Brochure

Application Modernization and Connectivity

Visual COBOL

MICRO FOCUS

Business Challenge

There is an increasing demand from users of business software for easier to use applications which flexibly 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 across teams regardless of which programming language was used to build it originally. Microsoft's .NET framework and the Java Virtual Machine (JVM) are widely seen as the standard frameworks of choice.

Moving to commodity hardware and managed code frameworks can return considerable cost savings and better 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 innovation was to rewrite business applications in newer languages, such as Java or C#. This strategy introduces considerable cost and risk for little business value. With Visual COBOL the application's core business logic is re-used enabling the application provider to deploy new functionality across a wide range of enterprise platforms including Linux, .NET, JVM or the Cloud. This strategy enables the organization to take advantage of next generation technology alongside traditional strengths and reliability of proven application investments.

IT organizations must also find ways to quickly onboard or transfer specialty skills and resources to support existing application investments, avoiding the pitfalls of siloed application development. By standardizing within integrated development environments (IDEs)—Visual Studio and Eclipse, IT teams can quickly onboard new developers already familiar with these modern tools and processes enabling them to take on new enterprise application maintenance and development tasks. This approach accelerates onboarding, reduces new hire costs and delivers a future strategy for building next generation talent.

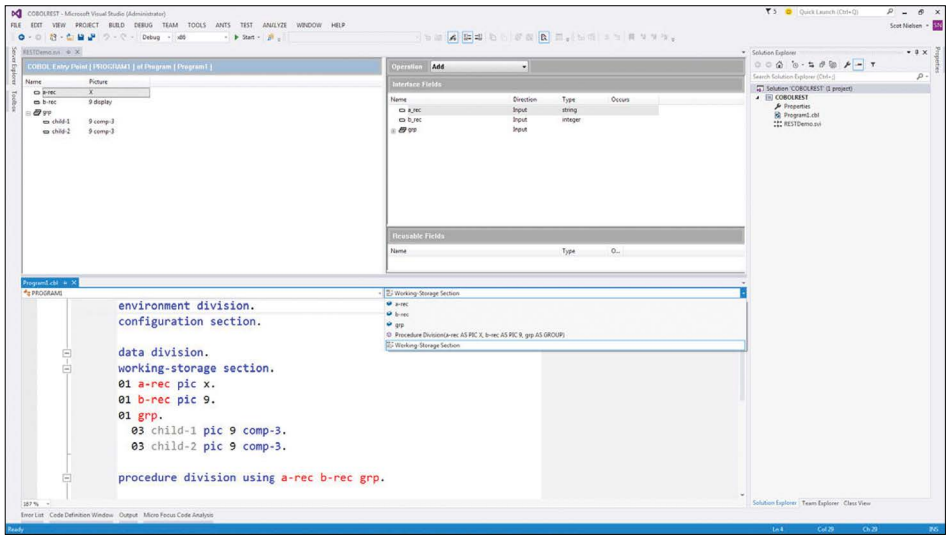

Figure 1. The future of COBOL application development using industry standard tools

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 available within Visual Studio and Eclipse provide developers with a modern development experience, consistent with that of Java and C# developers, facilitating fast, agile development and easy onboarding for new staff.

Patented compiler technology offers flexible deployment options across native platforms including Windows, UNIX and Linux systems as well as managed code environments such as .NET and the Java Virtual Machine (JVM). Existing COBOL applications seamlessly integrate with C# or Java systems enabling faster development and service delivery.

Business Benefit

Visual COBOL provides IT organizations with the ability to create new customer value from existing application investments. By re-using core application logic, Visual COBOL removes the risk associated with re-write or replacement strategies which expose the business to uncertain cost and extended delivery time frames.

With Visual COBOL, organizations can quickly and safely respond to new business requirements and modern IT user needs with predictable and highly cost-effective results.

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 within 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.

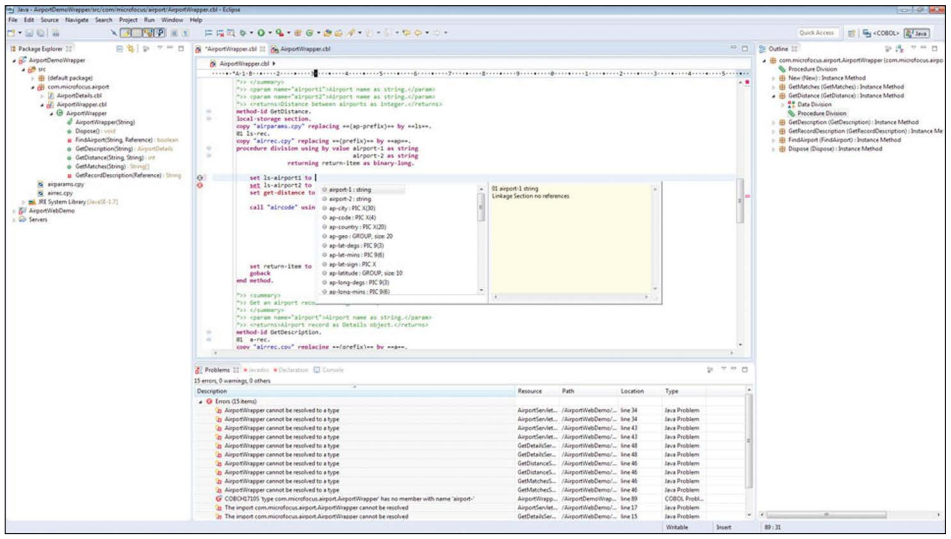

Figure 3. Mixed language COBOL and Java development within Eclipse

High Performance COBOL Runtime Environment

Micro Focus COBOL Server provides a high-performance and platform-portable runtime environment for the execution of COBOL applications. Consistent application behavior is provided across supported platforms and through use of a portable API layer, enabling developers to follow a write-once, deploy-anywhere approach.

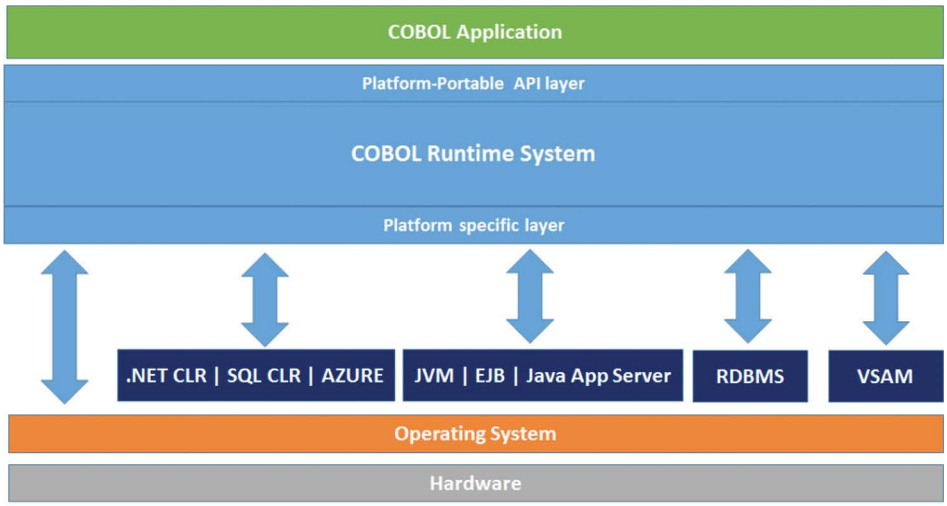

Figure 2. High performance, portable runtime execution environment for COBOL applications

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.

The latest release of Visual COBOL includes new compiler and runtime optimizations which have been shown to increase application performance on Intel-based systems by between 20-30% across a broad range of benchmarks.

Cross-Platform Deployment

The Micro Focus COBOL compiler and COBOL runtime system provides ubiquitous platform coverage enabling COBOL applications to be deployed across a range of distributed systems, ensuring compatibility and consistent behaviour across different platforms. Standard library routines callable from COBOL applications enable application developers scope to write-once, deploy-anywhere yet still access operating system level functionality.

ISVs can target multiple platforms with the same COBOL codebase, increasing market coverage and reducing application testing overheads. Application owners can swiftly change operating platforms to take advantage of commodity hardware and more agile, flexible deployment options.

This release of Visual COBOL includes support for the Docker container platform, further increasing flexibility for application development and deployment. New product packaging options enable Docker deployment to both Windows Server and Linux platforms.

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.

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.

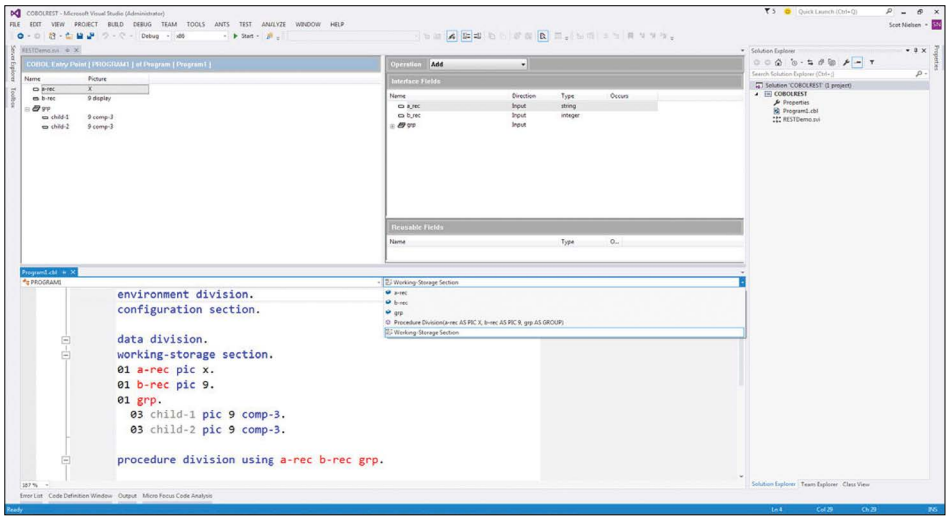

Figure 5. Creating a REST web service using the graphical web services toolkit provided in Visual COBOL

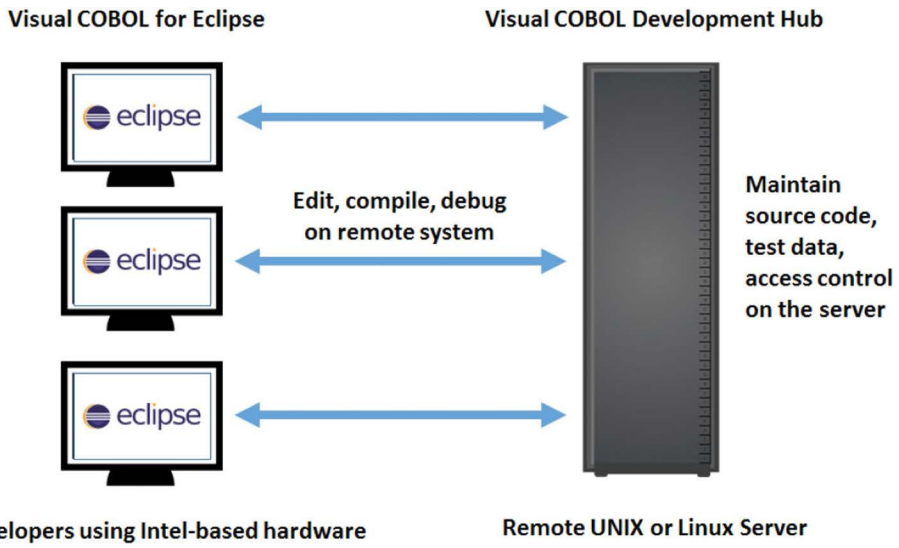

Figure 4. Visual COBOL for Eclipse remote development option

Reversible Debugging and Live Recorder

This release of Visual COBOL includes patented reversible debugging tools from Undo.

Unlike traditional debugging techniques, which allow the developer to step forwards through code as it executes, reversible debugging enables a developer to step back through the application's execution path. Somewhat akin to reviewing a CCTV recording, developers using reversible debugging tools can retrospectively examine application behavior to identify and resolve logic errors.

In addition to the standard reversible debugging tools available from within the Eclipse IDE, the Live Recorder utility allows developers to capture application execution at runtime. Live Recorder trace files can subsequently be loaded into Eclipse for debugging and diagnostic purposes. In this release of Visual COBOL, reversible debugging and Live Recorder tools are available as early adopter features on Red Hat only.

Agile Development

This release of Visual COBOL includes features designed to support Agile development teams.

The Micro Focus COBOL unit testing framework enables developers to create unit tests for new application code. Unit tests can be automated as part of a continuous integration system to provide immediate feedback to developers on the results of the latest checked-in code. The unit testing framework is available within the Visual Studio and Eclipse IDEs which provide tools for automatic test case creation and visualization of results.

Visual COBOL includes support for common Continuous Integration (CI) toolsets, such as Microsoft TFS and Jenkins. Documentation guides aid in configuring CI systems to compile COBOL applications and run unit testing.

CI pipelines can be augmented using additional validation checks such as coding standards rules from Micro Focus COBOL Analyzer, unit testing with code coverage, performance profiling and use of Live Recorder to capture failing test cases.

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 unified and seamless development experience within Eclipse. Compilation and debugging occur on the remote server while being initiated and controlled through the Eclipse IDE.

RESTful Web Services

Visual COBOL provides several options for building applications as part of an SOA or API development strategy. 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

The latest release of Visual COBOL includes new compiler and runtime optimizations which have been shown to increase application performance on Intel-based systems by between 20-30% across a broad range of benchmarks.

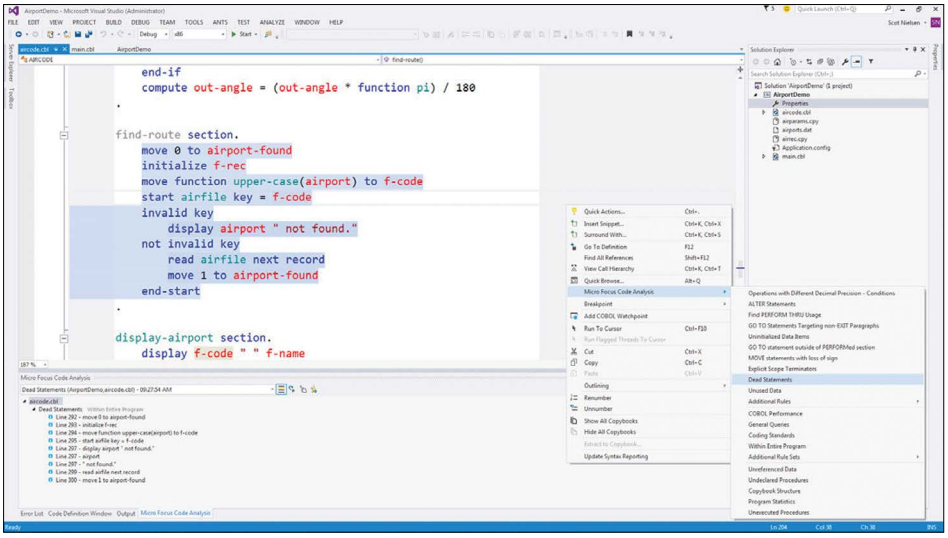

Figure 5. Creating a REST web service using the graphical web services toolkit provided in Visual COBOL

and deployed within the Micro Focus COBOL Server for SOA, a high-performance COBOL application server environment.

JSON

This release of Visual COBOL includes support for generating JSON data from COBOL records. The JSON GENERATE verb accepts a COBOL record structure and emits a JSON string; the verb also provides options for renaming or suppressing certain fields. As JSON rapidly becomes the standard for data interchange and with the predominance of tooling that can consume JSON, COBOL data in this format can be readily passed to external applications for processing.

Containers

The desire to deliver faster business results has accelerated Agile and DevOps practices within IT organizations. Engineering teams are achieving higher rates of success by embracing containerized architectures to support increased software deliveries.

Visual COBOL provides containerized development and deployment options using Docker.

"The Visual COBOL implementation has given us many benefits. Maintenance of our new, .NET-based solution is so much easier that we have realized a 20 percent IT cost savings. Our development team is more efficient and productive, thanks to the intuitive IDE, helping us respond faster to business requirements."

JOVENTINO MEDEIROS
Managing Director
Jove Informática

The Visual COBOL native web services toolkit supports containerized delivery of APIs which can be orchestrated by Kubernetes to provide scalable and reliable application delivery.

UI Modernization

For many organizations, the 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 frameworks provide unique options for overhauling aged user interfaces while preserving existing business logic. Modern UIs can be developed directly in COBOL or combined, using Java, C# or VB.NET for front end functionality and COBOL for backend service delivery.

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 directly 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.

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 traditional 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.NET 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 systems.

"Using Visual COBOL within our modernization project has given us unrivalled application platform stability, improved customer satisfaction and significant cost savings. Visual COBOL was the right choice for our business."

ALEJANDRO WYSS
CIO
Caja de Valores S.A.

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 where all data and sub programs are then 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 update or re-architect the existing COBOL applications and easing 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 mainframe application rehosting projects. Visual COBOL integrates with the leading transaction processing monitors and middleware vendors including IBM TX Series and Oracle Tuxedo. Our network of specialist partners, with decades of skill in application rehosting and modernization, ensures that the most effective solutions are provided for our customers.

Specialized compiler directives are provided to emulate source platform behavior and include platform-specific functions, such as Little Endian (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

Application Deployment

- COBOL Server

Contact us at:
www.microfocus.com

Like what you read? Share it.