

	Work Instruction Number WI-----	Revision Date 22/04/2024
	Author Valentin B.	
	Department Engineering Testing and Documentation	
Work Instruction Title WI – Setting up ChatEditor end-2-end and run them		

Table of Contents

Prerequisites – Software needed	2
Git*	2
Python and its libraries.....	2
Python	2
Python Libraries.....	3
Visual Studio Code*	2
Robot Framework and its libraries	3
Robot Framework	3
Robot Framework Libraries	4
ChatEditor	4
How to get the robot files.....	4
Getting the code from GitHub – private repository.....	4
Running the tests and interpreting the results	4
Opening the results.....	5
Change Summary.....	6
Final Notes	6

Prerequisites – Software needed

- **Git***
- **Python**
 - **Python Library: PyAutoGUI**
 - **Python Library: PyGetWindow**
 - **Python Library: OpenCV**
 - **Python Library: PyTesseract**
- **Visual Studio Code***
- **Robot Framework**
 - **Python Library: robotframework**
 - **Python Library: rpaframework[windows]**

The software list is recommended for running the e2e suite. The ones marked with * are not mandatory, but for making the process easier, they are nice to have.

- Without **Git** we would be unable to get the latest code from the server. A workaround for this is in place.
- Without **Visual Studio Code** we won't be able to read the code in an understandable way, besides that it has integrated a Terminal (CMD).

Git*

1. Download git from <https://git-scm.com/download/win> (64bit)
2. Open the installer (leave all the option as they are in the installer)
3. Next → Next → Next → Next → Next → Next → Next → Next → Next → Next → Next → Next → Next → Next → Next → Next → **Install**
4. Check *Launch Git*
5. Uncheck *View Release Notes*
6. Press **Finish**

Git should be installed now. The git terminal should open at the end.

Visual Studio Code*

1. Download Visual Studio Code from <https://code.visualstudio.com/download> (64bit)
2. Open the installer
3. Accept the agreement → Next → Next → Next → *Check "Add Open with Code" options* → Next → **Install**
4. Press **Finish**

Visual Studio Code should be installed. We can check this by opening it.

Python and its libraries

Python

1. Download Python from <https://www.python.org/downloads/release/python-3105/>
2. Press on "Download Python" button
 1. Open the installer **with administrator rights**
 2. Check *Add python.exe to PATH* → Install Now → Close

Python should be installed. We can check this by open a CMD window and type `python -version` the result should be the version that we installed (3.10).

Python Libraries

1. Open a CMD window **with administrator rights**;
2. Navigate to `C:\git`

Note: All the following commands should be run from `C:\Git`, unless expressed otherwise

3. Install PyAutoGUI library:

PyAutoGUI is a Python library that allows users to automate interactions with the Graphical User Interface. It enables remote control of the mouse and keyboard using Python code, which is useful for automating repetitive tasks or interactions with desktop applications.

- a. Type `pip install pyautogui`
- b. Press Enter;
- c. The library should be installed;

4. Install PyGetWindow library:

PyGetWindow is a Python library that allows users to access and interact with windows on their desktop. It provides functions to retrieve information about open windows, such as titles, dimensions, positions, etc., and to control interactions with these windows.

- a. Type `pip install pygetwindow`
- b. Press Enter;
- c. The library should be installed;

5. Install OpenCV library:

OpenCV is an open-source library specialized in image and video processing. It offers functions and algorithms to perform various image processing operations, including object recognition, face detection, image segmentation, camera calibration, and many others.

- a. Type `pip install opencv-python`
- b. Press Enter;
- c. The library should be installed;

6. Install PyTesseract library:

PyTesseract is a Python library that provides a simple interface for using the Tesseract Optical Character Recognition (OCR) engine. In other words, it allows users to extract text from images or documents.

- a. Type `pip install pytesseract`
- b. Press Enter;
- c. The library should be installed;

Robot Framework and its libraries

Robot Framework

Description:

- Robot Framework is a generic open-source automation framework for acceptance testing, acceptance test-driven development (ATDD), and robotic process automation (RPA).
- It utilizes a simple, keyword-driven syntax that allows both technical and non-technical users to write test cases in a readable and understandable format.
- Robot Framework supports a wide range of libraries and extensions for testing web applications, APIs, databases, desktop applications, and more.
- It provides built-in reporting and logging capabilities, making it easy to track test execution results and debug test cases.
- With its extensibility and flexibility, Robot Framework is widely used in agile and continuous integration/continuous deployment (CI/CD) environments for automating software testing and validation processes.

Using `pip`, which is the package manager for Python, install Robot Framework:

1. Open a CMD window;
2. Type `pip install robotframework`
3. Press Enter;

Robot Framework should be installed. We can check this by open a CMD window and type

`robot -version` the result should be the version that we installed.

Robot Framework Libraries

Using pip, which is the package manager for Python, install RPA.Windows and RPA.Desktop libraries:

1. Open a CMD window;
2. Type `pip install rpaframework[windows]`
3. Press Enter;

RPA.Windows & RPA.Desktop libraries should be installed. To check this by open a CMD window and type `pip list` which will generate a list of all installed Python packages and we can search in this list to see if the *RPA.Windows* and *RPA.Desktop* libraries are present.

ChatEditor

For installing or uninstalling ChatEditor please consult the appropriate Work Instruction.

How to get the robot files

A robot file is a set of instructions that contain the code (steps) that the framework should follow in order to execute the tests.

Getting the code from GitHub – private repository

4. Open in Visual Studio Code (Right click > open with Code) `C:\git`
5. In the **Terminal** write
`git clone https://github.com/AssuranceStudio/ChatEditor_e2e.git`
 Now the code should be downloaded from git server, when it is done;
6. Please see the section [\(Running and Interpreting the results\)](#)
7. In order to start the run of the tests you should
 - a. In a new terminal (`Ctrl+Shift+``)
 - b. Run `robot .\Tests`
8. The tests should be running.

Running the tests and interpreting the results

Notes:

After all the steps in the previous two chapters have been done (getting the software and getting the code), we should be able now to run the tests:

1. Open Visual Studio Code
2. Select File → Open Folder
3. Select the folder in which the project was cloned (it should be)
`D:\git\e2e_chat_editor_robot`
4. Open a new Terminal window and run:
 - a. For running all the tests it's simply
`robot .\Tests`
 - b. For running specific tests, it should be: (in *italic* we have the examples)
 - i. If we want to run a single file (the path to the file should be provided)
`robot .\PATH_TO_FILE.robot`

`robot .\Tests\Vocabulary\TouchChatEN\TouchChatEnglishSimbolstix\4BasicSS.robot`

- ii. If we want to run a single folder (path to folder)

`robot .\PATH_TO_FOLDER\`

`robot .\Tests\Vocabulary\TouchChatEN\TouchChatEnglishSimbolstix\`

- iii. If we want to run multiple files (path to files should be provided)

`robot .\PATH_TO_FILE1.robot .\PATH_TO_FILE99.robot`

5. After running the tests, we will be having a result in the terminal window:

```
=====
MultiChat15AdolescentSS & 4BasicSS          | PASS |
4 tests, 4 passed, 0 failed
=====
```

6. But if we want a full look of the run tests, we have to check log.html and report.html files – Explained in this [section](#).

Opening the results

After running the tests, two files with the results will be generated, **log.html** and **report.html**:

1. The **log.html** file contains a detailed log of all events that occurred during the execution of tests. It provides information such as the start and end time of each test, along with any logging messages or error information that occurred during the test run.
2. The **report.html** file presents a summarized and easily readable report, providing an overview of the test results. It includes information such as the total number of tests executed, the number of tests passed successfully, the number of failed tests, and any tests that were skipped.

The two files can be opened in a web browser window for viewing.

Notes:

- After each run, we should be keeping track of the results that we get;
- We can do that by renaming the files **log.html** and **report.html** to the name of what we ran;
- If we run multiple times, without renaming the folder, it can be difficult to get the true results.

Change Summary

Date	Pg.	Par.	Summary comments	Author	Approver
22-04-24	0	6	Created the WI	Valentin N	MN
22-04-24	-	-	Modified the WI	Marius N	JH, ED, VB
12-06-24	-	-	Updated the WI per changes requested	Marius N	JH, ED, VB

Final Notes

Please take in consideration that this framework is not at its final stages, given this, changes might arise.