



How to use User Tasks

Project name: User Task – Cloud Studio

Package version: 1.0.0

Version of document	Date	Description
1	20 August 2021	Document created

TABLE OF CONTENTS

INTRODUCTION	4
IMPORTANT RECOMMANDATION	5
General	5
Reuse the sample as a new project	5
DESCRIPTION	7
Settings	8
<i>Environment variables</i>	8
<i>Dependent packages</i>	8
Captures	8
Datatypes	8
<i>Configuration Item</i>	8
User Tasks	8
<i>Main Menu</i>	8
<i>Create File</i>	9
<i>Edit File</i>	9
<i>Make Your Changes</i>	10
<i>Delete File</i>	11
Automations.....	11
<i>manageConfigFiles</i>	11
<i>createConfigFiles</i>	11
<i>editConfigFiles</i>	12
<i>selectFile</i>	13
VERSION	14
SAP Build Process Automation	14
Target application	14
PREREQUISITES.....	15
Global setup	15
Specific steps to follow before launching the agent	15
EXPECTED OUTPUT	16

INTRODUCTION

This document describes the SAP Build Process Automation sample **User Task – Cloud Studio** 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.

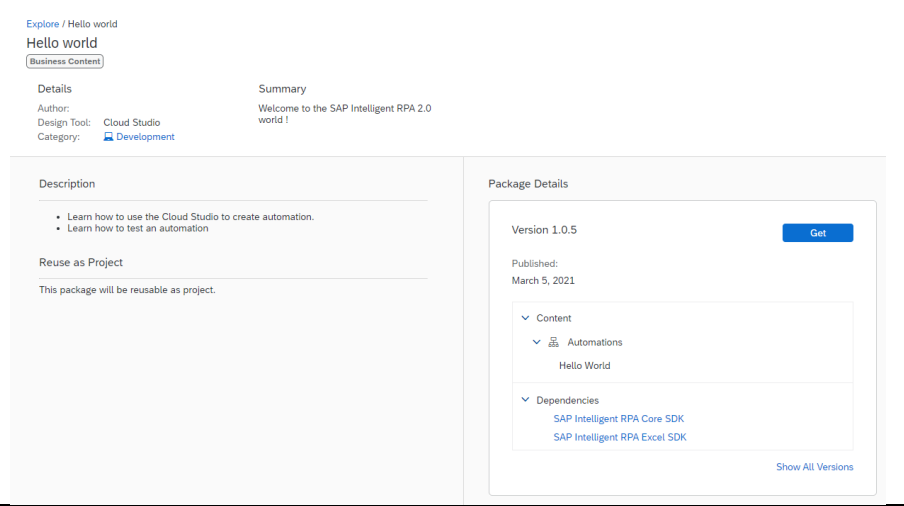
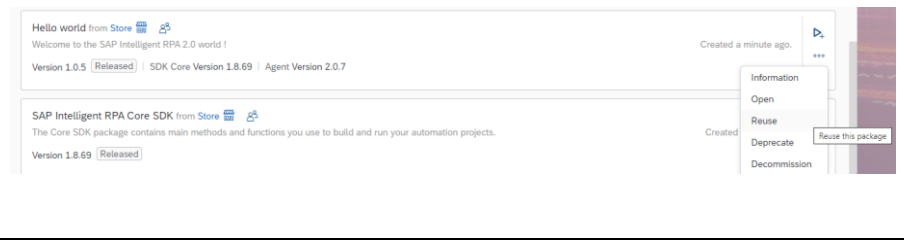
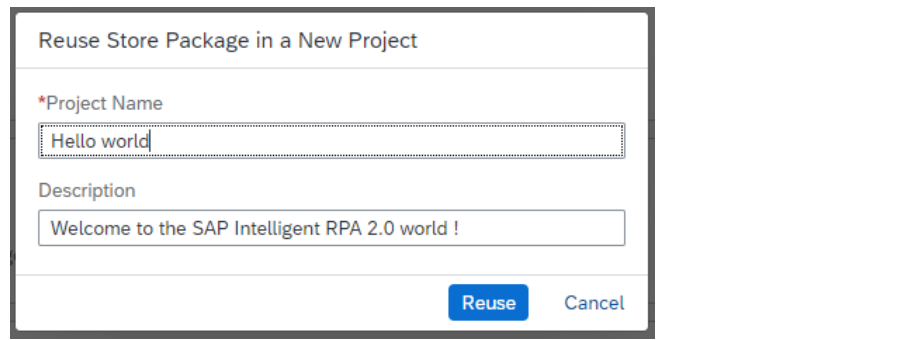
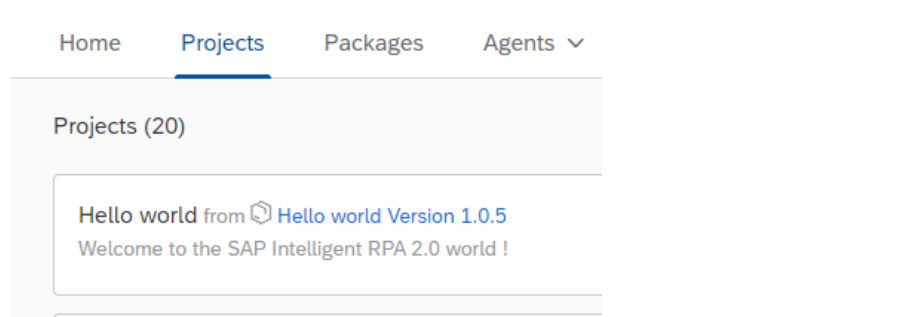
IMPORTANT RECOMMENDATION

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.

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

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 automation.

Environment variables

Name	Description	Type
folderPath	Folder path to store configuration files	String

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.

N/A

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.

Configuration Item

Name of attribute	Type	Description
Name	String	Name of the user
Country	String	Country from which user belongs to
Hobbies	String	List of user's hobbies
Agree	Boolean	If true, agree to policy

User Tasks

This section describes the user tasks used in this sample. It describes the structure of the user task.

Main Menu

Input: options (String) [List]

Output: decision (String), processor (String), status (String), selectedOption (String)

<p>Choose the action to perform from the drop-down.</p> <p>Actions: <input type="text" value=""/></p> <hr/> <p><input type="button" value="Continue"/> <input type="button" value="+ Click to Add a Reject Decision"/></p>	<p>This user task uses input options which is a list of actions that the user can perform – create, edit or delete configuration file.</p> <p>Output is the user selected option which gets stored in the variable selectedOption.</p>
--	--

Create File

Input: countryList (String) [List], hobbiesList (String) [List]

Output: decision (String), processor (String), status (String), selectedName (String), selectedCountry (String), selectedHobbies (String) [List], selectedAgreed (Boolean)

<p>Please provide the details below:</p> <p>Name: * <input type="text" value=""/></p> <p>Country: <input type="text" value=""/></p> <p>Hobbies: <input type="text" value=""/></p> <p>Please tick the checkbox below to agree to the policies:</p> <p>Agree: <input type="checkbox"/></p> <hr/> <p><input type="button" value="Continue"/> <input type="button" value="+ Click to Add a Reject Decision"/></p>	<p>This user task is responsible for creating a configuration file with the information provided in different components.</p> <p>Values in drop-down for Country and Hobbies come from list countryList and hobbiesList respectively.</p> <p>Name gets stored in selectedName. Country chosen is stored in variable selectedCountry. Multiple hobbies can be selected and gets stored as a list in selectedHobbies. Agree value is saved in selectedAgreed.</p>
---	---

Edit File

Input: filesToEdit (String) [List]

Output: decision (String), processor (String), status (String), selectedFileToEdit (String)

<p>Please choose a file to edit from the list below.</p> <p>Configuration File: <input type="text" value=""/></p> <hr/> <p style="text-align: right;"> <input type="button" value="Continue"/> <input type="button" value="+ Click to Add a Reject Decision"/> </p>	<p>This user task gets the list of all the configuration files created in the environment variable, folderPath. The list of files is stored in variable filesToEdit.</p> <p>User can choose any file to edit and the output gets saved in selectedFileToEdit.s</p>
---	---

Make Your Changes

Input: currentName (String), countryList (String) [List], hobbiesList (String) [List], currentAgreedStatus (Boolean), item {Configuration Item}

Output: decision (String), processor (String), status (String), updatedName (String), updatedCountry (String), updatedHobbies (String) [List], updatedAgreedStatus (Boolean)

<p>Please make the changes below.</p> <p>Name:* <input type="text" value="currentName"/></p> <p>Country: <input type="text" value=""/></p> <p>Hobbies: <input type="text" value=""/></p> <p>Agree: <input type="checkbox"/></p> <hr/> <p style="text-align: right;"> <input type="button" value="Continue"/> <input type="button" value="+ Click to Add a Reject Decision"/> </p>	<p>The user task is triggered from automation editConfigFiles and is responsible for making the changes.</p> <p>The Name attribute saved in the configuration file gets fetched from currentName variable and is shown in the user task. Similarly, the initial value of Agree attribute is stored in currentAgreedStatus. The components Country and Hobbies are a static list of all the countries, countryList and hobbies, hobbiesList. User can choose from the respective lists.</p> <p>The chosen values for Name, Country, Hobbies and Agree are saved in variables updatedName, updatedCountry, updatedHobbies and updatedAgreedStatus respectively.</p>
---	--

Delete File

Input: filesListforDeletion (String) [List]

Output: decision (String), processor (String), status (String), selectedFileToDelete (String)

<p>Choose the configuration file to delete:</p> <p>Config File: <input type="text"/></p> <p><input type="button" value="Continue"/> <input type="button" value="+ Click to Add a Reject Decision"/></p>	<p>The user task is responsible for deleting the chosen configuration file. The list of existing configuration files is saved in filesListforDeletion.</p> <p>The selected file name for deletion gets stored in variable selectedFileToDelete and the file is then, deleted.</p>
---	---

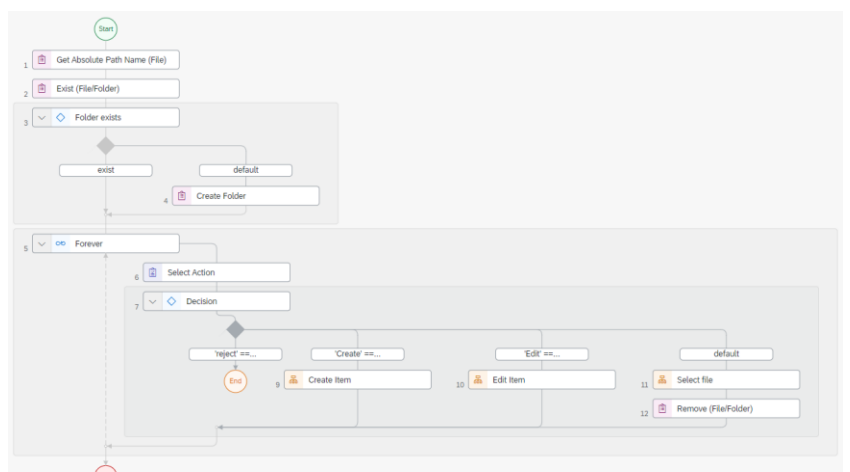
Automations

manageConfigFiles

Type: Attended

Input: N/A

Output: N/A

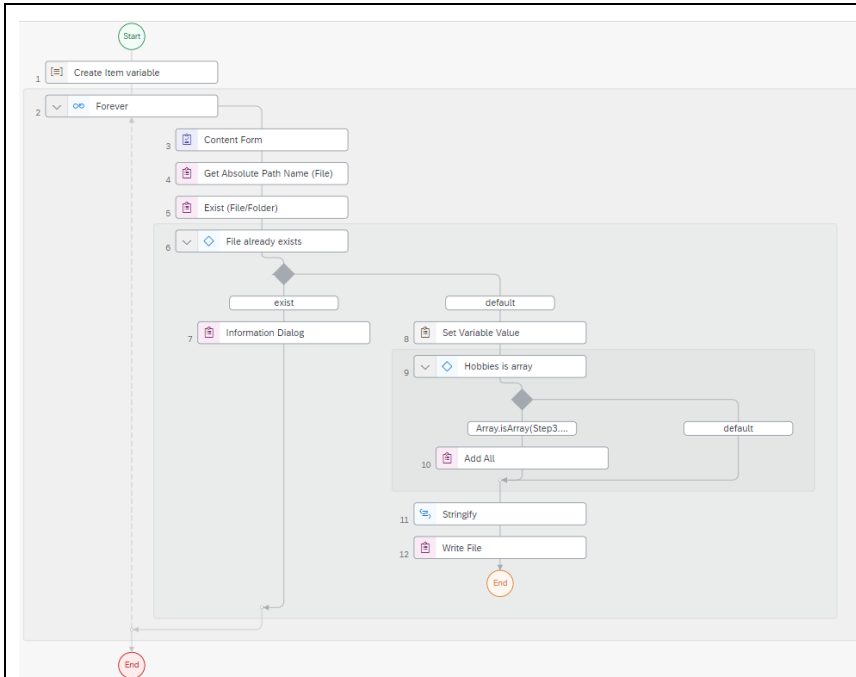
	<p>This automation is the main entry point.</p> <p>The agent will use the environment variable folderPath to check whether the folder exists or not and create/ edit/ delete the configuration files in that folder.</p> <p>The path existence is checked. If the path does not exist, then, folder is created. Else user is provided with the user task Main Menu to choose the action to perform.</p>
--	---

createConfigFiles

Type: Attended

Input: N/A

Output: N/A



This automation allows user to create new configuration files in the path provided in environment variable **folderPath**.

A configuration file is created using the variable **selectedName** provided in user task **Create File**. File format is ".json".

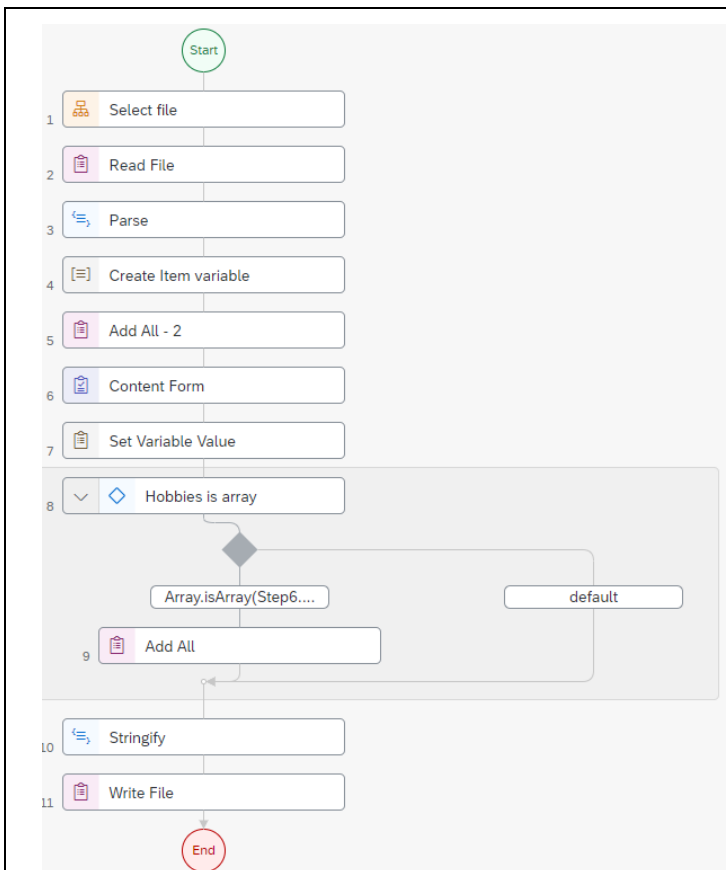
A forever loop checks for the existence of the file with the same name. If file name already exist, then, user is prompted with a message dialog. If not, then, the configuration file is created.

editConfigFiles

Type: Attended

Input: N/A

Output: N/A



This automation takes the environment variable **folderPath** to get the list of files to edit and then, finally, write the changes at the same path.

The list of files present in the path provided are shown to the user in user task **Edit File**. The selected file is then, read and parsed. The respective values are mapped to different fields in data type **Configuration Item**.

The user task **Make Your Changes** allows user to make relevant changes to the file.

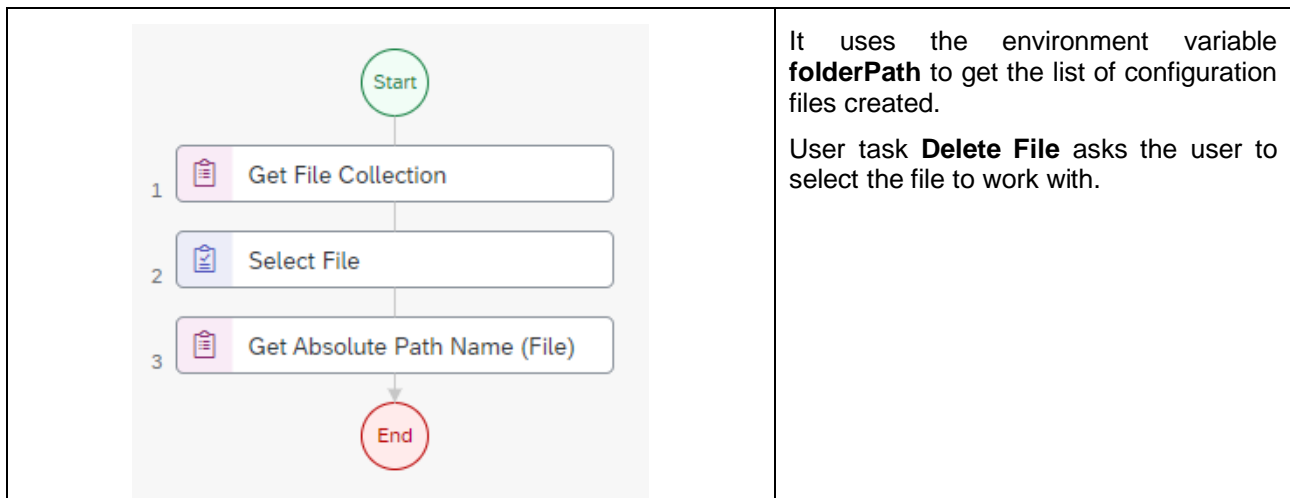
The custom script use stringify function to concatenate the changes and then, write the changes in the same file without changing the file name.

selectFile

Type: Attended

Input: N/A

Output: N/A



It uses the environment variable **folderPath** to get the list of configuration files created.

User task **Delete File** asks the user to select the file to work with.

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 **2.0.15** or higher.

The following SDK dependencies were used to generate this sample:

irpa_core	1.14.54
irpa_excel	1.14.54
irpa_outlook	N/A
irpa_pdf	N/A
irpa_ui5	N/A
irpa_word	N/A
irpa_powerpoint	N/A

See [documentation](#) for more details about the compatibility between SDK version and Desktop Agent.

Target application

N/A

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

- When you deploy your automation, set the environment variable **folderPath** with the location where configuration files need to be managed.

EXPECTED OUTPUT

N/A

www.sap.com/contactsap

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See www.sap.com/copyright for additional trademark information and notices.

THE BEST RUN

