IMPROVING THE MAINFRAME DEVELOPER EXPERIENCE WITH PL/I INCREMENTAL PARSING

Introduction

As developers, we want to know that the code we're writing is syntactically correct and error-free. It's essential for the programmer to have a comfortable and uninterrupted computing environment. Moreover, it's important for a developer to have proper navigation throughout the document they're working on. This is where incremental parsing comes into the picture.

Let's explore the need, features, and benefits of PL/I incremental parsing. There are several issues when developing without incremental parsing, including:

1. The developer must consistently save the file.
2. There are no real-time code updates as it's written.
3. The failure to save the file masks potential code issues.
4. Excessive time spent saving detracts from focusing on the code.

Incremental Parsing in PL/I

BMC has increased the overall efficiency and reliability of programming by reducing the delay between modifying code and detecting errors in a program by performing syntactic and semantic validation of programs at run time, during the editing phase. **With the introduction of incremental**

parsing to BMC AMI DevX Workbench for Eclipse, we aim to reduce the efforts of saving the file for every change and instead provide an uninterrupted coding experience.

How does it work?

Once a developer makes any change to the PL/I document and takes a pause, the incremental parsing gets triggered.

Let's look at a small example:

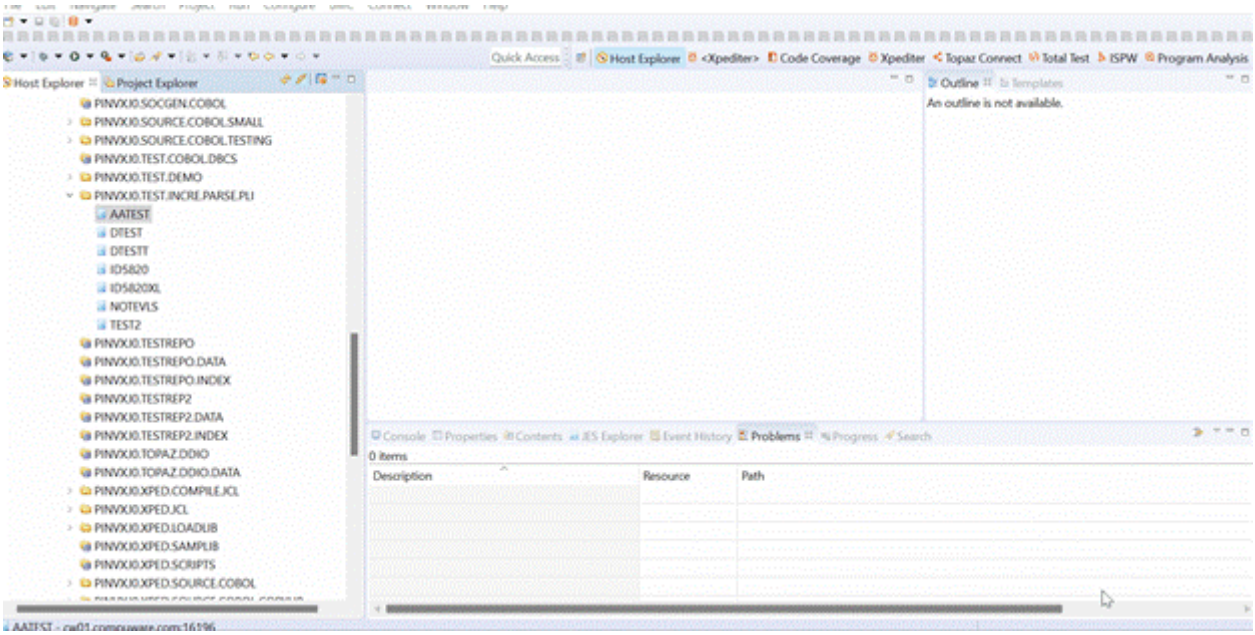

Figure 1. The changes are being reflected in the Outline view without saving the file.

If there's any syntax error, we get real-time updates about the error in the Problems view. Moreover, there's no performance impact from incremental parsing on the developer's day-to-day work.

Features

PL/I incremental parsing in BMC AMI DevX Workbench for Eclipse:

- Provides real-time updates in Outline and Problems views, along with problem markers.
- Supports inclusion of copybooks.
- Offers incremental parsing of preprocessor statements.
- Uses undo/redo commands to return to any state of the program.
- Allows navigating to the code line through Problems view.
- Works for multiple opened PL/I files.
- Supports hovering, content assist, code folding, and open declaration.

Benefits

These features improve the overall develop experience and increase efficiency by:

- Delivering runtime syntactic and semantic validation of programs.
- Allowing immediate downloading and parsing of new copybooks.
- Saving time and effort with no waiting.
- Offering easy navigation.

Conclusion

An uninterrupted programming experience is integral to any developer delivering high-quality work. Incremental parsing and run-time syntactical analysis of code improves the developer experience and helps them focus on what they do best—writing code that serves customers' needs.

Learn more about BMC's Eclipse-based mainframe IDE in the BMC AMI DevX Workbench for Eclipse data sheet.