IOblend

DEVELOPER EDITION / INSTALLATION

Install IOblend on Windows

From download to the first working local pipeline.

A practical installation guide

Set up Designer and the local Engine, configure your development environment, and build your first validated pipeline.



NOVEMBER 2026 / DEVELOPER EDITION

01 / PREPARE

Start with Developer Edition

Designer and Engine run locally on your workstation. Development is free to use; production deployment uses Enterprise Edition.

The first milestone

Complete the installation, open Designer and the Script window, then run one local Source → Transform → Sink pipeline. You can connect to remote sources later when your workstation has the relevant access.

RECOMMENDED WORKSTATION

Resource	Recommended / minimum
Processor	Intel or AMD; 3 GHz recommended (2 GHz minimum)
Memory	16 GB or more recommended (4 GB minimum listed)
Storage	30 GB or more on a local SSD (10 GB minimum listed)
Display	1024 × 768 or higher; DirectX 9/10-compatible graphics
Connectivity	Internet access for download and authorised remote data connections
Archive tool	7-Zip - use it to extract the full IOblend distribution

The packaged local Engine uses your workstation CPU, memory and storage. Keep project files on a local drive for the first installation, and use test data rather than production credentials.

Before downloading

Confirm you have permission to install the package and configure your workstation. The first setup updates environment variables and may be subject to your organisation’s endpoint policies.

02 / DOWNLOAD

Download and extract the package

The most common avoidable installation issue is an incomplete archive extraction.

01

Open the Developer Edition download page

Complete the registration form at ioblend.com/download/ and choose the Windows version shown in the confirmation.

02

Test and extract with 7-Zip

Obtain 7-Zip if needed. Use 7-Zip > Test archive, then 7-Zip > Extract. Avoid the standard Windows compressed-folder extractor for this distribution.

03

Use a short local installation path

Move the complete extracted folder to your local C: drive. Avoid executing directly inside the ZIP, from a network share or from a synchronised folder.

EXAMPLE INSTALLATION LOCATION

```
C:\IOblend\ioblend_designer\
  ioblend_designer.exe
```

Check before launching

The extracted folder should contain the application and its supporting components. Do not move only the executable out of the distribution folder.

03 / FIRST LAUNCH

Start Designer and initialise Windows

Keep the Designer and its Script/terminal window open during initialisation.

01

Launch the application

Open the extracted folder and run ioblend_designer. Create a shortcut if useful. The Script window and Playbook Designer should both appear.

02

Open the environment setup

In Designer select Setup > Setup Environmental Vars. When the dialog warns that environment variables will change, make a local backup before confirming.

WINDOWS COMMAND PROMPT - BACKUP

```
set > "%USERPROFILE%\Documents\ioblend_env_before.txt"
```

The environment listing may contain sensitive values. Keep the backup private and delete it according to your workstation policy once no longer needed.

03

Complete the setup and restart

Return to Designer, confirm the setup, and check the Script window for six success messages. Close and reopen Designer if required.

Ready to continue

Both windows should reopen without missing-file or component errors. If not, use the troubleshooting section before changing any playbooks.

04 / PROJECT SETUP

Create a clean development project

Separate playbooks, input data, outputs and runtime state from the start.

Create a project directory before adding pipeline components. The example is local to Windows; the Engine resolves these paths through the Run Parameters configuration.

SUGGESTED PROJECT STRUCTURE

```
C:\IOblendLabs\first-pipeline\
  config\          Run Parameters and settings
  playbooks\       JSON component definitions
  transforms\      External Python scripts
  source_data\     Local test files
  output\          Local sink results
  checkpoint\      Streaming checkpoints
  state\           Execution state and logs
  interaction\     Designer / Engine files
  jars\            Additional JDBC drivers
```

> **Keep the paths separate**
>
> Do not point a streaming source at the folder used by its destination. Generated output could otherwise be read again as new input. Use separate state and checkpoint locations for independent tests.

The project is a useful boundary for portable JSON definitions, non-secret configuration and repeatable test runs. Keep credentials and generated state outside shared source-control repositories.

05 / RUN PARAMETERS

Configure the project before building

Run Parameters tell Engine where the project lives and how development execution should behave.

In Designer select **Config > Project Run Params > New Run Params**. Enter the relevant values, save the configuration and reopen it to confirm the paths.

Setting	Typical first local value
Pipeline Location	.../first-pipeline/playbooks
PipeLine Name	first_pipeline_01
Transform Ext	.../first-pipeline/transforms
State Directory	.../first-pipeline/state
Run Parameters Path	.../first-pipeline/config
File System	local
Spark Master / SSH	local / local
Epochs	1 (one run; 0 for continuous operation)
Debug	true
Run Interactive	true
interaction_by_file	true

For the first lab

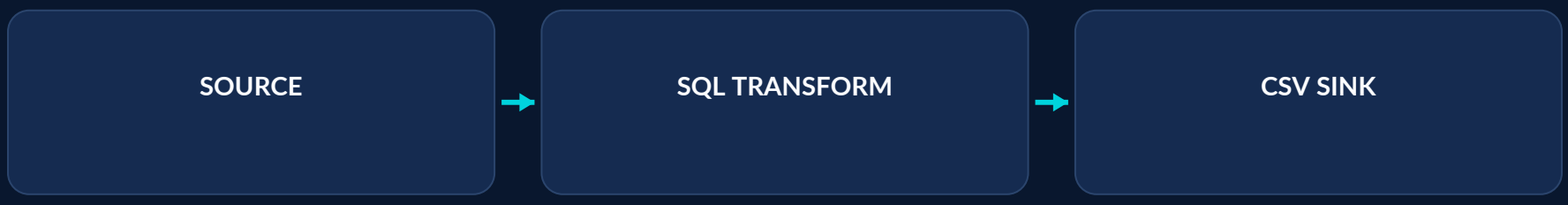
Use **Epochs = 1** and all three development-time debugging flags set to **true**. Production Enterprise jobs typically disable interactive development, but that belongs to the production deployment guide.

06 / FIRST PIPELINE

Build. Test. Add the next component.

Work through the pipeline one component at a time, verifying each result before expanding the design.

Start with a small CSV and a simple calculation. The goal is a correct end-to-end result, not a large demonstration pipeline. The Developer Handbook contains the full step-by-step exercise.



01

Add Source

Read a small local CSV. Confirm expected columns, row count and inferred schema in the interactive output.

02

Add SQL Transform

Create a simple aggregation or filter. Inspect the intermediate DataFrame and check the arithmetic.

03

Add Sink

Write to a different output folder. Open the actual CSV and verify record counts and calculated values.

Repeat the same loop

Once each component is verified, add the next source, transformation or sink. A single pipeline can contain several of each, with dependencies and run order controlling the flow.

07 / WORKSTATION ACCESS

Script execution and endpoint protection

Apply the controls approved for the developer machine; do not weaken system-wide protection simply to run the package.

PowerShell execution policy

If scripts are blocked, inspect the active policy first. Use the script-execution scope approved by the workstation administrator. The older installation instructions included a system-wide Bypass command; that should not be treated as a default setup requirement.

POWERSHELL - INSPECT CONFIGURED SCOPES

```
Get-ExecutionPolicy -List
```

If trusted IOblend, Java, Python or Spark files are blocked by endpoint software, request targeted approval for the verified files or processes. Avoid disabling antivirus or firewall protection globally.

Symptom	First check
Designer does not open	Run from the complete extracted local folder; inspect Script messages.
Files or modules are missing	Re-test and re-extract the archive with 7-Zip.
Environment setup incomplete	Review the six success messages, path entries and restart.
Unexpected access denied	Check project-folder permissions and endpoint security events.

Useful support information

Capture the exact first error, installed OS, IOblend Engine version and relevant logs. Remove credentials, tokens and sensitive paths before sharing.

08 / TROUBLESHOOTING

When Spark does not start

Diagnose the local environment first; do not change working pipeline SQL to solve a startup failure.

What you observe	Where to look first
No Java process appears	Incomplete extraction, Java environment or application permissions.
Java starts, then exits	Runtime compatibility, available RAM, driver/heap settings.
Java keeps running; Designer waits	Interaction folder writes, endpoint protection or local networking.
Spark starts; a component later fails	Source/transform/sink configuration and component logs.

Use the simple CSV lab to isolate installation issues. Check that the playbooks, configuration, state and interaction directories exist and that your user can write to the interaction folder.

COMMAND PROMPT - JAVA CHECKS

```
java -version
where.exe java
echo %JAVA_HOME%
```

SPARK LOCAL PROFILE - 16 GB+ WORKSTATION

```
--conf spark.driver.memory=8g
--conf spark.driver.extraJavaOptions=-Xms4g
```

Memory settings must match

Do not combine an 8g driver-memory limit with -Xms20g. If Spark initialisation still times out, the Developer Handbook includes interaction-file, endpoint-protection and loopback diagnostics.

09 / READY TO BUILD

Installation complete. Build something.

The next step is a small, working pipeline - built one component at a time.

The best first experiment is deliberately simple: connect a local CSV source, add a SQL transformation and write the result to a separate destination. Inspect each stage before continuing.

Your first success criteria

Designer and Script/terminal windows open correctly; project Run Parameters are saved; a Source returns the expected rows; a SQL transformation produces verified results; and the Sink writes the expected output.

CONTINUE THE LEARNING PATH

01 / Developer Handbook

Use the guided CSV lab, Run Parameters reference and troubleshooting sections.

02 / Official video tutorials

Move from the first source through transformations, sinks and streaming.

03 / Enterprise Edition

When your pipeline is ready for production, discuss remote deployment and operations.

DOWNLOAD DEVELOPER EDITION

DOCUMENTATION & VIDEOS

ioblend.com/download/ | ioblend.com/documentation/
Enterprise deployment? ioblend.com/contact/