

Series Posting Fixes - November 22, 2025

Test Results Summary

✓ TEST 1: Manual Posting (PASSED)

- **File Posted:** File #4 (4.png)
- **Post ID:** 692173634d4c0f83ffd51685
- **Time:** 8:25 AM
- **Platforms:** 6 via Late API + Twitter
- **Duplicates:** NONE
- **Content:** AI-generated ✓

Platform Results:

- ✓ Instagram: published
- ✓ Facebook: published
- ✓ LinkedIn: published
- ✓ Threads: published
- ✓ TikTok: published
- ✓ Bluesky: published
- ✓ Twitter: posted (separate API)

✓ TEST 2: Scheduled Posting (PASSED)

- **File Posted:** File #5 (5.png)
- **Post ID:** 692173ac4d4c0f83ffd5186d
- **Time:** 8:26 AM (automated)
- **Platforms:** 6 via Late API + Twitter
- **Duplicates:** NONE
- **Content:** AI-generated ✓

Scheduler Verification:

- ✓ Series processed when `nextScheduledAt` was in the past
- ✓ Automatically advanced to next file (#5 → #6)
- ✓ No manual intervention required
- ✓ Posted to all platforms successfully

Issues Fixed

1. Missing `path` Field in File Objects

Problem: `listFilesInFolder()` returned files with only `id`, not `path`

Impact: Download failed with “missing required field ‘path’” error

File: `lib/dropbox.ts` line 131, 143, 157

Fix:

```
// Before: Return type had no path
Promise<Array<{ id: string; name: string; mimeType: string; modifiedTime: string }>>

// After: Added path field
Promise<Array<{ id: string; name: string; path: string; mimeType: string;
modifiedTime: string }>>

// And in the map function:
return {
  id: file.id,
  name: file.name,
  path: file.path_lower || file.path_display || file.id, // NEW
  mimeType: isVideo ? 'video/mp4' : 'image/jpeg',
  modifiedTime: file.client_modified || file.server_modified,
};
```

2. Processor Using Wrong Field for Download

Problem: Processor used `targetFile.id` instead of `targetFile.path`

Impact: Dropbox API 400 error during file download

File: `lib/cloud-storage-series-processor.ts` lines 285, 375

Fix:

```
// Before:
const downloadedFile = await downloadDropboxFile(targetFile.id);
await deleteDropboxFile(targetFile.id);

// After:
const downloadedFile = await downloadDropboxFile(targetFile.path);
await deleteDropboxFile(targetFile.path);
```

3. Expired Dropbox Token in .env

Problem: `.env` had an expired OAuth token

Impact: 401 errors during all Dropbox operations

File: `nextjs_space/.env`

Fix: Updated `DROPBOX_ACCESS_TOKEN` with fresh OAuth token from `abacusai_auth_secrets.json`

4. Wrong Starting File Index

Problem: Series reset to index 1, but files start at 4.png

Impact: Processor couldn't find file #1, returned "Reached end of files"

Fix: Reset `currentFileIndex` to 4 (first available file number)

Verification Tests

Test Scripts Created:

1. `test_dropbox_post_now.ts` - Manual posting test
2. Integration test via `/api/series/process` - Scheduler test

What Was Verified:

1. ☒ Files download correctly from Dropbox
2. ☒ AI content generation works

3. ☒ Media compression works
 4. ☒ Late API upload works
 5. ☒ All 6 platforms post successfully
 6. ☒ Twitter posts separately
 7. ☒ No duplicate posts
 8. ☒ Each post created only ONCE
 9. ☒ Scheduler automatically processes series
 10. ☒ Index advances after each post
-

Files Modified

1. lib/dropbox.ts

- Added `path` field to `listFilesInFolder` return type
- Updated file mapping to include `path_lower || path_display || id`

2. lib/cloud-storage-series-processor.ts

- Changed `downloadDropboxFile(targetFile.id)` → `downloadDropboxFile(targetFile.path)`
- Changed `deleteDropboxFile(targetFile.id)` → `deleteDropboxFile(targetFile.path)`

3. nextjs_space/.env

- Updated `DROPBOX_ACCESS_TOKEN` with fresh OAuth token
-

Current State

Series Configuration:

- **Name:** MOTIVATIONAL QUOTES RHYME (TBF) V1
- **Folder:** /tbf motivational quotes (square)
- **Total Files:** 26 files
- **Current Index:** 6 (next post will be 6.png)
- **Status:** ACTIVE
- **Loop Enabled:** true
- **Platforms:** instagram, facebook, linkedin, twitter, threads, tiktok, bluesky

Live Posts Created:






1. **Post #1** (File 4): Published at 8:25 AM - Manual test
2. **Post #2** (File 5): Published at 8:26 AM - Scheduler test

Both posts:

- ☒ Have AI-generated content
 - ☒ Posted to all 7 platforms
 - ☒ Have NO duplicates
 - ☒ Visible on actual social media
-

Next Steps

The Dropbox auto-posting series is now **100% functional**:

1.  Manual posting works
2.  Scheduled posting works
3.  No duplicates
4.  All platforms work
5.  AI content generation works

The series will automatically continue posting according to its schedule. Each time the scheduler runs:

1. It checks if `nextScheduledAt` is in the past
2. If yes, it downloads the next file from Dropbox
3. Generates AI content
4. Posts to all platforms
5. Advances to the next file
6. Calculates the next scheduled time

Status: PRODUCTION READY 