

# Earnings Summary Script - CLI Reference and Workflow Guide

---

## Command Line Usage & Flags

The `run_summary.py` script supports the following CLI flags. You can combine them to control batch processing, file selection, confirmation prompts, and dry runs.

### Available Flags (One-liner format)

- `--dry-run` → Simulates the full pipeline. No files will be renamed, saved, or generated.
- `--skip-to <index|id>` → Starts at the Nth file or jumps to a filename substring match (e.g. `--skip-to 3`, `--skip-to tsla`).
- `--no-confirm` → Skips the interactive "confirm/deselect files" prompt. Ideal for automation.

### Flag Combinations (Examples)

Use Case	Command
Dry run with interactive selection	<code>python3 run_summary.py --dry-run</code>
Dry run with no prompts	<code>python3 run_summary.py --dry-run --no-confirm</code>
Resume from a specific index	<code>python3 run_summary.py --skip-to 2</code>
Resume from a specific file (e.g., TSLA)	<code>python3 run_summary.py --skip-to tsla</code>
Full automation (e.g., cronjob)	<code>python3 run_summary.py --no-confirm --skip-to tsla</code>
Test full automation with no output	<code>python3 run_summary.py --dry-run --skip-to tsla --no-confirm</code>

## Workflow Steps

### 1. Detect files:

2. Automatically finds all `.mp3` and `.pdf` files in the current directory.

### 3. Optional filtering:

4. If using `--skip-to`, begins from that file or index.

5. If not using `--no-confirm`, prompts user to confirm and deselect files interactively.

**6. Parse metadata:**

7. Ticker and quarter are extracted from filenames using regex.

8. Files are renamed to follow the format: `TICKER_qX_YYYY.ext`

**9. Transcribe audio / extract text:**

10. `.pdf` → parsed using PyPDF2

11. `.mp3` → processed via Whisper.cpp

**12. Summarize and quote extraction:**

13. Summarizes the text using an Ollama LLM

14. Extracts quotes using a custom quote extractor script

**15. YouTube script + voiceover + video:**

16. Generates a YouTube-friendly script

17. Converts script to voiceover (via macOS `say` + `ffmpeg`)

18. Builds a static image video (`ffmpeg`)

19. Triggers a dynamic slide-based video builder

**20. File output layout:**

```
summaries/  
  TSLA/  
    q1_2025/  
      summary_tsla_q1_2025.txt  
      youtube_script_tsla_q1_2025.txt  
      voiceover_tsla_q1_2025.mp3  
      video_tsla_q1_2025.mp4  
      video_dynamic.mp4  
      quotes_tsla_q1_2025.json
```

```
tsla_q1_2025_transcript.txt  
tsla_q1_2025_transcript.vtt
```

**1. End of process:**

2. Opens the `summaries/` folder in your browser

3. Logs status messages for each file



## Pro Tips

- Use `--dry-run` to validate everything before generating actual output.
- `--skip-to` is great for resuming a crashed batch or jumping to one file.
- `--no-confirm` lets you script this as part of a CI or cron job.
- Combine flags to streamline repeated workflows — see examples above for templates.
- You can embed these in a bash helper script like `summarize_all.sh` to speed things up.

---

Need help adding another flag like `--only` or `--list`? Just ask.