Deep Space - MCP Guide Sections:

- Claude
- ChatGPT
- Claude Code
- Codex
- Troubleshooting

Connect Claude to Deep Space

Applies to **claude.ai in the browser, the Claude Desktop app, and Claude on mobile**. Connectors are attached to your Claude account, so a connector you add in one place is available everywhere you are signed in to Claude.

Plans: custom connectors are available on Claude Free (one connector), Pro, Max, Team and Enterprise.

Individual users (Free, Pro, Max)

1. Open Claude and go to **Settings** → **Connectors** (also reachable from **Customize** → **Connectors**).
2. Choose **Add custom connector**.
3. Fill in:
    - **Name:** `Deep Space`
    - **Remote MCP server URL:** `https://mcp.deepspacesync.com/mcp`
    - Leave the **Advanced settings** (OAuth client ID / secret) empty. Deep Space identifies Claude automatically.
4. Click **Add**, then **Connect**.
5. A Deep Space sign-in window opens. Sign in with your normal Deep Space account (email/password, Google, Microsoft or Apple, whichever you use for Horizon).
6. **Choose the workspace** you want Claude to read from and click **Allow**.
7. Back in Claude the connector shows as connected.

Teams and Enterprise organisations

An organisation **Owner** adds the connector once:

1. **Organization settings** → **Connectors** → **Add** → **Custom** → **Web**.
2. Enter the server URL `https://mcp.deepspacesync.com/mcp`, leave the OAuth client fields empty, click **Add**.

Each **member** then connects with their own Deep Space login:

1. **Customize** → **Connectors**, find **Deep Space**, click **Connect**.
2. Sign in to Deep Space, choose a workspace, click **Allow**.

Every member sees only what their own Deep Space role permits.

Using it in a chat

1. Start a new chat. Click the **+** (attachments / tools) button, open **Connectors** and make sure **Deep Space** is switched on for this chat.
2. Ask a question, for example:
    - "Which Deep Space projects can you see?"
    - "List the extensions of time on the Riverside project that are still submitted."

Confidential - For Approved Customers Only

- "Total the approved purchase orders on project 25040 by supplier."

Claude calls the Deep Space tools and shows the results. If a tool needs a project, Claude first lists your projects and uses the exact name or ID.

Claude Desktop and mobile

Connectors added to your Claude account appear automatically in Claude Desktop and Claude mobile. You can also add the connector from Claude Desktop itself via **Settings** → **Connectors** → **Add custom connector** with the same URL.

Switching workspace or disconnecting

- **Disconnect:** Settings → Connectors → Deep Space → Disconnect (or Remove). This ends the session immediately.
- **Switch workspace:** disconnect, then Connect again and choose the other workspace. A connector is bound to exactly one workspace at a time. If you regularly need two workspaces, add the server twice with different names (for example "Deep Space – Hospital" and "Deep Space – Retail").

See Troubleshooting and FAQ if anything does not work.

Connect ChatGPT to Deep Space

ChatGPT connects to remote MCP servers through **developer mode** (custom MCP connections). Availability depends on your ChatGPT plan and, for Business, Enterprise and Edu workspaces, on a setting your workspace administrator controls. ChatGPT's menu names change between releases; the steps below describe the current layout and the older one.

Before you start (Business / Enterprise / Edu)

A workspace **owner or admin** must allow custom MCP connections: **Workspace settings** → **Permissions & roles** → **Connected data** (or **Connectors**) → enable **Developer mode / Create custom MCP connectors**.

1. Turn on developer mode

- Current ChatGPT: **Settings** → **Security and login** → **Developer mode** → **On**.
- Older builds: **Settings** → **Connectors (or Apps & Connectors)** → **Advanced settings** → **Developer mode** → **On**.

2. Add the Deep Space connection

1. Open **Settings** → **Connectors** (older builds) or the **Plugins / Connectors** area and select **+** / **Create** / **Add custom connector**.
2. Fill in:
    - **Name:** `Deep Space`
    - **Description:** `Read-only access to my Deep Space workspace`
    - **Connection / MCP server URL:** `https://mcp.deepspacesync.com/mcp`. Include the `/mcp` path.
    - **Authentication: OAuth**. Leave client ID and client secret empty — Deep Space recognises ChatGPT automatically.
3. Confirm the safety notice and click **Create**.
4. ChatGPT sends you to the **Deep Space sign-in** page. Sign in with your Deep Space account, **choose a workspace**, review what the connection can do and click **Allow**.
5. Back in ChatGPT the connection is listed with the available tools (`get_connection_context`, `list_projects`, `list_extensions_of_time`, `get_extension_of_time`, `list_purchase_orders`, `get_purchase_order`). You only see the tools your Deep Space role permits.

3. Use it in a chat

1. Start a new chat and open the **+** (tools) menu.
2. Under **Developer mode** / **Connectors**, switch on **Deep Space**.
3. Ask, for example:
    - "Which Deep Space projects can you see?"
    - "List the purchase orders on project 25040 that are still open."

- "Summarise the extensions of time on the Riverside project."

ChatGPT asks for confirmation before running tools the first time. All Deep Space tools are read-only, so approving them cannot change anything in Deep Space.

ChatGPT desktop app

The ChatGPT desktop app supports the same custom MCP connections. Use **Settings** → **MCP servers (or Connectors)** → **Add**, enter the same URL, choose OAuth and select **Authenticate** when prompted.

Switching workspace or disconnecting

Open the connection in **Settings** → **Connectors**, choose **Disconnect** (or **Delete**), then add it again and choose the other workspace. Each connection is bound to exactly one workspace.

See Troubleshooting and FAQ if anything does not work.
.

Connect Claude Code to Deep Space

Claude Code (the terminal agent) connects to the Deep Space MCP server directly from your machine. Sign-in happens in your browser; Claude Code receives the result on a local address.

Requirements: Claude Code 2.1 or later and a browser on the same machine (or the `--no-browser` flow described below).

1. Add the server

Available to all your projects:

```
Shell
claude mcp add --transport http --scope user deepspace
https://mcp.deepspacesync.com/mcp
```

(Omit `--scope user` to add it only to the current project directory.)

2. Sign in

Either run, from a shell:

```
Shell
claude mcp login deepspace
```

or, inside a Claude Code session, type `/mcp`, select **deepspace** and choose **Authenticate**.

Your browser opens the Deep Space sign-in page. Sign in, **choose the workspace**, review the access statement and click **Allow**. The page confirms that you can return to the terminal.

> The consent page shows a notice that the application runs on your own computer and will receive the connection on a local address (`localhost`). That is expected for Claude Code. Only continue if you started the sign-in yourself.

No browser on this machine (SSH, containers):

```
Shell
claude mcp login deepspace --no-browser
```

Claude Code prints the sign-in URL. Open it in any browser, complete the steps, then paste the full address of the final page (it starts with `http://localhost:...`) back into the terminal when prompted.

3. Use it

Ask Claude Code as you would ask Claude:

- "Use Deep Space to list the projects I can see."
- "Pull the extensions of time on the Hospital project and put them in a table."

Run `/mcp` at any time to see the connection status. Tokens are refreshed automatically; if the refresh ever fails, `/mcp` shows the server as needing authentication and you can sign in again.

Sharing the configuration with a team

Add the server at project scope and commit the generated `.mcp.json`:

```shell
claude mcp add --transport http --scope project deepspace
https://mcp.deepspacesync.com/mcp
```

```json
{
  "mcpServers": {
    "deepspace": {
      "type": "http",
      "url": "https://mcp.deepspacesync.com/mcp"
    }
  }
}
```

Each team member still signs in with their own Deep Space account and sees only what their own role permits.

Removing

```shell
claude mcp logout deepspace     # ends the Deep Space session
claude mcp remove deepspace     # removes the server from your configuration
```

See Troubleshooting and FAQ if anything does not work.

.

Connect Codex to Deep Space

OpenAI Codex (CLI, desktop app and IDE extension) supports remote MCP servers with OAuth sign-in. Codex uses a local address on your machine to receive the result of the sign-in.

Codex CLI

1. Add the server:

```
Shell
codex mcp add deepspace --url https://mcp.deepspacesync.com/mcp
```

2. Sign in:

```
Shell
codex mcp login deepspace
```

   Your browser opens the Deep Space sign-in page. Sign in, **choose the workspace**, review the access statement and click **Allow**.

   > The consent page notes that the application runs on your own computer and will receive the connection on a local address. That is expected for Codex.

3. Use it: start `codex` and ask, for example, "Use Deep Space to list my projects" or "Show the purchase orders on the Hospital project".

Check status with `codex mcp list` and remove the server with `codex mcp remove deepspace`.

config.toml

The command above writes this to `~/.codex/config.toml`; you can also add it by hand:

```
None
[mcp_servers.deepspace]
url = "https://mcp.deepspacesync.com/mcp"
```

If your network blocks random local ports, pin the sign-in port:

```
None
mcp_oauth_callback_port = 5555
```

Codex desktop app and IDE extension

1. Open **Settings** → **MCP servers**.
2. Add a server with the URL `https://mcp.deepspacesync.com/mcp`.
3. Select **Authenticate** when prompted and complete the Deep Space sign-in, workspace choice and **Allow** in the browser.

Each person signs in with their own Deep Space account and sees only what their own role permits.

See Troubleshooting and FAQ if anything does not work.
.

Troubleshooting and FAQ

Sign-in and connection

"This connection request has expired." The sign-in must be completed within 10 minutes. Start again from the AI application.

"Please finish in the browser where you signed in." The page was opened in a different browser or profile than the one that started the sign-in. Return to the original window, or start again.

"No workspaces available." Your Deep Space account is not a member of any active Horizon workspace. Ask your workspace administrator for access. If you have more than one Deep Space account, sign out of Deep Space in this browser (hub.deepspacesync.com) and start the connection again with the right account.

The AI application reports `access_denied` with "This Deep Space account is not enabled for the MCP connection". During the alpha, Deep Space enables the connection for selected accounts only. Ask your Deep Space contact to enable your account.

The AI application reports `access_denied` or "You do not have access to that workspace". The workspace you chose is no longer available to your account, or the sign-in was cancelled. Reconnect and choose a workspace from the list shown.

The connector connects but shows only `get_connection_context` and `list_projects`. Your Horizon role for that workspace does not include the PO Register or Extension of Time features (the pages you can open in Horizon). Ask your workspace administrator to update your role, then disconnect and reconnect (or wait up to an hour; entitlements are refreshed automatically).

"Couldn't reach the MCP server" (Claude) / connection failed (ChatGPT). Check the URL: it must be exactly `https://mcp.deepspacesync.com/mcp`. Connectors created with the temporary early-alpha `…run.app` address no longer work; remove and re-add them. Try again in a minute; then contact Deep Space support with the time of the attempt.

Claude Code or Codex: the browser shows the consent page but the terminal never continues. The sign-in result is delivered to a local address (`http://localhost:<port>`). Make sure the browser runs on the same machine as the terminal, or use `claude mcp login deepspace --no-browser` and paste the final URL back into the terminal.

Using the tools

"No accessible project matches …" Use the exact project name or the numeric project ID from `list_projects`. The error lists close matches when it finds any. Projects you cannot open in Horizon are not available.

"Your Deep Space role does not include this module." See the role note above; the module is hidden until your role allows it.

"Your Deep Space sign-in has expired; reconnect the server." The saved sign-in could not be refreshed (for example after a password change or an administrator revoking sessions). Disconnect and connect again.

Results look incomplete. Lists are paged (up to 100 records per call). Ask the assistant for the next page, or narrow the request with a status or search filter.

Frequently asked questions

Can the assistant change anything in Deep Space? No. Every tool is read-only and declared as such to the AI application. There is no tool that creates, edits, approves or deletes.

Can it see other workspaces? No. A connection is bound to the single workspace you approved. To read another workspace, add a second connection and choose that workspace.

Can it see projects or modules I cannot open in Horizon? No. Projects are limited to those you can view, and modules follow your Horizon role. Every request to Deep Space is made with your own login.

Where are my credentials stored? Your Deep Space password is only ever entered on the Deep Space sign-in page. The MCP server keeps an encrypted, expiring session token so it can act on your behalf; the AI application only receives a short-lived connection token.

How do I stop it? Disconnect or remove the connector in the AI application. The session is ended immediately. Signing out of Deep Space everywhere (or an administrator removing your workspace access) also ends it within an hour.

Which AI applications are supported? Claude (web, Desktop, mobile, Team/Enterprise), Claude Code, ChatGPT (developer mode, web and desktop) and Codex. Any MCP client that supports OAuth 2.1 with PKCE and Streamable HTTP should work.