# isCOBOL$^{TM}$ Evolve

## isCOBOL Evolve 2016 Release 1 Overview

# isCOBOL Evolve 2016 Release 1 Overview

## Introduction

Veryant is pleased to announce the latest release of isCOBOL™ Evolve, isCOBOL Evolve 2016 R1.

isCOBOL Evolve provides a complete environment for the development, deployment, maintenance, and modernization of COBOL applications.

isCOBOL 2016 R1 includes a new product named isUPDATER, which allows you to keep all COBOL application, modules and isCOBOL framework installed up to date.

isCOBOL 2016R1 also includes the ability to export Excel file format from isCOBOL IDE generated reports, native Android SDK integration to improve the development of mobile applications, an even better isCOBOL Debugger with several usability improvements, better runtime performance in several areas, and new UI features to continue to satisfy developers requests.

isCOBOL EIS 2016R1 provides many enhancements on web direct capabilities on CSS style managements and the support of Bootstrap styles and Font Awesome icons.

Details on these enhancements and updates are included below.

## isUPDATER: software updates for all COBOL applications

New software versions and program updates are vital steps in staying secure online, as security patches are often included in updates, as well as solutions for vulnerabilities and bugs. The goal of this new product is to assist you to keep all COBOL applications, native modules and isCOBOL libraries up to date.

The isUPDATER software:

- Provides an easy, one-click activation of applications
- Guarantees that you are always running the latest version of the application
- Eliminates complicated installation or upgrade procedures

Based on HTTP communication protocol, isUPDATER can be used in conjunction with any HTTP server like IIS or Apache. isCOBOL Server provides very basic HTTP services to make the adoption of this new feature painless.

With the server configuration file, it is possible to define the packages that must be kept up to date. It is also possible to define a client-side POSTUPDATE procedure for each package to be updated. This allows the developer to specify precise tasks to be executed on the client where packages are updated, after updating procedures are run.

On the server packages and versions are defined in the `swupdater.properties` property file, such as follows:

```
swupdater.version.iscobol=100

swupdater.zipfile.iscobol=isCOBOL2016.zip

swupdater.version.application=100

swupdater.zipfile.application=customer_application.zip
```

The files listed above are located on the root folder of the isCOBOL Server with HTTP services turned on, or any HTTP server of your choice.

On the client site a property file defines the address of HTTP server and the location of packages to be kept updated:

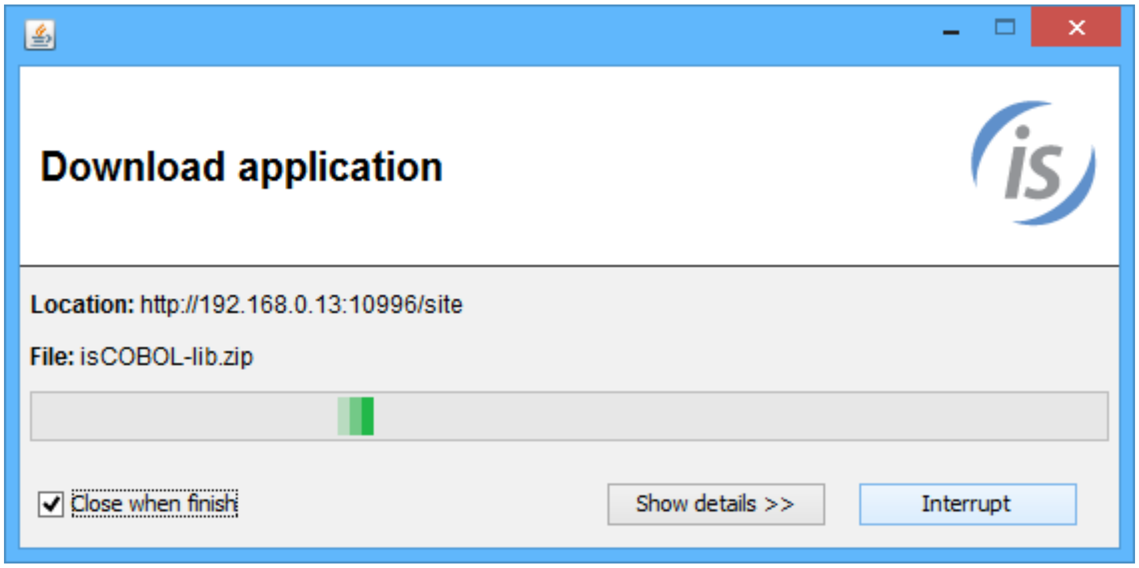
```
swupdater.site=http://192.168.0.13:10996/site

swupdater.version.iscobol=90

swupdater.directory.iscobol=c:/isCOBOL2016R1

swupdater.version.application=90

swupdater.directory.application=C:/Application

swupdater.mainclass=MAIN_APPLICATION
```

With the above basic configuration. the isUPDATER command on the client site can be invoked to start the updating process from a command prompt by typing:

```
iscupdater -c isupdater.properties
```

This command will launch the updating process as ruled on the server and on the client's configuration files. As depicted in Figure 1, *isUPDATER in action*, the command will show a window with progress information.

**Figure 1**, isUPDATER in action.



After updating the packages, as defined using the `swupdater.mainclass` property, the main  COBOL application program  named MAIN_APPLICATION will be executed.

Another use of the updater facility could be to run isCOBOL thin client after the updates have been downloaded, to allow the client to update its jar files.

This configuration automates the task of updating the jar files on the client when the version of the isCOBOL Server doesn't match the version of the isCOBOL thin client.

The following property file allows running the thin client after client is updated:

```
swupdater.site=http://192.168.0.13:10996/site

swupdater.version.iscobol=90

swupdater.directory.iscobol=c:/isCOBOL2016R1

swupdater.version.application=90

swupdater.directory.application=C:/Application

swupdater.mainclass=com.iscobol.gui.client.Client MAIN_APPLICATION
```

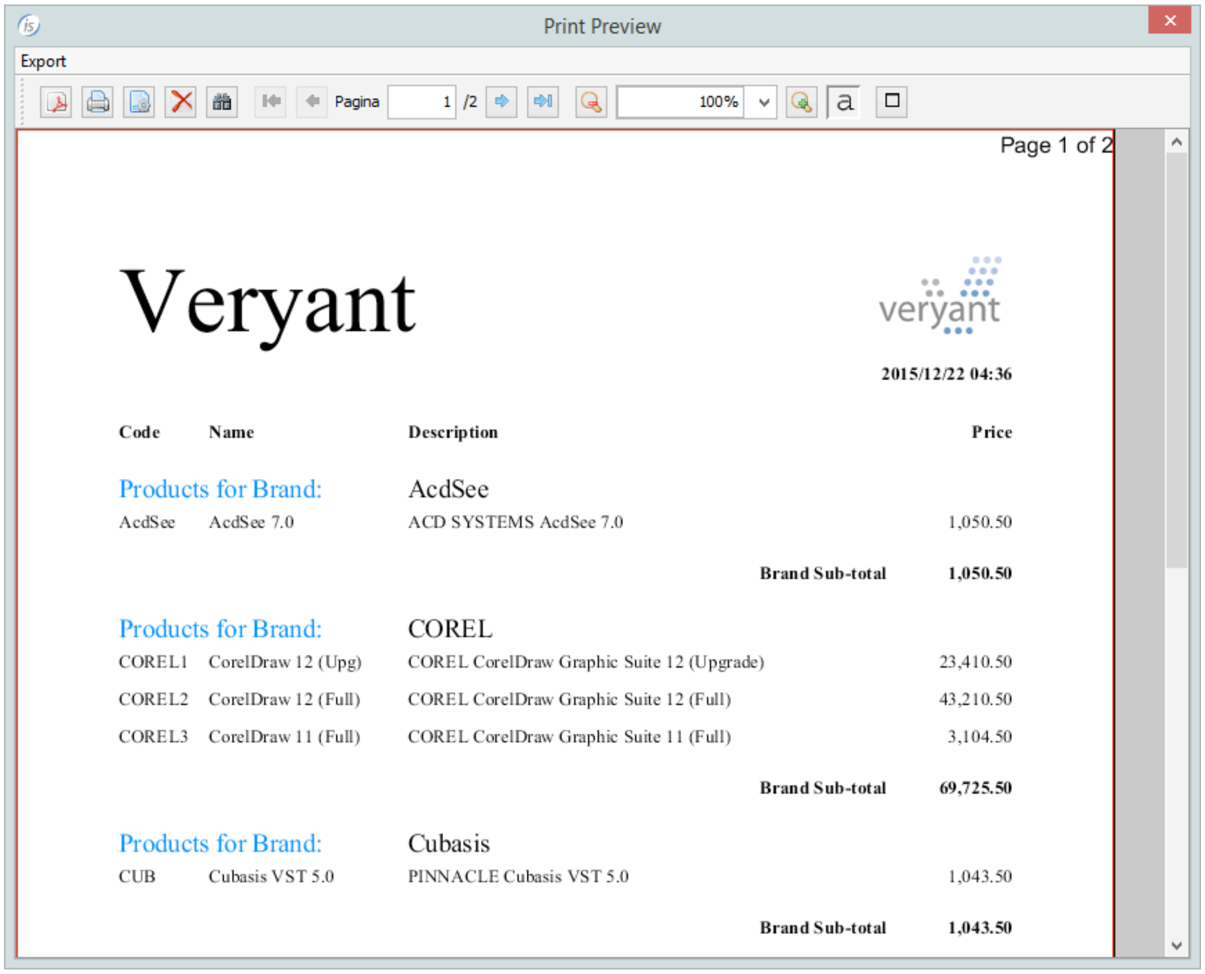
## isCOBOL IDE Enhancements

In isCOBOL Evolve 2016 R1, the isCOBOL IDE includes changes that improve the usability and productivity with additional new features such as Microsoft Excel file format export from the Report Designer generated programs.

### Report Designer

isCOBOL IDE 2016 R1 has enhanced the Report Designer by adding the ability for report programs to export into Microsoft Excel file formats.  As depicted in Figure 2, *New Export Menu*, during print preview the user can export the generated output to XLS, XLSX, PDF formats, or to print on a printer.

**Figure 2**. New Export Menu



Page 1 of 2

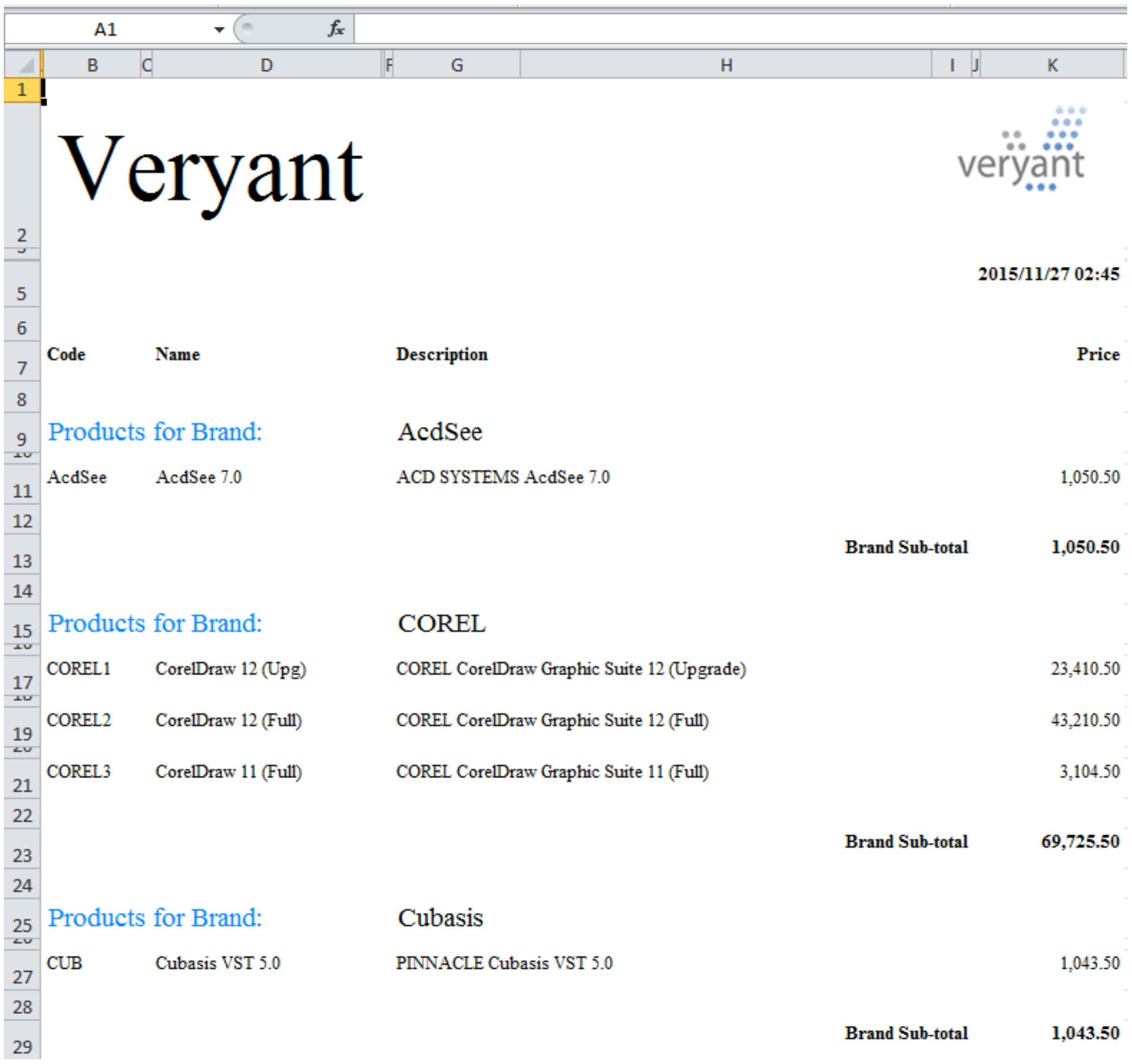
Veryant

2015/12/22 04:36

| Code | Name | Description | Price |
|---|---|---|---|
| Products for Brand: | | AcdSee | |
| AcdSee | AcdSee 7.0 | ACD SYSTEMS AcdSee 7.0 | 1,050.50 |
| | | Brand Sub-total | 1,050.50 |
| Products for Brand: | | COREL | |
| COREL1 | CorelDraw 12 (Upg) | COREL CorelDraw Graphic Suite 12 (Upgrade) | 23,410.50 |
| COREL2 | CorelDraw 12 (Full) | COREL CorelDraw Graphic Suite 12 (Full) | 43,210.50 |
| COREL3 | CorelDraw 11 (Full) | COREL CorelDraw Graphic Suite 11 (Full) | 3,104.50 |
| | | Brand Sub-total | 69,725.50 |
| Products for Brand: | | Cubasis | |
| CUB | Cubasis VST 5.0 | PINNACLE Cubasis VST 5.0 | 1,043.50 |
| | | Brand Sub-total | 1,043.50 |

The Figure 3, *Exporting example*, shows an Excel sheet automatically generated from the Export menu of print preview. The export feature converts all graphical definition like fonts, formats and images into Excel equivalent features, as seen in Figure 3.

**Figure 3**. Exporting example



Veryant

2015/11/27 02:45

| Code | Name | Description | | Price |
|---|---|---|---|---|
| Products for Brand: | | AcdSee | | |
| AcdSee | AcdSee 7.0 | ACD SYSTEMS AcdSee 7.0 | | 1,050.50 |
| | | | **Brand Sub-total** | **1,050.50** |
| Products for Brand: | | COREL | | |
| COREL1 | CorelDraw 12 (Upg) | COREL CorelDraw Graphic Suite 12 (Upgrade) | | 23,410.50 |
| COREL2 | CorelDraw 12 (Full) | COREL CorelDraw Graphic Suite 12 (Full) | | 43,210.50 |
| COREL3 | CorelDraw 11 (Full) | COREL CorelDraw Graphic Suite 11 (Full) | | 3,104.50 |
| | | | **Brand Sub-total** | **69,725.50** |
| Products for Brand: | | Cubasis | | |
| CUB | Cubasis VST 5.0 | PINNACLE Cubasis VST 5.0 | | 1,043.50 |
| | | | **Brand Sub-total** | **1,043.50** |

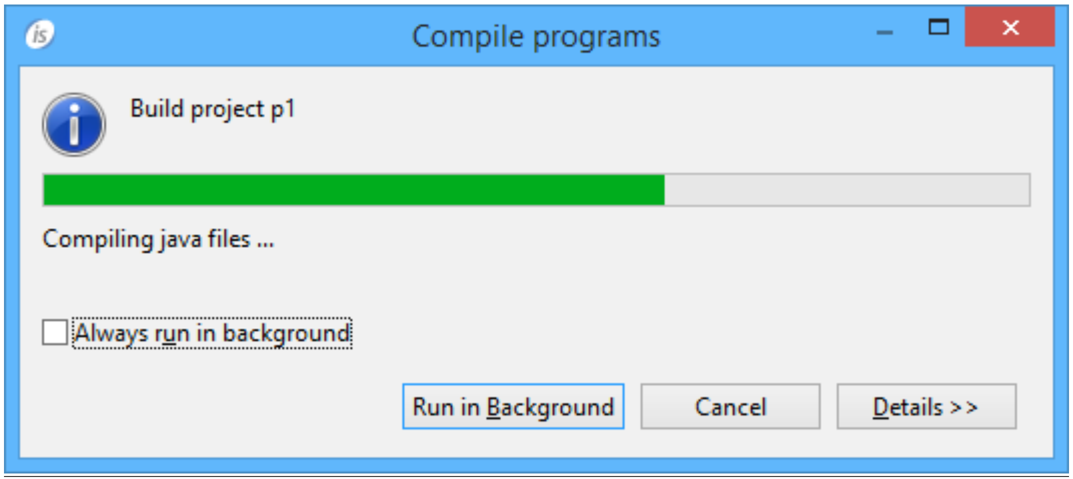
To better configure the output layout generations, the Export feature defaults and behaviors can be fine-tuned using the following properties:

```
iscobol.export.excel.cell_ignore_background=true|false
iscobol.export.excel.cell_ignore_borders=true|false
iscobol.export.excel.cell_locked=true|false
iscobol.export.excel.cell_numeric_format=valid excel numeric format
iscobol.export.excel.cell_wrap_text=true|false
iscobol.export.excel.collapse_row_span=true|false
iscobol.export.excel.detect_cell_type=true|false
iscobol.export.excel.force_page_breaks=true|false
iscobol.export.excel.freeze_page_header=true|false
iscobol.export.excel.ignore_images=true|false
iscobol.export.excel.remove_columns_space=true|false
iscobol.export.excel.remove_rows_space=true|false
iscobol.export.excel.whitepage_background=true|false
```

Background compilation

Compiles can now be run in background, allowing developers to continue working on the project during long compilations without wasting time (as shown in Figure 4, *Compile in Background mode*).

**Figure 4**. Compile in Background mode



Build project p1

Compiling java files ...

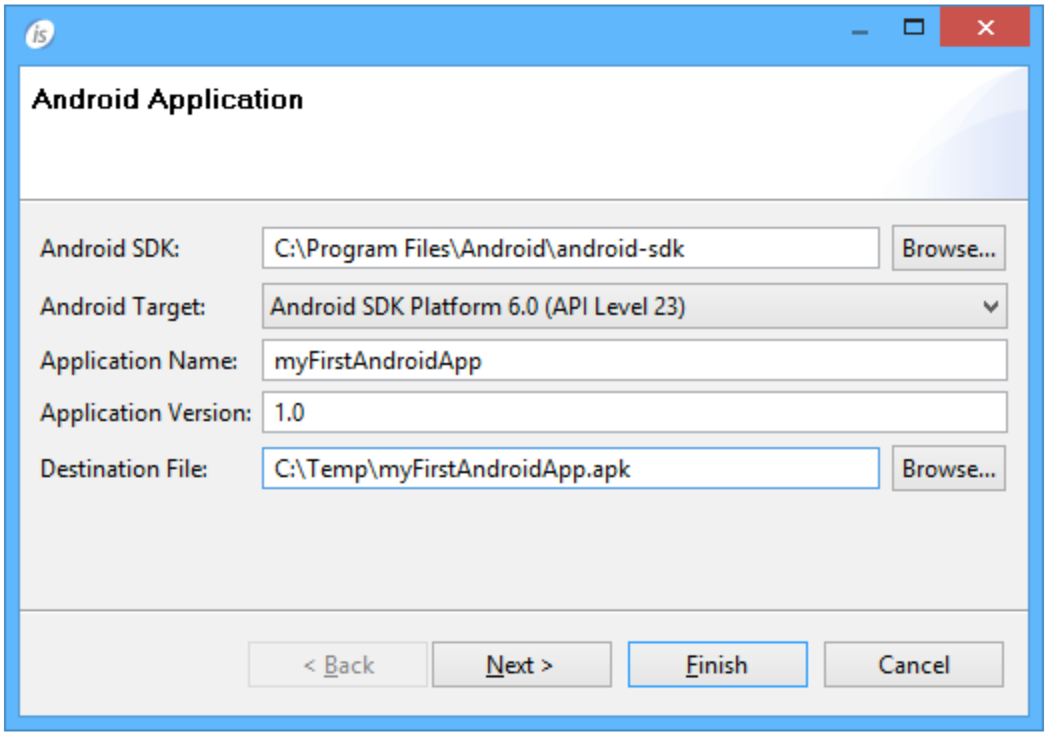
☐ Always run in background

## ANDROID SDK integration

In order to simplify mobile applications development, isCOBOL 2016R1 natively supports the generation of Android packages. As depicted in Figure 5, *New Dialog to create an Android Application,* a new Wizard named "Export to Android Application" is available in isCOBOL HTML project of isCOBOL IDE.

This new step-by-step wizard allows creation of APKs, without needing any external Eclipse plugin, just filling required fields to specify the SDK to be used, the application name, the application version and destination file.

**Figure 5**. New Dialog to create an Android Application



**Android Application**

| | |
|---|---|
| Android SDK: | C:\Program Files\Android\android-sdk |
| Android Target: | Android SDK Platform 6.0 (API Level 23) |
| Application Name: | myFirstAndroidApp |
| Application Version: | 1.0 |
| Destination File: | C:\Temp\myFirstAndroidApp.apk |

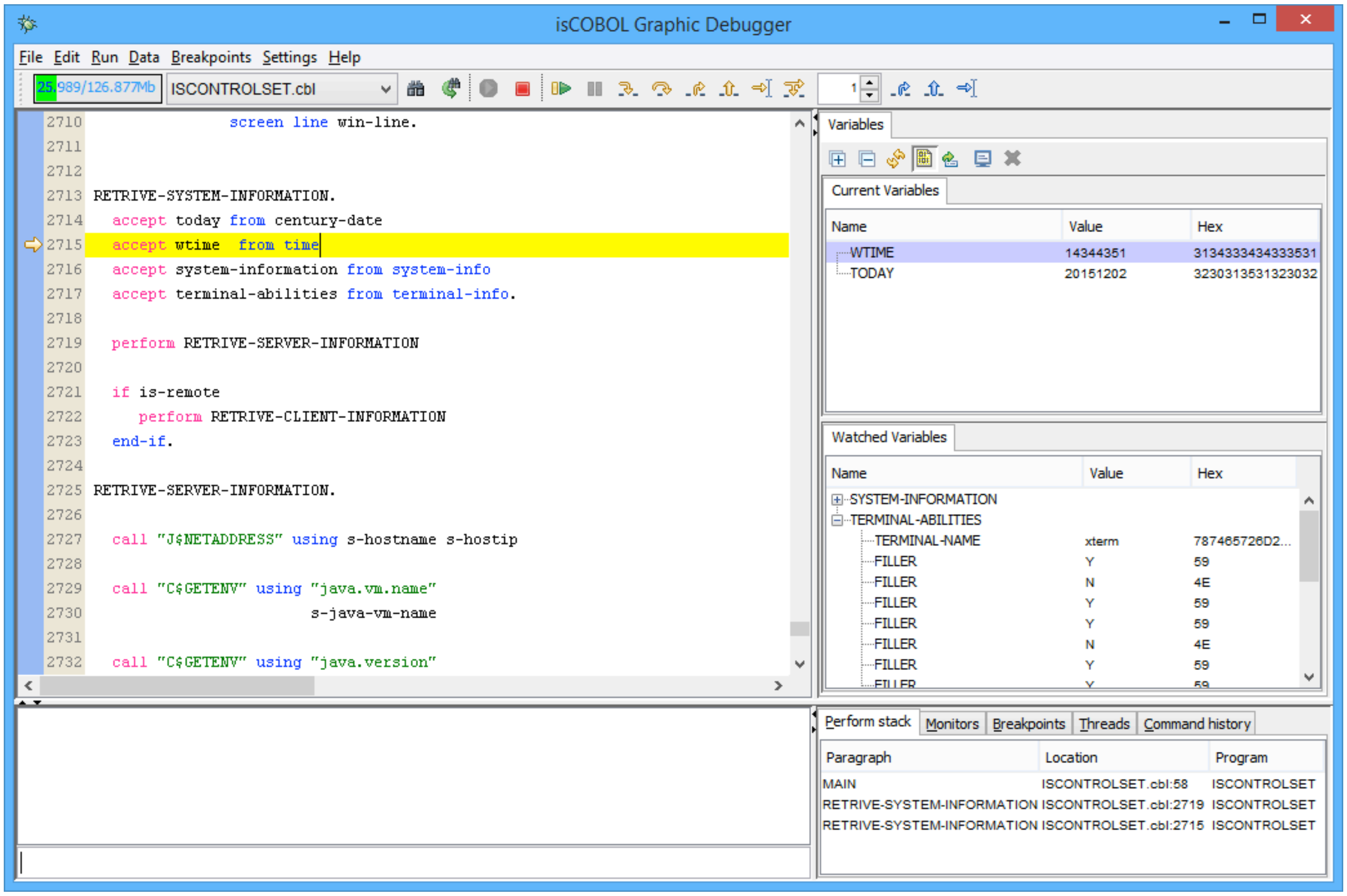
## Debugger Improvements

isCOBOL Debugger continues to be a core feature in isCOBOL Evolve, so it has been carefully improved improving usability and ease of use.

### New Current Variables View

isCOBOL Debugger now automatically shows the Current Variables to easily follow program execution's flow. As depicted in Figure 6, *New Current Variables View*, in the right part of the Debugger window, an area is reserved to show the Current Variables (list of variables used in the current and previous statements) and the Watched Variables, making it easy to monitor multiple variables during the stepping process.

**Figure 6**. New Current Variables View



```
                 screen line win-line.


RETRIVE-SYSTEM-INFORMATION.
   accept today from century-date
   accept wtime  from time
   accept system-information from system-info
   accept terminal-abilities from terminal-info.

   perform RETRIVE-SERVER-INFORMATION

   if is-remote
      perform RETRIVE-CLIENT-INFORMATION
   end-if.

RETRIVE-SERVER-INFORMATION.

   call "J$NETADDRESS" using s-hostname s-hostip

   call "C$GETENV" using "java.vm.name"
                         s-java-vm-name

   call "C$GETENV" using "java.version"
```

Current Variables

| Name | Value | Hex |
|---|---|---|
| WTIME | 14344351 | 3134333434333531 |
| TODAY | 20151202 | 3230313531323032 |

Watched Variables

| Name | Value | Hex |
|---|---|---|
| SYSTEM-INFORMATION | | |
| TERMINAL-ABILITIES | | |
| TERMINAL-NAME | xterm | 787465726D2... |
| FILLER | Y | 59 |
| FILLER | N | 4E |
| FILLER | Y | 59 |
| FILLER | Y | 59 |
| FILLER | N | 4E |
| FILLER | Y | 59 |
| FILLER | Y | 59 |

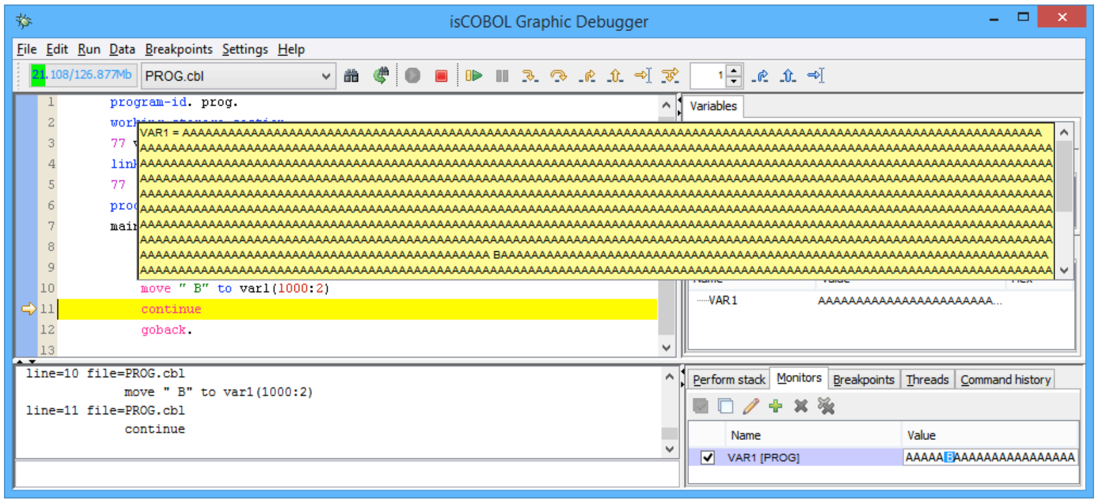
Perform stack

| Paragraph | Location | Program |
|---|---|---|
| MAIN | ISCONTROLSET.cbl:58 | ISCONTROLSET |
| RETRIVE-SYSTEM-INFORMATION | ISCONTROLSET.cbl:2719 | ISCONTROLSET |
| RETRIVE-SYSTEM-INFORMATION | ISCONTROLSET.cbl:2715 | ISCONTROLSET |

Other minor enhancements

The Figure 7, *New Management of long values,* shows the new management introduced in debugger interface to better manage long values in tool-tip and monitor.

**Figure 7**. New management of long values


isCOBOL Graphic Debugger
File Edit Run Data Breakpoints Settings Help
21.108/126.877Mb
PROG.cbl
Variables
Perform stack Monitors Breakpoints Threads Command history
Name
Value

```
program-id. prog.
wor
77 v
lin
77
pro
mai


    move " B" to var1(1000:2)
    continue
    goback.
```

```
line=10 file=PROG.cbl
          move " B" to var1(1000:2)
line=11 file=PROG.cbl
          continue
```

The “New button” (green plus icon) has been added in Monitor and Breakpoint Debugger views to simplify the creation of new Monitor or Breakpoint. Additionally, a new configuration property named `iscobol.debug.propfile` has been added to specify a different configuration file to store Debugger settings, to ease sharing preferences across team members.

## New User Interface Features

To stay up-to-date with new requests to develop modern and easy to use applications, Veryant engineers continue to improve UI with new controls, style and properties. An example?  The new ACCORDION style in tab-control.

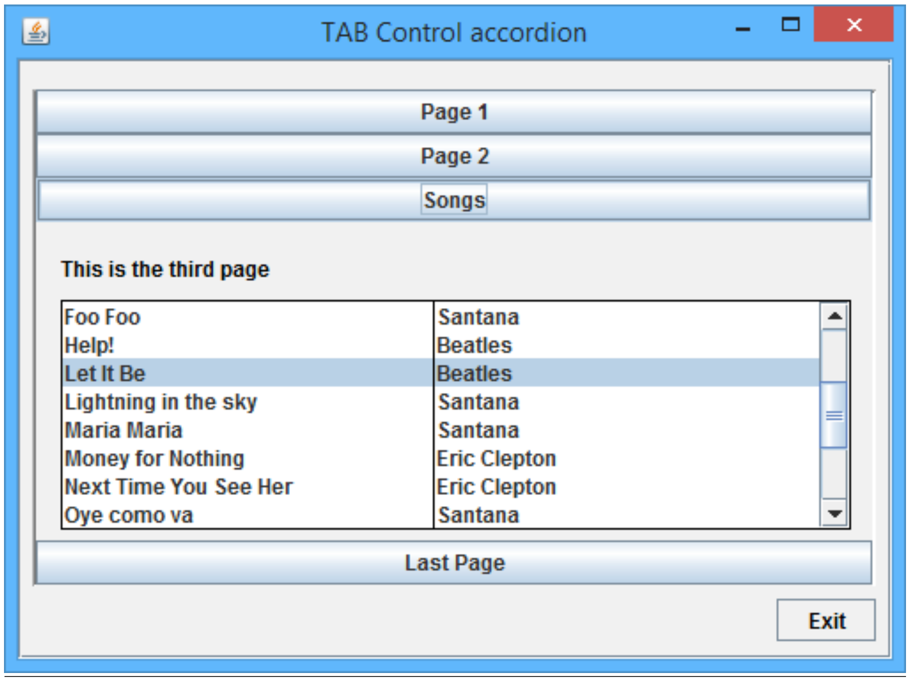
### New layout ACCORDION in TAB-CONTROL

isCOBOL 2016 R1 introduces a new layout in TAB-CONTROL that can be activated with the new style named ACCORDION.  Figure 8, *Accordion*, shows an example of this new style in use.  ACCORDIONs are useful to group controls that need to be visible when their owning page is made visible.

Code example:

```
  03 Tb1-accordion tab-control
     line 2 col 2 lines 17 cells size 68 cells
     accordion.
```

**Figure 8**. Accordion.



This is the third page

| | |
|---|---|
| Foo Foo | Santana |
| Help! | Beatles |
| Let It Be | Beatles |
| Lightning in the sky | Santana |
| Maria Maria | Santana |
| Money for Nothing | Eric Clepton |
| Next Time You See Her | Eric Clepton |
| Oye como va | Santana |

New control properties

New property in ENTRY-FIELD control:

- PROPOSAL-MIN-TEXT to show the proposals after a number of characters typed in the field. This works in conjunction with existing PROPOSAL properties.

New property in ENTRY-FIELD and COMBO-BOX controls:

- PLACEHOLDER to show a hint inside the control when the field is empty. This assists the user to understand which type of information is required in the editing field.

New property available in all controls with borders:

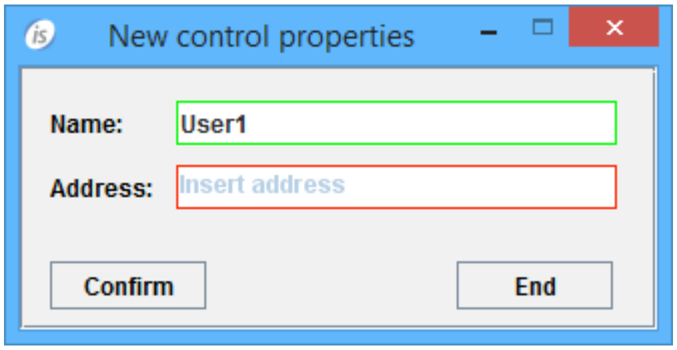
- BORDER-COLOR sets the border color when BOXED style is set (in entry-field it is set by default). It supports all color values and also RGB colors.

The following example shows how to use the new properties PROPOSAL-MIN-TEXT, PLACEHOLDER and BORDER-COLOR:

```
03 efname entry-field line 2 col 12 size 30
   border-color 11
   placeholder "Insert name"
   proposal-min-text 2.
03 efaddress entry-field line 4 col 12 size 30
   border-color 13
   placeholder "Insert address".
```

The image shown in Figure 9, *example of new properties*, shows the final result of BORDER-COLOR on the fields, and the PLACEHOLDER text is visible only on the second field as the entry field is still empty.

**Figure 9**. Example of new properties.



## Framework Improvements

isCOBOL framework is the heart of isCOBOL Evolve. Once again Veryant's engineers increased performance on many common statements, to allow the entire COBOL program to run faster than ever before. In addition to performance improvements, isCOBOL 2016R1 provides JISAM encryption, reduction of memory footprint on numeric fields and many other improvements.

### Performance and memory improvements

Many COBOL statements have been optimized in order to increase execution performance of batch COBOL programs. To take advantage of most performance improvements, the COBOL source code only require to be recompiled with isCOBOL compiler 2016R1.

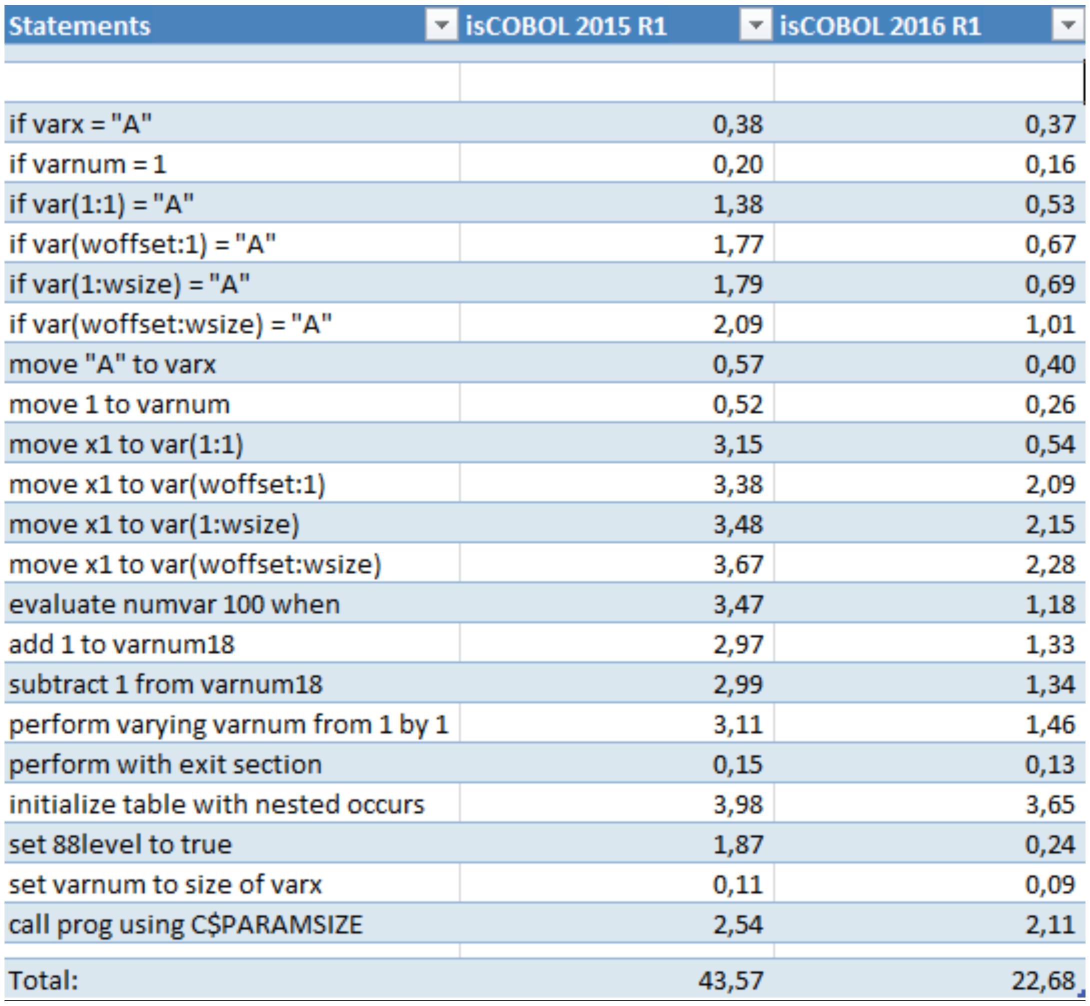
This is the list of statements that were improved in 2016R1:

- ADD,
- EVALUATE,
- EXIT SECTION,
- IF,
- MOVE,
- PERFORM,
- SET,
- SUBTRACT.

isCOBOL 2016R1 also optimizes COBOL programs that use C memory model (-cp) on several areas. The memory used to store numeric COBOL variables has been reduced.

The Figure 10, *Performance Comparison*, shows a comparison between isCOBOL 2016R1 and isCOBOL 2015R1. The performance test was executed on Windows 10 64 bit (CPU: Intel Core i-5 Processor 4440+, 3.10 GHz / RAM: 8GB / isCOBOL compiler option -dz / java used: Oracle JDK1.8.0_65 with option –server, time in seconds).

**Figure 10**. Performance Comparison



| Statements | isCOBOL 2015 R1 | isCOBOL 2016 R1 |
|---|---|---|
| | | |
| if varx = "A" | 0,38 | 0,37 |
| if varnum = 1 | 0,20 | 0,16 |
| if var(1:1) = "A" | 1,38 | 0,53 |
| if var(woffset:1) = "A" | 1,77 | 0,67 |
| if var(1:wsize) = "A" | 1,79 | 0,69 |
| if var(woffset:wsize) = "A" | 2,09 | 1,01 |
| move "A" to varx | 0,57 | 0,40 |
| move 1 to varnum | 0,52 | 0,26 |
| move x1 to var(1:1) | 3,15 | 0,54 |
| move x1 to var(woffset:1) | 3,38 | 2,09 |
| move x1 to var(1:wsize) | 3,48 | 2,15 |
| move x1 to var(woffset:wsize) | 3,67 | 2,28 |
| evaluate numvar 100 when | 3,47 | 1,18 |
| add 1 to varnum18 | 2,97 | 1,33 |
| subtract 1 from varnum18 | 2,99 | 1,34 |
| perform varying varnum from 1 by 1 | 3,11 | 1,46 |
| perform with exit section | 0,15 | 0,13 |
| initialize table with nested occurs | 3,98 | 3,65 |
| set 88level to true | 1,87 | 0,24 |
| set varnum to size of varx | 0,11 | 0,09 |
| call prog using C$PARAMSIZE | 2,54 | 2,11 |
| | | |
| Total: | 43,57 | 22,68 |

## Encryption on JISAM files

JISAM now supports data encryption declared using the WITH ENCRYPTION clause in the SELECT of indexed files. A new property named:

```
iscobol.file.encryption.key
```

allows to specify the key to be passed to the file handler. This property can be set dynamically in COBOL programs using the SET ENVIRONMENT statement to maximize security.

Code example:

```
 select pwd-file assign to "pwdfile"
        organization is indexed
        access mode is dynamic
        record key is pwd-key
        WITH ENCRYPTION
```

When the code runs, all data stored in the “pwdfile” (both .dat and .idx) will be encrypted based on the encryption key.

If the encryption key is missing, opening an encrypted file will result in a new error status “9X”, “Encryption's key missing”.

JUTIL command line utility has been updated to work with encrypted JISAM files. A new option -e=encryption_key needs to be added to the JUTIL command line to access encrypted files.

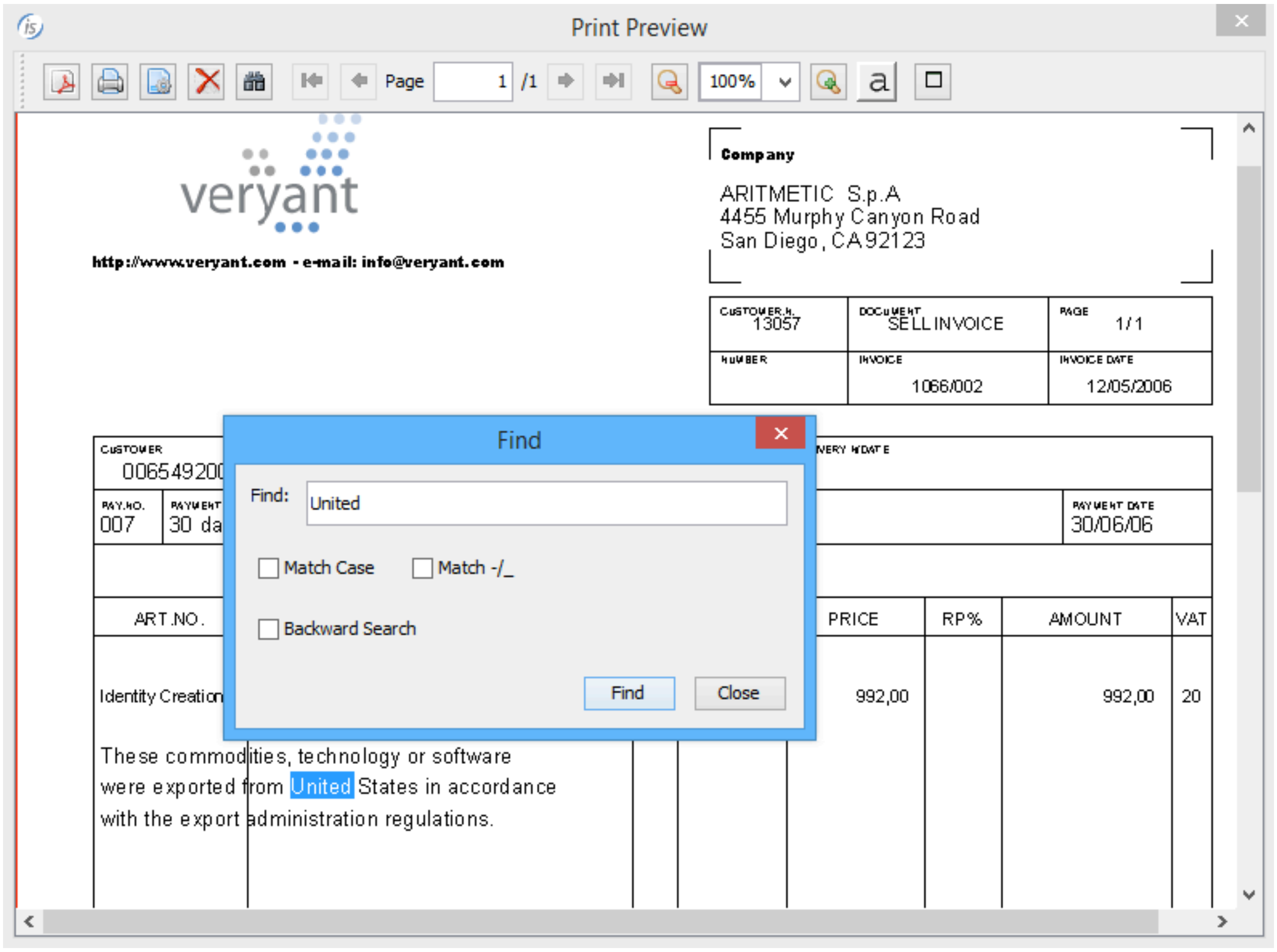
JUTIL command line examples:

```
jutil -e=MyKey123 –unload pwdfile outputbinpwd

jutil -e=MyKey123 –rebuild pwdfile
```

## Find features on print preview

As depicted in Figure 11, *Find in preview*, the "find" feature is now supported in the isCOBOL Print Preview. User of print preview can now easily search text forward or backward with the option to match -/_.

**Figure 11**. Find in Preview.



**Company**

ARITMETIC S.p.A
4455 Murphy Canyon Road
San Diego, CA 92123

http://www.veryant.com - e-mail: info@veryant.com

| | | |
|---|---|---|
| CUSTOMER N. 13057 | DOCUMENT SELL INVOICE | PAGE 1/1 |
| NUMBER | INVOICE 1066/002 | INVOICE DATE 12/05/2006 |

CUSTOMER 00654920

PAY.NO. 007 | PAYMENT 30 da | PAYMENT DATE 30/06/06

| ART.NO. | PRICE | RP% | AMOUNT | VAT |
|---|---|---|---|---|
| Identity Creation | 992,00 | | 992,00 | 20 |

These commodities, technology or software were exported from United States in accordance with the export administration regulations.

Find

Find: United

☐ Match Case ☐ Match -/_

☐ Backward Search

Find Close

Ability to know which process is locking a record

COBOL programs can call a new library routine named “C$LOCKPID” to find out which process is locking a record with c-tree file handler.

The routine returns the c-tree task number to be used in the Faircom utilities to monitor the connected clients.

Code example:

```
77  wctid  pic 9(7).
...
read myfile next with lock
if myfile-fs = "99"
   call "C$LOCKPID" giving wctid
end-if
```

## Other enhancements

Additional minor enhancements provided in isCOBOL 2016R1 framework:

- Ability to have all ACCEPT statements time out just as if there was a BEFORE TIME phrase present in the ACCEPT statement. A new configuration property `iscobol.accept_timeout` to specify the number of seconds to timeout.
  Configuration example:
  ```
  iscobol.accept_timeout=5
  ```
- Ability to set attributes for PDF generation through configuration properties, using the `iscobol.print.attribute.attribute-name=attribute-value` syntax.

  The attribute-name and attribute-value are the same used from WINPRINT-SET-ATTRIBUTE op-code in WIN$PRINTER library routine.

  Configuration examples to define jpeg compression to 95% and PDF author to “MyName”:
  ```
  iscobol.print.attribute.jpeg=95
  iscobol.print.attribute.author=MyName
  ```
- Improved PDF compression especially when the same images are used several times in the print job.

## isCOBOL Compiler Enhancements

isCOBOL Evolve 2016 R1 release includes many changes on the isCOBOL Compiler that improve its power and flexibility and simplifies some areas of migration from other COBOLs and Java interoperating with isCOBOL.

### EasyLinkage feature to ease calling isCOBOL from Java

EasyLinkage feature eases the COBOL program integration with Java, automatically exposing standard Java methods to set and read COBOL linkage section items and call legacy COBOL programs or their entry-points.

The isCOBOL compiler has been enhanced with new compiler properties to generate a “bridge” interface class that exposes methods and properties to read and write COBOL linkage section items. With this new feature, Java developers can take advantage of COBOL programs without being forced to know COBOL variable types and structure of parameters in the linkage section.

Considering the linkage section inside a COBOL program coded this way:

```
program-id. MYPROG.
...
linkage section.
01 lnk-p1                  pic x(10).
01 lnk-p2.
   03 lnk-desc             pic x(4).
   03 lnk-num              pic 99v99.
   03 lnk-price            pic 9(5) COMP-3.
   03 lnk-occurs occurs 3.
       05 lnk-tel          pic x(15).
```

the compiler automatically generates a java class named linkMYPROG.  This class can be used from a Java source like this:

```
...
/* create the instance of the bridge program */
linkMYPROG myLink = new linkMYPROG();

/* set value of the variables to be passed to the cobol program */
```

```
myLink.lnkP1.set("hello");
myLink.lnkDesc.set("desc");
myLink.lnkNum.set(1.78);
myLink.lnkPrice.set(77);
myLink.lnkTel.at(1).set("occurs1");
myLink.lnkTel.at(2).set("occurs2");
myLink.lnkTel.at(3).set("occurs3");

...
/* invoke the cobol program */
  rc = myLink.run();
...

/* get value from variables returned from cobol program */
System.out.println("return-code = " + rc);
System.out.println("lnkP1 =[" + myLink.lnkP1.toString() + "]");
System.out.println("lnkDesc =[" + myLink.lnkDesc.toString() + "]");
System.out.println("lnkNum =[" + myLink.lnkNum.tofloat() + "]");
System.out.println("lnkPrice =[" + myLink.lnkPrice.toint() + "]");
System.out.println("lnkTel(1) =[" + myLink.lnkTel.at(1).toString() + "]");
System.out.println("lnkTel(2) =[" + myLink.lnkTel.at(2).toString() + "]");
System.out.println("lnkTel(3) =[" + myLink.lnkTel.at(3).toString() + "]");
```

To better configure the output generated by the EasyLinkage feature, isCOBOL compiler supports the following properties:

`iscobol.compiler.easylinkage=true` to generate the bridge program

`iscobol.compiler.easylinkage.prefix=prefix` to use a different prefix

`iscobol.compiler.easylinkage.package=packagename` to name the package

`iscobol.compiler.easylinkage.cut=val1` to cut val1 prefix in the linkage variable names

`iscobol.compiler.easylinkage.decoration=false` to avoid the use of decorated variable names

## Object Oriented syntax check improvements

A new compiler Warning has been added for method invocation used without cast when the signature is ambiguous. This new Warning helps identify lines that could be better written using a  cast, to be sure which method will be invoked at runtime.

Compiling the code in the following example:

```
repository.
   class math as "java.lang.Math"
working-storage section.
77  w-result pic 9(18).
procedure division.
   set w-result = math:>max(11, 22)
```

this compiler Warning will appear on the line where the “max” method is invoked:

--W: #222 The method signature is ambiguous, use a (better) cast: max;

This occurs because the java class has 2 methods with the same name and same number of parameters but different types:

```
static int    max(int a, int b)

static long   max(long a, long b)
```

To avoid the Warning the programmer can easily control which cast needs to be applied on the method invocation. In the previous example:

```
  set w-result = math:>max(11 as long, 22 as long)
```

or:

```
  set w-result = math:>max(11 as int, 22 as int)
```

Enhanced compatibility with other COBOLs

The isCOBOL 2016 R1 compiler introduces additional new options:

- -d1 enforces Binary data with length <= 2 to be stored in 1 byte. It could be useful during Micro Focus or other COBOL migrations when binary files are stored with this rule.
- -dcr to use Realia COBOL sign encoding.  It could be useful when the sign encoding used in existing files must be maintained.

The library routine named C$XML has been improved to support two new op-codes, simplifying migration from ACUCOBOL-GT. This library should be used for compatibility with ACUCOBOL-GT COBOL only.  isCOBOL users can take advantage of powerful and native ways to support XML streams.

Two new library routines, CBL_WRITE_SCR_CHARS and CBL_READ_SCR_CHARS, have been added to ease migration from Micro Focus COBOL.

DISPLAY POP-UP statement with CONTROL clause syntax has been enhanced to simplify migration from RM/COBOL.

Code example:

```
DISPLAY POP-UP WINDOW-CONTROL-BLOCK
        LINE W-LINE POSITION W-POS
        CONTROL W-CONTROL.
ACCEPT W-STAT FROM EXCEPTION STATUS.
CLOSE POP-UP WINDOW-CONTROL-BLOCK CONTROL "WINDOW-REMOVE".
```

Starting from isCOBOL 2016, BTRIEVE and Pervasive ISAM file are directly supported from a new interface class called com.iscobol.io.DynamicBtrieve. This native interface requires installation of BTRIEVE or Pervasive SQL native client libraries to work correctly.

ISMIGRATE data migration utility has been enhanced to support direct BTRIEVE/Pervasive SQL data migration.

## IsCOBOL Server Improvements

Starting from isCOBOL 2016R1, isCOBOL Server has been improved to provide more functionality and better performance.

### Lost connection message

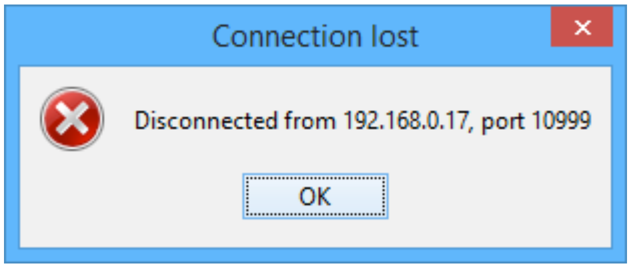
A message box 'Lost connection' is now automatically reported on Thin Client architecture if an abnormal session termination is detected. This assists end-users to identify a network stability problem.

If you would rather avoid this message, the new client option `–nodisconnecterr` has been added to the `iscclient` command:

```
iscclient –hostname ipserver –port 10999 –nodisconnecterr MYPROG
```

If no option is passed, by default the Connection lost error is displayed as shown in Figure 12, *Connection lost*.

**Figure 12**. Connection lost.


Connection lost

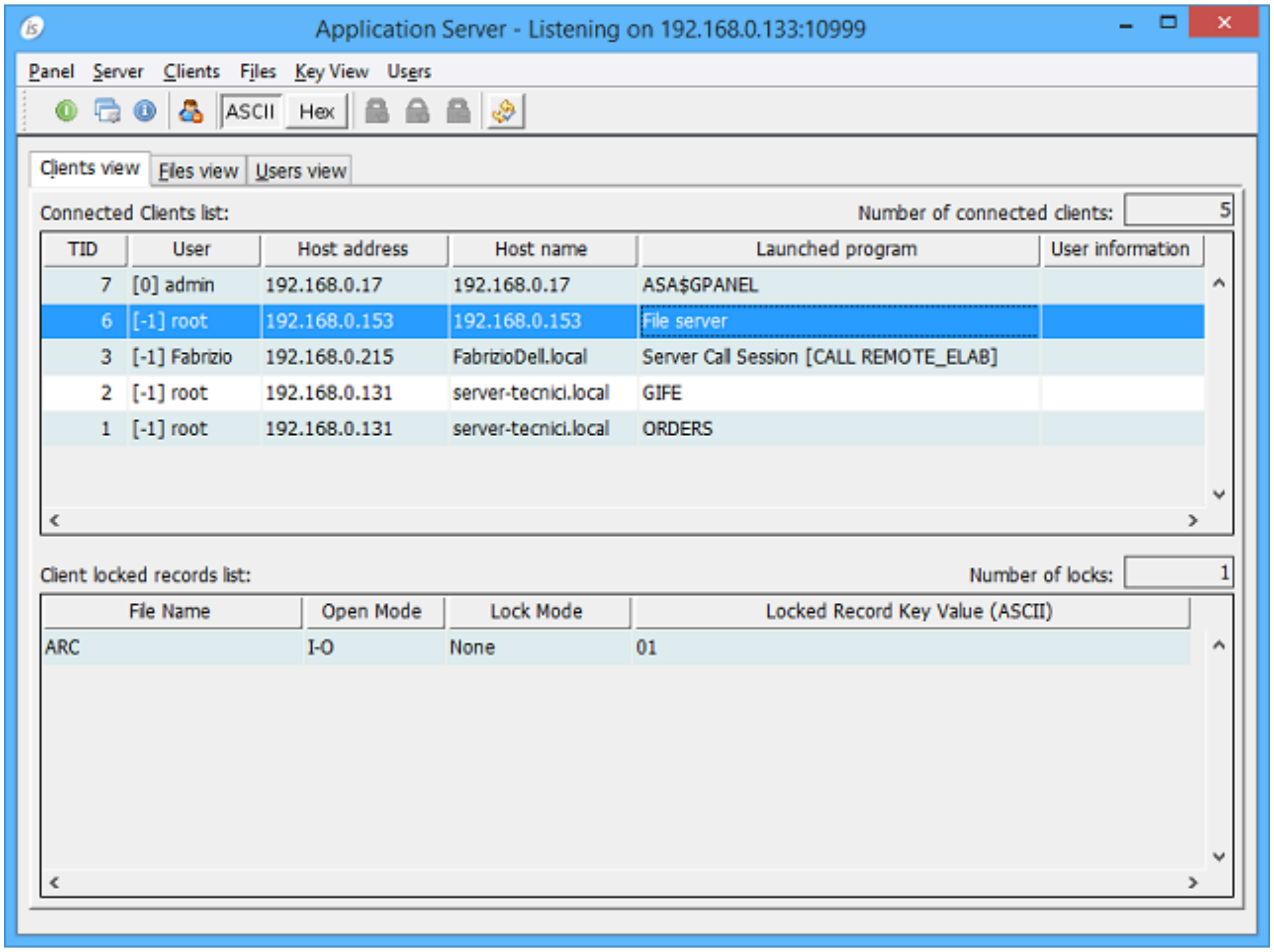
Disconnected from 192.168.0.17, port 10999

## Monitor users and locks of File Server

This new feature allows monitoring of active users and locks in the File Server through AS Panel. This allows administrators to have a complete list of connected users and locks if `iscobol.file.lock_manager` is set, as shown in Figure 13, *AS Panel*.

This same information can be retrieved from COBOL programs using the existing routines: A$LIST_USERS and A$LIST_LOCKS.

**Figure 13**. AS Panel.


Application Server - Listening on 192.168.0.133:10999
Panel Server Clients Files Key View Users
ASCII Hex
Clients view Files view Users view

Connected Clients list:

Number of connected clients: 5

| TID | User | Host address | Host name | Launched program | User information |
|---|---|---|---|---|---|
| 7 | [0] admin | 192.168.0.17 | 192.168.0.17 | ASA$GPANEL | |
| 6 | [-1] root | 192.168.0.153 | 192.168.0.153 | File server | |
| 3 | [-1] Fabrizio | 192.168.0.215 | FabrizioDell.local | Server Call Session [CALL REMOTE_ELAB] | |
| 2 | [-1] root | 192.168.0.131 | server-tecnici.local | GIFE | |
| 1 | [-1] root | 192.168.0.131 | server-tecnici.local | ORDERS | |

Client locked records list:

Number of locks: 1

| File Name | Open Mode | Lock Mode | Locked Record Key Value (ASCII) |
|---|---|---|---|
| ARC | I-O | None | 01 |

Remote calls in –cp mode

The remote calls now support a new protocol named iscp, to allow COBOL programs that use the C memory model (-cp compile option) to be called remotely.

Example:

```
iscobol.remote.code_prefix=iscp://192.168.0.101:10999
```

It's also possible to specify multiple remote settings to define different directories or servers where COBOL programs can be called.

Example:

```
iscobol.remote.code_prefix=iscp://server1:10999\niscp://server2:10999
```

## isCOBOL DataBase Bridge Improvements

isCOBOL Database Bridge, the tool that enables COBOL programs to use the  power of RDBMS without changing COBOL source code or learning ESQL, has been enhanced to support a new database. This increases the number of databases supported natively.

### New IBM Informix support

The new option -di allows generating EDBI routines for Informix. All EDBI routines are optimized to generate specific SQL syntax supported from IBM Informix DBMS.

### Improved performance

The EDBI routines now use a faster algorithm to execute COBOL's READ KEY SQL code on Microsoft SQL Server.  The EDBI RDBMS subroutines will benefit from this performance increase up to 30%.

### Multi-connection support for COBOL threads

Usually COBOL threads share DBMS connections unless the following property is used:

```
iscobol.jdbc.thread_connection=true
```

With this property turned on, each COBOL thread runs in a separate DBMS connection.

## isCOBOL EIS improvements

### Web Direct enhancements

Web Direct feature of isCOBOL EIS has been updated to give a new style to your fully interactive COBOL applications deployed to the cloud, without needing any change of basic application structure.

isCOBOL developers can take advantage of Bootstrap styles, Font Awesome icons and new CSS style screen section properties, to allow COBOL applications running inside a web browser to have a modern and stylish UI on desktop, tablet and mobile.

The following properties have been introduced to better manage CSS stylesheets:

To load an additional CSS you can use

```
iscobol.wd2.additional_stylesheet=filename
```

To force Bootstrap styling for all UI controls with no classes defined:

```
iscobol.wd2.style=bs
```

The compiler now supports a better management of CSS in the screen section with the addition of two new properties:

To set a css-style for UI controls and override any isCOBOL defaults use

```
css-base-style-name=style name
```

To assign a font-awesome icon to push-buttons

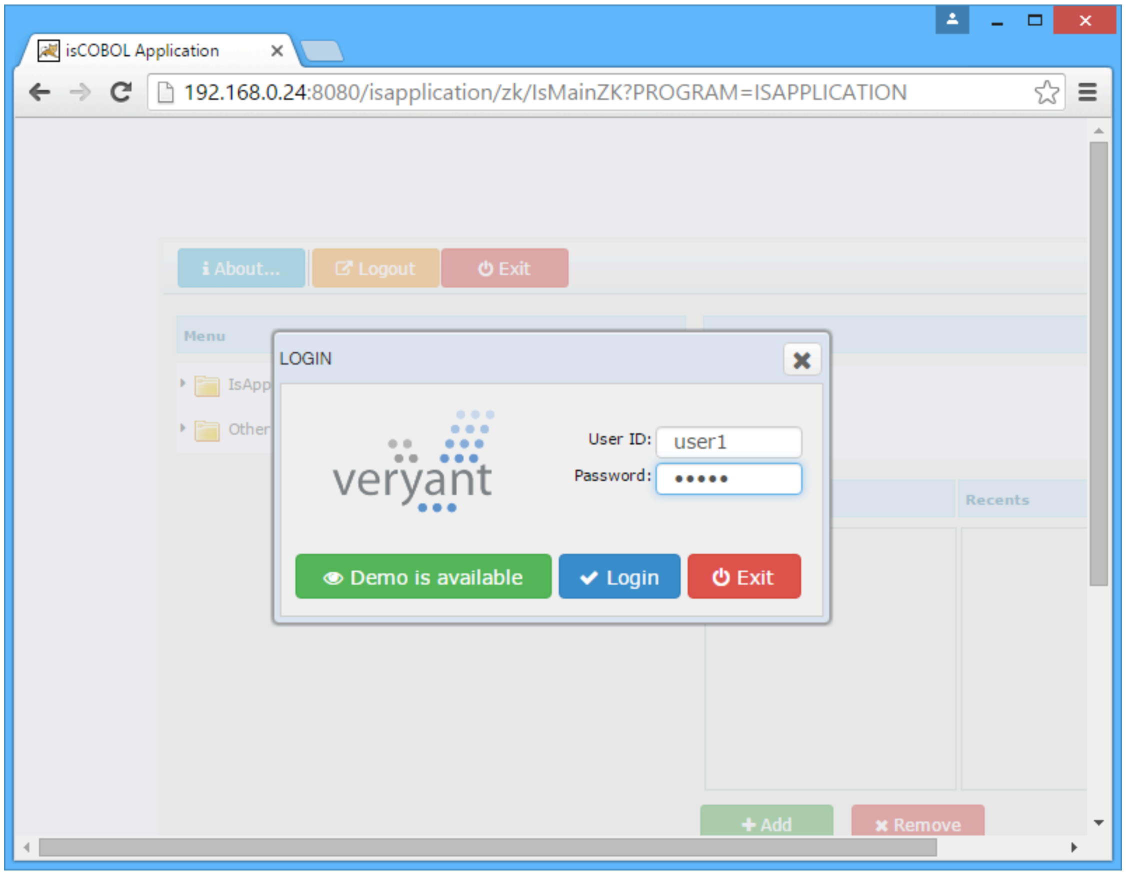
```
css-icon=icon=name
```

Code example:

```
03 PB-LOGIN Push-Button
   title "&Login"
   ...
   css-base-style-name "btn btn-primary"
   css-icon "fa-check".
```

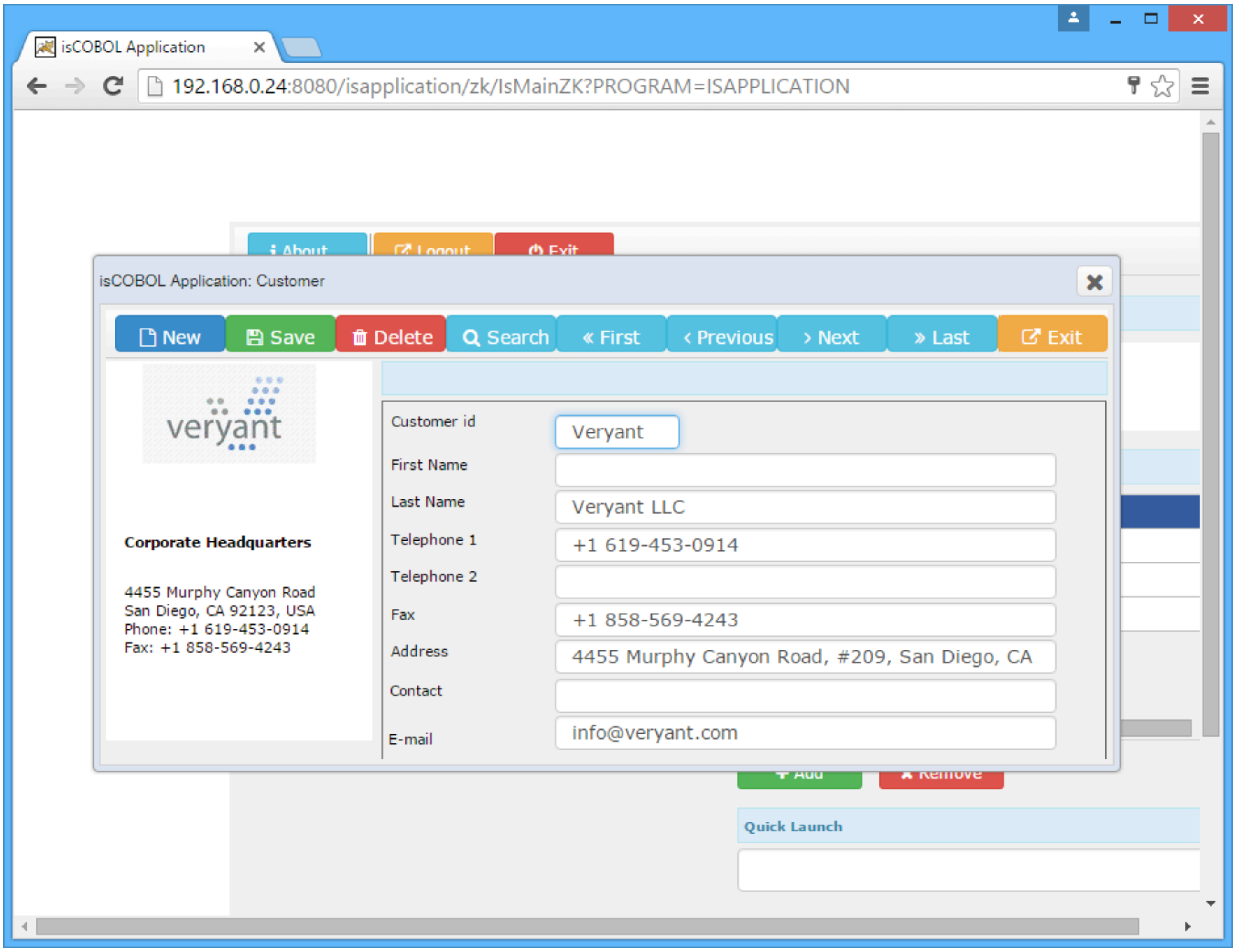
The result of the code above is shown in Figure 14, *Font Awesome style*

**Figure 14**. Font Awesome style.



With little adjustments on CSS styles and having bootstrap styling UI setting on configuration file, a COBOL application designed for desktop, can be deployed as depicted in Figure 15, *simple desktop application with Bootstrap styling.*

**Figure 15**. Simple desktop application with Bootstrap Styling.



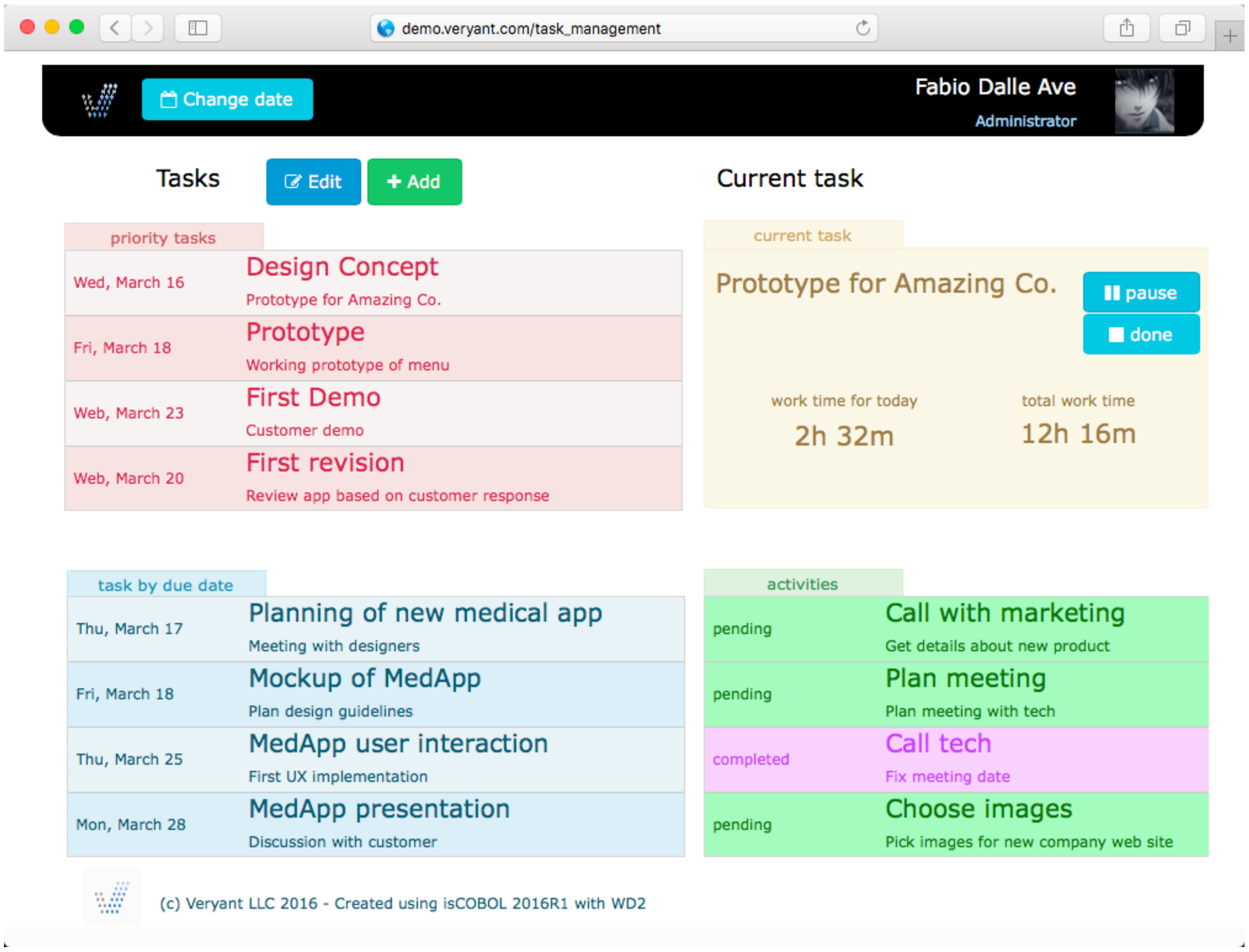
**Corporate Headquarters**

4455 Murphy Canyon Road
San Diego, CA 92123, USA
Phone: +1 619-453-0914
Fax: +1 858-569-4243

| | |
|---|---|
| Customer id | Veryant |
| First Name | |
| Last Name | Veryant LLC |
| Telephone 1 | +1 619-453-0914 |
| Telephone 2 | |
| Fax | +1 858-569-4243 |
| Address | 4455 Murphy Canyon Road, #209, San Diego, CA |
| Contact | |
| E-mail | info@veryant.com |

New capabilities provided in isCOBOL EIS 2016 let COBOL developers create, with a little effort on CSS styling, screen section based web applications that can be used on various devices. Figure 16 shows a *Tasks application* running on desktop browser while Figure 17 shows the same T*asks application* running inside tablet.

**Figure 16**, *tasks application*



priority tasks

| | |
|---|---|
| Wed, March 16 | Design Concept<br>Prototype for Amazing Co. |
| Fri, March 18 | Prototype<br>Working prototype of menu |
| Web, March 23 | First Demo<br>Customer demo |
| Web, March 20 | First revision<br>Review app based on customer response |

current task

Prototype for Amazing Co.

| work time for today | total work time |
|---|---|
| 2h 32m | 12h 16m |

task by due date

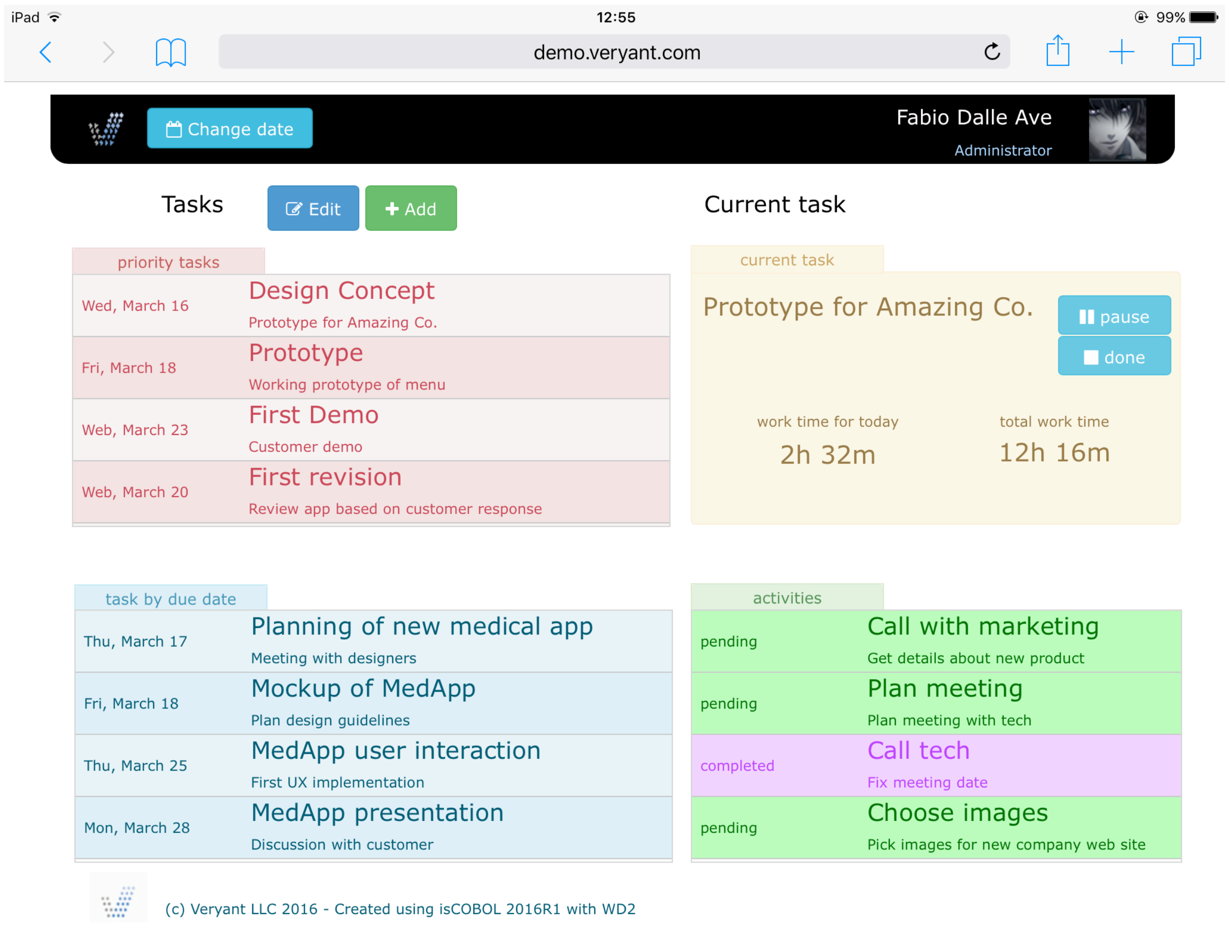
| | |
|---|---|
| Thu, March 17 | Planning of new medical app<br>Meeting with designers |
| Fri, March 18 | Mockup of MedApp<br>Plan design guidelines |
| Thu, March 25 | MedApp user interaction<br>First UX implementation |
| Mon, March 28 | MedApp presentation<br>Discussion with customer |

activities

| | |
|---|---|
| pending | Call with marketing<br>Get details about new product |
| pending | Plan meeting<br>Plan meeting with tech |
| completed | Call tech<br>Fix meeting date |
| pending | Choose images<br>Pick images for new company web site |

(c) Veryant LLC 2016 - Created using isCOBOL 2016R1 with WD2

**Figure 17**, *tasks application runs inside tablet*



Tasks

priority tasks

| | |
|---|---|
| Wed, March 16 | Design Concept<br>Prototype for Amazing Co. |
| Fri, March 18 | Prototype<br>Working prototype of menu |
| Web, March 23 | First Demo<br>Customer demo |
| Web, March 20 | First revision<br>Review app based on customer response |

Current task

current task

Prototype for Amazing Co.

| work time for today | total work time |
|---|---|
| 2h 32m | 12h 16m |

task by due date

| | |
|---|---|
| Thu, March 17 | Planning of new medical app<br>Meeting with designers |
| Fri, March 18 | Mockup of MedApp<br>Plan design guidelines |
| Thu, March 25 | MedApp user interaction<br>First UX implementation |
| Mon, March 28 | MedApp presentation<br>Discussion with customer |

activities

| | |
|---|---|
| pending | Call with marketing<br>Get details about new product |
| pending | Plan meeting<br>Plan meeting with tech |
| completed | Call tech<br>Fix meeting date |
| pending | Choose images<br>Pick images for new company web site |

(c) Veryant LLC 2016 - Created using isCOBOL 2016R1 with WD2

Configuration to activate stateless mode

A COBOL Services can be designed to be stateless or stateful, according to COBOL developers' needs. In case of migration from CGI COBOL programs, which are stateless by design, this could be useful to force isCOBOL services to run stateless, even if not designed to be.

A new property `iscobol.http.stateless` has been added to force services to run stateless.

Configuration examples:

```
iscobol.http.stateless=true
```