



Possibilities ^{Unlimited}

AssistEdge RPA 20.1

Community Edition - Customization
Guide



COPYRIGHT NOTICE

©2023 EdgeVerve Systems Limited, Bangalore, India. All Rights Reserved. This documentation is the sole property of EdgeVerve Systems Limited ("EdgeVerve"). EdgeVerve believes the information in this document or page is accurate as of its publication date; such information is subject to change without notice. EdgeVerve acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. This document is not for general distribution and is meant for use solely by the person or entity that it has been specifically issued to and can be used for the sole purpose it is intended to be used for as communicated by EdgeVerve in writing.

Except as expressly permitted by EdgeVerve in writing, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording, or otherwise, without the prior written permission of EdgeVerve and/ or any named intellectual property rights holders under this document.

EDGEVERVE SYSTEMS LIMITED

Electronics City, Hosur Road, Bangalore 560 100, India.

www.edgeverve.com

AE-RPA-CE-CUSTOMIZATION GUIDE

AUTHORIZED BY: EDGEVERVE SYSTEMS LIMITED

DATE: NOV, 27, 2023.

REVISION HISTORY

Product Version	Revision Version	Revision Date	Change Description
20.1	1.0	27/11/2023	Final Version

Contents

1	Introduction	5
1.1	About this Guide	5
2	Configuration in Automation	6
2.1	Creating DLL for Custom Application.....	6
2.1.1	Custom App Launch.....	7
2.2	Code Editor	12
2.2.1	Steps to Create Code Editor DLL.....	12
2.3	Microbot.....	13
2.3.1	Steps to Create Microbot.....	13
2.3.2	Microbot.....	13
2.3.3	Microbot Input Outputs.....	14
3	Community Edition Upgrade	15
3.1	Prerequisites	15
3.2	Installation of Community Edition	15
3.3	Data Migration	22
3.4	License Expiry Notification	25

1

Introduction

The AssistEdge RPA Community Edition is a lean version of the AssistEdge RPA product with powerful automation capabilities and easy-to-use user interface. The feature-rich AssistEdge Automation Studio is a key component of the Community Edition. It enables users to create and configure process automations. The Automation Studio includes automation configuration capabilities for varied and heterogeneous technology applications.

1.1 About this Guide

This Customization Guide contains steps that enable user to configure various automations which are not configurable using the Automation Studio. It mainly focuses on configuration related to 20.x release.

2

Configuration in Automation

This section provides steps that enable the user to configure various automations that are not configurable through Automation Studio for the 20.x release.

The automations are:

- [Custom Application](#)
- [Code Editor](#)
- [Micro Bot](#)

2.1 Creating DLL for Custom Application

Follow below steps for creating DLL:

1. In Visual Studio, create new project with output type as **Class Library** using .NET framework 4.6.

Note: Make sure that DLL is built using x86 platform and DLL name is same as Namespace.

2. Add Proton.Utilities.CodePluginInterfaces, PROTON.Utilities.Interfaces, and Proton.Miscellaneous.Framework DLLs (present in EVA bot directory) references in the project.
3. Add reference to PresentationFramework and WindowsBase to use the Dispatcher to access the mainthread when required.
4. Implementing ICodeplugin interface gets following methods:
 - a. Launch – is used to launch custom applications.
 - b. DoSignIn – is used to perform sign in for custom applications.
 - c. DoAutomation – is used to perform automation in custom applications.
 - d. DoReset – is used to perform reset automation in custom applications.
 - e. DoDispose – is used to dispose custom applications.
5. Once the launch is complete:
 - a. Raise the *LaunchCompleted* event.
 - b. Pass the code plugin object as sender.
 - c. Set the application object in the *appContext InstanceObject*, set *IsLaunched* as True and pass the *appContext* object in the *LaunchEventArgs* in the *LaunchCompleted* event.

Note: Raise all the events in main thread, i.e. using Dispatcher.

2.1.1 Custom App Launch

```

public ApplicationContextForAutomation Launch()
{

//Code to launch the custom application goes here

//Events for handling form size change in SE this.SizeChanged -= MyCustomApp_SizeChanged;
this.SizeChanged += MyCustomApp_SizeChanged;

//Set application context

applicationContextFormAutomation = new ApplicationContextForAutomation();
applicationContextFormAutomation.InstanceName = "MyCustomApp";

//Assign your application instance object in the context
applicationContextFormAutomation.InstanceObject = objIE;
applicationContextFormAutomation.InstanceType = objIE.GetType().ToString(); LaunchEventArgs
launchEventArgs = new LaunchEventArgs(); launchEventArgs.appContext =
applicationContextFormAutomation;

//Code to dock the application in SE
System.Windows.Application.Current.Dispatcher.BeginInvoke((Action)delegate ()
{

if (SEFramework.UseDefaultDockBehaviour)

{

AddNewApplicationTabUIWithObject("MyCustomApp", this.Handle, this);
Proton.Miscellaneous.Framework.Win32.SetParent(childHandle, this.Handle);
Proton.Miscellaneous.Framework.Win32.ShowWindow((IntPtr)childHandle, 3);this.Show();

}

}

```

6. In **SizeChanged** event generated as shown below, set height and width of the application to be equal to that of the *usercontrol*.

Custom App SizeChanged:

```
private void MyCustomApp_SizeChanged(object sender, EventArgs e)

{

    if (objIE != null)

    {
```

7. Once *LaunchCompleted* event is raised:
 - a. EVA bot calls the *DoSignIn* method of the code plugin if *SignIn* is configured for the Custom Application.
 - b. Raise the *SignInCompleted* event once the sign in is done with *SignInEventArgs* having either *SignInStatus* as Completed or Failed using *SignInStatus* Enumeration.

If this method takes more time than the value configured in the sign-in manager in IDE for the code plugin, then EVA bot marks the sign-in as failed for this custom application. User gets User Id and password in *inputParamDict*.

Custom App *DoSignIn*:

```

public bool DoSignIn(Dictionary<string, string> inputParamDict, ref Dictionary<string, string>
outPutParamDict, ApplicationContextForAutomation appContext)

{

//Code to perform sign in into your application goes here

//On successful sign in, raise the SignInCompleted event SignInEventArgs signInEventArgs = new
SignInEventArgs(); signInEventArgs.SignInStatus = SignInStatus.SignInCompleted;

System.Windows.Application.Current.Dispatcher.BeginInvoke((Action)delegate ()

{

SignInCompleted(this, signInEventArgs);

```

8. Application automation logic is to be implemented in *DoAutomation* method.
 - *InputParamDict* contains all inputs parameters configured for this custom application in format of *KeyValuePair*, where keys correspond to field name and value corresponds to its value.
 - *OutPutParamDict* contains *KeyValuePair* of all the output parameters, where keys correspond to field name and value corresponds to its value which is required to be modified or assigned by user depending on the output from custom application.
 - *To return the extracted fields to EVA bot*: Raise *FillExtractedFieldsEvent* with *ExtactedFieldsEventArg* having dictionary containing *extractedFields* assigned equal to *outputPutParamDict*.

Note: Ensure this event is raised using Dispatcher.

9. To mark search completed or failed, raise *SearchStatusEvent* with *SearchStatusEventArgs* having either *searchCompleted* or *SearchFailed* as enumeration in *eventArgs*.

Custom App DoAutomation:

```

public Dictionary<string, string> DoAutomation(Dictionary<string, string> inputParamDict, ref
Dictionary<string, string> outputParamDict, ApplicationContextForAutomation appContext)

{

//Code to automate the application goes here; inputParamDict will have the input data configured for
automation

//Raise event for filling extracted fields

ExtractedFieldsEventArgs extractedFieldEventArgs = new ExtractedFieldsEventArgs();
extractedFieldEventArgs.ExtractedFields = new Dictionary<string, string>();
extractedFieldEventArgs.ExtractedFields.Add("Key from outputParamDict", "Extracted value for the
key");System.Windows.Application.Current.Dispatcher.BeginInvoke((Action)delegate ()

{

FillExtractedFieldsEvent(this, extractedFieldEventArgs);

});

//Raise SearchStatusEvent when search completed with appropriate status SearchStatusEventArgs
searchStatusEventArgs = new SearchStatusEventArgs(); searchStatusEventArgs.SearchStatus =
SearchStatus.SearchCompleted:

```

10. Custom Application is reset whenever the user resets the process in EVA bot, to achieve that need to implement *DoReset* method. After reset, user has to raise *SearchStatusEvent* as *ResetCompleted* or *ResetFailed* using *SearchStatus* Enumeration.

Note: Ensure that this event is to be raised using Dispatcher as shown below.

Custom App *DoReset*:

```
public void DoReset(string applicationName, Dictionary<string, object> inputParamDict)
{

//Code to reset the application goes here. This method gets called on Reset button click in SE

//Raise SearchStatusEvent on successful reset

SearchStatusEventArgs searchStatusEventArgs = new SearchStatusEventArgs();
searchStatusEventArgs.SearchStatus = SearchStatus.ResetCompleted;
System.Windows.Application.Current.Dispatcher.BeginInvoke((Action)delegate ()
{
```

11. Custom Application has to be reloaded whenever user reloads the applications in EVA bot, to achieve that user needs to implement *DoDispose* method. Whenever Reload is clicked, *DoDispose* method is called first and then launched.

Note: *DoDispose* method is used to dispose the customization and reset the application.

Custom App *DoDispose*:

```
public void DoDispose(string applicationName, Dictionary<string, object> inputParamDict)

{
```

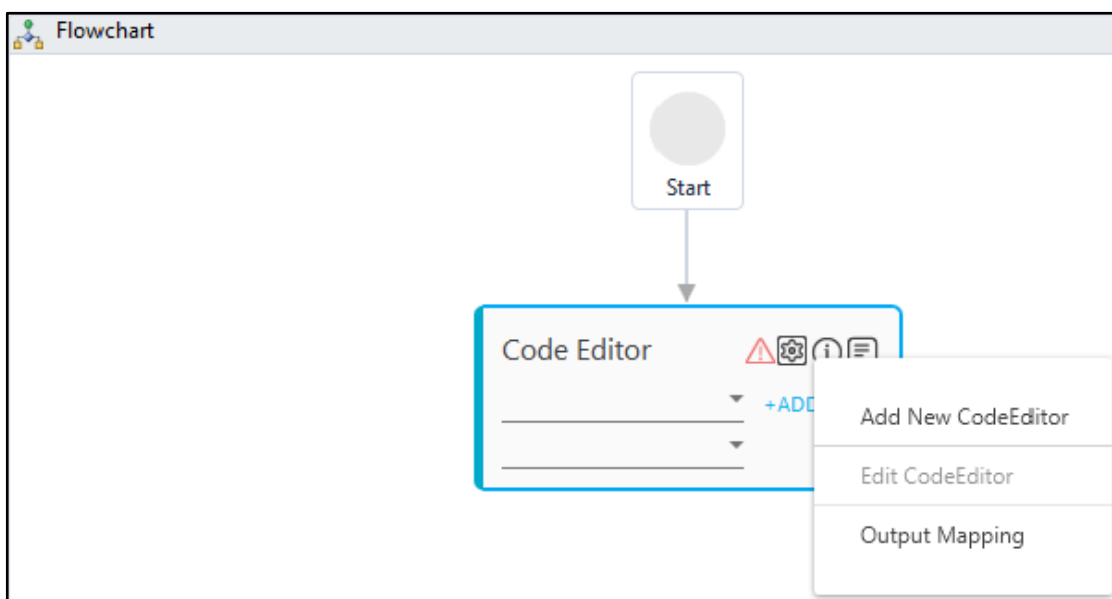
Note: All custom plugin DLLs must be built using the supported .NET 4.6 framework.

2.2 Code Editor

The Code Editor helps the user to write custom code for a particular step in the automation. If the user is unable to automate a particular step using Automation Studio, add the Code Editor for that particular step and write custom code.

2.2.1 Steps to Create Code Editor DLL

1. In Automation Studio, search for **Code Editor** activity in canvas tool.
2. Drag and drop the Code Editor activity in the Flowchart designer area.
3. Click the **Settings** icon, and then click **Add New CodeEditor**.



The Script Editor window appears. Write your customization code. You can also add reference to any thirdparty dll from the reference window.



4. Once you are done, click **Compile**.
5. Once compilation is completed, click **Compile and Upload**.

Note: Click **Settings**, and then click **Output Mapping** to define the parameters and map the same.

2.3 Microbot

Microbot enables the user to write code for reusable components. This is a feature to extend the product boundary by creating reusable components which are inserted anywhere in the process to perform custom logic.

2.3.1 Steps to Create Microbot

1. In Visual Studio, create a new project with output type as Class Library using .NET framework 4.6.
2. Add SE.Core.Automation.Interfaces DLL and SE.Core.Automation.Models DLL references in the project.
3. Implement IAECODEPackage using SE.Core.Automation.Interfaces.
4. Once IAECODEPackage is implemented, the user gets execute method and execution completed event.

2.3.2 Microbot

```
public class MyMicrobot : IAECODEPackage
{

    public event EventHandler ExecutionCompleted; public void Execute(StudioContext context)
    {

        //Microbot code goes here
    }
}
```

1. To map input and output parameters, custom attributes are defined to be set on .NET class properties to set or get values from output and input of micro-bot.
2. Once execution is complete, the user must raise the execution complete event.

2.3.3 Microbot Input Outputs

```
[ArgumentDirection(Direction = DirectionType.Input)]public int Input1
{
    get; set;
}

[ArgumentDirection(Direction = DirectionType.Output)]public int Output1
{
    get;set;
}
```

Note: Ensure each microbot has a distinct main dll.

3 Community Edition Upgrade

The Community Edition Upgrade functionality helps you migrate your data from AssistEdge Community Edition 18.0 to AssistEdge Community Edition 20.1. As a AssistEdge Community Edition user, you can now upgrade to AssistEdge Community Edition 20.1, the latest offering, to explore more advanced features of Automation Studio. You can also view your license expiry notification.

The following activities are involved for the Community Edition upgrade:

- [Prerequisites](#)
- [Installation of Community Edition](#)
- [Data Migration](#)
- [License Expiry Notification](#)

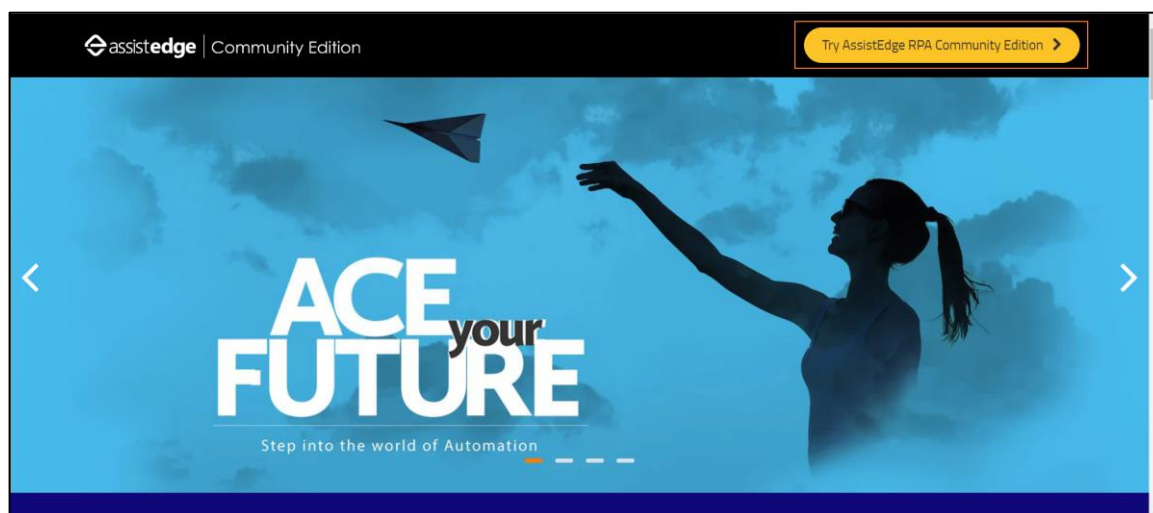
3.1 Prerequisites

- Node.js installation is recommended on the system before installation of Community Edition.

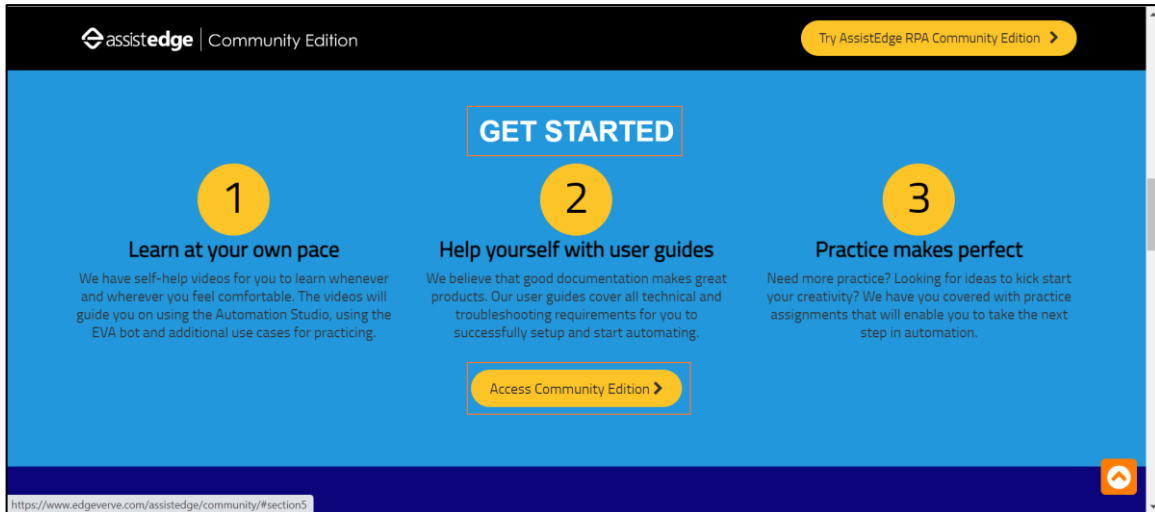
3.2 Installation of Community Edition

To install and upgrade from AssistEdge Community Edition 18.0 to AssistEdge Community Edition 20.1:

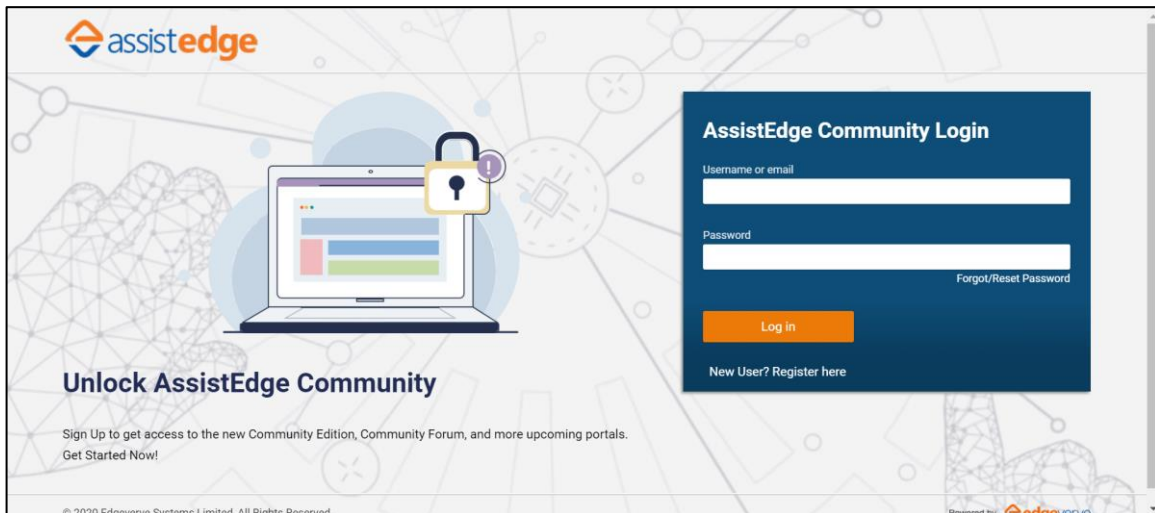
1. Go to [AssistEdge Community Edition](#) website and click **Try AssistEdge RPA Community Edition**.



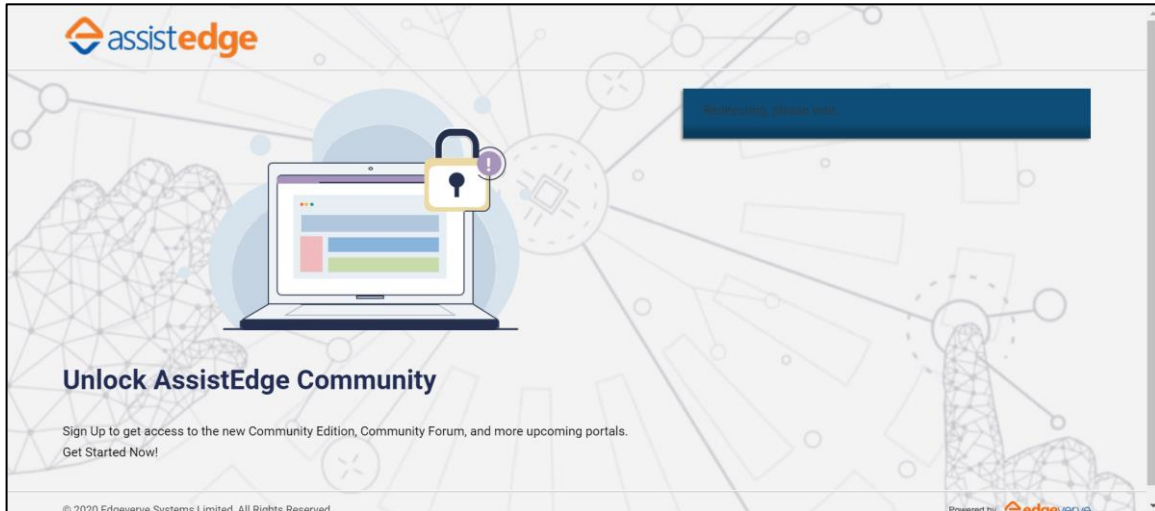
- In the **Get Started** area, click **Access Community Edition**. A new browser window opens.



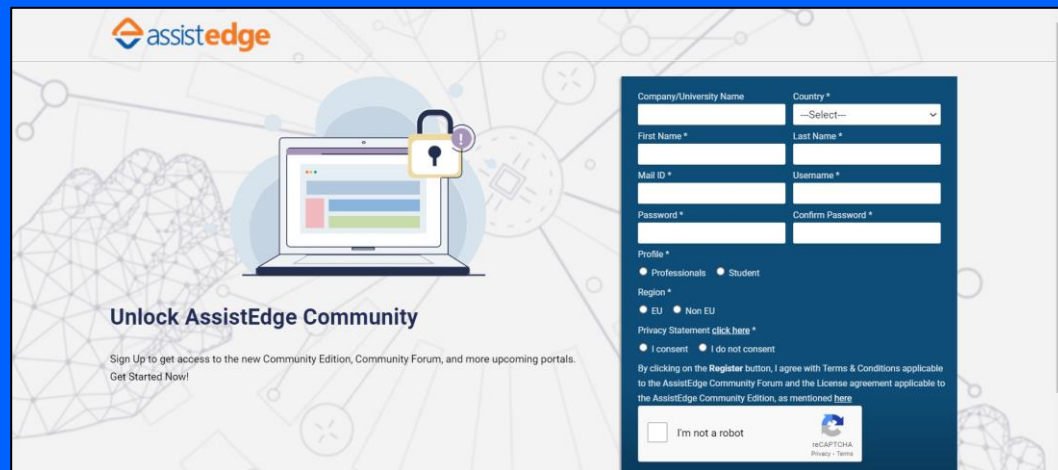
- In **AssistEdge Community Login**, provide your registered **Username or email** and **Password**, and then click **Log in**.



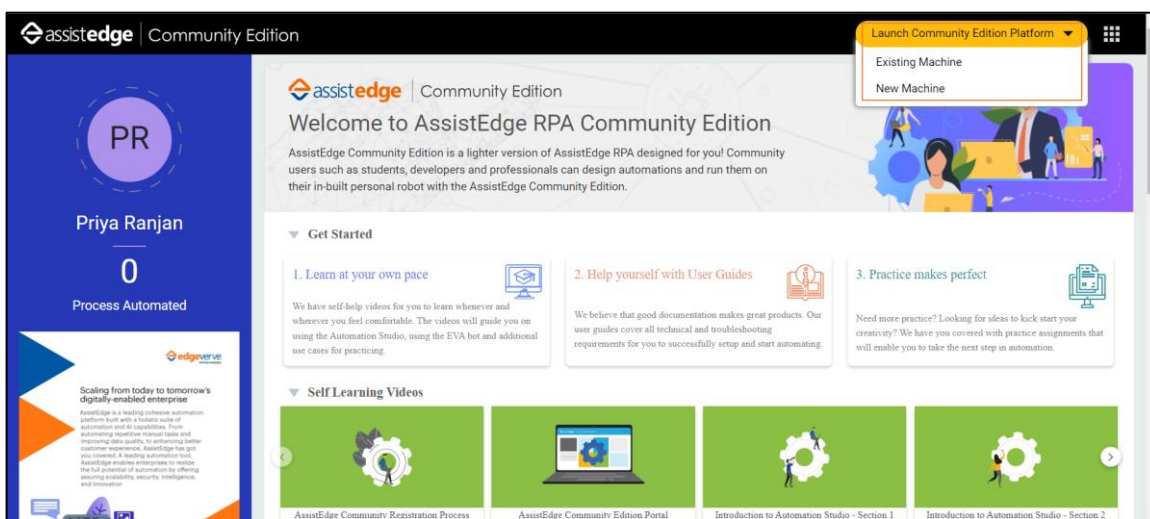
You are redirected to the **AssistEdge Community** page.



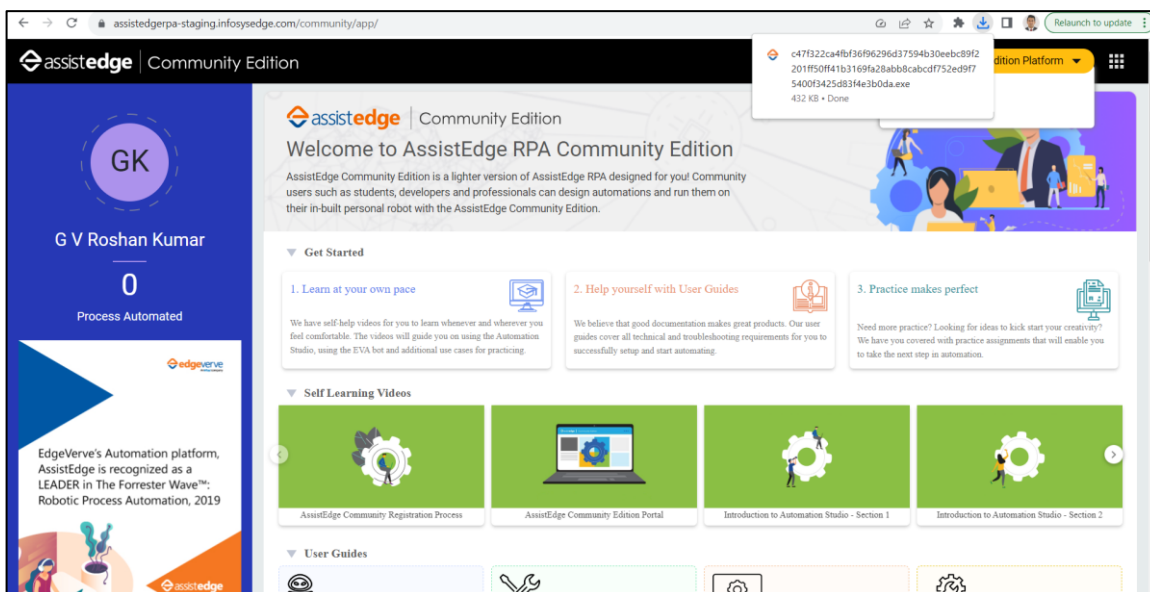
Note: If you are a first-time user, click **New User? Register here** and provide appropriate details to register for the AssistEdge Community Login.



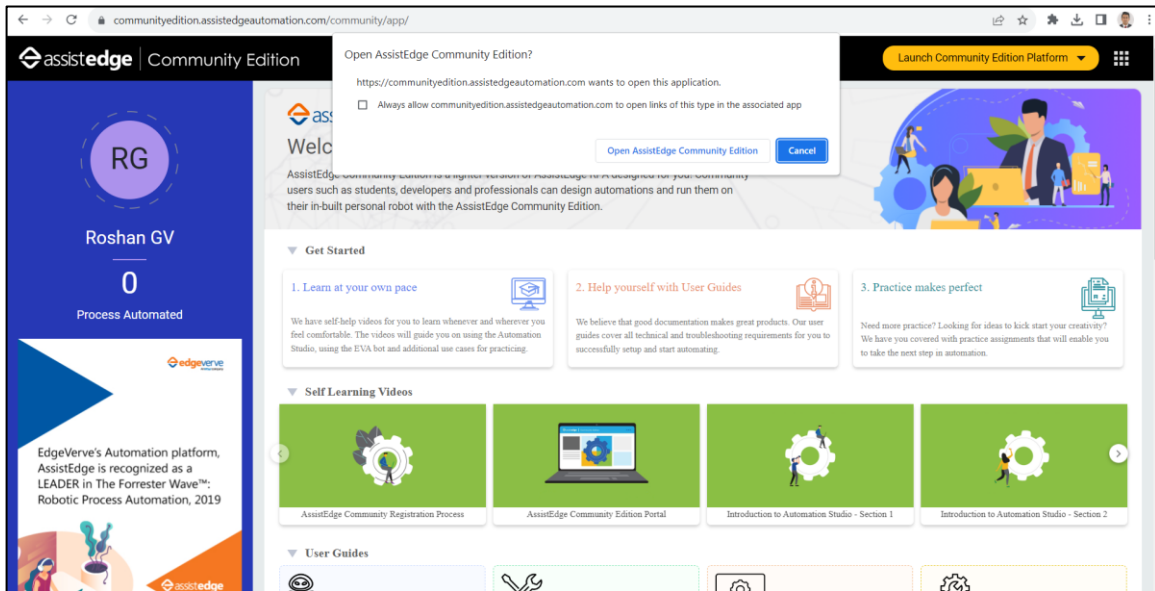
- In the **AssistEdge Community** page, click **Launch Community Edition Platform** to launch the latest version of AssistEdge Community Edition.



- If you are the first-time user of AssistEdge Community Edition 20.1, click **New Machine** from the dropdown list. The Community Edition setup gets downloaded.



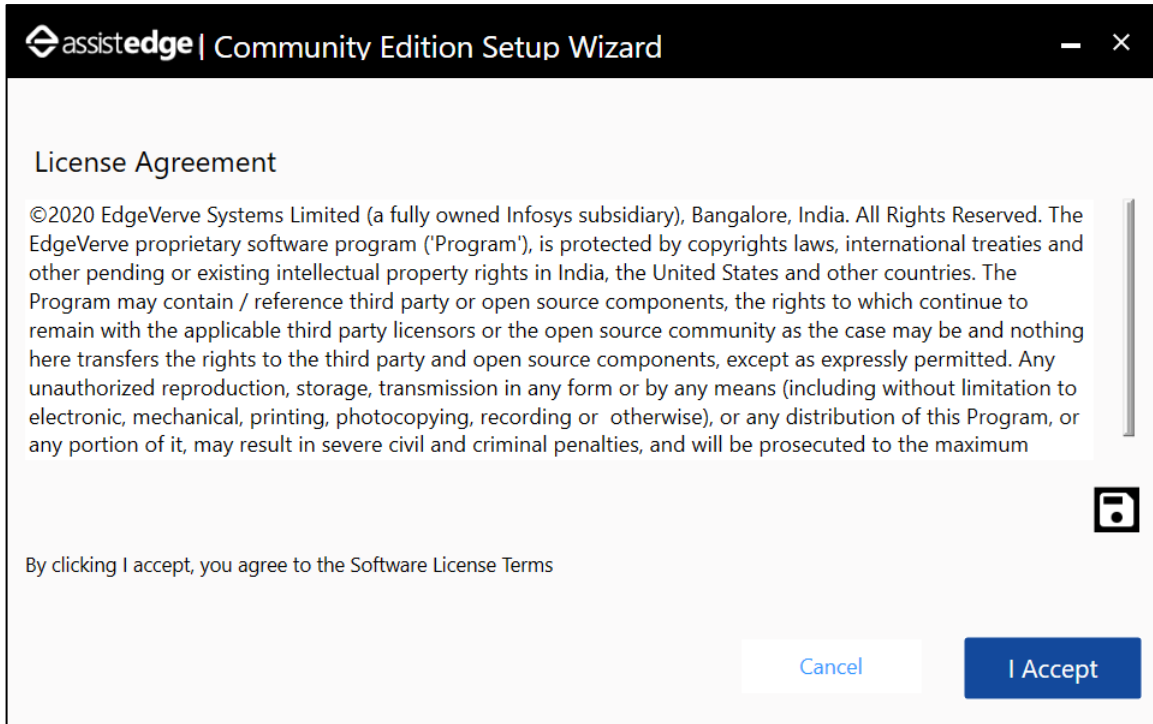
- If you are the second-time user of AssistEdge Community Edition 20.1, click **Existing Machine**. A confirmation message appears. Click **Open AssistEdge Community Edition** to launch the Community Edition.



- Click the downloaded setup to download the **AssistEdge Community Edition**. The download progress is displayed.

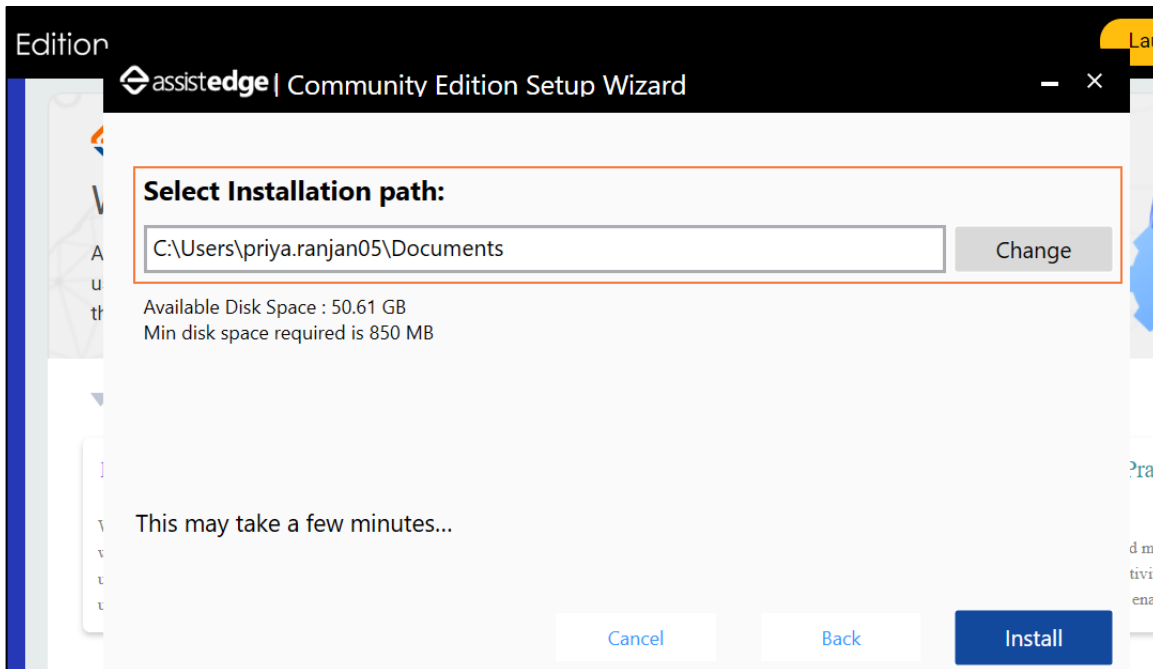


6. In Community Edition Setup Wizard, click **I Accept**.



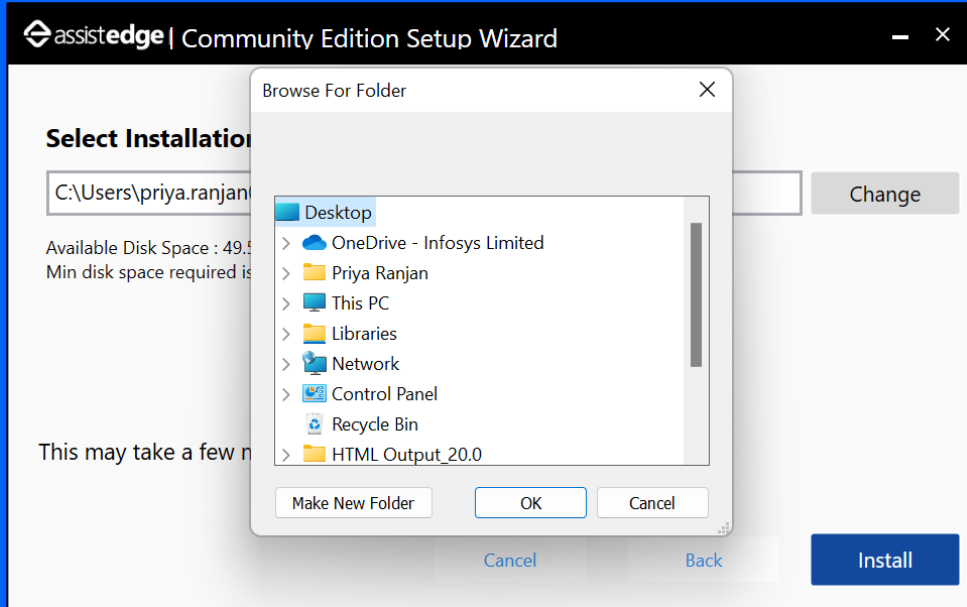
7. To install the AssistEdge Community Edition, select the appropriate installation path and click **Install**.

Note: By default, **C:\Users*<user name>*\Documents** is selected as the installation path. Click **Change** to browse for appropriate folder and change the installation path.

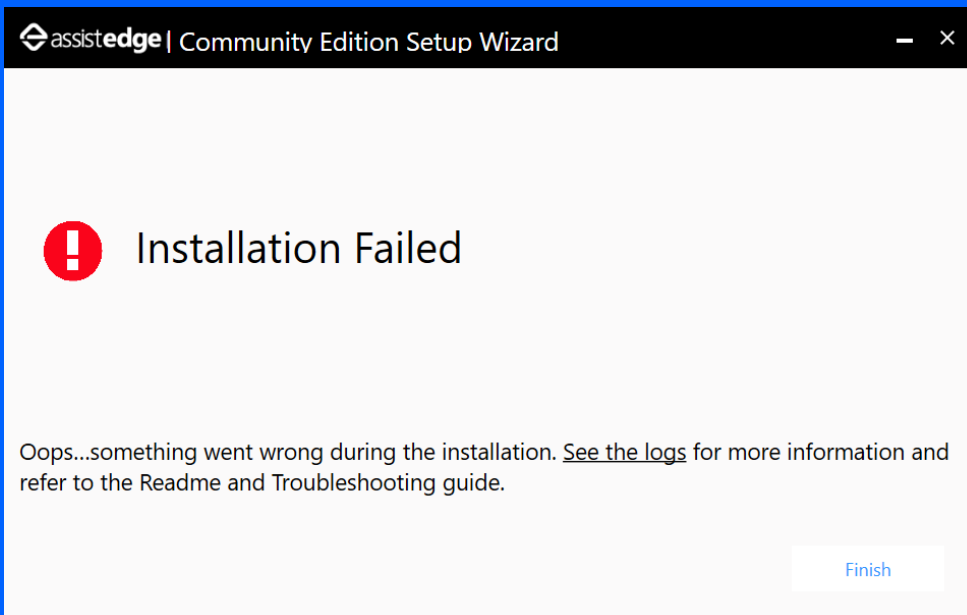


Note: If you do not want to install the Community Edition now, click **Cancel**. If you want to go to previous screen, click **Back**.

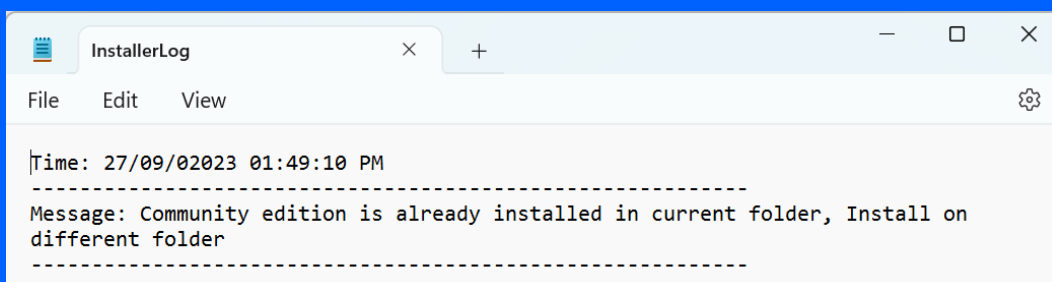
Note: If you are the second-time user of Community Edition, ensure to select an installation path different than the previous Community Edition installed on your machine. Click **Change** to browse for appropriate folder and change the installation path.



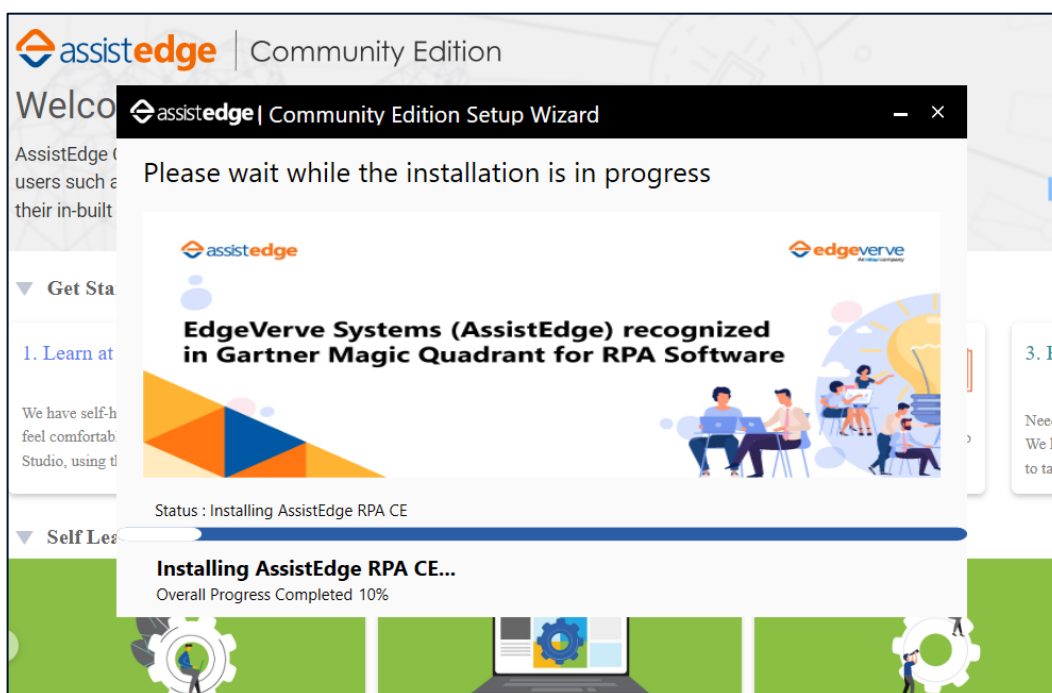
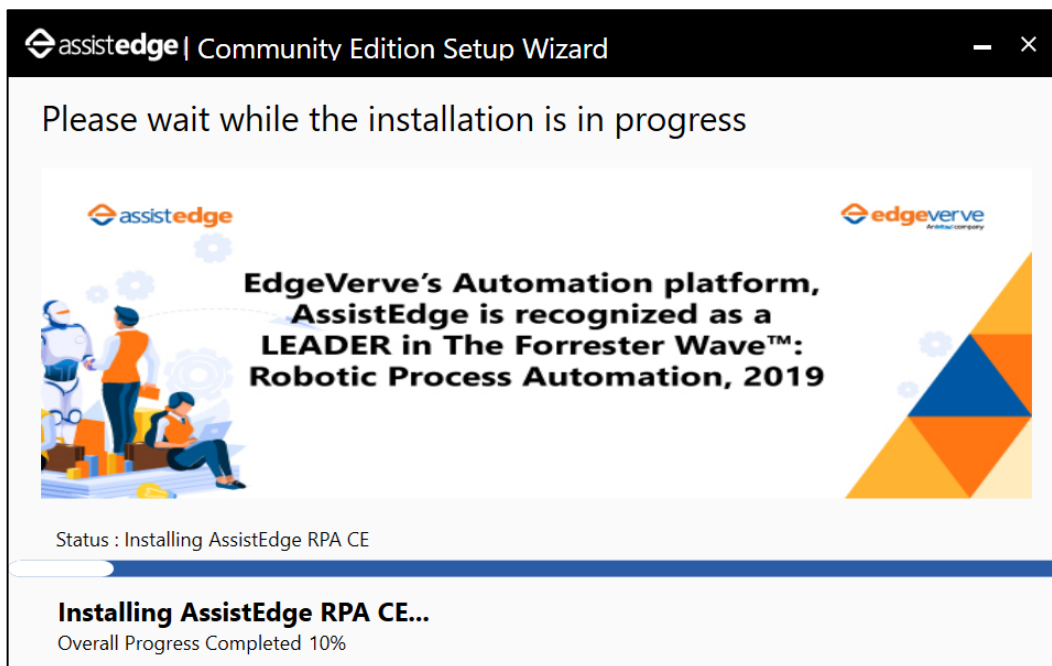
If the Community Edition is already existing in the selected installation path, the following message is displayed.



Click **See the logs** for more information.



An information message regarding the installation progress is displayed.



Note: For uninstallation of Community Edition, ensure to exit all the running applications from the system tray, if any.

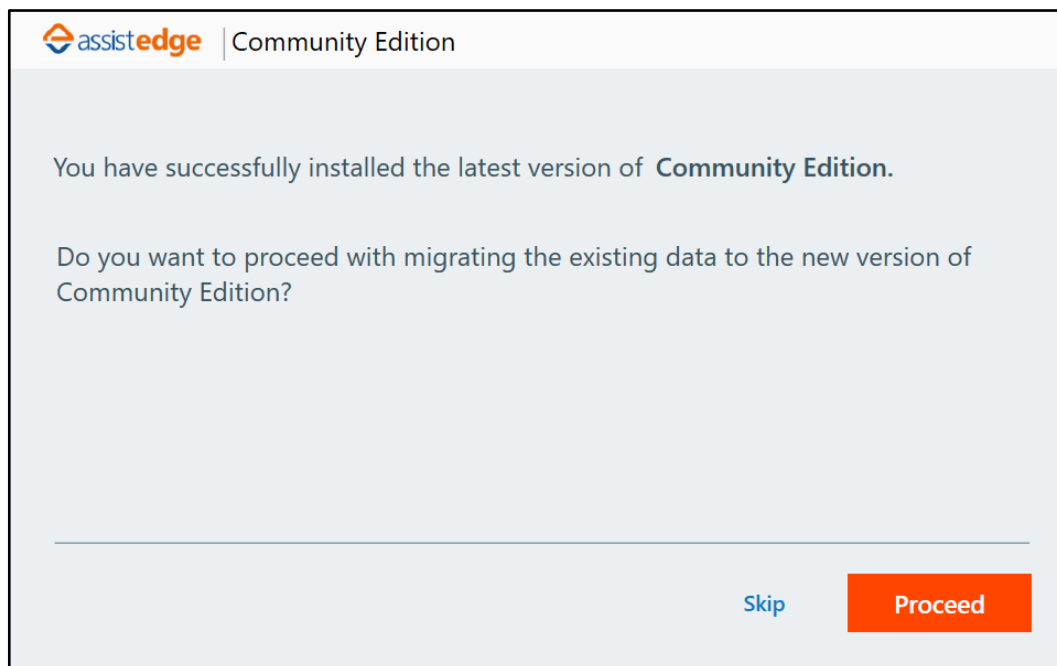
3.3 Data Migration

To complete data migration:

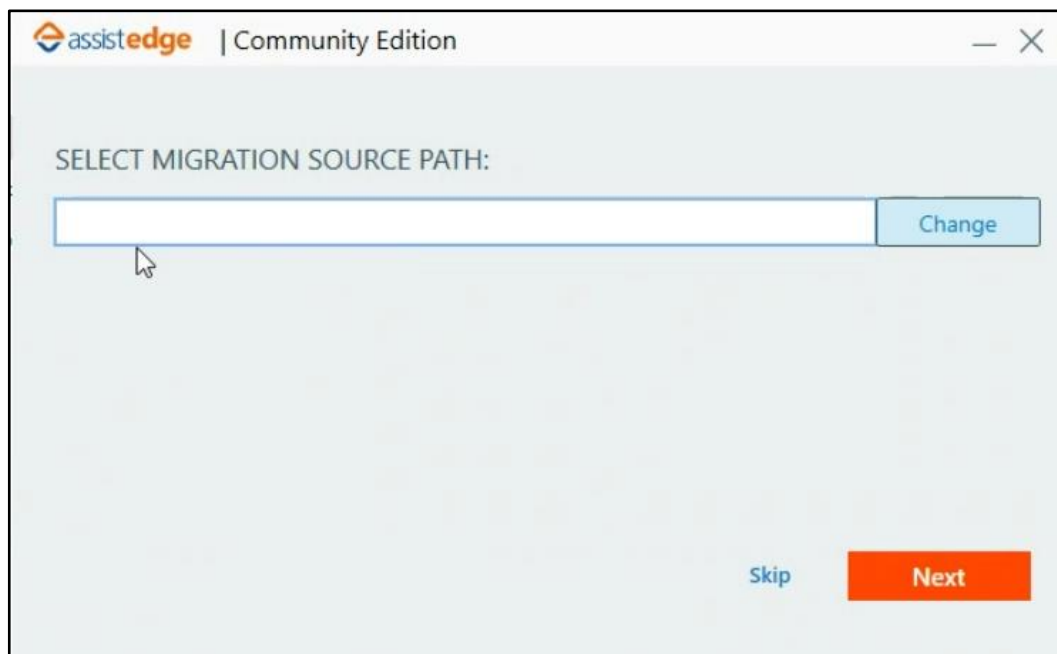
1. Ensure the installation is completed successfully (see [Installation for Community Edition](#)).
2. Click **Proceed** to migrate your entire AssistEdge Community Edition 18.0 data into the new version, that is,

AssistEdge Community Edition 20.1.

OR, Click **Skip** if you want to skip the data migration process.

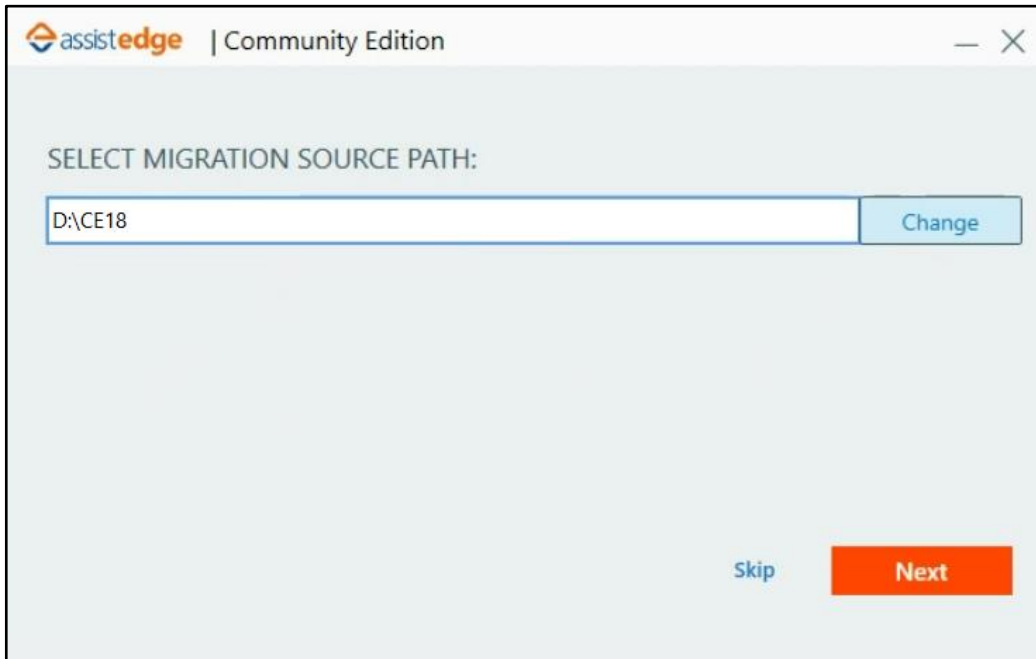


3. Click **Change** to select the migration source path of the previous version of AssistEdge Community Edition installed on your machine.

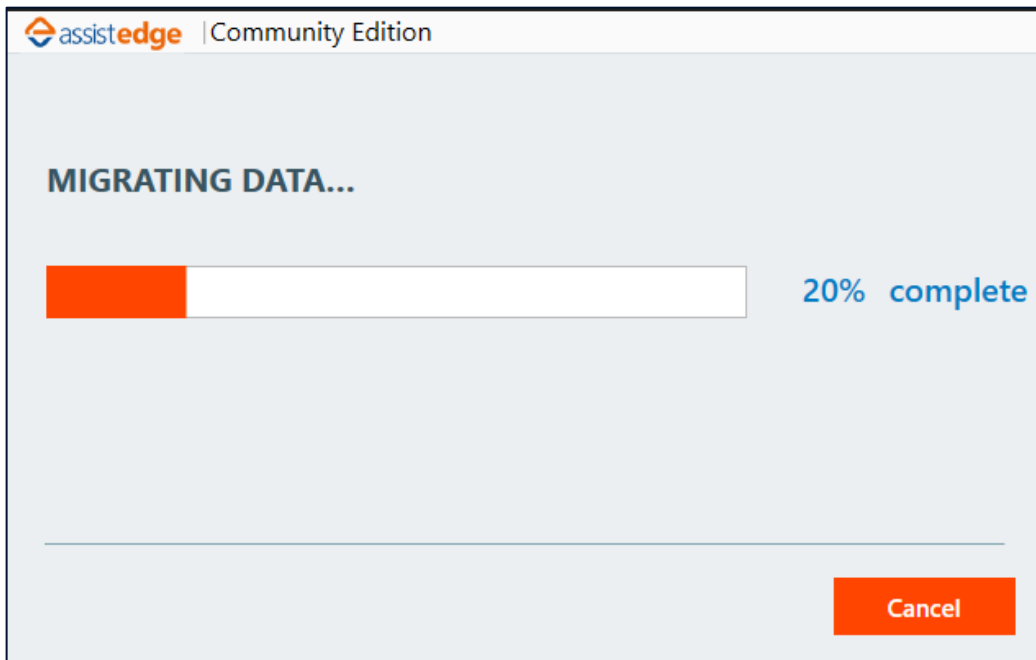


4. After selecting migration source path, click **Next** for data migration.

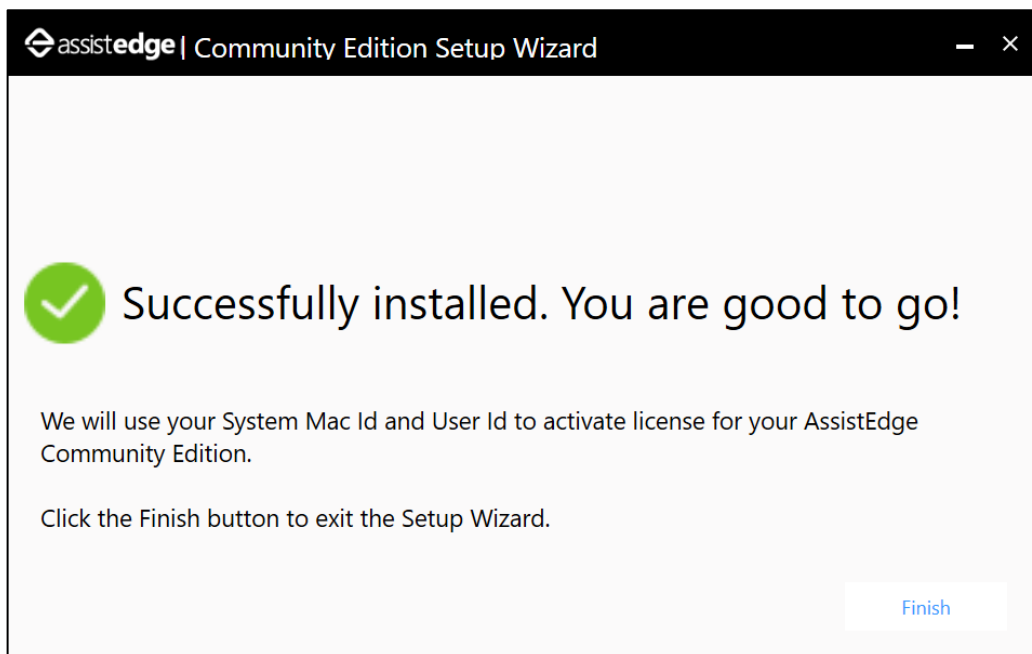
Note: The migration source path should be before folder name: EdgeVerve Systems Limited. For example, if you have installed the previous version under **CE18** folder name as shown in the following figure, then select the migration source path as **CE18**.



The data migration process status is displayed. To abort the migration, click **Cancel**.



The latest version of AssistEdge Community Edition is successfully installed.



5. Click **Finish** to exit the **Setup Wizard**.
You are now able to access all the available features of AssistEdge Community Edition 20.1 for successful automation.

Note: There is no change to the EVA bot.

3.4 License Expiry Notification

On launching the latest AssistEdge Community Edition 20.1, the Studio splash screen displays the respective **License expiry date**.

Note:

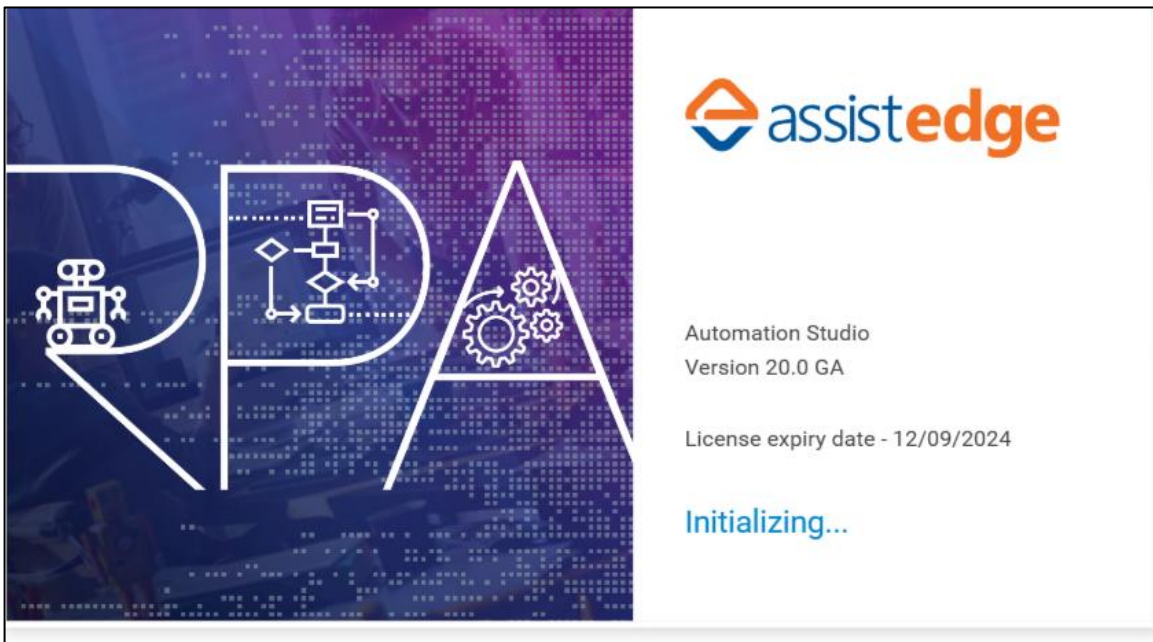
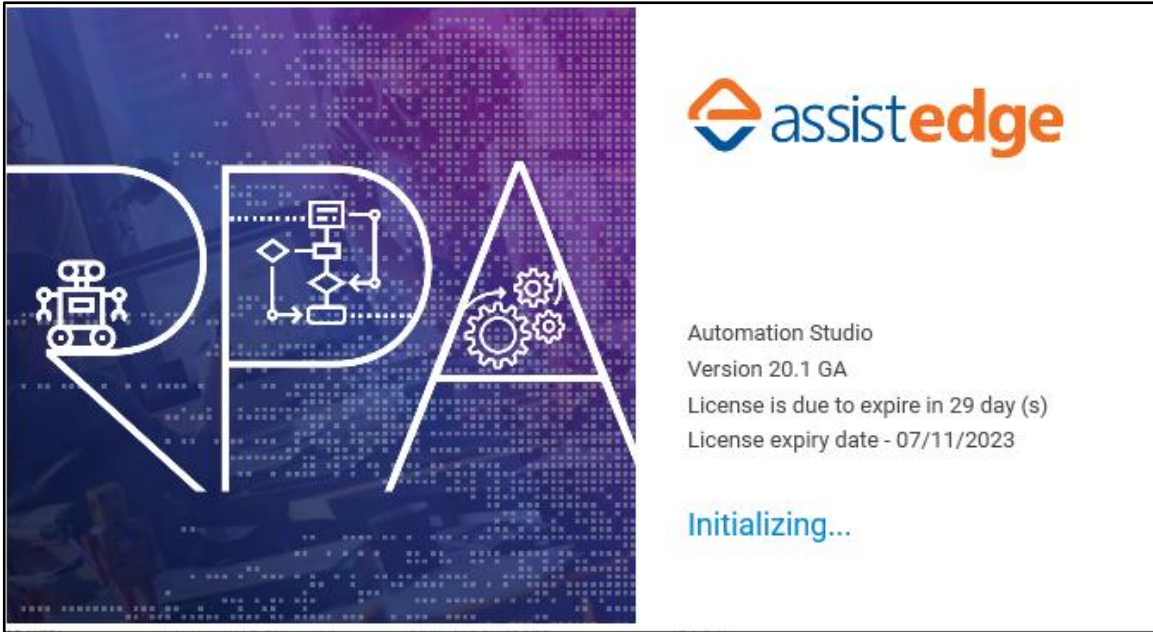
License expiry date - <dd/mm/yyyy> is always displayed.

When 30 or less days are left, following message is displayed on the screen.

License is due to expire in <> day (s)

License expiry date - <dd/mm/yyyy>

The following are the sample screenshots for license expiry:



About EdgeVerve

EdgeVerve Systems Limited, a wholly owned subsidiary of Infosys, is a global leader in developing digital platforms, assisting clients to unlock unlimited possibilities in their digital transformation journey. Our purpose is to inspire enterprises with the power of digital platforms, thereby enabling our clients to innovate on business models, drive game-changing efficiency and amplify human potential. Our platforms portfolio across Automation (AssistEdge), Document AI (XtractEdge), and Supply Chain (TradeEdge) helps inspire global enterprises to discover & automate processes, digitize & structure unstructured data and unlock the power of the network by integrating value chain partners. EdgeVerve, with a deep-rooted entrepreneurial culture, our innovations are helping global corporations across financial services, insurance, retail, consumer & packaged goods, life sciences, manufacturing telecom and utilities, and more. EdgeVerve. Possibilities Unlimited.

www.edgeverve.com