

ExO Upgrade Agent — User Guide

Install the skills and run an upgrade with your AI coding agent

Welcome, and thanks for helping us test the ExO Upgrade Agent. This guide walks you through installing the skills, running an upgrade end to end, and knowing what to expect at each step so nothing is surprising. It should take about 10 minutes to get to your first generated plan.

What this is: a set of AI *skills* for Claude Code or Codex that analyze a legacy Contentful space and help you migrate it to Experience Orchestration (ExO). You talk to your agent in plain language; the skills do the mechanical work.

What this is not: there is no app to launch and no program of ours you run directly. The skills are instructions plus bundled code that *your* agent runs. It is non-destructive — it reads your space and creates new ExO drafts for review; it never changes or deletes your existing content.

1. Prerequisites

- **Claude Code or Codex**, installed and working.
- **Node.js 24 or newer** — check with `node --version` . (nodejs.org)
- **Python 3** — check with `python3 --version` .
- **macOS or Linux**. Windows is not supported in this beta.
- Your Contentful **CMA token**, **Space ID**, and **environment** (e.g. `master`) for the space you

want to upgrade. The agent asks for these; they are used only to talk to Contentful's API and are saved locally, never sent anywhere else.

2. Install the skills

You received a zip named `exo-upgrade-skills-<date>.zip` . Unzip it — you get one folder, `exo-upgrade-skills/` , containing the skill set. Install it one of two ways.

Option A — with the skills CLI (recommended)

From the folder that contains `exo-upgrade-skills/` :

```
npx skills add ./exo-upgrade-skills --copy -g --skill '*' --agent claude-code codex -y
```

- `npx` comes with Node — nothing extra to install; it fetches the small `skills` helper on demand.
- `--copy` copies the files in, so you can delete the unzipped folder afterward.
- `-g` installs globally, so the skills work from any folder.
- Drop `codex` (or `claude-code`) from `--agent` if you only use one.

Option B — copy the folders by hand (no CLI)

If you'd rather not use `npx`, just copy each skill folder into your agent's skills directory:

```
# Claude Code:
cp -R exo-upgrade-skills/* ~/.claude/skills/
# Codex:
cp -R exo-upgrade-skills/* ~/.agents/skills/
```

Either way: restart your agent session afterward so it picks up the new skills.

3. Run an upgrade

Open your agent in the folder you want to work from, then start the one entry-point skill:

```
/contentful-exo-upgrade
```

That's the only skill you invoke by hand. It sets things up and then guides you through the rest of the flow conversationally — you say things like "upgrade" and it runs the right skill for you. You do **not** run the other skills yourself.

What happens, step by step

1. **Setup + fetch your space.** The entry-point skill creates a local working folder (`.contentful-exo-upgrade/`, next to your `.git`), asks for your token / space / environment, and downloads your space into a local database. It then greets you with an inventory (content types, entry counts, locales). *Expect: a short Q&A, then a summary of your space.*

1. **(Optional) Codebase analysis.** If you have the space's website codebase handy, you can point the

agent at it to map how pages compose from components. *Expect: a markdown analysis of your pages and components. This step is optional and read-only.*

1. **Upgrade — generate the plan.** Say `upgrade` . The agent analyzes the space, generates the ExO model (Components, Experience Templates, Data Assemblies), publishes it to a **Modeling Workspace** you can review in the Contentful UI, and writes a plan. *Expect: a Modeling Workspace URL to open and review, plus a `plan.md` .* To keep a first upgrade small and low-risk, select just a few pages when you run codebase analysis — that scopes the model to that slice.

1. **Apply entries (the ETL) — only when you ask.** Say `apply the transformation / run the ETL` to migrate the actual content entries into ExO Fragment and Experience **drafts**. *Expect: draft entities created in your space for review. Nothing is published live; nothing existing is changed.* You can stop after any step, review, and continue later from the same folder. Ask the agent questions about your space at any point ("how many entries do I have?") — it answers from the local database.

Where the output goes

Everything the agent produces stays in the visible working folder it created:

<code><your-folder>/</code>	
<code> .contentful-exo-upgrade/</code>	
<code>.env</code>	your token + space + environment (kept local, gitignored)
<code>spaces/<space>-<env>/</code>	the downloaded copy of your space (local database)
<code>codebase-analysis/</code>	codebase analysis, if you ran it
<code>runs/<timestamp>/</code>	one folder per upgrade run:
<code>plan.md</code>	the human-readable upgrade plan
<code>report.md</code>	a summary of the run
<code>artifacts/</code>	the generated ExO model + supporting files

You can inspect or delete this folder any time. To start completely fresh, delete

`.contentful-exo-upgrade/` and run `/contentful-exo-upgrade` again.

4. Updating to a newer build

There's no auto-update during the beta. When we send a newer zip, re-run the install command from Section 2 (it overwrites the previous copy), then restart your agent session.

5. The skills, at a glance

You only ever invoke `/contentful-exo-upgrade` . It runs the others for you as the flow progresses — this table is just so you know what each one does.

Skill	What it does	When it runs
<code>contentful-exo-upgrade</code>	Entry point. Sets up the working folder, fetches your space, loads context, and greets you.	You run this first (the only one you invoke).
<code>contentful-exo-upgrade-codebase-analysis</code>	Analyzes your website codebase to map pages → components and the Contentful fields behind them.	Optional, early — if you point it at your code.
<code>contentful-exo-upgrade-plan</code>	The main upgrade workflow: analyze the space, generate the ExO model, publish the Modeling Workspace, write the plan.	When you say "upgrade".
<code>contentful-exo-upgrade-apply-modeling-workspace</code>	Publishes or updates the Modeling Workspace from the generated model.	Used by the upgrade step.
<code>contentful-exo-upgrade-apply-entries</code>	The entry-level ETL: migrates content entries into ExO Fragment / Experience drafts .	Only when you explicitly ask to run it.
<code>contentful-exo-upgrade-save-chat</code>	Saves the conversation history to a file.	Any time, to keep a record.

6. Glossary

- **ExO (Experience Orchestration)** — Contentful's model that separates content from presentation into Components (layout), Content Types (content), Data Assemblies (data), and Experiences / Fragments.
- **Modeling Workspace** — the Contentful UI surface where you review and approve the AI-generated model before anything is migrated.
- **Plan (`plan.md`)** — a human-readable summary of the current model, the proposed ExO model, and the mappings.
- **ETL / apply entries** — the step that moves your actual content entries into ExO, created as drafts for review.
- **Working folder (`.contentful-exo-upgrade/`)** — the local folder holding your downloaded space,

analysis, and per-run outputs. Visible and safe to delete.

7. Feedback

Please tell us what worked, what confused you, and anything that broke (with the step and any error text). That feedback is the whole point of the beta — thank you!