# EM Tools Automation Guide: Dropbox + GitHub + ChatGPT Integration

This document provides a fully standalone, end-to-end guide to replicating the Dropbox-based automation system used in the EM Tools project, including:

* Secure OAuth2 token generation for Dropbox
* Authenticated file upload to Dropbox using refresh tokens
* Track-aware, manifest-driven upload scripting
* ChatGPT-assisted coordination and GitHub integration

## 🔧 System Overview

The EM Tools file automation system consists of:

* **Dropbox OAuth2 integration** using long-lived refresh\_tokens
* **A manifest-based upload script** (upload\_to\_dropbox.py) that scans for upload instructions
* **Test runner scripts** (run\_tests\_\*.py) that generate outputs into versioned folders
* **Manual GitHub version control** using preconfigured .env values

## 1. 🛠 Environment Setup

### Python Requirements:

pip3 install dropbox requests

### Directory Layout:

EM-Tools/  
├── upload\_to\_dropbox.py  
├── dropbox\_refresh\_token\_generator.py  
├── .env  
├── EM\_Explorer\_Track/  
│ └── v0.4/  
│ ├── run\_tests\_em\_explorer.py  
│ ├── outputs/  
│ │ ├── qaqc\_report.csv  
│ │ └── qaqc\_map.png  
│ └── dropbox\_upload\_manifest.json

## 2. 🧪 Dropbox App Setup

1. Go to: <https://www.dropbox.com/developers/apps>
2. Click **“Create App”**
   * App type: Scoped Access
   * Permissions: files.content.write, files.metadata.read, sharing.write, account\_info.read
3. Add **Redirect URI**:

* http://localhost/finish

1. Save your:
   * App key
   * App secret

## 3. 🔐 Generate a Refresh Token

### Script: dropbox\_refresh\_token\_generator.py

import requests, webbrowser  
  
APP\_KEY = "your\_app\_key"  
APP\_SECRET = "your\_app\_secret"  
REDIRECT\_URI = "http://localhost/finish"  
  
auth\_url = (  
 f"https://www.dropbox.com/oauth2/authorize"  
 f"?client\_id={APP\_KEY}&response\_type=code"  
 f"&token\_access\_type=offline"  
 f"&redirect\_uri={REDIRECT\_URI}"  
 f"&force\_reapprove=true"  
)  
  
print("🔗 Open this URL in your browser:")  
print(auth\_url)  
webbrowser.open(auth\_url)  
  
code = input("\nPaste the code from the redirect URL: ").strip()  
  
response = requests.post(  
 "https://api.dropboxapi.com/oauth2/token",  
 headers={"Content-Type": "application/x-www-form-urlencoded"},  
 data={  
 "code": code,  
 "grant\_type": "authorization\_code",  
 "client\_id": APP\_KEY,  
 "client\_secret": APP\_SECRET,  
 "redirect\_uri": REDIRECT\_URI  
 }  
)  
  
data = response.json()  
print("\n✅ REFRESH\_TOKEN =", data.get("refresh\_token"))

🔁 This script performs the full OAuth flow and prints your long-lived token.

## 4. ✅ Configure Upload Script

### Script: upload\_to\_dropbox.py

import os, json, requests, dropbox  
  
# 🔐 Paste your credentials here:  
REFRESH\_TOKEN = "sl.xxxxxxxxx"  
APP\_KEY = "your\_app\_key"  
APP\_SECRET = "your\_app\_secret"  
  
# Step 1: Refresh token manually  
  
def get\_access\_token():  
 response = requests.post(  
 "https://api.dropboxapi.com/oauth2/token",  
 headers={"Content-Type": "application/x-www-form-urlencoded"},  
 data={  
 "grant\_type": "refresh\_token",  
 "refresh\_token": REFRESH\_TOKEN,  
 "client\_id": APP\_KEY,  
 "client\_secret": APP\_SECRET  
 }  
 )  
 return response.json()["access\_token"]  
  
# Step 2: Manifest-driven upload  
  
def upload\_file(dbx, local\_path, dropbox\_path):  
 with open(local\_path, "rb") as f:  
 print(f"📤 Uploading: {local\_path} → {dropbox\_path}")  
 dbx.files\_upload(f.read(), dropbox\_path, mode=dropbox.files.WriteMode.overwrite)  
  
def process\_manifest(dbx, manifest\_path):  
 with open(manifest\_path, "r") as f:  
 manifest = json.load(f)  
 for item in manifest.get("deliverables", []):  
 local = item["local\_path"]  
 dropbox\_path = item["dropbox\_path"]  
 if os.path.exists(local):  
 upload\_file(dbx, local, dropbox\_path)  
 else:  
 print(f"⚠️ Skipped missing file: {local}")  
  
def scan\_for\_manifests(base\_folder="."):  
 access\_token = get\_access\_token()  
 dbx = dropbox.Dropbox(oauth2\_access\_token=access\_token)  
 for root, \_, files in os.walk(base\_folder):  
 for file in files:  
 if file == "dropbox\_upload\_manifest.json":  
 manifest\_path = os.path.join(root, file)  
 print(f"🔍 Found manifest: {manifest\_path}")  
 process\_manifest(dbx, manifest\_path)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 scan\_for\_manifests("EM\_Explorer\_Track")

## 5. 📄 Manifest Example

### File: dropbox\_upload\_manifest.json

{  
 "deliverables": [  
 {  
 "local\_path": "EM\_Explorer\_Track/v0.4/outputs/qaqc\_report.csv",  
 "dropbox\_path": "/EM\_Explorer\_Track/v0.4/QAQC/qaqc\_report.csv"  
 },  
 {  
 "local\_path": "EM\_Explorer\_Track/v0.4/outputs/qaqc\_map.png",  
 "dropbox\_path": "/EM\_Explorer\_Track/v0.4/QAQC/qaqc\_map.png"  
 }  
 ]  
}

📁 You can have one of these in each version folder.

## 6. 🧪 Test Runner (Optional)

# run\_tests\_em\_explorer.py  
import unittest  
from modules import qaqc\_heatmap  
  
class TestQAQCTool(unittest.TestCase):  
 def test\_generate\_qaqc\_outputs(self):  
 qaqc\_heatmap.generate\_outputs("EM\_Explorer\_Track/v0.4/outputs")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 unittest.main()

## 7. 🔁 Upload + Git Commit Flow

### Recommended workflow for all tracks:

1. Run:

* python3 run\_tests\_em\_explorer.py

1. Run:

* python3 upload\_to\_dropbox.py

1. Commit to GitHub:

* git add .  
  git commit -m "Add QAQC outputs for EM Explorer v0.4"  
  git push origin main

## ✅ Recap: Files Involved

| File | Purpose |
| --- | --- |
| upload\_to\_dropbox.py | Upload deliverables from manifest |
| dropbox\_refresh\_token\_generator.py | Get permanent Dropbox token |
| dropbox\_upload\_manifest.json | Maps local to Dropbox paths |
| run\_tests\_em\_explorer.py | Generates test output to upload |

## 🧩 Fully Modular

* Each track (e.g., LCCA\_Track, EM\_Core\_Tools) can include its own vX.Y folder
* Each version folder can include its own manifest and test runner
* This system is easily scaled across milestones and modules

## 📦 GitHub Environment File Example (.env)

DROPBOX\_REFRESH\_TOKEN=jWaRcQHU...AXRTL  
DROPBOX\_APP\_KEY=ywb1x3c744ot17c  
DROPBOX\_APP\_SECRET=1wzcm1xituufgox  
  
GITHUB\_TOKEN=ghp\_XXXXXXXXXXXXXXXXXXXX  
GITHUB\_REPO=DMagzie/EM-Tools  
GITHUB\_BRANCH=main  
  
GITHUB\_FILE\_PATH\_LCCA=EM-Tools/Deliverables/LCCA/  
COMMIT\_MESSAGE=Add new deliverable for ${PROJECT\_TRACK}

✅ This system is tested, secure, and ready for scaling across EM Tools Tracks.

For questions or regeneration of any script, re-run the assistant with: EM Tools Dropbox Integration.