



# Catapult Your z/OS Application Development Experience From Legacy to Legendary With VS Code Agent Mode and the Model Context Protocol

February 2026

Dr. Peter Haumer ([phaumer@us.ibm.com](mailto:phaumer@us.ibm.com), <https://www.linkedin.com/in/haumer/>)

Development Lead IBM Z Developer Experience



## Disclaimer

- IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.
- The numbers are a result of adopting IBM tools and industry research data with some client verified assumptions. No warranties are made with respect to the actual improvement your organization may achieve.

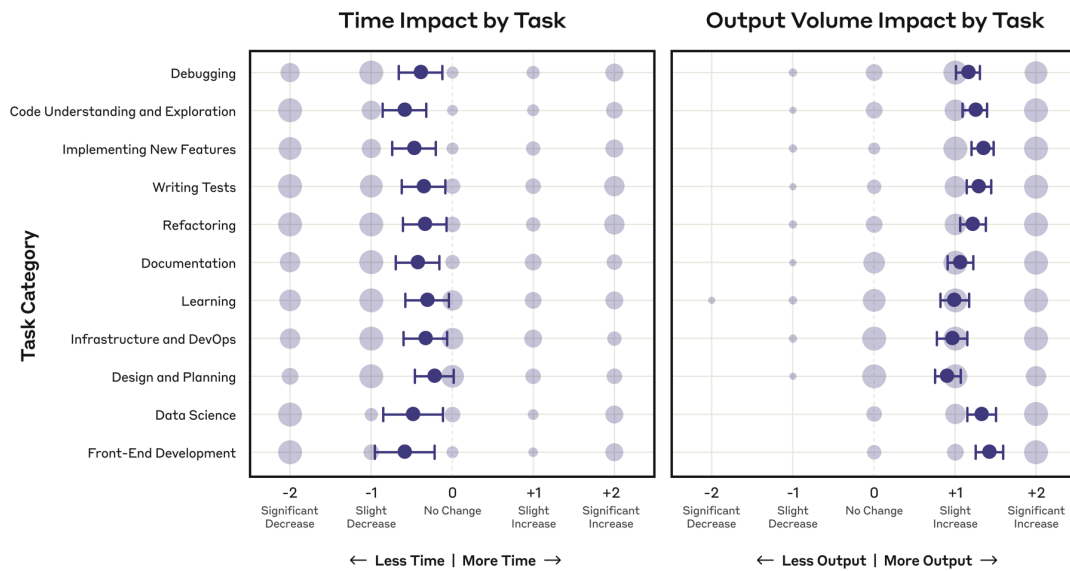
## Discussion topics

1. AI-supported developer experience
2. Model Context Protocol (MCP)
3. Integrating MCP with developer tools
4. Making AI and MCP central to the user experience: Project Bob
5. Demos

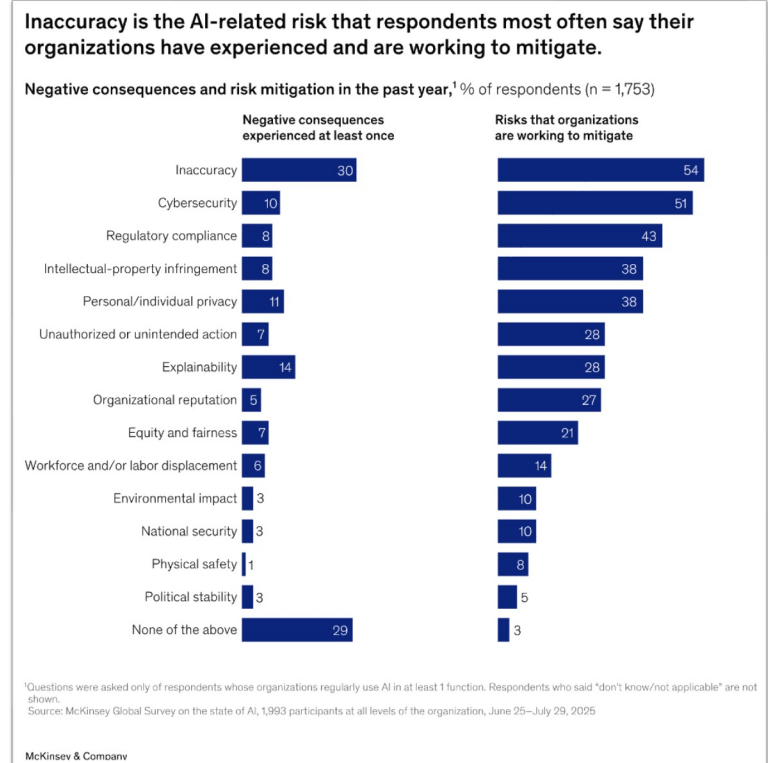
# AI-supported developer experience

# AI adoption in the software industry continues despite risks

- Most software development organizations realize the undeniable benefits
- Risk have been accepted, but awareness and education are key



<https://www.anthropic.com/research/how-ai-is-transforming-work-at-anthropic>



<https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai>

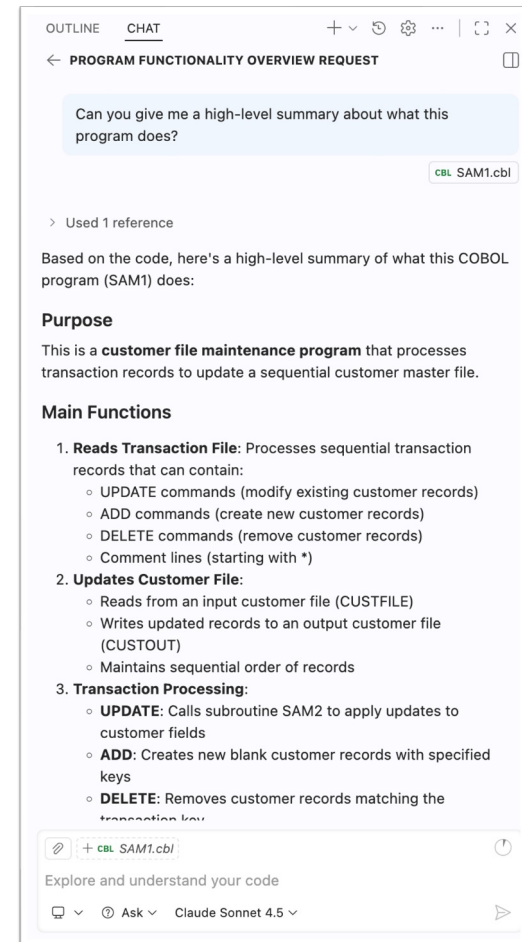


## Some core AI developer experience paradigms

- **Chat:** interactive dialogues for answers, document generation, coding requests, data source integrations and more
- **Agentic workflows:** complex multi-steps plans and task execution
- **Vibe coding:** natural language to code inside the editor
- **Inline annotations,** suggestions, code completion
- **In-place context menus,** e.g., Problem view and Code Actions integrations
- Dedicated **code review panels:** organizing all issues found with actions to fix
- AI power **debugging and runtime analysis**
- **Diff views:** side-by-side presentation of suggested code changes for approval
- **Smart search:** describe the code you are looking for
- **Code visualization:** generated control flow and data structure diagrams
- **Terminal integration:** code completion, command suggestion based on context
- **Git integration:** summarize changes for commit, create PRs based on team template

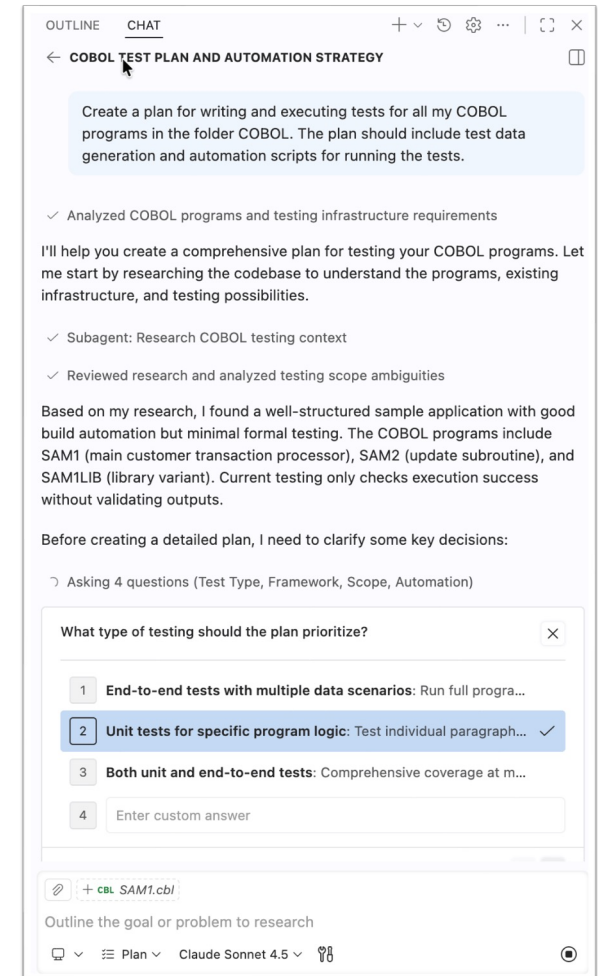
## Some core AI developer experience paradigms: Chat

- Natural language interactions with your LLM.
- When integrated in your editor, ask questions about your code, generate code to insert, analyze and fix bugs, etc.
- Context can be open program files, issues in your Problems view, other resources in your workspace, resources on the internet, etc.
- Sessions setting the context of your current interaction allowing you to refer to previous answers.
- Chat support modes that can enable support for different Agents.



## Some core AI developer experience paradigms: Agents

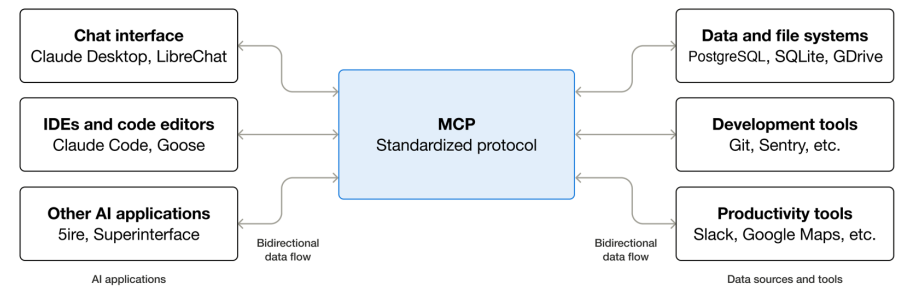
- Agents can handle complex tasks that can go beyond LLM interactions.
- Agents can be distributed across your entire organization, and their interactions can be orchestrated.
- This presentation will focus on local Chat agents that run in your editor.
- **Modes** to customize the experience and set scope, for example
  - Ask: Google-like queries and request: suggest three alternatives for how to write this algorithm.
  - Plan: Perform complex analysis of a problem and plan a multi-step solution collecting user feedback to refine the plan.
- **Skills** to define the scope of conversions and LLM prompts to use by defining the persona the agent should assume, essential background information for the complex tasks to be performed.
- **Prompt files** to provide prepared/custom instructions for perform complex task that can easily be executed from Chat, e.g. `/test` to generate unit tests for the selected code using your team's rules and style.
- **Model Context Protocol** to provide the agent with access to special resources, tool capabilities, and current data that an LLM would be able to access otherwise.



# Model Context Protocol

## Model Context Protocol (MCP)

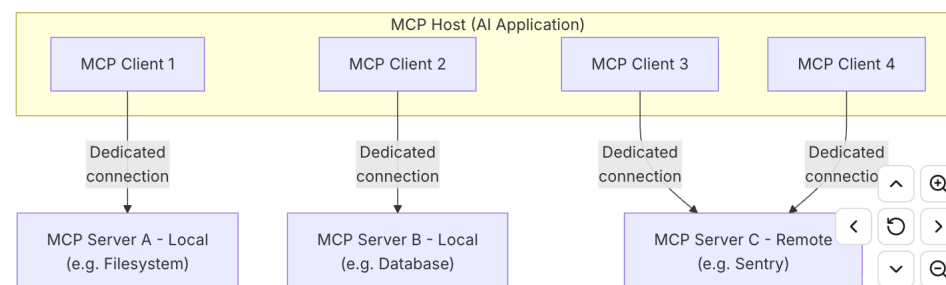
- Allows LLMs to connect to live data sources
- Referred to as USB-C for AI: allowing you to plug in many different data sources
- Created by Anthropic, but evolved into accepted standard by most vendors
- Open-source SDKs for many languages available
- Very large library of MCP implementations available



<https://modelcontextprotocol.io/docs/getting-started/intro>

## MCP architecture

- Three architectural components:
  - **MCP Host:** AI application that manages clients and server connections, such as the Chat window in your IDE
  - **MCP Client:** Performs interactions with specific MCP server
  - **MCP Server:** Program and server that can access and provide the requested data. Offers services in the forms of **tools**, **resources**, **prompts**, and other features
- Host and Client typically provided by frameworks such as VS Code
- Servers are the special tools that make the LLM smarter by providing real-time data, such as
  - Git repository data
  - z/OS resources
  - UI test framework
  - Docker containers
  - Jira
  - Slack



<https://modelcontextprotocol.io/docs/learn/architecture#concepts-of-mcp>

## MCP: the REST API for the Age of AI (1/2)

An application developer writing code using a REST API to use an external service needs to follow the exact protocol

- needs to know what REST API to call
- needs to know the exact type, format and content of the input parameters
- needs to know to format of the response and parse it exactly as it is provided

REST API are defined by a formal machine-readable specification.

**Authentication** ▾

**JES Jobs** ▾

**MVS Datasets** ▸

**POST** /datasets Creates a new partitioned dataset or a sequential dataset provided with dataset properties ▾ 🔒

**DELETE** /datasets/{dsn} Deletes a dataset or a dataset member provided the dataset/member name ▾ 🔒

**GET** /datasets/{dsn}/content Gets content of an existing dataset member or sequential dataset ▸ 🔒

Gets content of an existing dataset member or sequential dataset.

**Parameters** Try it out

Name	Description
<b>dsn</b> * <small>required</small> string <small>(path)</small>	Dataset name. Ex. TST.ACCESS.WRITE.SE or TST.PO(MEMBER) <input type="text" value="dsn"/>
Scan boolean <small>(header)</small>	Return the results of a source scan in the resulting JSON Available values : true, false <input type="text" value="--"/>
E-Tag boolean <small>(header)</small>	Returns the checksum for the file Available values : true, false <input type="text" value="--"/>
Record-Range string <small>(header)</small>	The range of records to be retrieved from the dataset file. Format SSS-EEE or SSS,NNN, example: 1000-1050 or 1000,51. Cannot be specified with header 'Etag' or 'Scan' <input type="text" value="Record-Range"/>
Host-Encoding string <small>(header)</small>	Optional host encoding parameter <input type="text" value="Host-Encoding"/>
Local-Encoding string <small>(header)</small>	Optional local encoding parameter <input type="text" value="Local-Encoding"/>

**Responses**

Code	Description	Links
200	Successful response	No links

## MCP: the REST API for the Age of AI (2/2)

MCP servers are accessed by the AI agent based on your interactions using natural language.

- Agents uses your NL Chat prompt to understand your intentions.
- LLMs are great understanding NL. So, the agent uses MCP to read the NL descriptions of the tools and resources provided by the servers and matches based on that.
- LLM provides the required inputs based on contextual information it has, e.g. previous interactions.
- No strict data types and format required.
- LLMs can work with the services results in many formats.
- Humans describe what they want and LLM, Agent, MCP figure which services are relevant, how to call them, and how to deal with the results.

The screenshot displays the MCP Inspector v0.19.0 interface. On the left, the 'Transport Type' is set to 'Streamable HTTP' and the 'URL' is 'http://localhost:3005/mcp'. The 'Connection Type' is 'Via Proxy', and the status is 'Connected'. The main panel shows a list of tools: 'zowe-get-dataset-content', 'zowe-upload-dataset', 'zowe-issue-tso-command', and 'zowe-allocate-like-dataset'. The 'zowe-get-dataset-content' tool is selected, showing its description and a 'Run Tool' button. The 'Tool Result' is 'Success', displaying a sample of dataset content with copyright information. A 'History' panel at the bottom shows a list of tool calls: '1. initialize', '2. tools/list', '3. tools/call', and '4. tools/call'.

## Example: LLM and agent deciding which tools to use

<https://modelcontextprotocol.info/docs/tutorials/writing-effective-tools/>

Agent does not find MCP tool and runs Zowe CLI command based on its LLM training

OUTLINE IBM BOB

Show me the list of files in the data set PHAUMER.SAMPLE.COBOL 6k / 200k 0.03

I'll help you list the files in the MVS dataset PHAUMER.SAMPLE.COBOL. To do this, I need to use a command that can interact with the mainframe dataset.

Run: ●

```
zowe files list ds "PHAUMER.SAMPLE.COBOL" --attributes
```

Auto-approved commands

Now I need to list the members (files) within the dataset. Let me use the appropriate command to list all members:

Run: ●

```
zowe files list am "PHAUMER.SAMPLE.COBOL"
```

```
SAM1
SAM2
```

Auto-approved commands

**Task Completed**

The dataset PHAUMER.SAMPLE.COBOL contains 2 members:

- SAM1
- SAM2

Dataset attributes:

- Organization: PO-E (Partitioned Organization Extended)
- Record format: FB (Fixed Block)
- Logical record length: 80
- Block size: 32720
- Volume: USRVS1

IBM Software © 2026 IBM Corporation

Agent finds MCP tool that matches the NL intent. Matches file to member in the context of data set.

OUTLINE IBM BOB

Show me the list of files in the data set PHAUMER.SAMPLE.COBOL 27k / 200k 0.11

I'll help you list the files in the data set PHAUMER.SAMPLE.COBOL using the Zowe MCP tool.

I want to use a tool on the zopeneditor-sample MCP server:

**zopeneditor-sample** ● Completed

**zowe-list-members**  
Get all of the data set members for a given data set. Before execution, if no data set is specified ask the user to enter a data set name providing a text prompt.

```
"{\n  \"dsn\": \"PHAUMER.SAMPLE.COBOL\\n\"}
```

**Task Completed**

The data set PHAUMER.SAMPLE.COBOL contains 2 members:

- SAM1
- SAM2

These are the COBOL program members stored in the partitioned data set.

Agent finds tool to access local file system and select that because "folder" is mentioned.

OUTLINE IBM BOB

Show me the list of files in the folder COBOL 27k / 200k 0.03

I'll list the files in the COBOL folder for you.

I want to view the top level files in this directory:

COBOL

**Task Completed**

The COBOL folder contains 3 files:

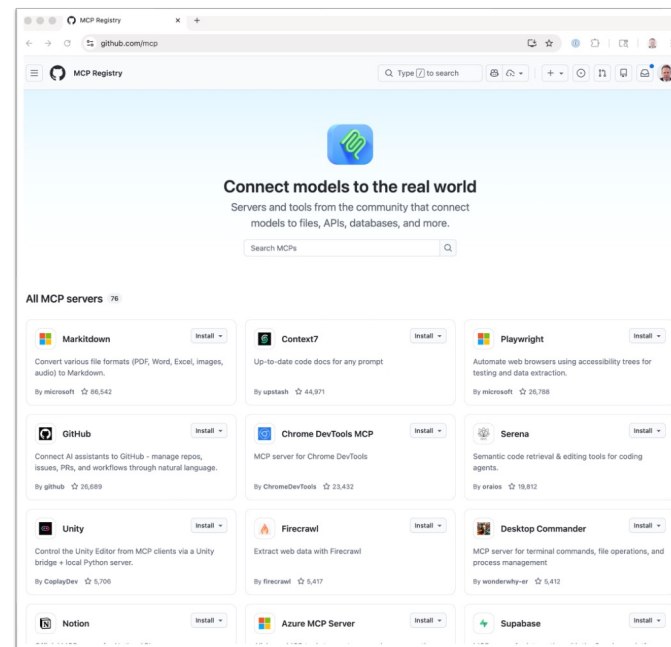
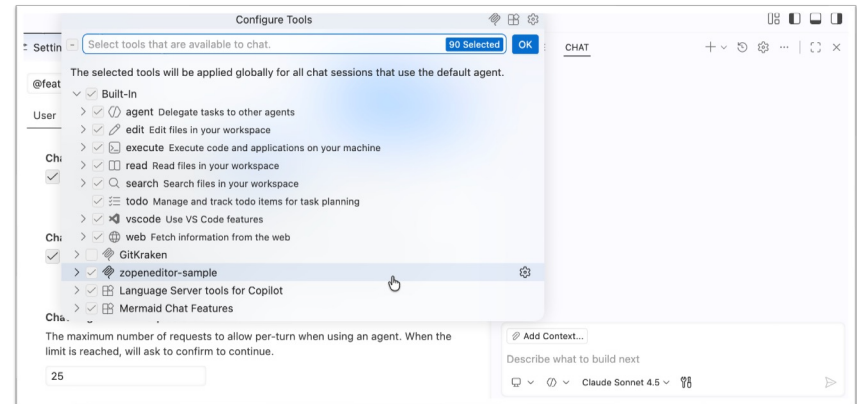
- SAM1.cb1
- SAM1LIB.cb1
- SAM2.cb1

Ambiguity can be factor and requires clear language by the MCP tool developer and the user writing their prompts. MCP tools are deterministic. LLMs are not deterministic.



## VS Code Chat: MCP out of the box

- VS Chat branded GitHub Chat available out of the box.
- Select from a large collection of LLMs billed via GitHub.
- Various ways for installing and registering MCP servers: gallery integrated into Extensions view or separate Web site.
- Select individual tools and resources in drop-down.
- Caution: Local MCP servers can run arbitrary code and access resource on your machine. Only add servers from trusted sources and review the publisher and server configuration before starting it.



## Many alternative MCP agents available

- MCP is recognized as a standardized protocol
- MCP tools can run in many different environments
- Popular VS Code extensions:  
CodeGPT, Sourcegraph Cody, Amazon Q
- VS Code forks, reinventing AI-centric developer experience:  
Cursor, Codex, Windsurf, Antigravity, IBM Bob
- Use the same MCP servers in many of these alternatives.

The screenshot displays the VS Code interface with the MCP settings panel open on the left and the IBM Bob chat interface on the right.

**MCP Settings Panel:**

- General:** MCP (Enabled)
- Chat:** zopeneditor-sample (Enabled)
- Editor:** Scope: Project
- Tools:** Tools this MCP is using (Search tools: All)
- Bob version:** 0.0.14
- Bob Docs:** Bob Docs

**MCP Tools List:**

Tool Name	Description	Always allow
zowe-list-datasets	List the data sets for a given data set pattern. Before execution, if no data set pattern is specified ask the user to enter a data set pattern providing a text prompt.	Enabled
zowe-list-members	Get all of the data set members for a given data set. Before execution, if no data set is specified ask the user to enter a data set name providing a text prompt.	Enabled
zowe-get-dataset-content	Get the contents of a specified sequential data set or partitioned data set member. Before execution, if no name is specified ask the user to enter a data set name providing a text prompt.	Enabled
zowe-upload-dataset	Upload the specified local file to a Sequential or Partitioned data set. Before execution, if no local file path or data set name is specified ask the user to enter a local file path or data set name ...	Enabled
zowe-issue-tso-command	Issue a TSO (Time Sharing Option) command. Before execution, if no command is specified ask the user to enter a TSO command providing a text prompt.	Enabled
zowe-allocate-like-dataset	Allocate a new data set similar to an existing one. Before execution, if no data set names are specified ask the user to enter the new data set name and the template data set name providing a text prompt.	Enabled
zowe-search-dataset	Perform Search method for a string or regex pattern search within PDS datasets and members.	Enabled

**IBM Bob Chat Interface:**

- Avatar:** IBM Bob (A blue robot wearing a hard hat and holding a code block).
- Greeting:** Hi, I'm IBM Bob. Ask me questions or let me code for you.
- Workflows:** Java modernization (Active)
- Recent Tasks:**
  - Run a zcodescan on all the programs inside the COBOL folder and create a nice graphical quality report with pie and bar charts. (3 hours ago - 0.77)
  - Run a zcodescan on the currently open program and list issues by severity. (3 hours ago - 0.36)
  - Run a ZCodeScan of the currently open program and summarize the results. (3 hours ago - 0.37)
- Input:** What would you like me to help with? (@ to add context, / for commands, hold shift to drag in files/images)
- Advanced:** Enabled

# Building your own MCP server

- SDKs available for several popular languages.
- Many other open-source projects available to provide additional abstraction layers.
- Two transport protocols:
  - stdio (e.g., for standalone programs and script)
  - http (e.g., running the MCP server as a local web server)
- A Typescript-based server could run an Express web server to host the MCP server.
- MCP tool is a wrapper for functional capabilities and services.
- MCP resource is access to data such as log files, database queries, etc.
- Sample for registering an MCP tool providing NL description for tool and input/output schema. Output can be test or json.

```
server.registerTool(  
  'zowe-get-dataset-content',  
  {  
    title: 'Get contents of data set member or sequential data set',  
    description:  
      'Get the contents of a specified sequential data set or partitioned data set member. Before execution, if no name is specified ask the user to enter a data set name providing a text prompt.',  
    inputSchema: {  
      dsn: z  
        .string()  
        .describe('A qualified Sequential data set or PDS member name (e.g. IBUSER.JCL.SRC(MEM))')  
    },  
  },  
  async ({ dsn }) => {
```

<https://modelcontextprotocol.io/docs/sdk>

Model Context Protocol

Documentation Specification Registry Community

Get started

What is MCP?

About MCP

Architecture

Servers

Clients

Versioning

Develop with MCP

Connect to local MCP servers

Connect to remote MCP Servers

Build an MCP server

Build an MCP client

**SDKs**

Security

Developer tools

MCP Inspector

Extensions

MCP Apps

Search...

On this page

Available SDKs

Getting Started

Next Steps

Copy page

## Develop with MCP SDKs

Official SDKs for building with Model Context Protocol

Build MCP servers and clients using our official SDKs. SDKs are classified into tiers based on feature completeness, protocol support, and maintenance commitment. Learn more about [SDK tiers](#).

### Available SDKs

- TypeScript
- Python
- Go
- Kotlin
- Swift
- Java
- C#
- Ruby
- Rust
- PHP

### Getting Started

Each SDK provides the same functionality but follows the idioms and best practices of its language. All SDKs support:

- Creating MCP servers that expose tools, resources, and prompts
- Building MCP clients that can connect to any MCP server
- Local and remote transport protocols
- Protocol compliance with type safety

Visit the SDK page for your chosen language to find installation instructions, documentation, and examples.

**IDz on VS Code with Z Open Editor:**

**MCP tools for  
z/OS Enterprise development**



# IBM Z Open Editor

A free modern editing experience for IBM Z Enterprise languages



**Z** **COBOL** **JCL**  
**HLASM** **PL/I**  
**REXX**

VS Code style editing



- Search-Replace across files
- Searchable Outline view
- Code snippets

Integrates with Zowe



Explorers for z/OS MVS, USS, and JES

Preview and open remote Copybooks

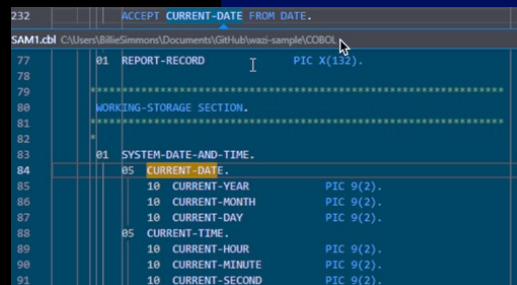
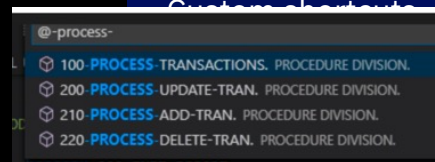
Added value features such as Search & sortable z/OS Resources table

## Advanced Language Servers

COBOL 6.5  
 PL/I 6.2  
 HLASM z/OS 3.1  
 REXX  
 JCL

Edit embedded code for  
 CICS 6.3, IMS 15.1.0 and  
 SQL DB2 for z/OS 12.1

- Real-time syntax checking while you type
- Sortable Outline view
- Code and Variables completion
- Declaration code-lenses and go-to-definition
- Find all references for variables/paragraphs
- Refactoring such as Rename Symbol
- Preprocessor support



Integrates with Git SCM



Runs in the Cloud



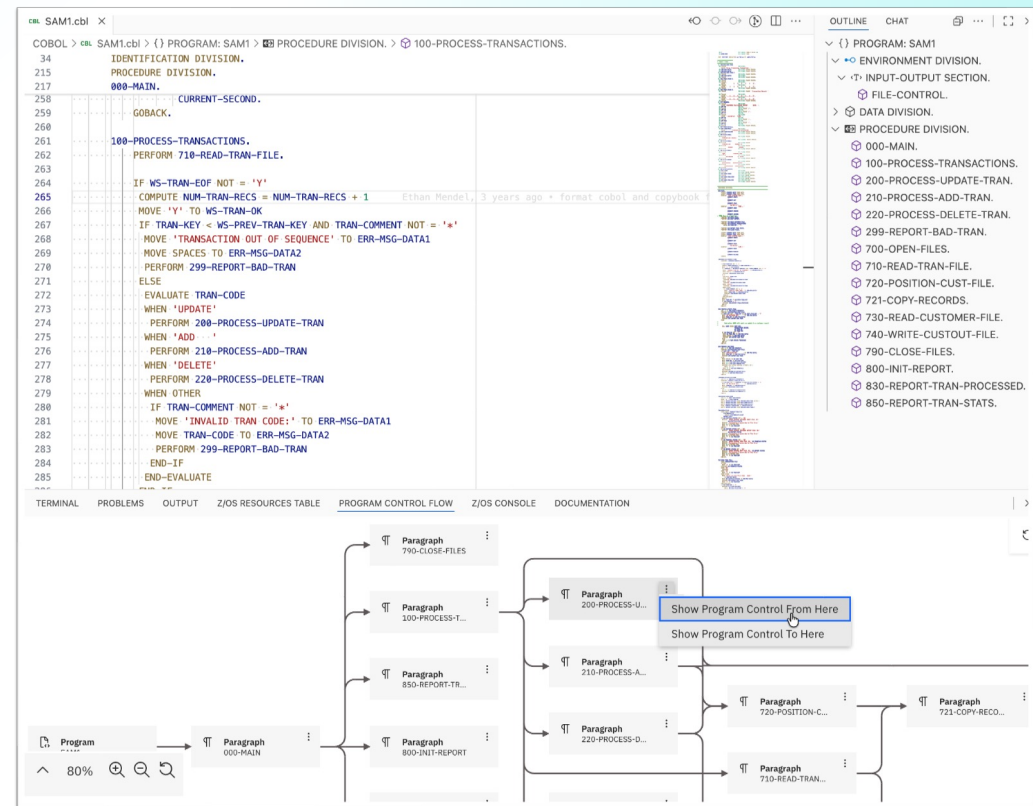
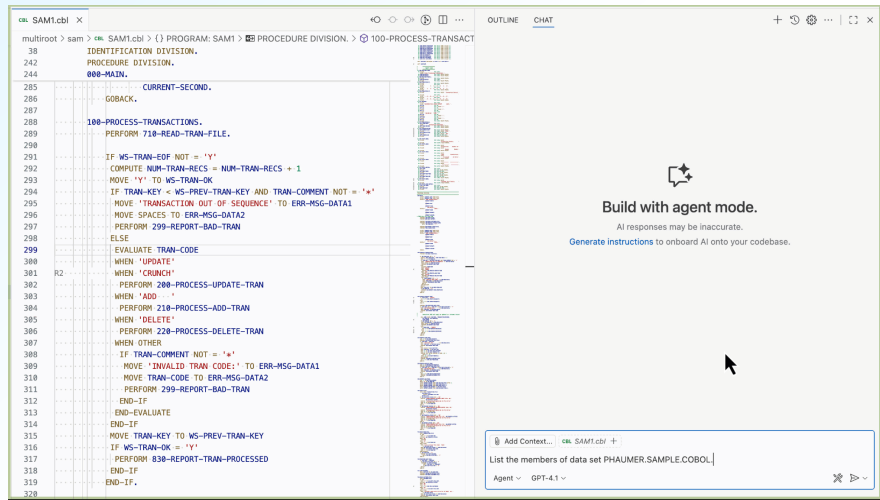
Integrates with AI agents via built-in MCP server

# Release Highlights IDz on VS Code v6

Major release update switching to VS Code 1.102 and Java 21.

Key capabilities added

- Agent Mode: MCP servers for AI chat
- Program Control Flow Graph Brower
- Data Flow Graph Brower
- CICS 6.3 support for COBOL and PL/I
- Expand Source Code for PL/I incl. macros



## Layered solution components: Zowe as the basis for Z Open Editor and Debug

### IBM Developer for z/OS on VS Code (IDz EE and ADFz)

Commercial Enterprise editor on VS Code

- Support for Z Open Editor
- Debug capabilities with Z Open Debug
- User Build integration with DBB
- Other advanced capabilities for enterprises
- Model Context Protocol servers for AI Chat integration

### IBM Z Open Editor (free)

Enterprise level extension for z/OS development, free download from the marketplace

- First class editing support for z/OS application development for COBOL, PL/I, HLASM, REXX, and JCL.
- Use of common z/OS servers (RSE) across VS Code and Eclipse
- Flexible encoding conversions

### Zowe Explorer (open-source)

Visual studio code extension for interacting with the mainframe. Built as graphical UI on top of Zowe CLI code. Allows a user to interact with data sets, UNIX files, and jobs that are stored on z/OS.

### Zowe CLI (open-source)

Nodejs-based command line interface to interact with the mainframe. Allows a user to interact with data sets, UNIX files, jobs, TSO, z/OS Console and more. Uses z/OSMF REST API to interact with mainframe. Many plugins available such as CICS, DB2, IMS, MQ, FTP.

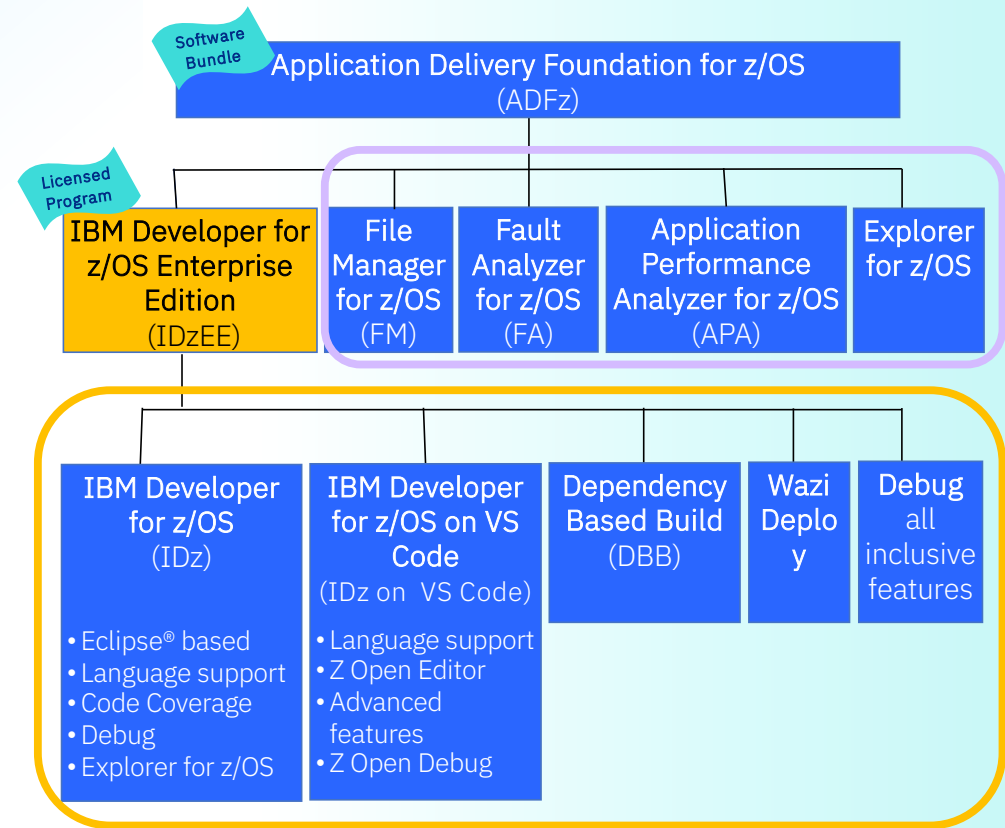
# IBM Developer for z/OS Enterprise Edition (IDzEE)

## What is IDzEE

- A **licensed program** of integrated development tools
- One enterprise license
- Choice of Integrated Development Environments
- Enables teams to adopt agile practices
- Enables a continuous integration and continuous deployment (CI/CD) pipeline
- Supports legacy and modern source code managers (SCM)
- Accelerate application development process
- Automates deployments
- .... and more

## ADFz adds

Tools to aid in accessing mainframe data and assist in problem determination by providing abend analysis and interactive reports on application resource consumption



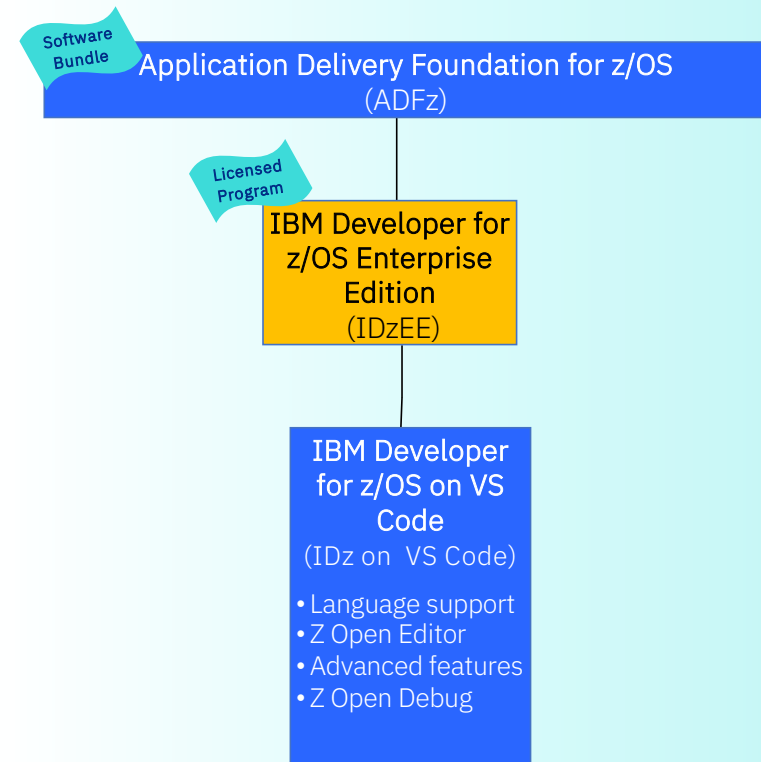
# What are the IDz on VS Code Advanced Capabilities?

Z Open Editor 4.0 (3/24) started adding Advanced Capabilities that need to be unlocked with IDzEE or ADFz activation kits after 90 days.

Incrementally added new and refined capabilities since then every month!

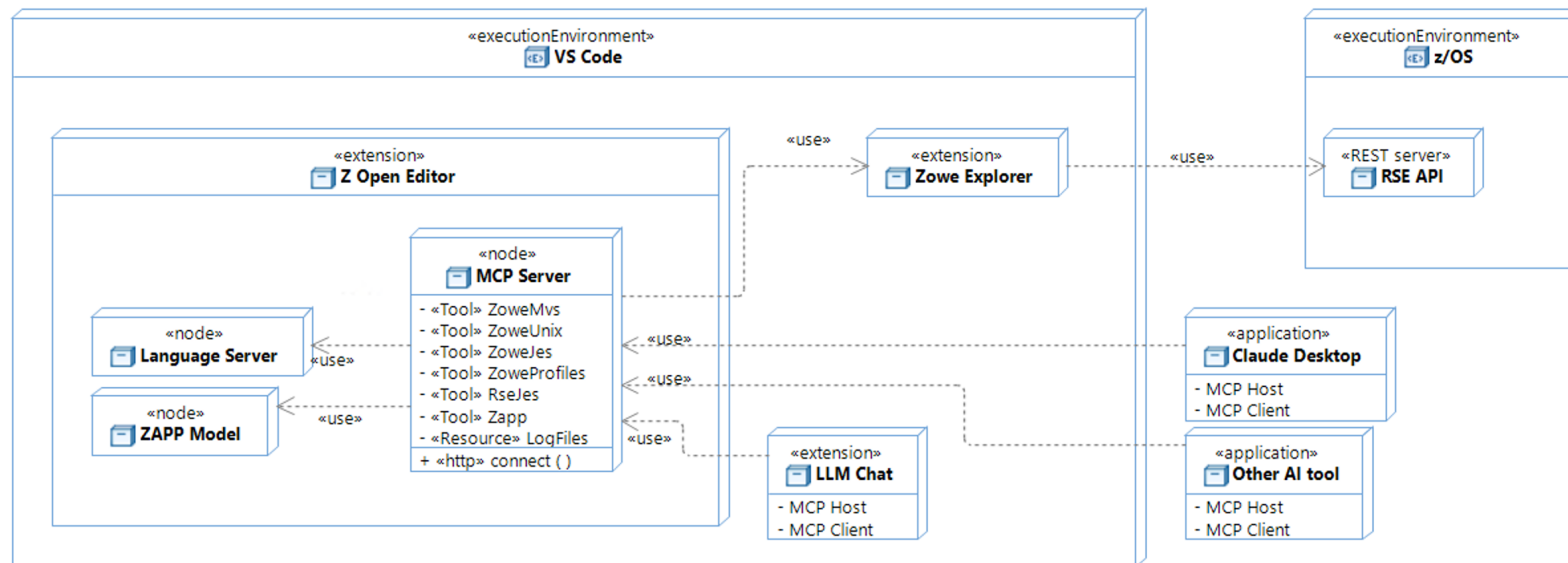
List of capabilities available with 6.4.0

- z/OS Resources Table
- Data Elements View
- Program Control Flow Graph Brower
- Preprocessor support for COBOL and PL/I
- Language detection for data set members
- Problem View for DBB User Build
- HLASM custom macros support
- IBM ZCodeScan
- 3270 Remote Connection Emulator
- Agent Mode: MCP servers for AI chat



## Z Open Editor's Model Context Protocol server

- Z Open Editor implements the MCP tools and resources for many of its capabilities.
- MCP server runs as local web server inside the extension.
- Every VS Code window instance will run its own server with the specific local context.



# IDz on VS Code MCP tools

Almost 50 tools available in these areas

- Zowe Profiles management tools
- Zowe MVS tools
- Zowe UNIX System Services tools
- Zowe JES tools
- IBM RSE API tools
- Z Open Editor tools
- Z Open Editor log file resource
- ZAPP tools
- User build tools
- ZCodeScan tools
- Wazi Deploy tools

**IBM Z Open Editor MCP Tools**

These are all the Model Context Protocol (MCP) capabilities available in Z Open Editor. See the section [Agent Mode](#) for details.

### Zowe Profile Tools

You can ask about your Zowe™ Client profiles to see what is available, what is in use by Z Open Editor, and to obtain or retire JSON Web Token (JWT) authentication.

Tool Name	Tool Description
<code>zowe-get-default-profile &lt;type&gt;</code>	Retrieve the default Zowe profile of a given type as configured in the Zowe team configuration files, defaults to type RSE.
<code>zopeneditor-get-default-zowe-profile</code>	Retrieve IBM Z Open Editor's default Zowe profile as configured in the <code>zopeneditor.zowe</code> user settings.
<code>zowe-profile-connection-check</code>	Retrieve the status of the connection to the Rest API server.
<code>zowe-list-profiles &lt;type&gt;</code>	List Zowe profiles configured in the Zowe team configuration file by type, defaults to all profiles.
<code>zowe-profile-login</code>	Log in with the editor's set profile to obtain a JWT.
<code>zowe-profile-logout</code>	Log out from the server with the editor's set profile and remove the JWT from the team configuration file.
<code>zopeneditor-change-default-zowe-profile</code>	Change the IBM Z Open Editor default Zowe profile.

### Zowe MVS Tools

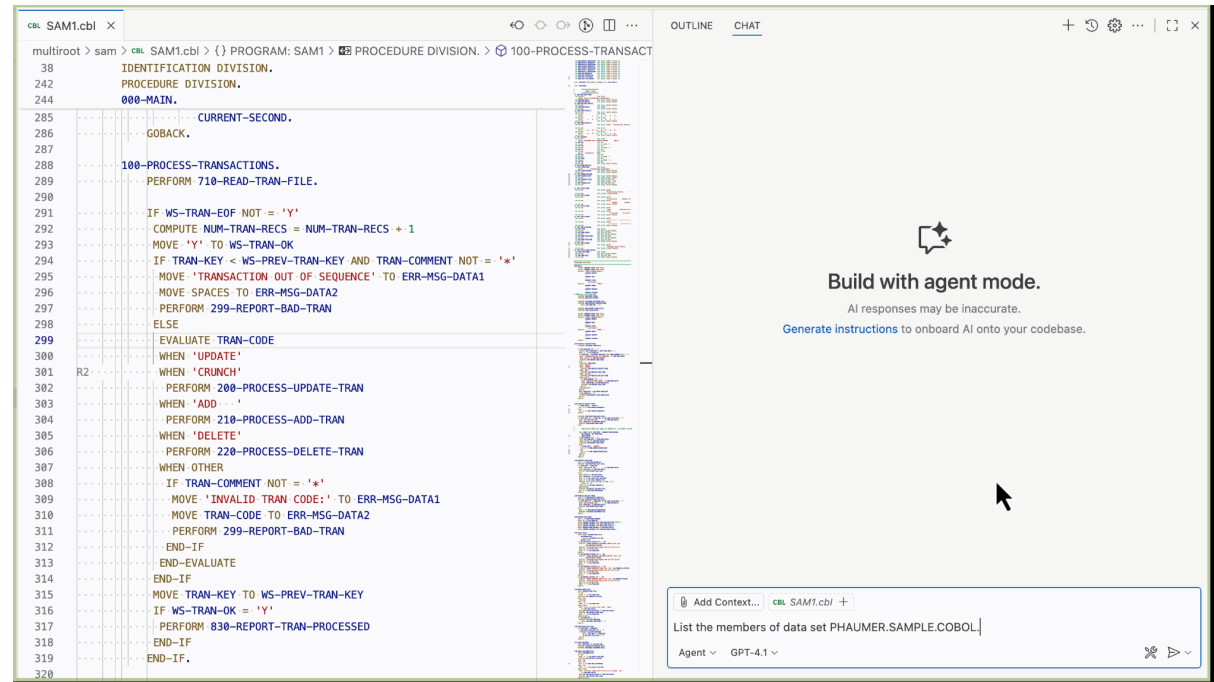
You can manage data sets and members on z/OS® and run TSO commands.

Tool Name	Tool Description
<code>zowe-list-datasets &lt;filter&gt;</code>	Retrieve data sets that match the specified filter pattern.
<code>zowe-list-members &lt;dataSetName&gt;</code>	List members of the specified data set.
<code>zowe-get-dataset-content &lt;name&gt;</code>	Download the specified data set member.
<code>zowe-allocate-like-dataset &lt;dataSetName&gt; &lt;existingDataSet&gt;</code>	Allocate a new specified data set similar to an existing one.
<code>zowe-issue-tso-command &lt;command&gt;</code>	Execute the specified TSO command.

## IDz on VS Code Agent Mode

Example use cases:

- List the members of the data set USER.SAMPLE.COBOLE that start with the prefix SAM.
- What are the jobs of user IBMUSER grouped by status?
- Submit the JCL in the currently open editor as a job.
- Using the zopeneditor.log resource and the ZAPP property groups, analyze why copybook include files could not be found for the currently open program?
- Can you explain the ZCodeScan issues listed in the Problems view for COBOL developers and how to fix them?
- Who made the last few changes on this program looking at the Git history?



**IBM Bob**

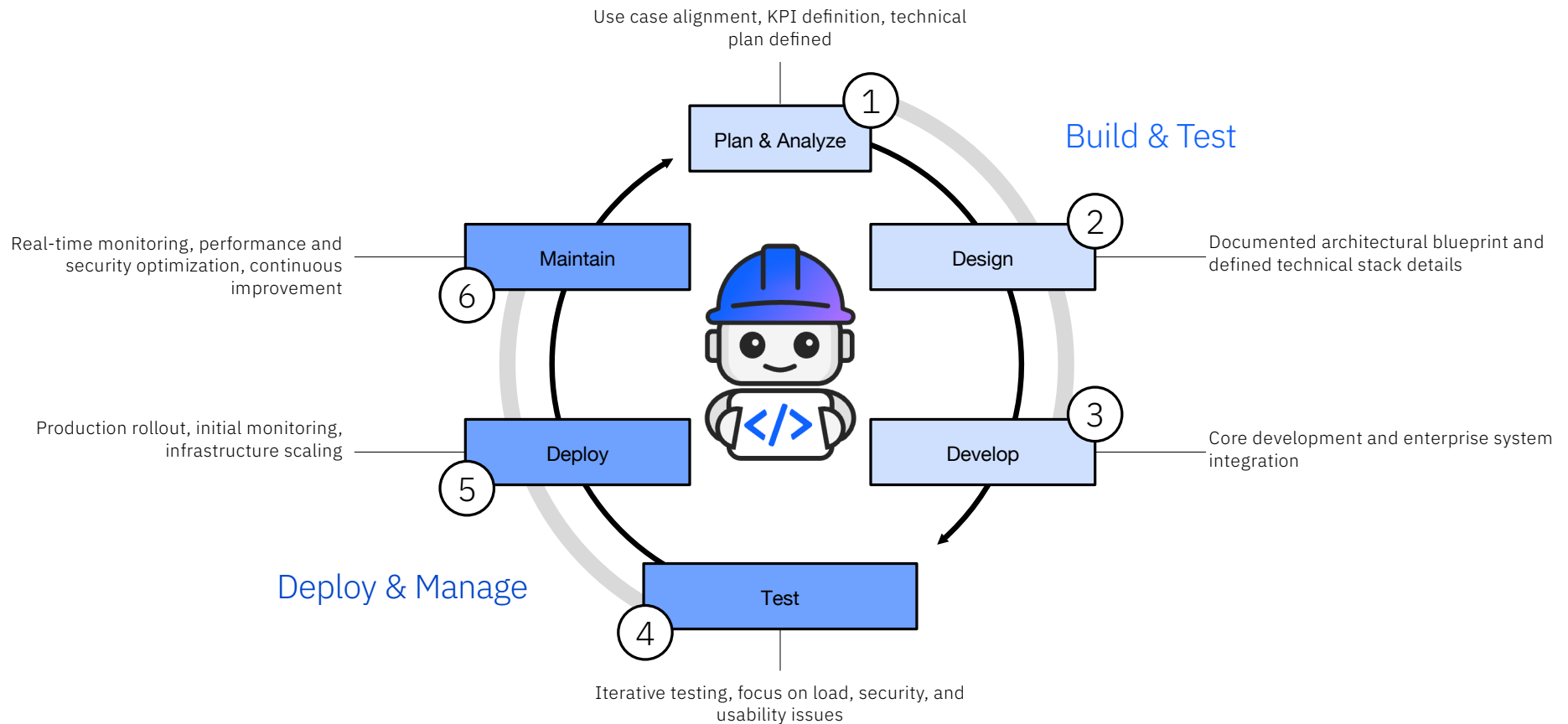
**Your AI-powered coding partner**

**and**

**Z Open Editor's Best Friend**

# Introducing IBM Bob, the AI partner across the SDLC

Delivering AI across the SDLC requires a different architecture. Bob is built for that.



# Developers need AI across the entire SDLC

Enterprises are entering a new phase of AI-driven development where tools don't just suggest or autocomplete code: they operate and coordinate across systems

## Execute multi-step workflows, not just suggestions

Outcomes require orchestration, retries, and validation, not one-off prompts.

## Operate across the full SDLC within existing environments

Value is unlocked when AI operates inside existing delivery pipelines, not alongside them.

## Understand entire repositories, not isolated files

Safe orchestration requires repository-level context, not just single-file reasoning.

## Embed governance, security and human oversight by design

Enterprise AI must enforce policy, security and accountability as work happens.

**95%** of IBM's AI-assisted development involved **complex or multi-step workflows**, not isolated code completions\*

# Bob for the entire Software Development Lifecycle





## Instant understanding and action

---

- Interprets your intent with precision
- Analyzes code intelligently
- Creates clear, actionable To-Do lists for immediate use

## Versatile development modes

---

-  Plan Mode:  
Plan and design before implementation
-  Code Mode:  
Implement features and fix bugs efficiently
-  Ask Mode:  
Get answers to technical questions
-  Advanced Mode:  
Use advanced features, such as MCP

## Enterprise-grade capabilities

---

- Automated deployment playbooks: Streamline your release process
- Comprehensive code reviews: Maintain quality standards
- Intelligent code refactoring: Optimize performance and readability

## Why teams choose Bob

---

- Accelerates development cycles
- Reduces technical debt
- Empowers developers of all experience levels
- Ensures consistent code quality across projects

# Bob in our workflow

When AI fits the workflow, adoption and results follow



Adoption at scale...

~30k (and counting) IBM devs using Bob

*Rapid enterprise-wide adoption*

~95% activation after onboarding

*Near-universal activation rate*

~65% daily usage

*Sustained daily active usage*



...with measurable developer impact

~45% average improvement

*Dramatic improvements to productivity*

90%+ reduction in effort for certain tasks

*Repetitive work largely automated away*

Developers moved up the value stack

*Measurable move from manual coding to task orchestration*

## Smart Scribble

Literate coding lets you write code with **AI assistance** right inside the editor.

Just type **plain-language instructions** exactly where the code belongs.

I generate the implementation **in-context** and display a diff of the changes.

## Code Actions

Inject **AI-powered assistance** directly into your workflow, so that you stay in the zone.

Snap the selected snippet – complete with **filename and line numbers** – into the chat with a single click.

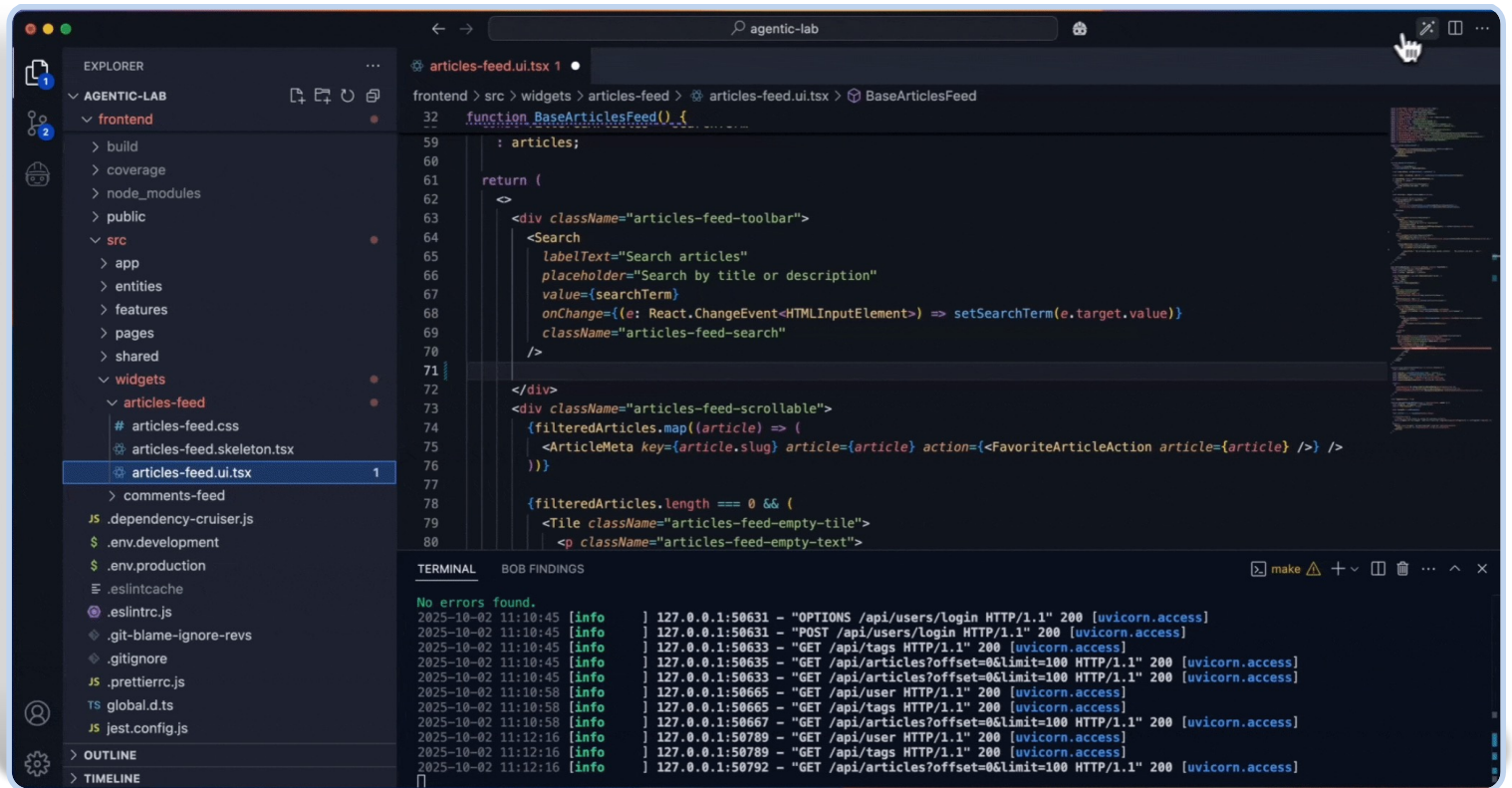
I'll dissect the code on the spot and surface **inline suggestions or improvements**.

What can I do?

# Literate Coding & Beyond

Check Out More Videos

- ▶ [Literate Coding](#)
- ▶ [Code Actions](#)

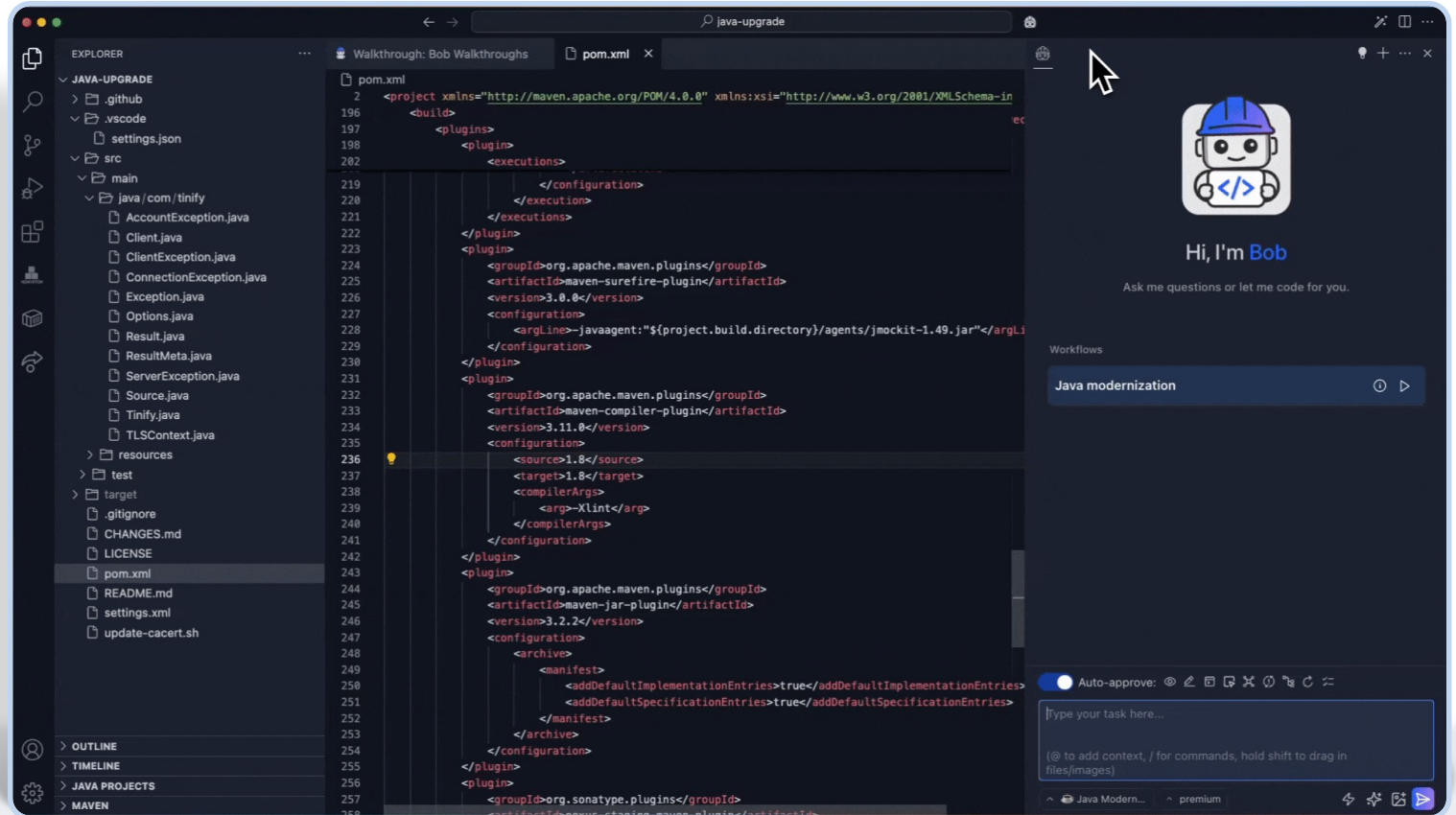


What can I do?

# Modernize Java Applications

Check Out More Videos

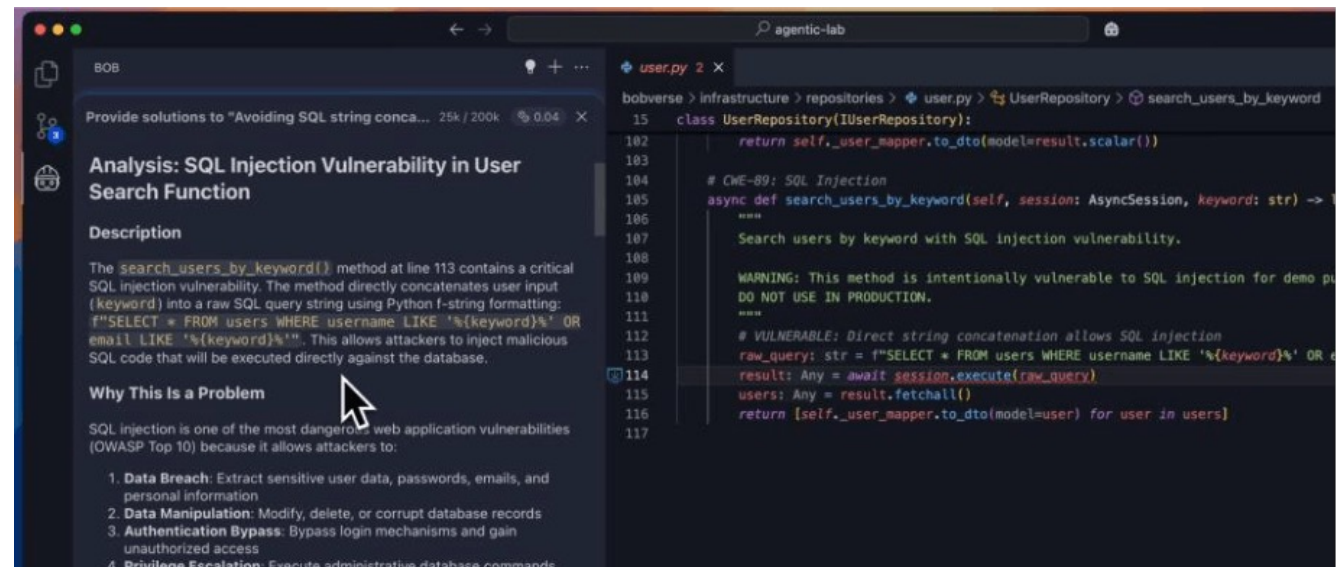
- ▶ [Java Upgrade](#)
- ▶ [Liberty Replatforming](#)
- ▶ [UI Modernization](#)



## Shift-left security: governance & security built into workflows

Bob embeds security and compliance directly into AI-driven workflows so protection happens during execution, not after deployment or incidents

By integrating guardrails, validation, and oversight with clear checkpoints and review points, Bob enables faster delivery while maintaining trust and auditability.



The screenshot shows a development environment with a code editor on the right and a security analysis tool (Bob) on the left. The code editor displays a Python class `UserRepository` with a method `search_users_by_keyword`. The analysis tool highlights a vulnerability in the `search_users_by_keyword` method at line 113, where user input is concatenated into a raw SQL query string. The analysis tool provides a detailed description of the vulnerability, a warning message, and a list of potential impacts.

```
class UserRepository(IUserRepository):
    182     return self._user_mapper.to_dto(model=result.scalar())
    183
    184     # CWE-89: SQL Injection
    185     async def search_users_by_keyword(self, session: AsyncSession, keyword: str) -> List[User]:
    186         """
    187         Search users by keyword with SQL injection vulnerability.
    188
    189         WARNING: This method is intentionally vulnerable to SQL injection for demo purposes.
    190         DO NOT USE IN PRODUCTION.
    191         """
    192         # VULNERABLE: Direct string concatenation allows SQL injection
    193         raw_query: str = f"SELECT * FROM users WHERE username LIKE '{keyword}%' OR email LIKE '{keyword}%'". This allows attackers to inject malicious SQL code that will be executed directly against the database.
    194         result: Any = await session.execute(raw_query)
    195         users: Any = result.fetchall()
    196         return [self._user_mapper.to_dto(model=user) for user in users]
    197
```

**Analysis: SQL Injection Vulnerability in User Search Function**

**Description**

The `search_users_by_keyword()` method at line 113 contains a critical SQL injection vulnerability. The method directly concatenates user input (`keyword`) into a raw SQL query string using Python f-string formatting: `f"SELECT * FROM users WHERE username LIKE '{keyword}%' OR email LIKE '{keyword}%'". This allows attackers to inject malicious SQL code that will be executed directly against the database.`

**Why This Is a Problem**

SQL injection is one of the most dangerous web application vulnerabilities (OWASP Top 10) because it allows attackers to:

1. **Data Breach:** Extract sensitive user data, passwords, emails, and personal information
2. **Data Manipulation:** Modify, delete, or corrupt database records
3. **Authentication Bypass:** Bypass login mechanisms and gain unauthorized access
4. **Privilege Escalation:** Execute administrative database commands

“Catch issues early through shift-left security checks, with governance and auditability across changes

# IBM watsonx Code Assistant for Z

Accelerated application lifecycle

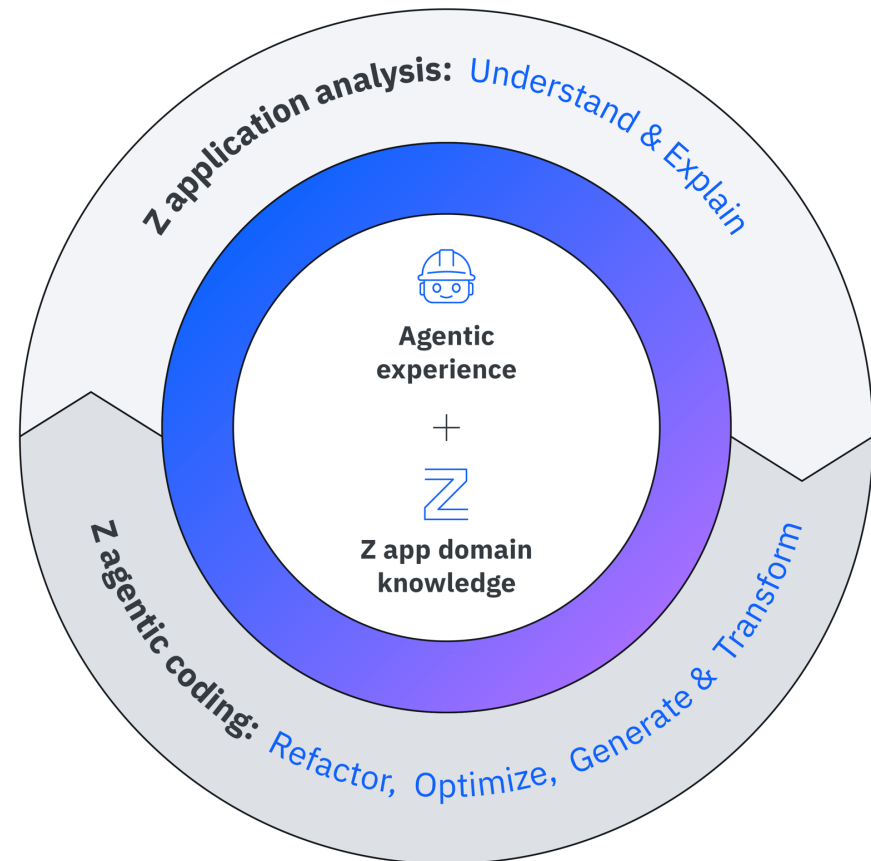
## Agentic experience

brings purpose-built AI and automated tooling for end-to-end modernization workflows, from application analysis to coding

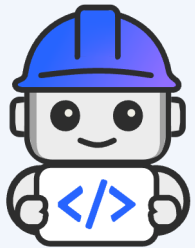
## Z application domain knowledge

infuses tooling with Z specific language and middleware context

AI-assisted mainframe application modernization



# Ask Bob



IBM Bob

## Get Bob!

[Join the waitlist](#) →



## Follow Bob on YouTube

[IBM Bob](#) →  
(<https://www.youtube.com/@ibm-bob>)

## Video Playlists

[Your agentic coding partner](#) →

[Java Modernization](#) →

[Deploy with confidence](#) →

[From observability to action](#) →

[Application Modernization on IBM i](#) →

**DEMO**

**Z Open Editor's MCP server**

**in**

**IBM Bob**

## Demo script

- Open a program and check why it could not find copybooks
- Check my Zowe configurations
- List and sort my my data sets and review data set contents that have COBOL files
- Compare a program in data set with local copy
- Run a ZCodeScan on the file, explain results, and propose fixes
- Fix highest severity issues
- Run a build to check that the program still builds
- Run the application by submitting JCL on check the spool files for errors
- Run a ZCodeScan on all my programs and create a fancy quality report

The screenshot displays the IBM Zowe IDE interface. The main editor shows a COBOL program named SAM1.cbl with the following code:

```
IDENTIFICATION DIVISION.  
PROCEDURE DIVISION.  
  
200-PROCESS-UPDATE-TRAN.  
ADD +1 TO NUM-UPDATE-REQUESTS.  
PERFORM 728-POSITION-CUST-FILE.  
IF CUST-KEY NOT = TRAN-KEY OR WS-CUST-FILE-EOF = 'Y'  
MOVE 'NO MATCHING KEY: ' TO ERR-MSG-DATA1  
MOVE TRAN-KEY TO ERR-MSG-DATA2  
PERFORM 299-REPORT-BAD-TRAN  
ELSE  
Subroutine SAM2 will apply an update to a customer record  
CALL SAM2 USING CUST-REC,  
TRANSACTION-RECORD,  
WS-TRAN-OK,  
WS-TRAN-MSG  
IF WS-TRAN-OK NOT = 'Y'  
MOVE WS-TRAN-MSG TO ERR-MSG-DATA1  
MOVE SPACES TO ERR-MSG-DATA2  
PERFORM 299-REPORT-BAD-TRAN  
ELSE  
ADD +1 TO NUM-UPDATE-PROCESSED  
END-IF.  
  
210-PROCESS-ADD-TRAN.  
ADD +1 TO NUM-ADD-REQUESTS.  
PERFORM 728-POSITION-CUST-FILE.  
IF CUST-KEY = TRAN-KEY  
MOVE 'DUPLICATE KEY: ' TO ERR-MSG-DATA1  
MOVE TRAN-KEY TO ERR-MSG-DATA2  
PERFORM 299-REPORT-BAD-TRAN  
ELSE  
MOVE SPACES TO WS-CUST-REC  
MOVE TRAN-KEY TO WS-CUST-KEY  
MOVE +0 TO WS-CUST-ACCT-BALANCE  
MOVE +0 TO WS-CUST-ORDERS-TID  
PERFORM TEST AFTER VARYING I FROM 1 BY 1  
UNTIL I > 12  
MOVE +0 TO WS-CUST-MONTH(I)  
END-PERFORM  
PERFORM 740-WRITE-CUSTOM-FILE  
ADD +1 TO NUM-ADD-PROCESSED  
END-IF.  
  
220-PROCESS-DELETE-TRAN.  
ADD +1 TO NUM-DELETE-REQUESTS.  
PERFORM 728-POSITION-CUST-FILE.  
IF CUST-KEY NOT = TRAN-KEY OR WS-CUST-FILE-EOF = 'Y'  
MOVE 'NO MATCHING KEY: ' TO ERR-MSG-DATA1  
MOVE TRAN-KEY TO ERR-MSG-DATA2  
PERFORM 299-REPORT-BAD-TRAN
```

The ZCodeScan analysis window on the right provides the following details:

- Run a ZCodeScan of the open program, explain the results, and propose fixes.**
- Problem:** Without `END-CALL`, the program may not accurately determine when the called program's execution ends, potentially causing subsequent statements to execute unintentionally.
- Current Code Pattern:**

```
CALL 'WS-SUBPROGRAM' USING WS-NUM1, WS-NUM2, WS-RESULT.
```
- Proposed Fix:**

```
CALL 'WS-SUBPROGRAM' USING WS-NUM1, WS-NUM2, WS-RESULT  
END-CALL
```
- Recommended Action Plan:**
  - Priority 1 (HIGH): Add missing `END-READ` and `END-CALL` clauses (lines 304-307, 391-393, 426-428)
  - Priority 2 (MEDIUM): Replace `ACCEPT DATE/TIME` with `CURRENT-DATE` (lines 218-219, 244-245)
  - Priority 3 (MEDIUM): Refactor nested `IF` statements to `EVALUATE` (lines 267-286, 288-290, 308-314, 413-417)
- Task Completed**
- ZCodeScan analysis complete for SAM1.cbl. Found 11 code quality issues:**
  - HIGH Severity (3 Issues):**
    - Missing `END-READ` clauses at lines 391-393 and 426-428
    - Missing `END-CALL` clause at lines 304-307
  - MEDIUM Severity (8 Issues):**
    - 4 occurrences of deprecated `ACCEPT DATE/TIME` (lines 218, 219, 244, 245) - should use `CURRENT-DATE` instead
    - 4 nested `IF` statements (lines 267-286, 288-290, 308-314, 413-417) - should use `EVALUATE` for better maintainability
- Recommended Fixes:**
  - Add `END-READ` and `END-CALL` clauses to prevent scope termination issues
  - Replace `ACCEPT DATE/TIME` with `CURRENT-DATE` for better precision and timezone support
  - Refactor nested `IF` statements to `EVALUATE` statements for improved readability and maintainability

The terminal window at the bottom shows the following output:

```
IBM BOB FINDINGS ...  
Filter (e.g. text, **/*):  
- End Clause is Required (IBM ZCodeScan) [Ln 304, Col 13]  
- End Clause is Required (IBM ZCodeScan) [Ln 391, Col 12]  
- End Clause is Required (IBM ZCodeScan) [Ln 426, Col 12]  
- Use CURRENT-DATE rather than ACCEPT DATE or ACCEPT TIME (IBM ZCodeScan) [Ln 218, Col 12]
```

## Try it...

free IBM Z Open Editor  
<https://ibm.github.io/zopeneditor-about>

and provide feedback

Try our free Red Hat  
Developer Sandbox

<https://developers.redhat.com/developer-sandbox>

## Learn more about

### Z Open Editor Docs:

<https://ibm.github.io/zopeneditor-about/>

Architecture deep-dive and demo:

<https://youtu.be/8Z0GPN7Ld2w>

Day in the life: mainframe Git workflow video series:

[https://mediacenter.ibm.com/playlist/details/1\\_0o0vq1h9/categoryId/189147203](https://mediacenter.ibm.com/playlist/details/1_0o0vq1h9/categoryId/189147203)

Webinar on Developer on z/OS on Cloud:

[https://ibm.webcasts.com/starthere.jsp?ei=1720262&tp\\_key=daf37cfc24](https://ibm.webcasts.com/starthere.jsp?ei=1720262&tp_key=daf37cfc24)

### IBM Wazi as a Service

<https://www.ibm.com/cloud/wazi-as-a-service>

Demo Video Playlist:

[https://mediacenter.ibm.com/playlist/details/1\\_5q5qeehb/](https://mediacenter.ibm.com/playlist/details/1_5q5qeehb/)

z/OS Development in the IBM Cloud Demo:

<https://youtu.be/OdsfPiOfKto>

Peter Haumer  
[phaumer@us.ibm.com](mailto:phaumer@us.ibm.com)  
<https://www.linkedin.com/in/haumer>

# BACKUP

## IBM ZCodeScan

- A modern linting experience for COBOL.
- Will highlight issues within your source code and Problems view.
- Open web view with detailed problem description.
- Customize which rules your organization wants to utilize and the severity level of each rule.
- In addition to VS Code integration ZCodeScan is also a CLI that can run on z/OS or other platforms to perform scans as part of your build automation pipeline.
- Use it to define Pull Request quality gates with GitLab or GitHub.
- Integrate with DBB builds.

The screenshot displays the IBM ZCodeScan interface. The top pane shows COBOL source code for 'SAM1.cbl'. A rule violation is highlighted on line 219: 'ACCEPT CURRENT-TIME FROM TIME.'. A tooltip for this rule is visible, stating: 'Rule: Use CURRENT-DATE rather than ACCEPT DATE or ACCEPT TIME. Category: Program Structure - Severity: MEDIUM. CURRENT-DATE provides a full timestamp including Date (with 4-digit year), Time (hours, minutes, seconds, milliseconds) and Time zone offset (from UTC). It works consistently across modern COBOL compilers and systems. ACCEPT DATE or ACCEPT TIME only give basic values (e.g., YYMMDD, HHMMSS) with less precision and no timezone info.'

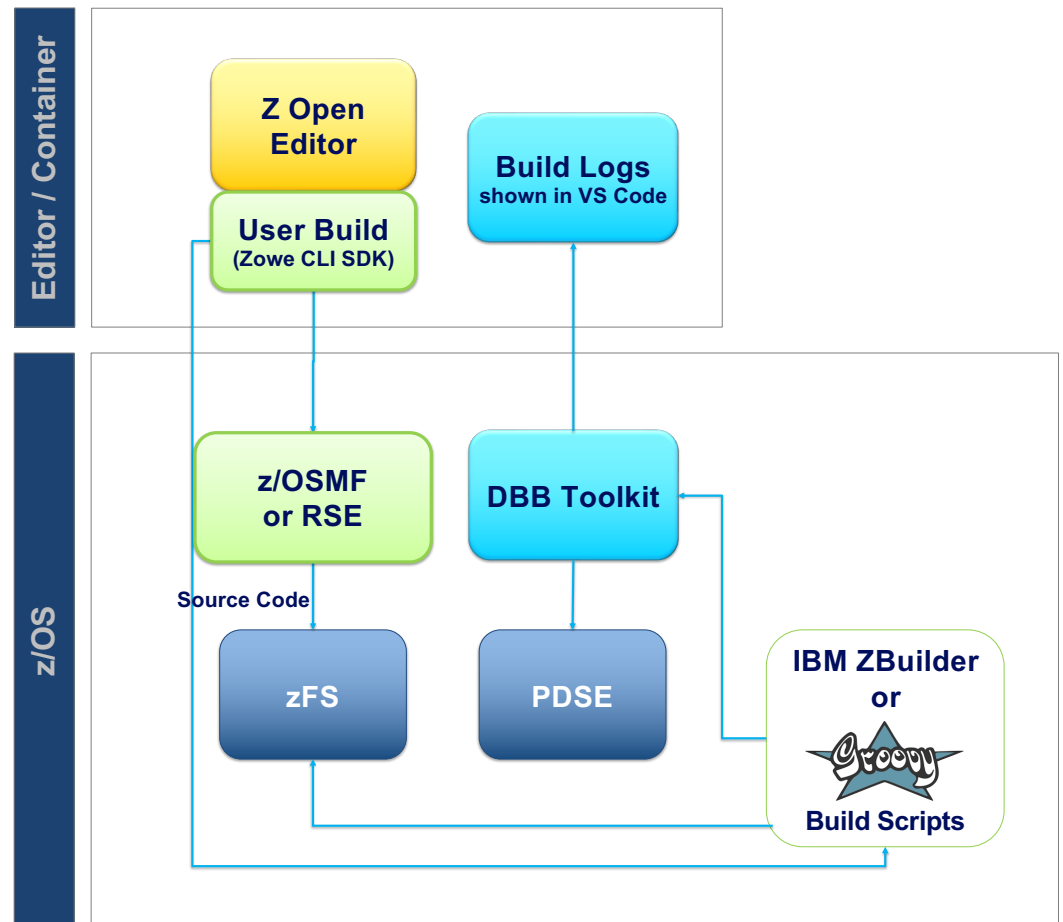
The bottom pane shows the 'PROBLEMS' view, listing several issues:

- Use of uninitialized variables. (IBM ZCodeScan) [Ln 215, Col 8]
- Use CURRENT-DATE rather than ACCEPT DATE or ACCEPT TIME (IBM ZCodeScan) [Ln 218, Col 12]
- Use CURRENT-DATE rather than ACCEPT DATE or ACCEPT TIME (IBM ZCodeScan) [Ln 219, Col 12]**
- Use CURRENT-DATE rather than ACCEPT DATE or ACCEPT TIME (IBM ZCodeScan) [Ln 244, Col 12]
- Use CURRENT-DATE rather than ACCEPT DATE or ACCEPT TIME (IBM ZCodeScan) [Ln 245, Col 12]
- End Clause is Required (IBM ZCodeScan) [Ln 304, Col 13]
- End Clause is Required (IBM ZCodeScan) [Ln 391, Col 12]
- End Clause is Required (IBM ZCodeScan) [Ln 426, Col 12]

## Dependency Based Build in IBM Z Open Editor

### User Builds of a single program

- Adds build when editing sources locally using Git.
- Uses Zowe profile and personal settings in addition to ZAPP.
- Start a build with simple right-click or shortcut.
- Build operation uses language server to find local copybooks and uploads with program.
- Executes build with settings specified in ZAPP.
- Downloads build logs and populates VS Code Problems views with compiler warnings and info messages.



# watsonx Code Assistant for Z

# What are Bob's strengths?

Delivering AI at scale with ROI requires a unique approach. Bob is designed for that.

## A Trusted Development Partner Your Teams Can Rely On

Enable consistent, confident engineering across the SDLC, with an AI partner teams want to adopt

## Modernization That Improves Velocity Without Sacrificing Quality

Scale software development while maintaining quality

## Cost Control & Transparency That Sustains Innovation

Optimize, control, and reduce costs across the SDLC

## Shift-left without adding friction

Detect issues earlier, reduce rework, and deliver with confidence

45% Productivity improvement across SDLC tasks

40% Faster delivery for complex, engineering work

40% Lower AI compute costs with task-aware execution and routing

65% Daily usage with higher confidence and trust

80% Efficiency gains in structured workflows

90% Time savings on repetitive SDLC tasks

Source for ROI metrics: IBM Client 0

# IBM watsonx Code Assistant for Z

Accelerated application lifecycle

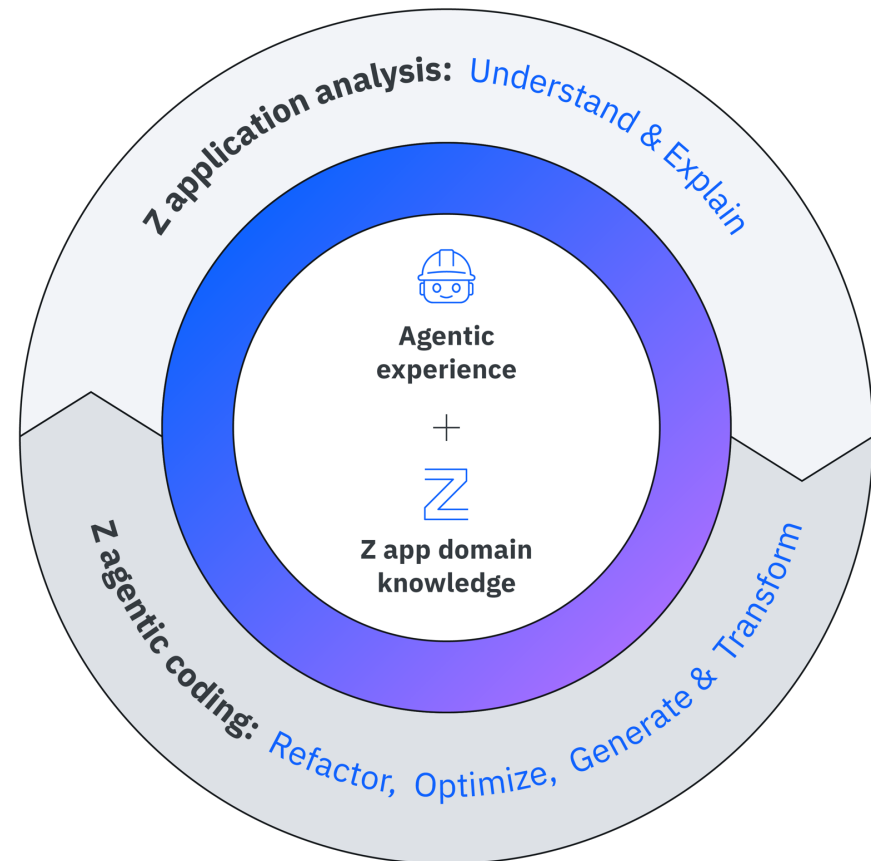
## Agentic experience

brings purpose-built AI and automated tooling for end-to-end modernization workflows, from application analysis to coding

## Z application domain knowledge

infuses tooling with Z specific language and middleware context

AI-assisted mainframe application modernization



# Agentic mainframe developer experience

Empower developers with AI agents that proactively understand goals, automate tasks, and collaborate as intelligent partners, driving development forward with smart, goal-aware assistance

## Z application analysis

Plan and analyze mainframe applications to identify how to enhance or modernize applications before implementation

*"What is the impact of changing this business rule in my application?"*

*"What is the impact if I add this capability into my application?"*

*"What parts of my application use this table?"*




## Z agentic coding

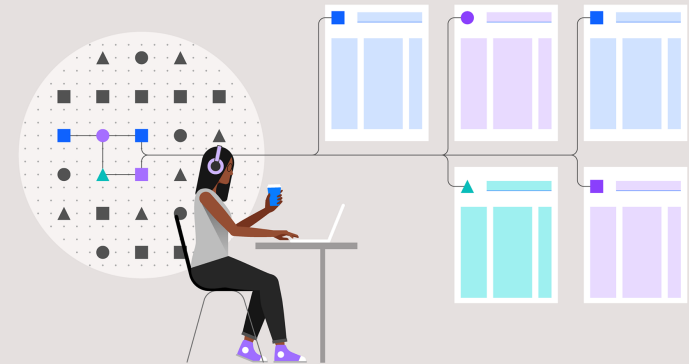
Explain, generate, and modify mainframe application code; refactor monolithic code to reduce complexity

*"Create a new COBOL program that does X using my coding standards"*

*"Is there duplicate code in my program?"*

*"Refactor this code to remove GOTO statements"*

-  **Boost Productivity** – Innovate faster by offloading routine tasks to AI
-  **Accelerate Delivery** – Achieve faster time to value with goal-driven automation
-  **Improve Quality** – Identify issues early with continuous AI guidance



# Z application analysis

Understand and document mainframe applications, enriching AI outputs with enterprise specifics, driving smarter, context-aware insights

Sample agentic chat prompts for Z application analysis:

*"Build an architectural plan to add multi-factor authentication into my application."*

*"What programs are impacted if I update the Motor Policy table?"*

*"What parts of my application use the Customer table?"*



## Understand applications

In-depth, enterprise-wide analysis and visualization of applications, programs, and their dependencies, enabling stronger business alignment for modernization initiatives, reduced risk, and faster time to market



## Document applications

Automatically generate high-quality documentation of complex Z application structure, dependencies, and business logic using an agentic workflow and Z Understand metadata, reducing reliance on experts and accelerating onboarding



## Enhance outputs with enterprise context

Ability to include enterprise specifics, such as data dictionaries and business glossaries, into code explanations and documentation, enabling the LLM to deliver precise and business-aware analysis

# Z agentic coding

Execute complex, time-intensive coding tasks by leveraging intelligent AI agents and automated workflows to achieve faster, more efficient outcomes

Sample chat prompts for Z agentic coding:

*"Update this program to insert a new row into the existing table that is based on the customer number."*

*"Explain what this COBOL module does and help me refactor into a series of modular business services."*

*"Based on the errors listed in the log output, resolve the errors accordingly for the programs in my workspace."*



## AI-assisted coding

Implement new features by generating, refining, and transforming code with an agentic chat interface that leverages MCP-enabled tools, accelerating delivery and improving code quality



## Streamline and simplify code

Analyze code blocks and programs to refactor monolithic code into modular, maintainable structures, reducing technical debt and lowering change risk



## Optimize and debug code

Improve performance and maintainability with prioritized insights and automated error detection and resolution, reducing defect-related downtime and enabling faster, higher-quality releases with greater confidence

# Example workflow - Update policy table

Sample developer prompt

*“I need to add a column to the Motor Policy Table that captures if the vehicle is an electric car. Can you help me to update all the programs with this field.”*

AI agents automatically identify dependencies and generate code, ensuring coding rules are applied, before compile and build, to accelerate delivery, reduce errors, and ensure consistent quality across enterprise modernization initiatives

