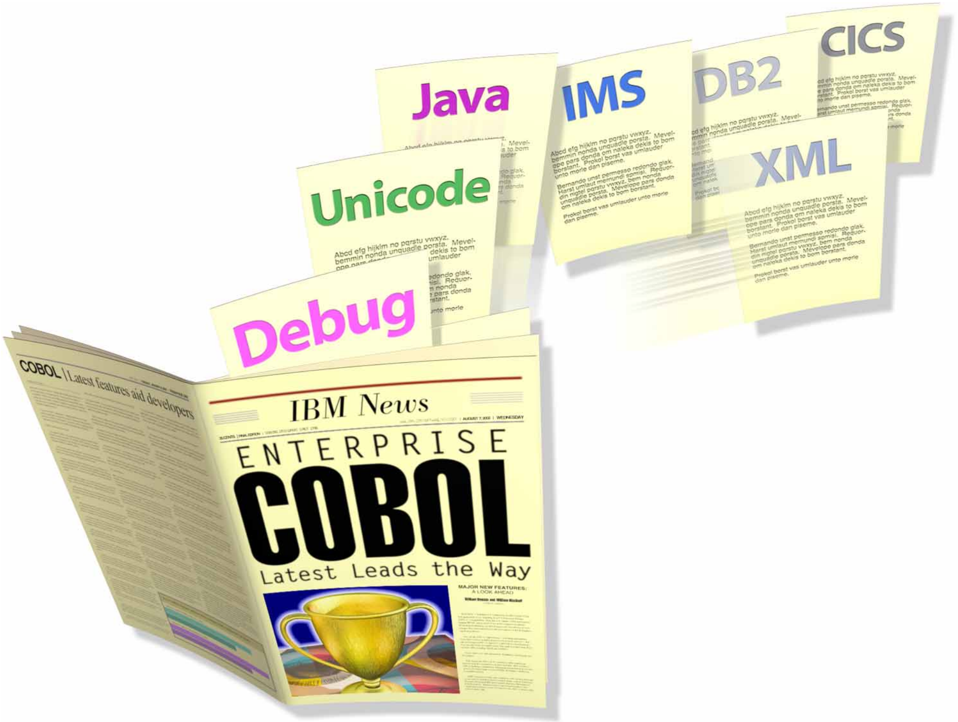
Application development for today's changing marketplace

IBM® Enterprise COBOL for z/OS™ and OS/390® Version 3 Release 2 (5655-G53)



Highlights

IBM Enterprise COBOL for z/OS and OS/390 provides the COBOL functions that you need to integrate COBOL applications with Web-oriented business processes. Enterprise COBOL retains all the function of earlier releases, such as:

- Java™ interoperability, supported by object-oriented syntax, provides access to, and the reuse of, Java business and data objects from USS, batch, and now IMS® environments.
- XML support enables business messages to be delivered in XML and processed by COBOL.
- Unicode™ support enables the internationalization of applications, the processing of international data, and the sharing of string data with Java.
- WebSphere® interoperability provides access from batch and USS environments to Java processing in WebSphere.
- Compatibility with WebSphere tools, including WebSphere Studio Enterprise Developer, supports remote development, compiling, error identification, debugging, and enhanced COBOL XML processing tools. The XML tools include automatic generation of high-speed adapter modules for parsing inbound XML documents and populating a COBOL data structure, and for generation of outbound XML from a COBOL data structure.

Enterprise COBOL delivers the following new function:

- Java-COBOL interoperability in IMS
- Support for Debug Tool Utilities and Advanced Functions
- Enhancements to object-oriented syntax for Java interoperability
- Enhanced support for Unicode in DB2® COBOL applications
- Parameterized initialization of the Java virtual machine (JVM)
- Support for object arrays as method arguments
- Support for COBOL class definition with a main method
- Execution of object-oriented COBOL programs from batch JCL

With the capabilities of this release, developers can do the following tasks:

- Simplify the componentization of COBOL programs and enable interoperability with Java components across distributed applications.
- Promote the exchange and use of data in standardized formats, including XML and Unicode.
- Facilitate the reuse of existing applications in WebSphere and in traditional z/OS and OS/390 environments.
- Enable interoperability with Java when an application runs in an IMS Java dependent region.

- Utilize new debugging functions in Debug Tool.

With Enterprise COBOL, COBOL and Java applications can interoperate smoothly in the e-business world. As a result of such interoperation, software developers can leverage more than 30 years' worth of applications in new endeavors, capitalizing on existing investments.

Java-COBOL interoperability in IMS

Enterprise COBOL supports interoperation between the COBOL and Java languages when an application runs in an IMS Java dependent region. In particular, 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.
- Develop object-oriented COBOL applications that contain a main routine capable of invoking Java routines.

Enhanced support for Debug Tool, and support for Debug Tool Utilities and Advanced Functions

Enterprise COBOL supports new features of IBM Debug Tool and of IBM Debug Tool Utilities and Advanced Functions for z/OS and OS/390. Debug Tool Utilities and Advanced Functions is purchased separately from Debug Tool.

The following new features of Debug Tool are supported:

- Support for debugging optimized code - you can debug COBOL programs that have been compiled with the `OPTIMIZE` and the `TEST(NONE,SYM,`*`xxx`*`)`compiler options. You can:
  - Set breakpoints.
  - Inspect the value of variables.
  - Step through a program one statement at a time.
  - Run the program until it reaches the next breakpoint.
- The Debug Tool `GOTO` command is enabled for COBOL programs that have been compiled with the `NOOPTIMIZE` option and any variant of the `TEST` option.
- Condition names and file names may be specified in `LIST` commands.

In addition, if Debug Tool Utilities and Advanced Functions is installed with Debug Tool, you can benefit from the following new function:

- Playback - you can record and replay application execution paths and data values.
- Automonitor - variables that are referenced in the current statement will automatically be displayed in the monitor window.

Enhancements to object-oriented syntax for Java interoperability

Enterprise COBOL supports Java-based object-oriented syntax to facilitate the interoperation of COBOL and Java programs. The syntax is based upon the facilities of the Java Native Interface, 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 let you easily call services that the Java Native Interface provides. These services include Unicode- and EBCDIC-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.

Using Unicode in DB2 COBOL applications

Enterprise COBOL simplifies the programming of applications using the Unicode support provided by COBOL and DB2. 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 CCSIDs explicitly in SQL DECLARE VARIABLE statements.

Parameterized initialization of JVM

The COBOL run time automatically initializes the Java virtual machine as needed for COBOL applications that use object-oriented syntax for Java interoperability. Enterprise COBOL introduces a new environment variable, COBJVMINITOPTIONS, which lets you specify options that you want the COBOL run-time environment to pass to the JVM.

Object arrays as method arguments

Enterprise COBOL supports the use of Java object arrays. Object references of type jobjectArray can go between COBOL and Java 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 an application that uses object-oriented COBOL syntax can start with a COBOL class definition. The application can be run by using the `java` command, and can pass string values as command-line arguments. This method of starting an application can be used in environments where applications are expected to be started 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.

Execution of object-oriented COBOL programs from batch JCL

In certain cases, you can bind COBOL programs that use object-oriented syntax as modules in PDSE data sets and run them by using batch JCL statements.

XML support

Enterprise COBOL introduces XML capabilities to COBOL. The support includes a high-speed parser that enables your COBOL programs to:

- Process XML documents in the principal run-time environments, such as CICS, IMS, and MQSeries®.
- Populate COBOL data structures with the content of XML documents.

For example, in a business-to-business environment, XML support enhances your existing high-performance IMS transactions written in COBOL, by accepting XML data as input. XML data can be placed in, and retrieved from, the IMS messages queue.

WebSphere interoperability

You can use the Java interoperability capabilities of COBOL to access enterprise beans that run on a J2EE-compliant EJB server, such as WebSphere Application Server. To do this, the client environment must support a Java-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 to invoke methods on enterprise beans

Improved application development

Enterprise COBOL for z/OS and OS/390 provides a set of intrinsic functions including string handling, financial capabilities, statistical functions, and mathematical functions. You can also use the COBOL CALL statement to take advantage of Language Environment services for everything from storage management to condition handling. The condition-handling support enables you to write programs in which error handling is done in a separate routine that is loaded only when needed. You do not have to write the error-handling routines in assembler; with Language Environment you can write them in COBOL.

Enterprise COBOL for z/OS and OS/390 offers support for recursive calls in COBOL, structured programming, improved interoperability with other languages, and dynamic link library support. The Enterprise COBOL for z/OS and OS/390 run-time library, Language Environment, also supports PL/I, C/C++, and Fortran programs.

Ease into migration

Enterprise COBOL for z/OS and OS/390 gives you a migration path from OS/VS COBOL, VS COBOL II, IBM COBOL for MVS & VM, and IBM COBOL for OS/390 & VM. Except for OS/VS COBOL programs and any programs previously compiled with the CMPR2 compiler option, your current programs will continue to compile and to run without modification, while you selectively update existing applications to take advantage of the new functions. For OS/VS COBOL programs and any programs previously compiled with the CMPR2 compiler option, convert them into 1985 Standard programs that can be compiled with Enterprise COBOL. Use either the optional product COBOL and CICS Command-level Conversion Aid, 5648-B05, or use the COBOL conversion option included in Debug Tool Utilities and Advanced Functions, 5655-J18. The CICS Command-level Conversion capability is also part of the Debug Tool Utilities and Advanced Functions.

COBOL across platforms

Enterprise COBOL for z/OS and OS/390 is part of a large family of compatible compilers, application development tools, and maintenance tools. In addition to Enterprise COBOL, IBM offers IBM COBOL compilers for Windows®, AIX®, VSE®, and AS/400®. Host-based development products include ISPF, File Manager, Fault Analyzer, and Debug Tool. 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 of interactive applications.

Software prerequisites

Enterprise COBOL and its generated object programs run under the following S/390 operating systems:

- z/OS, Version 1 Release 1 (5694-A01) or later
- OS/390, Version 2 Release 10 (5647-A01)

VM/CMS is not supported.

The following Language Environment elements provide the execution environment and library of COBOL run-time services that are required to compile and run COBOL applications using Enterprise COBOL:

- On z/OS, Version 1 Release 1: z/OS Language Environment element plus PTFs for APAR PQ62947 and PQ52626
- On z/OS, Version 1 Release 2: z/OS Language Environment element plus PTFs for APAR PQ65174 and PQ52626
- On z/OS, Version 1 Release 3: z/OS Language Environment element plus PTFs for APAR PQ65175
- On z/OS, Version 1 Release 4: z/OS Language Environment element plus PTFs for APAR PQ65176
- On OS/390, Version 2 Release 10: OS/390 Language Environment element plus PTFs for APAR PQ62947 and PQ52626

Support for object-oriented COBOL syntax (Java interoperability) requires:

- IBM Developer Kit for OS/390, Java 2 Technology Edition, SDK 1.3.1 or later

For installation on z/OS:

- z/OS SMP/E element
- SMP/E, Version 8 Release 1 (5668-949), if you are using SMP/E to maintain your z/OS products

For installation on OS/390:

- OS/390 SMP/E element
- SMP/E, Version 8 Release 1 (5668-949), if you are using SMP/E to maintain your OS/390 products

One of the following assemblers is required for customization:

- z/OS High Level Assembler
- OS/390 High Level Assembler

Support for Unicode requires:

- For z/OS: z/OS Support for Unicode (5694-A01)
- For OS/390: OS/390 Support for Unicode (5647-A01)

Support for CICS requires:

- CICS Transaction Server for OS/390, Version 1 Release 3 or later
- Support for integrated CICS translator

requires CICS Transaction Server, Version 2 Release 1 or later

The following prerequisites are needed for DB2:

- DB2 for OS/390, Version 6 (5645-DB2) or later
- Support for DB2 integrated coprocessor requires DB2 Universal Database, Version 7 (5675-DB2) or later
- Use of Unicode in DB2 COBOL applications requires DB2 APAR PQ61320

One of the following prerequisites is needed for IMS:

- IMS, Version 7 with PTFs for APAR PQ53944 and PQ54039
- IMS, Version 8

The following prerequisite is needed for interactive debugging:

- Debug Tool for z/OS and OS/390, Version 3 Release 1 (5655-H32)

For more information

See your client representative or call IBM DIRECT at 1-800-IBM-CALL in the US and Canada. To learn more about these tools, visit the COBOL Web site at www.ibm.com/software/ad/cobol.

IBM Corporation
Marketing Communications, Servers
Route 100
Somers, NY 10589

Printed in United States of America, 08/2002