

FileJump Sync - Write-up - 20251229

Architecture: Apple Silicon (M2) & Android (Termux) **Protocol:** rclone bisync (Newer Wins) + Cloud-to-Cloud Archiving **Status: Stable** (Hash Checks Disabled, Priority Queue Enabled)

PART 1: CRITICAL PREREQUISITES

Both devices must use these specific settings to avoid 501 Not Implemented errors.

1. **Vendor:** owncloud (Strictly. Do NOT use Nextcloud or Infinitescale).
2. **Safety Lock:** Run this command immediately after config on any new device:

Bash

```
rclone config update filejump chunk_size 0
```

PART 2: MAC OS SETUP (Apple Silicon)

A. The Engine (filejump_engine.sh)

Features: "Obsidian First" priority, Safety Net archiving, and modtime comparison (no hash errors).

Bash

```
cat << 'EOF' > /Users/nirmal/filejump_engine.sh
#!/bin/zsh
# HARDCODED PATH FOR APPLE SILICON
RCLONE="/opt/homebrew/bin/rclone"
LOGfile="/tmp/filejump.log"
LOCKfile="/tmp/filejump_engine.lock"
# ARCHIVE LOCATION
BACKUP_DIR="filejump:Bisync_Archive/Obsidian"

if [ -f "$LOCKfile" ]; then
    # If lock is older than 30 mins, kill it. Otherwise, exit.
    if test `find "$LOCKfile" -mmin +30`; then
        rm -f "$LOCKfile"
    else
        exit 0
    fi
fi
```

```

echo $$ > "$LOCKfile"; trap "rm -f $LOCKfile" EXIT

run_job() {
    local MODE="$1"; local LOCAL="$2"; local CLOUD="$3"

    local EXCLUDES=(
        --exclude ".sync/**"
        --exclude ".DS_Store"
        --exclude ".Trash/**"
        --exclude "*.download"
        --exclude "*.conflict*"
        --exclude "**/workspace.json"
        --exclude "**/.obsidian/cache/**"
    )

    # FLAGS: --compare size,modtime (Fixes 'hash missing' errors)
    local FLAGS="--verbose --remove-empty-dirs --recover --conflict-
resolve newer --compare size,modtime"

    if [ "$MODE" = "push" ]; then
        $RCLONE copy "$LOCAL" "$CLOUD" --verbose $EXCLUDES >> "$LOGfile"
2>&1
    elif [ "$MODE" = "sync" ]; then
        if [[ "$LOCAL" == *Obsidian* ]]; then
            $RCLONE bisync "$LOCAL" "$CLOUD" \
                $FLAGS \
                --backup-dir2 "$BACKUP_DIR" \
                $EXCLUDES \
                >> "$LOGfile" 2>&1
        else
            $RCLONE bisync "$LOCAL" "$CLOUD" \
                $FLAGS \
                $EXCLUDES \
                >> "$LOGfile" 2>&1
        fi
    fi
}

# --- PRIORITY QUEUE ---
# 1. Obsidian (Always runs first)
run_job "$1" "/Users/nirmal/Obsidian Mac" "filejump:Obsidian"

# 2. Standard Folders
run_job "sync" "/Users/nirmal/Downloads" "filejump:Synced Downloads"
run_job "sync" "/Users/nirmal/Screenshots" "filejump:Synced Screenshots"
run_job "sync" "/Users/nirmal/Documents" "filejump:Synced Documents"
EOF
chmod +x /Users/nirmal/filejump_engine.sh

```

B. The Watcher (launch_watcher.sh)

Monitors file system events and triggers the engine.

Bash

```
cat << 'EOF' > /Users/nirmal/launch_watcher.sh
#!/bin/zsh
(/opt/homebrew/bin/fswatch -o -l 5 -r \
"/Users/nirmal/Downloads" \
"/Users/nirmal/Screenshots" \
"/Users/nirmal/Documents" \
"/Users/nirmal/Obsidian Mac" | while read num_events; do
    /Users/nirmal/filejump_engine.sh push
done) &
while true; do sleep 300; /Users/nirmal/filejump_engine.sh sync; done
EOF
chmod +x /Users/nirmal/launch_watcher.sh
```

C. The Launch Agent (plist)

Keeps the system running at login.

Bash

```
cat << 'EOF' > ~/Library/LaunchAgents/com.filejump.sync.plist
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>Label</key>
    <string>com.filejump.sync</string>
    <key>ProgramArguments</key>
    <array>
        <string>/bin/zsh</string>
        <string>/Users/nirmal/launch_watcher.sh</string>
    </array>
    <key>RunAtLoad</key>
    <true/>
    <key>KeepAlive</key>
    <true/>
</dict>
</plist>
EOF
```

PART 3: ANDROID SETUP (Termux)

A. Background Service (`start_sync.sh`)

Runs at boot. Handles debounced auto-sync, includes Camera folder.

Bash

```
mkdir -p ~/.termux/boot
cat << 'EOF' > ~/.termux/boot/start_sync.sh
#!/data/data/com.termux/files/usr/bin/bash
termux-wake-lock
LOGfile="$HOME/sync_log.txt"
TRIGGER="/data/data/com.termux/files/usr/tmp/sync_trigger"
REMOTE="filejump"
BACKUP_DIR="filejump:Bisync_Archive/Obsidian"

run_job() {
    # OBSIDIAN (With Archive)
    EXCLUDES="--exclude .sync/** --exclude .trash* --exclude .dtrash* \
        --exclude *.crdownload --exclude *.tmp --exclude \
*.opdownload \
        --exclude *.part --exclude *.conflict* --exclude \
**/workspace.json \
        --exclude **/workspace-mobile.json --exclude \
**/.obsidian/cache/**"

    rclone bisync "$1" "$REMOTE:$2" \
        --verbose --remove-empty-dirs --recover \
        --conflict-resolve newer \
        --backup-dir2 "$BACKUP_DIR" \
        --compare size,modtime \
        $EXCLUDES \
        >> "$LOGfile" 2>&1
}

run_standard_job() {
    # STANDARD FOLDERS (Camera/Docs/Downloads)
    EXCLUDES="--exclude .sync/** --exclude .trash* --exclude *.download \
        --exclude *.conflict* --exclude .DS_Store --exclude \
.thumbnails/**"

    rclone bisync "$1" "$REMOTE:$2" \
        --verbose --remove-empty-dirs --recover \
        --conflict-resolve newer \
        --compare size,modtime \
        $EXCLUDES \
        >> "$LOGfile" 2>&1
}
```

```

execute_job() {
    echo "--- Sync Started at $(date) ---" >> "$LOGfile"
    run_job "$HOME/storage/shared/Obsidian" "Obsidian"
    run_standard_job "$HOME/storage/downloads" "Synced Downloads"
    run_standard_job "$HOME/storage/dcim/Screenshots" "Synced Screenshots"
    run_standard_job "$HOME/storage/shared/Documents" "Synced Documents"
    run_standard_job "$HOME/storage/dcim/Camera" "Android Camera"
    echo "--- Sync Finished at $(date) ---" >> "$LOGfile"
}

# PROCESS 1: Smart Debouncer (10s delay)
(
    while true; do
        if [ -f "$TRIGGER" ]; then
            sleep 10
            CURRENT=$(date +%s); FILE=$(stat -c %Y "$TRIGGER")
            if [ $((CURRENT - FILE)) -ge 10 ]; then
                rm -f "$TRIGGER"; execute_job
            fi
        else
            sleep 2
        fi
    done
) &

# PROCESS 2: Timer (15 Mins)
(while true; do sleep 900; execute_job; done) &

# PROCESS 3: Watcher (Includes Camera)
inotifywait -m -r -e close_write,moved_to,create,delete \
    "/storage/emulated/0/Download" \
    "/storage/emulated/0/DCIM/Screenshots" \
    "/storage/emulated/0/Documents" \
    "/storage/emulated/0/Obsidian" \
    "/storage/emulated/0/DCIM/Camera" \
    2>/dev/null | while read path action file; do
        touch "$TRIGGER"
    done
EOF
chmod +x ~/.termux/boot/start_sync.sh

```

B. The Widget (Pull_Cloud.sh)

Foreground execution. Does not hang on black screen.

Bash

```

mkdir -p ~/.shortcuts
cat << 'EOF' > ~/.shortcuts/Pull_Cloud.sh
#!/data/data/com.termux/files/usr/bin/bash

# 1. KILL Background Tasks (Stop the watcher so we can sync safely)
pkill -f "inotifywait"
pkill -f "start_sync.sh"
pkill -f "rclone"
termux-toast "Syncing... Please wait"

# 2. SETUP VARIABLES
REMOTE="filejump"
BACKUP_DIR="filejump:Bisync_Archive/Obsidian"
LOGfile="$HOME/sync_log.txt"

# 3. DEFINE SYNC JOBS
run_job() {
    EXCLUDES="--exclude .sync/** --exclude .trash* --exclude .dtrash* \
        --exclude *.crdownload --exclude *.tmp --exclude \
*.opdownload \
        --exclude *.part --exclude *.conflict* --exclude \
**/workspace.json \
        --exclude **/workspace-mobile.json --exclude \
**/.obsidian/cache/**"

    rclone bisync "$1" "$REMOTE:$2" \
        --verbose --remove-empty-dirs --recover \
        --conflict-resolve newer \
        --backup-dir2 "$BACKUP_DIR" \
        --compare size,modtime \
        $EXCLUDES \
        >> "$LOGfile" 2>&1
}

run_standard_job() {
    EXCLUDES="--exclude .sync/** --exclude .trash* --exclude *.download \
        --exclude *.conflict* --exclude .DS_Store --exclude \
.thumbnails/**"

    rclone bisync "$1" "$REMOTE:$2" \
        --verbose --remove-empty-dirs --recover \
        --conflict-resolve newer \
        --compare size,modtime \
        $EXCLUDES \
        >> "$LOGfile" 2>&1
}

# 4. RUN THE SYNC
echo "--- Manual Sync Started at $(date) ---" >> "$LOGfile"

```

```
run_job "$HOME/storage/shared/Obsidian" "Obsidian"
run_standard_job "$HOME/storage/downloads" "Synced Downloads"
run_standard_job "$HOME/storage/dcim/Screenshots" "Synced Screenshots"
run_standard_job "$HOME/storage/shared/Documents" "Synced Documents"
run_standard_job "$HOME/storage/dcim/Camera" "Android Camera"
echo "--- Manual Sync Finished at $(date) ---" >> "$LOGfile"

# 5. RESTART THE WATCHER (In Background – Don't wait for it!)
~/.termux/boot/start_sync.sh > /dev/null 2>&1 &

# 6. NOTIFY AND CLOSE
termux-toast "FileJump Sync Complete ✅"
exit 0
EOF
chmod +x ~/.shortcuts/Pull_Cloud.sh
```

C. Android Permissions Checklist (Do Not Forget)

1. **Storage:** `termux-setup-storage` must be run.
2. **API:** `pkg install termux-api` must be installed.
3. **Boot: Termux:Boot** app must be installed and launched once.
4. **Battery:** Set **Termux** and **Termux:Boot** to "Unrestricted" or "Don't Optimize" in Android Settings -> Apps -> Battery.