

Hello Mainframe How are you today ?

Using Zowe in an AI world with MCP AI Agents

Andrew Toews, Gestell
Joe Winchester, IBM

The Mainframe Knowledge Crisis



240B+

lines of COBOL
in production today



70%

of mainframe devs
retiring by 2030



\$3T+

daily transactions
run on mainframes

The gap between retiring expertise and incoming talent is widening

Model Context Protocol

A universal standard that lets AI assistants connect to external tools and data sources.



Tools

AI can call functions — list jobs, read datasets, submit JCL



Prompts

Pre-built workflows — onboarding, failure diagnosis, code review



Resources

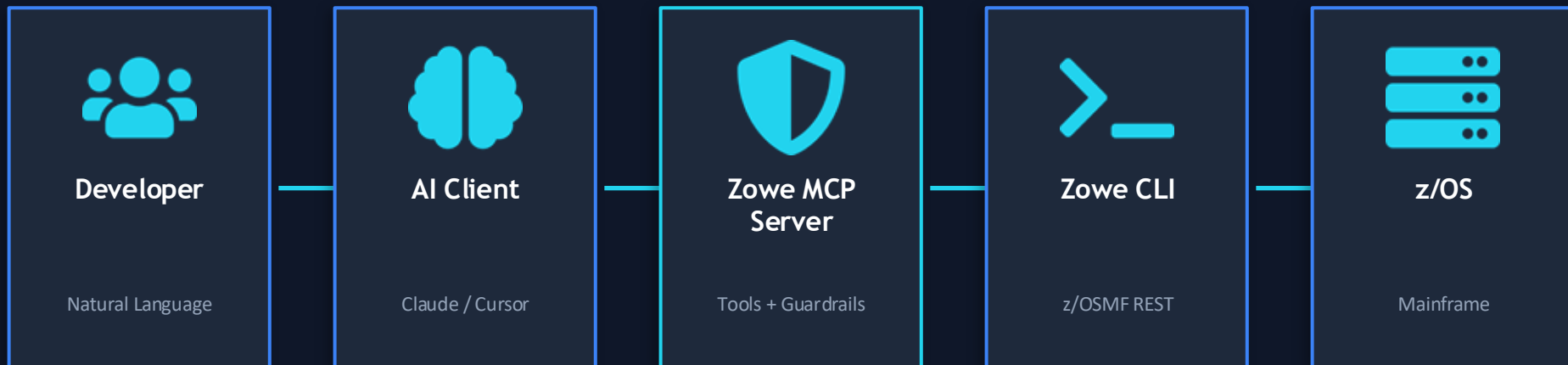
Reference material — z/OS concepts, JCL syntax, error codes



Think of it as
USB-C for AI

One protocol,
any data source

Architecture



"Show me why the payroll job failed last night"

→ The AI translates natural language into the right Zowe CLI commands, retrieves spool output, and explains the ABEND code in plain English.

10 MCP Tools

Each tool has a safety classification



list_jobs

List jobs by owner/prefix/status



get_job_status

Get detailed job information



get_job_output

Retrieve spool + error analysis



submit_job

Submit JCL for execution



list_datasets

List datasets by HLQ pattern



list_members

List PDS members



view_dataset

Read COBOL, JCL, copybooks



tso_command

TSO commands with guardrails



console_command

MVS console with guardrails



explain_error

Look up z/OS error codes



Safe (read-only)



Cautious (side effects)



Varies (guardrail-checked)

Built-In Guardrails

The AI cannot execute destructive commands. Period.



SAFE

Read-only operations

LISTDS, LISTCAT,
DISPLAY, STATUS



CAUTIOUS

Side effects — logged & flagged

SUBMIT, ALLOCATE,
ALTER, COPY



BLOCKED

Cannot be executed by AI

DELETE, CANCEL,
PURGE, FORCE, FORMAT

Before using zowe you will need:

- Node Package Manager needed

- Bun install: `curl -fsSL https://bun.sh/install | bash`

Demo

End-to-End Workflow Execution

```
| DEMO.SAMPLE.COPYLIB | present |
| DEMO.SAMPLE.JCL | present |
| DEMO.SAMPLE.OBJ | present |
| DEMO.SAMPLE.LOAD | present (RECFM=U, LRECL=0) |
| DEMO.SAMPLE.SYSDEBUG | present |

Required members:
| Library | Required members | Status |
|----|----|----|
| DEMO.SAMPLE.COPYLIB | CUSTCOPY, TXNCPY, TRANREC | all present |
| DEMO.SAMPLE.COBOL | INITVSAM, SETBAL, GENTRAN, PROCTXN, VIEW | all present |
| DEMO.SAMPLE.JCL | DEFCAT, DEFACC, DEFTXN, INITVSAM, VERIFY, CREATEAC, SETBAL, GENTRAN, PROCTXN, VIEW, VIEWCOMP | all present |

Bootstrap verdict: READY.

Workflow (Customer Account Path)
| Step | Job ID | Retcode |
|----|----|----|
| DEMO.SAMPLE.JCL(DEFCAT) | JOB00167 | CC 0000 |
| DEMO.SAMPLE.JCL(DEFACC) | JOB00168 | CC 0000 |
| DEMO.SAMPLE.JCL(DEFTXN) | JOB00169 | CC 0000 |
| DEMO.SAMPLE.JCL(INITVSAM) | JOB00170 | CC 0000 |
| DEMO.SAMPLE.JCL(VERIFY) | JOB00171 | CC 0004 |
| DEMO.SAMPLE.JCL(CREATEAC) | JOB00172 | CC 0004 |
| DEMO.SAMPLE.JCL(SETBAL) | JOB00173 | CC 0004 |
| DEMO.SAMPLE.JCL(GENTRAN) | JOB00174 | CC 0000 |
| DEMO.SAMPLE.JCL(PROCTXN) | JOB00175 | CC 0004 |
| DEMO.SAMPLE.JCL(VIEW) | JOB00176 | CC 0004 |
| DEMO.SAMPLE.JCL(VIEWCOMP) | JOB00177 | CC 0000 |

All retcodes are in the allowed set (CC 0000/0004/0008), so workflow continued successfully.

Validation
1. Missing DEMO.SAMPLE.* datasets: none.
2. VSAM objects:
- DEMO.ACCOUNTS.CLUSTER: present
- DEMO.TXN.CLUSTER: present
- DEMO.USERCATALOG: not a valid DSN pattern in this environment (qualifier length limit); the actual created catalog from DEFCAT is DEMO.VSAM.CATALOG, and it is present.
3. VIEWCOMP: JOB00177, CC 0000, spool retrieved successfully.

Final result: PASS (customer account workflow completed end to end).
```

Summarize recent commits
? for shortcuts
47% context left

AI submits JCL, monitors jobs, validates return codes, checks VSAM objects — all through natural language

Demo

COBOL Source Inspection & Diagnostics

```
STMT NO. MESSAGE
 2 IEFC001I PROCEDURE ELAXFCDC WAS EXPANDED USING SYSTEM LIBRARY SYS1.PROCLIB
 4 IEFC002I INCLUDE GROUP ELAXF WAS EXPANDED USING SYSTEM LIBRARY SYS1.PROCLIB
 50 IEFC001I PROCEDURE ELAXFLNK WAS EXPANDED USING SYSTEM LIBRARY SYS1.PROCLIB
 52 IEFC002I INCLUDE GROUP ELAXF WAS EXPANDED USING SYSTEM LIBRARY SYS1.PROCLIB
 ICH70001I IBMUSER  LAST ACCESS AT 23:56:31 ON SATURDAY, FEBRUARY 21, 2026
 IEFA111I VIEWCOMP IS USING THE FOLLOWING JOB RELATED SETTINGS:
          SMA=BELOW,TIOT SIZE=64K,DSEHQSHR=DISALLOW,G0GB1A5=JOB
 IEF236I AL...

• Called zowe-mcp-server.zowe_walt_async_task({"task_id":"b022c123-2933-4cd7-87f8-6ba0c1a0b0c2","max_wait_ms":120000})
  ↳ Task b022c123-2933-4cd7-87f8-6ba0c1a0b0c2 completed.
    Started: 2026-02-22T03:02:26.505Z
    Completed: 2026-02-22T03:02:57.933Z
    Command: zowe zos-jobs view spool-file-by-id JOB00177 106 --base-profile global_base --zosmf-profile zosmf
    Success: true
    Exit code: 0
    **stdout**
    12/05 V3 R1 BINDER          23:57:49 SATURDAY FEBRUARY 21, 2026
    BATCH EMULATOR JOB(VIEWCOMP) STEP(LINK      ) PGM= HEWL          PROCEDURE(LINK      )
    IEW227RI B352 INVOCATION PARAMETERS - LIST,MAP
    IEW2322I 1220 1          INCLUDE 00J0000

    1          *** M O D U L E M A P ***

    -----
    CLASS  C_CODE          LENGTH =          1808  ATTRIBUTES = CAT,          LOAD, RMODE=ANY
    OFFSET =          0  IN SEGMENT 001          ALIGN = DDLWORD

    SECTION  CLASS          ----- SOURCE -----
    OFFSET  OFFS...

• Called zowe-mcp-server.zowe_walt_async_task({"task_id":"483ee6fd-fd8c-4d78-b2db-0e518e9b4aab","max_wait_ms":120000})
  ↳ Error: Interrupted

■ Conversation Interrupted - tell the model what to do differently. Something went wrong? HLT `feedback` to report the issue.

• The VIEW and VIEWCOMP jobs had some warnings (were not CC Code 0)... Can you pull from JES and report to me warnings and how to fix?

• Working (4s + esc to interrupt)

• Implement (feature)

? for shortcuts
77% context left
```

AI retrieves compiler spool output, analyzes diagnostics, reads COBOL source, and suggests patch-level fixes

```
model: gpt-5.3-codex high /model to change
directory: ~/Projects/gestell/zowe/zowe-mcp-server
```

Tip: New Try the **Codex App** with 2x rate limits until April 2nd. Run 'codex app' or visit <https://chatgpt.com/codex?app-landing-page=true>

Review demo/README.md and do the customer account workflow end to end

I'll inspect demo/README.md first, identify the exact customer account workflow steps, then execute them in order and report each result with any fixes needed along the way.

Explored

- ↳ Read README.md
- ↳ List demo

I found that demo/README.md just points to the instruction set, so I'm now pulling the workflow docs (00 through 40) to execute the exact customer account path end to end.

Explored

- ↳ Read 20-run-workflow.md, 10-bootstrap.md, 00-overview.md, 40-failure-handling.md, agent-end-to-end.md, 30-validate.md

I've mapped the full instruction chain and will run it exactly: bootstrap (DEMO.SAMPLE.* presence, attribute remediation, uploads, member checks), ordered job execution (DEFKAT through VIEWCOMP), then validation (DEMO.* dataset checks + VIEWCOMP spool). I'll also keep a run log of step/job_id/retcode and only stop early if a defined blocking condition occurs.

Calling zowe-mcp-server.zowe_list_datasets({"pattern":"DEMO.SAMPLE.*"})

Executing dataset listing step (25s • esc to interrupt)

```
model: gpt-5.3-codex high /model to change
directory: ~/Projects/gestell/zowe/zowe-mcp-server
```

Tip: New Try the **Codex App** with 2x rate limits until April 2nd. Run 'codex app' or visit <https://chatgpt.com/codex?app-landing-page=true>

> Review demo/README.md and do the customer account workflow end to end

• I'll inspect demo/README.md first, identify the exact customer account workflow steps, then execute them in order and report each result with any fixes needed along the way

• **Explored**

- ↳ Read README.md
- ↳ List demo

• I found that demo/README.md just points to the instruction set, so I'm now pulling the workflow docs (00 through 40) to execute the exact customer account path end to end.

• **Explored**

- ↳ Read 20-run-workflow.md, 10-bootstrap.md, 00-overview.md, 40-failure-handling.md, agent-end-to-end.md, 30-validate.md

• I've mapped the full instruction chain and will run it exactly: bootstrap (DEMO.SAMPLE.* presence, attribute remediation, uploads, member checks), ordered job execution (DEFKAT through VIEWCOMP), then validation (DEMO.* dataset checks + VIEWCOMP spool). I'll also keep a run log of step/job_id/retcode and only stop early if a defined blocking condition occurs.

• **Calling** zowe-mcp-server.zowe_list_datasets({"pattern":"DEMO.SAMPLE.*"})

• **Executing dataset listing step** (24s • esc to interrupt)

> Summarize recent commits

? for shortcuts

96% context left

```
| DEMO.SAMPLE.LOAD | present (RCFM=U, LRECL=0) |
| DEMO.SAMPLE.SYSDEBUG | present |
```

Required members:

```
| Library | Required members | Status |
|---|---|---|
| DEMO.SAMPLE.COPYLIB | CUSTCOPY, TXNCOPY, TRANREC | all present |
| DEMO.SAMPLE.COBOL | INITVSAM, SETBAL, GENTRAN, PROCTXN, VIEW | all present |
| DEMO.SAMPLE.JCL | DEFCAT, DEFACC, DEFTXN, INITVSAM, VERIFY, CREATEAC, SETBAL, GENTRAN, PROCTXN, VIEW, VIEWCOMP | all present |
```

Bootstrap verdict: READY.

Workflow (Customer Account Path)

```
| Step | Job ID | Retcode |
|---|---|---|
| DEMO.SAMPLE.JCL(DEFCAT) | JOB00167 | CC 0000 |
| DEMO.SAMPLE.JCL(DEFACC) | JOB00168 | CC 0000 |
| DEMO.SAMPLE.JCL(DEFTXN) | JOB00169 | CC 0000 |
| DEMO.SAMPLE.JCL(INITVSAM) | JOB00170 | CC 0008 |
| DEMO.SAMPLE.JCL(VERIFY) | JOB00171 | CC 0004 |
| DEMO.SAMPLE.JCL(CREATEAC) | JOB00172 | CC 0004 |
| DEMO.SAMPLE.JCL(SETBAL) | JOB00173 | CC 0004 |
| DEMO.SAMPLE.JCL(GENTRAN) | JOB00174 | CC 0008 |
| DEMO.SAMPLE.JCL(PROCTXN) | JOB00175 | CC 0004 |
| DEMO.SAMPLE.JCL(VIEW) | JOB00176 | CC 0004 |
| DEMO.SAMPLE.JCL(VIEWCOMP) | JOB00177 | CC 0008 |
```

All retcodes are in the allowed set (CC 0000/0004/0008), so workflow continued successfully.

Validation

1. Missing DEMO.SAMPLE.* datasets: none.

2. VSAM objects:

- DEMO.ACCOUNTS.CLUSTER: present
- DEMO.TXN.CLUSTER: present
- DEMO.USERCATALOG: not a valid DSN pattern in this environment (qualifier length limit); the actual created catalog from DEFCAT is DEMO.VSAM.CATALOG, and it is present.

3. VIEWCOMP: JOB00177, CC 0008, spool retrieved successfully.

Final result: PASS (customer account workflow completed end to end).

Use Cases



Day-One Developer Onboarding

New hire asks 'What do I have access to?' and the AI walks them through their datasets, programs, and JCL — no green screen training needed.



Overnight Job Failure Triage

Operations asks 'What failed last night?' — AI finds the ABEND, pulls spool, explains the error, and points to the source line.



Legacy Code Understanding

Developer asks 'What does the payroll program do?' — AI reads the COBOL, explains business logic, traces copybooks and data flow.



Compliance & Audit Support

Auditor asks 'Show me all jobs that touched customer data this week' — AI queries, filters, and summarizes with full traceability.

What's Next

Open Source

The Zowe MCP Server is an Open Mainframe Project contribution — available on GitHub today.

Live z/OS Integration

Moving from mock mode to production Zowe CLI connections with full z/OSMF support.

Expanded Tool Coverage

DB2 queries, CICS transactions, IMS interactions — bringing more of z/OS into the AI's reach.

Enterprise Guardrails

RACF-aware permissions, audit logging, configurable safety policies per team and environment.

Get Started

One command to install. Works with any MCP-compatible AI client.

```
npx -y @gestell/zowe-mcp
```

Or add to your MCP config:

```
{  
  "command": "npx",  
  "args": ["-y", "@gestell/zowe-mcp"]  
}
```



github.com/Gestell-AI/zowe-mcp

Your feedback is important!

Submit a session evaluation for each session you attend:

www.share.org/evaluation

