



# isCOBOL™ Evolve

## isCOBOL Evolve 2011 Release 1 Overview

Copyright © 2011 Veryant LLC.  
9930 Derby Lane, Suite 202, Westchester, IL 60154, 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.

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

## Table of Contents

<b>Introduction.....</b>	<b>4</b>
<b>isCOBOL IDE Enhancements .....</b>	<b>4</b>
Multi-language editing capabilities.....	4
ANSI fixed format editing tools .....	5
Remote compilation .....	5
<b>Dynamic GUI Design Enhancements.....</b>	<b>7</b>
Docking Windows.....	7
Propose Values .....	7
<b>isCOBOL Load Balancer.....</b>	<b>8</b>
<b>Additional isCOBOL 2011 R1 enhancements .....</b>	<b>8</b>

## isCOBOL Evolve 2011 Release 1 Overview

### Introduction

Veryant is pleased to announce the latest release of isCOBOL™ Evolve (formerly the isCOBOL Application Platform Suite), isCOBOL Evolve 2011 Release 1 (R1). isCOBOL Evolve provides a complete environment for the development, deployment, maintenance, and modernization of COBOL applications. isCOBOL 2011 R1 includes several enhancements to the isCOBOL™ Integrated Development Environment (IDE), new graphical user interface (GUI) features, and a new isCOBOL™ Load Balancer add-on product for large-scale deployments with thin client GUI s.

Details on these enhancements and other new features are included below.

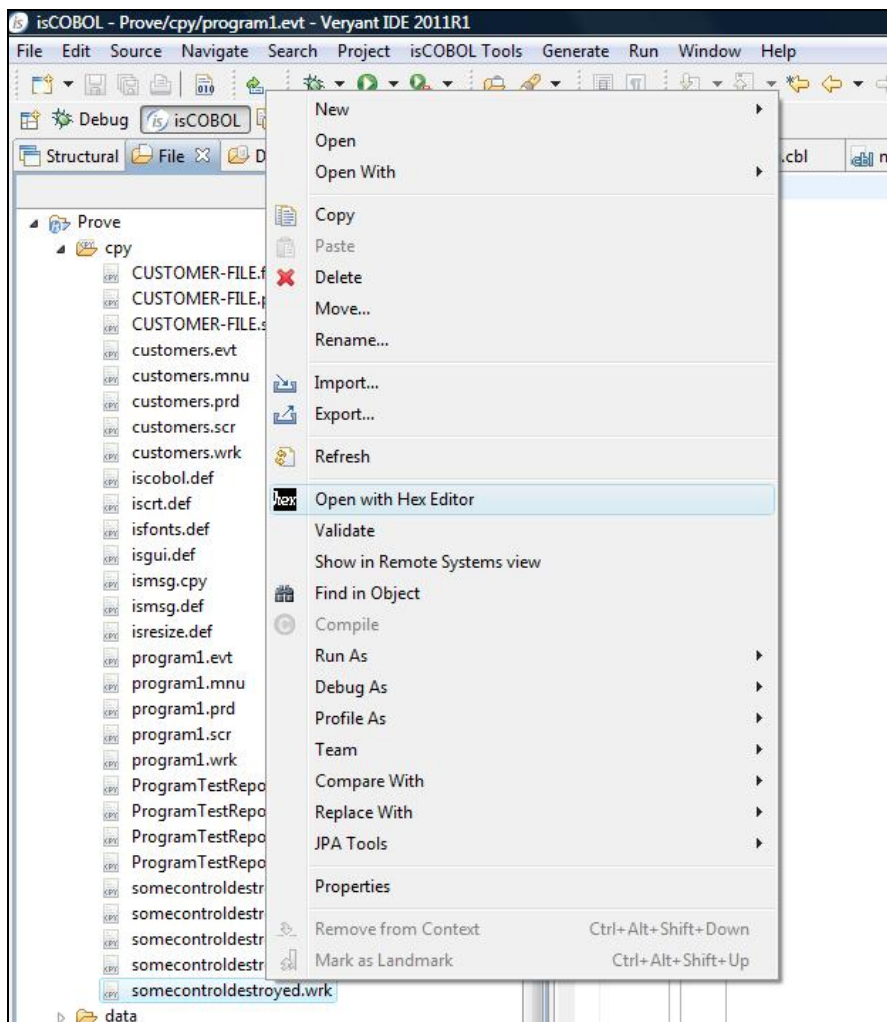
### isCOBOL IDE Enhancements

In isCOBOL Evolve 2011 R1, the isCOBOL IDE has been updated to support Eclipse Helios, the latest in the Eclipse release train. The isCOBOL IDE also offers several new productivity enhancements, including:

#### Multi-language editing capabilities

isCOBOL IDE users can access Java, XML, and HTML editing capabilities, as well as a Hex editor and Version Control with CVS and Subversion, providing a streamlined, industry-standard environment for building, deploying and testing applications across the entire software lifecycle, regardless of language.

To enable this functionality, a developer associates a particular file extension with the desired editor in the isCOBOL IDE (from 'Window → Preferences → General → Editors → File Associations'). For example, as depicted in Figure 1, *Opening a Hex Editor in the isCOBOL IDE*, a Hex Editor can be directly accessed from a pop-up menu in the IDE window after an association has been made.

**Figure 1.** Opening a Hex Editor in the isCOBOL IDE

### ANSI fixed format editing tools

Programmers accustomed to using a horizontal ruler to show current cursor column position when editing ANSI fixed format code can now enable a ruler from the isCOBOL IDE. By default the horizontal ruler is not visible. The ruler can be enabled from 'Window → Preferences → isCOBOL → Editor'.

### Remote compilation

The addition of a remote compilation feature in the isCOBOL IDE allows developers to incorporate server-side pre-compilers and translators into the application build step. As depicted in Figure 2, *Configuring Remote Compilation*, developers can push files to a server,

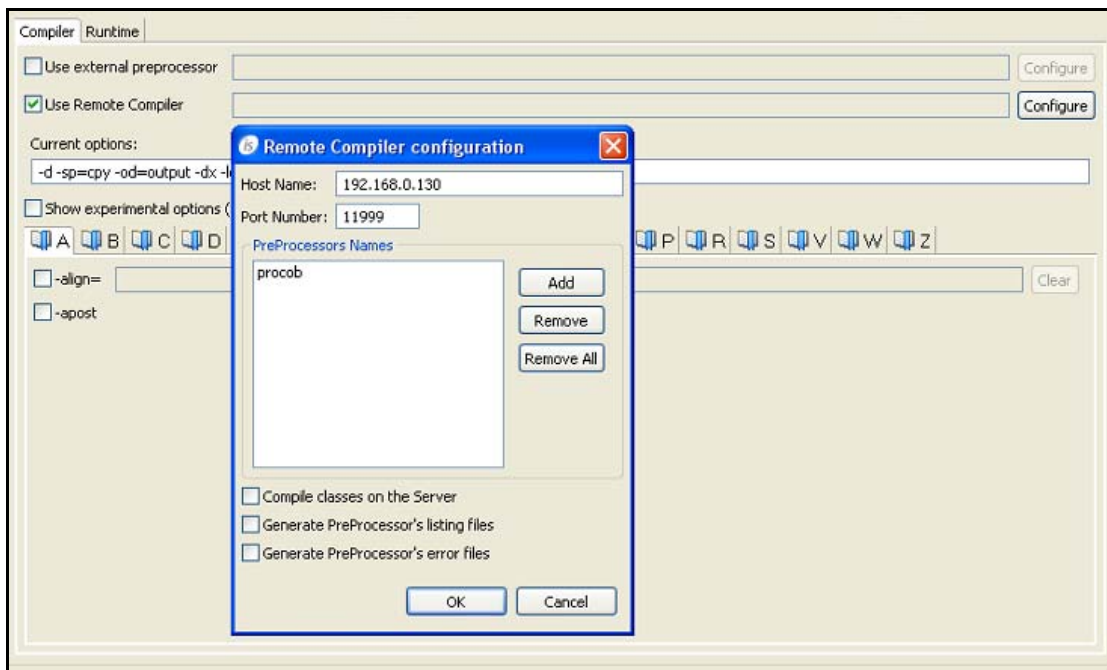
preprocess, compile and then pull errors and .class files back into the isCOBOL IDE with a single click of the mouse. Assuming there is a Remote Compiler listening on a server, the isCOBOL IDE can be instructed to delegate the compilation.

This feature is activated by performing the following steps:

1. Click on *Project* menu
2. Select *Properties*
3. Expand *Settings* and select *Compile/Runtime* options
4. Check the *Use Remote Compiler* option
5. Click on the *Configure* button and input the necessary information

*Host Name* and *Port* are the address and the port of the server where the Remote Compiler is listening. *PreProcessors Names* is the list of preprocessors that must be invoked to precompile the source file (click on the Add button to add items to the list). The remaining settings are optional.

**Figure 2.** Configuring Remote Compilation



Now, each time a compilation is issued, it will be done remotely. To disable the feature and compile locally, uncheck the *Use Remote Compiler* option.

The remote compilation feature is also available outside of the isCOBOL IDE from the isCOBOL Compiler command line.

### **Dynamic GUI Design Enhancements**

In 2011 R1, developers can add docking windows to GUIs developed with isCOBOL Evolve, as well as enable form entry fields to propose values to end-users from a preset list or a user's history.

#### Docking Windows

By enabling docking windows, application end-users can attach multiple windows to a single container window and the windows will resize accordingly.

The syntax to create the container is:

```
display docking window
    layout w-layout
    [...]
    handle h-main.
```

The syntax to create a window inside the container is:

```
display dockable window upon h-main
    upon-leaf var-leaf
    [...]
```

#### Propose Values

With the addition of the PROPOSAL property to the ENTRY-FIELD control, isCOBOL developers can now automatically propose values from a preset list or a user's history to end-users completing GUI form entry fields.

Multiple values are allowed if specified between parentheses. If the Proposals-Unsorted style is set, the position of the new proposal can be controlled by the Proposal-Index property.

More information about ENTRY-FIELD control can be found in the isCOBOL 2011 R1 documentation.

### **isCOBOL Load Balancer**

The isCOBOL Load Balancer has been introduced for organizations distributing COBOL applications with 100% portable thin client user interfaces. An add-on product to isCOBOL Application Server, the load balancer distributes workloads among clusters of server processes running on the same or different machines for enhanced horizontal scalability or workload separation.

Once the isCOBOL Load Balancer is started, it waits for connections from isCOBOL Clients and when a connection is requested, it evaluates the best server for satisfying the request, and then supplies the address of that server to the isCOBOL Client. From this point on, the isCOBOL Client communicates with the isCOBOL Server directly and the connection between the isCOBOL Client and the isCOBOL Load Balancer is closed. Therefore, shutting down the isCOBOL Load Balancer will not impact current connections.

Options and parameters for the isCOBOL Load Balancer can be set from the isCOBOL Load Balancer panel in the isCOBOL IDE. When operating, the isCOBOL Load Balancer checks the connections to configured servers at regular intervals. The interval is set at 60 seconds by default, but the actual number of seconds can be customized through a property.

More information can be found on the isCOBOL Load Balancer in the isCOBOL Application Server chapter of the isCOBOL Evolve User's Guide.

### **Additional isCOBOL 2011 R1 enhancements**

Other advances made in isCOBOL 2011 R1 include:

- Improved database performance, up to 33%, by reducing overhead on read workloads with ESQL FETCH statements.
- Improved Veryant ESQL Generator performance of up to 30% on READ NEXT and READ PREVIOUS statements.
- Added compatibility with mainframe and ANSI 85 COBOL syntax to further simplify transitions to the isCOBOL platform, including:
  - VALUE in REDEFINES
- Enhanced ACUCOBOL-GT compatibility
  - character color 8192
  - modify entry-field styles



- New compiler error and warning messages
  - new warning: " With decimal point comma, comma cannot be separator between two digits"
  - new error: "Invalid host variable" for ESQL statements
  - new error: "ESQL invalid end statement: /\*" for unclosed Oracle hints
  - compiler option: -wref to enable "reference modifier out of range" warnings