

Solution overview

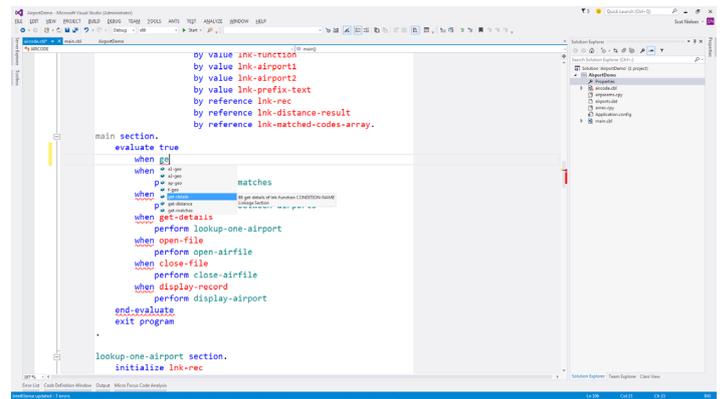
BUSINESS CHALLENGE

There is an increasing demand from users of business software for easier to use applications which integrate with other business systems. As a result IT organizations are being asked to deliver modern user interfaces and integrate multiple business systems. At the same time IT continually strives to reduce operational costs and risk by standardizing on common platforms for all their applications. This way, the same tools and processes can be used regardless of which programming language the IT team built the applications with. Microsoft's .NET framework and the Java Virtual Machine (JVM) are widely seen as the standard frameworks of choice.

Moving to commodity hardware and standard frameworks can return considerable cost savings and business agility. But often, IT organizations must contend with long-standing applications that run on aging or even unsupported hardware and software environments. Previously, IT organizations and ISVs considered that their only option to deliver new features and deploy into the standard frameworks was to rewrite business applications in newer languages, such as Java or C#. This introduces considerable cost and risk for little business value. With Visual COBOL the application can remain in COBOL and the application provider can choose to deploy across a wide range of enterprise platforms including .NET or JVM. This way the organization can retain the benefits of the platform as well as the traditional strength and reliability of existing application investments.

IT organizations must also find ways to transfer skills and resource to support existing application investments, avoiding the pitfalls of siloed application development. By standardizing within the leading integrated development environments (IDEs) – Visual Studio and Eclipse – with advanced COBOL tooling, IT organizations can quickly train new developers already familiar with the tools and processes. This increases business agility and reduces cost.

Fig.1 The future of COBOL application development using industry standard tools



SOLUTION OVERVIEW

Visual COBOL is a suite of software products designed to meet the needs of IT organizations with existing application investments written in the COBOL language.

Advanced COBOL application development tools based around Visual Studio and Eclipse provide developers with a high-end development experience, consistent with that of Java and C# developers, facilitating cross-skilling and agile development.

Patented compiler technology offers flexible deployment options across Windows, UNIX and Linux systems including .NET and JVM platforms. Existing COBOL applications can seamlessly integrate with C# or Java systems to provide backend service delivery.

BUSINESS BENEFIT

Visual COBOL provides IT organizations with the ability to create new customer value from existing application investments. Visual COBOL removes the risk associated with more radical approaches which expose the business to uncertain cost and extended time frames.

By reusing proven application logic to deliver new business requirements, organizations can quickly and safely respond to the needs of modern IT users with predictable and highly cost-effective results.

Solution overview

FEATURE OVERVIEW

- High performance COBOL runtime environment
- Highly compatible upgrade path for existing Micro Focus Net Express and Server Express users
- Application development within Visual Studio or Eclipse
- Application development and deployment across Windows, Unix and Linux platforms, application servers, databases and middleware components
- Direct generation to Java bytecode and Microsoft Intermediate Language (MSIL) for deployment in JVM or .NET platforms
- A modern COBOL language syntax supporting new language constructs for object-oriented development and traditional procedural styles
- Web Services toolkit for building Service Oriented Architectures (SOA) based on existing COBOL applications.

DETAILED FEATURE OVERVIEW

Patented compiler technology

The Visual COBOL compiler includes patented technology that offers unique and highly versatile options for COBOL application development. COBOL programs can be compiled to a variety of executable formats including:

- **Intermediate code (.int)** a Micro Focus platform portable executable format
- **Generated code (.gnt)** a Micro Focus executable format optimized for the target platform
- **Shared Object (.so)** native shared object executable format for UNIX/Linux platforms
- **Windows Executable (.exe/dll)** native Windows executable formats

- **Java bytecode (.class)** COBOL compiled to Java bytecode and executable with the JVM

- **.NET Assembly (.exe/.dll)** COBOL compiled to MSIL and executable within the Microsoft Common Language Runtime (CLR).

The Visual COBOL compiler offers support for a wide variety of modern and older COBOL dialect variants and includes ANSI and ISO standards, Enterprise COBOL and many others.

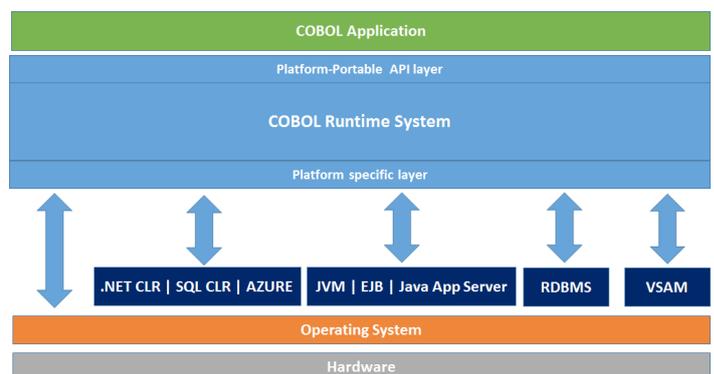
High performance COBOL runtime environment

Micro Focus COBOL Server provides a high-performance, platform-portable runtime environment for the execution of COBOL applications.

Consistent application behaviour is provided across supported platforms and through use of a portable API layer, enabling developers to follow a write-once, deploy-anywhere approach.

COBOL Server provides many services essential to a range of COBOL applications such as file processing, sorting and relational database support. The COBOL Server also provides application tracing and diagnostic tools.

Fig.2 High performance, portable runtime execution environment for COBOL applications

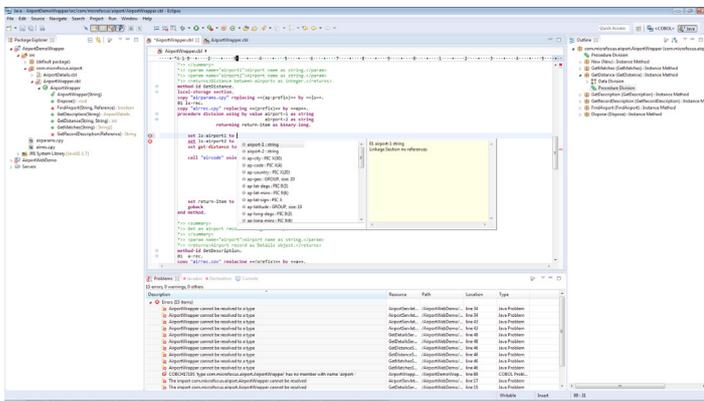


Solution overview

ADVANCED COBOL DEVELOPMENT TOOLS

Visual COBOL extends the Visual Studio and Eclipse IDEs to provide a rich COBOL development environment within the world's most popular application development platforms.

Fig.3 Mixed language COBOL and Java development within Eclipse



Visual COBOL provides advanced editing and debugging features within the IDE:

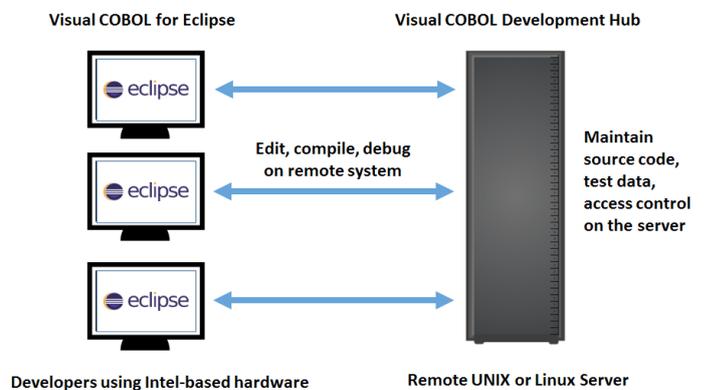
- Continuous background compilation catches syntax errors as they occur providing instant feedback to the developer
- Auto-complete prompts assist the developer when coding language constructs. It automatically offers suggestions and access to framework APIs and documentation
- Advanced debugging tools support JIT, attach-to-process, and remote debug options
- Multiple tool windows provide extra information about the structure and layout of the application

- Inline editor views provide a single window view of a COBOL program, incorporating all copybooks used and the result of COPY REPLACING
- Several code navigation tools assist developers in identifying points of interest in the code, including go to definition, find all references, and navigation bars
- Snippets and templates give developers access to commonly used constructs and can be customized to meet specific developer and development team needs
- Code analysis tools enable developers to create customizable analysis queries that can be executed on demand or following a build. Out-of-the box queries identify performance bottlenecks and dead code.

Remote development for UNIX and Linux

Productivity offered by the Eclipse platform extends to COBOL application development on UNIX platforms. A remote development option enables developers to use Eclipse on Windows or Linux platforms while application source code and data resides on a remote UNIX server. This feature enables teams to retain the traditional client-server approach to UNIX application development, while delivering a seamless development experience within Eclipse. Compilation and debugging occur on the remote server while being initiated and controlled through the Eclipse IDE.

Fig.4 Visual COBOL for Eclipse remote development option



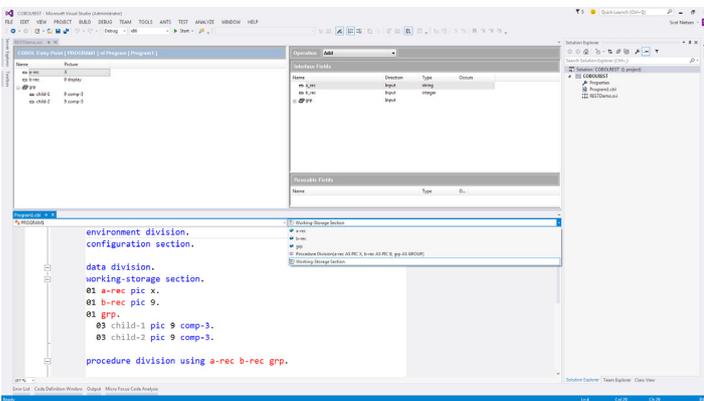
Solution overview

RESTful Web Services

Visual COBOL provides several options for developing applications as part of an SOA. Once compiled for use in .NET or JVM environments, COBOL applications can be invoked within the native framework support for web services.

In addition, Visual COBOL provides a native code, COBOL application server which offers both J2EE application server integration and a Web Services option for SOAP or REST-JSON based web services. A graphical toolset enables developers to quickly create web services from existing COBOL applications. Once created, services can be debugged and deployed within the Micro Focus COBOL Server for SOA, a high-performance COBOL application server environment.

Fig.5 Creating a REST web service using the graphical web services toolkit provided in Visual COBOL



UI modernization

For many organizations, application user experience is often not in keeping with the needs of today's users. Green-screen or command line driven data entry is commonplace and an updated modern look and feel is highly desirable.

Visual COBOL's integration with .NET and JVM provides unique options for overhauling aged user interfaces while preserving existing business logic. Modern UIs can be developed directly in COBOL or combined, employing Java or C#/VB to provide front end functionality, and COBOL to provide backend functions.

Java

Visual COBOL provides several options for creating composite applications which use both COBOL and Java. These include:

- JNI mechanisms using the Micro Focus OO COBOL Java domain
- Java EE using Java Connector Architecture (JCA) for EJB integration
- Direct bytecode generation using COBOL JVM

The COBOL JVM option enables COBOL applications to be compiled to .class files. This enables COBOL applications to run directly within the JVM and become accessible to other JVM languages. COBOL developers can also make use of the Java SDK or invoke functionality implemented by other languages in separate .class files.

Microsoft .NET

Visual COBOL compiler technology enables COBOL applications to be compiled directly to Microsoft Intermediate Language (MSIL) assemblies. This enables COBOL applications to run directly within the Common Language Runtime (CLR) and become accessible to other .NET languages, such as C# or VB. COBOL developers can also make use of .NET framework APIs or invoke functionality implemented by other languages within separate assemblies.



Solution overview

Modern COBOL syntax

The COBOL language has been enhanced to better support the needs of application developers working within .NET and JVM platforms. An extended and lightweight object-oriented syntax is available for creating classes in COBOL or can be used to invoke object-oriented code from regular procedural COBOL.

These extensions also add modern constructs to COBOL programming, such as the ability to declare local fields.

SmartLinkage

When compiling COBOL applications for Java bytecode or MSIL, the Visual COBOL compiler can be instructed to automatically produce wrapper classes which can be used by C#/VB or Java developers. These classes assist other developers in calling existing procedural COBOL applications by implementing routines that hide the complexity involved in converting to and from COBOL type system.

Run units

Application developers and architects can face challenges when reusing procedural COBOL applications in multi-user environments, such as web-based deployments in .NET and JVM. COBOL programs were traditionally architected to isolate user state within separate processes and which conflicts with the web server models in .NET and JVM.

Micro Focus run unit technology solves this challenge by enabling a top-level COBOL program, all data and sub programs to be isolated from other users of the application. Multiple instances of the top-level program can be constructed, each corresponding to individual users, removing the need to re-architecture the existing COBOL applications and smoothing the transition to .NET and JVM platforms.

Data tools

A comprehensive suite of COBOL data tools is provided for the editing and maintenance of COBOL data files. A COBOL file editor supports formatted data editing of COBOL data types and a data file converter, enabling files to be converted from one format to a wide variety of alternatives.

Application rehosting

Visual COBOL is the leading choice for application rehosting projects emanating from a variety of hardware platforms. Visual COBOL integrates with the leading transaction processing monitors and middleware vendors. This network of specialist partners, with decades of skill in application rehosting projects, ensures the most effective solutions are provided for our customers.

Specialised compiler directives are provided to emulate source platform behaviour and include platform-specific functions, such as LE routines.

Data conversion tooling is available and provides support for ASCII and EBCDIC data formats.

PRODUCTS

Application development

- Visual COBOL for Visual Studio
- Visual COBOL for Eclipse
- Visual COBOL Development Hub

Integrated development environments

- Visual Studio 2012, 2013, 2015
- Eclipse 4.2, 4.3, 4.4

Application deployment

- COBOL Server

Solution overview

PLATFORMS

Microsoft Windows

- Windows - 7, 8, 8.1, 10
- Windows Server -2003, 2008 R2, 2012 R2
- Windows XP – deployment only
- Microsoft Azure 2.6 SDK

IBM AIX

- 6.1, 7.1

HP-UX

- 11i v3 (11.31) on Itanium

Solaris

- 10, 11 SPARC
- 11 Intel

SUSE

- 11, 12 on Intel and System z

Red Hat

- 6, 7 on Intel and System z

Oracle Linux

- 6, 7 on Intel
- Red Hat Compatible Kernel
- Oracle Unbreakable Kernel

Java

- 1.6, 1.7, 1.8

Java Application Servers

- Tomcat 7
- JBOSS 7.1
- Oracle WebLogic 12.1
- IBM WebSphere 8.5

.NET

- All supported frameworks versions

Relational Databases

- Oracle 11g R1-R2, 12c
- Microsoft SQL Server 2008 R2, 2012, 2014
- IBM DB2 9.5, 9.7, 10.1,10.5
- Postgres 9.4

Middleware

- Oracle Tuxedo and Tuxedo ART 12.1.3c
- IBM TXSeries for Multiplatforms 8.2

For additional information please visit: www.microfocus.com

© 2015 Micro Focus Limited. All rights reserved. MICRO FOCUS, the Micro Focus logo, among others, are trademarks or registered trademarks of Micro Focus Limited or its subsidiaries or affiliated companies in the United Kingdom, United States and other countries. All other marks are the property of their respective owners.