



## IBM Enterprise PL/I for z/OS, Version 3.7

---

### Highlights

---

- ***Generates XML code to support PL/I structures***
- ***Enables your IT systems to exchange and use data in standardized formats, including XML and Unicode***
- ***Improves compatibility of PL/I programs and Java components across distributed applications***
- ***Provides enhanced debugging support***
- ***Enables you to reuse existing applications in traditional z/OS and Web service environments***
- ***Includes compatibility improvements to ease upgrading from earlier PL/I compilers***
- ***Delivers integrated CICS and SQL preprocessors***

### **Enabling a comprehensive business integration strategy**

To compete effectively, your organization must be able to connect business components end to end with suppliers, partners, employees and customers. Such connectivity provides the foundation for a flexible business with an open, integrated operating environment—positioning the organization to quickly take advantage of opportunities or respond to threats in realtime.

Unfortunately, many existing IT systems were not designed to address these objectives—or the increasing dependence on Web services and service-oriented architecture (SOA). And you can't afford the downtime or added expense and risk of rewriting core applications. What's needed is a comprehensive business-integration strategy that is designed to help you integrate, modernize and manage the existing applications, data and skill sets that can ease your organization's transformation.

### **Integrate, modernize and manage your application assets with Web services capabilities**

The IBM Enterprise PL/I for z/OS, Version 3.7 compiler can help you integrate your PL/I and Web-based business processes in Web services, XML, Java™ and PL/I applications. You're able to take advantage of more than 30 years of IBM experience in application development to facilitate your enterprise application integration and SOA initiatives. Enterprise PL/I for z/OS is an integral part of the comprehensive application-development environment delivered with IBM Rational® Developer for System z™ software—providing a robust, integrated development environment (IDE) for PL/I and connecting Web services, Java Platform, Enterprise Edition (Java EE) applications and traditional business processes.

An advanced IBM z/OS® system based compiler helps you create and maintain critical, line-of-business PL/I applications targeted to run on your z/OS systems. As a result, you can realize the full potential of your current z/OS investments and provide access to IBM DB2®, IBM CICS® and IBM IMS™ systems, as well as other data and transactions systems. Enterprise PL/I for z/OS also delivers XML enablement, parsing and generation to further integrate your business-to-business (B2B) Web applications. And, enhanced performance and usability features make upgrading easier, increase debug support and provide smooth integration with Rational Developer for System z, CICS and IMS software.

#### **Facilitate Web interoperability using XML parsing and generation**

Enterprise PL/I for z/OS allows existing PL/I transactions to process inbound and outbound XML data directly within the applications. It provides a high-speed parser that enables PL/I programs to parse XML documents in Extended Binary Coded Decimal Interchange Code (EBCDIC), American Standard Code for Information Interchange (ASCII) or Unicode Transformation Format (UTF)-16. Using the IBM PL/I Simple API for XML (SAX) parser, this XML can then be passed to other applications, even those running on other platforms—including IMS and CICS environments.

Enterprise PL/I also supports the generation of XML using a built-in function, so you're able to dump the contents of a structure as XML into a buffer. You can use this XML code to enhance your existing high-performance IMS and CICS transactions that have been written in PL/I. By enabling these transactions to send and receive XML documents, you're better positioned to support a B2B environment.

#### **Improve compatibility of PL/I programs and Java components**

Because earlier versions of the Enterprise PL/I compiler support ASCII and UTF-16, adding support for the Institute of Electrical and Electronics Engineers (IEEE) decimal floating-point standard means that the compiler can receive, manipulate and send Java data without any

translation. To further improve Java interoperability, Enterprise PL/I for z/OS also provides a thread-safe PL/I library and multithreading statements (ATTACH, WAIT, DETACH) as part of the PL/I language supported by the compiler.

### **Enhance application performance**

You get better hardware performance with Enterprise PL/I for z/OS, Version 3.7 because the BASR instruction is used instead of the BALR instruction. Additional performance enhancements include:

- *UNPKU to convert some PICTURE to WIDECHAR (rather than making a library call).*
- *Use of FIXED BIN(63) as an intermediary to speed conversions of FIXED DEC with large precision to FLOAT.*
- *Inlined CHAR built-in when applied to CHAR expressions.*
- *Improved code generated for conversions of FIXED BIN(p,q) to unscaled FIXED DEC.*

Features and benefits in earlier releases of the Enterprise PL/I for z/OS compiler continue to improve your application performance. Specifically, under the ARCH(7) option, the compiler supports extended immediate facility in z/OS, which provides 32-bit immediate-operand versions of instructions, including ADD IMMEDIATE, AND IMMEDIATE, COMPARE LOGICAL IMMEDIATE, EXCLUSIVE OR IMMEDIATE, INSERT IMMEDIATE, LOAD IMMEDIATE, OR IMMEDIATE, and SUBTRACT LOGICAL IMMEDIATE. These, and other, instructions enable the compiler to produce shorter and faster code sequences.

Under the ARCH(5) option, the compiler can use CVBG and CVDG to convert between FIXED BIN and FIXED DEC (and some PICTURE commands) when the precisions are too large for the conversion to be done with CVB or CVD.

In addition, conversions between dates using any of the three supported DB2 date-time patterns or the corresponding date-time patterns without any punctuation can now be performed inline. And code for REPATTERN can be inlined if the target date has a DB2 date-time pattern, and the source has either a DB2 date-time pattern or a date-time pattern that starts with YYYYMMDD.

### **Increase usability**

With the 3.7 release, Enterprise PL/I provides many usability enhancements that save time for development staff. Several are highlighted below. Support for the IEEE decimal floating-point standard has already been discussed earlier under improvements to PL/I and Java compatibility.

The new STACKADDR built-in function returns the address of the current dynamic save area (register 13 on z/OS) and makes it easier for programmers to write their own diagnostic code.

With the new MAXNEST option, excessive nesting of BEGIN, DO, IF and PROC statements are flagged by the compiler. The new (and non-default) suboption NOELSEIF of the RULES option also enables the compiler to flag any ELSE statement that is immediately followed by an IF statement and suggest that it be rewritten as a SELECT statement. And under the new (and non-default) suboption NOLAXSTG of the RULES option, the compiler flags where a variable A is declared as BASED on ADDR(B) and STG(A) > STG(B). This flag occurs not only when B is AUTOMATIC, BASED or STATIC with constant extents (as the compiler did before), but also when B is a parameter declared with constant extents.

Using the new QUOTE option, developers can name alternate code points for the quote ( ' ) symbol since this symbol is not code-page invariant. And the new XML compiler option allows them to specify that the tags in the output of the XMLCHAR built-in function be either in all upper case or in the case in which they were declared.

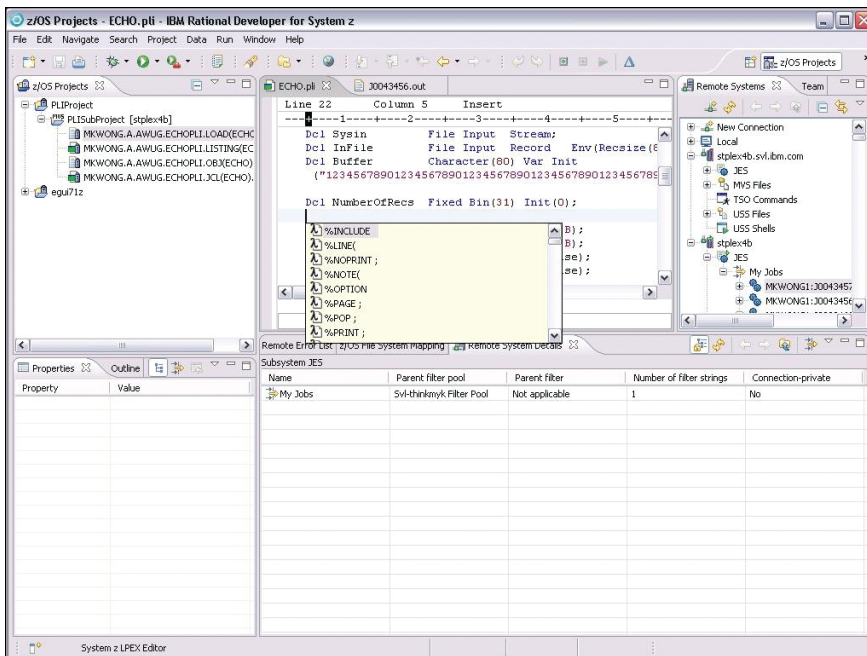
The MACRO preprocessor now supports a new suboption that enables a developer to choose whether it should process only %INCLUDE statements or whether it should process all macro statements.

The line-of-business (DB2) SQL preprocessor option has also been enhanced. When it is selected, the integrated SQL preprocessor will now generate DB2 precompiler style declares for all \*LOB\_FILE, \*LOCATOR, ROWID, BINARY and VARBINARY SQL types, in addition to the BLOB, CLOB and DBCLOB SQL types already supported.

### **Provide a higher level of debug support**

The Enterprise PL/I compiler supports both the level of IBM Debug Tool feature included in the version 3.7 full-function offering, and the separately orderable IBM Debug Tool Utilities and Advanced Functions for z/OS product, which includes automonitor support for PL/I programs. Automonitor support automatically displays the values of the variables referenced in the current statement in the Debug Tool monitor window. The AUTOMONITOR output also includes the value of the variable about to be changed by the assignment.

In version 3.7, the TEST option has been enhanced so you can choose to view the source in the listing and in the Debug Tool source window as that source would appear after a user-specified preprocessor had been run (or after all the preprocessors had been run).



Tools such as Rational Developer for System z can improve developer productivity when editing and debugging PL/I applications.

### Integrate CICS and SQL preprocessors

With integrated CICS and SQL preprocessors, you don't need to run a separate job step that precompiles EXEC CICS or EXEC SQL statements into PL/I code. Instead, the compile step handles these statements the

same way it handles any use of the macro facility. And because debugging is run against the source code fed to the compiler, you can now debug against the source you wrote instead of against the code the CICS or SQL precompiler produced.

### Help boost program quality and serviceability

The use of certain variables, statements and values can lead to poor performance or unexpected results. To help improve program quality, the Enterprise PL/I compiler in version 3.7 automatically flags for the following:

- Use of *AUTO* (and *STATIC*) variables as tables in *TRANSLATE* and *VERIFY*.
- Use of a function to set the initial value in a *DO* loop.
- Duplicate *WHEN* values in *SELECT* statements even if *'* is specified as a *WHEN* value.
- Any *%DECLARE* that does not specify any attributes, thus leading to easier detection.
- *SUBSTR* references where the third argument is zero, since while technically valid, this is almost certainly a coding error.
- Specification of a scale factor in a *FLOAT* declaration since this is invalid and may point to a typo in the source where *FLOAT* was typed but *FIXED* was meant.
- Bit prefix operands that do not have the *BIT* attribute.

You can also have the compiler optionally flag storage overlay problems where the base variable is a parameter.

To improve serviceability, the compiler now flags a CLOSE statement for a file if that statement occurs in an ENDFILE ON-block. In this way, you're able to detect a user error that is usually resolved only after a PMR has been opened. In addition, ON ERROR blocks that do not start with ON ERROR SYSTEM are flagged, thus making it less likely users will have ON ERROR blocks that go into "infinite" loops.

### **Ease migration**

Designed to simplify upgrades, Enterprise PL/I for z/OS delivers features that make it more compatible with the IBM PL/I for MVS and VM, Version 2.3 and IBM OS PL/I, Version 2.3 compilers. You're able to ease migrations with:

- *Improved default initialization of variables.*
- *Diagnostics to flag storage overlay problems.*
- *Aggregate and storage listings that provide the same information as offered by the old compiler.*
- *Messages to highlight erroneous code as well as possible upgrade problems.*

Enterprise PL/I for z/OS, Version 3.7 also provides support for IBM DB2, Version 9, including:

- *STDSQL(YES/NO).*
- *CREATE TRIGGER (also known as multiple SQL statements).*
- *FETCH CONTINUE.*
- *SQL style comments ('--') embedded in SQL statements.*
- *Additional SQL TYPES, such as BLOB\_FILE, CLOB\_FILE, DBCLOB\_FILE, XML AS, BIGINT, BINARY and VARBINARY.*
- *Listing the DB2 coprocessor runtime options.*
- *Special handling of EXEC SQL DECLARE VARIABLE statements.*
- *Special handling of EXEC SQL xxx SQL TYPE IS xxx statements.*

### **Becoming a more flexible business**

Enterprise PL/I for z/OS, Version 3.7 provides capabilities designed to streamline your business processes, enabling smooth integration with the systems of your customers, trading partners, suppliers and employees. It can help create your SOA and enable application reuse, so that you can make the most of your existing infrastructure and more easily modernize your applications for enhanced business competitiveness.

### **For more information**

To learn more about how IBM Enterprise PL/I for z/OS, Version 3.7 software can help your enterprise in the transformation to a flexible, performance-driven business, contact your IBM representative or IBM Business Partner, or visit:

[ibm.com/software/awdtools/pli/plizos](http://ibm.com/software/awdtools/pli/plizos)

To learn more about IBM Rational software, visit:

[ibm.com/software/rational](http://ibm.com/software/rational)

[ibm.com/software/awdtools/rdz](http://ibm.com/software/awdtools/rdz)

To see the hardware and software requirements for the IBM Enterprise PL/I for z/OS, Version 3.7 software, please visit:

[ibm.com/software/awdtools/pli/plizos/sysreq](http://ibm.com/software/awdtools/pli/plizos/sysreq)



© Copyright IBM Corporation 2007

IBM Corporation  
Software Group  
Route 100  
Somers, NY 10589  
U.S.A.

Produced in the United States of America  
11-07  
All Rights Reserved

DB2, CICS, IBM, the IBM logo, IMS, Rational, System z and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems in the United States, other countries or both.

Other company, product and service names may be trademarks or service marks of others.

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.