



IBM Enterprise COBOL for z/OS, Version 4.1

Highlights

- ***Supports Java interoperability with COBOL programs through object-oriented syntax***
- ***Provides enhanced XML support that enables you to share components as services***
- ***Enables application internationalization, international data processing and sharing string data with Java via support for Unicode***
- ***Includes support for and exploitation of IBM DB2 for z/OS, Version 9 software***
- ***Delivers enhanced support for IBM Debug Tool, Version 8.1 software***
- ***Facilitates IBM WebSphere Application Server interoperability within batch and IBM z/OS environments***
- ***Includes compatibility with the enhanced COBOL XML processing tools of IBM Rational Developer for System z software to improve support for remote development, compiling, error identification and debugging***

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 realtime.

Unfortunately, many IT systems weren't designed to address these objectives or to support the Web services and service-oriented architectures (SOAs) that are essential for transforming your enterprise into a flexible business with an open, integrated operating environment. But rewriting your existing applications to bring them in line with today's standards can be expensive and risky—and it could potentially create downtime that you just can't afford. To remain competitive, you need a complete business integration 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.

Integrating COBOL applications with Web services

With IBM Enterprise COBOL for IBM z/OS®, Version 4.1 software, you can take advantage of more than 30 years of IBM experience in application development to integrate COBOL and Web-based business processes in Web services, XML, Java™ and COBOL applications. Such interoperability enables you to capitalize on existing IT investments, while smoothly incorporating new, Web-based applications into your organization's infrastructure.

Supporting XML-based documents

Enterprise COBOL for z/OS, Version 4.1 provides set-level operations between XML documents—represented as character strings—and COBOL groups or elementary data items. Besides handling input XML messages (through `XML PARSE`), the software can also handle output XML messages. The `XML GENERATE` statement creates XML directly from a group data item, eliminating the need to use `MOVE CORRESPONDING` and `STRING`

statements to build XML messages. This function makes it easier to support outbound XML-based messaging. Enterprise COBOL for z/OS also includes a high-speed parser that enables your COBOL programs to:

- *Process inbound XML documents in principal run-time environments, such as IBM CICS®, IBM IMS™ and IBM WebSphere® MQ*
- *Populate COBOL data structures with the content of XML documents. For example, in a business-to-business (B2B) environment, XML support enhances your existing high-performance IMS transactions written in COBOL by accepting XML data as input. XML data can therefore be placed in and retrieved from the IMS messages queue.*

Enterprise COBOL for z/OS, Version 4.1 also provides additional enhancements to support XML, including:

- *An extended XML GENERATE statement with new syntax that gives you more flexibility and control over the form of the XML document you generate. The statement also supports generation of XML documents encoded in UTF-8 Unicode.*

- *The WITH ATTRIBUTES phrase, which enables eligible items in the XML document to be generated as XML attributes instead of elements.*
- *The WITH ENCODING phrase, which allows you to specify the encoding of the generated document.*
- *NAMESPACE and NAMESPACE-PREFIX phrases, which allow you to generate XML documents that use XML namespaces.*
- *The WITH XML-DECLARATION phrase, which includes the version and encoding information in the generated document.*

The XML PARSE statement has been enhanced as well. Enterprise COBOL for z/OS now supports the z/OS System Services XML parser as an alternative to the existing XML parser included in the COBOL library. The z/OS System Services XML parser:

- *Makes the latest IBM parsing technology available to COBOL users.*
- *Offloads COBOL XML parsing to IBM System z™ Application Assist Processor (zAAP) specialty processors.*
- *Offers improved support for parsing XML documents that use XML namespaces.*

- *Includes direct support for parsing XML documents that are encoded in UTF-8 Unicode.*
- *Supports parsing of very large XML documents, one buffer of XML at a time.*

Four new special registers are now available for namespace processing during execution of XML PARSE statements:

- *The XML PARSE statement has been extended with new syntax WITH ENCODING and RETURNING NATIONAL that gives you control over the assumed encoding of input XML documents, to facilitate parsing in Unicode.*
- *A new compiler option, XML PARSE, enables you to control whether the z/OS System Services parser of the existing COBOL parser is used for XML PARSE statements. With the XMLPARSE(COMPAT) option, XML parsing is fully compatible with Enterprise COBOL for z/OS, Version 3 software. With the XMLPARSE(XMLSS) option, you can use the z/OS System Services parser with new XML parsing capabilities.*

Unicode support

Enterprise COBOL for z/OS provides Unicode support that enables you to internationalize your applications and process international data. Specifically, this release enables a broad use of Unicode data for COBOL applications, including the following types of data encoded in Unicode:

- *External decimal (national decimal)*
- *External floating point (national floating point)*
- *Numeric-edited of USAGE NATIONAL*
- *National-edited*

This release supports data encoded in Unicode (USAGE NATIONAL) that is analogous to the COBOL language and semantics for data encoded in a traditional Extended Binary Coded Decimal Interchange Code (EBCDIC) code page (USAGE DISPLAY). It is therefore possible to update existing EBCDIC applications (with any copy-books) by replacing the picture symbol X or A with the picture symbol N. It also enables you to replace USAGE DISPLAY with USAGE NATIONAL (explicitly or implicitly). By adding the GROUP-USAGE NATIONAL clause to some declarations of groups that contain only USAGE NATIONAL data, you can minimize required changes to procedural code.

You can simplify the programming of applications by using the Unicode support included in Enterprise COBOL and, optionally, the support provided by the IBM DB2® data server. In most cases, when using the DB2 integrated coprocessor, the code pages for host variables are handled implicitly, and it is not necessary to specify coded character set identifiers (CCSIDs) explicitly in structured query language (SQL) DECLARE VARIABLE statements.

Support for IBM DB2 9 information management software

Enterprise COBOL for z/OS takes advantage of SQL support available with IBM DB2 9 information management software. Using DB2 coprocessor support (SQL compiler option), you can use DB2 SQL features, such as multiple-row FETCH, multiple-row INSERT and longer names for SQL identifiers, and table and column names.

This release of Enterprise COBOL for z/OS features enhanced DB2 support, including:

- *DB2 9 exploitation.*
- *Improvements in coprocessor integration and usability.*
- *Support for new SQL data types and new SQL syntax provided by DB2 9.*

- *DB2 options shown in the compiler listing (DB2 9 only).*
- *Expanded SQL communications area (SQLCA) and SQL descriptor area (SQLDA) control blocks in the compiler listing (all DB2 releases).*
- *A new compiler option SQLCCSID to coordinate the CCSID between COBOL and DB2.*

Support for IBM Debug Tool for z/OS, Version 8.1 software

A new suboption of the TEST compiler option enables support for the IBM Debug Tool GOTO and JUMPTO commands when debugging optimized production programs.

Enterprise COBOL for z/OS, Version 4.1 supports new features of the Debug Tool, Version 8.1. The previous version of Enterprise COBOL was available with alternate-function or full-function offerings. The alternate-function offering included the COBOL compiler. The full-function offering included both the COBOL compiler and the basic debug capability found in Debug Tool for z/OS. Enterprise COBOL, Version 4.1 provides the COBOL compiler only. It is not available with a full-function offering. The latest version of the basic debug capability can be obtained in Debug Tool for z/OS, Version 8.1, which can be ordered separately.

As an alternative to Debug Tool, Version 8.1, Debug Tool Utilities and Advanced Functions for z/OS, Version 8.1 includes the basic functions in Debug Tool, Version 8.1 and delivers additional advanced debugging capability to help maximize the availability of z/OS applications. Debug Tool Utilities and Advanced Functions for z/OS gives you the debugging capability for applications written in COBOL, PL/I, C/C++ and Assembler that run in a variety of environments, such as batch, TSO, CICS, IMS, DB2 and UNIX® System Services. It offers more productivity enhancements when used with IBM Rational® Developer for System z software to debug applications from the workstation using remote debug capabilities. It also includes tools to help identify old OS/Microsoft Visual Studio and Visual Studio COBOL II source code; to help upgrade the code to Enterprise COBOL; to provide the tools you need in order to determine how thoroughly your code has been tested (code coverage).

Debug programs that use national data

When national data is displayed in a formatted dump or using the Debug Tool LIST command, the data can be automatically converted to an EBCDIC representation using the code page specified in the CODEPAGE compiler

option. You can use the Debug Tool MOVE command to assign values to national data items, and you can move national data items to or from group data items. You can also use national data as an operand in Debug Tool conditional commands, such as IF or EVALUATE.

Debug object-oriented COBOL syntax

Object-oriented COBOL syntax has been retargeted for Java technology-based object-oriented programming to improve interoperation between COBOL and Java languages.

Separate debugging information from the load module to improve manageability

The Debug Tool support also enables you to debug COBOL programs that have been compiled with the OPTIMIZE and the TEST (NONE, SYM, xxx) compiler options, enabling you to set breakpoints, inspect the value of variables, step through a program and run the program until it reaches the next breakpoint. If IBM Debug Tool Utilities and Advanced Functions, Version 8.1 software is installed with Debug Tool, you can also use the playback function to record and replay application execution paths and data values. Alternately, you can use the automonitor function to automatically display variables referenced in the current statement in the monitor window.

Additional interoperability

Enterprise COBOL for z/OS also facilitates interoperability with a number of other technologies.

Object-oriented syntax for Java interoperability

Enterprise COBOL supports Java technology-based, object-oriented syntax to facilitate the interoperation of COBOL and Java programs. The syntax is based on the facilities of the Java Native Interface (JNI), the primary means that Java provides for interoperating with non-Java programs. Object-oriented COBOL syntax is designed to enable you to write COBOL programs that:

- *Define classes, with methods and data implemented in COBOL.*
- *Create object instances of Java or COBOL classes.*
- *Invoke methods on Java or COBOL objects.*
- *Write classes that inherit from Java classes or from other COBOL classes.*
- *Define and invoke overloaded methods.*

A special register and copybook enable you to easily call services that the JNI provides. These services include Unicode standard–based and EBCDIC language–based services for handling strings and for managing local and global object references. Enterprise COBOL provides full support for the `OPTIMIZE` compiler option for programs that contain object-oriented syntax for Java interoperability.

Object arrays as method arguments

Enterprise COBOL for z/OS supports the use of Java object arrays. Object references of type `javaObjectArray` can go between COBOL and Java languages and can be specified as:

- *Arguments in* `INVOKE...USING`.
- *Values in* `INVOKE...RETURNING`.
- *Parameters on the* `PROCEDURE DIVISION...USING phrase-of-method definitions`.
- *Values on method definitions in* `PROCEDURE DIVISION...RETURNING`.

COBOL class definition with main method

In Enterprise COBOL for z/OS software, an application that uses object-oriented COBOL syntax can start with a COBOL class definition. You can run the application by using the Java command and pass string values as command-line arguments. You can use this method to start an application in an environment where you expect to start applications by using the Java command, and in other environments (such as IMS Java regions) that require applications to start with the main method of a Java class file.

Implementation of object-oriented COBOL programs from batch JCL

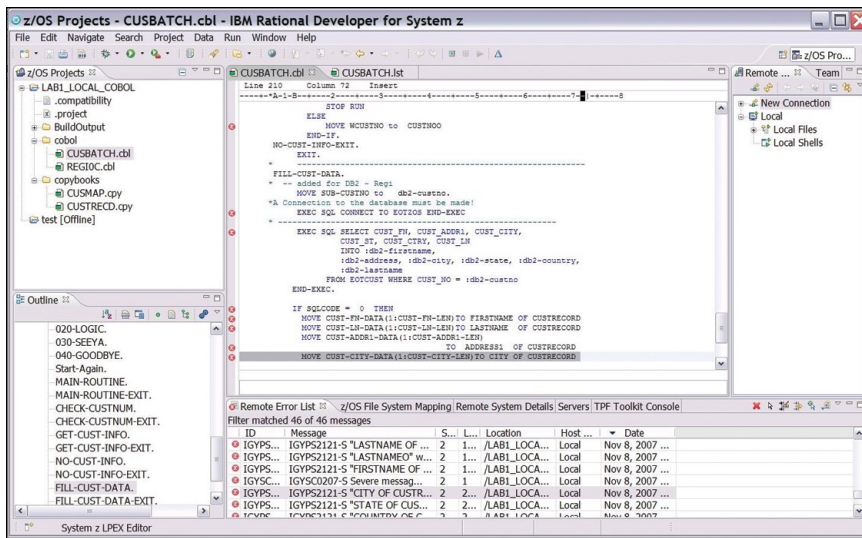
In certain cases, you can bind COBOL programs that use object-oriented syntax as modules in partitioned data set extended (PDSE) data sets and run them by using batch job control language (JCL) statements.

Smooth interoperability between Java and COBOL languages in IMS

Enterprise COBOL for z/OS supports interoperation between the COBOL and Java languages when an application runs in an IMS Java language–dependent region. This support enables you to call a COBOL routine from an IMS Java application. You can use Java to develop the messaging portion of an application and call COBOL to access IMS data sources. You can also develop object-oriented COBOL applications that contain a main routine capable of invoking Java routines.

IBM WebSphere Application Server interoperability

You can use the Java interoperability capabilities of COBOL to access enterprise beans that run on a Java Platform, Enterprise Edition (Java EE) technology–compliant Enterprise JavaBeans (EJB) server, such as IBM WebSphere Application Server software. To do this, the client environment must support a Java



Tools such as Rational Developer for System z support Enterprise COBOL for z/OS and can help improve the productivity of your COBOL developers.

technology-based object request broker (ORB). The client COBOL application can use COBOL INVOKE statements to access the following programming interfaces:

- *Java Naming and Directory Interface (JNDI) to locate EJB services and components*
- *Java ORB interface to invoke methods on enterprise beans*

Enhanced COBOL support for enterprise beans in IBM WebSphere Application Server for z/OS software

Enterprise COBOL for z/OS, Version 4.1 includes enhanced run-time support and initial development tools in Rational Developer for System z software that allow you to load COBOL programs dynamically into WebSphere Application Server for z/OS regions. The programs can be accessed from enterprise beans running in those regions, enabling the business logic in the COBOL program to be accessed in the same way as any other

enterprise bean. Enterprise COBOL also enables you to manage DB2 connections, so the COBOL program can access DB2 databases using EXEC SQL statements.

Delivering new functionality in version 4.1

Additional enhancements in Enterprise COBOL for z/OS, Version 4.1 include the following:

- *Enhanced performance of COBOL application programs by exploiting new IBM z/Architecture® instructions. The performance of COBOL Unicode support (USAGE NATIONAL data) has been significantly improved.*
- *Compiler options can now be specified in a data set.*
- *When the XREF compiler option is used, the compiler listing includes cross-references of COPY statements, libraries and data sets.*
- *DFSMS large-format data sets are now supported.*
- *Internal compiler limits have been increased to allow you to compile very large and complex COBOL source programs.*

Ease into migration

Enterprise COBOL for z/OS gives you a migration path from OS/Visual Studio COBOL, Visual Studio COBOL II, IBM COBOL for MVS™ and VM, and IBM COBOL for IBM OS/390® and VM. With the exception of OS/Visual Studio COBOL programs and any programs that were previously compiled with the CMPR2 compiler option, your current programs can continue to compile and run without modification, while you selectively update existing applications to take advantage of new functions.

You can convert OS/Visual Studio COBOL programs and any programs that were previously compiled with the CMPR2 compiler option into 1985 Standard programs, which can then be compiled with Enterprise COBOL for z/OS. The Debug Tool Utilities and Advanced Functions application has a load module analyzer that can help identify which of your programs were compiled with the OS/Visual Studio COBOL compiler. You can also use the COBOL conversion option or the CICS command-level conversion capability included in Debug Tool Utilities and Advanced Functions to convert your programs.

Workstation-based development

Rational Developer for System z software provides an interactive, workstation-based environment to help you create, maintain and reuse applications for traditional processing or for inclusion in an SOA. It includes support for traditional development using COBOL or CICS, but also has the ability to generate XML from COBOL constructs to help integrate existing COBOL applications into a Web services environment. Rational Developer for System z software can help developers who continue to maintain existing COBOL applications as well as those who may be working on integrating existing COBOL applications into an SOA.

In addition to providing a workstation interface to the Debug Tool, Rational Developer for System z software also integrates with IBM File Manager and Fault Analyzer tools. 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 from the workbench. By integrating with Fault Analyzer tools, Rational Developer for System z software enables you to browse Fault Analyzer ABEND reports on CICS, IMS, batch, Java, WebSphere and other run times from the workbench interface.

COBOL across platforms

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

You can also take advantage of IBM's extensive suite of COBOL maintenance tools to improve your existing applications. These tools assist you with source-code conversion from former ANSI standards to ANSI 85 COBOL syntax, code analysis and reporting, CICS source conversion, Report Writer code support and regression testing for interactive applications.



For more information

To learn more about IBM Enterprise COBOL for z/OS, Version 4.1 software, contact your IBM representative or IBM Business Partner, or visit:

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

© Copyright IBM Corporation 2008

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

Produced in the United States of America
01-08
All Rights Reserved

CICS, DB2, IBM, the IBM logo, IMS, MVS, OS/390, Rational, System z, WebSphere, z/Architecture 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.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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.