**Knowledge Transfer Document: Zip Report Utility for Robot Framework**

**Objective**

This document details a utility designed to archive test execution reports automatically at the end of a Robot Framework test run. The utility copies result files to a timestamped folder and compresses the folder into a zip file for organized storage.

**Utility Overview**

* **Purpose:**
  + Automates the archiving of test execution results.
  + Ensures every test execution’s reports are stored in a dedicated, timestamped folder.
  + Compresses the archived folder into a .zip file for efficient storage.
* **Pre-requisite:**
  + **Install 7-Zip:**  
    Download and install [7-Zip](https://www.7-zip.org/). Add the 7-Zip executable path to your system's environment variables for command-line access.  
    Example Path:

plaintext

Copy code

C:\Program Files\7-Zip

**Key Functionalities**

1. **Create Timestamped Folder:**
   * Generates a folder named Automation\_Report\_<YYYY\_MM\_DD\_HH\_MM> using the current system date and time.
2. **Move Results to Archive Folder:**
   * Copies all files and subfolders from the Results directory into the newly created folder.
3. **Zip the Archived Folder:**
   * Compresses the archive folder using 7-Zip and saves it to the project’s root directory.

**Implementation Details**

**Folder Structure**

The utility assumes the following project folder structure:

python

Copy code

Project/

├── RobotPython/

│ ├── Results/ # Contains the test execution results

│ ├── Archive\_Reports/ # Folder to store archived test reports

│ ├── Utilities/

│ │ └── zip\_report.py # The utility script

│ ├── Archive\_Reports.zip # Consolidated zip file of all archived reports

│ ├── Tests/

**Python Utility: zip\_report.py**

python

Copy code

import os

import subprocess

from datetime import datetime

import shutil

import time

# Paths Configuration

RESULTS\_PATH = r"RobotPython/Results"

ARCHIVE\_PATH = r"RobotPython/Archive\_Reports"

ZIP\_PATH = r"RobotPython/Archive\_Reports.zip"

# Create Timestamped Folder

def create\_archive\_folder():

now = datetime.now()

timestamp = now.strftime("%Y\_%m\_%d\_%H\_%M")

archive\_folder = os.path.join(ARCHIVE\_PATH, f"Automation\_Report\_{timestamp}")

os.makedirs(archive\_folder, exist\_ok=True)

return archive\_folder

# Move Results to Archive Folder

def move\_results\_to\_archive(archive\_folder):

if os.path.exists(RESULTS\_PATH):

for item in os.listdir(RESULTS\_PATH):

source = os.path.join(RESULTS\_PATH, item)

destination = os.path.join(archive\_folder, item)

try:

shutil.move(source, destination)

print(f"Moved: {source} -> {destination}")

except Exception as e:

print(f"Error moving {source}: {e}")

# Compress Folder with 7-Zip

def compress\_folder(archive\_folder):

try:

if os.path.exists(ZIP\_PATH):

# Append to existing zip

print("Appending to existing zip...")

subprocess.run(["7z", "a", ZIP\_PATH, archive\_folder], check=True)

else:

# Create a new zip

print("Creating new zip...")

subprocess.run(["7z", "a", ZIP\_PATH, archive\_folder], check=True)

print(f"Archive created at {ZIP\_PATH}")

except subprocess.CalledProcessError as e:

print(f"Error zipping archive: {e}")

# Main Execution

if \_\_name\_\_ == "\_\_main\_\_":

print("Starting archive process...")

archive\_folder = create\_archive\_folder()

move\_results\_to\_archive(archive\_folder)

time.sleep(2) # Wait to ensure all operations are complete

compress\_folder(archive\_folder)

print("Archive process completed.")

**How to Integrate with Robot Framework**

1. **Add the Utility as a Listener:**  
   Configure Robot Framework to run the zip\_report.py utility after test execution is complete. Use the --listener option:

bash

Copy code

robot --listener Utilities/zip\_report.py Tests/

1. **Run Tests:**  
   Execute your Robot Framework tests as usual. The utility will automatically archive and zip the results upon completion.

**Execution Workflow**

1. **Setup and Pre-requisite:**
   * Install and configure 7-Zip.
   * Add the utility script to the Utilities folder in your project.
2. **During Execution:**
   * Test cases run and generate results in the Results folder.
3. **Post-Execution:**
   * The utility:
     + Creates a timestamped folder under Archive\_Reports.
     + Moves all result files and subfolders into the archive folder.
     + Compresses the archive folder into a zip file.

**Advantages**

* **Automation:** No manual intervention is required for archiving reports.
* **Organization:** Timestamped folders make it easy to track test execution history.
* **Storage Efficiency:** Zipped archives reduce storage usage and make sharing reports easier.

**Example Output**

After execution, the project structure will look like this:

python

Copy code

Project/

├── RobotPython/

│ ├── Results/ # (Empty after execution)

│ ├── Archive\_Reports/

│ │ └── Automation\_Report\_2024\_11\_20\_14\_30/

│ ├── Archive\_Reports.zip # Contains all archived folders

│ ├── Utilities/

│ ├── Tests/

**Best Practices**

1. **Ensure 7-Zip Installation:**  
   Verify that 7-Zip is installed and the environment variable is correctly set. Use 7z in the command line to confirm.
2. **Periodic Cleanup:**  
   Regularly review and clean up the Archive\_Reports folder and the Archive\_Reports.zip file to manage disk space.
3. **Relative Paths:**  
   Use relative paths in the script for easier portability across environments.

**Conclusion**

This utility automates the post-test execution process, ensuring reports are archived and zipped systematically. It is a vital tool for maintaining well-organized and accessible test execution results in Robot Framework projects.