

Mac Workflow for Tracking Daily Research Progress

Ronald (Ryy) Glenn Thomas

2025-02-14

Table of contents

1	Introduction	1
2	Step 1: Organize Your Research Folder Structure	4
3	Step 2: Implement a Daily Logging System	4
4	Step 3: Automate Dictation & Summarization via ChatGPT	5
4.1	Step 3.1: Preparing a Dictation Prompt	5
4.2	Step 3.2: Dictating Notes	5
4.3	Step 3.3: Automate Log Updates	6
5	Step 4: Version Control with Git	6
5.1	Automate Daily Git Commits	7
6	Step 5: Search & Retrieve Past Notes	7
7	Summary of the Workflow	8

1 Introduction

Tracking daily research progress can quickly become overwhelming without a structured workflow. This guide outlines an ef-



efficient system using a well-organized directory structure, automated logging, ChatGPT for dictation and summarization, and Git for version control. By implementing this workflow, you can maintain clear, searchable research logs while reducing manual overhead.

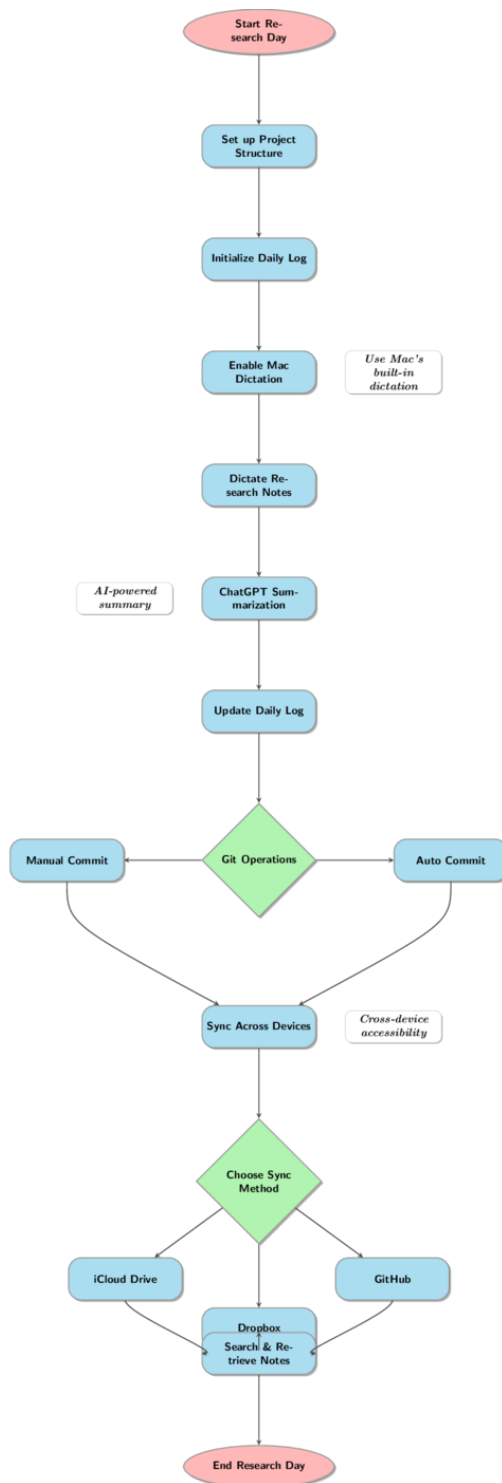


Figure 2: Research Workflow Overview

2 Step 1: Organize Your Research Folder Structure

A structured directory keeps your research files easily accessible and prevents clutter. Use the following command to create a research workspace:

```
mkdir -p ~/prj/research_update ~/prj/{X1,X2,X3,X4,X5,X6,X7,X8,X9,X10}
```

Each project folder should contain: - **notes.md** – Running log of project progress - **references.bib** – Citation management - **data/** – Datasets and related files - **coding/** – Code and analyses - **figures/** – Graphs and visualizations - **tables/** – Data summaries - **archive/** – Storage for non-current files



Figure 3: Folder Structure Example

3 Step 2: Implement a Daily Logging System

Maintaining a centralized daily log helps consolidate research updates. Store daily progress in:

```
```sh
~/prj/daily_log.md
```

To streamline the process, we use a script to automatically generate a structured prompt for ChatGPT.

## 4 Step 3: Automate Dictation & Summarization via ChatGPT

Mac's built-in dictation tool allows hands-free logging, while ChatGPT refines your notes into concise summaries.

### 4.1 Step 3.1: Preparing a Dictation Prompt

Use this Bash script to copy a structured prompt to your clipboard: Call it `rsc` (research summary clipboard).

```
#!/bin/bash

current_time=$(date +"%Y-%m-%d %H:%M:%S")
current_dir=$(basename "$PWD")

prompt="I'm about to dictate research experiment notes.\nWhen I'm done, provide a concise summary."

echo -n "$prompt" | pbcopy
echo "Research prompt copied to clipboard. Paste it into ChatGPT when ready."
```



Figure 4: Mac Dictation Feature

### 4.2 Step 3.2: Dictating Notes

1. Open ChatGPT and follow these steps:
2. copy text from clipboard into the prompt box.
3. Enable Mac's dictation by pressing **Control** twice.

4. Dictate your research notes.
5. Copy and paste the generated summary into your daily log.
6. Use the following script to append the summary to your daily log:

### 4.3 Step 3.3: Automate Log Updates

To append the summarized notes to your daily log, use the following script:

```
#!/bin/bash

Get the current directory name
current_dir=$(basename "$PWD")

Get the current date and time
current_time=$(date +"%Y-%m-%d %H:%M:%S")

Get the clipboard content (MacOS pbpaste)
clipboard_content=$(pbpaste)

Echo the output
#
echo "$clipboard_content" >> ~/prj/research_update/daily_log.md
echo "" >> ~/prj/research_update/daily_log.md

Confirm success
echo "Update for $current_dir appended to daily_log.md in ~/prj/research_update"
push update file to github
cd ~/prj/research_update
git commit -m "Daily log update $(date +%Y-%m-%d)"
git push origin main
```

## 5 Step 4: Version Control with Git

Keeping track of your research logs is crucial. Automate version control with Git:

```
cd ~/prj/research_update
git init
git add daily_log.md
git commit -m "Daily log update $(date +%Y-%m-%d)"
git push origin main
```



Figure 5: Git Workflow

## 5.1 Automate Daily Git Commits

Schedule an automatic commit every night using crontab:

```
0 23 * * * cd ~/prj/research_update && git add . && git commit -m "Daily log $(date)" && git push origin main
```

## 6 Step 5: Search & Retrieve Past Notes

Use ripgrep to find past research logs:

```
#!/bin/bash
current_dir=$(basename "$PWD")
rg $current_dir ~/prj/research_update/daily_log.md | cut -c6-
```



Figure 6: Search Example



Figure 7: Cloud Syncing

## 7 Summary of the Workflow

1. Organize project folders.
2. Set up a centralized daily log.
3. Use ChatGPT for dictation and summarization.
4. Automate log updates with a script.
5. Track changes with Git and schedule daily commits.
6. Easily search and retrieve past notes.

By integrating this workflow, you'll reduce friction in tracking your research, making daily documentation effortless and structured. Would this system work for you? Let me know how it can be improved!