CoDIR AI Installation Manual

Windows Version

Official GitHub Repository: https://github.com/RobinCop63/CoDIR-IA

(Available from December 1, 2025)

Table of Contents

Chapter 1 – General Introduction

1.1 Purpose of This Manual

This manual provides step-by-step guidance for the complete installation of CoDIR AI on a Windows 11 environment.

It is intended for anyone who wishes to deploy the project locally, even without prior experience in Python development.

The goal is to achieve a functional, stable, and secure environment capable of running the CoDIR AI Streamlit dashboard.

1.2 Target Audience

This guide is designed for:

- Executives, CFOs, transformation managers, or consultants who want to test CoDIR AI on their workstation
- Users with basic computer knowledge
- Non-technical profiles who want to install a local AI in a simple and reproducible way

1.3 Recommended Working Environment

Component	Minimum Requirement	Optimal Requirement
Operating System	Windows 10 (64-bit)	Windows 11 Pro 64-bit
Processor	Intel i5 or AMD equivalent	Intel i7 / Ryzen 7
RAM	8 GB	16 GB or more
Free Storage	10 GB minimum	50 GB on SSD
Graphics Card	Optional	NVIDIA RTX with 8 GB VRAM
Internet Connection	Yes, for installation	Recommended for updates
Software	PowerShell 7+, VS Code, Python 3.13	Same

1.4 About the CoDIR AI Project

CoDIR AI (short for Comité de Direction IA / AI Executive Committee) is an open-source multi-AI orchestrator designed to support decision-makers in their daily decision-making processes.

It allows you to connect multiple artificial intelligence models (OpenAI, Google Gemini, Anthropic, Mistral, etc.) in a single, simple, and powerful interface.

1.5 Typical Folder Structure

Once installed, the CoDIR AI folder will look like this:

```
CoDIR-IA/
├── app.py
├── launch.bat
├── requirements.txt
├── .env
├── venv/
│   ├── Scripts/
│   └── Lib/
```

```
├── providers/
│   ├── openai_provider.py
│   ├── gemini_provider.py
│   └── mistral_provider.py
├── outputs/
└── prompts/
```

Chapter 2 – System Prerequisites

2.1 Checking Your Windows Version

Before any installation, verify that your system is up to date.

Open the Start menu → Settings → System → About.

Verify that you are using Windows 10 or 11 – 64-bit.

2.2 Updating the System

1. Open Settings → Windows Update
2. Click Check for updates
3. Install all available updates
4. Restart your computer

2.3 Installing Python 3.13

1. Go to https://www.python.org/downloads/windows/
2. Download Python 3.13 (Windows Installer 64-bit)
3. **Check "Add Python to PATH" before installing**
4. Click Install Now
5. Open PowerShell and verify:

```
python --version
```

2.4 Installing Git

1. Go to https://git-scm.com/downloads
2. Download the Windows 64-bit version
3. Run the default installation
4. Verify in PowerShell:

```
git --version
```

2.5 Checking PowerShell 7+

1. Open PowerShell and type:

```
$PSVersionTable.PSVersion
```

2. If the version is lower than 7.0, download the latest version from GitHub.

2.6 Installing Visual Studio Code

1. Download from https://code.visualstudio.com/
2. Install with default options
3. Install the extensions: Python, Streamlit, .ENV files support, GitLens

2.7 Final Verification

Open PowerShell and test:

```
python --version
git --version
code --version
```

Chapter 3 – Installing CoDIR AI

3.1 Creating the Working Folder

1. Open File Explorer
2. Create a folder at the root of your drive or in OneDrive:

```
C:\Users\<your_name>\OneDrive\CoDIR-IA\
```

3. Copy all files from the public CoDIR AI folder (e.g., downloaded from GitHub) into this folder.

It must contain at minimum: app.py, launch.bat, requirements.txt, .env, providers/, outputs/, prompts/

💡 *Avoid paths that are too long or contain spaces (this can sometimes cause Python errors).*

3.2 Opening the PowerShell Terminal

1. Click in the folder's address bar and type:

```
powershell
```

2. A PowerShell window opens directly in your CoDIR AI folder. You should see:

```
PS C:\Users\<your_name>\...\CoDIR-IA>
```

3.3 Creating the Virtual Environment

The virtual environment isolates your Python installation from the rest of the system.

In PowerShell, type:

```
python -m venv venv
```

Once completed, you will see a venv folder appear in your CoDIR-IA folder.

3.4 Activating the Environment

Still in PowerShell:

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

You'll know everything is working if your command prompt becomes:

```
(venv) PS C:\Users\<your_name>\...\CoDIR-IA>
```

⚠️ *If you get an error "Activate.ps1 cannot be loaded", first run this command:*

```
Set-ExecutionPolicy Unrestricted -Scope Process
```

3.5 Installing Dependencies

Now install all required Python modules:

```
pip install -r requirements.txt
```

✅ Wait for the installation to complete before continuing.

If you see warnings like "already satisfied" or "successfully installed", that's a good sign.

Chapter 4 – Initial Configuration

4.1 The .env File

The .env file contains all the project's confidential variables (API keys, ports, preferences).

You can open it with Visual Studio Code or a simple Notepad.

Example:

```
OPENAI_API_KEY=sk-xxxxxxxxxxxxxxxxxxxxxxxxxxxx
ANTHROPIC_API_KEY=sk-ant-xxxxxxxxxxxxxxxxxxxx
GEMINI_API_KEY=AIzaSy...
MISTRAL_API_KEY=xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
PORT=8501
```

⚠️ *Never share this file on the Internet or GitHub.*

4.2 Verifying Internal Paths

Some files (like launch.bat) contain a line that points to your local folder.

Open launch.bat and verify the following line:

```
cd "C:\Users\<your_name>\OneDrive\CoDIR-IA"
```

💡 *The path in quotes must exactly match your local CoDIR AI folder.*

4.3 Testing .env File Reading

You can test if the keys are being read correctly with this command:

```
python -c "import os; print(os.getenv('OPENAI_API_KEY'))"
```

✅ If the key is displayed (or at least the beginning), your .env file is being detected correctly.

4.4 Configuring Streamlit

The CoDIR AI dashboard is powered by Streamlit.

You can verify its proper installation:

```
streamlit --version
```

If the command responds with a version number, everything is ready.

Otherwise:

```
pip install streamlit
```

4.5 Local Network Configuration

The default port used is 8501.

If you want to change it (for example, in case of a network conflict):

- Open the .env file
- Modify the line:

```
PORT=8502
```

Streamlit will then launch the application on the new port.

Chapter 5 – Launching and Testing

5.1 Verifying the Environment

Make sure your virtual environment is active:

```
(venv) PS C:\Users\<your_name>\...\CoDIR-IA>
```

If not, run:

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

5.2 Launching via the .bat File

Double-click on launch.bat

A window opens with the message:

```
========================================
🚀 Launching CODIR AI
========================================
```

Then a few seconds later, Streamlit starts your dashboard.

5.3 Accessing the Interface

Your browser will automatically open to:

```
http://localhost:8501
```

💡 *If it doesn't open automatically, open it yourself and paste the address.*

5.4 Functionality Tests

Once the application is displayed:

- Verify that the home page opens without errors
- Test each AI model (OpenAI, Gemini, etc.) to validate the connections
- If you get a "missing API key" error, go back to your .env file

5.5 Proper Server Shutdown

When you want to close CoDIR AI:

- Return to the PowerShell window
- Press CTRL + C to stop Streamlit
- Then close the window

5.6 Cleanup and Backup

Before leaving definitively:

- Close your virtual environment:

```
deactivate
```

- Back up your CoDIR AI folder to OneDrive
- Verify that the .env file is excluded from any public sharing

Chapter 6 – Troubleshooting Common Issues

6.1 The launch.bat File Won't Start

Symptom: nothing happens or the window closes immediately.

Possible causes:

- The CoDIR AI folder path in the .bat is incorrect
- The venv virtual environment hasn't been created

Solution:

- Open your launch.bat file with Notepad
- Verify the line:

```
cd "C:\Users\<your_name>\OneDrive\CoDIR-IA"
```

- If the path is correct, verify that the venv folder exists
- Otherwise, recreate it:

```
python -m venv venv
```

6.2 Error "Activate.ps1 Cannot Be Loaded"

Cause: PowerShell execution policy blocks local scripts.

Solution:

```
Set-ExecutionPolicy Unrestricted -Scope Process
```

Then rerun:

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

6.3 Streamlit Won't Start

Symptom:

```
ModuleNotFoundError: No module named 'streamlit'
```

Solution:

```
pip install streamlit
```

If the problem persists:

```
pip install --upgrade streamlit
```

6.4 API Keys Not Recognized

Symptom:

```
KeyError: 'OPENAI_API_KEY'
```

Or error message in Streamlit.

Solution:

- Open your .env file
- Verify the syntax:

```
OPENAI_API_KEY=sk-xxxxxxxxxxxxx
```

Restart CoDIR AI after modification:

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

```
streamlit run app.py
```

6.5 Python Is Not Recognized

Symptom:

```
python is not recognized as an internal command
```

Solution:

- Reinstall Python by checking "Add Python to PATH"
- Or manually add this path to environment variables:

```
C:\Users\<name>\AppData\Local\Programs\Python\Python313\
```

6.6 Streamlit Port Conflict

Symptom:

```
OSError: [Errno 98] Address already in use
```

Solution:

Open .env and change the line:

```
PORT=8502
```

Then relaunch:

```
streamlit run app.py
```

6.7 Frozen or Slow Interface

Possible causes:

- Too many tabs open in the browser
- Streamlit cache is saturated

Solution:

Close other tabs and clear the cache:

```
streamlit cache clear
```

Chapter 7 – Updates and Maintenance

7.1 Updating Python Dependencies

Open PowerShell in your CoDIR AI folder.

Activate the environment:

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

Update all modules:

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

💡 *Do this once a month to stay up to date.*

7.2 Updating Project Code (GitHub)

If you cloned the project via GitHub:

```
git pull origin main
```

This updates your local folder with the latest public version of CoDIR AI.

7.3 Regular Backups

Create a copy of the CoDIR-IA folder on your OneDrive or Synology NAS.

Essential items to back up:

- The /outputs folder
- Your custom prompts
- The .env file

7.4 Environment Cleanup

If you want to start fresh:

```
deactivate
rmdir /s /q venv
python -m venv venv
pip install -r requirements.txt
```

7.5 Version Control

To check the current version:

```
python --version
streamlit --version
git --version
```

Note these somewhere, useful for support exchanges.

7.6 Complete Reinstallation

If you want to reset everything:

- Back up your .env
- Delete the entire CoDIR-IA folder
- Download the latest version from GitHub

- Follow chapters 3 to 5 of this manual again

Chapter 8 – Best Practices and Security

8.1 Protecting API Keys

- Never share your .env file
- If you send your project, create a public version without keys
- Use personal API accounts, never shared ones

8.2 Privacy and Local Data

- CoDIR AI does not collect or send any data without your action
- Files are processed locally on your computer

💡 *Logs generated by Streamlit can be deleted at any time.*

8.3 Organizational Best Practices

- Group your work files in /outputs
- Don't add large files to /providers
- Keep a clear and identical structure between machines (useful for support)

8.4 Windows Security

Activate and keep up to date:

- Windows Defender or Bitdefender
- Automatic Windows updates
- OneDrive backups
- A strong password on your Windows account

8.5 Responsible Use

Open source implies transparency and respect:

- Cite sources when you reuse the project
- Participate in the community (feedback, improvements)
- Never distribute sensitive content generated by AI without human verification

8.6 Next Steps

Once you've mastered this Windows version, you can:

- Test the macOS version (specific chapters coming soon)
- Contribute to the project's evolution on GitHub
- Join the CoDIR AI community (via LinkedIn or the future website)

Appendices A to D

Appendix A – Technical Glossary

Term	Simplified Definition
API Key	Unique access key provided by a service (OpenAI, Google, etc.) to authenticate requests.
.env	File containing environment variables (keys, ports, etc.), never shared publicly.
Virtual Environment (venv)	Isolated Python space where project modules are installed.
Streamlit	Python framework for creating interactive web interfaces.
Git / GitHub	Version control tools for tracking changes and collaborating on a project.
Prompt	Input text sent to an AI to generate a response.
Token	Unit of measurement for text read or written by an AI model (~4 English characters).
PATH	System variable telling Windows where to find executable programs.
Requirements.txt	List of Python libraries to install for the project.
PowerShell	Modern and powerful Windows terminal.
Multi-AI Orchestrator	Software connecting multiple AIs in a single interface.

Appendix B – Complete List of Python Libraries

Library	Main Function
streamlit	User interface and dashboard
openai	Connection to GPT models (ChatGPT, GPT-4, GPT-5)
anthropic	Connection to Claude models
google-generativeai	Connection to Gemini (Google)
mistralai	Connection to Mistral and Mixtral models
python-dotenv	Reading variables from the .env file
requests	HTTP requests to external APIs
pandas	Manipulation and display of tabular data
tiktoken	Calculating the number of tokens per message
langchain (optional)	Managing logical chains of AI agents
pyyaml	Reading and writing YAML files

Appendix C – CoDIR AI Version History

Version	Main Changes
v14.0 (Aug 2025)	First stable version, full multi-AI support
v14.2	Streamlit error fixes + offline test mode
v14.5	Simplified .env file integration and launch optimization
v14.8	Gemini 2.0 compatibility
v15.0 (Oct 2025)	Public open source version (Windows)
v15.1 (planned)	macOS version + NVIDIA/Apple Silicon GPU optimization

Appendix D – Credits and Acknowledgments

Development and Co-Authors

- **Robin Sauzet** – Creator of the CoDIR AI project, design, testing, and documentation
- **ChatGPT (OpenAI GPT-5)** – AI co-author and technical documentation generator

Contributions and Frameworks

- **OpenAI** – for GPT-4 and GPT-5 models
- **Anthropic** – for the Claude series
- **Google** – for Gemini (via google-generativeai)
- **Mistral AI** – for Mixtral
- **Hugging Face** – for the open source ecosystem
- **Streamlit** – for the dashboard framework
- **Python Software Foundation** – for the project base

Thank you to everyone who participates in making artificial intelligence more accessible, transparent, and ethical. CoDIR AI is first and foremost a collective project.

CoDIR AI – Windows Installation Manual

Final Version – November 2025

Authors: Robin Sauzet & ChatGPT (SASU Hi! Gestion)

Logo: Camille Belloir
https://www.linkedin.com/in/camille-belloir/