

GETTING STARTED

The AI Agent Workshop

From the Repository to the Work, Step by Step

David Bories

*Tried by the Flow — A Processual Ontology of the Datum, implementation
Getting Started — The AI Agent Workshop — From the Repository to the Work,
Step by Step*

Conceived and directed by David Bories — Albi, France.

Written using generative artificial intelligence.

Translation produced by generative artificial intelligence, under the supervision of the author.

July 2026.

Version 0.3.0-en, pocket format.

© David Bories, 2026. Released under the Creative Commons Attribution 4.0 International licence (CC BY 4.0). Any use, including commercial use, is permitted provided the author and the source are credited.
<https://creativecommons.org/licenses/by/4.0/deed.en>

The full text of this volume and of the whole corpus is freely available at <https://artisanat-de-la-donnee.fr>

contact@artisanat-de-la-donnee.fr

Table of Contents

Who This Document Is For	3
0. The Rule of Arbitration Between Sources (for the Agent)	5
1. Prerequisites	7
2. The Folder Structure	9
3. What the User Provides	11
4. Launching the Workshop, Agent by Agent	13
5. Validating at Each Milestone	15
6. Retrieving and Installing	17
7. Making It Evolve	19
If You Need Help	21
Translation choices	23

Note on the translation — This translation re-fixes the source text without displacing its substance. A few terms of the corpus have no obvious equivalent in English; the choices made, and how they are to be read, are set out in the section “Translation choices” at the end of the volume. This translation is not frozen: it will evolve with readers’ feedback and with the improvement of the means of translation. The French version is authoritative.

Who This Document Is For

For the **data artisan** (a craft yet to be created) who accompanies an entrepreneur, or for the entrepreneur themselves if they have the competences. It describes how to launch the **AI agent workshop** — in order to produce, from a statement of activity, the **activity manual** and the **software work** that is its active version.

This document is deliberately **dated and concrete**: it names tools, commands, a folder structure. It lives in the repository and is revised with it. The doctrine is in the manual *Technical Data*, chapter 12.

0. The Rule of Arbitration Between Sources (for the Agent)

Before any generation, the agent holds the corpus as its jurisdiction, according to five rules of precedence. **One.** For **substance** — what is true of the datum —, the trilogy is authoritative; in case of apparent tension, the tome prevails over any continuity volume and over the synthesis. **Two.** For **realization** — how to do it —, the continuity volumes are authoritative (*Technical Data*, the SGQA); their most recent version arbitrates. **Three.** The **synthesis** summarizes and does not replace: it cannot be cited against a tome. **Four.** The **canonical lexicon** settles definitions; a term absent from the lexicon is flagged as outside the corpus, never improvised. **Five.** The **dated examples** (figures, tools, sectors) commit nothing: they illustrate form, never substance. Every answer of the agent must be able to name its source and the rule that makes it prevail.

1. Prerequisites

- **The digital workshop** (*Technical Data*, chapter 10): a machine capable of running a local AI model.
 - **A coding agent** installed: Vibe Code (Vibe CLI) — supervised access to the file system, the shell and tools; it reads, runs and writes code.
 - **git** installed, to version the work (backup and memory of its evolution).
 - **The base**: the Artisanat-de-la-donnee repository (URL to be confirmed: <https://github.com/Dawebxx/Artisanat-de-la-donnee>), cloned or downloaded as a zip and copied onto the local machine.
-

2. The Folder Structure

Once the base has been retrieved, the working folder looks like this:

```
artisanat-de-la-donnee/
├── README.md
├── commencer.md
├── corpus/
│   ├── tome_1_le_fleuve_et_le_canal.md
│   ├── tome_2_de_la_source_au_torrent.md
│   ├── tome_3_la_trace_ou_la_donnee.md
│   ├── artisan_de_la_donnee.md
│   ├── systeme_gestion_qualite_activite.md
│   ├── donnees_techniques.md
│   └── ... (all the volumes)
├── atelier/
│   ├── plan-developpement-atelier-agents.md
│   ├── chartes-agents.md
│   ├── secteurs/
│   │   └── secteur-bijouterie.md
│   └── gabarits/
├── mon-activite/
│   ├── enonce.md
│   ├── manuel/
│   └── oeuvre/
```

← this document

← the knowledge base (substance)

← the prompts of the AI Agents

← sector documents (starter)

← empty artefact templates

← the entrepreneur's workspace

← TO BE FILLED IN

← produced by the substance layer

← produced by the form layer (PWA)

The corpus/ folder and the atelier/ folder are the knowledge base and the instructions; mon-activite/ is the only folder that one fills in and that fills itself.

3. What the User Provides

The user does **not** write prompts: the prompts are the **charters** (atelier/chartes-agents.md). They provide only two things:

1. **The statement of their activity** — in mon-activite/enonce.md: what they do, what they sell, to whom, with what. In free language, without worrying about form.
2. **The choice of a sector** — a document from atelier/secteurs/ that *starts* the reflection (for example secteur-bijouterie.md). It gives the axis; the statement gives the singular reality.

Everything else — the mapping, the manual, the architecture, the code — is produced by the AI Agents.

4. Launching the Workshop, Agent by Agent

Open Vibe Code at the root of the folder, and give it corpus/ and atelier/ as its knowledge base. Then run through the equipment itinerary (*Technical Data*, chapter 11) **in order**. For each agent, the instruction is the same: “Play the role defined by your charter; read the input artefact; produce the output artefact.”

Order	AI Agent	It is given	It writes in
1	Discovery	enonce.md + sector	manuel/brief-activite.
2	Activity	brief-activite	manuel/modele-activite
3	Modeller	+ corpus	
4	Traceability	modele-activite	manuel/tracabilite.md
—	Flows & Data (the manual is assembled: the four files above)	modele-activite	manuel/flux-et-donnees
5	Hexagonal Architect	the manual	oeuvre/dossier-hexagonal
6	Business Logic Engineer	dossier-hexagonal	oeuvre/ (core + tests)
7	Drawing / Fixation Engineer	model + dossier	oeuvre/ (input, storage, emails)
8	Monitoring Engineer	flux-et-donnees	oeuvre/ (dashboard)
9	Circulation Engineer (if needed)	dossier-hexagonal	oeuvre/ (exports, email port, API)
10	Validator	the work + the manual	oeuvre/rapport-validation living documentation

The **Corpus Keeper** rereads each artefact before moving on to the next (vocabulary, invariants, the invoicing safeguard). The **Orchestrator** — here, that is you — keeps the order and triggers the milestones.

5. Validating at Each Milestone

This is where real validation takes place, when the work is put into practice. Before moving from one agent to the next, the **referent** (the artisan or the entrepreneur) reads the artefact produced and checks it against the reality of the activity. Under Vibe Code, nothing is written without supervision: one rereads, one corrects, one relaunches the agent if needed. The gesture holds: *deliver a work over which the client can take back control.*

6. Retrieving and Installing

At the end of the itinerary:

- mon-activite/manuel/ contains the **activity manual** (substance, lasting).
- mon-activite/oeuvre/ contains the **work**: a native, local-first core, **installable as a PWA** on the entrepreneur's computer and smartphone.

A git repository is initialized on mon-activite/: backup and memory of its evolution. The PWA is installed. The activity runs.

7. Making It Evolve

The software is not frozen. For an adjustment or a new feature, one re-launches **only the agent concerned** with the artefact to be corrected, one validates, one versions. This is the cycle of the itinerary: **immersion** → **adjustment** → **expansion**.

Three scripts at the root of the repository equip this cycle, all connected to the same generation port (`lib-agent.sh`, which dispatches to Vibe Code or Claude Code):

- **init.sh** — the complete itinerary: from the statement to the manual, then to the work, up to the first delivery.
- **evolve.sh** — an evolution (design, feature, fix): on a **beta branch**, validated at the test gate, switched to production and **delivered** as a version.
- **debug.sh** — fixing a defect **on the current branch**, without a beta branch or a release. It serves the case that the tests of the core do not see: **interface wiring** (a form that triggers nothing). Its debugging agent *reproduces* the defect and inspects `adaptateurs/interface/app.js` (event listeners, selectors) in addition to the core; it does not conclude “everything is green” on the sole strength of the tests. No version is delivered; a record in `progress.md` is written only if one asks for it.

At the **test gate** common to these scripts, when `node --test` fails, there are three outcomes: **d** lets the agent correct and loops back; **c** continues the itinerary when one has corrected things oneself (“corrected manually”); **n** stops.

If You Need Help

The entrepreneur can follow this itinerary alone if they have the competences, be accompanied by the **data artisan**, or delegate the whole of it to them. The artisan knows how to manage the AI Agents: it is their craft — the one the corpus calls for creating.

Translation choices

These notes say how to read the English words that render the terms of the corpus; they are fixed for the whole English translation by the *Translation Charter of the Corpus*.

- **The work** renders *l'œuvre*: the software that the workshop produces and that the client can take back in hand; the **activity manual** (*manuel d'activité*) is its substance, the work its active version. The folder names of the repository (*oeuvre/*, *manuel/*, *mon-activite/*, *enonce.md...*) keep their French spelling: they are the names on disk.
- **The data artisan** (*l'artisan de la donnée*) is the referent whose domain is the datum itself; the corpus calls for creating this craft. The **referent** is whoever reads, for the activity, what the user cannot read alone.
- **The AI agent workshop** (*l'atelier d'agents IA*); the **digital workshop** (*l'atelier numérique*) is the machine that runs it. The names of the agents are given in English (*Discovery*, *Activity Modeller*, *Corpus Keeper*, *Orchestrator...*).
- **Substance / form** (*fond / forme*): what is true of the datum independently of techniques, against what depends on them; the **substance layer** produces the manual, the **form layer** produces the work.
- **Drawing** and **fixation** (*prélèvement*, *fixation*): the gesture that takes a value from the flow, and the operation that gives it a medium.
- **Tome** names each of the three tomes of the trilogy; **continuity volume**, the volumes that extend it (*Technical Data*, the SGQA). This volume is an **implementation** volume (*mise en œuvre*); **realization** renders *réalisation*, how to do it.
- **Itinerary** renders *parcours* (the equipment itinerary); *path* is kept for the three paths of Tome 3.
- **Immersion** → **adjustment** → **expansion** (*imprégnation* → *ajustement* → *expansion*): the cycle by which the client first takes in the base, then adjusts it to their real needs, then extends it.
- **Titles**. *Getting Started* — *The AI Agent Workshop* renders *Commencer* — *L'atelier d'agents IA*; *Technical Data* renders *Données techniques*.

Getting Started — The AI Agent Workshop

—

From the Repository to the Work, Step by Step

—