IBM Enterprise PL/I, V4.2

Enable the integration of business critical PL/I applications with modern web technology

Highlights

- Enables the creation, maintenance, and modernization of business-critical PL/I applications on z/OS® systems
- Utilizes the latest z/Architecture® through z196 facilities, the UNROLL compiler option and changes to inline code generation for improved application performance
- Delivers enhanced middleware support through SQL preprocessor improvements
- Improves application debugging with the compiler supporting typed structures in the IBM® Debug Tool
- Improves Web interoperability through the XMLCHAR built-in function
- Leverages productivity with new options and messages
- Integrates with IBM Rational® Developer for System z® and IBM Rational Team Concert™ providing a modern development environment and a collaborative team environment

To make your business as agile and responsive as possible, you need to be able to connect your business components end to end with your suppliers, partners, employees, and customers, and you need to position your organization to quickly take advantage of opportunities by responding to challenges in real time. Unfortunately, many IT systems weren't designed to address these objectives or to support Web services and service-oriented architecture (SOA) that are essential for transforming an enterprise into a flexible business with an open, integrated operating environment. You could rewrite your applications in a different programming language in order to address these objectives, but rewriting your applications would be expensive and risky, and it could potentially create downtime that you just can't afford. To remain competitive, you need a complete business strategy to help you modernize, integrate, and manage existing applications, data, and skill sets to ease your organization's transformation into a more flexible business.

Integrates, modernizes and manages assets with web services capabilities

With Enterprise PL/I for z/OS V4 you can leverage more than 30 years of IBM experience in application development to facilitate your new On Demand Business endeavors, helping integrate PL/I and web-based business processes in web services, XML, Java, and PL/I applications. This compiler's interoperability lets you capitalize on existing IT investment while smoothly incorporating new, web-based applications as part of your organizations infrastructure.

Enterprise PL/I for z/OS is an integral part of the comprehensive application development environment delivered with IBM Rational Developer for IBM 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

IBM Enterprise PL/I is a leading-edge, z/OS-based compiler that maximizes middleware by providing access to IBM DB2®, CICS®, and IMS™ systems.

Enterprise PL/I for z/OS V4 underscores the continuing IBM commitment to the PL/I programming language on the z/OS platform.

Facilitates 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 IBM IMS and IBM CICS environments.

Enterprise PL/I for z/OS 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 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 business-to-business (B2B) environment.

In Enterprise PL/I for z/OS V4.2, XML generation through the XMLCHAR built-in function, supports XML attributes and the omission of null values for improved web interoperability. The former gives you greater flexibility in controlling the style of the XML generated while the latter allows you to produce simpler, less cluttered XML.

Utilizes IBM z/Architecture for application performance improvements

Enterprise PL/I for z/OS V4.2 utilizes the following z196 facilities for improved application performance:

- High-word
- Floating-point extension
- Population-count

The V4.2 release also provides these additional performance improvements:

- The new UNROLL compiler option lets you control loop unrolling.
- The compiler now generates inline code:
  - To resolve the ULENGTH and USUBSTR built-in functions for UTF-8 strings
  - For MEMINDEX (p, n, x) where x is WCHAR(1) just as it previously did if x was CHAR(1)
  - For STG (x) where x is a BASED variable using REFER when all NONVARYING BIT in x are ALIGNED and all other elements in x are UNALIGNED

Improves application debugging

As of this release the compiler supports typed structures in the Debug Tool so that you can debug code containing not only untyped structures but also code using HANDLE variables and typed structures.

Delivers enhanced middleware support through SQL preprocessor support

In Enterprise PL/I for z/OS, with the integrated SQL preprocessor, it is not necessary to run a separate job step that precompiles EXEC SQL statements into PL/I code. Instead, the compile step will handle EXEC SQL statements in the same way that it handles any use of the MACRO facility. Also, since debugging is against the source code fed to the compiler, you can debug against the source you wrote (rather than what the SQL preprocessor produced).

In Enterprise PL/I for z/OS V4.2, the SQL preprocessor:

- Resolves host variable names in exactly the same way the compiler itself does thereby eliminating the quiet generation of incorrect code for some SQL statements
- Is smaller (in its load module size) than in earlier releases - so the compiler footprint is now much smaller.
- Runs faster than in earlier releases
- Enables the SQL TYPE attribute to be used in a declare whenever any other PL/I data attribute is used so that there are no longer any restrictions on its use and so that your code no longer needs to know how it is represented
- Honors the following compiler options so that defaults are applied correctly and unsuitable host variables are rejected as appropriate:
  - DEFAULT(ANS/IBM)
  - DEFAUT(SHORT(HEX/IEEE))
  - DEFAULT((NON)NATIVE)
  - DEFAULT(ASCII/EBCDIC)
  - DEFAULT((NO)EVENDEC)
  - RULES((NO)LAXCTL)
- Improves the processing of the PRECISION attribute
- Recognizes the UNSIGNED and COMPLEX attributes and rejects their use in any host variable

- Supports the PACKAGE statement so that your SQL code can be structured in the same manner as CICS and other PL/I code
- Ensures that DSNHMLTR is declared in every outermost procedure containing code that needs it.
- No longer puts an unprintable character in the source when emitting code to set the SQLAVDAID
- Allows indicator arrays to have any lower bound
- Emits the SQL parameter list structure with fewer unions, fewer I nit clauses, and no additional declares based on elements of the structure.

Several other improvements in this release provide enhanced middleware support:

- The new BY DIMACROSS form of assignments makes it easier to write code to handle the results of SQL multi-row fetch.
- The compiler and the preprocessors (rather than just the SQL preprocessor when parsing EXEC SQL code) now all support < >, as a not-equals symbol.
- The new INDICATORS built-in function makes it easy to declare an array to be used as a SQL indicator variable with a structure.

Leverages productivity with new or changed options and messages

With the 4.2 release, the Enterprise PL/I for z/OS compiler:

- Provides a new compiler option to conditionally erase the listing generated from a preprocessor phase unless the phase produced some messages - so that the compiler listing will be smaller and contains fewer distracting copies of the source
- Explicitly supports some use of adjustable BASED without REFER
- Supports comparisons of POINTER to ‘’ and ‘’b with the same semantics as in the previously supported assignments of ‘’ and ‘’b to POINTER
- Has raised the maximum number of distinct include files allowed from 2047 to 4095
- Will apply the NONASSIGNABLE attribute to any parameter declared with the INONLY attribute, and there the compiler will flag any assignment to a parameter declared as INONLY

In addition, you can now exert more control over your code quality using these enhancements to the RULES compiler option:

- Under the new NOSELFASSIGN suboption of the RULES compiler option, the compiler flags assignments of variables to themselves.
- Under the new NOLAXRETURN suboption of the RULES compiler option, the compiler generates code to raise an ERROR when a RETURN with an expression is executed from a procedure coded without RETURN or vice versa.
- Also, under RULES(NOLAXENTRY), the compiler will not flag names starting with DSN, and under RULES(NOUNREF), the compiler will not flag names starting with DSN or SQL.
- The compiler supports a new STRICT/LOOSE suboption to RULES(NOLAXENTRY) so that under the LOOSE suboption OPTIONS(ASM) routines are not flagged when they do not specify a parameter list.

Also, in this release, the compiler issues new or improved messages:

- When a comma is missing in a structure declare
- When source contains invalid shift-in and shift-out bytes

Other Enterprise PL/I for z/OS features

Provides compatibility for PL/I programs and Java components

Because it supports the Institute of Electrical and Electronics Engineers(IEEE) decimal floating point standard, the Enterprise PL/I for z/OS compiler can receive, manipulate and send Java data without any translation.

Built-in functions provide support for UTF-8 and UTF-16. One example is the ULENGTH function, which returns the number of UTF-8 or UTF-16 characters in a CHAR or WIDECHAR string, respectively. A second important example is the USUBSTR function which returns the UTF-sensitive substring of a CHAR or WIDECHAR string.

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

Ease into migration

Enterprise PL/I for z/OS gives you a migration path from OS PL/I V2 and PL/I for MVS™ and VM compilers. Our Compiler and Runtime Migration Guide provides you with all the information that you might need to move your applications to a new run-time (run-time migration) and to compile your source programs with the new compiler (compiler migration). Migrating to the new compiler allows your existing applications to take advantage of new functions.

Workstation-based development

Rational Developer for System z provides an interactive, workstation-based environment to help you create, maintain, and reuse applications. Rational Developer for System z includes support for traditional development using PL/I, but also has the ability to generate web services interfaces from PL/I constructs to ease creation of web services from existing PL/I applications.

Rational Developer for System z provides a workstation interface to Debug Tool, and is also integrated with IBM File Manager and Fault Analyzer. File Manager integration enables you to access Keyed Sequence Data Set (KSDS) files from the Rational Developer for System z workbench, and gives you the ability to browse and update data sets. By integrating with Fault Analyzer, Rational Developer for System z enables you to browse Fault Analyzer ABEND reports on CICS, IMS, batch, Java, WebSphere®, and other run times. Rational Developer for System z supports Enterprise PL/I and helps improve the productivity of PL/I developers. Within the workbench you can show the context-sensitive editor, as well as a compiler listing that indicates errors from a compilation. A simple click on a diagnostic message takes you to the line of source code in error.

IBM Rational Team Concert for System z, also an Eclipse-based offering, allows you to boost programming productivity with a collaborative team environment that makes it easy to manage your distributed software projects and teams.

PL/I across platforms

Enterprise PL/I for z/OS is part of a family of compatible compilers, application development tools, and maintenance tools. Along with Enterprise PL/I for z/OS, IBM offers PL/I compilers for multiple platforms as well as IBM File Manager, IBM Fault Analyzer, and Debug Tool. As mentioned previously, the recommended workstation-based development environment is Rational Developer for System z.

Summary of features and benefits

The following table summarizes the features and benefits for Enterprise PL/I for z/OS V4.

Table 1. Summary of new features and benefits

Feature	Benefit
Designed for System z and the zEnterprise™ System	Utilizes the latest z/Architecture through z196 facilities, the UNROLL compiler option and changes to inline code generation for improved application performance.
Maximizes middleware	Delivers enhanced middleware support through SQL preprocessor improvements to facilitate application integration and modernization.
Modernize applications	XML generation through the XMLCHAR built-in function, supports XML attributes and the omission of null values for improved web interoperability.
Improves application debugging	The compiler supports typed structures in the IBM Debug Tool so that you can debug code containing not only untyped structures but also code using HANDLE variables and typed structures.
Leverages productivity with new options and messages	New or changed messages and options improve your programming practices. New or changed rules option gives you more control over your code.
Provides compatibility for PL/I and Java components	• Supports the Institute of Electrical and Electronics Engineers (IEEE) decimal floating point standard, so the compiler can receive, manipulate and send Java data without any translation • Supports UTF-8 and UTF-16 through built-in functions. • Provides a thread-safe PL/I library and multithreading statements (ATTACH, WAIT, DETACH) as part of the PL/I language for improved Java interoperability
Ease of migration	Gives you a migration path from OS PL/I V2 and PL/I for MVS and VM compilers to easily move your applications to a new run-time (run-time migration) and to compile your source programs with the new compiler (compiler migration). Migrating to the new compiler allows your existing applications to take advantage of new functions.
Integrates with a modern development environment	Rational Developer for System z (a separate product) boosts developer productivity by making it easy to edit, compile, and debug PL/I applications from your workstation.
Integrates with a collaborative team environment	Rational Team Concert (a separate product) unifies development teams by making it easy to manage your distributed software projects and teams.

Software requirements

The following table presents the software requirements for Enterprise PL/I for z/OS V4.2.

Table 2. Software requirements

Operating system	Software	Hardware
z/OS	V1.13	Enterprise PL/I for z/OS V4.2

Upgrade to Enterprise PL/I for z/OS V4.2

Upgrade to the latest Enterprise PL/I compiler and get more out of your zEnterprise investment and stay ahead of competitors on the technology curve.

For more information

To learn more about IBM Enterprise PL/I for z/OS V4.2, contact your IBM representative or IBM Business Partner, or visit: ibm.com/software/awdtools/pli/plizos.

To learn more about IBM Rational Developer for System z software, visit: ibm.com/software/rational/products/developer/systemz/

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

Produced in the United States of America
September 2011