



MCS Engine: Quick Install & Get Started

Contents

- Quick start** 3
 - macOS (Apple Silicon) 3
 - Linux (x86_64) 3
- Requirements** 3
- Unpack the download** 4
 - macOS only: clear the download quarantine (required) 4
- Install** 4
 - What gets installed 4
 - Install options 5
- Set up your license** 5
- Run** 5
 - Web interface (default) 5
 - Don't have model keys yet? Get a few from OpenRouter 5
 - Loading a config (and trying the demos) 6
 - A few session concepts 6
 - Add MCS to your PATH (optional) 6
 - Command-line mode (Pro Max and Enterprise) 6
- Useful commands** 7
- Configuration (optional)** 7
- Upgrading and uninstalling** 7
- Support** 7

Multi-model Reasoning, Deliberation, and Decision System Osinix (osinix.com)

MCS runs a structured deliberation between multiple AI models. They reason, respond to each other, and produce a defensible result for high-stakes work (legal, financial, code review, and more). You bring your own model API keys; MCS orchestrates them.

How to read this guide. MCS installs and runs the same way on Linux and macOS: unpack a tarball, run `./install.sh`, add your license, and start the engine. Where a step differs by platform, look for a **macOS** or **Linux** note. Everything without such a note applies to both. macOS users have one extra one-time step (clearing the download quarantine); it is called out clearly below.

Quick start

Use the block for your platform. Replace `<version>` with the version in the filename you actually downloaded, and use the binary that matches your tier (`mcs_basic`, `mcs_pro`, `mcs_pro_max`, or `mcs_enterprise`; your package ships the one for your license).

macOS (Apple Silicon)

```
# Open Terminal: press Command+Space, type "Terminal", press Return.
cd ~/Downloads
tar xzf mcs-pro_max-<version>-macos_arm64.tar.gz

# Required one time: clear the macOS download block on the unpacked folder.
xattr -dr com.apple.quarantine mcs-pro_max-<version>-macos_arm64

cd mcs-pro_max-<version>-macos_arm64
./install.sh                                # installs to ~/osinix/mcs

cp /path/to/your/license.key ~/osinix/mcs/config/license.key
~/osinix/mcs/bin/mcs_<tier>                 # your tier's binary; the browser opens automatically
```

Linux (x86_64)

```
cd ~/Downloads
tar xzf mcs-pro_max-<version>-linux_x86_64.tar.gz
cd mcs-pro_max-<version>-linux_x86_64
./install.sh                                # installs to ~/osinix/mcs

cp /path/to/your/license.key ~/osinix/mcs/config/license.key
~/osinix/mcs/bin/mcs_<tier>                 # your tier's binary; the browser opens automatically
```

MCS opens its web interface in your browser. In it, add each model (its name, endpoint, and API key), then click **Start** to run the deliberation. The rest of this guide explains each step and the options.

The paths shown are the **defaults**. The installer lets you choose your own install, sessions, and temp directories; if you do, adjust the later paths to match.

Requirements

- **macOS:** macOS 12 (Monterey) or later, on **Apple Silicon** (M1, M2, M3, or newer). Intel Macs are not supported.
- **Linux:** x86_64.
- A **license key** from Osinix.
- An **API key** for each model provider you want on your panel (for example OpenAI, Anthropic, Google).

A network connection is needed for any **non-local** models you call (cloud providers) and for the once-a-day license check. Fully local models need no connection of their own. MCS keeps working offline for a grace period; see the Licensing Guide for details.

Unpack the download

Open a terminal:

- **macOS:** press **Command+Space**, type **Terminal**, press Return.
- **Linux:** open your terminal application.

Go to where the file downloaded and unpack it (use the exact filename you got):

```
cd ~/Downloads
tar xzf mcs-pro_max-<version>-macos_arm64.tar.gz    # macOS
tar xzf mcs-pro_max-<version>-linux_x86_64.tar.gz    # Linux
```

This creates a folder with the same name. Everything you need is inside it.

macOS only: clear the download quarantine (required)

macOS marks anything downloaded from the internet as quarantined and blocks it from running until you clear that mark. Do this once, on the unpacked folder:

```
xattr -dr com.apple.quarantine mcs-pro_max-<version>-macos_arm64
```

If you skip this, macOS will refuse to open the engine and it will look broken. This one command prevents that. (Linux has no equivalent step.)

If macOS still blocks it. After clearing the quarantine you should not see a block. If you ever do, open **System Settings > Privacy & Security**, scroll to the **Security** section, and if you see a message that a program from Osenix was blocked, click **Open Anyway**. Then run the command again.

Install

From inside the unpacked folder:

```
./install.sh
```

The installer is **interactive by default**. It asks three questions, each showing a default in brackets. Press Return to accept a default or type your own:

- **Install directory**, default `~/osinix/mcs`
- **Sessions directory**, default `<install>/sessions`
- **Temp directory**, default `/tmp`

It validates that each directory exists and is writable *before* copying anything, shows a summary, and installs.

What gets installed

Your package contains one engine binary matching your license tier. The installer detects it automatically and installs:

- the engine binary, to `<install>/bin/mcs_<tier>`
- a sample `mcs.conf`, to `<install>/config/mcs.conf`
- the Osenix public key for offline license checks, to `<install>/config/`
- the example `demos/` folder and `START_HERE.txt`, to `<install>/`

Your `license.key` is **not** in the package. You add it yourself (next section).

Install options

```
./install.sh -y                # non-interactive: accept all defaults
./install.sh --prefix=/opt     # install under /opt (may need sudo)
./install.sh --sessions-dir=DIR # set the sessions directory
./install.sh --temp-dir=DIR    # set the temp directory
./install.sh --upgrade         # reinstall, keep your existing mcs.conf
./install.sh --uninstall       # remove the installation
./install.sh --help            # show all options
```

If an installation already exists, the installer offers to upgrade (keep your config), do a fresh install (overwrite config), or cancel.

Once installed, everything MCS needs lives under `~/osinix/mcs`, so you can delete the unpacked download folder.

Set up your license

Place your license key file here (the same location on both platforms):

```
~/osinix/mcs/config/license.key
```

That is the default location the engine looks in. If you keep it elsewhere, point to it with `--license-key /path/to/license.key` or the `MCS_LICENSE_KEY` environment variable. The current working directory is never searched, so when you run MCS from somewhere other than the install folder, use one of those.

Don't have a key yet? Email licensing@osinix.com for a trial.

Run

Web interface (default)

```
~/osinix/mcs/bin/mcs_pro_max
```

Use the binary that matches your license tier: `mcs_basic`, `mcs_pro`, `mcs_pro_max`, or `mcs_enterprise`. (Your package ships the one for your tier.) If you installed somewhere other than the default, use that path instead.

MCS picks an open port in the range 8700-8799, starts its web server, and opens your default browser automatically (using `open` on macOS, `xdg-open` on Linux). The terminal also prints the URL (`http://localhost:<port>`) in case you want to open it yourself.

In the web interface you build your **panel**. For each model, click to add a participant, pick its **provider** from the menu (Anthropic, OpenAI, OpenRouter, HuggingFace), which fills in the endpoint for you, or choose **Custom** to type any other endpoint. Then enter the **model name** and that provider's **API key**. For example: provider Anthropic, model `claude-sonnet-4-6`, your key. Add two or three models, give the session a question, and click **Start**. The full walkthrough (modes, moderator, rounds) is in the User Guide.

Don't have model keys yet? Get a few from OpenRouter

MCS needs at least two models to deliberate. The quickest way to get several different models with one account is OpenRouter:

1. Sign up at openrouter.ai and create an API key.
2. Add a little credit (about 5 dollars is plenty for testing).
3. At openrouter.ai/models, copy a model "slug" like `openai/gpt-4o` or `google/gemini-2.5-flash`.

4. In MCS, pick **OpenRouter** as the provider (the endpoint fills in for you), paste the slug into **Model**, and paste your OpenRouter key into **API key**.

A good starter panel is two different models plus a moderator, so you can watch them actually deliberate rather than agree.

Loading a config (and trying the demos)

You don't have to build the panel by hand. If you already have a session `.conf`, load it in the GUI and the participants appear as editable fields. The install also includes ready-made examples in the `~/osinix/mcs/demos/` folder. Start with `hello_session`: load its `.conf`, and if it runs, your install and keys are good.

After loading any demo, you must replace its placeholders for each participant: the **model name**, the **endpoint**, and the **API key**, all from the same provider. One thing to watch: the GUI **masks the API key field**, so the `REPLACE_WITH_YOUR_API_KEY` placeholder is hidden. It is easy to miss that the key is still a placeholder, so type your real key over each one. Each demo's `README.md` (and a `NOTES.md` where present) explains its setup; read it before running.

To skip the auto-launch, or choose a browser:

```
mcs_pro_max --no-browser      # start the server, don't open a browser
mcs_pro_max -b firefox        # open in a specific browser
```

A few session concepts

A session is defined by a handful of things, set either in the GUI fields or in a `.conf` file. Three of them are **required**: the engine refuses to start without them.

- **Agenda (required)**: the question or task the panel works on.
- **Participants (required, at least 2)**: the models on the panel, each with a role. You need a minimum of two.
- **Moderator (required)**: one of those participants, named to run the session. It opens, keeps order, and writes the final result. The moderator name must match one of your participants.
- **Your own documents (optional)**: you can attach an input file for the panel to work on, such as a legal brief, a contract, a code file, or notes. In the GUI, use the upload control after you load or build the config.
- **Rounds and turns**: a *turn* is one model speaking; a *round* is one full pass through the panel. `max_rounds` limits how long a session runs.
- **Mode**: *roundtable* (models speak in a fixed order each round) or *directed* (the moderator chooses who speaks next; Pro and up).
- **Parallel**: models answer the same prompt at the same time within a round, independently, then converge (Pro and up).

This is just the shape of it. For how to write a config, the complete list of options, and the modes explained in depth, see the **User Guide**.

Add MCS to your PATH (optional)

So you can type `mcs_pro_max` from anywhere:

```
# macOS (zsh):
echo 'export PATH="$HOME/osinix/mcs/bin:$PATH"' >> ~/.zshrc

# Linux (bash):
echo 'export PATH="$HOME/osinix/mcs/bin:$PATH"' >> ~/.bashrc
```

Open a new terminal afterward for it to take effect.

Command-line mode (Pro Max and Enterprise)

Pro Max and Enterprise can run a full session in the terminal, without the web UI, from a session config file:

```
mcs_pro_max --cli -s session.conf
```

The session config format and a full walkthrough are in the **User Guide**. Run `mcs_pro_max --help` for every option.

Useful commands

```
mcs_pro_max --info      # version, license health, and where files live
mcs_pro_max --version   # version only
mcs_pro_max --show-config # print the resolved configuration and exit
mcs_pro_max --deactivate # free this machine so the license can move elsewhere
mcs_pro_max --help      # full usage
```

`--deactivate` releases this machine's seat with the license server so the license can move to another computer. There is **no separate activate command**: MCS binds a machine to your license automatically the first time you run it on that machine, as long as a seat is free. So to move your license, run `--deactivate` on the old machine, then simply start MCS on the new one; it activates on that first run. If every seat is already in use, MCS shows "**Paused (machine not licensed)**" until you deactivate one of the other machines. See the Licensing Guide.

Configuration (optional)

`~/osinix/mcs/config/mcs.conf` holds engine settings only. Sensible defaults are already set and most users change nothing. **API keys are not stored here**. Each belongs to a panel participant, as above.

A few settings you might touch:

- `server_port`: fix the web port (otherwise auto-chosen in 8700-8799). Useful for bookmarks or a reverse proxy.
- `max_rounds`: the default deliberation rounds when a session doesn't set its own (default 5). Per-tier ceilings cap it: Basic 8, Pro 20, Pro Max 40, Enterprise 100 (Enterprise admins can adjust this).
- `turn_delay_ms`: pause between turns for readability (default 1000).

Uncomment a line in `mcs.conf` to override it.

Upgrading and uninstalling

To upgrade, run the new package's installer with `--upgrade`; your `mcs.conf`, license, and sessions are preserved:

```
./install.sh --upgrade
```

To remove MCS:

```
./install.sh --uninstall
```

Support

- **General help and bug reports:** support@osinix.com
 - **Licensing and activation:** licensing@osinix.com
 - **Documentation:** <https://osinix.com/docs>
-

© Osinix. See `LICENSE.txt` for terms.