

Phase 8: Data Management & Deployment

Objective:

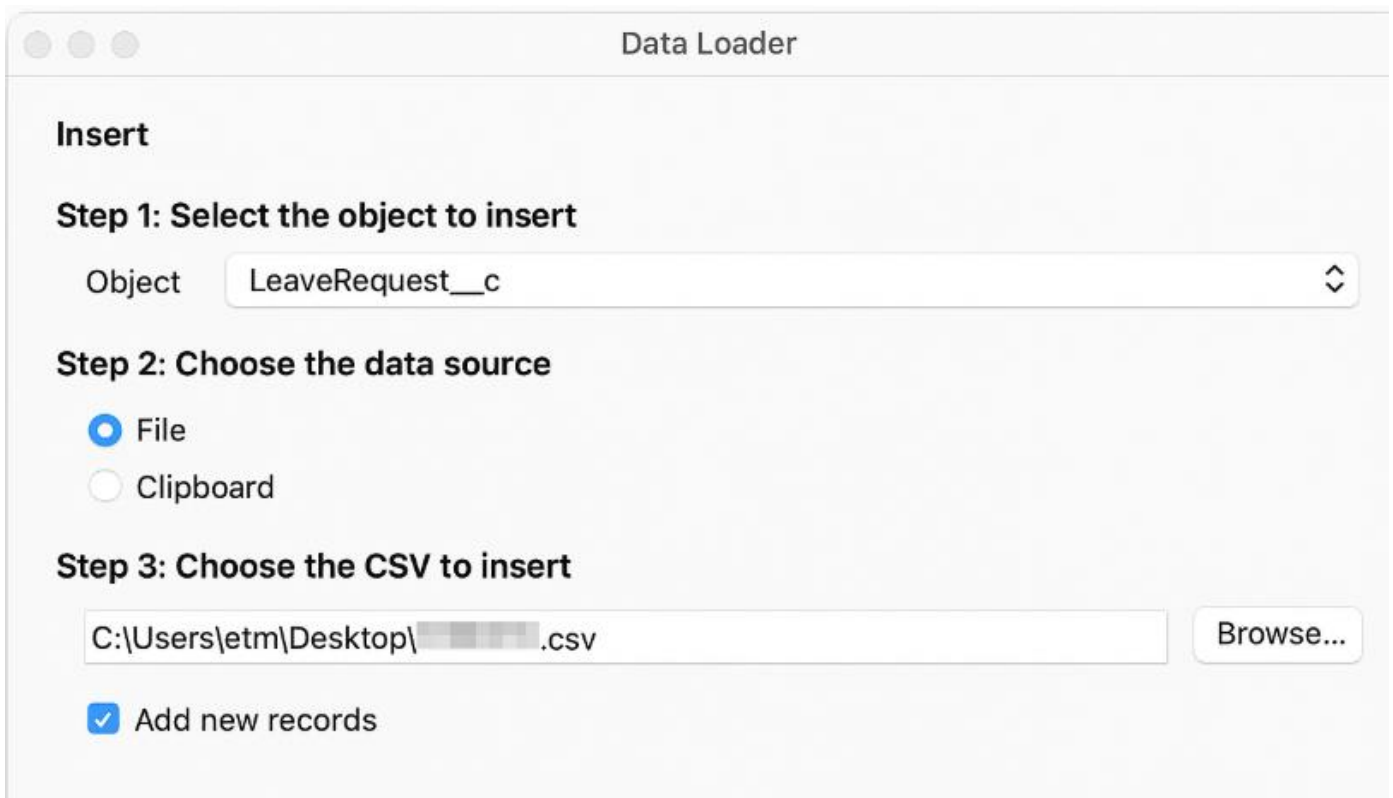
To ensure smooth and error-free deployment of Leave Tracker customizations from development to higher environments (UAT, Production) using best practices and tools.

Data Import Wizard

- Used to load initial employee data (Users, basic LeaveRequest__c records).
- Supports CSV files.
- Best for small/medium data volumes (< 50,000 records).

Data Loader

- Used to perform bulk operations (insert, update, delete, upsert) for large datasets.
- Example: Bulk upload of historical leave records.
- Supports command-line interface for automation.



The screenshot shows the 'Data Loader' application window. The title bar reads 'Data Loader'. The main content area is titled 'Insert'. Below this, there are three steps:

- Step 1: Select the object to insert**
Object: LeaveRequest__c (selected from a dropdown menu)
- Step 2: Choose the data source**
☒ File
☐ Clipboard
- Step 3: Choose the CSV to insert**
C:\Users\etm\Desktop\[redacted].csv (text input field)
Browse... (button)

At the bottom, there is a checkbox labeled 'Add new records' which is checked.

Change Sets

Used Change Sets for deploying metadata between environments (Sandbox → Production).

Included:

- Custom Objects (LeaveRequest__c)
- Apex Classes (LeaveRequestController, Triggers)
- Lightning Web Components (applyLeave, myLeaves, leaveRequest)
- Email Templates & Flows

VS Code Deployment Setup

Prerequisites

- Install Visual Studio Code.
- Install Salesforce CLI (SFDX).
- Install Salesforce Extension Pack in VS Code.
- Connect Org → `sfdx force:auth:web:login -d -a DevHub`.
- Sample Project Structure (VS Code)

```
leave-tracker-project/
├─ force-app/
│   └─ main/
│       └─ default/
│           ├── classes/      (Apex Classes)
│           ├── lwc/          (Lightning Web Components)
│           ├── objects/      (Custom Objects)
│           ├── triggers/     (Apex Triggers)
│           └─ email/         (Email Templates)
├─ sfdx-project.json
└─ .gitignore
```

- Retrieve metadata from source org
 - `sfdx force:source:retrieve -m ApexClass,CustomObject,LWC`
- Deploy metadata to target org
 - `sfdx force:source:deploy -p force-app/main/default`
- Run all tests before deployment
 - `sfdx force:apex:test:run --resultformat human --codecoverage`

Deployment Checklist

- Code Quality Check (PMD, Prettier for LWC).
- Apex Test Coverage $\geq 75\%$.
- Run Validation Deployment (test without committing).
- Backup Production metadata before final push.
- Post-deployment steps (activate Flows, verify Email Deliverability).

CI/CD Integration (Optional Advanced)

- GitHub Actions: Automate deployments on every push.
- Copado / Gearset: Enterprise-grade release management.