

KBFC Data Import ETL Toolkit Installation Guide

This document outlines how to set up the KBFC Data Import ETL Toolkit on a Windows workstation. It covers software requirements, installation steps, configuration, helper/launcher setup, and testing.

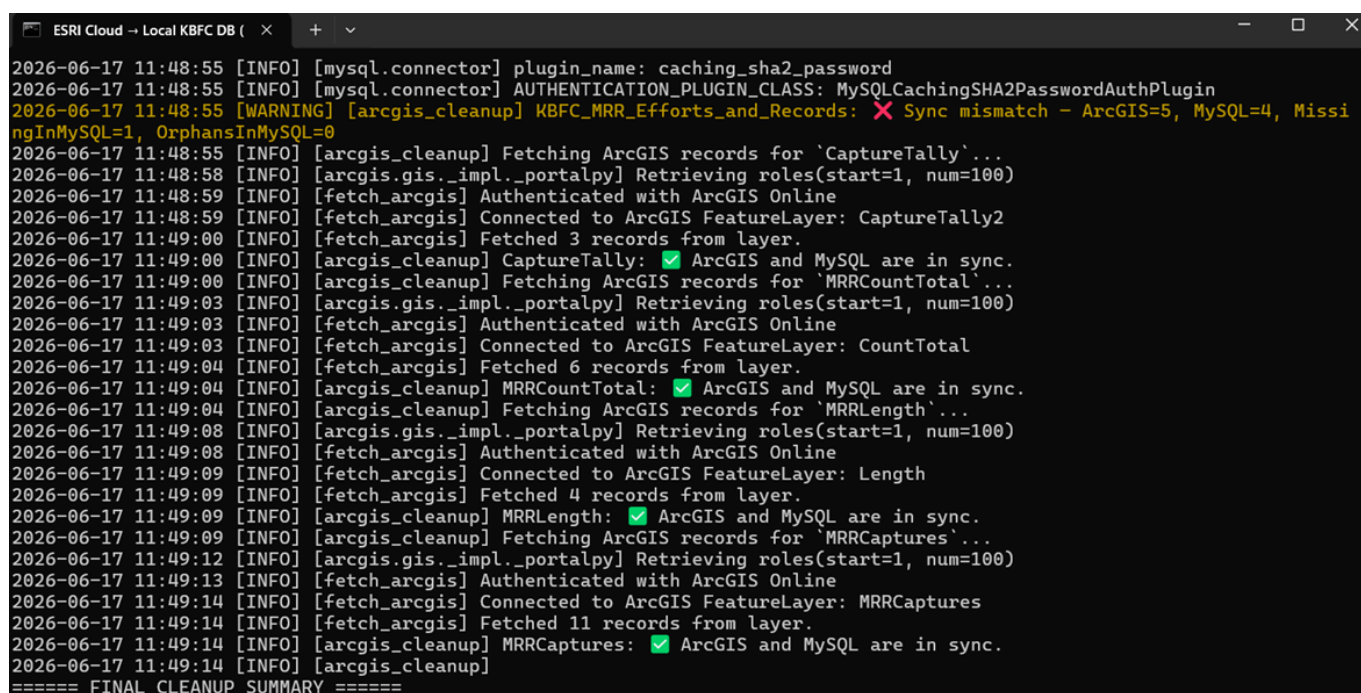
Areas to Confirm Before Installation

- Final install path. This guide assumes the toolkit is installed at `C:\Scripts\KBFC_Webhooks`.
- Target API environment: test API or production API.
- ArcGIS account and Survey123 layer URLs for the target organization.
- MySQL username, password, schema name, and whether the workstation uses port `3306`.
- Whether the scanner Streamlit launcher is required on this workstation.
- Whether desktop shortcuts and the custom `myapp://` URL protocol should be installed.

What This Project Installs

ArcGIS Import Tool

- What it is: A local import tool for moving Survey123/ArcGIS feature layer data into MySQL.
- Description: Connects to configured ArcGIS layers, imports records into local database tables, updates changed records, and can clean up local records that no longer exist in the ArcGIS source.



```
ESRI Cloud -> Local KBFC DB
2026-06-17 11:48:55 [INFO] [mysql.connector] plugin_name: caching_sha2_password
2026-06-17 11:48:55 [INFO] [mysql.connector] AUTHENTICATION_PLUGIN_CLASS: MySQLCachingSHA2PasswordAuthPlugin
2026-06-17 11:48:55 [WARNING] [arcgis_cleanup] KBFC_MRR_Efforts_and_Records: ✗ Sync mismatch - ArcGIS=5, MySQL=4, MissingInMySQL=1, OrphansInMySQL=0
2026-06-17 11:48:55 [INFO] [arcgis_cleanup] Fetching ArcGIS records for 'CaptureTally'...
2026-06-17 11:48:58 [INFO] [arcgis.gis.impl.portalpy] Retrieving roles(start=1, num=100)
2026-06-17 11:48:59 [INFO] [fetch_arcgis] Authenticated with ArcGIS Online
2026-06-17 11:48:59 [INFO] [fetch_arcgis] Connected to ArcGIS FeatureLayer: CaptureTally2
2026-06-17 11:49:00 [INFO] [fetch_arcgis] Fetched 3 records from layer.
2026-06-17 11:49:00 [INFO] [arcgis_cleanup] CaptureTally: ✓ ArcGIS and MySQL are in sync.
2026-06-17 11:49:00 [INFO] [arcgis_cleanup] Fetching ArcGIS records for 'MRRCountTotal'...
2026-06-17 11:49:03 [INFO] [arcgis.gis.impl.portalpy] Retrieving roles(start=1, num=100)
2026-06-17 11:49:03 [INFO] [fetch_arcgis] Authenticated with ArcGIS Online
2026-06-17 11:49:03 [INFO] [fetch_arcgis] Connected to ArcGIS FeatureLayer: CountTotal
2026-06-17 11:49:04 [INFO] [fetch_arcgis] Fetched 6 records from layer.
2026-06-17 11:49:04 [INFO] [arcgis_cleanup] MRRCountTotal: ✓ ArcGIS and MySQL are in sync.
2026-06-17 11:49:04 [INFO] [arcgis_cleanup] Fetching ArcGIS records for 'MRRLength'...
2026-06-17 11:49:08 [INFO] [arcgis.gis.impl.portalpy] Retrieving roles(start=1, num=100)
2026-06-17 11:49:08 [INFO] [fetch_arcgis] Authenticated with ArcGIS Online
2026-06-17 11:49:09 [INFO] [fetch_arcgis] Connected to ArcGIS FeatureLayer: Length
2026-06-17 11:49:09 [INFO] [fetch_arcgis] Fetched 4 records from layer.
2026-06-17 11:49:09 [INFO] [arcgis_cleanup] MRRLength: ✓ ArcGIS and MySQL are in sync.
2026-06-17 11:49:09 [INFO] [arcgis_cleanup] Fetching ArcGIS records for 'MRRCaptures'...
2026-06-17 11:49:12 [INFO] [arcgis.gis.impl.portalpy] Retrieving roles(start=1, num=100)
2026-06-17 11:49:13 [INFO] [fetch_arcgis] Authenticated with ArcGIS Online
2026-06-17 11:49:14 [INFO] [fetch_arcgis] Connected to ArcGIS FeatureLayer: MRRCaptures
2026-06-17 11:49:14 [INFO] [fetch_arcgis] Fetched 11 records from layer.
2026-06-17 11:49:14 [INFO] [arcgis_cleanup] MRRCaptures: ✓ ArcGIS and MySQL are in sync.
2026-06-17 11:49:14 [INFO] [arcgis_cleanup]
===== FINAL CLEANUP SUMMARY =====
```

KBFC API Sync Tool

- What it is: A local export/sync tool for sending validated MySQL records to the KBFC API.
- Description: Reads prepared records from the local database, validates required fields and formats, sends records to the configured API environment, and tracks API record IDs so future syncs can update existing records.

```

Local KBFC DB → Collaborative
2026-06-17 11:52:58 [INFO] [validate_project] [Project] Project USGS_pre2024 already known to exist.
2026-06-17 11:52:58 [INFO] [process_mmr_efforts] Mapping MRR-EFFORT records
2026-06-17 11:52:58 [INFO] [process_mmr_efforts] Find Record ID MRR-EFFORT
2026-06-17 11:52:58 [INFO] [api_client] GET https://test.kbfishc.org/effort/7A785C82-EFDE-4853-BECB-0EAE993E44E8 -> Status: 200
2026-06-17 11:52:58 [INFO] [process_mmr_efforts] Skipping id 7A785C82-EFDE-4853-BECB-0EAE993E44E8 -5ea773b3-50d2-49f0-8b
c6-17587843e6d5 - Already up to date (EditDate: 2025-06-27 17:17:26+00:00, datemodified: 2026-04-27 13:11:30.267000+00:
00)
2026-06-17 11:52:58 [INFO] [process_mmr_efforts] Mapping MRR-CAPTURES records
2026-06-17 11:52:58 [INFO] [process_mmr_efforts] Find Record ID MRR-CAPTURES
2026-06-17 11:52:58 [INFO] [api_client] GET https://test.kbfishc.org/mrrcaptures/D4798F4A-F4BB-469E-8C34-6B0D0F57C922 ->
Status: 200
2026-06-17 11:52:58 [INFO] [process_mmr_efforts] Skipping id D4798F4A-F4BB-469E-8C34-6B0D0F57C922 -057ea4a5-b90b-4a07-8b
db-7ed5d499aa39 - Already up to date (EditDate: 2025-06-27 17:17:26+00:00, datemodified: 2026-04-27 13:11:30.497000+00:
00)
2026-06-17 11:52:58 [INFO] [api_client] GET https://test.kbfishc.org/site/11R -> Status: 200
2026-06-17 11:52:58 [INFO] [validate_site] [Site] Site 11R exists in API.
2026-06-17 11:52:58 [INFO] [validate_project] [Project] Project USGS_pre2024 already known to exist.
2026-06-17 11:52:58 [INFO] [process_mmr_efforts] Mapping MRR-EFFORT records
2026-06-17 11:52:58 [INFO] [process_mmr_efforts] Find Record ID MRR-EFFORT
2026-06-17 11:52:58 [INFO] [api_client] GET https://test.kbfishc.org/effort/DD99F672-B419-478C-8609-D95EC2B37777 -> Stat
us: 200
2026-06-17 11:52:58 [INFO] [process_mmr_efforts] Skipping id DD99F672-B419-478C-8609-D95EC2B37777 -de5abdcf-0d04-4ff4-a6
62-94f616d0695a - Already up to date (EditDate: 2025-06-27 17:17:10+00:00, datemodified: 2026-04-27 13:11:30.640000+00:
00)
2026-06-17 11:52:58 [INFO] [process_mmr_efforts] Mapping MRR-CAPTURES records
2026-06-17 11:52:58 [INFO] [process_mmr_efforts] Find Record ID MRR-CAPTURES
2026-06-17 11:52:58 [INFO] [process_mmr_efforts] Inserting new record for id None - Missing Mapping Id, POST insert
2026-06-17 11:52:58 [INFO] [post_mrrcapture] [mrrcaptures] Posting MRR capture for PITTag: 3DD.003C08DF8D
2026-06-17 11:52:58 [INFO] [api_client] POST https://test.kbfishc.org/mrrcaptures -> Status: 400

```

Scanner File Loader

- What it is: A local file loading tool for PIT tag scanner workflows.
- Description: Loads scanner output files, prepares detection records, and supports the detection import process when scanner workflows are enabled for the workstation.

Import Remote Detection Files → Internal DB → Collaborative KBFC DB

Select Folder

Choose Folder

*.txt files in the chosen folder

Choose Folder containing scanner .txt files.

No .txt files loaded.

Select: Site / Deployment

Don't see your Site or Deployment? Update them on your portal: [Open site](#)

Site

11R (USGS) • UPKB

Deployment

1

Selection Status

Pick a folder on the left to list .txt files.

Current selection

Site: 11R (UUKLOxySpr5)

Org: USGS • Type: Detection • Status: Active • Lat: 42.39883804321289 • Lon: -121.82357025146484 • Watershed: UPKB

Deployment: —

Upload to KBFC Database

Process & Post records

Web Log Review Tool

- What it is: A browser-based log review page.
- Description: Provides a local way to review import results, validation errors, API responses, skipped records, and troubleshooting details after a run.

KBFC Monitoring - Summary Page

Open KBFC Data Integration Workflow

Import Remote Detection Files

Validation Rules by Survey & Route

ESRI Cloud → Local KBFC DB Sync Summary

Last Updated

2026-06-17 11:49:39 AM PDT

Status

Issues

Errors

0

Out of Sync

Yes

Show Detailed Sync Status

View Raw Log

Local KBFC DB → Collaborative KBFC DB Summary

Last Upload

2026-06-17 11:52:59 AM

Survey

MRR Efforts and Records

Total Records

22

New Inserted/Updated

0

Failed

5

Up-to-Date Records

17

2026-06-17 11:52:59 AM

Remote Deployments and Event Log

0

0

0

0

Errors From Local KBFC DB → Collaborative KBFC DB (last 200)

View Raw Log

Timestamp	Level	Script	Survey	Type	GlobalID	ObjectID	Message
2026-06-17 11:52:58 AM	ERROR	process_mrr_efforts	MRR Efforts and Records	MRR- CAPTURES	5ea773b3-50d2-49b0-8bc5-17387843e5d5	2	[E33a]mrrcaptures) Failed to post MRR capture for 300.003070d274. Status: 400 Response text: {"statusCode":400,"error":"Bad Request","message":{"processdatetime: Datetime must be between Fri Jun 27 2025 09:59:00 GMT-0700 (Pacific Daylight Time) and Fri Jun 27 2025 10:03:44 GMT-0700 (Pacific Daylight Time)"}}
2026-06-17 11:52:58 AM	ERROR	process_mrr_efforts	MRR Efforts and Records	MRR- CAPTURES	de5abdcf-0d34-4f84-a662-9485160d0959	4	[E33a]mrrcaptures) Failed to post MRR capture for 300.003038b0f0. Status: 400 Response text: {"statusCode":400,"error":"Bad Request","message":{"processdatetime: Datetime must be between Fri Jun 27 2025 10:12:00 GMT-0700 (Pacific Daylight Time) and Fri Jun 27 2025 10:14:00 GMT-0700 (Pacific Daylight Time)"}}

```
2026-06-17 11:49:16 [INFO] [api_client] API client initialized with xapikey only.
2026-06-17 11:49:16 [INFO] [arcgis_import] ArcGIS ingest reference validation is enabled (Project/Site checks).
2026-06-17 11:49:16 [INFO] [arcgis_import] Starting ArcGIS → MySQL import for 5 layers.
2026-06-17 11:49:16 [INFO] [arcgis_import] Fetching ArcGIS layer for 'KBFC_MRR_Efforts_and_Records'...
2026-06-17 11:49:20 [INFO] [arcgis.gis_impl_portalpy] Retrieving roles(start=1, num=100)
2026-06-17 11:49:20 [INFO] [fetch_arcgis] Authenticated with ArcGIS Online
2026-06-17 11:49:21 [INFO] [fetch_arcgis] Connected to ArcGIS FeatureLayer: KBFC_MRR_Efforts_and_Records_Form_V2_2
2026-06-17 11:49:21 [INFO] [fetch_arcgis] Fetched 5 records from layer.
2026-06-17 11:49:21 [INFO] [arcgis_import] KBFC_MRR_Efforts_and_Records: 5 records fetched.
2026-06-17 11:49:21 [INFO] [api_client] GET https://test.kbfishc.org/project/USGS_pre2024 -> Status: 200
2026-06-17 11:49:21 [INFO] [api_client] GET https://test.kbfishc.org/project/123 -> Status: 404
2026-06-17 11:49:21 [WARNING] [arcgis_import] KBFC_MRR_Efforts_and_Records: Skipping globalid=48bca57c-cea0-413a-8dc9-50176e22b11a due to reference validation failure: unknown ProjectID '123'
2026-06-17 11:49:21 [INFO] [arcgis_import] Inserting records into 'KBFC_MRR_Efforts_and_Records' ...
2026-06-17 11:49:21 [INFO] [mysql.connector] package: mysql.connector.plugins
2026-06-17 11:49:21 [INFO] [mysql.connector] plugin_name: caching_sha2_password
2026-06-17 11:49:21 [INFO] [mysql.connector] AUTHENTICATION_PLUGIN_CLASS: MySQLCachingSHA2PasswordAuthPlugin
2026-06-17 11:49:21 [INFO] [arcgis_import] KBFC_MRR_Efforts_and_Records: Inserted:0 , Updated:4
2026-06-17 11:49:21 [WARNING] [arcgis_import] KBFC_MRR_Efforts_and_Records: Sync mismatch. ArcGIS=5, MySQL=4, Missing in MySQL=1, Orphans in MySQL=0
2026-06-17 11:49:21 [INFO] [arcgis_import] Fetching ArcGIS layer for 'CaptureTally'...
2026-06-17 11:49:24 [INFO] [arcgis.gis_impl_portalpy] Retrieving roles(start=1, num=100)
2026-06-17 11:49:25 [INFO] [fetch_arcgis] Authenticated with ArcGIS Online
2026-06-17 11:49:25 [INFO] [fetch_arcgis] Connected to ArcGIS FeatureLayer: CaptureTally2
2026-06-17 11:49:25 [INFO] [fetch_arcgis] Fetched 3 records from layer.
2026-06-17 11:49:25 [INFO] [arcgis_import] CaptureTally: 3 records fetched.
2026-06-17 11:49:25 [INFO] [arcgis_import] Inserting records into 'CaptureTally'...
2026-06-17 11:49:26 [INFO] [arcgis_import] CaptureTally: Inserted:0 , Updated:3
2026-06-17 11:49:26 [INFO] [arcgis_import] CaptureTally: ArcGIS and MySQL are in sync.
2026-06-17 11:49:26 [INFO] [arcgis_import] Fetching ArcGIS layer for 'MRRCountTotal'...
2026-06-17 11:49:29 [INFO] [arcgis.gis_impl_portalpy] Retrieving roles(start=1, num=100)
2026-06-17 11:49:29 [INFO] [fetch_arcgis] Authenticated with ArcGIS Online
2026-06-17 11:49:29 [INFO] [fetch_arcgis] Connected to ArcGIS FeatureLayer: CountTotal
2026-06-17 11:49:30 [INFO] [fetch_arcgis] Fetched 6 records from layer.
2026-06-17 11:49:30 [INFO] [arcgis_import] MRRCountTotal: 6 records fetched.
2026-06-17 11:49:30 [INFO] [arcgis_import] Inserting records into 'MRRCountTotal'...
2026-06-17 11:49:30 [INFO] [arcgis_import] MRRCountTotal: Inserted:0 , Updated:6
2026-06-17 11:49:30 [INFO] [arcgis_import] MRRCountTotal: ArcGIS and MySQL are in sync.
2026-06-17 11:49:30 [INFO] [arcgis_import] Fetching ArcGIS layer for 'MRRLength'...
2026-06-17 11:49:33 [INFO] [arcgis.gis_impl_portalpy] Retrieving roles(start=1, num=100)
2026-06-17 11:49:33 [INFO] [fetch_arcgis] Authenticated with ArcGIS Online
2026-06-17 11:49:34 [INFO] [fetch_arcgis] Connected to ArcGIS FeatureLayer: MRRLength
```

Workflow Reference Page

- What it is: A browser-based workflow reference page for understanding the data integration process.
- Description: Summarizes the major import, validation, sync, and troubleshooting steps so users can review how the toolkit moves data between systems.

KBFC Import Tool - Summary Page

KBFC Data Integration Workflow

Debug Overview

Follow these steps if something looks wrong or data is missing:

1. Check Sync Status (sync_status.html) to see if ArcGIS and MySQL record counts match.

2. Look at the Import Log (arcgis_import.html) to confirm when the last ArcGIS → MySQL import ran and whether it had errors.

3. Check the API Upload Log (raw_api_log.html) for errors when pushing data to the KBFC API.

4. If records are missing from KBFC API but present in MySQL, it is likely an API upload issue.

5. If records are missing in MySQL, verify ArcGIS data and rerun the import script.

6. For persistent errors, review the raw logs or database to confirm whether the globalid matches ArcGIS records.

View Workflow Diagram

Overview

This system automatically transfers data from **ESRI Cloud** to the **Local KBFC database** and then uploads it to the **KBFC API**.

The process runs in steps: fetching new records, saving them in the database, checking for differences, and logging results.

1. ESRI Cloud Data Fetching

Goal: Retrieve updated records from ESRI Cloud Feature Services.

1. fetch_arcgis_data() queries ESRI Cloud layers using REST API with authentication token.

2. Each configured ArcGIS layer in config.yml has survey, table, index, layer_url, and table_name.

3. Data is retrieved in paged requests (default 1,000 records per page).

4. Supports full sync (1=1) or incremental sync (editDate >= cutoff).

2. Data Transformation (Mapping)

Local Launcher Tool

3 / 15

- What it is: An optional helper that allows desktop shortcuts or browser links to start local tools.
- Description: Starts configured local utilities, such as the scanner file loader, without requiring the user to run commands manually.

Software Requirements

Install these before configuring the project:

- Windows 10 or Windows 11.
- Git for Windows.
- Python 3.10.x. Existing notes reference Python **3.10.10**; use Python 3.10 unless the project is retested on a newer major/minor version.
- MySQL Server 8.x.
- MySQL Workbench 8.x.
- Visual Studio 2019 or Microsoft C++ Build Tools. This is useful for packages that may need native build support.
- Internet access to:
 - ArcGIS Online / Survey123 feature services.
 - KBFC API test or production environment.
 - Python package index during dependency installation.
- Optional scanner launcher requirements:
 - FastAPI.
 - Uvicorn.
 - Streamlit.
 - PyInstaller.

Most Python dependencies are pinned in **requirements.txt**.

Accounts and Credentials Required

Collect these before setup. Do not commit or share real values in documentation.

- ArcGIS username.
- ArcGIS password.
- KBFC API base URL.
- KBFC API key.
- MySQL host.
- MySQL port, normally **3306**.
- MySQL username.
- MySQL password.
- MySQL schema/database name, normally **kbfc**.

Recommended Folder Layout

Install the repository at:

```
C:\Scripts\KBFC_Webhooks
```

This path is recommended because the included batch files call this directory directly when they start the tools and activate the Python environment. The helper launcher and protocol registration file also assume this path by default.

If another path is used, update every path reference in:

- `autorun_arcgis_to_mysql.bat`
- `autorun_mysql_to_kbfc.bat`
- `manual_runner.bat`
- `helper.py`
- `myapp_protocol.reg`
- Desktop shortcuts.

Installation Steps

1. Install Git

Install Git for Windows, then confirm:

```
git --version
```

2. Install Python

Install Python 3.10.x and enable "Add Python to PATH" during installation.

Confirm:

```
python --version  
pip --version
```

Expected Python version:

```
Python 3.10.x
```

3. Install MySQL

Install MySQL Server 8.x and MySQL Workbench 8.x.

During setup:

- Record the root or service user password.
- Keep the default port `3306` unless the deployment requires another port.
- Confirm the MySQL service starts automatically.

Confirm MySQL is running by opening MySQL Workbench and connecting to `localhost`.

4. Clone the Repository

Clone the repository into the expected install directory. Replace `<REPOSITORY_URL>` with the project repository URL:

```
New-Item -ItemType Directory "C:\Scripts" -Force
git clone <REPOSITORY_URL> C:\Scripts\KBFC_Webhooks
cd C:\Scripts\KBFC_Webhooks
```

If the repository is already present, update it:

```
cd C:\Scripts\KBFC_Webhooks
git pull
```

5. Create the Python Virtual Environment

From the repository root:

```
python -m venv venv
.\venv\Scripts\activate
python -m pip install --upgrade pip
python -m pip install -r requirements.txt
```

Confirm key packages import:

```
python -c "import arcgis, mysql.connector, requests, yaml, streamlit;
print('Python dependencies OK')"
```

6. Create the MySQL Schema

Open MySQL Workbench and create the database/schema:

```
CREATE DATABASE IF NOT EXISTS kbfc;
USE kbfc;
```

Set `kbfc` as the default schema in Workbench before running the table scripts.

7. Create Tables and Views

Run the SQL scripts in `sql_scripts/` against the `kbfc` schema.

Standard ETL install order:

```
create_kbfc_mrr_efforts_and_records_table.sql
create_capture_tally_table.sql
create_mrr_count_total_table.sql
create_mrr_length_table.sql
create_mrr_captures_table.sql
create_kbfc_remote_deployments_table.sql
create_detection_site_configuration_table.sql
create_id_lookup.sql
create_view_Efforts_and_Records.sql
create_view_Remote_Deployments_and_Event_Log.sql
create_scanner_pittag_detections.sql
```

After running the scripts, confirm expected tables and views exist in Workbench.

8. Create `config.yml`

Create or update `config.yml` in the repository root. Use placeholders first, then replace with deployment values:

```
mysql:
  host: localhost
  port: 3306
  user: YOUR_MYSQL_USER
  password: YOUR_MYSQL_PASSWORD
  database: kbfc
  use_pure: true
  connection_timeout: 10
  get_warnings: true
  raise_on_warnings: true
  autocommit: false

arctgis_login:
  username: YOUR_ARCGIS_USERNAME
  password: YOUR_ARCGIS_PASSWORD

api:
  base_url: https://YOUR_KBFC_API_HOST
  api_key: YOUR_KBFC_API_KEY
  jwt: false

arctgis_layers:
  - survey: MRREffortsandRecords
    table_index: 0
    layer_url: YOUR_FEATURE_LAYER_URL
    table_name: KBFC_MRR_Efforts_and_Records
```

Add each required ArcGIS layer from the deployment source list. The active configuration should include only the layers intended for this workstation/environment.

9. Update Batch Files

Open these files and verify the paths match the install directory and virtual environment:

- `autorun_arcgis_to_mysql.bat`
- `autorun_mysql_to_kbfc.bat`
- `manual_runner.bat`

The batch files should use this path consistently:

```
cd /d "C:\Scripts\KBFC_Webhooks"  
call "C:\Scripts\KBFC_Webhooks\venv\Scripts\activate.bat"
```

10. Optional: Build the Helper Launcher

Only do this if the workstation needs the Streamlit scanner/file loader launcher.

Update paths in `helper.py`:

```
PYTHONW = r"C:\Scripts\KBFC_Webhooks\venv\Scripts\pythonw.exe"  
PYTHON  = r"C:\Scripts\KBFC_Webhooks\venv\Scripts\python.exe"  
STREAMLIT_APP = r"C:\Scripts\KBFC_Webhooks\kbfc_scanner\scanner_import.py"
```

Build the executable:

```
.\venv\Scripts\activate  
pyinstaller --onefile --noconsole helper.py
```

Expected output:

```
dist\helper.exe
```

Optional helper install location:

```
New-Item -ItemType Directory "<HELPER_INSTALL_DIR>" -Force  
Copy-Item ".\dist\helper.exe" "<HELPER_INSTALL_DIR>\helper.exe" -Force
```

11. Optional: Register the `myapp://` Protocol

Only do this if browser-based launcher links are required.

Update `myapp_protocol.reg` so it points to the correct `helper.exe` path.

Example for repo-local helper:

```
@="\"C:\\Scripts\\KBFC_Webhooks\\dist\\helper.exe\" \"%1\""
```

Double-click the `.reg` file and accept the Windows registry prompt.

12. Optional: Update Desktop Shortcuts

Update shortcuts to point to the correct batch files or HTML file:

- ArcGIS import shortcut: `autorun_arcgis_to_mysql.bat`
- KBFC API export shortcut: `autorun_mysql_to_kbfc.bat`
- Log reader shortcut: `kbfc_log_reader.html`

If desktop shortcuts, symbolic links, or `myapp://` protocol links are not set up, the tools can still be run directly from PowerShell using the commands in the testing section.

Testing

Run tests in this order. Stop and fix failures before moving to the next test.

1. Environment Test

```
cd C:\Scripts\KBFC_Webhooks
.\venv\Scripts\activate
python --version
python -m pip check
python -c "import arcgis, mysql.connector, requests, yaml; print('Environment OK')"
```

Expected result:

- Python reports version `3.10.x`.
- `pip check` reports no broken requirements.
- Import command prints `Environment OK`.

2. MySQL Connection Test

Run this from the repository root after `config.yml` is complete:

```
python -c "from utils.utils_general import load_config; import mysql.connector;
c=load_config()['mysql']; cn=mysql.connector.connect(host=c['host'],
user=c['user'], password=c['password'], database=c['database'],
port=c.get('port',3306)); print('MySQL OK'); cn.close()"
```

Expected result:

```
MySQL OK
```

3. Database Object Test

In MySQL Workbench:

```
USE kbfc;  
SHOW TABLES;  
SHOW FULL TABLES WHERE Table_type = 'VIEW';  
SELECT COUNT(*) FROM KBFC_MRR_Efforts_and_Records;
```

Expected result:

- Tables exist.
- Views exist.
- Count queries execute without SQL errors.

4. ArcGIS Import Test

Run the cleanup and import manually before relying on batch files:

```
cd C:\Scripts\KBFC_Webhooks  
.\venv\Scripts\activate  
python cleanup_orphans.py  
python arcgis_to_mysql_importer.py
```

Expected result:

- No unhandled Python traceback.
- MySQL target tables receive or update records.
- Generated log output appears under `logs/`.
- `logs\arcgis_cleanup.html` and/or `logs\arcgis_import.html` are created when conversion runs.

Check MySQL counts after import:

```
USE kbfc;  
SELECT COUNT(*) FROM KBFC_MRR_Efforts_and_Records;  
SELECT COUNT(*) FROM MRRCaptures;
```

5. KBFC API Sync Test

Use the test API first unless this is an approved production deployment.

```
cd C:\Scripts\KBFC_Webhooks
.\venv\Scripts\activate
python kbfc_import_tool.py
```

Expected result:

- No unhandled Python traceback.
- Records are posted, updated, skipped, or logged with validation errors.
- Generated log output appears under `logs/`.
- `logs\kbfc_import.html` is created when conversion runs.

Review the logs for:

- API 400 responses.
- Missing required fields.
- Invalid PIT tags.
- Missing site/project references.
- Skipped records that should have posted.

6. Batch File Test

Double-click or run:

```
cd C:\Scripts\KBFC_Webhooks
.\autorun_arcgis_to_mysql.bat
.\autorun_mysql_to_kbfc.bat
```

Expected result:

- The batch files activate the correct virtual environment.
- They run from the correct project folder.
- Output/log files update.

7. Run Without Shortcuts or Protocol Links

Use this path if desktop shortcuts, symbolic links, or `myapp://` protocol links have not been configured yet.

Activate the environment:

```
cd C:\Scripts\KBFC_Webhooks
.\venv\Scripts\activate
```

Run the ArcGIS to local database import:

```
python cleanup_orphans.py
python arcgis_to_mysql_importer.py
```

Run the local database to KBFC API sync:

```
python kbfc_import_tool.py
```

Open the log reader manually:

```
file:///C:/Scripts/KBFC_Webhooks/kbfc_log_reader.html
```

If scanner file loading is enabled, start it directly:

```
.\venv\Scripts\python.exe -m streamlit run kbfc_scanner\scanner_import.py --
server.headless false
```

Expected result:

- The tools run without requiring shortcuts or protocol registration.
- Logs are generated under `logs/`.
- The log reader opens from the local HTML file.
- The scanner file loader opens in the browser if Streamlit is started.

8. Log Reader Test

Open:

```
file:///C:/Scripts/KBFC_Webhooks/kbfc_log_reader.html
```

Expected result:

- The page opens in the browser.
- Recent import/API logs are visible or linked.

9. Optional Scanner Launcher Test

If the helper launcher was installed, start it:

```
.\dist\helper.exe
```

In another terminal:

```
Invoke-RestMethod http://127.0.0.1:8765/health
```

Expected result:

```
ok
--
True
```

Launch the Streamlit app through the helper or directly:

```
.\venv\Scripts\python.exe -m streamlit run kbfc_scanner\scanner_import.py --
server.headless true
```

Expected result:

- Streamlit starts on a local port.
- The scanner import screen loads in the browser.

10. Optional `myapp://` Protocol Test

After registering the `.reg` file, test a launcher URL from a browser or the relevant HTML page.

Expected result:

- Windows prompts to open the helper.
- The helper starts.
- The target Streamlit app or configured command launches.

Operational Checks After Installation

After a successful installation, confirm:

- `config.yml` points to the intended API environment.
- The active ArcGIS layers match the intended organization/survey version.
- MySQL tables and views exist in the `kbfc` schema.
- `autorun_arcgis_to_mysql.bat` runs successfully.
- `autorun_mysql_to_kbfc.bat` runs successfully.
- Logs are being written under `logs/`.
- The log reader opens.
- Scanner launcher works if required.
- Desktop shortcuts point to the correct install path.

Troubleshooting

Python Dependency Install Fails

Confirm the virtual environment is active:

```
.\venv\Scripts\activate
```

Then retry:

```
python -m pip install --upgrade pip
python -m pip install -r requirements.txt
```

If native package builds fail, install or repair Visual Studio Build Tools.

MySQL Connection Fails

Check:

- MySQL service is running.
- `config.yml` host, port, username, password, and database are correct.
- The `kbfc` schema exists.
- The user has privileges on the `kbfc` schema.

ArcGIS Import Fails

Check:

- ArcGIS credentials.
- Layer URLs.
- Network/VPN access.
- Survey layer/table indexes.
- Target MySQL table names.

API Sync Fails

Check:

- `api.base_url`.
- `api.api_key`.
- Whether the API environment is test or production.
- Validation errors in `logs/`.
- Required lookup/reference data for site, project, deployment, effort, and capture records.

Batch File Opens and Immediately Closes

Run it from PowerShell so the error remains visible:

```
.\autorun_mysql_to_kbfc.bat
```

Also confirm the `cd` path and virtual environment path inside the batch file are correct.

Maintenance Notes

- Pull repo updates before deployments:

```
git pull
```

- Reinstall dependencies after `requirements.txt` changes:

```
.\venv\Scripts\activate  
python -m pip install -r requirements.txt
```

- Rebuild `helper.exe` after changing `helper.py`:

```
pyinstaller --onefile --noconsole helper.py
```

- Re-run SQL migration scripts only after confirming they are safe for existing data.
- Back up MySQL before major schema changes or production sync changes.

Open Items for Final Guide

- Add exact production API base URL if production deployment is required.
- Add the approved ArcGIS layer list for each organization/survey version.
- Confirm whether `config.yml` should be created from a template file.
- Confirm exact SQL script order with a fresh database restore test.
- Review screenshots for public release and replace any image that exposes credentials, private URLs, or organization-specific paths.