Integrate your PL/I applications with web services, XML and Java™.

Enterprise PL/I for z/OS, Version 4.1

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.

**Highlights**

- Enables the creation, maintenance, and modernization of business-critical PL/I applications on z/OS® systems
- Exploits the latest z/Architecture® for application performance improvements
- Improves SQL processing
- Improves application debugging with the IBM Debug Tool through compiler enhancements
- Delivers enhanced XML parsing support
- Leverages productivity with new options
- Empowers your developers by providing a modern development environment
- Unifies your developers by providing a collaborative team environment

1. Separate product: IBM Rational® Developer for System z
2. Separate product: IBM Rational Team Concert for System z
Integrates, modernizes and manages assets with web services capabilities

With Enterprise PL/I for z/OS 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 organization’s 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.

This new version of Enterprise PL/I for z/OS V4.1, 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 V4.1, XML validation is supported through the XML System Services parser and the new PLISAXD built-in routine. PLISAXD has the same set of arguments as the PLISAXLD built-in subroutine except that it also has an additional, mandatory argument that is the address of an optimized schema representation.

Exploits IBM z/Architecture for application performance improvements

Additional exploitation of the hardware is implemented in the compiler in order to improve performance of the generated code:

• The ARCH option controls the code generated by adjusting the instructions, scheduling, and other optimizations for a specific architecture level of the system, Enterprise PL/I for z/OS V41 introduces the new ARCH(9) option to support the latest z/Architecture. The following new facilities are used to produce code targeting the latest z/Architecture:
  • Load/store-on-condition
  • Distinct-operands
• The compiler will now inline code to resolve references to elements of structures with complex REFER if all the elements are byte-aligned. Previously the compiler would have resolved these references through costly library calls.

As mentioned above, zEnterprise™ System hardware exploitation has been implemented in the Enterprise PL/I for z/OS compiler through the addition of the ARCH(9) option. This option enables the use of new instructions from the load/store-on-condition, and the distinct-operands facilities to be exploited in the compiler generated code. This is designed to provide better performing applications tuned for the
Improves SQL preprocessing
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 V4.1, the SQL preprocessor:
- Supports the XREF option
- Always lists the SQL options in effect.

Improves CICS preprocessing
In Enterprise PL/I for z/OS, with the integrated CICS preprocessor, it is not necessary to run a separate job step that precompiles EXEC CICS statements into PL/I code. Instead, the compile step will handle EXEC CICS 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 CICS precompiler produced).

Leverages productivity with new options
In Enterprise PL/I for z/OS V4.1, the compiler now allows initialization of typed structures. In particular, the INIT attribute will be allowed on leaf elements of a DEFINE STRUCTURE statement, and the new VALUE type-function can then be used to initialize or assign to a variable having that structure type.

In this release, the rules option provides more control over code:
- NOGLOBALDO flags loops with a control variable declared in a parent block
- NOPADDING flags structures with padding

The compiler now flags:
- OPTIONS(MAIN COBOL/FORTRAN).
- All assignments to REFER objects.
- PROC(REENTRANT) when neither RENT nor DFT(NONASGN) is in effect.
- Through the new DEPRECAT E option, the usage of user specified built-in functions, user variable, or include files. Using this option, you can more easily enforce internal quality and naming standards.

As well with the 4.1 release, Enterprise PL/I for z/OS:
- Allows the RESPECT(DATE) option to be overwritten
- Accepts static label arrays declared without emitting a warning message, if the arrays are declared as NONASGN

The SAA/SAA2 suboption of the LANGlvl option has been removed since the older `SAA` compilers are no longer in service.

The (NO)STORAGE suboption of the CHECK option has been removed since LE (Language Environment®) provides the equivalent function more dynamically.

Provides compatibility for PL/I programs and Java components
Because it supports the Institute of Electrical and Electronics Engineers.
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, shown here, supports Enterprise PL/I and helps improve the productivity of PL/I developers. The windows displayed 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.

(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.
### Summary of features and benefits

<table>
<thead>
<tr>
<th>Feature</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designed for IBM platforms</td>
<td>• Exploits z/Architecture®.</td>
</tr>
<tr>
<td><strong>Maximizes middleware</strong></td>
<td>• Exploits the latest middleware (DB2, CICS, IMS) to facilitate application integration and modernization.</td>
</tr>
<tr>
<td>Modernize applications</td>
<td>• Easy to extend your applications on a modern web infrastructure.</td>
</tr>
</tbody>
</table>
| Improved debug information | • Reduces the size of the generated object by placing the statement number in the side file.  
• Generates more specific information for the Debug Tool automonitor function.  
• Includes information in the side file identifying the source files for declares, references and assignments.  
• Generates information to identify the implicit locator reference when a variable is BASED on the ADDR of an array element. |
| Improved SQL and CICS preprocessing | • With the integrated SQL preprocessor or CICS preprocessor, it is not necessary to run a separate job step that precompiles EXEC SQL statements or EXEC CICS statements into PL/I code. Instead, the compile step will handle EXEC SQL statements or CICS 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 or CICS precompiler produced). |
| Exploits hardware support for IEEE 754 decimal floating-point data | • Improves the accuracy and performance of decimal floating-point calculations for commercial applications. |
| 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. |
| Integrated development environment¹ | • Rational Developer for System z boosts developer productivity by making it easy to edit, compile, and debug PL/I applications from your workstation. |
| Collaborative team environment² | • Rational Team Concert for System z unifies development teams by making it easy to manage your distributed software projects and teams. |
| IBM service and support | • Provides responsive platform and cross-platform support that meets or exceeds customer expectations.  
• Teams with subject matter experts in compiler development for dedicated support excellence. |

¹ Separate product: IBM Rational Developer for System z
² Separate product: IBM Rational Team Concert for System z
Upgrade now!
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 how IBM Enterprise PL/I for z/OS, Version 4.1 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/

To learn more about IBM Rational Developer for System z software, visit: ibm.com/software/awdtools/rdz

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Software</th>
<th>Compiler</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS</td>
<td>V1.12</td>
<td>• Enterprise PL/I for z/OS V4.1</td>
</tr>
</tbody>
</table>

© Copyright IBM Corporation 2010
IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
09-10
All Rights Reserved

CICS, DB2, IBM, the IBM logo, IMS, Language Environment, Rational, System z, WebSphere, z/Architecture, zEnterprise, 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, Inc. 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.