Integrations

- AI MCP Server

AI MCP Server

VeloxFactory can be operated directly through an AI assistant. A dedicated **MCP Server**, available as an add-on for VeloxFactory customers, exposes report configs, rendering, print dispatching, and user administration through the Model Context Protocol (MCP), so any MCP-compatible AI assistant can manage VeloxFactory through plain conversation instead of the frontend or hand-written API calls.

> **In short:** if you can describe the task, your assistant can do it. No API documentation to read, no frontend clicks to remember, and not tied to any single AI provider.

What It Lets You Do

The MCP Server gives your assistant access to everything VeloxFactory can do, organised into a few practical areas.

Manage reports by talking, not clicking. Create, update, and organise report configs, connection configs, and report contexts through conversation. "Clone the KanBan label as a QA version and rename the fields" is a request your assistant can carry out directly.

Render and preview on demand. Ask for any report to be rendered with real or example data, and get a finished PDF back, opened automatically, ready to view. No need to log into the frontend or assemble an API request by hand. This includes swapping an image in on the fly for a single render, a local path, a remote URL, or a Base64 upload, without editing the report configuration itself.

Handle files without the round-trip. Uploading a new `.jrxml` template, swapping a logo, or attaching a resource image is as simple as pointing your assistant at the file on your computer. It takes care of the technical side.

Dispatch print jobs. Have a rendered document sent straight to a print task, check its status, or mark it as printed, the same dispatching VeloxFactory already does, just requested in a sentence.

Administer users and access. Create accounts, adjust permissions, reset passwords, or issue API tokens for new integrations, day-to-day administration handled conversationally.

Keep track of history. Pull up past renderings, review what was generated when and by whom, and clean up records that are no longer needed.

At a Glance

Coverage is complete, every part of VeloxFactory is reachable, not just the rendering side:

Area	What's covered
Users	List, view, create, and update accounts; change passwords; enable/disable access; issue or revoke API tokens
Report Contexts	List, view, create, update, and delete
Connection Configs	List, view, create, update, delete, and test a database connection
Common Report Resources	List, view, create, update, and delete (logos, images, shared files)
Report Configs	List, view, create, update, delete, manage resources/parameters/fields, generate previews, and render (including per-render image overrides via `resourceOverrides`)
Report Resources	Link or unlink a Common Report Resource to a report
History Records	List, view, create, update, delete, print, and clean up linked print tasks
Print Tasks	List, view, create, update, delete, and set status/printed state
Local Utilities	Automatically open rendered PDFs, URLs, and output folders on your machine

55 tools in total, covering the full VeloxFactory API.

What This Looks Like in Practice

A few examples of the kind of request the MCP Server is built for:

- "Render the A5 KanBan label for article 4561287-154 and show me the PDF."
- "I designed a new label in Jaspersoft Studio, onboard it as a report config."
- "Create a new user for our warehouse lead with admin rights."
- "Print this delivery note to the shop floor printer and let me know once it's done."
- "Show me every report config connected to the QA database."

- "Render this label but use this product photo instead of the linked one, just for this one print."

Each of these would otherwise mean a frontend session or a small API script. With the MCP Server, they are a single message.

Built-In Know-How

Access to the tools is only half the story. Alongside the MCP Server, kiwi software also provides a companion **Skill**, a packaged layer of accumulated know-how that teaches the assistant how to use those tools well, not just that they exist.

In practice, this is what separates "the assistant can technically call the API" from "the assistant makes the same good judgment calls a person who knows VeloxFactory inside out would make": choosing the right output format for the situation, batching multiple renders sensibly instead of opening five windows in a row, staying within an instance's rate limit, and following VeloxFactory's own conventions rather than generic assumptions.

The most visible example: with the Skill in place, the assistant can **design a brand-new report template from scratch** (a label, delivery note, or certificate) without ever opening Jaspersoft Studio. It knows the exact conventions a `.jrxml` file needs to work in VeloxFactory (parameter naming, example values, and so on) and checks its own work with a preview before calling the template done.

> □□ The Skill ships together with the MCP Server as part of the same add-on, there is nothing separate to install or configure.

Who It Is For

The MCP Server is aimed at teams that already run VeloxFactory and want a faster way to handle the routine work around it, onboarding new report templates, day-to-day administration, one-off renderings, or quick checks on what has already been generated.

It is a natural fit for anyone comfortable working with an AI assistant who would rather describe a task than click through a frontend or write an API call, including teams without dedicated developer resources to spare on integration work. Since MCP is an open, provider-independent standard, it is not limited to a single AI product.

Getting It

The MCP Server is built and maintained by kiwi software as an add-on to a VeloxFactory license. It runs locally alongside your MCP-compatible AI assistant (Claude being one example) and connects to your VeloxFactory instance using your own API credentials. Nothing about your data passes through a third party.

Setup is handled together with kiwi software as part of onboarding, typically alongside the initial VeloxFactory rollout or added to an existing installation.

> ℹ **Interested?** Reach out via the Get VeloxFactory page, or try the live demo first at demo.veloxfactory.kiwi-software.dev.