SAP Discovery Center Mission: SAP Build Process Automation and Document Classification

Package version: 1.0.2

THE BEST RUN SAP

Version	Date	Description
1	May 27, 2021	Document created

TABLE OF CONTENTS

INTRODUCTION

This document describes the SAP Build Process Automation sample **DOX** and provides the following information:

- Description (functional and technical)
- Version used to generate this sample

It also contains information on prerequisites, such as the steps to follow before launching the agent.

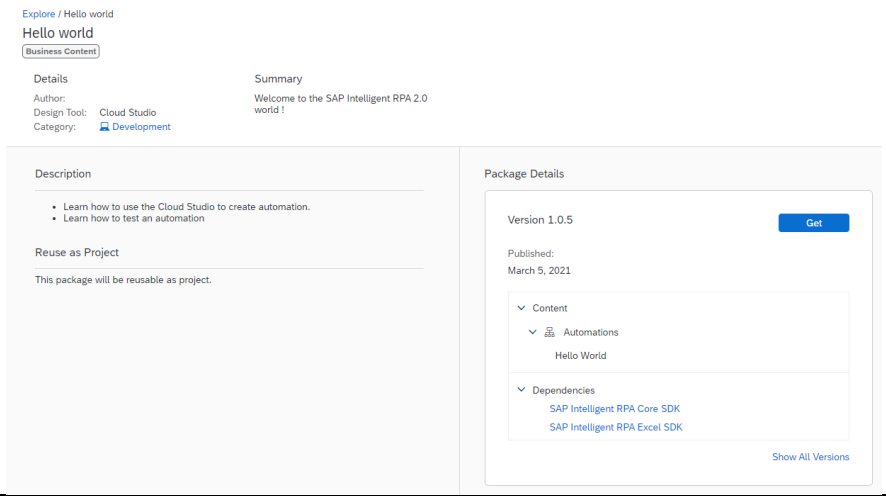
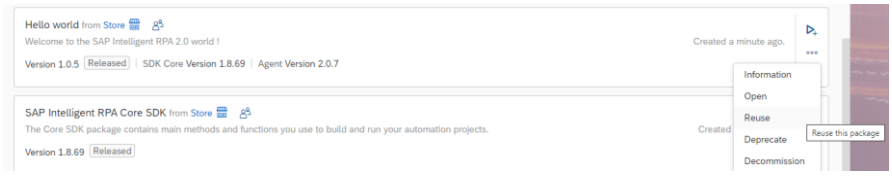
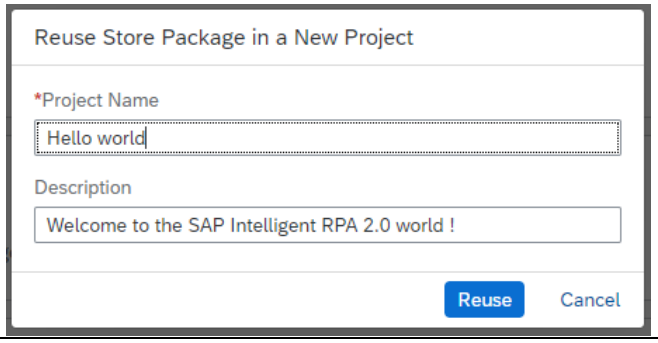
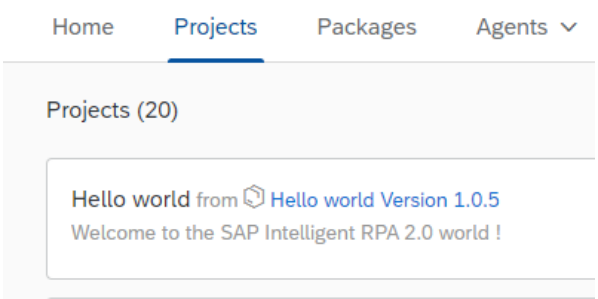
IMPORTANT RECOMMANDATION

General

To use this sample, you need to have a basic knowledge and understanding of SAP Build Process Automation tool. At the very least you need to know how to build an automation, add and modify activities and generate a package.

Reuse the sample as a new project

Note: screenshot might display a different name than the one of this sample.

From the Cloud Factory, open the Store tab and select the sample you want to retrieve. Click on the **Get** button.	
Once the package is retrieved, open the Packages tab of the Cloud Factory. Click on the Options button of the package you just retrieved and select the option **Reuse**.	
Set a name for the project to be created.	
Open the project that has just been created.	
If needed, update the content of this project, and generate a new package from it.	

You need to execute this procedure to be able to open the project and see all its content (the captured applications, the declared items, the automations, etc.).

DESCRIPTION

This package contains captures, datatype and automations that are described below. See chapter Version for more details about the version of the Desktop Agent and the SDK dependencies.

Settings

This section describes the settings of the project such as environment variables or dependent packages that are used in the automations.

Environment variables

Name	Type	Description
workingDirectory	String	Folder where attached files are temporary saved
uaaClientId	String	The uaa.cliendId field from the service key
uaaClientSecret	Password	The uaa.cliendSecret field from the service key
uaaUrl	String	The uaa.url field from the service key
serviceUrl	String	The url field from the service key
searchMailCriteria	Email Search Criteria	Criteria used to search mails

Dependent packages

N/A

Captures

This section describes the captures which were made to pilot the application in this sample. It will also describe the different methods which were used to capture the pages and declare the items.

Datatypes

This section describes the datatype used in this sample. It describes the structure of the datatype and where it is used in the automations.

DCResult

Name	Type	Description
status	String	Required. Status of the service for the processing of the document submitted in the automation. Note: this is the content of the status field (see documentation)
predictions	Any	The result from the document classification. Note: this is the content of the predictions field (see documentation)

Automations

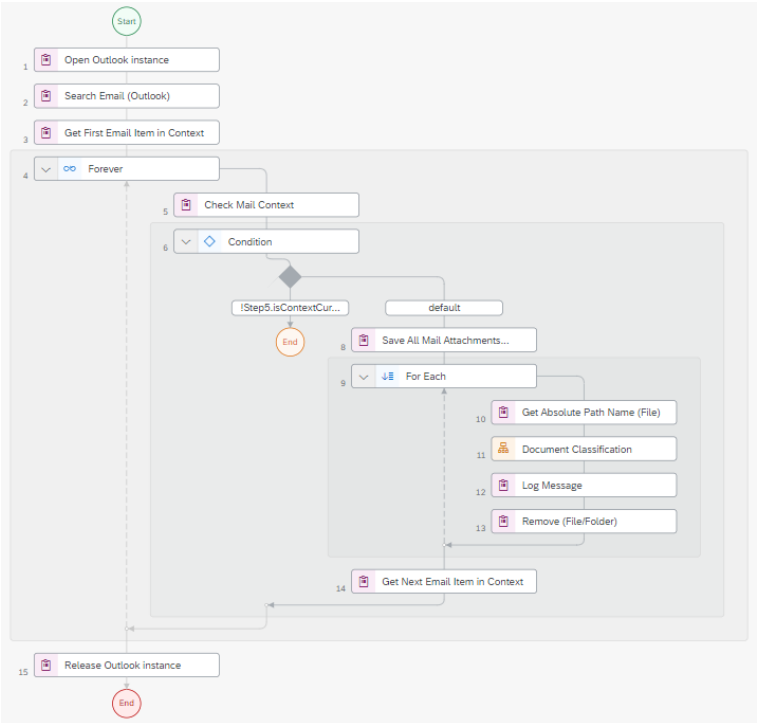
Outlook Process

Type: Attended

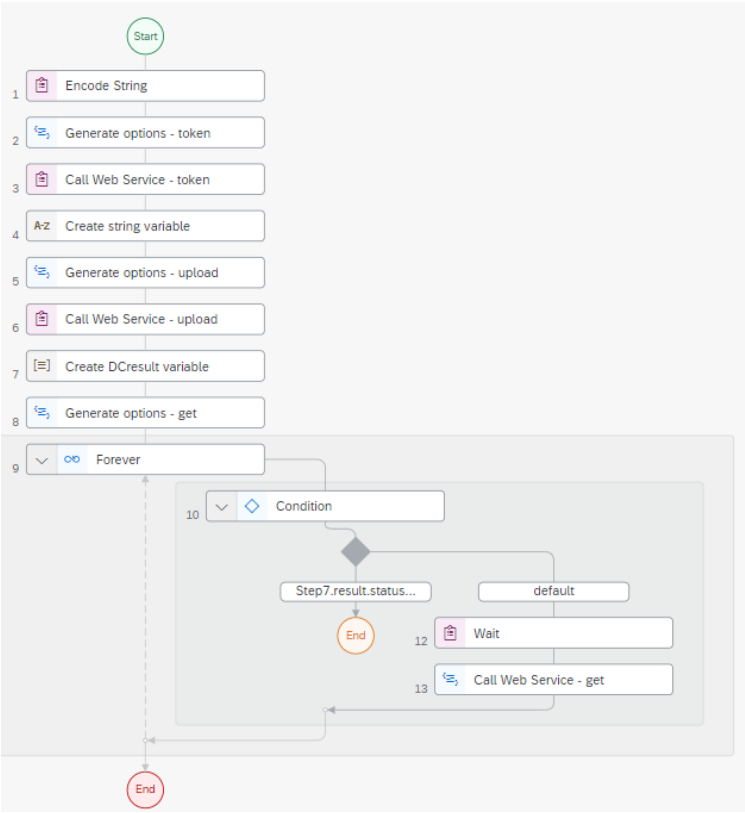
Input: None

Output:

Name	Type	Description
result	DCResult	JSON containing the predictions and the status of the extraction (DONE or FAILED).

	The agent first opens an Outlook instance and searches for e-mails according to the criteria set as environment variable. The selected e-mails are available in a list stored in the context. The Forever loop works on this list as long an e-mail is available. The agent saves temporarily the documents attached in the mail and for each one of them, it will launch the Document Classification automation (see below). After getting the results the agent removes the files for storage location and moves to the next e-mail in the context.

Document Classification
Type: Attended
Input: None
Output: None

	The agent first encodes the *clientId* and *clientSecret*. Then it executes a query to get the token which will be used to authenticate to the service. Get file collection delivers the name of the upload file which is used in the next step. The file is then uploaded to the service and the document ID is retrieved from the response. At this point, we create a variable of type DC*Result* (see datatype chapter) and we initialize the status with PENDING value as it always takes some time for the service to process the document. Then the agent will periodically (every 3 seconds) ask the service for the results. The status and predictions fields are stored in the DC*Result* variable (in *Status* and *Predictions*) When the Status is not PENDING anymore (could be DONE or FAILED), the agent exits the loop and the automation end.

	Note: when this automation is used in another one, it is highly recommended to check the value of **status**. If it is DONE, then you can proceed. Otherwise, an error was raised during the process.

VERSION

The product versions used to generate this sample are detailed below. This sample is provided “as is”, with no warranty that it will work correctly with other versions. If some versions of your software are different (such as the tool version or the target application version), you may need to recapture the application and/or update the workflow activities.

SAP Build Process Automation

This sample targets the Desktop Agent **2.0.10** or higher.

The following SDK dependencies were used to generate this sample:

irpa_core	1.10.69
irpa_excel	1.10.69
irpa_outlook	1.10.69
irpa_pdf	N/A
irpa_ui5	N/A
irpa_word	N/A

See documentation for more details about the compatibility between SDK version and Desktop Agent.

Target application

Document Classification Service	Date of last test: 20th May 2021

PREREQUISITES

Global setup

SAP Build Process Automation must be installed in accordance with the installation guide available here. An SAP Build Process Automation Factory must be available with a suitable environment (containing an agent). All information can be found in the “Getting Started” section accessible via the above link.

Specific steps to follow before launching the agent

1. Set up an instance of the SAP Document Classification service on your tenant
2. Generate the service key and get the following information from the key:
    a. url
    b. uaa.clientid
    c. uaa.clientsecret
    d. uaa.url
3. When you deploy the package, set the environment variables with the following information:

workingDirectory	The path of the folder used to temporary store the files from the mails
serviceUrl	url from the service key
uaaClientId	uaa.clientid from the service key
uaaClientSecret	uaa.clientsecret from the service key
uaaUrl	uaa.url from the service key
mailSearchCriteria	Criteria used to search emails in the automation

EXPECTED OUTPUT
The automation **Document Classification** returns two variables:
Status = DONE
Predictions = list of document types

www.sap.com/contactsap

THE BEST RUN SAP®