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_token`s
- **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"**
3. App type: Scoped Access
4. Permissions: `files.content.write`, `files.metadata.read`, `sharing.write`, `account_info.read`
5. Add **Redirect URI**:

```
http://localhost/finish
```

6. Save your:
7. App key
8. App secret

---

3. 🔐 Generate a Refresh Token

Script: `dropbox_refresh_token_generator.py`

```python
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)

```python
# 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
   ```

2. Run:

   ```
   python3 upload_to_dropbox.py
   ```

3. 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_XXXXXXXXXXXXXXXXXXX
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`.