



isCOBOL™ Application Platform Suite

isCOBOL APS 2009 Release Overview

Copyright © 2009 Veryant.
2415 East Camelback Road, Suite 700, Phoenix, AZ 85016, U.S.A.

All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution and recompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Veryant and its licensors, if any.

isCOBOL is a trademark or registered trademark of Veryant in the U.S. and other countries. All other marks are property of their respective owners.

Table of Contents

Introduction	4
COBOL and Java Integration	4
Accessing COBOL data from Java	4
Calling COBOL programs from Java	5
Javadoc comments	6
isCOBOL	6
Screen Designer	7
Integrated Event Editor	8
Import Facility	8
Thin client performance	8
Transitioning to isCOBOL APS	8
Additional enhancements	10

isCOBOL APS 2009 Release Overview

Introduction

Veryant is proud to announce the newest version of its isCOBOL™ Application Platform Suite (APS), isCOBOL APS 2009. isCOBOL APS provides a complete environment for the development, deployment, maintenance, and modernization of COBOL applications. The latest release of isCOBOL APS enhances COBOL and Java integration, expands the capabilities of the Eclipse-based Integrated Development Environment (IDE), and offers several improvements that make transitioning to isCOBOL from alternative COBOL platforms even easier.

Details on key features of the 2009 release are included below.

COBOL and Java Integration

By blending the best of Java and COBOL, isCOBOL APS enables developers to add the power and flexibility of the Java platform to existing COBOL programs and make COBOL investments easily accessible to the Java world.

Accessing COBOL data from Java

A new utility known as COBFILEIO has been added in the isCOBOL APS 2009 release. The COBFILEIO utility reads COBOL source code and automatically generates Java classes to access COBOL files and records -- eliminating the time-consuming requirement of writing individual interface programs or using proprietary middleware to directly access COBOL data from Java programs.

The COBFILEIO utility works by generating Java class interface programs to access each COBOL file and record type in a program. These Object Oriented (OO) COBOL programs are able to open and close data files and do READ/WRITE operations. Javadocs for the file and record classes are automatically generated by the COBFILEIO utility as well.

From the COBOL programmer's perspective, there is no need to know OO programming to use this tool. From the Java programmer's perspective, there is no need to be familiar with COBOL data types or their underlying storage format.

To use this utility, programs are first compiled with the `-efd` option to generate XML files. Next, those XML record structure descriptions are used by the COBFILE IO utility to produce OO programs that directly interact with the COBOL data files and prepare a record to WRITE or READ information.

Java programs can now access individual record fields using Java data types. The record is managed as an object with set/get accessor methods for each elementary field. Because the COBFILEIO utility generates COBOL and Java source code for the classes, this source code becomes part of an application and allows for further customization if desired. This utility allows organizations to seamlessly integrate COBOL records and files into a Java development environment and takes advantage of rapid development features such as "code assist" context sensitive content completion.

The COBFILEIO utility is located in the utility.jar file. Additional documentation on this tool can be found in the isCOBOL APS 2009 User Guide under the Utilities section.

Calling COBOL programs from Java

In the isCOBOL APS 2009 release, Veryant has also simplified the process for Java developers to access COBOL programs and routines. Now calling COBOL programs from Java can be as simple as using the line of code below:

```
returnCode = IsCobol.call("MYCOBOLPROG", args);
```

An example illustrating how to set an environment variable (runtime framework property) and how to cancel a program is as follows:

```
import com.iscobol.java.IsCobol;

public class MyJavaClass {
    public static void main( String args[] ) {
        int rc;
        // Calling a COBOL main program.
        IsCobol.setEnvironment("errors_ok", "2");
        rc = IsCobol.call("MYCOBOLPROGRAM", null);
        IsCobol.cancel("MYCOBOLPROGRAM");
    }
}
```

Javadoc comments

isCOBOL APS has been enhanced in the 2009 release to allow for the inclusion of Javadoc style comments in COBOL source code. Javadoc is a tool in the Java JDK which generates API documentation in HTML format from doc comments placed in source code.

With isCOBOL, Javadoc documentation is created based on special directives and comments inside OO COBOL source code. These directives are `((DOC))` and `((END-DOC))`, defining the start point and the end point of where to write Javadoc comments.

A Javadoc comment is set off from code by standard COBOL comment tags `(*)`. This comment line could contain a HTML snippet or Javadoc tag.

Javadoc tags allow special description parameters such as method `@param`, method returns `@returns` or minor things like the name of developer `@author` and version number of class or method `@version` to be included as comments in source code.

Here is a COBOL Javadoc source code example:

```
IDENTIFICATION DIVISION.  
*(( DOC ))  
* The <code>CatgyRec</code> class represents records in the  
* CatgyFile.  
*  
* @author John Doe  
* @version 1.0  
* @see CatgyFile  
*  
*(( END-DOC ))  
CLASS-ID. CatgyRec as "com.internal.cobolio.CatgyRec".  
ENVIRONMENT DIVISION.
```

Because the Javadoc tool reads java source, java source needs to be generated from the isCOBOL compiler using the `-jj` option to take advantage of this feature.

isCOBOL™ IDE enhancements

Built on Eclipse, the isCOBOL IDE features a COBOL code editor, integrated debugger with remote debugging capability, the ability to debug COBOL and Java together, and all of the other features available to the Eclipse community.

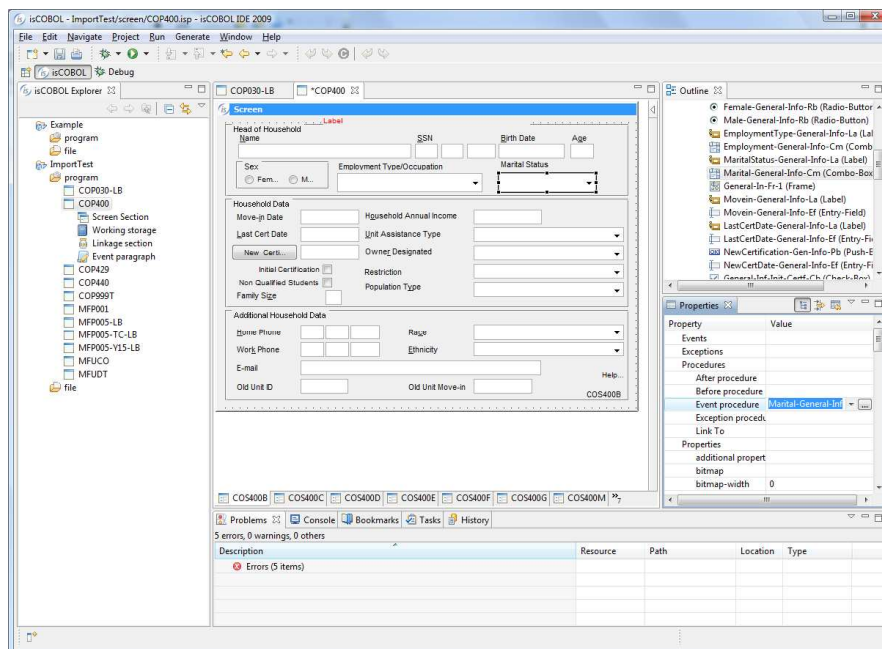
Core COBOL development functions such as COBOL syntax highlighting, compiling, and project handling were included in previous release of the Eclipse-based isCOBOL IDE included with isCOBOL APS 2008 SP1. In the 2009 release, several new productivity enhancements have been added to the IDE including a GUI-based Screen Designer, an integrated Event Editor, and an Import Facility.

Screen Designer

The GUI-based Screen Designer available in the 2009 release of the isCOBOL IDE, is a WYSIWYG drag-and-drop editing tool that lets developers design screens from a graphical "Palette View" repository and drop controls into a screen view. This feature assists with standardization and templates, and enhances productivity by automatically generating back end COBOL source code. Since the isCOBOL Runtime Environment is 100 percent Java, when a developer is finished designing screens, he or she has actually created fully portable Java SWING interfaces.

When working with the Screen Designer, a developer can select a control and the Properties View will display all of its associated properties. Default properties values for each control type can be configured using the isCOBOL Screen Designer Default menu. Once all desired controls have been added to a screen, the Align pop-up menu can be used to fine tune the size and position of each control.

Figure 1. isCOBOL IDE Screen Designer Screen Shot



Integrated Event Editor

An integrated Event Editor is now offered with the isCOBOL IDE to facilitate the connection between screen control design elements and the back end execution events they trigger. Now, as screens are designed, custom source code paragraphs can be easily associated with screen controls. For each structured program, it is also possible to specify the rules to follow in generation tasks, such as generating ANSI or Terminal source format or generate source code in one program.

Import Facility

With the addition of an import facility wizard, existing Totem™ and AcuBench® projects and programs can be brought into the Eclipse-based isCOBOL IDE. When using the Import Facility, an import wizard reads *.psf files and allows a developer to choose programs to be imported.

Other additions to the isCOBOL IDE include a GUI interface for working storage. Now working storage variables can be set through a graphical interface.

Thin client performance

Offered as part of the isCOBOL Runtime Environment, isCOBOL Server delivers powerful thin client and distributed processing.

In the 2009 release of isCOBOL APS, integration between isCOBOL™ Server and isCOBOL™ ISAM Server has been enhanced and overhead has been reduced. Now, when isCOBOL ISAM Server is accessed in thin client mode the need to start a separate isCOBOL ISAM Server process has been eliminated, thus improving overall performance.

Transitioning to isCOBOL APS

isCOBOL APS supports current COBOL language syntaxes and is highly compatible with today's common COBOL dialects. The transition to isCOBOL technology is generally as easy as a version upgrade of your current COBOL development and deployment tools.

Veryant aims to continue simplifying transitions to the isCOBOL platform with every new product release. In isCOBOL APS 2009, compatibility with common COBOL dialects has been further enhanced.

- Several new ACUCOBOL-GT compatibility options have been added to the product including new compiler option such as:
 - o `-zy` to return 8-digits dates for ACCEPT FROM DATE statement
 - o `-vh` to HIGHLIGHT all ACCEPT
 - o `-cax` to specify the file assignment as external
 - o `-smat` to use Mixed formats (ANSI/Terminal) in source and copy files
- New framework properties have been added to the product that also benefit customers moving from ACUCOBOL-GT to isCOBOL, including:
 - o `iscobol.gui.temporary_controls` to change default styles
 - o `iscobol.terminal.cursor_type=n` to change the cursor shape
 - o `iscobol.colormap.default` to map the default character display
- For organizations moving from Micro Focus COBOL to isCOBOL APS, new compiler options have been added to the isCOBOL Compiler to support COBOL dialect specific options such as:
 - o `-cm` to address NEXT SENTENCE, LOCK
 - o `-dcmi` to use numeric format with -C IBMCOMP
- Other items that have been added to better support transitions from Micro Focus COBOL include the added support of:
 - o PERFORM UNTIL EXIT syntax
 - o EVALUATE with IS LESS or GREATER syntax
 - o INITIALIZE ALL TO VALUE
 - o 01 [Constant-name-1] CONSTANT AS ...
 - o \$IF and \$SET directives
 - o Micro Focus file status codes
 - o Arithmetic operations between START OF and LENGTH OF clauses in constant values

Moving to isCOBOL from IBM Mainframe COBOL and RM/COBOL has also been further simplified with three new JUTIL options; `-loadtext` and `-unloadtext` to load and unload

records from and to line sequential files, and -loadr2 to load files created with RM/COBOL's Recover2 program.

Additional enhancements

The following new features or enhancements were also added in the isCOBOL APS 2009 release:

- Full support for MySQL, DB2, and DB2/400 (the AS/400 version of DB2 from IBM), to include functionality such as record locking, optimization, and other database-specific features.
 - o The isCOBOL™ ESQL Generator also can now generate EDBI routines for MySQL, DB2, and DB2/400
- isCOBOL ISAM Server is now also available on Windows 64-bit and Mac OSX.