activities-to-a-repository

Object Repository Best Practices

Incorporate the following best practices while working with Object Repository:

Reusability : You can achieve reusability through different means, such as local elements, snippets, and UI libraries in the Object Repository.

Use Descriptive Names: Use meaningful and descriptive names for objects in the repository. Avoid generic names such as "Button1" and "Link2" as they can become confusing as time progresses.

Organize Objects: Organize objects within a repository by grouping them together by screen, type of object (buttons, input fields, etc.), or any other logical categorization.

Scalability: It is important to design your object repository so that it can grow with your application. As new features are added, ensure the repository can support these changes without becoming inefficient.

Version Control: When publishing a new version of the same UI library, make sure to properly add the new version number to the Publish window.

Resources

Object Repository

https://docs.uipath.com/studio/standalone/2023.4/user-guide/about-object-repository