VS Code Extension for ABL

New & Noteworthy + Roadmap

Gilles Querret
Riverside Software





About the speaker

- Founded Riverside Software in 2007
- Continuous integration, source code analysis and development environment with OpenEdge
- Official SonarSource partner

VS Code - History

- Announced in April 2015 by Microsoft
- First stable release in April 2016
- Code editor based on Electron framework
  - Files and workspaces
  - Debugger support
  - Terminal
  - Source control
  - Remote development + web based development
  - Containers
  - Extensions + marketplace

VS Code – LSP & DAP

- Language Server Protocol and Debug Adapter Protocol
- Component between the code editor and target language
- Open specification
- Language Servers and Debug Adapters can be reused in any other editor

ABL Extension - History

- Announced in November 2021 during the POET Summit
- First release on the marketplace in July 2022

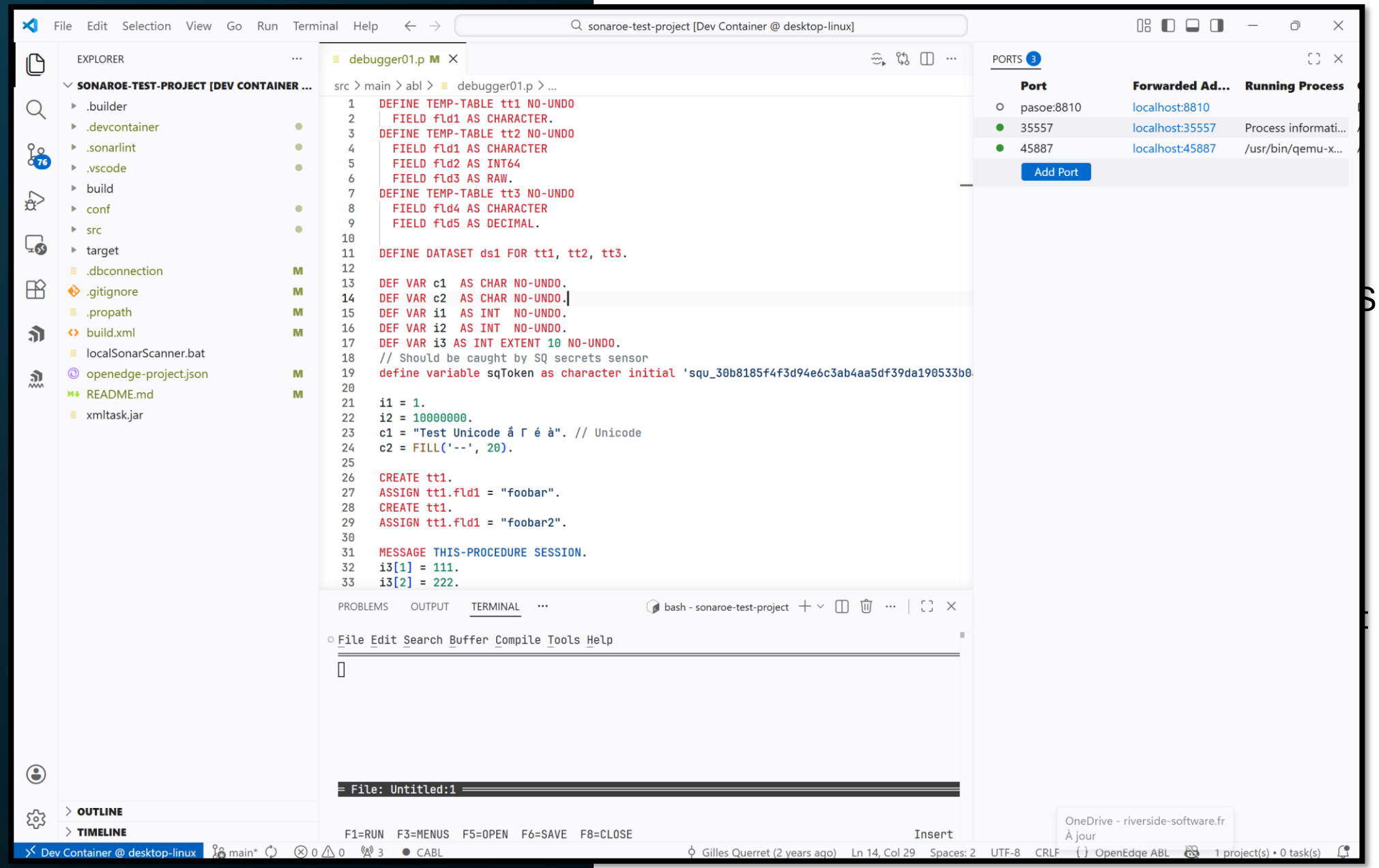
- Syntax highlight
- Code completion
- Task execution
- Hover
- Definition
- Debugger

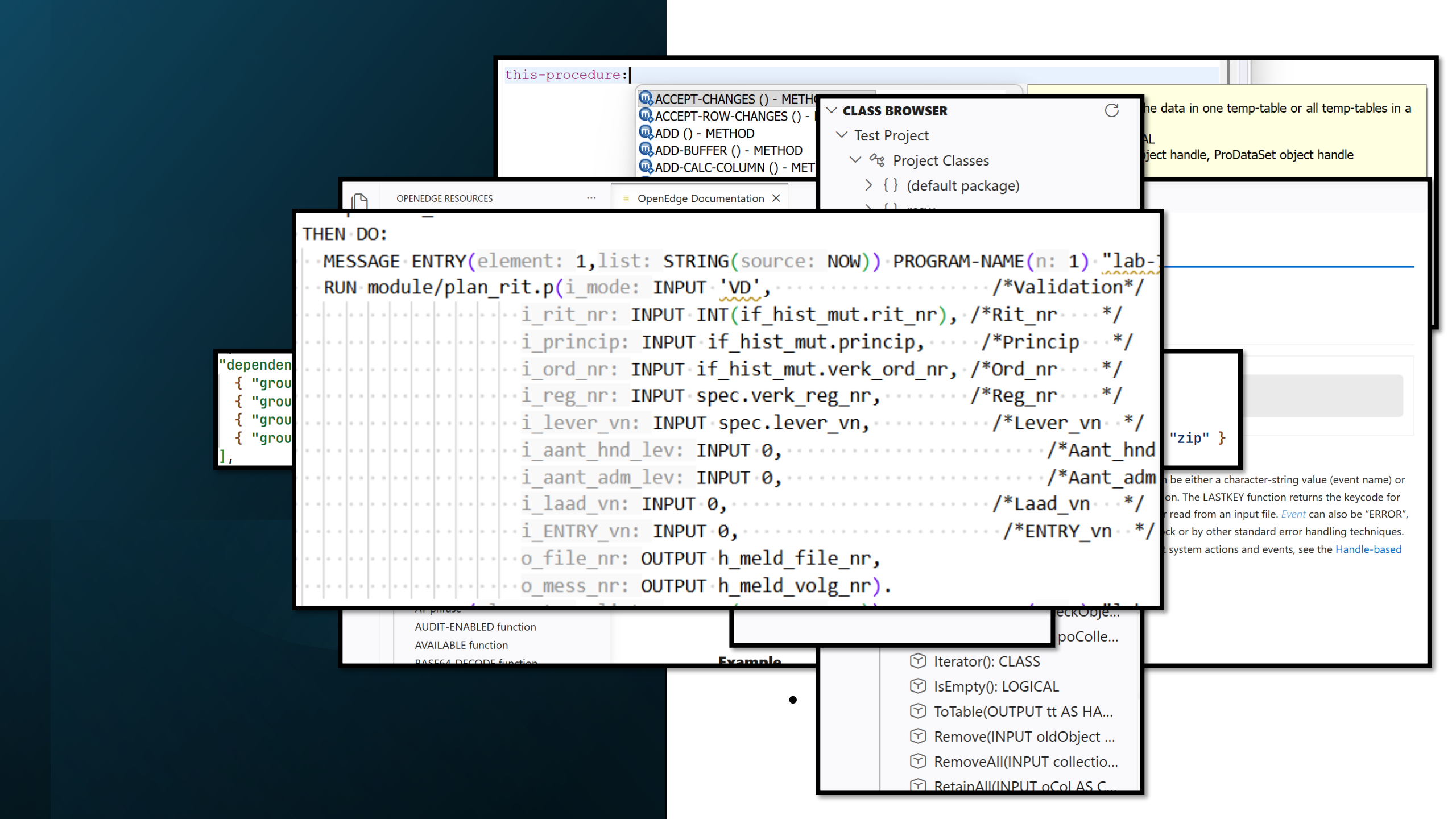
ABL Extension – Sponsored by Progress

- Announced in June 2025 during the French PUG event

ABL Extension – Figures

- 25k installs from the marketplace, 65k from OpenVSX
- 90% stable / 10% prerelease
- 95% VS Code – 3% Devin – 1% Cursor – 15 other user agents
- 80% 12.8, then 12.2, 11.7, 13.x, then non-LTS versions
- 70% Windows, 30% Linux, then MacOS
- Mean size: 5k files per project, up to 70k


File Edit Selection View Go Run Terminal Help
sonaroe-test-project [Dev Container @ desktop-linux]
EXPLORER
SONAROE-TEST-PROJECT [DEV CONTAINER ...
.builder
.devcontainer
.sonarlint
.vscode
build
conf
src
target
.dbconnection
.gitignore
.propath
build.xml
localSonarScanner.bat
openedge-project.json
README.md
xmltask.jar
OUTLINE
TIMELINE
debugger01.p M
src > main > abl > debugger01.p > ...
1 DEFINE TEMP-TABLE tt1 NO-UNDO
2 FIELD fld1 AS CHARACTER.
3 DEFINE TEMP-TABLE tt2 NO-UNDO
4 FIELD fld1 AS CHARACTER
5 FIELD fld2 AS INT64
6 FIELD fld3 AS RAW.
7 DEFINE TEMP-TABLE tt3 NO-UNDO
8 FIELD fld4 AS CHARACTER
9 FIELD fld5 AS DECIMAL.
10
11 DEFINE DATASET ds1 FOR tt1, tt2, tt3.
12
13 DEF VAR c1 AS CHAR NO-UNDO.
14 DEF VAR c2 AS CHAR NO-UNDO.
15 DEF VAR i1 AS INT NO-UNDO.
16 DEF VAR i2 AS INT NO-UNDO.
17 DEF VAR i3 AS INT EXTENT 10 NO-UNDO.
18 // Should be caught by SQ secrets sensor
19 define variable sqToken as character initial 'squ_30b8185f4f3d94e6c3ab4aa5df39da190533b0
20
21 i1 = 1.
22 i2 = 10000000.
23 c1 = "Test Unicode ắ Γ é à". // Unicode
24 c2 = FILL('--', 20).
25
26 CREATE tt1.
27 ASSIGN tt1.fld1 = "foobar".
28 CREATE tt1.
29 ASSIGN tt1.fld1 = "foobar2".
30
31 MESSAGE THIS-PROCEDURE SESSION.
32 i3[1] = 111.
33 i3[2] = 222.
PROBLEMS OUTPUT TERMINAL
bash - sonaroe-test-project
File Edit Search Buffer Compile Tools Help
File: Untitled:1
F1=RUN F3=MENUS F5=OPEN F6=SAVE F8=CLOSE
Insert
PORTS 3
Port
Forwarded Ad...
Running Process
pasoe:8810
localhost:8810
35557
localhost:35557
Process informati...
45887
localhost:45887
/usr/bin/qemu-x...
Add Port
OneDrive - riverside-software.fr
À jour
Dev Container @ desktop-linux
main*
CABL
Gilles Querret (2 years ago)
Ln 14, Col 29
Spaces: 2
UTF-8
CRLF
OpenEdge ABL
1 project(s) • 0 task(s)


this-procedure:
ACCEPT-CHANGES () - METH
ACCEPT-ROW-CHANGES () -
ADD () - METHOD
ADD-BUFFER () - METHOD
ADD-CALC-COLUMN () - MET
CLASS BROWSER
Test Project
Project Classes
{ } (default package)
OPENEDGE RESOURCES
OpenEdge Documentation
THEN DO:
MESSAGE ENTRY(element: 1,list: STRING(source: NOW)) PROGRAM-NAME(n: 1) "lab-
RUN module/plan_rit.p(i_mode: INPUT 'VD', /*Validation*/
i_rit_nr: INPUT INT(if_hist_mut.rit_nr), /*Rit_nr */
i_princip: INPUT if_hist_mut.princip, /*Princip */
i_ord_nr: INPUT if_hist_mut.verk_ord_nr, /*Ord_nr */
i_reg_nr: INPUT spec.verk_reg_nr, /*Reg_nr */
i_lever_vn: INPUT spec.lever_vn, /*Lever_vn */
i_aant_hnd_lev: INPUT 0, /*Aant_hnd
i_aant_adm_lev: INPUT 0, /*Aant_adm
i_laad_vn: INPUT 0, /*Laad_vn */
i_ENTRY_vn: INPUT 0, /*ENTRY_vn */
o_file_nr: OUTPUT h_meld_file_nr,
o_mess_nr: OUTPUT h_meld_volg_nr).
"dependen
{ "grou
{ "grou
{ "grou
{ "grou
],
"zip" }
AUDIT-ENABLED function
AVAILABLE function
Example
Iterator(): CLASS
IsEmpty(): LOGICAL
ToTable(OUTPUT tt AS HA...
Remove(INPUT oldObject ...
RemoveAll(INPUT collectio...
RetainAll(INPUT oCol AS C...

Dependency Management

- **The problem**
- OpenEdge projects depend on shared libraries (.pl, r-code, sources)
- Today: manual copy, shared network drives, no versioning
- Upgrading a shared library means touching every project

- **The goal**
- Declare dependencies in openedge-project.json
- Language server resolves and downloads them automatically
- Transitive dependencies handled transparently

Dependency Management

- **Maven artifact format**
- groupId / artifactId / version — standard Maven coordinates
- Classifiers: sources (.zip), r-code (.zip), procedure libraries (.pl)
- Hosted on a Nexus / Artifactory repository

- **Language server integration**
- Maven Resolver (Eclipse Aether) — no CLI Maven required
- Repository authentication via settings.xml
- Transitive dependency graph resolved automatically

Dependency Management

- **Declaration in openedge-project.json**
- *{ "groupId": "com.example", "artifactId": "mylib", "version": "1.0.0" }*

- **Status & roadmap**
- Direct dependencies: available in current builds
- Transitive resolution: in progress (Maven Resolver)
- Publish your own libs: Nexus/Artifactory + Maven deploy
- Progress OpenEdge artifacts: planned for future releases

Extension Entrypoints

- **What are entrypoints?**
- VS Code extensions can expose a public API to other extensions
- Consumed with vscode.extensions.getExtension(‘id’).exports

- **What the ABL extension exposes**
- Language server instance and connection handle
- Project model: propath, databases, build output
- Commands: compile, run, check syntax — callable programmatically

Extension Entrypoints

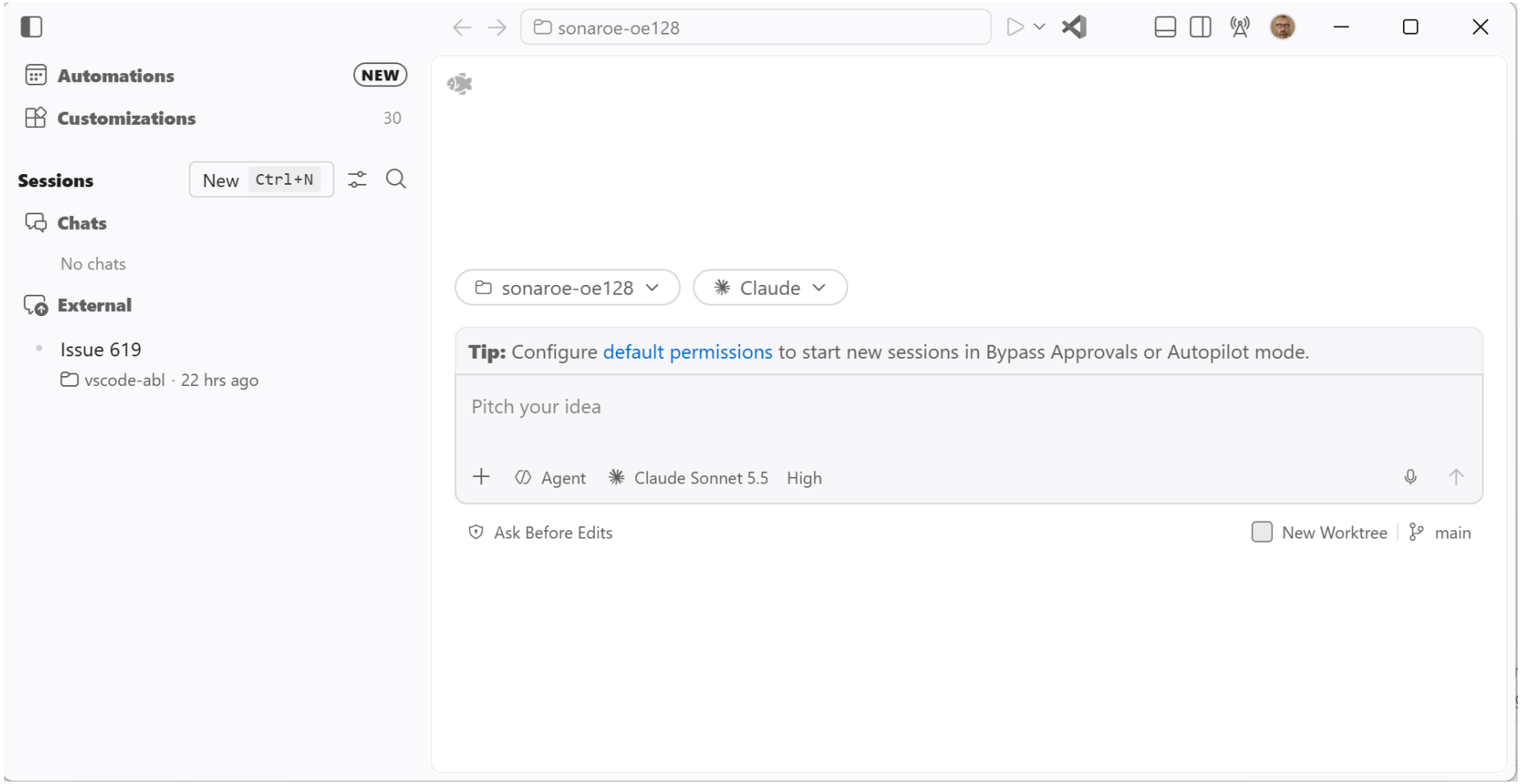
- **Use cases for third-party extensions**
- SonarLint: retrieve propath to analyse ABL files in context
- Custom build tools: trigger compile/run without shell scripts
- AI extensions: query project model to provide accurate context

- **Roadmap**
- Stable, versioned API contract
- Event system: notify consumers on project reload or LS restart

Language Model Tool API

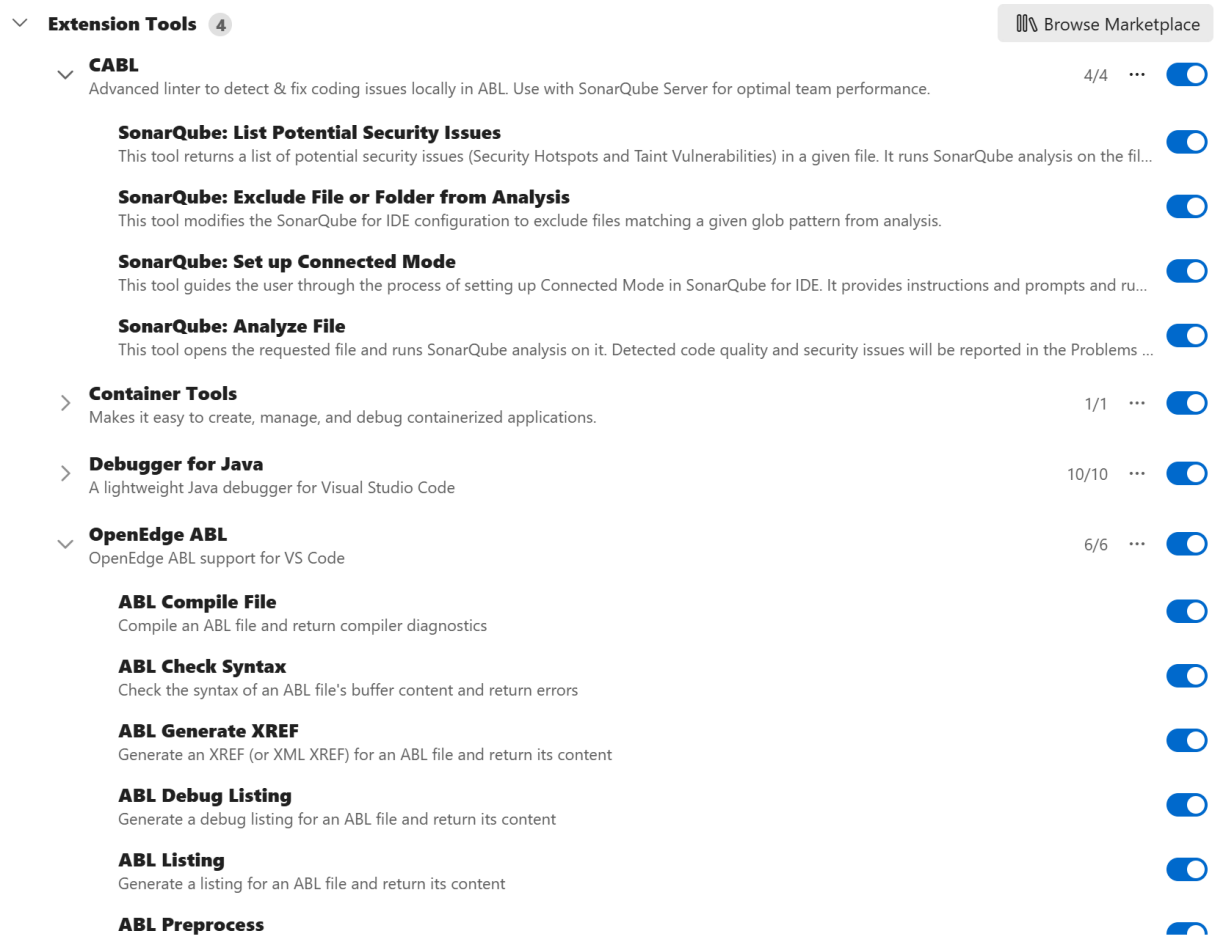
- Function that can be invoked as part of a language model request
  - Retrieve information from a database
  - Perform some calculation
  - Call online API
  - …
- Not executed by the LLM
- LLM only generates the parameters used by the tool

- Check syntax on generated files
- Create XREF
- Execute debugger
- Database schema
- …



Language Model Tool API

VS Code Agent mode



Language Model Tool API

VS Code Agent mode

Not on the roadmap (in the core extension)

- No GUI for .Net
- No PASOE management



Feedback

GitHub issues:
https://github.com/vscode-abl/vscode-abl/issues