WATS Feature Release Note – MCP Server for On-Premises Installations (Beta) – 26.2

Major Feature Areas

MCP Server for On-Premises Installations (Beta)

The WATS MCP Server lets AI assistants such as Claude, ChatGPT, Microsoft Copilot Studio and VS Code work directly with WATS data through the Model Context Protocol (MCP). It has been available in beta for WATS Cloud since June 2026. With this release it is also available for on-premises installations as a beta feature.

An MCP client connects to your WATS server, signs in as a WATS user, and can then call a set of analysis tools on that user's behalf. The assistant sees only what the user can see in WATS: every tool runs under the connected user's permissions and product group or level restrictions, and a tool the user lacks permission for is not even listed to the client.

Tools

More than 60 tools, the same that power the Alvea Assistant in WATS Cloud, organised as a drill-down from overview to single unit:

- Overview and health. Where does production stand right now, and what needs attention first.
- Exploration. Slice yield, repairs and unit flow by any report dimension to find where problems concentrate.
- Scoped analysis. Analyse the test steps of one product and test operation: failures, trends, retests, timing, OEE and Gage R&R.
- Deep dive. Examine one step, component, report or asset in detail.
- Single unit. Follow one serial number through its history.
- Utilities. Look up valid filter values, products and systems, and create links that open the matching WATS report with the filter already applied.

The complete list of tools with a description of each is shown in Control Panel > Alvea & connectivity > MCP tools. The page requires the MCP Administration permission, which the Administrator and Manager roles have by default.

What is different on-premises

- **No Alvea Assistant**. The in-app assistant depends on cloud AI services. Its pages, permissions and account switch are hidden. This release covers the MCP Server only.
- **No documentation search**. SearchDocuments and FetchDocuments use a search index hosted by WATS in Azure. They are not listed on-premises, and the assistant is told that documentation search is unavailable. For documentation questions, use Help chat at https://helpchat.wats.com or from the Help menu in WATS. All other tools are available.

- **Network requirements**. Hosted assistants must reach your WATS server, and OAuth requires your WATS server to reach the assistant vendor's website. See below.

Otherwise tool behavior, permissions and administration pages are identical to WATS Cloud.

Enabling MCP

MCP is off by default. The account owner, an Administrator or a user with the Account permission turns it on with Enable MCP in Control Panel > Account. Turning it off blocks every connection and token immediately, and all MCP pages show that MCP is disabled.

Three permissions control who can use it:

- **Beta features**. Required for all MCP use during the beta. Now enabled by default on-premises and granted to Administrator, Manager and Analyzer.
- **MCP Access**. Connect MCP clients and create tokens. Granted to Administrator and Manager.
- **MCP Administration**. See and revoke all user's connections and tokens, and switch tools on or off. Granted to Administrator and Manager.

Administration pages under Control Panel > Alvea & connectivity:

- MCP tools. Every tool with description and an Enabled switch for the whole installation. Tools that belong together, such as the two Gage R&R tools or the report viewer and its attachment reader, switch as one. The toolbar shows enabled count out of total.
- OAuth sessions. All users' connected apps: app name, user, created, last used, status, request count. Revoke.
- JWT tokens. All users' tokens: name, user, created, last used, expiry, status, request count. Revoke.

Token and session creation and revocation, and tool switches, are recorded in the change log.

Connecting to the MCP Server

The MCP address is your WATS URL plus /mcp, shown with a copy button under My Settings > MCP. The tab is visible to users with MCP Access when MCP is enabled.

Two ways to sign in. Both act as the connected user, with that user's permissions and restrictions.

OAuth connection

Recommended for hosted assistants and for local clients that support it. Add the MCP URL in the assistant. It opens WATS in the browser, where you log in if needed and reach the consent page.

The consent page leads with the app's verified domain, for example "claude.ai wants to connect to your WATS account", and names the WATS user the app will act as. It warns the first time an app domain connects to your installation, and if the app sends the access to a different domain than its own. Technical details shows the app's self-declared name, identity URL, return address, signed-in user and WATS instance. Nothing is granted until you click Connect. If you did not start this from an app, click Cancel.

WATS identifies OAuth clients with Client ID Metadata Documents (CIMD): the client identifies itself with an HTTPS URL on its vendor's domain, and WATS downloads a small JSON document from it to verify the client's name and allowed return addresses. There is no client registration, and nothing is stored until a logged-in user has approved. The flow is OAuth 2.1 with PKCE, and WATS publishes standard discovery metadata, so clients find the login flow on their own. Most clients select CIMD automatically; if asked, choose Client ID Metadata Document.

A connected app receives short-lived access tokens that it renews automatically. The connection expires after 90 days without use and can be revoked at any time under My Settings > MCP or Control Panel > OAuth sessions. Revoked connections stay listed as Revoked for at least 24 hours, then are removed.

JWT token

For clients that cannot do OAuth or do not support CIMD, such as Copilot Studio, and for closed networks. Create a token under My Settings > MCP > JWT tokens: name it, choose 7, 30 or 90 days or a custom date up to 365 days ahead, and save. The token is shown once. Copy it immediately, WATS does not store it.

Paste it into the client as a bearer token:

Authorization: Bearer <token>

A JWT token is accepted only by the MCP endpoint. It acts as you with all your permissions and restrictions, so treat it like a password: one token per client, short lifetimes, revoke on suspected leak. Users revoke their own tokens under My Settings, administrators any token under Control Panel > JWT tokens. Revoked and expired tokens are removed automatically after their expiry date.

Every MCP request re-checks that the user exists, is not expired, and still holds MCP Access and Beta features. Losing either cuts off the user's connections and tokens on the next request.

Network requirements for on-premises installations

- Hosted assistants (such as Claude web, desktop and mobile, ChatGPT, Copilot Studio) run in the vendor's cloud and connect over the internet. They need an inbound HTTPS path to /mcp and a certificate from a public certificate authority. Internal CA and self-signed certificates are not trusted by the vendors' servers.
- Local clients (Claude Code, Claude Desktop, VS Code) run on the user's machine and only need to reach WATS over LAN or VPN. An internal certificate works.
- OAuth requires the WATS server to reach the vendor's website to download the client's metadata document: public DNS resolution for the vendor's domain and outbound HTTPS on port 443. The metadata URL must resolve to a public address, so this applies to local clients as well. If either is missing, login fails before the consent page.
- JWT tokens need no DNS resolution of the vendor's domain, no outbound HTTPS and no browser login. WATS fetches nothing and sends no outbound traffic. With a local client, the whole setup stays inside your network.