MADA

Setup Reference Card

LLNL's Multi-Agent Design Assistant • GitHub: https://github.com/llnl/mada | https://github.com/llnl/mada-tools • Documentation: https://software.llnl.gov/mada/develop/ | https://software.llnl.gov/mada-tools/develop/

MADA is a multi-agent design assistant platform that helps users build custom AI agents and orchestrate them using Microsoft Agent Framework. The broader MADA ecosystem also includes **MADA Tools**, a Python-based repository that provides reusable AI tooling for those agents, including MCP servers, skills files, and other resources that support agent workflows. Together, MADA and MADA Tools provide both the framework for creating agents and the tools they use to get work done.

1 · Install

> **Tip:** More information on installation instructions can be found in the documentation for each project:
>
> - MADA Installation
> - MADA Tools Installation

Installation of MADA requires Python >= 3.11 to be running on your machine.

```
python -m venv venv      # create virtual environment
source venv/bin/activate  # activate virtual environment
pip install mada         # install mada
pip install mada-tools    # install mada-tools
```

Verify the install — open a new terminal in your project, then:

```
mada --help               # help menu
mada-tools --help          # help menu
mada-tools --version        # version
mada-tools available-servers  # display available MCP Servers
```

2 · Connect a Model Provider — Model Access Gateway (MAG).

MADA needs a provider that serves LLM models.

Step 1 — Create a personal access token

Use the links below to create your keys and save them somewhere safe

- MAG https://amsc-docs-d762d2.gitlab.io/model-access-gateway/#quick-start
    - *Instructions to get an AmSC account, request access to a project, and generate PAT (Personal Access Token) to access MAG are listed above.*

Step 2 — Set environment variables

Export your environment variables so they can be seen by MADA.

```
# MAG
export AMSC_I2_API_KEY=<key>
export OPENAI_BASE_URL="https://i2-api.genesis.american-science-cloud.org/"
```

Step 3 — Edit Configuration

Edit the `mada/configs/example_helpful_critic.json` configuration file from the MADA repository to use exported environment variables.

```
# MAG
{
  "model": {
    "provider": "openai",
    "model": "gpt-5.6",
    "api_key": "${AMSC_I2_API_KEY}",
    "base_url": "${OPENAI_BASE_URL}"
  },
...
```

Step 4 — Connect

Ensure the connections are working by running the following commands.

```
mada-cli mada/configs/example_helpful_critic.json  # Use configuration file to start mada

# you should now see the output below
MADA Multi-Agent Orchestrator
==================================================

Chat Session Manager
==================================================
Existing sessions:
  1. 2026-05-26 07:07:54 | a7da842f-a44b-402a-b640-a1ae76d31f02

Options:
  [n] New session
  [s] Select session
  [d] Delete session
  [a] Delete ALL sessions
  [q] Quit

Enter choice: n
Created and selected a new session.

Initializing agents and MCP servers...
Status: Connection Successful: Orchestrator initialized with 0 MCP Servers and 3 agents
Model: gpt-5.4 from livai

No MCP tools available (LLM-only agents)

Chat with the agents (type 'quit' to exit)
--------------------------------------------------

You: Hi

Agents:
--------------------
Hi! How can I help?
```

3 · Per-Project Settings

Tip: More documentation on configuring MADA and MADA Tools can be found in their respective sets of documentation:

- MADA Configuration
- MADA Tools Configuration

We used the Example/Helpful Critic configuration above that only has LLM agents without any tools. These configurations can be changed to include MCP tools by filling in the `“mcp_servers”` key with a list of available MCP servers. See the MADA Vertex CFD configuration for an example.

These MCP Servers are defined in the mada-tools repo and have a corresponding configuration file. We use this configuration to start the MCP Servers so the LLM can have access to their tools. See the MADA Tools Vertex CFD configuration for an example. The mada-tools repo is designed to create and integrate modular MCP Servers as the user sees fit.

> **IMPORTANT:** The ports in both configuration files must match in order for the connection to be successful

```
mada-tools start-servers mada-tools/configs/vertex_cfd_servers.json   # start mcp servers with mada-tools
mada-cli mada/configs/orchestrator_vertex_cfd.json # Use mada to start llm agent interaction

# you should now see the output below
MADA Multi-Agent Orchestrator
==================================================

Chat Session Manager
==================================================
Existing sessions:
  1. 2026-07-07 12:16:21 | 02fff2a1-33a2-4e43-b9b4-0b558a581cb2

Options:
  [n] New session
  [s] Select session
  [d] Delete session
  [a] Delete ALL sessions
  [q] Quit

Enter choice: n
Created and selected a new session.

Initializing agents and MCP servers...
Status: Connection Successful: Orchestrator initialized with 2 MCP Servers and 3 agents
Model: gpt-5.4 from livai

Available tools:
  • JobManagementAgent: flux
  • SimulationAgent: vertex_cfd

Chat with the agents (type 'quit' to exit)
--------------------------------------------------

You: What tools are available

Agents:
--------------------
Available tools:

- JobManagementAgent — executes computational workflows
- SimulationAgent — creates simulation parameter runs and analyzes results

There is also:
- multi_tool_use.parallel — lets me call multiple specialist tools in parallel when appropriate

If you want, I can also explain when I'd use each one.
```

4 · Everyday Commands

TIP: More documentation on using MADA and MADA Tools can be found in their respective sets of documentation:

- Using MADA
- MADA Tools CLI and Using MADA Tools

Command	What it does
mada-tools	
mada-tools available-servers	View available servers.
mada-tools start-servers path/to/mcp_config.json	Start MCP servers.
mada-tools stop-servers	Stop running MCP servers.
mada-tools servers-status	Check status of MCP servers.
mada-tools restart-servers path/to/mcp_config.json	Restart MCP servers.
mada	
mada-cli path/to/agent_config.json	Start MADA in CLI mode.
mada-gradio path/to/agent_config.json	Start MADA in GUI mode. Uses Gradio.
mada-openai-api path/to/agent_config.json	Start MADA in OpenAI API mode.