https://cloudblue.com

Documentation Developer Resources

Connect MCP

This article has been generated from the online version of the documentation and might be out of date. Please, make sure to always refer to the online version of the documentation for the up-to-date information.

Auto-generated at September 9, 2026

Give your AI agent a seat at the table in CloudBlue Connect®

CloudBlue Connect® exposes its core capabilities through the **Model Context Protocol (MCP)**, an open standard for connecting AI agents to external systems. Once you connect your agent (e.g. Claude, Cursor, Copilot, or your own), it can look up a subscription, file a Helpdesk case, pull the latest price list, or check a fulfillment request — the same actions you'd take in the Connect UI or through the API, but driven by a conversation instead of a screen.

What is MCP?

MCP is a standardized way for an AI agent to discover what a system can do and act on it, without you or your developers writing custom integration code for every tool the agent might need to use. Think of it as a USB-C port for AI applications: instead of a different cable (a different bespoke integration) for every device, one standard connector works everywhere it's supported.

Before MCP, connecting an AI agent to Connect meant writing and maintaining custom code against Connect's REST API for every workflow you wanted to automate. With MCP, Connect describes its own capabilities in a form your agent already knows how to read, and your agent takes it from there.

How it works

Three parts, each doing one job:

Platform or Component	Description
Connect (the MCP server)	Publishes a catalog of tools—things like "list pending fulfillment requests" or "create a price list"—and executes them when called.
Your MCP client	The bridge inside your AI tool (Claude Desktop, Claude Code, Cursor, and others) that connects to Connect, authenticates, and hands your agent the tool catalog.
Your agent	The assistant you're actually talking to. It reads your request, decides which Connect tools it needs, and calls them—in sequence, if the task takes more than one step.

A typical exchange looks like this:

```
You:    "Show me every open Helpdesk case with High priority."
Agent:  discovers Connect's Helpdesk tools → calls helpdesk_list_cases
        (priority=High, state=open) → reads the result → summarizes it for you
```

Nothing about this changes what Connect will let an agent do on your behalf. An agent connecting through MCP is bound by the same account permissions as any other integration.

What you can do with it

As a Vendor:

- **Catalog and pricing.** Point your agent at your own product catalog or price list and have it reconcile that against what's already in Connect, flagging gaps instead of you doing it line by line.
- **Usage reporting.** Hand your agent a usage file in whatever shape it comes in, and let it map that into the structure Connect expects—instead of reformatting it by hand every reporting period.
- **Fulfillment and support.** Ask your agent to check on a batch of fulfillment requests, or triage and respond to Helpdesk cases, the same way you'd ask a teammate.

As a Distributor:

- **Onboarding and contracts.** Let your agent cross-reference a new vendor or reseller against your CRM and confirm what's ready to be issued in Connect.
- **Marketplace integration.** Use your agent to help map your catalog and listings into a marketplace you're integrating with.
- **Reporting.** Ask for a custom rollup of subscriptions, usage, or billing data that doesn't exist as a canned report today.

These are starting points, not the limit of what's possible. Anything exposed as a Connect MCP tool is fair game for your agent to combine into a larger workflow.

What MCP is not

- **It's not a replacement for the Connect REST API.** MCP is a second door into the same house, built for agents rather than application code. If you're building a traditional integration, the REST API is still the right tool.
- **It's not a bulk data export.** MCP tools operate the way the Connect UI does — one scoped action at a time — not as a firehose for pulling your entire dataset.
- **It's not unrestricted access.** Every tool call is checked against your account's permissions before it runs. An agent can only do what the account behind it is already allowed to do.