

Amp + reMarkable: Review Documents by Hand

This guide tells you how Amp sends documents to your reMarkable, reads what you write on them, and makes the changes. It also tells you how to set it up for yourself.

What it does

1. You ask Amp to make a document, for example a plan or a design.
2. Amp sends the document to your reMarkable as a PDF.
3. You write notes on the tablet with the pen.
4. You close the document. The tablet syncs your notes to the reMarkable cloud.
5. You tell Amp "done". Amp looks at each page, reads your handwriting, and changes the document.
6. Amp overwrites the existing PDF in place under the same stable title. Repeat the review until the document is correct.

Versioned copies (v2, v3, v4) are created only when you explicitly request them. Ask to preserve an earlier version or its annotations before an update.

The parts

- **reMarkable cloud.** Your tablet syncs to it. You need a reMarkable Connect subscription.
- **remarkable-mcp.** A small open-source program (github.com/SamMorrowDrums/remarkable-mcp). It connects to the reMarkable cloud and gives Amp tools to browse, upload, and render pages as images.
- **Token.** A key that lets remarkable-mcp use your reMarkable account. You make it one time.
- **Amp secret.** Amp keeps the token in your personal secrets, as `REMARKABLE_TOKEN`. The token never shows in a thread.
- **Amp skill.** A personal skill named `remarkable`. It tells Amp how to do the loop and starts remarkable-mcp when Amp needs it.

How it runs in Amp

```
Amp thread (any project, orb or laptop)
|
| 1. loads the "remarkable" skill
v
remarkable-mcp starts inside the thread's machine
|
| 2. reads REMARKABLE_TOKEN (your Amp secret)
```

```
      v
reMarkable cloud <---- sync ----> your tablet
```

- The skill is personal. Amp loads it in every thread you start: every project, no project, cloud orbs, and your own computer.
- Each thread starts its own copy of remarkable-mcp with `uvx`. No server runs all the time, and you do not need your laptop to be on.
- Because the program runs on the same machine as the thread, Amp can upload the PDFs that it makes in that thread.

How the skill works

The skill has two files:

- `SKILL.md`: the instructions that Amp follows.
- `mcp.json`: tells Amp to start remarkable-mcp and which tools to show.

When you ask for the reMarkable, Amp does these steps:

1. **Check.** It calls `remarkable_status` to make sure it is connected.
2. **Upload.** It puts the document in the top-level folder `/Amp Uploads`. It uses a different folder only if you name one. It does not make other folders.
3. **Read.** When you say "done", it renders each page as an image with your ink on it. The AI reads the handwriting from the image. You do not need an OCR service.
4. **Confirm.** It tells you what it read on each page before it changes anything. It asks about writing that it cannot read.
5. **Update.** It changes the source and overwrites the existing PDF or document in place under its stable title by default.

Safety rules in the skill:

- Delete is turned off. You delete items on the tablet yourself.
- If the connection drops during an upload, Amp checks the folder before it tries again. This stops duplicate uploads.

How to use it

Say things like:

- "Send this plan to my reMarkable."
- "I finished my notes on the plan. Read them and make the changes."
- "Update the existing document in place."

Tips:

- **Close the document on the tablet** before you say "done". Notes sync only after you close the document and the tablet is online.
- Write clearly, near the text that you want to change. Arrows, crossing out, and short notes all work.

- For long documents, tell Amp which pages have notes.

Set it up for yourself

You need: an Amp account, a reMarkable with Connect, and a computer with `uv` installed (`brew install uv` on a Mac).

Step 1. Make a token.

1. Go to my.remarkable.com/device/desktop/connect and get a one-time code.
2. On your computer, run:

```
uvx remarkable-mcp --register YOUR_CODE
```

This saves the token in the file `~/.rmapi`. Keep this file private (`chmod 600 ~/.rmapi`).

Step 2. Save the token as an Amp secret.

Run this command. It sends the file to Amp without showing the token:

```
tr -d '\n' < ~/.rmapi |  
amp secrets set REMARKABLE_TOKEN \  
--user --secret --data-file -
```

Step 3. Add the skill.

Ask a teammate who has the skill for its share URL, then tell Amp: "Import this shared skill into my personal skills: <URL>".

Or make it yourself. Tell Amp: "Create a personal skill named remarkable", and give it this `mcp.json`:

```
{  
  "remarkable": {  
    "command": "uvx",  
    "args": ["remarkable-mcp"],  
    "includeTools": [  
      "remarkable_status", "remarkable_browse",  
      "remarkable_recent", "remarkable_search",  
      "remarkable_read", "remarkable_image",  
      "remarkable_upload", "remarkable_markdown_to_pdf",  
      "remarkable_mkdir", "remarkable_move",  
      "remarkable_rename"  
    ]  
  }  
}
```

Step 4. Test it. Start a new thread and say: "Send a one-page test PDF to my reMarkable."

Write on it, close it, and say "done".

Limits and things to know

- **Old threads** do not know about a new skill. Tell them "run reload_skills", or start a new thread.
- **Orbs that started before you saved the secret** do not have the token. Start a new thread.
- **The token gives full access** to your reMarkable account. Every orb you start has it. If you lose trust in it, remove the device at my.remarkable.com and make a new token.
- **Shared (multiplayer) threads:** other people who prompt your thread can use your reMarkable through the skill.
- **Puck** can read the skill but cannot run remarkable-mcp, so it cannot send to your tablet.
- **Times can be wrong.** The tablet clock can differ from the cloud. Amp looks at the pages, not the "last modified" time.

If something goes wrong

- **Amp says it has no reMarkable tools:** tell it to load the `remarkable` skill and run `reload_mcp`.
- **"No reMarkable token found":** the `REMARKABLE_TOKEN` secret is missing. Do Step 2 again, then start a new thread.
- **Amp cannot see your notes:** close the document on the tablet, check Wi-Fi, wait a few seconds, and ask again.