

60 Years in the Making: Building a language server for PL/I

Mark Sujew

TypeFox

Vit Gottwald

 **BROADCOM®**

 **LangDev
CON2026**

Torremolinos, 8-9 October, 2026

<https://langdevcon.org>

Disclaimer

Certain information in this presentation may outline Broadcom's general product direction. This presentation shall not serve to (i) affect the rights and/or obligations of Broadcom or its licensees under any existing or future license agreement or services agreement relating to any Broadcom software product; or (ii) amend any product documentation or specifications for any Broadcom software product. This presentation is based on current information and resource allocations as of October 2026, and is **subject to change or withdrawal by Broadcom at any time without notice. The development, release and timing of any features or functionality described in this presentation remain at Broadcom's sole discretion.**

Notwithstanding anything in this presentation to the contrary, upon the general availability of any future Broadcom product release referenced in this presentation, Broadcom may make such release available to new licensees in the form of a regularly scheduled major product release. Such release may be made available to licensees of the product who are active subscribers to Broadcom maintenance and support, on a when and if-available basis. The information in this presentation is not deemed to be incorporated into any contract.

Broadcom may use any feedback provided by you related to a Broadcom product or this presentation for any Broadcom business purposes (including but not limited to, preparation, reproduction, and distribution of derivative works based upon such feedback), without any obligation to you including consent or payment.

Copyright © 2026 Broadcom. All rights reserved. The term "Broadcom" refers to Broadcom Inc. and/or its subsidiaries. Broadcom, the pulse logo, Connecting everything, CA Technologies and the CA Technologies logo are among the trademarks of Broadcom.

THIS PRESENTATION IS FOR YOUR INFORMATIONAL PURPOSES ONLY. Broadcom assumes no responsibility for the accuracy or completeness of the information. TO THE EXTENT PERMITTED BY APPLICABLE LAW, BROADCOM PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON INFRINGEMENT. In no event will Broadcom be liable for any loss or damage, direct or indirect, in connection with this presentation, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if Broadcom is expressly advised in advance of the possibility of such damages.





BROADCOM[®]

MAINFRAME SOFTWARE



LangDev
CON2026

Torremolinos, 8-9 October, 2026

<https://langdevcon.org>



IBM System/360 at NASA JPL. Source: NASA Jet Propulsion Laboratory (JPL)

<https://www.jpl.nasa.gov/images/slic-e-of-history-farewell-to-the-ibm-360/>

z17 CPC

Released 2025

5 nm

208 Cores

2nd Gen AI Chip

64 TB of Memory

32 CPCs in a sysplex



Source: IBM



Quiz #1

What is the #1
programming language
in the world
?



Quiz #1

What is the #1
programming language
in the **mainframe** world
?



COBOL

Comon Business-Oriented Language



Code4z - Mainframe DevX Solution

★ Planned

Access



Zowe
Explorer



Explorer
for
Endeavor



Data Editor
for
Mainframe

Analyze



Code4z
Static
Analysis



COBOL
Control
Flow



Endeavor
MCP

Edit



COBOL



HLASM



JCL



Easytrieve



Rexx

Database Access

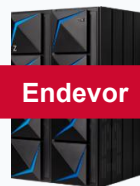


DBM-Db2
Schema
Management

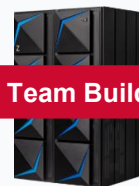


Database
API
Generator

Build



Endeavor



Team Build

Testing and Diagnostics



Debugger
for
Mainframe



Abend
Analyzer



JCL Check

Integrations



Hostbridge

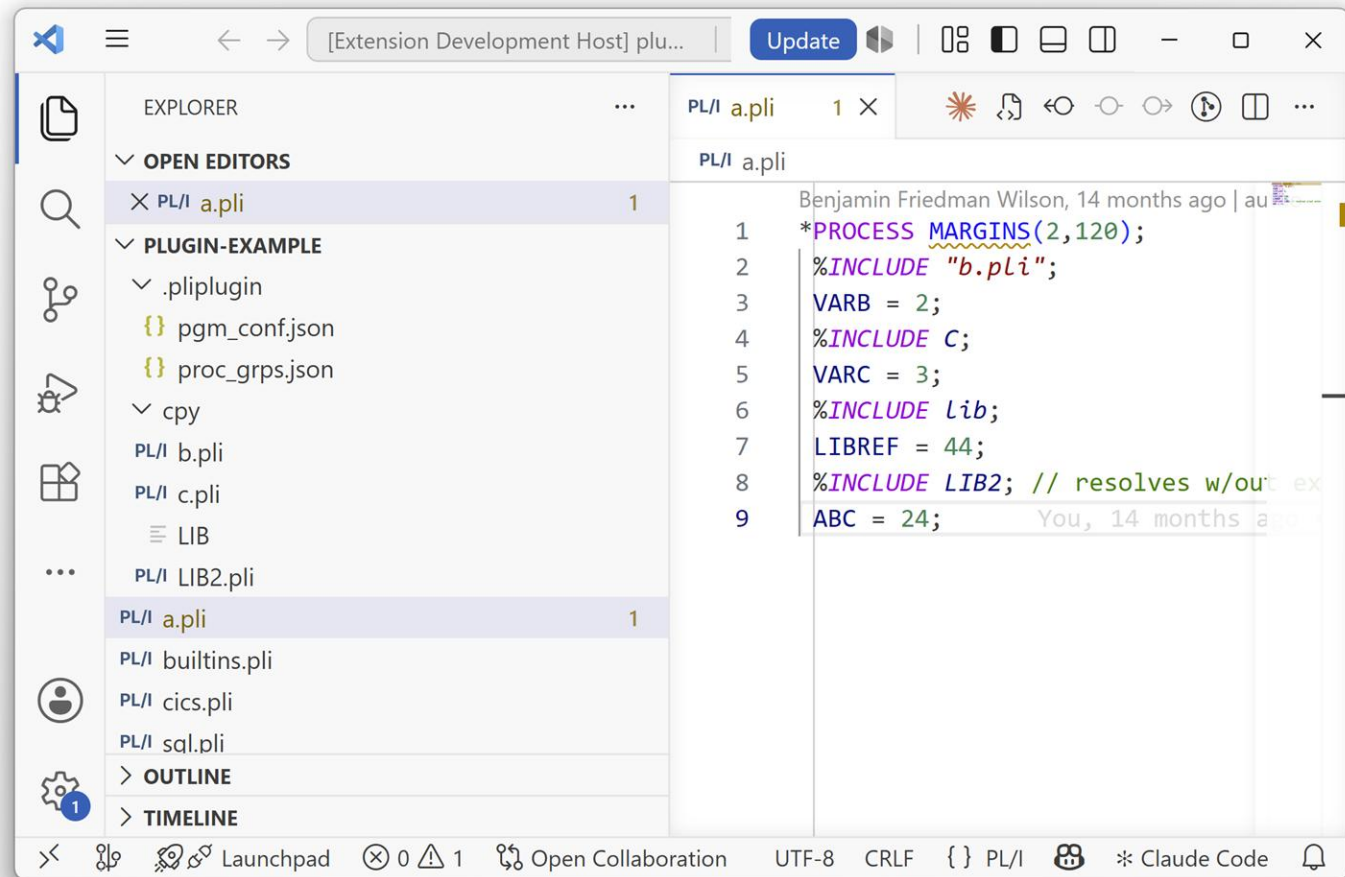
Quiz #2

What is the #2
programming language
in the **mainframe** world
?



PL/I

Running in VSCode & more



PL/I Code

```
DCL (APPLE_COUNT, CARROT_COUNT, CHERRY_COUNT, TOMATO_COUNT)
    FIXED BIN(31) INIT(0);
DCL (INPFILE, OUTFILE) FILE RECORD;
READ FILE (INPFILE) INTO (REC_INP);
DO WHILE (^EOF);
    SELECT(REC_INP.NAME);
        WHEN('APPLE') APPLE_COUNT=APPLE_COUNT+REC_INP.COUNT;
        WHEN('CARROT') CARROT_COUNT=CARROT_COUNT+REC_INP.COUNT;
        WHEN('CHERRY') CHERRY_COUNT=CHERRY_COUNT+REC_INP.COUNT;
        WHEN('TOMATO') TOMATO_COUNT=TOMATO_COUNT+REC_INP.COUNT;
        OTHERWISE PUT SKIP LIST('Unexpected order ',REC_INP.NAME);
    END;
    READ FILE (INPFILE) INTO (REC_INP);
END;
```



PL/I Syntax

- Verbose statement syntax
- Configurable syntax
- Unusual cross-reference semantics
- Non-context-free grammar
- No reserved keywords

DCL IF FIXED;

DCL THEN FIXED;

IF THEN = IF THEN...



Cross-Reference Semantics

```
// Structure declaration
DCL
  1 A,
    2 (X, Y, Z) CHAR(8),
    2 B,
    3 C FIXED;

// Fully qualified
PUT(A.B.C);
// Shortcut
PUT(C);
// Partial qualification
PUT(A.C);
```

Macro PL/I

```
%FILE_IN: PROC (FILE, INTO) STATEMENT;  
    ANSWER("DCL " || FILE || " FILE RECORD INPUT;");  
    ANSWER("READ FILE (" || FILE || ") INTO (" || INTO || ");");  
END;  
%ACT FILE_IN;  
  
FILE_IN FILE(SOME_FILE) INTO(SOME_CONTENT);  
FILE_IN FILE(OTHER_FILE) INTO(OTHER_CONTENT);  
  
// Generates:  
DCL SOME_FILE FILE RECORD INPUT;  
READ FILE (SOME_FILE) INTO (SOME_CONTENT);  
DCL OTHER_FILE FILE RECORD INPUT;  
READ FILE (OTHER_FILE) INTO (OTHER_CONTENT);
```



Macro PL/I

- Subset of PL/I
- Additional macro directives:
 - INCLUDE
 - ANSWER
 - ...
- Executed as a preprocessor in the compiler
- Live-Execution in the language server
- ...there are even more preprocessors



Technical Challenges

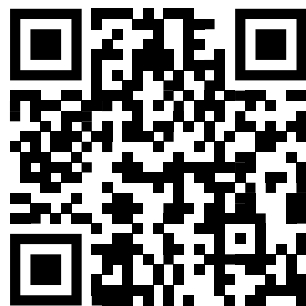
- Large, historically grown code bases
- Lack of syntax/behavior documentation
- No stand-alone compilation units
- Implementing preprocessors
 - Rerun on every keystroke
 - Multiple preprocessors!
 - Integrating with the language server



Preprocessor Challenges

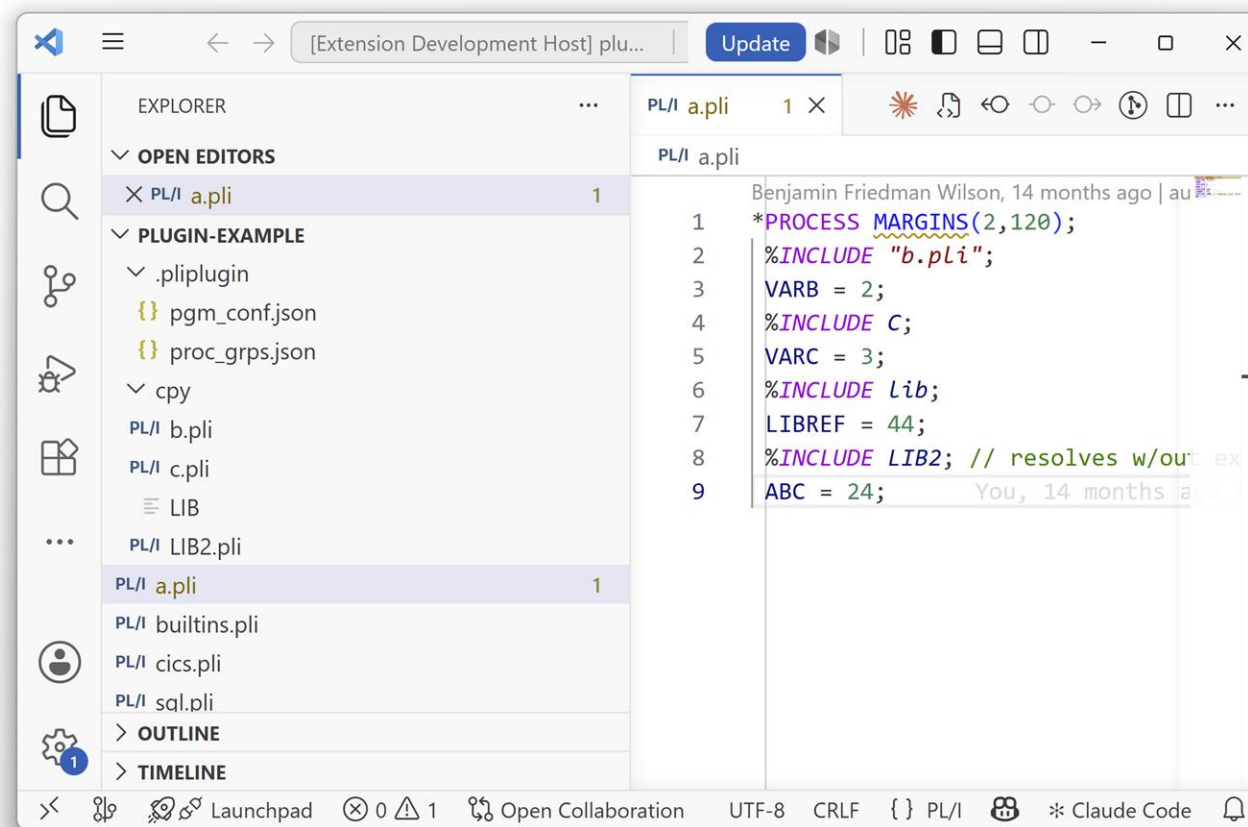
- Pure functions: `String -> { String, SourceMap }`
- Concat/Merge sourcemaps into one
- Endresult: `{ String, SourceMap }`
- Translate all positional information back and forth:
 - `{ URI, Position } -> Offset`
 - `Offset -> { URI, Position }`





Live-Demo

PL/I extension in VSCode



<https://github.com/zowe/zowe-pli-language-support>



Q&A