# extend Interoperability Suite 10.1.0

## Release Notes

**Micro Focus**
**The Lawn**
**22-30 Old Bath Road**
**Newbury, Berkshire RG14 1QN**
**UK**
**http://www.microfocus.com**

**2017-01-23**

# Contents

# extend Release Notes

These release notes contain information that might not appear in the Help. Read them in their entirety before you install the product.

**Note:** This document contains links to external web sites. Micro Focus cannot be responsible for the contents of the website or for the contents of any site to which it might link. Web sites by their nature can change very rapidly and although every attempt is made to keep links up-to-date, Micro Focus cannot guarantee that external links will always work as expected.

# extend System Requirements

**Note:** This product includes software developed by the University of California, Berkeley and its contributors.

### Hardware Requirements

extend software has the following requirements:

**For Windows:**

- The amount of disk space needed to install the ACUCOBOL-GT development system is typically less than 35 MB.
- AcuBench® requires at least 20 MB for installation.
- You need an additional 40 MB to install all of the other extend products.
- Use of .NET controls with the runtime and thin client requires .NET Framework 4.0.

**For all other platforms:**

- The amount of disk space needed to install all extend products is typically less than 35 MB.

### Supported Operating Systems

**Note:** AcuBench no longer supports Windows XP or Windows Server 2003 (or any Windows platforms that pre-date these). This is because AcuBench utilizes a number of third-party libraries that have ceased to be supported on those platforms.

For this release, platform support has been added for Windows Server 2016.

For a full list of the supported operating systems, check the Product Availability section on the Micro Focus SupportLine Web site: *http://supportline.microfocus.com/prodavail.aspx*.

### Additional Requirements

**AcuServer:**

- Each server machine must be networked to UNIX, Linux, or Windows clients with TCP/IP. TCP/IP is not sold or supplied by Micro Focus.
- All servers must have a copy of the AcuServer license management file.
- Windows clients can run any TCP/IP software that uses a `WINSOCK2` compliant `ws2_32.dll`.
- Unless you have an unlimited license for AcuServer, all UNIX servers must run the current version of `acushare`, which is included on the AcuServer distribution media.
- All servers must have a copy of the license file activated by the product installation script. This file is named `acuserve.alc`.
- Client machines must have an ACUCOBOL-GT AcuServer-enabled runtime. All Windows runtimes Version 5.0 and later are AcuServer-enabled. To verify that your UNIX runtime is AcuServer-enabled, type `runcbl -v` in a Command prompt and look for this line.

  ```
  AcuServer client
  ```
- Servers being accessed by the ACUCOBOL-GT Web runtime must have a multiple-user ACUCOBOL-GT runtime license that accommodates each concurrent user that is anticipated. (If you anticipate 100 concurrent users of the Web runtime, you need a 100-user runtime license on the server in addition to the AcuServer license file. Alternatively, runtime users can install a local or network floating license for the runtime themselves.

**AcuBench:**

- Intel Pentium III CPU, 300 MHz; Intel Pentium IV, 2 GHz recommended
- 128 MB of RAM recommended
- 120 MB of available hard disk space recommended
- mouse
- 800 x 600 VGA display or better; 1024 x 768 VGA display recommended

**AcuSQL:**

- Your COBOL application must run on a Windows system or a UNIX system supported by Micro Focus. Unless otherwise indicated, the references to Windows in this manual denote supported Windows operating systems. Where necessary, individual versions of those operating systems are referred to by their specific version numbers.
- AcuSQL must be installed with the ACUCOBOL-GT development system on your Windows or UNIX system.
- If using a database other than Microsoft SQL Server, you must have a working ODBC level 2 API connection to your database, including any required networking software support.
- For SQL Server, if running the AcuSQL interface to Microsoft SQL Server, you must have the SQL Server client software from Microsoft. Use the Query Analyzer to see if the SQL Server client software from Microsoft is on your system. For information on opening the Query Analyzer, see the SQL Server client documentation. If the Query Analyzer opens and you are able to connect to the database, the client libraries are most likely all present. Your SQL Server data source may be hosted on one or more of the supported server operating systems.
- If you are running the AcuSQL interface to MySQL, you must have the following software:

    - MySQL 5.0 Database Server Version 5.0.18 or later (Generally Available release). Testing was done with MySQL 5.0.18 Standard.
    - MySQL Connector/ODBC Version 3.51.11 or later (Generally Available release). Testing was done with the `libmyodbc3-3.51.12.so` library. This file is available from *http://dev.mysql.com*.

    You can check the version of your server by connecting using `mysql`. The version prints upon connection. For example:

    ```
    [testing ]: mysql
    Welcome to the MySQL monitor. Commands end with ; or \g.
    Your MySQL connection id is 29 to server version:
    5.0.18-standard
    Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
    ```

    Once in MySQL, you can also use the following:

    ```
    mysql> select version();
    +-----------------+
    | version()       |
    +-----------------+
    | 5.0.18-standard |
    +-----------------+
    1 row in set (0.09 sec)
    ```

- If your application accesses DB2 data, IBM's DB2 Connect™ software is recommended. Access to DB2 databases has been tested with DB2 Connect. However, any vendor's properly configured ODBC level 2 API connectivity software should work. Your DB2 data source may be hosted on one (or more) of the supported operating systems.

**Acu4GL (for ODBC) driver requirements:**

Your ODBC driver must include the following functions:

- all Core ODBC driver functions
- the Level 1 function SQLColumns
- the Level 1 function SQLTables

Depending on the method of record locking you choose, your driver may also need to support some of the following function calls:

- SQLSetStmtOption
- SQLSetScrollOptions
- SQLExtendedFetch
- SQLSetPos

See A_ODBC_LOCK_METHOD in the extend online help for more information.

To test the capabilities of your ODBC driver, we have included a driver test program on your Acu4GL for ODBC installation disks. You can also consult your driver documentation to ensure that it meets these requirements.

# Windows Installation

## License Files for Windows

You may request a license file for one or more users. The number of users (user limit) is set in your license agreement with Micro Focus.

When you receive your products, the package includes product codes and product keys for every product you ordered. You must have the product codes and product keys to create the license file.

When you install or update your license file, place it in the appropriate directory for your version of Windows. The location is:

```
C:\ProgramData\Micro Focus\extend\x.x.x\x86\product-license.alc
```

(Where *x.x.x* is the product version number.)

**Note:** This location is new to extend 9.0 and later, and differs from past versions of extend products. See *Appendix C - Changes Affecting Previous Versions* in the *ACUCOBOL-GT Appendices* manual for details.

When you have multiple users or products, you may copy the license file onto each machine containing the corresponding product or place the products and license file on a shared drive. Each product must be able to locate its license file in order to function.

**Note:** The ACUCOBOL-GT Transaction Server runtime license file is named `wrun32.1lc`. If you are using other *extend* products, the license files must be concatenated into a special file.

## The Activator Utility

The Activator Utility automates the process of creating a license file.

During installation, select **Install License Activator** on the Installation Settings page to install the Activator Utility (`activator.exe`). It is installed in the `\AcuGT\bin` sub-directory of the installation directory.

**Note:** Always use the version of the Activator supplied with the version of the product you have installed.

To create the required license files during installation, select **Launch License Activator** on the Installation Settings page. This will launch the Activator Utility when the selected products have installed, enabling you to enter the product code and key pairs required to create the license files.

You can also create license files after the installation by running the Activator Utility (`activator.exe`) from the location detailed above or from the Start menu.

During product installation, if the Activator detects the presence of an existing license file, the extension of the existing file is changed before a new license file is created. For example, `runcbl.alc` is renamed `runcbl.al!`. If the Activator is unable to rename the existing license file, it quits with an error message, and no new license file is created. If a license file with the back-up extension already exists, the Activator attempts to overwrite it. If that fails, the Activator quits with an error message and no new license file is created. On Windows platforms, file attributes such as Read Only are also preserved.

## Changing or Updating a Windows License File

If you need to alter your license file information, contact your Micro Focus *extend* representative for updated product code(s) and product key(s). When you receive them, launch the Activator and enter the new information as prompted.

**Note:** If you have copied a license file to a non-default directory, remember to replace that file with a copy of the updated license file.

# Installation on Windows Platforms

The Windows installation requires little interaction; the setup program copies the files into a directory you designate, or `C:\Program Files\Micro Focus\extend` *`x.x.x`* (where *x.x.x* is the version number) by default. On 64-bit machines, 32-bit executables are installed to `C:\Program Files (x86)`.

Before you run an installation, you should ensure that you have your product codes and product keys, and the product media to hand. Refer to the appropriate installation instructions below.

Also, you should ensure that you do not have another version of the extend Interoperability Suite referenced in the PATH system environment variable, as having more than one version specified may cause unexpected results.

After the installation is complete, if you have installed both the compiler and runtime, you can begin to compile and execute your COBOL programs. Basic compilation and execution techniques are described in *Compiling Your Programs* and *Running Your Programs*.

**Note:** If you move or delete any `.dll` files that have been installed, your products may not run as expected.

AcuBench and AcuXDBC™ must be installed locally (on the client). Server-side products such as AcuServer™, AcuXDBC™ Server and AcuConnect® are to be installed only on server machines.

## Installation Steps

**Attention:** The installation is supplied in two formats: `.exe` and `.msi`. On the installation CD, the `.exe` is located in the top level folder, and the `.msi` is located in the `msi` folder. If you plan to install Xcentrisity Business Information Server, or start AcuServer or AcuConnect from the installer, you must run the installation with administrator privileges, from an account that is in the Administrator group.

If you install from the CD, it will automatically run the `.exe` version with administrator privileges. If you run the `.msi`, it does not automatically run with administrator privileges. To run the `.msi` with administrator privileges, click **Start**, and in the **Search programs and files** field, type `cmd.exe`, then in the list displayed, right-click **cmd.exe** and select **Run as administrator**: this opens a command prompt with administrator privileges, where you can run the install by typing the full path name of the `.msi` file.

Follow these steps to install your products.

1. Insert the product CD, or use the provided link to download the installation package from the Micro Focus website.

   If you are using the CD, it runs the `.exe` version automatically.
2. If the `.exe` version does not start automatically, or you are using the downloaded installation package, do one of the following:

   - Navigate to the CD drive, then double-click the `.exe`.

- Navigate to the location of the downloaded file, then double-click it. (See the above note if you are running the `.msi` version.)

The installation starts.

3. On the Welcome page, click **Next**.
4. On the **End-User License Agreement** page, select **I accept the terms in the License Agreement**, then click **Next**.
5. On the **Select Installation Folder** page, click **Browse** and select installation directories for 32-bit and 64-bit (if applicable) products. Alternatively, you can accept the default location(s), then click **Next**.
6. On the **Product Selection** page, select the required products, then click **Next**.

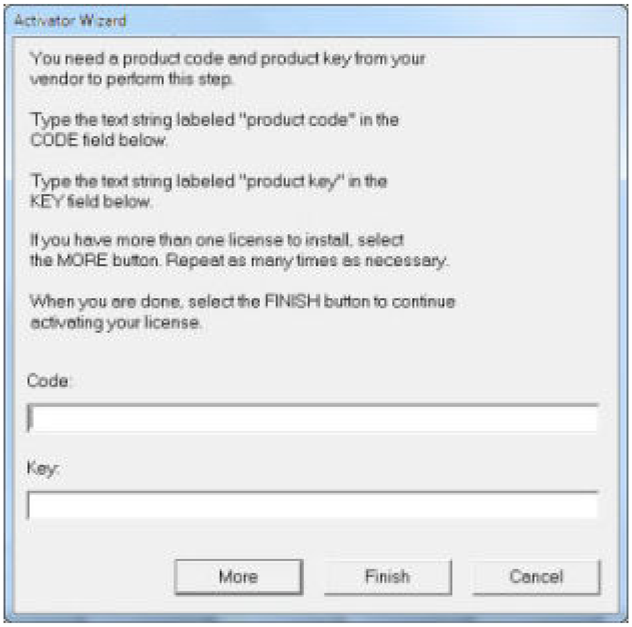
   You must have product codes and keys to activate each product; however, you can install all products now, and activate those products for which you do not have a license at a later date. (Be aware that if you attempt to use products for which you do not have a license, you may receive error messages indicating that no license file is available.)

   **Note:** Ensure you select **Documentation** on this page to install and access the product user guides.
7. On the **Installation Settings** page, select one or more of the following, then click **Next**:

| Select | To |
|---|---|
| Install License Activator | Install a copy of the License Activator |
| Launch License Activator | Launch the License Activator at the end of the installation process |
| Start AcuServer | Start AcuServer as a service |
| Start AcuConnect | Start AcuConnect as a service |

**Note:** Starting either AcuServer or AcuConnect as a service will only occur if their pre-requisites are satisfied before you run the installer, and only if you have run the `.exe` or `.msi` version of the installer with administrator privileges (see note above).

If you selected **Launch License Activator** on the **Installation Settings** page, the **Activator Wizard** appears.



Activator Wizard

You need a product code and product key from your vendor to perform this step.

Type the text string labeled "product code" in the CODE field below.

Type the text string labeled "product key" in the KEY field below.

If you have more than one license to install, select the MORE button. Repeat as many times as necessary.

When you are done, select the FINISH button to continue activating your license.

8. Type your first product code and key in the appropriate fields.

The License Activator is case-insensitive and displays only uppercase characters. It also ignores embedded spaces and separating characters. Product codes and keys do not contain the letters "O" or "I".

**Caution:** If you have a license for both the Windows runtime (`wrun32.exe`) and an Alternate Terminal Manager (ATM) runtime (`run32.exe`) for the same machine, be aware that the Activator Utility creates a license file named `wrun32.alc` for each of them. To avoid a situation in which the Activator Utility overwrites the license file for the second runtime:

- Make a backup copy of the Windows runtime license file prior to creating (and renaming) the ATM runtime license.
- Create the ATM runtime license and rename it to match the executable (change `wrun32.alc` to `run32.alc`) before creating the Windows runtime license.

**9.** If you have more than one code and key pair to enter, select **More** after typing the first code/key pair. Repeat this process until you have entered all code and key pairs, then click **Finish**.

Each time you press **More**, the License Activator creates a separate license file for the product code and key you entered and returns you to the code and key entry screen.

**10.** Click **Finish** on the **Installation Complete** page to complete the installation.

**Note:** If license activation was successful, but you get a message during product startup indicating that the license file cannot be found, the license file may not be in the correct directory. The License Activator determines where to place the license file based on entries in the Windows registry. If no registry entry is found, the license file is placed in the same directory as the License Activator executable file, which is the `\AcuGT\bin` sub-directory of the default installation directory. If this is not the location of the product's executable file, move the license file to the directory containing the corresponding executable file; for example, move `wrun32.alc` to the directory containing `wrun32.exe`.

# Windows 64-bit Installations

There are 64-bit versions of most extend products. These 64-bit versions are installed using a separate 64-bit version of the installer. The installation process follows the same steps as described in *Installation on Windows Platforms*, with the following notable exceptions.

When running the 64-bit installer, if no 64-bit version exists for a selected product (for example, AcuBench), the 32-bit version is installed.

Products such as AcuConnect and AcuServer have 32-bit and 64-bit versions, and both are installed if you select these products during installation. You can also decide which version of the product to start on completion of the installation.

The AcuXDBC product is broken down into three installations: for the Data Interface, you can install both the 32-bit and 64-bit versions; and for the AcuXDBC Server and Enterprise Edition, you must chose which version to install.

By default, all 64-bit product versions are installed in the `Program Files` directory, and 32-bit product versions (and any supporting non-64-bit tools) are installed in the `Program Files (x86)` directory; although, you can change these locations during the installation. All the 32-bit versions are fully supported and functional in a 64-bit environment.

**Remember:** When running the license activator after the installation, the 64-bit version of the Activator utility is run, which installs license files into both the 32-bit and 64-bit directories. When running the Activator utility from the command prompt, make sure you are using the 64-bit command prompt to ensure the correct licenses are generated and placed in the correct locations; otherwise, if the 32-bit Activator utility is run, only license files for 32-bit products will be generated.

# BIN-REDIST and REDIST Installation Directories

The extend Windows distribution contains two directories: `BIN-REDIST` and `REDIST`.

`REDIST` contains thin client files that should be distributed along with the thin client.

`BIN-REDIST` contains Microsoft Redistributable files. These files are required in cases where the ACUCOBOL-GT `bin` directory and runtime are placed on a shared drive and users then map to that drive. The `BIN-REDIST` directory should be placed inside the shared `bin` directory.

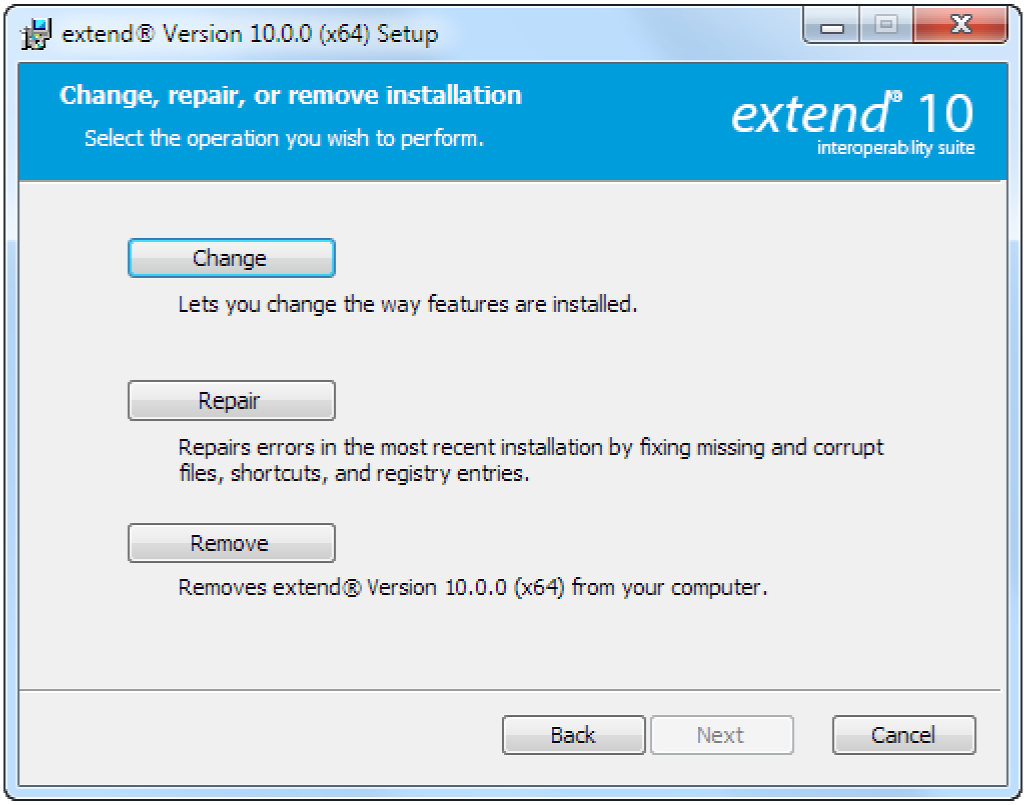
# Uninstalling or Modifying Your Installation

You can uninstall or amend your current installation in one of two ways: you can run the installer again, or use the Programs and Features section in Windows. Both enable you to run the Installation Wizard, where you can perform the following program maintenance:

**Change** Enables you to add or remove products to and from your current installation. Any products that were already checked that you uncheck are uninstalled. Any additions are installed using the default installation path (`C:\Program Files\Micro Focus\extend` *`x.x.x`* - where *x.x.x* is the version number). There is no option to change to a non-default location, but you can overcome this with some products by copying the installed files from the default location to your preferred location. Note that this method will not work for AcuXDBC, AcuBench, and any server products that are registered as services.

**Repair** Enables you to reinstall the currently installed products.

**Remove** Enables you to remove all products of your installation.



extend® Version 10.0.0 (x64) Setup

**Change, repair, or remove installation**

Select the operation you wish to perform.

Lets you change the way features are installed.

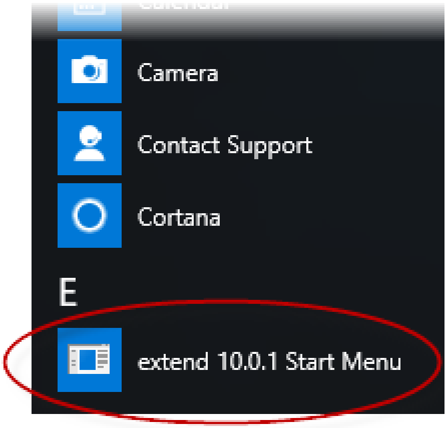
Repairs errors in the most recent installation by fixing missing and corrupt files, shortcuts, and registry entries.

Removes extend® Version 10.0.0 (x64) from your computer.

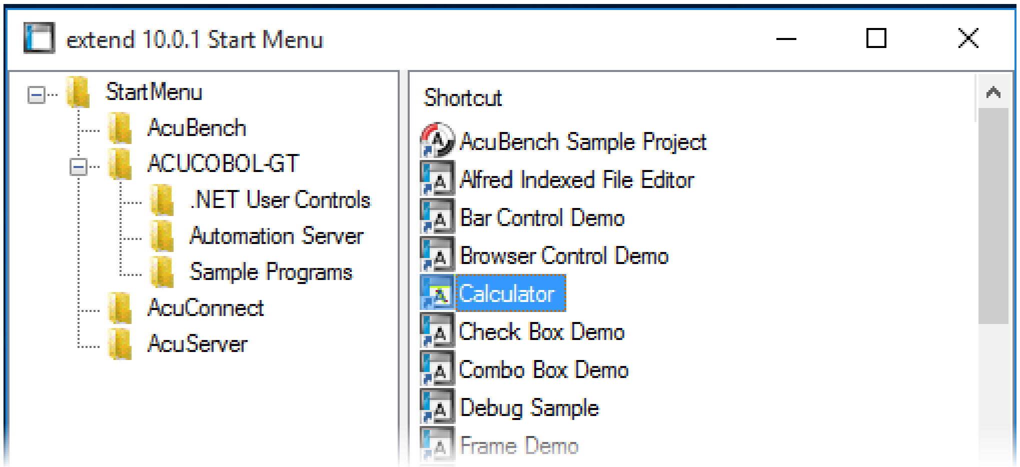
# Launching extend Products on Windows 8 or Later

Since the introduction of Windows 8, the Windows program menu does not display programs in a hierarchical way. For the extend Interoperability Suite, this would mean that all the programs relating to AcuBench, ACUCOBOL-GT, AcuConnect, etc... would be displayed as a flat list, and it would be difficult to distinguish which programs were related to which products.

In order to show the programs grouped within the products they belong to, the extend Start Menu has been introduced. After installation of the extend Interoperability Suite, there will be one entry in the Windows programs menu - **extend x.x.x Start Menu** (where *x.x.x* is the version number of the installed product):



Launch this application to display a hierarchy that looks similar to the Windows program menus prior to Windows 8. Simply navigate the folders displayed in the left-hand pane to display shortcuts to the related programs in the right-hand pane. Select the required shortcut: at which point, the extend Start Menu closes and the program is launched.



# Compiling Your Programs

The ACUCOBOL-GT compiler is most easily used via AcuBench. However, it is also possible to compile from the command prompt or the Windows Start menu, to establish an association between your source files and the compiler executable, or to set up one or more icons to compile individual files. For information regarding creating file associations and setting up icons, see your Windows documentation.

When you compile, if the system returns "Bad command or file name," you did not add the location of the compiler to the PATH environment variable, or you have not rebooted since installing the software.

You can check the current definition of PATH by typing `path` and pressing **Enter** at the command prompt. The default path is `C:\Program Files\Micro Focus\extend x.x.x\AcuGT\bin` (or `c:\Program Files (x86)\...` for 64-bit installations). If you do not update PATH with this entry, you must type the full path and program name to the compiler each time you compile; alternatively, you can create a `.bat` file.

Once you have verified that the compiler is in your PATH (or you have decided that you will specify the full path every time you compile), you are ready to compile your programs. Refer to *ACUCOBOL-GT User's Guide* > *Compiler and Runtime* > *Using the Compiler* for a complete list of compile options. You can also display a complete list of options by running the compiler command:

```
ccbl32 -help
```

One commonly used option is `-o`. This option is used to specify the name of the output object file. Note that ACUCOBOL-GT uses the naming convention `.acu` to indicate an ACUCOBOL-GT object file. If `-o` is not used, the compiler will name the file *`source-name`*`.acu`. If your source includes COPY files, you can copy the COPY files to your current working directory, specify their location with the `-Sp` option followed by the complete path of one or more directories, or set the COPYPATH environment variable. A typical compilation command might look like:

```
ccbl32 -Sp c:\work\lib -o sample.cbx sample.cbl
```

# Running Your Programs

After a successful compilation, the resultant object file is ready for immediate execution (no link step is required). To run your program, make a note of the full path to your object file and return to the Windows desktop.

There are a variety of ways to run your program under Windows. The basic methods are:

- Running a command from the **Start** menu.
- Placing an icon for each program in a Program Group or folder, and then starting the program by double-clicking its icon.
- Naming your COBOL object files with a common file extension and then associating the extension with the runtime. After the association is established, you can run a program by double-clicking on the name of the file as it appears in Windows.
- Using the **wrun32** (or **crun32** if you are using the console runtime) command at the command prompt.

This document describes how to run a program from the **Start** menu only (although, the command line option is almost identical). The advantage of this method is that it requires no special setup. However, it is more time-consuming (than other methods) because you must type the command line options every time you execute the program. For greater convenience, it is best to set up a program icon: to do this, or to create a file association to the runtime, please refer to your Windows documentation.

1. Click the **Start** button.
2. On the Start menu, in the entry field, type the name of your ACUCOBOL-GT Windows runtime.

   The default name is `wrun32.exe`. If you changed the name of the runtime executable, enter that name.
3. Following the name of the runtime, enter any runtime options required, followed by the path and name of the COBOL executable program you want to run.
4. After you have entered the complete command line, press **Enter** to execute the program.

**Note:** If you did not add the `\bin` directory path to the PATH system environment variable (typically, that is: `c:\Program Files\Micro Focus\extend` *`xxx`*`\AcuGT\bin`), you will need to specify the full path to the runtime (`c:\Program Files\Micro Focus\extend` *`xxx`*`\AcuGT\bin\wrun32.exe`) each time in the command (and not simply `wrun32.exe`).

You can use the COBOL configuration variable DEFAULT_PROGRAM to specify the name of the program to execute when no program is specified on the command line. See *Appendix H* in the ACUCOBOL-GT documentation set for more configuration information.

# Printing and Spooler Issues

ACUCOBOL-GT includes extended support for printing under Windows. In addition to the basic print spooler access procedures described below, the WIN$PRINTER runtime library routine provides easy access to extended Windows print spooler capabilities. See the entry for "WIN$PRINTER" in *ACUCOBOL-GT Appendices Guide* > *Appendix I*. Rules for printer handling are illustrated in the *ACUCOBOL-GT User's Guide* > *Compiler and Runtime* > *Filename Interpretation* > *Assigning Files to Printers.*

Under Windows, you may print directly to the printer by defining PRINTER in the configuration file as "-D PRN". Be aware that this does not prevent other programs from printing at the same time and as a result you may get intermixed pages.

You may also print using the Windows spooler, even if your reports have embedded control codes. The spooler allows many programs to create print files at the same time, and also allows the user to do other tasks while the report is being printed.

**Note:** The configuration variable WIN_SPOOLER_PORT allows you to divert printer output to a file or port through the Windows print spooler. For more information, see *ACUCOBOL-GT Appendices Guide* > *Appendix H* of the ACUCOBOL-GT manual set.

Before you assign your print file to the Windows spooler, you must decide whether you want to control the format of each page directly (with embedded control codes) or whether you want the print spooler to format the pages.

## Spooler Formatting

There are two ways to use the Windows spooler to format your print file: "-P SPOOLER" and "-Q <*printername*>". See Direct Control for information on controlling the formatting yourself.

### -P SPOOLER

If you want to use the default printer and font, simply assign your print file to "-P SPOOLER". For example, to assign "PRINTER1" to the spooler, enter the following line in your COBOL configuration file ("CBLCONFI"):

```
PRINTER1  -P  SPOOLER
```

By default, the runtime system assigns the "PRINTER" device to the spooler. You may change this in the configuration file by assigning "PRINTER" to some other name.

When the runtime opens a file assigned to "-P SPOOLER", it automatically initiates a job with the Windows spooler and constructs print pages in accordance with your program. The runtime uses the default printer and font. If the user looks for the job in the spooler, it is named with the current title of the ACUCOBOL-GT window.

**Note:**

The Windows spooler operates by drawing your report on each page. It constructs its own control codes to handle formatting. If you assign your print file to "-P SPOOLER" and your file contains device-dependent control sequences (such as those used to shift to a condensed font, or to print a form and then fill it in), the codes will be passed to the spooler as data and thus will not be interpreted correctly. If you have reports that depend on embedded control codes, you should print those directly to the device, or assign the print file to "-P SPOOLER-DIRECT," as described below.

### -Q <printername>

If you want the Windows spooler to format the pages of your report, but you want to use a particular printer, assign your print file to:

```
PRINTER1 -Q \\printername
```

in the configuration file (`CBLCONFI`). *Printername* is the printer designation as given in the **Devices and Printers** screen. The name may be up to 80 characters long and contain embedded spaces. The name may not include the semicolon character (;) or be surrounded by single or double quotes. The pages are printed in the manner described in "-P SPOOLER", above. The sample programs `graphprn.cbl` and `prndemox.cbl` contain examples of these functions.

To determine a valid printer name, use the WIN$PRINTER library routine to obtain the name of the desired printer. (This is described in *Appendix I* under the WINPRINT-SET-PRINTER operation code in "Specifying a Printer".) Then add the following line to your code:

```
MOVE "-Q \\printername" TO WS-PRINTER-NAME.
```

When the runtime opens a file assigned to "-Q <*printername*>", it sets the Windows print spooler to use this printer. The printer driver must be installed on the computer from which you print. If *printername* is not recognized by the runtime, a dialog box allows you to choose a printer manually.

**Note:**

If you want to access a printer using a UNC path, you have to print directly to the printer by defining PRINTER as "-D PRN". If you use the UNC path, Windows formatting is not supported.

## Direct Control

If you want to control the format of the printout yourself using embedded control codes, simply assign your print file to `-P SPOOLER-DIRECT` or to `-Q <printername>` using the `DIRECT=ON` option. For example, to assign the print job "PRINTER1" to the spooler and retain direct control over formatting, enter the following line in your COBOL configuration file (`CBLCONFI`):

```
PRINTER1  -P  SPOOLER-DIRECT
```

Or, use the following command to assign PRINTER1 to the spooler for printing to a specific printer while retaining direct formatting control:

```
PRINTER1 -Q printername;DIRECT=ON
```

Both of these methods cause the print job to be sent to the printer via the Windows spooler, but the program does not use the spooler to format the pages. You must use embedded control codes to handle formatting (much as you would under UNIX if you used the UNIX spooler).

When using the `-P SPOOLER-DIRECT` option, you may use the WIN$PRINTER library routine to choose a printer, but because you completely control the printer, the various options provided by WIN$PRINTER are ignored. For example, WIN$PRINTER does not set the page size, page orientation, or font. Information returned from WIN$PRINTER, such as number of lines and columns on the page, may not be accurate and should not be used. This subject is discussed in detail in Appendix I "*Library Routines*" of the ACUCOBOL-GT manual set.

Because some print drivers do not flush the last page, be sure to end your last page with a form-feed (for example, WRITE ... BEFORE ADVANCING PAGE). This ensures that all pages are printed. The ACUCOBOL-GT runtime ensures that no extra blank pages are printed at the end.

If you code WRITE...AFTER ADVANCING PAGE instead of WRITE...BEFORE ADVANCING PAGE, you might receive a blank last page. This is because a blank line written on the new page causes the Windows subsystem to flush the page for some print drivers. ACUCOBOL-GT ensures that entirely empty lines are not sent to the device (only the form-feed will be sent). But it is essential that:

- You have specified trailing space removal in your COBOL code (the default for print files).

- You have set the configuration option MIN-REC-SIZE to "0".

Your other option is to specify WRITE... BEFORE ADVANCING PAGE to avoid this potential problem.

If the user looks for the job in the spooler, it is named with the current title of the ACUCOBOL-GT window.

# Printing Multiple Jobs Simultaneously

If you need to print multiple jobs at the same time, you must open multiple File Descriptors that point to "-P SPOOLER" or "-P SPOOLER-DIRECT" simultaneously. For example, you may have two simultaneous print jobs:

```
SELECT FIRST-FILE
       ASSIGN TO PRINTER "-P SPOOLER".

SELECT SECOND-FILE
       ASSIGN TO PRINTER "-P SPOOLER".

..PROCEDURE DIVISION.

..

       OPEN OUTPUT FIRST-FILE.
       OPEN OUTPUT SECOND-FILE.
```

and both will print to the default Windows printer without interfering with each other. You can call WIN$PRINTER USING WINPRINT-SETUP before one or both of the OPEN statements. Each file may have individual file status variables or may refer to a common file status variable.

This does not mean that you can open a single File Descriptor multiple times. For example, the following will return file status indicating that the file is already opened:

```
SELECT FIRST-FILE
       ASSIGN TO PRINTER "-P SPOOLER".

..

PROCEDURE DIVISION.

..

       OPEN OUTPUT FIRST-FILE.
       OPEN OUTPUT FIRST-FILE.
```

This is normal behavior and is consistent with the way file handling is implemented in COBOL and in other programming languages.

If you are using only the verbs OPEN, CLOSE, and WRITE, no further changes to your code are needed. If you are using WIN$PRINTER functionality (other than WINPRINT-SETUP) you will need to specify which print job is affected. This can be done in two ways:

1. The simplest way is to execute the WIN$PRINT operation immediately after an OPEN or WRITE statement on the intended job. Every execution of OPEN and WRITE sets the current job as the default so that subsequent activity using WIN$PRINTER is automatically directed to the job that was last accessed with an OPEN or WRITE statement.

   In this situation, if you have multiple jobs running, and you close one of them, the runtime switches to the next job in the list. For example, if you are printing jobs 1, 2, and 3, and you close job 2, the close command sets the current job to 3. If there is no job 3, the runtime attempts to set to the job that preceded the closed job (which in this case is job 1). If there are no jobs, the current job is initialized.
2. The other method is to use the WINPRINT-SET-JOB operation of the WIN$PRINTER library routine. This operation is described in *Appendix I* of the ACUCOBOL-GT manual set.

# UNIX Installation

## License Files for UNIX

UNIX users may request a license file for one or more users.

When you receive your product(s), the package includes product codes and product keys for every product you ordered. You require theses product codes and product keys to create license files.

### The Activator Utility

The Activator utility automates the process of creating a license file. On UNIX platforms, the Activator utility operates through a command-line interface.

By default, the Activator utility program (`activator`) is placed in the same directory as the runtime and other binary executable files.

If you did not install the Activator utility with your other products, simply copy the file onto your computer and run it as you would any other executable.

### Changing or Updating Your UNIX License Files

If you need to alter a license file, contact your Micro Focus *extend* representative for updated product codes and product keys. When you receive them, launch the Activator utility and enter the new product codes and product keys as prompted.

## Installation Under UNIX

To install *extend* products on UNIX or Linux systems, you must have the product media, and the product codes and product keys for the products you intend to install. Your products are delivered via FTP.

### Installation Process

1. Download the product from the Micro Focus SupportLine site.

   **Note:** Before running the installer, ensure that the `tar` utility in on your PATH.

2. Enter the following:

   ```
   /path/to/installer/installer-name [options]
   ```

   where *installer-name* will be something similar to `setup_acucob1010pmk59shACU`.

   The following options can be included:

   | Option | Description |
   |---|---|
   | `-d` *`installation-path`* | Specifies a new default install location offered during the installation. If not specified, the default location is the current working directory (.). Any specified directory must already exist. |

| Option | Description |
|---|---|
| | This option can be combined with the `-EULA` option, but if it is, it will specify the location of the extracted EULA, and not the location of the installed product. |
| `-EULA` | Extracts the EULA, in `.htm` format, to a location of your choosing. The product installation will not continue when using this option. |
| `-help` | Displays the available options that can be appended to the installation command. |

**Note:** If you install ACUCOBOL-GT as a shared object library and you don't install to the default location, you need to set an appropriate library path variable specifying the location of the shared objects. For example, on an AIX system, you would need to set the LIBPATH environment variable. Note that if you log in as root or a superuser, this variable must also be set in root's environment for ACUCOBOL-GT to start. Additionally, see *SHARED_LIBRARY_PREFIX configuration variable*.

After installation, you must use the Activator utility to license the products installed.

3. From the installation directory, enter the following command to run the Activator utility:

   ```
   ./bin/activator
   ```
4. At the prompt, type the product code and product key pairs that came with your product package, pressing **Enter** after each pair: this updates the license file. Repeat this cycle until the code/key pairs for each product you have ordered are entered.

   **Note:** Each product searches for its license file in the same directory in which its executable resides. If you move the product's executable to a new directory, you must move its license file to the same location.
5. To start the acushare license manager service, enter the following:

   ```
   acushare -start
   ```
6. Navigate to the `sample` sub-directory of your installation directory and try compiling and running the **tour** program, using the following commands:

   ```
   ccbl tour.cbl
   runcbl tour.acu
   ```
7. If you get the message `Can't find entry for 'terminal' in 'term-lib`", you need to configure your terminal for ACUCOBOL-GT. See *Configuring Your Terminals.*
8. Once you have the sample program running, we recommend that you edit the `cblconfig` file supplied with ACUCOBOL-GT to meet the needs of your site. In particular, you should configure it to support the printers you have attached to your system.
9. If you are using shared memory, see the instructions for configuring acushare in *ACUCOBOL-GT User's Guide* > *Runtime Manual* > *Shared Memory* > *Acushare Utility Program*.

# SHARED_LIBRARY_PREFIX Configuration Variable

If you install ACUCOBOL-GT as a shared object library and you don't install to the default location, you need to set an appropriate library path variable (LIBPATH or LD_LIBRARY_PATH) specifying the location of the shared objects. The SHARED_LIBRARY_PREFIX variable helps the runtime find `libaclnt.so` (or `libaclnt.sl`) in case the LIBPATH (or LD_LIBRARY_PATH) variable is not set. libclnt.so (or .sl) is needed for AcuServer and AcuConnect support.

If a shared library name is specified without any directory information and the system call fails to load the shared library, the runtime will try to load the shared library from each of the directories specified in the SHARED_LIBRARY_PREFIX configuration variable.

The default value for SHARED_LIBRARY_PREFIX is `/opt/acucorp/`*`xxx`*`/lib`: `/opt/acu/lib`. The format of the value of SHARED_LIBRARY_PREFIX is the same as FILE_PREFIX. You can set SHARED_LIBRARY_PREFIX in the configuration file or environment, or programatically with the SET verb.

Note that the runtime searches for and loads `libaclnt.so` (or `libaclnt.sl`) using the default value of SHARED_LIBRARY_PREFIX. This happens before reading the configuration file, environment, or running any COBOL code.

You can set SHARED_LIBRARY_PREFIX to an empty value if you do not want to use it.

Also, if the license is for AcuTSL, in a transactional server environment such as CICS, the runtime will add `/opt/acucorp/`*`xxx`*`/bin/runcbl.1lc` and `/opt/acu/bin/runcbl.1lc` to its list of license files to check. First the runtime checks `$ACUCOBOL/etc/license.acu`, then `/etc/license.acu`. If neither exists, the runtime will check `/opt/acucorp/`*`xxx`*`/bin/runcbl.1lc` and finally `/opt/acu/bin/runcbl.1lc`.

For cases in which users install ACUCOBOL-GT in the default location, `/opt/acucorp/`*`xxx`* or `/opt/acu`, and they have a license file, `runcbl.1lc` in their bin directory, they will not need to copy the license to `/etc/license.acu`.

# Configuring Your Terminals

ACUCOBOL-GT requires data about the video environment it is running in. On Windows machines, it directly examines the hardware and configures itself appropriately. On UNIX and VMS machines, you must provide a description of the terminal you are using. This section describes briefly how to provide that information. Additional details are provided in the *Terminal Manager* section of the *ACUCOBOL-GT User's Guide.*

On systems that do not configure themselves automatically, describing the terminal to ACUCOBOL-GT involves two steps:

1. First, identify the terminal by setting the "TERM" variable.
2. Second, ensure that the terminal's characteristics are accurately described in the terminal database file.

## TERM Variable

ACUCOBOL-GT determines the type of terminal you are using by looking at the setting of the "TERM" variable. On UNIX and Linux machines, TERM is an environment variable; on VMS machines, it's a symbol. Samples for both UNIX and VMS are presented below.

TERM should be set to the name of one of the entries in the terminal database. You can examine the database file for valid names. The first field of each entry consists of a list of accepted names. Some common names are "vt100", "tv925", and "wy50" for VT100, Televideo 925, and Wyse 50 terminals, respectively.

Note that on most UNIX systems, the TERM environment variable is initialized as part of the login procedure. You will need to change this only if the name used is not one listed in the terminal database. On VMS systems, the TERM symbol defaults to "vt100" if it is not defined. Thus, you need to define the TERM symbol only if you want to use a terminal that is not VT100 compatible or if you want to use some advanced features of your terminal.

As an example, suppose you want to use a VT220 terminal. In the database, "vt220" is one of the accepted names for this type of terminal. On a VMS system, you would use the following command

```
TERM == "vt220"
```

On UNIX systems with the Bourne or Korn shell, the command would be:

```
TERM=vt200; export TERM
```

Using the C shell, the equivalent command is:

```
setenv TERM vt220
```

You may want to leave the TERM variable at its current setting to maintain compatibility with other software. If the setting is not correct for ACUCOBOL-GT, you can set the "A_TERM" variable instead. If both the

A_TERM and TERM variables are set, ACUCOBOL-GT uses the definition of A_TERM. This allows you to have different settings for ACUCOBOL-GT and your other software.

## Terminal Database

ACUCOBOL-GT comes with a database of terminal descriptions. On UNIX machines, this is called "a_termcap". On VMS machines it is called "A_TERMS.DAT". This database contains encoded descriptions of many types of terminals. You need to select the terminal type in the database that most closely matches the terminal you are using. If you need to, you can add your own entries in the database.

By default, the terminal database should reside in a pre-selected directory on your machine. On UNIX machines, this is the "/etc" directory; on VMS machines, it is the "SYS$LIBRARY" directory. If you want to place your terminal database somewhere else, then you must define the variable "A_TERMCAP" to be the full name of the database file. For example, on a VMS system, you could place the database in the "SYS $LOCAL" directory with the following command:

```
A_TERMCAP == SYS$LOCAL:A_TERMS.DAT
```

On UNIX systems (using the Bourne shell), you might use the command:

```
A_TERMCAP=/usr/local/etc/a_termcap; export A_TERMCAP
```

The *ACUCOBOL-GT User's Guide* contains more information about setting up terminals and making full use of their capabilities, and selecting terminal types.

# Changes Affecting Version 10.1.0

If you are upgrading from a version of the extend Interoperability Suite prior to version 10.1.0, you need to be aware of the following changes that may affect your existing applications.

**TRANSLATE_TO_ANSI configuration variable**

This variable has been superceded by the COBOL_CHARACTER_SET configuration variable, and should be replaced in your applications. See the *COBOL_CHARACTER_SET* details for more information.

**W$FONT library routine**

W$FONT may load slower than in the past, due to the new Unicode support.

The runtime now checks to see if a chosen font has a width that seems too large, and if so, every character is validated. With ANSI fonts, this is about 250 characters. With Unicode fonts, this can be up to 65,000 characters, meaning that the test can take considerably longer to load.

If you encounter W$FONT loading slower than in previous versions and you do not plan to use the Unicode support, set *V71-FONT-WIDTHS* to TRUE to skip the width validation.

# What's New

The following items are new for this release:

## ACUCOBOL-GT Enhancements

This section includes the enhancements related to ACUCOBOL-GT.

### ECN-4362 Permanent large file support for UNIX

Change Number: ECN-4362

Type of Change: Enhancement

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

All UNIX ports now have large file support permanently enabled. As of extend 10.1.0, the *USE_LARGE_FILE_API* configuration variable is no longer used.

### ECN-4363 Unicode support

Type of Change: Enhancement

Product: ACUCOBOL-GT

Module: Runtime

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: All

**DESCRIPTION:**

The extend 10.1.0 release introduces support for Unicode by introducing the ability to encode using the UTF-8 and UTF-16 character encodings. For certain aspects of extend, you can configure the encoding type, and for other aspects, a specific encoding is used as default.

The introduction of this support means that your programs have the ability to understand, process, and display any Unicode character handled by those encodings. As well as this ECN, refer to the following ECNs for details on how this support affects the different areas of the extend Interoperability Suite.

- ECN-4411
- ECN-4416
- ECN-GL552

The Windows runtime terminal manager now includes built-in Unicode support - for all screen I/O, the data is dynamically converted from the system code page, using UTF-16 encoding. From the point of view of your COBOL programs, little has changed. You still use all the verbs you normally use, and the data in your program remains unchanged, with one exception, described below.

If you rely on the TRANSLATE_TO_ANSI configuration variable, this no longer works, and has been replaced by the new configuration variable COBOL_CHARACTER_SET, which takes a number of different values, but has 3 pre-defined values:

**OEM**

Specifies that all data in the COBOL program is encoded in the OEM (DOS) character set.

**ANSI**

Specifies that all data in the COBOL program is encoded in the current ANSI code page character set.

**UTF-8**

Specifies that all data in the COBOL program is encoded in Unicode (UTF-8 format).

**Note:** As this is a multi-byte character set, you may need to expand some of your data variables to account for the larger size.

Additionally, setting this variable to a numeric value means that all data in the COBOL program is encoded in the ANSI code page with that value; for example: 437 is used in the United States. Setting the variable to ANSI is the same as setting it to the value of the current code page.

You can use this variable to dynamically change the specified character set if your data uses a mixture of formats; for example, data read from an XML file might be in UTF-8 format, while the rest of the data in your program might be in ANSI format.

If you have any existing UTF-16 encoded data, you must translate it to another supported encoding before it can be handled; see ECN-4367 for a solution.

# ECN-4376 Option to disable JNI_OnLoad function

Change Number: ECN-4376

Type of Change: Enhancement

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

You can now disable the normal processing performed by the JNI_OnLoad function in the runtime library. Set the environment variable *ACU_RUNCBL_JNI_ONLOAD_DISABLE* to 1 to disable this processing.

# ECN-4378 New vutil -unload option -k #

Type of Change: Enhancement

Product: ACUCOBOL-GT

Module: vutil

New Version: 10.1.0

Machines Affected: all

Known Versions Affected: N/A

**DESCRIPTION:**

A new option has been added to `vutil -unload` to specify the key order in which to read records.

Specify `-k #`, where # is the key number by which to read files. By default, `vutil -unload` reads the records in primary key order.

If the `-r` option is also specified on the same command line, the option that occurs later will be in effect.

# ECN-4389 Avoid function name collision with Tuxedo

Type of Change: Enhancement

Product: ACUCOBOL-GT

Module: Vision

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

An internal function named `Fread()` in the Vision library clashes with a similarly named function in Tuxedo. The Vision function has been renamed in order to avoid the conflict.

# ECN-4397 Vision file locking compatibility

Type of Change: Enhancement

Product: ACUCOBOL-GT

New Version: 10.1.0

Module: vutil

Machines Affected: All

**DESCRIPTION:**

A new vutil option has been introduced to enable you to test the compatibility of your file system when setting the 'asynchronous read' method of file locking for Vision files (V_LOCK_METHOD 2). When this method is set, the file system must be able to perform atomic file operations; that is, it must not allow read operations to view the intermediate state of a file currently being written to in another process. If you use this locking method on file systems that cannot perform operations atomically, it can lead to spurious file errors or file corruption.

The new option is `-wtest`, and can be run from the command line or from the **WTest** tab from the AcuBench Vision File Utility. It takes one mandatory and one optional argument: an existing directory on your file system (in which a temporary test file is created and written to/read from during the test), and an optional number indicating for how many seconds the test should be run.

```
vutil -wtest /usr1/vis 60
```

This example creates the temporary test file in the `/usr1/tmp` directory and run the test for 60 seconds. If the time argument is omitted or is less than 30, the test will run for 30 seconds.

While the test is running, a series of dots ("....") are displayed to show that the test is progressing normally.

During the test, blocks of characters are written to the temporary file and read back in a separate thread until either the indicated time elapses or an inconsistent read is encountered. A message will then be displayed showing the results of the test.

If the test passes, this indicates that the system did not allow the thread reading the data to see the data in an intermediate state, and thus V_LOCK_METHOD 2 can work reliably in this environment.

If a block is read that contains inconsistent data, this indicates that the file system does not ensure that data written in a single system call can always be read back consistently, and intermediate results may be seen. In this case, V_LOCK_METHOD 2 cannot operate reliably, and must not be used.

**Note:** The file system type and its mount options may affect the results of this test. Different file systems on the same system may produce different results. Ensure that you run this test on the file systems that will hold the Vision files that are to be accessed using the V_LOCK_METHOD 2 configuration.

# ECN-4399 vutil -rebuild process using multiple threads

Type of Change: Enhancement

Product: ACUCOBOL-GT

Module: vision

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

A new option has been introduced to `vutil -rebuild` that enables you to use multiple threads when rebuilding multi-keyed, Vision 6-formatted (or later) Vision files, which can improve the performance of the rebuild. For example:

```
vutil -rebuild -j #threads my-vis-file
```

where *#threads* must be a number between 2 and 120, and specifies the number of threads to be used in the rebuild process. Each thread is used to rebuild a key in the file; a new thread being spawned when a previous one completes rebuilding a key, until all keys are rebuilt.

The optimum number of threads that gives the best performance will vary depending on your system, and so trial and error in specifying different numbers of threads may be required. As a general guideline, specify a similar number of threads as your system has CPUs.

An important consideration when specifying the number of threads is that each thread is allocated memory as if it were its own version of vutil; therefore, the memory you have allocated for the V_BULK_MEMORY, V_BUFFERS, MAX_FILES, etc... variables is multiplied by the number of threads.

Another consideration is that rebuilding using multiple threads also uses additional temporary disk space, to hold the intermediate results of each key rebuild. This could be up to double the amount of space that a non-threaded rebuild would use.

# ECN-4411 Unicode support for XML technologies

Type of Change: Enhancement

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

The XML technologies in ACUCOBOL-GT now support Unicode to the extent described below.

XML data is now stored internally as UTF-8 encoded data before being transferred to a data item in a COBOL program, where it is then encoded according to the COBOL_CHARACTER_SET configuration variable (see ECN-4363 for more details). Similarly, when transferring a COBOL data item to an XML document, the data is encoded into UTF-8 from the character set defined by the configuration variable.

Transferring data to or from a COBOL data item can be done in many ways, for example, reading the data (AcuXML), getting the data via C$XML CXML-GET-DATA, setting the data via C$XML CXML-SET-DATA, etc... This re-encoding affects all of those methods, and also the XML PARSE and XML GENERATE statements. It does not however, affect any usage performed using the XML Extensions suite.

# ECN-4416 Unicode character translation library routines

Type of Change: Enhancement

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

There are six new library routines that can translate data items from one encoding to another; those supported encodings are UTF-8, UTF-16, ANSI, and ISO-8859-1.

- `C$UTF16-UTF8` translates from UTF-16 to UTF-8
- `C$UTF8-UTF16` translates from UTF-8 to UTF-16
- `C$COBOL-UTF8` translates from the current ANSI code page defined by the COBOL_CHARACTER_SET variable to UTF-8
- `C$UTF8-COBOL` translates from UTF-8 to the current ANSI code page
- `C$88591-UTF8` translates from ISO-8859-1 to UTF-8
- `C$UTF8-88591` translates from UTF-8 to ISO-8859-1

All of these routines are called in exactly the same way:

```
CALL translation-routine USING source, sourcelen
                               [,destination [,destinationlen]]
```

where:

***translation-routine***

One of the C$ routines listed above.

***source***

The string that you are translating. It must be either a POINTER (that you must set to a valid value) or an alphanumeric item.

***sourcelen***

The number of characters you want to translate. If this value is 0, then the size of the source data item is used (not valid when source is a POINTER). If this value is -1, the source is assumed to be terminated by a low-value character, and the entire string is translated.

***destination***

If given, is where the translated characters will be moved. If not given, the return-code will be the number of characters needed in the destination item to hold the entire source string. This data item can be either a POINTER or an alphanumeric data item. If it is a POINTER, you must set it to a valid value.

***destinationlen***

The number of characters that can be held in the destination data item. If this parameter is -1, or is not specified, then the length of the destination is used.

The return value is the number of characters moved to the destination data item, or the number that would be needed (when the destination item is missing or NULL).

If fewer characters are placed in the destination than there is room for, the routine will pad the destination with spaces.

**Examples**

Using the following data definitions:

```
01 my-string-1 pic x(100).
01 my-string-2 pic x(100).
01 my-pointer pointer.
01 my-len signed-int.
01 alloc-len signed-int.
```

In the following example, since *sourcelen* is 0, the CALL translates all 5 characters (ABcde) into UTF-8, placing the result into *my-string-1*.

```
CALL C$88591-UTF8 using "ABcde", 0, my-string-1.
```

In the following example, the CALL translates the 5 characters in my-string-1 (ABcde) into UTF-8, and places the result in *my-string-2*, which is then padded with UTF-8 spaces. The return-code is 5 (characters).

```
MOVE "ABcde" to my-string-1
CALL "C$COBOL-UTF8 using my-string-1, 5, my-string-2, 50
```

In the following example, although the source (*my-string-1*) is 26 characters, the CALL translates only the first 10 characters into UTF-16. The amount of space required for translation is calculated, allocated, and then translated into that buffer, which is then freed upon completion of the translation.

```
MOVE "abcdefghijklmnopqrstuvwxyz" to my-string-1.
CALL "C$UTF8-UTF16" using my-string-1, 10.

MOVE return-code to my-len.
MULTIPLY my-len by 2 GIVING alloc-len.  *> UTF-16 uses 2 bytes per character

CALL "M$ALLOC" using alloc-len my-pointer.
CALL "C$UTF8-UTF16" using my-string-1, 10, my-pointer, my-len.
CALL "M$FREE" using my-pointer.
```

# ECN-4422 PDF Printing Support

Type of Change: Enhancement

Product: ACUCOBOL-GT

Module: Runtime

New Version: 10.1.0

Machines Affected: All

DESCRIPTION:

Use the `-P PDF` *`filename`* syntax in your runtime configuration file to write output to a PDF file named filename. There are a number of configuration variables that enable you to tailor the formatting of the text; they cover such things as font and font-size selection, margin settings, and page orientation. See *Spooler Print Formatting* for more details.

**Example:**

With no modifications to an existing COBOL program, you can use the runtime configuration file to print to a PDF file instead of a printer.

If the COBOL program contains the following ASSIGN clause:

```
SELECT PRINT-FILE ASSIGN to PRINTER
```

You can assign PRINTER to be a PDF file, and configure it, using such variables in your runtime configuration file:

```
PRINTER -P PDF report.pdf
PDF_PAGE_SIZE LETTER
PDF_FONT_NAME HELVETICA
```

**Note:** If you want to specify a PDF file directly in your source code, you can include the assignment in the SELECT statement instead of the configuration file: `SELECT PRINTER ASSIGN to "-P PDF report.pdf"`.

# ECN-4426 New warning for vutil -unload -t

Type of Change: Enhancement

Product: ACUCOBOL-GT

Module: vutil

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

The `vutil -unload -t` option now issues a warning if a record containing a newline character is written to the output file. Since line sequential files use newline characters to indicate the end of a record, records that themselves contain newline characters should not be written to line sequential files. Use a binary sequential file format (`vutil -unload -b`) for such records.

The warning displayed is as follows:

```
Warning: record containing newline written to line sequential file
```

# ECN-4429 The extend UNIX product now uses an executable installer

Type of Change: Enhancement

Product: ACUCOBOL-GT

Module: All

New Version: 10.1.0

Machines Affected: all UNIX

**Description:**

To install extend on UNIX platforms, you must now run an executable installer, instead of the previous installation process of extracting the installation from a `.tar` file. The new installer will be named similarly to the old tar archives; for example, instead of:

```
acucob1010pmk10stACU.tar
```

the installer will be named:

```
setup_acucob1010pmk10stACU
```

In order to run the installer, the tar utility must be specified in your PATH. The installer can be run with or without additional options. When run with no options specified, the installer prompts you to accept the terms of the License Agreement. If you want to review the EULA documentation before accepting, you must answer 'N', and then rerun the installer again, including the `-EULA` option.

The default installation path is to install in the current directory (.); you may accept the default or specify another destination. If you want to change the default installation directory that is prompted, run the installer using the `-d` *`path/to/install`* option.

**Tip:** The installed product contains several files and directories, so it is advisable to install to a container directory to hold them all.

You can also run the installer with the `-help` option to display the available options.

For details on the full UNIX installation, including pre- and post-requisites, see the *UNIX Installation* section.

# ECN-4430 Modifying key structure of existing Vision files

Type of Change: Enhancement

Product: ACUCOBOL-GT

Module: vutil

New Version: 10.1.0

Machines Affected: all

**DESCRIPTION:**

The vutil Vision utility now includes functionality to modify the key structure of an existing Vision file. You can add, delete and modify keys.

This functionality is only available for version 6 or later Vision files that are not encrypted.

# ECN-4432 Multi-thread detection

Type of Change: Enhancement

Product: ACUCOBOL-GT

Module: Runtime

New Version: 10.1.0

Machines Affected: All

Known Versions Affected: All

**DESCRIPTION:**

The runtime has been enhanced so that it is now more thread-aware. The concept of threading was previously possible, but was implemented completely within the runtime, meaning that it was not inherently thread-safe, as it was not aware of multiple OS threads. Multiple OS threads are not currently supported, but are often present when an application uses external libraries (Java, .NET, etc...).

The runtime can now detect multiple OS threads during startup. If detected, the following message is output and the runtime stops:

```
Multiple OS threads detected
```

If the message appears, it means that your COBOL program has somehow started multiple OS threads. You need to figure out how those threads were started, and prevent them from starting.

**Note:** The check for multiple threads, and therefore the appearance of this message, only happens when running in debug mode, and the message will only appear once per session.

# Acu4GL Enhancements

This section includes the enhancements related to Acu4GL.

## ECN-GL552 Unicode support in Acu4GL for Microsoft SQL Server

Type of Change: Enhancement

Product: Acu4GL

Module: MSSQL

New Version: 10.1.0

Machines Affected: Windows only

**DESCRIPTION:**

This ECN extends ECN-4363, in which Acu4GL for SQL Server can now read and write Unicode data from ANSI data in a COBOL program. It uses the COBOL_CHARACTER_SET configuration variable (as described in ECN-4363) to determine how to interpret the data, and then writes that data to the database. The columns being read and written must be of type *nchar* or *nvarchar* (not *char* or *varchar*).

This ECN introduces a new configuration variable, A_MSSQL_CREATE_UNICODE_COLUMNS. If set to TRUE, then the interface will create *nchar* or *nvarchar* columns for all text data. Note that existing tables are not affected. The default value for this variable is FALSE, meaning that default behavior is the same as in prior versions.

Visit the following web page for a list of supported code pages for MSSQL: *https://technet.microsoft.com/en-us/library/ms186356(v=sql.105).aspx*

From extend 10.1.0 onwards, The `Acu4GLFD` utility is able to produce an appropriate FD for an existing SQL server table that has columns defined as *nchar* or *nvarchar*.

## ECN-GL556 Support added for latest ODBC Microsoft SQL Server drivers

Type of Change: Enhancement

RPI Number: 623862

Product: Acu4GL

Module: mssql

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: All

**DESCRIPTION:**

When using Acu4GL for Microsoft SQL Server, we now support ODBC Driver 11, 12, 13 for SQL Server. This results in the additional permissible values for the A_MSSQL_ODBC_DRIVER_NAME configuration variable:

ODBC Driver 11 for SQL Server

ODBC Driver 12 for SQL Server
ODBC Driver 13 for SQL Server

# AcuToWeb Enhancements

This section includes the enhancements related to AcuToWeb.

## ECN-AW001 Introducing AcuToWeb

Type of Change: Enhancement

Product: AcuToWeb

Module: AcuToWeb

New Version: 10.1.0

Machines Affected: Windows and Linux platforms

**DESCRIPTION:**

AcuToWeb is a new product to the extend Interoperability Suite, and is available as part of the normal Windows installer. It is an innovative technology that displays the user interface portion of your server-based ACUCOBOL-GT application to a client-side browser.

The client-side browser must have HTML 5 and WebSocket support; this can be found in such browsers as IE 10+, Edge, Firefox 38+, Chrome 43+, Safari 7+, IOS Safari 6+.

AcuToWeb consists of the AcuToWeb Gateway, AcuConnect, ACUCOBOL-GT runtime, your COBOL application and data, and also requires a web server (such as IIS, Apache, or AcuToWeb even includes its own web server). Client-side users only require a browser that supports HTML 5, unless your application needs access to the client, for example, client-side printing or other client-side resources, in which case AcuToWeb Desktop is required. AcuToWeb Desktop facilitates the communication from the server-side COBOL program (such as calls to Win$Printer) and provides printing functionality on the client.

AcuToWeb can be deployed in the following ways:

- In a single-tiered environment, where the web server, AcuToWeb, AcuConnect, ACUCOBOL-GT runtime and COBOL program and data all reside on single machine.
- In a multi-tiered environment, where the web server is configured to tunnel the requests to a AcuToWeb gateway residing on a different machine, and then the AcuToWeb gateway is configured to communicate with AcuConnect, ACUCOBOL-GT runtime and COBOL programs and data residing on another machine.

AcuToWeb functionality is facilitated by the AcuConnect thin client technology, and so access file, alias file and thin client connection configuration are all pre-requisites to AcuToWeb.

For more information, see *AcuToWeb* in the product documentation.

# AcuXDBC Enhancements

This section includes the enhancements related to AcuXDBC.

## ECN-XD108 AcuXDBC QUERY_TIMEOUT configuration variable

Type of Change: Enhancement

Product: AcuXDBC

New Version: 10.1

Module: AcuXDBC

Machines Affected: All

RPI Number: 1104447

**DESCRIPTION:**

A query through AcuXDBC has a default timeout of 2 seconds, but you can now configure the length of the timeout using a new configuration variable, QUERY_TIMEOUT, in your `acuxdbc.cfg` configuration file, before making a connection to the data.

In the `acuxdbc.cfg` file, set QUERY_TIMEOUT to a value (in seconds), or set to 0 (zero) so that the query never times out.

# ECN-XD109 AcuXDBC dynamic logging

Type of Change: Enhancement

Product: AcuXDBC

New Version: 10.1.0

Module: AcuXDBC

Machines Affected: All

**DESCRIPTION:**

For JDBC data sources, if you did not enable transaction logging during the initial setup, you can now enable it dynamically when connecting to the data source each time.

You can now append the required GENESIS variables to your connection string to enable logging for that connection only; for example, the following string sets full logging:

`jdbc://uid/pwd/xvision:acuxdbc.cfg@20222:`*`machinename`*`!`
`acuxdbc04,GENESIS_LOGFILE=`*`/path/to/logfile.log`*`,GENESIS_LOGOPTS=3`

**Note:** For ODBC connections on Windows, you can use the Logging tab on the ODBC Data Source Administrator to permanently enable logging for a particular data source name (DSN); see *AcuXDBC Setup: Logging Tab*.

**Configuration file:**

To extend this dynamic logging further, a new configuration file, `vortex.properties`, is available, which is used if it is in the same directory as `vortex.jar`.

```
# VORTEXJava logging
# logfile is the name of the logfile, e.g. /tmp/logfile
#logfile=/tmp/logfile
# logopts is a list of options, separated by + (optional)
#logopts=FULL+MULTI+TIME
#
# VORTEXjdbc connection commands:
# These are sent to the target after the connection succeeds.
# A maximum of 20 statements are sent and they must be consecutively numbered
# without gaps.
#
#jdbcsql0=set option logfile '/tmp/test_c3po.log'
#jdbcsql1=set option plan on
#
# VORTEX network parameters
# connect_timeout - seconds to wait on a connect
```

```
#connect_timeout=15
# read_timeout - seconds to wait on a read response
#read_timeout=1440
# key_connect - integer key to use for uid/pwd obfuscation. Must match the
#               -kn vtxnetd option value.
#key_connect=1999
#environment_variables=VORTEX_HOST_LOGFILE=/tmp/test,VORTEX_HOST_LOGOPTS=FULL
```

Uncomment (and edit, where required) the options in the configuration file; for example, to enable full logging:

```
logfile=/tmp/logfile
logopts=FULL
```

The valid `logopts` keywords are:

**FULL**

Verbose output

**MULTI**

Append a time-stamp to the logfile name

**TIME**

Inserts time-stamps in the logfile.

You can specify more than one option, separated by +; the order in which you specify each option is not important.

The VORTEXjdbc commands are typically `set option` commands; no query statements are allowed here. The statements are sent immediately after the connection is made.

The Vortex network parameters affect the network connection.

The `environment_variables` value is appended to the connection string to make it easier to manage server-side logging.

# BIS for extend Enhancements

This section includes the enhancements related to BIS for extend.

## ECN-4384 BIS support for IPv6 addressing on UNIX

Type of Change: Enhancement

Product: ACUCOBOL-GT

Module: Business Information Server

New Version: 10.1.0

Machines Affected: UNIX

**DESCRIPTION:**

IPv6 addressing support has been added to the UNIX version of Xcentrisity Business Information Server (BIS) for extend. This will allow BIS to run in a network that uses IPv6 (128-bit) IP addresses. Largely, the Apache web server manages the IP addresses of the requests from clients, but there are two areas where BIS is affected:

- The `{{Trace}}` and `{{Debug}}` tags now accept IPv6 addresses, for the IP option, in the following forms:
    - `x:x:x:x:x:x:x:x`

    - `x:x:x:x:x:x:d.d.d.d`
    - `d.d.d.d/p`
    - `x:x:x:x:x:x:x:x/p`
    - `x:x:x:x:x:x:d.d.d.d/p`

    Where:

    `d` is a decimal numeric value, less that or equal to 255; `x` is a hexadecimal numeric value, less than or equal to `0xffff`; `p` is a decimal value, less than or equal to 99.

    The special notation of two consecutive colons (`::`) indicates a sequence of multiple groups of 16-bit zeros. The `::` can only appear once in an address. It can also be used to compress the leading and/or trailing zeros in an address.

    The `/p` format of the address specifies the prefix-length of the address. The prefix-length indicates the number of leftmost contiguous bits in the address that are valid.
- When a request is received from an IPv6 address, for licensing purposes, the address is interpreted as such:
    - An IPv6 address of `fe80::/10` is considered a link local address.
    - An IPv6 address of `fec0::/10` is considered a site local address.
    - A link or site local address is considered a private IP address.
    - An IPv6 address of `::1/128` is considered a loopback IP address.

# XML Extensions for extend Enhancements

This section includes the enhancements related to XML Extensions for extend.

## ECN-4386 XML Extensions enhancements

Type of Change: Enhancement

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

There have been a number of enhancements to XML Extensions in this release of extend.

### Controlling whitespace

The main change is the enhanced control over whitespace handling when importing a file. The default behavior has changed to preserve all whitespace elements, that is, space characters, line feeds (LF), carriage returns (CR) and tabs (TAB); previously, only space characters were preserved.

The introduction of two new library routines enables you to set non-default behavior: you can strip out all LF, CR and TABs as before, or any subset of these characters. You can also normalize whitespace, which replaces any consecutive sequence of whitespace characters with a single space.

See the documentation on *XML SET WHITESPACE-FLAGS* and *XML GET WHITESPACE-FLAGS* in the documentation for more details on how to control whitespace.

### Addition information returned for non-successful operations

The `XML GET STATUS-TEXT` statement now obtains additional status lines for any errors or warnings. Each such line is now prefixed with `[`*nn*`]`, where *nn* is the error or warning number that the extra line describes. This is particularly helpful when there are warnings and a fatal error for the same statement. The "Called from" line now includes this prefix as well. This additional information is provided for the import, export or input of document name, the ambiguous element name and its content, and the extraneous import element name and its content. The information will eliminate the need for using `XML TRACE` in many instances during or after development of a program that uses XML Extensions.

The `XML GET STATUS-TEXT` statement now obtains additional status lines for any errors or warnings. Each such line is now prefixed with `[`*nn*`]`, where *nn* is the error or warning number that the extra line describes. This is particularly helpful when there are warnings and a fatal error for the same statement. The "Called from" line now includes this prefix as well. This additional information is provided for the import, export or input of document name, the ambiguous element name and its content, and the extraneous import element name and its content. The information will eliminate the need for using `XML TRACE` in many instances during or after development of a program that uses XML Extensions.

### General XML TRACE improvements

There were several improvements made and defects fixed that concern the `XML TRACE` statement:

- When a style sheet fails to load, and is then referenced a second time in the same program, a crash no longer occurs.
- On UNIX, a failed transform to a memory document now frees the memory allocated for the transformed document. This matches the behavior on Windows.
- Several possible buffer overruns when passed path-names or other user-provided strings too long for the internal buffers have been eliminated.
- The `XML TRACE` statement no longer crashes if the file name provided cannot be opened, for example, when a path is specified for the trace file that does not exist, or the user does not have write permissions to the directory indicated; instead, the `XML TRACE` statement fails with a 73 error if the trace file cannot be opened.
- An ambiguous import target data-name could cause XML Extensions to attempt a store outside the data area defined in the program for the import, thus causing memory corruption. Improvements in resolving ambiguous data-names now limit the possibility of this happening. If the situation does happen, the store outside the target data area is suppressed and an error 72 is returned for the `XML IMPORT` statement.

  Taking the following problem test case, the program defined a data-name twice, once within the target data item for the import and once outside the target data item. Import element qualification was not provided in this test case, but now XML Extensions uses its knowledge of the target data-name to implicitly qualify the element name; that is, only data-names subordinate to the target data-name are considered when resolving element names in the imported document to data-names in the importing program. When the target data-name itself is ambiguous, a -05 warning is produced. When the import element name is ambiguous (that is, matches more than one data-name defined subordinate to the target data-name), a -06 warning is produced. XML Extensions has also been changed to choose the lexically last definition of an ambiguous data-name, instead of the first.

# Resolved Issues

The following are resolved issues for the extend products.

## ACUCOBOL-GT ECN List

This section includes the ECNs relating to ACUCOBOL-GT:

### ECN-4412 RENEW-TIMEOUT fails to renew the timeout

Type of Change: Correction

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

When using a Format 2 ACCEPT statement, with the BEFORE TIME phrase, the configuration variable RENEW_TIMEOUT is now correctly renewing the timeout when a user starts entering data into the first control. Previously, this was failing.

The variable now also has an additional setting, RESTART or 2, which resets the timer after each keystroke within the field.

### ECN-4413 Mouse shapes on thin clients

Type of Change: Correction

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

ACUCOBOL-GT has a feature in which the mouse changes shape when positioned over classic-style fields (not controls) based on the type of data allowed in the field. This feature had been turned off when using a thin client.

### ECN-4414 Runtime memory error when path greater than 80 characters

Type of Change: Correction

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

Machines Affected: All UNIX and console mode Windows

**DESCRIPTION:**

The runtime no longer fails with a memory error if it is executed via a path containing more than 80 characters; previously, a memory error may have occurred.

# ECN-4415 Updating STATUS-BAR controls

Type of Change: Correction

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

When using a status bar in your application, you might just DISPLAY a new STATUS-BAR control each time you need to update its contents. Whilst this is a valid technique, it could look like a memory leak, and so is not considered best practice.

The preferred technique is to MODIFY the existing STATUS-BAR control, rather than creating a new one; the runtime can only show a single STATUS-BAR control, regardless of how many are actually attached to the window.

A new configuration variable, SINGLE-STATUS-BAR, has been introduced to help control this. If set to the default of TRUE, then displaying a STATUS-BAR control on a canvas that already has one will result in the old control being destroyed before creating the new one. When set to FALSE, a new STATUS-BAR control will be created and displayed, as in prior versions.

# ECN-4417 Microsoft Calendar control crashes on exit

Type of Change: Correction

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

RPI Number: 1104310

Machines Affected: Windows

Known Versions Affected: 10.0.1

**DESCRIPTION:**

When destroying the Microsoft Calendar Control ActiveX control, the runtime no longer crashes.

# ECN-4418 C$XML PARSE-FILE hangs

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

RPI Number: 1104470

Machines Affected: Windows

Known Versions Affected: 10.0.1

**DESCRIPTION:**

C$XML no longer hangs when parsing an XML file that contains UTF-8 encoded characters. Previously, some types of characters could cause the hang.

# ECN-4419 Extra segment too large with many files and alternate keys

Type of Change: Enhancement

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

RPI Number: 1103149

Machines Affected: All

**DESCRIPTION:**

As from extend 10.1.0, file descriptors in a single object file are no longer limited to a size of 64KB; they now occupy regular memory, as do other data descriptors. Previously, file descriptors than contained many files, and/or many alternate keys could exceed the limit in place.

# ECN-4420 SORT TABLE memory error when more than 32767 entries

Type of Change: Correction

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

RPI Number: 1104773

Machines Affected: All

**DESCRIPTION:**

Performing the SORT TABLE operation on a table with more than 32767 entries no longer results in a memory error.

# ECN-4421 Slow compilation when using -Ze compiler option

Type of Change: Correction

Incidents: 2863349

RPI Number: 1104982

Product: ACUCOBOL-GT

Module: compiler

New Version: 10.1.0

Machines Affected: all

Known Versions Affected: all

**DESCRIPTION:**

The compilation time for a program that is compiled with the -Ze option and that contains many symbols has been improved.

# ECN-4423 XFD comment directive not inserting comments

Type of Change: Correction

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

RPI Number: 1104752

Machines Affected: All

**DESCRIPTION:**

Comments are now being inserted correctly in the XFD file when using the XFD COMMENT directive. Previously, they were not if the XFD file was in its default XML format.

# ECN-4424 `AcuGT.exe` not registered during install

Type of Change: Correction

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

The Automation Server (`AcuGT.exe`) is now automatically registered when installing a product on Windows that includes the runtime.

# ECN-4425 Incorrect data returned by thin client

Type of Change: Correction

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

RPI Number: 1104404

Machines Affected: All

**DESCRIPTION:**

Version 10 enhancements resulted in a subtle change in the ordering of messages from the thin client, in a way that can cause COBOL code to continue executing with slightly different timing than before. One side-effect of this is that data 'prefetched' from the client can get used by the COBOL program after circumstances have changed, invalidating the data's current value. In particular, under unusual circumstances, two sets of prefetched data can be present on the server, causing invalid results. (Prefetching data is a network optimization technique that reduces overall traffic.)

In the current release, only the most recent set of prefetched data is retained, ensuring that stale data is not accidently retrieved by the program.

# ECN-4427 New variable to turn off effects of ECN-4137

Type of Change: Correction

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

RPI Number: 1105652

Machines Affected: All

**DESCRIPTION:**

A new configuration variable, *ECN_4137*, when set to 0 enables you to turn off the effects of ECN-4137. Since ECN-4137 corrects an error in floating point rounding, it is not recommended to turn this off unless you fully understand the consequences.

# ECN-4428 Using NEXT SENTENCE with version 6.2 (and earlier) semantics

Type of Change: Correction

Product: ACUCOBOL-GT

Module: vutil

New Version: 10.1.0

RPI Number: 1105397

Machines Affected: All

**DESCRIPTION:**

The NEXT SENTENCE syntax now works as expected for programs compiled with *-Z62* or earlier). Previously, NEXT SENTENCE could change behavior when using older program semantics, causing unexpected results.

**Note:** The NEXT SENTENCE syntax is considered obsolete, and should not be used if possible.

# ECN-4431 SET *var1* TO ADDRESS OF *var2* fails

Type of Change: Correction

Incidents: 2869786

RPI Number: 1106086

Product: ACUCOBOL-GT

Module: Runtime

New Version: 10.1.0

Machines Affected: 64-bit Windows only

Known Versions Affected: All

**DESCRIPTION:**

The SET *var1* TO ADDRESS OF syntax no longer causes a MAV. Previously, this SET construct would fail to set a variable to the right value under certain circumstances - this would only happen on 64-bit Windows, and then only if the address of the variable was larger than 0x00FFFFFFFF. This seemed to have been limited to setting indexed variables on Windows 2012 Server.

# ECN-4433 Multi-monitor issues

Type of Change: Correction

Incidents: 2865396, 2834985

RPI Number: 1105219, 1101449

Product: ACUCOBOL-GT

Module: Runtime

New Version: 10.1.0

Machines Affected: Windows only

Known Versions Affected: All

**DESCRIPTION:**

The runtime is now multi-monitor aware. Previously, it was not, which lead to the following issues: when a COBOL program was running on a secondary monitor, the automatic placement of floating windows would be incorrect; when maximizing a resizable window, the maximum size would reflect the geometry of the primary monitor; and when an application was not placed on the primary monitor, its tooltips would not show on the correct monitor.

# ECN-4434 XFD-related compilation problems

Type of Change: Correction

Incidents: 2860903

RPI Number: 1104535

Product: ACUCOBOL-GT

Module: compiler

New Version: 10.1.0

Machines Affected: all

Known Versions Affected: all

**DESCRIPTION:**

The following compilation errors relating to XFD files are now resolved:

- Selecting a different object code version with the compiler -Z# option no longer results in a higher XFD version being generated than was selected with the -F# option. Previously, compiling for versions 4.1 through 5.2 would force the XFD version to be 4, and compiling for versions 6.0 through 8.*x* would force the XFD version to be 5.
- When an XFD NAME directive contains a character greater than or equal to 0x80 that is not the start of a UTF-8 sequence, the compiler no longer hangs, and now properly encodes and includes the character(s) in the XFD file.

# ECN-4435 PLACEMENT values ignored for 64-bit Windows applications

Type of Change: Correction

Incidents: 2868194

RPI Number: 1105643

Product: ACUCOBOL-GT

Module: Runtime

New Version: 10.1.0

Machines Affected: 64-bit Windows only

Known Versions Affected: All

**DESCRIPTION:**

The special PLACEMENT values (`TVPLACE-FIRST`, `TVPLACE-LAST`, and `TVPLACE-SORT`) when adding items to a list now work as expected for 64-bit Windows applications.

# ECN-4436 Runtime freezes when getting a double parameter from a .NET method

Type of Change: Correction

Incidents: 2870776

RPI Number: 1106140

Product: ACUCOBOL-GT

Module: Runtime

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: All

**DESCRIPTION:**

When a .NET method has an `out double` parameter, and the program does not contain `DECIMAL-POINT IS COMMA` in the Special-Names paragraph, and the program uses a configuration variable to modify the decimal point, the runtime no longer goes into an infinite loop when returning the double value.

# ECN-4437 Method Not Found when passing a NULL object handle

Type of Change: Correction

Incidents: 2871800

RPI Number: 1106205

Product: ACUCOBOL-GT

Module: Runtime

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: All

**DESCRIPTION:**

When a .NET method has a stored object as a parameter, it was not possible to pass a null object to the method: a `Method Not Found` message was generated. Note that not all methods that take objects will allow a null object; now in those cases, a runtime exception is raised.

# ECN-4441 FILE-TRACE-TIMESTAMP setting not honored

Type of Change: Correction

Incidents: 2875622

RPI Number: 623013

Product: ACUCOBOL-GT

Module: stdlib

New Version: 10.1.0

Machines Affected: all

Known Versions Affected: at least 8.1.3+

**DESCRIPTION:**

The value of the FILE-TRACE-TIMESTAMP configuration variable is now honored. Previously, if this variable was set in a configuration file, the timestamp feature would be turned on regardless of the setting.

# ECN-4442 Error when running BIS/Apache setup scripts on UNIX

Type of Change: Correction

RPI Number: 614826

Product: ACUCOBOL-GT

Module: config_bis_apache.sh

New Version: 10.1

Machines Affected: Solaris 10.0

Known Versions Affected: 10.0

**DESCRIPTION:**

You no longer receive the following error when running the `config_bis_apache.sh` script when setting up the Apache Web server to run with BIS:

```
./config_bis_apache.sh: IFS: cannot unset
```

# ECN-4443 GDI resource leak

Type of Change: Correction

Incidents: 2874421

RPI Number: 1106537

Product: ACUCOBOL-GT

Module: Runtime

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: 9.2.0 and later

**DESCRIPTION:**

The runtime no longer leaks GDI resources when certain TAB control properties are used. Previously, use of these properties would cause problems with the display of the screen after a while.

# ECN-4444 Check box values not updating via thin client

Type of Change: Correction

Incidents: 2866717

RPI Number: 1105581

Product: ACUCOBOL-GT

Module: Runtime

New Version: 10.1.0

Machines Affected: All

Known Versions Affected: Unknown - at least 10.0 and later

**DESCRIPTION:**

Check box values now update correctly even when focus is not on the item. Previously, when displaying a Screen Section item during an embedded procedure, the value of a check box within the screen would not change unless that check box has the focus before the click started; you had to click the check box twice to see the updated value.

# ECN-4445 Compiler crash when listing COPY files

Type of Change: Correction

Incidents: 2873985

RPI Number: 1106518

Product: ACUCOBOL-GT

Module: ccbl32.exe

New Version: 10.1.0

Machines Affected: All

Known Versions Affected: All

**DESCRIPTION:**

When using the compiler to list the opened files (-Ms option), and a COPY/REPLACING phrase goes past column 72, the compiler no longer crashes.

Previously, if compiling from AcuBench, this error could crash AcuBench, which could lead to information being lost, or damage to the project.

# ECN-4448 Using the *AUTOFILL* property on 64-bit server with AcuThin

Type of Change: Correction

Incidents: 2876422

RPI Number: 1106800

Product: ACUCOBOL-GT

Module: Runtime

New Version: 10.1.0

Machines Affected: All 64-bit

Known Versions Affected: 10.0.0 and later

**DESCRIPTION:**

When using the *AUTOFILL* property with AcuThin connecting to a 64-bit server, the runtime no longer crashes.

# ECN-4449 Memory error upon runtime shutdown

Type of Change: Correction

Incidents: 2880941

RPI Number: 1107413

Product: ACUCOBOL-GT

Module: runtime

New Version: 10.1.0

Machines Affected: all

Known Versions Affected: 7.1.0+

**DESCRIPTION:**

ECN-3137 introduced a potential memory error when the runtime shuts down; this has now been fixed. The error was possible when multiple programs were in use, and could cause a crash or unpredictable behavior.

# ECN-4450 Compiler memory error on large programs

Type of Change: Correction

Incidents: 2880723

RPI Number: 1107400

Product: ACUCOBOL-GT

Module: compiler

New Version: 10.1.0

Machines Affected: all

Known Versions Affected: all

**DESCRIPTION:**

A potential memory error when compiling large programs has been fixed. Previously, an internal table was not resized correctly and this led to an invalid memory reference. This error resulted in a crash or undefined behavior.

# ECN-4451 EXTFH interface incorrectly referenced FCD field on 64-bit port

Type of Change: Correction

Incidents: 2881260

RPI Number: 1107565

Product: ACUCOBOL-GT

Module: EXTFH interface

New Version: 10.1.0

Machines Affected: all 64-bit

Known Versions Affected: all

**DESCRIPTION:**

The ACUCOBOL-GT file system interface to EXTFH no longer incorrectly accesses the maximum record length field in the FCD on 64-bit ports. Previously, it would, and this would lead to different and incorrect behavior compared to 32-bit ports.

# ECN-4452 Tree height incorrect after a multi-threaded vutil rebuild

Type of Change: Correction

Incidents: 2882760

RPI Number: 1107678

Product: ACUCOBOL-GT

Module: Vision

New Version: 10.1.0

Machines Affected: all

Known Versions Affected: 10.1.0 Beta

**DESCRIPTION:**

After a multi-threaded rebuild of a Vision file, the tree height of the resulting file is no longer reported as 0 by `vutil -info`.

# ECN-4453 Vision 3 files grew too large with new large file offsets

Type of Change: Correction

Incidents: 2882759

RPI Number: 1107661

Product: ACUCOBOL-GT

Module: Vision

New Version: 10.1.0

Machines Affected: all

Known Versions Affected: 10.0.0+

**DESCRIPTION:**

Previously, the maximum file size of Vision 2 and 3 files was not explicitly checked in Vision. When a file grew too large, it would cause a system error (file status 30) due to an invalid file offset. The introduction of large operating system file offsets in extend 10.0.0 caused this failure mode to not work anymore, and an invalid Vision 2 or 3 file would result.

A test has now been added to the file size of Vision 2 and 3 files. When such a file is about to grow too large, it will be marked as broken with file status 98,88. If this file status is encountered, the file would best be rebuilt to a file format that supports larger file sizes (Vision 4, 5, or 6). The maximum size of a Vision 2 or 3 file is the same as the maximum size of a Vision 4 or 5 segment.

# Acu4GL ECN List

This section includes the ECNs relating to Acu4GL:

## ECN-GL553 Acu4GL fails to use the extra WHERE constraint

Type of Change: Correction

Product: ACUCOBOL-GT

Module: vutil

New Version: 10.1.0

RPI Number: 1105530, 1105669

Machines Affected: All

**DESCRIPTION:**

The external variable *A4GL-WHERE-CONSTRAINT* is now working as expected. Previously, a test that checks bounds errors was failing, which caused the variable to be ignored.

## ECN-GL554 A-MSSQL-ROWCOUNT error

Type of Change: Correction

Incidents: 2865176

RPI Number: 1105225

Product: Acu4GL

Module: MSSQL

New Version: 10.1.0

Machines Affected: Windows only

Known Versions Affected: All

**DESCRIPTION:**

Switching back and forth from a ROWCOUNT of 1 and 0 no longer causes a READ operation to fail with an `Invalid cursor state` error.

### ECN-GL555 WRITE or REWRITE may fail

Type of Change: Correction

Incidents: 2868949

RPI Number: 1106128

Product: Acu4GL

Module: MSSQL

New Version: 10.1.0

Machines Affected: Windows only

Known Versions Affected: All

**DESCRIPTION:**

After a READ operation that returns data for a NULL column value, a WRITE or REWRITE operation no longer fails or truncates the data.

## AcuBench ECN List

This section includes the ECNs relating to AcuBench:

### ECN-WB135 When linking a COPY file, the wrong path is stored

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

Linking to COPY files outside of a project's directory structure is now working correctly. Previously, if you linked to a COPY file that was located outside of the project's directory, AcuBench would prepend `./screen` to the file name, making it inaccessible at compile time.

### ECN-WB136 Quick Watch context menu option not working

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

The Quick Watch option on the context menu is now working correctly (that is, displaying the **Quick Watch** window). Previously, it did not, although the other access methods to the **Quick Watch** window did.

# ECN-WB137 Integrated debugger breaks on the wrong line

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

The integrated debugger no longer breaks on the wrong line. Previously, it would often break on the line preceding the one containing the breakpoint.

# ECN-WB138 Change in default behavior for the HOME key

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

From version 10.0 onwards, AcuBench is using an updated code editor base from a third party, and as a result, the default behavior of the HOME key has changed.

By default, the HOME key now toggles between column 1 and the first non-space character of the current line.

If you prefer to revert to the previous behavior, which just moves the cursor to column 1, a new option has been added to **Tools** > **Options** > **Code Editor** > **General**: check **HOME key goes to column 1** to revert to the old behavior.

# ECN-WB471 Full printing functionality restored

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

The following printing functionality, which was reduced in version 10.0 of AcuBench, has been restored:

- Color and Zebra printing
- Printing of line numbers
- Setting of margins
- Printing selections
- Printing boundary lines

# ECN-WB472 Code Insight not displaying control properties

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

Known Versions Affected: 10.0.0 and later

**DESCRIPTION:**

The **Code Completion** functionality (enabled via **Tools** > **Options** > **Code Editor** > **Code Insight**) is now working correctly.

# ECN-WB473 Auto-generated code may not be inserted at the beginning of lines

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

Auto-generated code is now being inserted correctly. Previously, various operations would not add code to the right location of the source file; for example:

- Adding a new event paragraph could add the first (comment) line to the end of the last line, instead of starting a new line.
- Adding a code snippet could add the snippet to the current location, instead of starting with the next line.

# ECN-WB474 Comment Block and Uncomment Block not working correctly

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

When using the Comment Block or Uncomment Block options, the editor now correctly applies the action to the selection. Previously, if the cursor was positioned in column 1 of a line, that line was being interpreted as part of the block, when in fact it should not have been affected.

# ECN-WB475 Modifications in the Event Editor not changing the Modified state

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

When making changes to the Event Editor code, AcuBench now shows the project as Modified. Previously, certain actions made in the Event Editor were not recognized as a modification.

# ECN-WB476 Files that have no extension are not treated as COBOL

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION of problem or enhancement:**

When opening a file with an unknown or no file extension, AcuBench now correctly prompts if you want to treat the file as COBOL. Previously, the prompt was not being displayed for files with no extension.

# ECN-WB477 Copy and Paste an entry field fails to copy the autofill properties correctly

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

When copying and pasting an `ENTRY-FIELD` control in the Screen Designer, the `autofill` properties are now copied correctly. Previously, they could be populated with spurious values.

# ECN-WB478 AcuBench builds the wrong project

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

AcuBench no longer loses track of the currently active project when AcuBench first starts up. Previously, in certain circumstances, if you right-clicked a project that was not the currently active project, and selected **Build**, AcuBench would build the active project, and not the project that was clicked.

# ECN-WB631 BACKSPACE key shifts line to the right

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

When the **Virtual Space** option is checked (**Tools** > **Options** > **Code Editor** > **General**), the current line no longer erroneously shifts to the right if the cursor is beyond the end of a line and you press the BACKSPACE key. Previously, it would insert a number of spaces at the beginning of the line, essentially shifting the line to the right.

# ECN-WB633 Wrong cursor position after Replace All operation

Type of Change: Correction

Incidents: 2865101

RPI Number: 1105183

Product: AcuBench

Module: AcuBench.exe

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: 10.0.0 and later

**DESCRIPTION:**

When replacing text using **Replace All**, the cursor now returns to its current position when the operation is complete.

# ECN-WB633 and ECN-WB634 Code parameters (tooltips) not showing

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

When editing code, if the **Code Parameters** check box is checked (**Tools** > **Options** > **Code Editor** > **Code Insight** > **Code Parameters**), tooltips are now displaying correctly; for example, when you type PERFORM, a tooltip describing the PERFORM verb now displays. Previously, this tooltip was not displaying when the check box was checked.

# ECN-WB635 Relative copybook filenames not recognized

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

RPI Number: 1105303

Machines Affected: All

**DESCRIPTION:**

AcuBench now detects relative filenames, and correctly applies them within a generated COPY line. Previously, it would use the full given name of a file, potentially causing compilation errors.

# ECN-WB636 Wrong code generated for FONT handles

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

RPI Number: 1105303

Machines Affected: All

**DESCRIPTION:**

The following problems, which were a consequence of ECN-WB940 (the ECN that added code to destroy handles on shutdown), have been fixed:

- If a handle was a font handle of a standard font, the value should not be NULL at initialization.
- If a handle was subordinate to a REDEFINES clause, it should not be given any value.

# ECN-WB637 Incorrect code completion text

Type of Change: Correction

Incidents: 2866050

RPI Number: 1105253

Product: AcuBench

Module: AcuBench.exe

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: 10.0 and later

**DESCRIPTION:**

The code completion text offered when coding an OPEN statement has now been changed to correctly state I-O (instead of IO).

# ECN-WB638 Tab color not saved in `.psf` file

Type of Change: Correction

Incidents: 2866695

RPI Number: 1105369

Product: AcuBench

Module: AcuBench.exe

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: 10.0 and later

**DESCRIPTION:**

When setting the tab color or tab color variable for the Screen Painter, the changes are now saved in the program's `.psf` file. Previously, they would not, and so would be reset every time you reopened the project.

# ECN-WB639 AcuBench crash when using ActiveX controls

Type of Change: Correction

Incidents: 2859164

RPI Number: 1105821

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

Certain ActiveX controls should no longer cause AcuBench to crash. Previously, a crash could occur when regenerating a workspace, or even when displaying a screen in the Screen Painter; this was apparently caused by a change, implemented by Microsoft, in the way that ActiveX controls are created.

# ECN-WB640 Bookmarks and errors don't follow edits

Type of Change: Correction

Incidents: 2869475

RPI Number: 1105821

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

Bookmarks and error lines now follow the appropriate lines when the source file is edited. Previously, adding additional lines would display the correct behavior, but when deleting lines, the bookmarks would not move with the appropriate lines of text. Also, when the problem occurred, double-clicking an error message would take you to the line named in the error message, and not necessarily the line that the error was on.

# ECN-WB641 Syntax highlighting of numbers

Type of Change: Correction

Incidents: 2873028

RPI number: 1106367

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

Syntax highlighting now works as expected. Previously, some number formatting was failing, in that it was coloring too many characters.

# ECN-WB642 Unable to remove keyboard mappings

Type of Change: Correction

Incidents: 2873031

RPI Number: 1106370

Product: AcuBench

Module: AcuBench.exe

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: 10.0.0 and later

**DESCRIPTION:**

It is now possible to remove a keyboard mapping from AcuBench. Previously, you could only add a new mapping, or change an existing one.

# ECN-WB643 Replace in Files count incorrect

Type of Change: Correction

Incidents: 2868692

RPI Number: 1105823

Product: AcuBench

Module: AcuBench.exe

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: 10.0.0 and later

**DESCRIPTION:**

When performing a Replace in Files operation, AcuBench no longer reports the wrong number of replacements for each file modified. Previously, it did, although the final result count was correct.

# ECN-WB644 AcuBench projects incorrectly marked as modified

Type of Change: Correction

Incidents: 2868397

RPI Number: 1106447

Product: AcuBench

Module: AcuBench.exe

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: 10.0.0 and later

**DESCRIPTION:**

AcuBench no longer marks a project as being modified when opening then closing the Events Paragraph without actually making any edits.

# ECN-WB645 Missing background color in Code Editor

Type of Change: Correction

Incidents: 2875850

RPI Number: 1106726

Product: AcuBench

Module: AcuBench.exe

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: 10.0.0 and later

**DESCRIPTION:**

When your Code Editor format has items that use both background and foreground colors, the Code Editor now shows the correct background color. Previously, it would only show the standard background color.

**Note:** This feature only works in AREA B, since other areas tend to use their own background color, which is different than it was in previous versions.

# ECN-WB646 The Code Completion list for the CALL statement constantly displayed

Type of Change: Correction

Incidents: 2876067

RPI Number: 1106750

Product: AcuBench

Module: AcuBench.exe

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: 10.0.0 and later

**DESCRIPTION:**

The Code Completion functionality is now working as expected when displaying code options for the CALL statement. Previously, after entering CALL and an opening quote ("), a list box would display the possible callable functions, but would not close after one was selected; there was no other way to close the list without closing AcuBench.

# ECN-WB647 Multi-line selection using Shift-click not working

Type of Change: Correction

Incidents: 2874774

RPI Number: 1106577

Product: AcuBench

Module: AcuBench.exe

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: 10.0.0 and later

**DESCRIPTION:**

The Shift-click method of selecting multiple lines at once is now working as expected. Previously, when selecting the last line in the selection (by holding down Shift and clicking the line number), only that line would show as selected.

# ECN-WB648 Unable to auto-locate error in source code

Type of Change: Correction

Priority: Medium

Incidents: 2880039, 2879926

RPI Number: 1107255, 1107257

Date: 2016-11-16

Product: AcuBench

Module: AcuBench.exe

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: 10.0.0 and later

**DESCRIPTION:**

When double-clicking on an entry in an output window, you are now correctly positioned on the relevant line of source code. Previously, if you double-clicked an entry from the Find in Files tab, or an error in the Build tab, you were only taken to the first line of the code.

# ECN-WB649 Polar SpellChecker ActiveX control fails to display

Type of Change: Correction

Incidents: 2882355

RPI Number: 1107634

Product: AcuBench

Module: AcuBench.exe

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: 10.0.0 and later

**DESCRIPTION:**

Some ActiveX controls (the Polar SpellChecker control being one) now display correctly in AcuBench. Previously, they would not display, which was due to Data Execution Prevention being enabled in the development environment (Visual Studio 2012) used to build the product.

# ECN-WB960 Duplicate code when destroying fonts and bitmaps

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

More efficient code has been introduced when destroying auto-created fonts and bitmaps. Although not technically incorrect, AcuBench would previously generate duplicate code, destroying fonts and bitmaps multiple times (once for each control they were used in); now, those handles are only destroyed once.

# ECN-WB961 Certain Report Writer defaults not saved

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

Report Writer defaults are now saved correctly. Previously, some of the default values were not saved. In particular, if you set the default title to empty, the value would not be saved, and the default would remain as it was prior to making the default change.

## ECN-WB962 Opening projects from newer versions of AcuBench

Type of Change: Correction

Product: AcuBench

Module: AcuBench

New Version: 10.1.0

Machines Affected: All

**DESCRIPTION:**

You can now open a project in an earlier version of AcuBench than the one that was used to create it.

Prior to extend 9.2.0, attempting to open a project file that was created with a later version of AcuBench would fail. For example, trying to open a project created by AcuBench 9.2.5, with AcuBench 8.1.3 would result in an error message, and the open would fail.

# AcuSQL ECN List

This section includes the ECNs relating to AcuSQL:

## ECN-SQL149 Insert of DATE fails

Type of Change: Correction

Product: AcuSQL

Module: runtime

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: 9.2.0 and later

**DESCRIPTION:**

When using SQL Server Native Client 11.0, the following error may be produced when inserting dates: `Invalid character value for cast specification`.

Inserting dates with an ODBC driver requires those dates to be in YYYY-MM-DD format. If your dates do not follow this format, the error is reporting correctly. Previous SQL drivers did not enforce this requirement, and so your code when using an older driver would not have produced the error. From SQL Server Native Client 11.0 onwards, this format is enforced, hence the error.

To mitigate the need to modify your programs, you can use a new configuration variable, which will update your dates to the required format before they are written to the SQL Server database. However, if possible, you should modify your COBOL programs so that date values are in YYYY-MM-DD format before being sent to the AcuSQL driver; thus, no relying on the variable.

To use the new configuration variable, ASQL_AUTO_DATE_FORMAT, set it to TRUE, which will attempt to automatically format the inserts to a DATE column to the YYYY-MM-DD format. This variable is only used if the source is a PIC X(n) data type, the target is a DATE type, and the length of the source (n) is not correct for a DATE column (which, at the time of writing, is 10). For example, if these conditions are met, a value of 12345678 is modified to 1234-56-78. There is no mechanism to specify another format for the date (for example, YYYYMMDD vs DDMMYYYY).

The default value of this variable is FALSE, to preserve historical behavior.

# ECN-SQL150 Performance degradation

Type of Change: Correction

Product: AcuSQL

Module: runtime

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: 9.1.2 and later

**DESCRIPTION:**

ECN-SQL138 fixed an issue in which multiple result sets could not be accessed concurrently from the same connection. This fix had the result of slowing down performance dramatically on large tables if that concurrent access was not needed.

A new configuration variable, ASQL_FAST_ACCESS, can be applied in situations where SELECT queries are solely available to a single request, which will improve performance greatly. Set the variable to TRUE to use this, which results in SELECT queries not being treated as cursors. If you do attempt to retrieve other information from the same connection whilst this is set, you may get a `"Connection busy"` error.

The default value is FALSE.

# ECN-SQL151 Error on SELECT after executing a Stored Procedure

Type of Change: Correction

Product: AcuSQL

Module: runtime

New Version: 10.1.0

Machines Affected: Windows

Known Versions Affected: 9.2.0 and later

**DESCRIPTION:**

After executing a stored procedure, you can now perform subsequent SELECT on the same connection, without error. Previously, this would fail with the error `Connection is busy with results for another command`. It would only happen when using the SQL Native Client 11.0, but was technically an error with all SQL Server drivers.

## ECN-SQL153 Invalid buffer length on 64-bit Windows

Type of Change: Correction

Incidents: 2869786

RPI Number: 1105860

Product: AcuSQL

Module: asqlsrvr.dll

New Version: 10.1.0

Machines Affected: Windows x64 only

Known Versions Affected: All

**DESCRIPTION:**

When using AcuSQL on a 64-bit Windows machine to connect to an SQL Server, you no longer receive an `Invalid buffer length` error when using the SQLDA data item.

## ECN-SQL154 Connection busy error when interspersing CURSOR with UPDATE

Type of Change: Correction

Incidents: 2868616

RPI Number: 1106010

Product: AcuSQL

Module: SQL Server

New Version: 10.1.0

Machines Affected: Windows only

Known Versions Affected: 9.2.0 and later

**DESCRIPTION:**

When interspersing CURSOR fetches with UPDATE, INSERT, or DELETE statements, you no longer receive a `Connection is busy` error.

# AcuXDBC ECN List

This section includes the ECNs relating to AcuXDBC:

## ECN-XD110 Too many cursors error when using AcuXDBC Server

Type of Change: Correction

RPI Number: 1107117

Product: AcuXDBC

Module: AcuXDBC

New Version: 10.1.0

Machines Affected: All

Known Versions Affected: All

**DESCRIPTION:**

You should no longer receive a 'Too many cursors' error when using a JDBC connection through AcuXDBC Server.

# Updates and SupportLine

Our Web site gives up-to-date details of contact numbers and addresses.

## Further Information and Product Support

Additional technical information or advice is available from several sources.

The product support pages contain a considerable amount of additional information, such as:

- The *Product Updates* section of the Micro Focus SupportLine Web site, where you can download fixes and documentation updates.
- The *Examples and Utilities* section of the Micro Focus SupportLine Web site, including demos and additional product documentation.
- The *Support Resources* section of the Micro Focus SupportLine Web site, that includes troubleshooting guides and information about how to raise an incident.

To connect, enter *http://www.microfocus.com* in your browser to go to the Micro Focus home page, then click *Support*.

**Note:** Some information may be available only to customers who have maintenance agreements.

If you obtained this product directly from Micro Focus, contact us as described on the Micro Focus Web site, *www.microfocus.com*. If you obtained the product from another source, such as an authorized distributor, contact them for help first. If they are unable to help, contact us.

Also, visit:

- The Micro Focus Community Web site, where you can browse the Knowledge Base, read articles and blogs, find demonstration programs and examples, and discuss this product with other users and Micro Focus specialists.
- The Micro Focus YouTube channel for videos related to your product.

## Information Needed by Micro Focus SupportLine

When contacting Micro Focus SupportLine, please include the following information if possible. The more information you can give, the better Micro Focus SupportLine can help you.

- The name and version number of all products that you think might be causing an issue.
- Your computer make and model.
- System information such as operating system name and version, processors, and memory details.
- Any detailed description of the issue, including steps to reproduce the issue.
- Exact wording of any error messages involved.
- Your serial number or works order (WO) number.

To find out these numbers, look in the subject line and body of your Electronic Product Delivery Notice email that you received from Micro Focus, or on the box in which the product was supplied, and on the red card supplied in the DVD case .

# Index