SISMO SFTP FILE UPLOADER

License

This project is licensed under the MIT License - see the LICENSE file for details. You are free to use, modify, and distribute it, but the software is provided as is, without warranty. The authors accept no liability for any damages.

SftpClient Setup and Usage Guide - macOS

This guide explains how to prepare, configure, and run SftpClient.jar on macOS. It also includes instructions for scheduling the application to run daily at 05:00 AM and for removing the scheduled task.

Note: *SftpClient is a Java-based application that connects to an SFTP server and uploads files to a specified directory.*

Prerequisites

1. **Java Runtime Environment (JRE)**
   Ensure the jre folder is included in the same directory as the SftpClient.jar.

2. **Environment Variables**
   The run_macos.sh script requires several environment variables to be set. Update these to match your environment before execution.

3. **Required Folders**
   The following folders in the root directory of the application (created during the zip packaging) are used by default for file uploads:

   - user_indicator: for stock-related indicator files.
   - macro_indicator: for macro indicator files.
   - portfolio: for portfolio files.

You may create your own folders and update the environment variables in run_macos.sh accordingly.

Environment Variables

Below are key environment variables, their purposes, and usage examples:

SFTP_PORTFOLIO_FILE_MAPPER

- **Purpose**: Maps external IDs to substrings in file names for portfolios, enabling automatic renaming before upload.
- **Example**: *PortfolioExternalId1-Portfolio1, PortfolioExternalId2-Portfolio2*
- **Mock File Name**: Portfolio1_Report.csv renamed to PortfolioExternalId1-[timestamp]-[operation].csv (The substring Portfolio1 in the file name matches the mapping PortfolioExternalId1-Portfolio1. The file is renamed to PortfolioExternalId1-[timestamp]-[operation].csv before being uploaded)

SFTP_PORTFOLIO_FILE_MAPPER_DEFAULT_OPERATION

- **Purpose**: Specifies the default operation for portfolio files.
  - F = replace file and delete history
  - M = modify file
- **Example:** M
  Files will be uploaded with the “modify” (update) operation unless specified otherwise.

SFTP_USER_INDICATOR_FILE_MAPPER

- **Purpose**: Same logic as for portfolios, but applies to user indicator files.
- **Example**: *UserExternalId1-UserIndicator1*, *UserExternalId2-UserIndicator2*
- **Mock File Name**: UserIndicator1_Data.csv renamed to UserExternalId1-[timestamp]-[operation].csv and zipped before upload

SFTP_USER_INDICATOR_FILE_MAPPER_DEFAULT_OPERATION

- **Purpose**: Default operation for user indicator files.
- **Example**: M
  Files will be uploaded with the “modify” operation unless specified otherwise.

SFTP_MACRO_INDICATOR_FILE_MAPPER

- **Purpose**: Same mapping logic, applied to macro indicator files.
- **Example**: *MacroExternalId1-MacroIndicator1*, *MacroExternalId2-MacroIndicator2*
- **Mock File Name**: MacroIndicator1_Stats.csv ⍰ renamed to MacroExternalId1-[timestamp]-[operation].csv

SFTP_MACRO_INDICATOR_FILE_MAPPER_DEFAULT_OPERATION

- **Purpose**: Default operation for macro indicator files. F means replace the file, and M means modify the file.
- **Example**: F
  Files will be uploaded with the “replace” operation unless specified otherwise.

✅ *Ensure all environment variables are correctly set in run_macos.bat before running the application.* If you encounter issues, check the log file (log.txt) in the application’s directory for details.

Manual Execution Steps

1. **Edit the Script**
2. **Make the File Executable**:
   - Open the terminal and navigate to the directory containing run_macos.command.
   - Run the following command to ensure it is executable:

```
chmod +x run_macos.command
```

3. **Run the Script**
   Double-click the run_macos.command file in Finder or run command in the terminal:

~~~
  chmod +x run_macos.command
```bash
  ./run_macos.command
~~~

4. **Verify Execution**
   The script checks for the Java runtime and executes SftpClient.jar.

5. **Check Logs**:
   - The script writes logs to /tmp/sftpclient_debug.log.
   - Open the log file to verify the execution or debug any issues:

```
cat /tmp/sftpclient_debug.log
```

Steps to Run SftpClient.jar Manually using Terminal

1. **Adjust Environment Variables:**
   - Open run_macos.sh in a text editor.
   - Update the environment variables with the correct values for your environment.

2. **Run the Script:**
   - Make the script executable:

```
chmod +x run_macos.sh
```

   - Execute the script:

```
./run_macos.sh
```

3. **Verify Execution:**
   - The script will check for the presence of the Java runtime and execute SftpClient.jar.

Schedule SftpClient.jar to Run Daily at 05:00 AM

1. **Prepare the Script:**
   - Ensure schedule_macos.sh is in the same directory as run_macos.sh.

2. **Run the Scheduler Script:**
   - Execute schedule_macos.sh to create a launchd task:

```
./schedule_macos.sh
```

3. **Verify the Task:**

- Check if the task is loaded:

```
launchctl list | grep com.user.sftpclient
```

Remove the Scheduled Task

1. **Run the Removal Script:**
   - Execute remove_schedule_macos.sh to remove the launchd task:

```
./remove_schedule_macos.sh
```

2. **Verify Removal:**
   - Check if the task is unloaded:

```
launchctl list | grep com.user.sftpclient
```

Important Notes

⚠️⚠️⚠️

- All .sh scripts (run_macos.sh, schedule_macos.sh, remove_schedule_macos.sh) must be in the same directory as SftpClient.jar and the jre folder.
- The folders user_indicator, macro_indicator, and portfolio are created during packaging. You may use custom folders by updating run_macos.sh.
- If you encounter errors, check logs in the Task Scheduler or in log.txt.

File Structure

The directory should have the following structure:

```
/your-directory
 ├── run_macos.sh
 ├── schedule_macos.sh
 ├── remove_schedule_macos.sh
 ├── SftpClient.jar
 ├── README_MACOS.md
 ├── README_MACOS.pdf
 ├── jre/
 ├── user_indicator/
 ├── macro_indicator/
 ├── portfolio/
```

Troubleshooting

- **Java Runtime Not Found**
  Ensure the jre folder is present and correctly populated.

- **Task Scheduler Errors**
  Confirm you have the required permissions. Check the History tab for details.

- **Incorrect Environment Variables**
  Double-check the variables in run_macos.sh for typos or invalid values.