

# 1

## Overview of *extend*<sup>®</sup> Version 7.2

---

### Key Topics

Introduction .....	1-2
Interoperating With ACUCOBOL-GT .....	1-4
Enhanced Access to Data .....	1-6
Compatibility With Other COBOLs.....	1-6
Performance .....	1-9
ACUCOBOL-GT Compiler .....	1-10
ACUCOBOL-GT Runtime.....	1-17
AcuBench.....	1-22
AcuConnect.....	1-24
AcuODBC and AcuODBC Server .....	1-27
AcuServer.....	1-29
AcuSQL.....	1-31
Acu4GL.....	1-32
Technical Support.....	1-33

---

## Introduction

Acucorp is pleased to announce *extend*<sup>®</sup>7.2, the next release in our Interoperability Series. Among many enhancements to the ACUCOBOL-GT<sup>®</sup> Development System and *extend*7 product suite, Version 7.2 introduces the following top features:

- A new AcuSort utility program distributed with the runtime provides a way to perform external sort and merge operations.
- New compile options make Java and .NET interoperability easier to implement.
- AcuBench<sup>®</sup> now includes context-sensitive help, the ability to build an AcuBench workspace or program from the Windows command line, and a new autoload capability to ease the creation of combo boxes, grids, and list boxes.
- AcuSQL<sup>®</sup> now includes support for the popular open source database, MySQL, plus the ability to call stored procedures using embedded SQL.
- ACUCOBOL-GT Thin Client users can take advantage of the new automatic update feature, new capabilities in the C\$COPY library routine to access files on the display host, and the ability to set a new thin client command line variable to control the trace level.

In addition, you will also find these new features:

- The ability to use the Windows NT and UNIX native logon facilities with the new SECURITY\_METHOD configuration variable in AcuConnect<sup>®</sup>, AcuODBC<sup>®</sup>, and AcuServer<sup>®</sup>.
- WIN\$PRINTER has been enhanced to allow access to more settings in the Windows printer dialog box, including page columns and print margins.
- Debugging in a distributed processing environment
- Support for working with reference modified arrays in the runtime debugger

- A new option to order the ACUCOBOL-GT runtime in shared libraries instead of the statically linked version
- Configurable error reporting from the compiler that includes enhanced parsing and error reporting capabilities. For example, you can specify the number of errors to report before the compiler exits and allow the compiler to skip entire sections of code once an error is detected.
- AcuConnect, AcuODBC, and AcuServer include system logging enhancements that allow programmers to write to the system log or event notification system using the new C\$SYSLOG library routine.
- Dozens of performance enhancements, including simple math operations, native code generation, sort performance, improved processing time for the profiler utility, and compiling to a remote server

This *Release Overview* highlights of new features in Version 7.2 for the following technologies:

- ACUCOBOL-GT Compiler
- ACUCOBOL-GT COBOL Virtual Machine™ (runtime)
- AcuBench
- AcuConnect
- AcuODBC and AcuODBC Server
- AcuServer
- AcuSQL
- Acu4GL®

Unless otherwise indicated in this overview, please refer to the respective product manuals for complete details. See each product's online *Release Notes* for information not included in the manuals.

## Interoperating With ACUCOBOL-GT

Refer to the following sections for details on:

- Interoperating With Java
- Interoperating With .NET

### Interoperating With Java

Previously, to call a COBOL program from Java, Java programmers used the ACUCOBOL-GT native Java Application Programming Interface (API). Like the .NET API, the Java API gives programmers low-level control over the COBOL program. Starting with Version 7.2, programmers can use one of two new compiler options to provide Java-to-COBOL interoperability instead. The new options are “--javaclass” and “--javamain”.

When you specify the “--javaclass” option at compile time, the compiler generates a “.java” file in addition to a “.acu” file. This “.java” file is a Java class that calls the COBOL program being compiled. Java programmers can then invoke this class as they would any native Java code. “--javamain” is the same as “--javaclass” except that it generates a class with a main method added.

Refer to section 2.2.1 of *A Guide to Interoperating with ACUCOBOL-GT* for more details on the Java compiler options.

By offering compiler options as well as a native Java API, ACUCOBOL-GT makes Java interoperability easier and more flexible. It also mirrors the .NET solution, making multi-language interoperability intuitive for developers. As before, you can call Java from COBOL using our C\$JAVA library routine. This has not changed in Version 7.2.

## Interoperating With .NET

Starting with Version 7.2, ACUCOBOL-GT includes two additional ways to call a COBOL program from .NET:

1. Via two new .NET compiler options: “--netdll” and “--netexe”.  
COBOL programmers can use these compiler options to automatically generate Microsoft Intermediate Language (MSIL) objects in addition to their COBOL objects at compile time. .NET programmers can then invoke the MSIL objects as they would any native .NET resource, gaining access to the ACUCOBOL-GT runtime and COBOL program automatically.

The “--netexe” compiler option generates a .NET executable file for command line execution.

The “--netdll” compiler option generates a .NET dynamic link library (DLL) that gives .NET assemblies—both executables and DLLs—an interface to your COBOL program.

See section 5.4.1 of *A Guide to Interoperating with ACUCOBOL-GT* for more details on these compiler options.

2. Using a native .NET interface, `wrunet.dll`, programmers can interact with the COBOL program at the API level. This interface contains a class, `CVM`, that encapsulates the ACUCOBOL-GT runtime. With the `CVM`, the .NET programmer can programmatically start the ACUCOBOL-GT runtime and control a COBOL program from a very low level. .NET programmers can include the interface in a Visual Studio .NET project and take advantage of the native .NET development environment.

See section 5.4.2 of *A Guide to Interoperating with ACUCOBOL-GT* for more details on the .NET API.

These new features make interoperability with .NET much easier than ever before, making your programs more powerful—and for ISVs, more marketable—than standalone COBOL programs.

As before, you can call a COBOL program from .NET using our .NET proxy and COM server, and you can call .NET from COBOL using our **NETDEFGEN** utility and COBOL language extensions. With Version 7.2, you can use the **Ctrl** key to select multiple classes in **NETDEFGEN**.

## Enhanced Access to Data

When using the EXTFH interface, you can now indicate whether the file system you are accessing can handle variable length files. Three new configuration variables correspond to indexed, relative, and sequential files:

A\_EXTFH\_VARIABLE\_IDX  
A\_EXTFH\_VARIABLE\_REL  
A\_EXTFH\_VARIABLE\_SEQ

When the file system does not process variable length files, set these configuration variables to “0” and the EXTFH interface treats variable length records as fixed length.

With these options, ACUCOBOL-GT allows you to readily work with file systems like SFS that would otherwise not process variable length files.

For more information, see Appendix H in Book 4, *ACUCOBOL-GT Appendices*.

## Compatibility With Other COBOLs

ACUCOBOL-GT has been enhanced to make it more compatible with other COBOL systems.

Refer to the following section for details on:

- Compiler Allows PIC X(n) COMP-5
- Screen Attribute Library Routines
- New SORT-RETURN Special Register

- FILE\_ALIAS\_PREFIX
- WHEN = and WHEN NOT = in an EVALUATE Statement

## Compiler Allows PIC X(n) COMP-5

To provide compatibility with other COBOLs, it is now possible to compile programs that contain PIC X(n) COMP-5 data items without an error. In programs that assign an alphanumeric literal value to a PIC X(n) COMP-5 item, the results are undefined.

For example, the following code fragment causes NUM to have an undefined number and the resulting value for the last line will be "100":

```
NUM PIC X(5) COMP-5.  
ALPHANUM PIC X(9) .  
MOVE "ABC" TO NUM.  
MOVE "1,000" TO NUM.  
MOVE ALPHANUM TO NUM.  
MOVE "100" TO NUM.
```

Note that a PIC X(n) data item used with COMP-5 cannot be signed.

## Screen Attribute Library Routines

The ACUCOBOL-GT runtime includes several new library routines that allow the runtime to read and write characters and attributes on the screen. For more information, please see the section in this overview titled "New and Enhanced Library Routines."

## New SORT-RETURN Special Register

Version 7.2 includes a new special register known as SORT-RETURN, which can be used for two purposes.

- To determine the status of a SORT that's just finished. You can determine the success or failure of a SORT by examining this variable after the SORT returns. A value of "0" indicates success, and a non-zero value indicates failure.
- To interrupt a SORT that is currently running. By setting this variable in an input or output procedure, you stop SORT processing immediately after the next RELEASE or RETURN statement is performed. By setting this variable in a DECLARATIVES paragraph (if you are not using input or output procedures), you stop SORT processing immediately after the next implicit RELEASE or RETURN is performed.

The special register SORT-RETURN is of type SIGNED-INT. This register is primarily for compatibility purposes, and there are other ways to perform these functions in ACUCOBOL-GT. For instance, to get status on a SORT, use the file status variable of the SORT file. This gives information in addition to success or failure. Also, if you are using input procedures, you can halt a SORT more simply by returning from the procedure as if you had reached the end of the file of an end-of-file condition.

## FILE\_ALIAS\_PREFIX

This variable allows you to specify a list of strings to prefix to a file name before searching for that name in the configuration file or environment. For more information, please see the later section on "New and Enhanced Configuration Variables."

## WHEN = and WHEN NOT = in an EVALUATE Statement

The compiler now allows an equals sign ("=") in WHEN and WHEN NOT phrases in an EVALUATE statement.

## Performance

Version 7.2 introduces a variety of performance enhancements to both the compiler and runtime that improve:

- Math Operations
- Sorting Performance
- Native Code Generation
- Processing Time in the Profiler
- Compiling to a Remote Server

Please refer to the ECN list for details on these and other performance benefits.

### Math Operations

Code generation for the SET ADDRESS OF statement has been improved so that the compiler is more effective in recognizing variants of the literal “1”, for example, “+1” or “01”.

### Sorting Performance

The default value of the SORT\_FILES configuration variable is changed from “4” to “8” and the maximum allowable value is increased to “64”. The default value of the SORT\_MEMORY configuration variable is also increased from “4” to “32”. The addition of two COMP-3 data items and numeric key comparisons for the runtime SORT/MERGE verbs also make sorting operations more efficient.

## Native Code Generation

Version 7.2 includes several native code improvements, including a new “De” compiler switch, which ensures that move code works properly for any machine architecture, regardless of how data is aligned. Binary comparisons between signed and unsigned values and for the comparison of native data types to zero (or null) have also been improved.

## Processing Time in the Profiler

To reduce the file size and processing time for the “acumon.xml” file, the runtime profiler no longer writes out records for paragraphs that have zero execution count and zero execution time. To collect zero count paragraphs, use the new “-p0” runtime flag. To track the real time spent executing and, at shutdown, to write the total to the runtime error output file, use the new “-time” option.

## Compiling to a Remote Server

In previous versions, when compiling to a remote object file, the compiler opened the object file directly and wrote to it in several parts, seeking various locations in the file. The compiler now detects that the object file is remote and creates a local temporary object file first. When the compilation is successful, the compiler then copies that file to the remote computer, saving time.

## ACUCOBOL-GT Compiler

Refer to the following section for details on:

- Compiler Options in Listing File
- Compiler Options in Object Header
- New Compiler Switches

- ACUCOBOL-GT Utilities
- New and Enhanced Library Routines

## Compiler Options in Listing File

Options given to the compiler are now listed on the first page of the listing file, if one is created. An example of the format for this listing is as follows:

```
iobench.cbl      Sat Sep 18 06:55:54 2005 ACUCOBOL-GT v7.2.0 Page: 0001
..\bin\ccbl32 -Lfo @.lst iobench.cbl
CBLFLAGS: -v -Za
COPYPATH: my path;.
```

Note that the CBLFLAGS and COPYPATH lines are not listed if those variables are not set.

## Compiler Options in Object Header

The options used to compile a COBOL program are now embedded in the header of the resulting object file. You can use a new “cblutil -info” option (“-x”) to display this information.

## New Compiler Switches

Version 7.2 introduces the following compiler switches.

<b>-De</b>	The compiler generates different code for moves of binary data (such as pointers) depending on whether the variable is on an appropriate boundary for the compile switches given. In cases where the definition of a binary linkage data item includes the SET ADDRESS verb, this new flag ensures that the move code works properly for any machine architecture, regardless of how the data is aligned.
<b>--javaclass</b>	Generates a “.java” file in addition to a “.acu” file. The “.java” file has the same prefix as the “.acu” file and is placed in the same directory. This “.java” file is a Java class that calls the COBOL program being compiled.
<b>--javamain</b>	Generates a “.java” file in addition to a “.acu” file. The “.java” file has the same prefix as the “.acu” file and is placed in the same directory. This “.java” file is a Java class with a main method added. The class calls the COBOL program that is being compiled.
<b>--netexe</b>	Generates a .NET executable file for command line execution. The name of the executable is the name of the program followed by “.exe.” All valid ACUCOBOL-GT command line options can be specified with the executable, as well as special Linkage Section parameters.
<b>--netdll</b>	Generates a .NET dynamic link library (DLL) that gives .NET assemblies—both executables and DLLs—a programmatic interface to your COBOL program. All COBOL entry points are exposed as .NET methods along with ACUCOBOL-GT runtime properties and methods. This allows .NET programmers to set ACUCOBOL-GT command options and call runtime interfaces from their .NET assembly. Refer to <i>A Guide to Interoperating with ACUCOBOL-GT</i> , section 4.5.1 for more details.

- Qm** Specifies the number of errors the compiler reports before it exits, allowing the compiler to skip entire sections of code when it finds an error. The option must be followed by a positive numeric argument, which is the maximum number of errors the compiler reports before it exits. The default value is “100”.
- Qp** Allows the compiler to skip entire sections of code when it finds an error. When this option is used, the compiler skips to the next period (“.”) when an error is detected. This option can be useful when an error causes subsequent cascading errors that are actually not errors. When the subsequent “errors” are not reported, the developer can focus on the first real error in the COBOL program.

## ACUCOBOL-GT Utilities

Refer to the following section for details on:

- AcuSort
- Adding comments with cblutil

### AcuSort

A new **AcuSort** utility program distributed with the ACUCOBOL-GT runtime enables you to sort or merge Vision indexed, relative, binary sequential, and line sequential files. In some cases, AcuSort can offer performance improvements over the SORT verb when working with large files. This external sort function is invoked from the command line and can take a number of parameters. **AcuSort** parameters may appear directly on the command line, or they may be included in a separate text file.

For details about how to use this new utility, please refer to section 3.8, “External Sort Utility—AcuSort,” in the *ACUCOBOL-GT User's Guide*.

## Adding comments with cblutil

You can now embed a comment in an object library created with **cblutil**, the object file utility. When you use the new “-c” flag to add a comment, that comment is displayed any time the object library is examined with the “-info” flag.

In the following example, a comment is embedded in a library called “mylib”:

```
cblutil -lib -o mylib -c "Special library for testing." obj1.acu
      obj2.acu
```

More information is available in section 3.2.2, Book 1, *ACUCOBOL-GT User's Guide*.

## New and Enhanced Library Routines

In addition to the descriptions in this *Release Overview*, ACUCOBOL-GT runtime library routines are described in complete detail in Appendix I of Book 4, *ACUCOBOL-GT Appendices*.

### Screen attribute library routines

The ACUCOBOL-GT runtime includes several new library routines that allow the runtime to read and write characters and attributes on the screen. The following 16 library routines allow you to implement this new functionality.

CBL_CLEAR_SCR	Clears the entire screen using a specified character and attribute
CBL_GET_CSR_POS	Returns the cursor position
CBL_GET_SCR_SIZE	Returns information about the size of the screen
CBL_READ_SCR_ATTRS	Reads a string of attributes from the screen
CBL_READ_SCR_CHARS	Reads a string of characters from the screen

CBL_READ_SCR_CHATTRS	Reads a string of characters and their attributes from the screen
CBL_SET_CSR_POS	Moves the cursor
CBL_SWAP_SCR_CHATTRS	Swaps a string of characters and their attributes with a string from the screen
CBL_WRITE_SCR_ATTRS	Writes a string of attributes to the screen
CBL_WRITE_SCR_CHARS	Writes a string of characters to the screen
CBL_WRITE_SCR_CHARS_ATTR	Writes a string of characters to the screen, giving them all the same attribute
CBL_WRITE_SCR_CHATTRS	Writes a string of characters and their attributes to the screen
CBL_WRITE_SCR_N_ATTR	Writes a specified attribute to a string of positions on the screen
CBL_WRITE_SCR_N_CHAR	Writes a specified character to a string of positions on the screen
CBL_WRITE_SCR_N_CHATTR	Writes a specified character and attribute to a string of positions on the screen
CBL_WRITE_SCR_TTY	Writes a string of characters to the screen starting at the current position and scrolling

Please refer to Appendix I in *ACUCOBOL-GT Appendices* for the syntax and descriptions of these routines.

## C\$COPY

The C\$COPY library routine “@[DISPLAY]:” function has new capabilities to access files on the display host (thin client). If a file name on the client starts with a special directory specifier, the thin client attempts to locate the file in special Windows directories. The special directory names are as follows:

<APPDATA>	C:\Documents and Settings\ <user>\Application Data</user>
<COMMON_APPDATA>	C:\Documents and Settings\All Users\Application Data

<COMMON_DOCUMENTS>	C:\Documents and Settings\All Users\Documents
<DESKTOP>	C:\Documents and Settings\<user>\Desktop
<LOCAL_APPDATA>	C:\Documents and Settings\<user>\Local Settings\Application Data
<MYDOCUMENTS>	C:\Documents and Settings\<user>\My Documents

For syntax information on this new C\$COPY feature, please refer to Appendix I in *ACUCOBOL-GT Appendices*.

## C\$SYSLOG

A system logging facility has been added to the runtime, allowing COBOL programmers to write to the system log or event notification system (on Windows) in the event of a serious error that administrators need to know about.

The new C\$SYSLOG system library routine can be used to open, write to, and close the system log. This routine sends messages to **syslog** on UNIX systems that have the **syslog** function. On UNIX systems without the **syslog** function, and on other non-Windows operating systems, this routine sends messages to the console. On Windows systems, this routine sends messages to the event log, which can be viewed using the Event Viewer applet available on Windows.

The runtime also sends messages to the system log when it detects broken files. This function allows COBOL programmers to notify system administrators automatically when a broken file is detected, instead of relying on individual users to report such errors.

For detailed syntax and parameter information regarding the use of this library routine, please refer to Appendix I in *ACUCOBOL-GT Appendices*.

## WIN\$PRINTER

The W\$PRINTER routine has been enhanced to:

- Allow for greater control of the page columns

- Make the page margins set in the standard Windows Print Setup dialog box available to the runtime

You can use these new features as follows:

- Specify the start position of the page as an absolute value from the left edge using WPRTUNITS-CENTIMETERS-ABS, WPRTUNITS-INCHES-ABS, and WPRTUNITS-CELLS-ABS.
- Use the new WINPRINT-SETUP-USE-MARGINS op-code (instead of WINPRINT-SETUP) so the runtime can use the page margins specified in the Windows Print Setup dialog box.
- Use the new WINPRINT-GET-MARGINS op-code to get the current value of the page margin.

These new capabilities extend WIN\$PRINTER so you can take advantage of additional printing options enabled in Windows and have greater control over printed output.

For complete information on these new features, refer to Appendix I in Book 4, *ACUCOBOL-GT Appendices*.

## ACUCOBOL-GT Runtime

Refer to the following sections for details on:

- Added Support for Character Sets With “-Q”
- Reference Modified Arrays in Runtime Debugger
- New and Enhanced Configuration Variables
- New Runtime Command Line Options

## Added Support for Character Sets With “-Q”

The “-Q” option that allows you to simulate traditional style printing using the Windows print spooler has a new CHARSET configuration entry, which enables you to specify any of the character sets defined in “fonts.def”.

For complete information on this new feature, refer to the section, “Routine to Handle the Windows Print Spooler” in Appendix I in Book 4, *ACUCOBOL-GT Appendices*.

## Reference Modified Arrays in Runtime Debugger

You can now work with reference modified items in arrays in the debugger. For example, if you want to display the fifth through eighth characters in the tenth item in an array, you can use the syntax:

```
d table-item(10)(5:4)
```

## New and Enhanced Configuration Variables

In addition to the descriptions in this *Release Overview*, all ACUCOBOL-GT runtime configuration variables are described in complete detail in Appendix H of Book 4, *ACUCOBOL-GT Appendices*.

A\_EXTFH\_VARIABLE\_IDX  
A\_EXTFH\_VARIABLE\_REL  
A\_EXTFH\_VARIABLE\_SEQ

These variables indicate whether the file system you are accessing with the the EXTFH interface can handle variable length files. Setting this variable to the default of “1” (on, true, yes) causes the EXTFH interface to pass the minimum and maximum record lengths to the file system for variable length files as defined in the COBOL program. Setting this variable to “0” (off, false, no) causes the EXTFH interface to ignore the variable record length defined in the COBOL program, instead passing a record of length equal to the maximum record length.

You can specify the variable separately for indexed, relative, and sequential files. For example:

```
A_EXTFH_VARIABLE_IDX=0  
A_EXTFH_VARIABLE_REL=0  
A_EXTFH_VARIABLE_SEQ=1
```

When the file system does not process variable length files, set these configuration variables to “0” and the EXTFH interface treats variable length records as fixed lengths.

If the file system does process variable length files, set the configuration variables to “1” (or do not set them at all).

## AXML\_IGNORE\_EMPTY\_DATA

When writing data, previous versions of AcuXML generated tags for all data items in the file. A new AXML\_IGNORE\_EMPTY\_DATA configuration variable, allows you to omit empty and zero-filled data from the XML output file. If you have many empty records, this makes the output more concise and easier to read. Refer to Appendix H in Book 4, *ACUCOBOL-GT Appendices*, for more information on this new variable.

## FILE\_ALIAS\_PREFIX

This variable allows you to specify a list of strings to prefix to a file name before searching for that name in the configuration file or environment. Data and code file search paths are described in more detail in section 2.7.2 of the *ACUCOBOL-GT User's Guide*.

When searching for a file alias, the runtime constructs the file alias name by prepending the first string listed in FILE\_ALIAS\_PREFIX to the file name and searches for that name in the environment or configuration file. If the name is not found, the runtime constructs a new name by prepending the second string in FILE\_ALIAS\_PREFIX to the file name and searches for that alias.

This process is repeated with each string in FILE\_ALIAS\_PREFIX until a file alias name is found or the end of the list is reached.

## FREEZE\_AX\_EVENTS

During the processing of an ActiveX event, the Windows and thin client runtimes attempt to suspend subsequent ActiveX events until the first event has completed. The thin client runtime also attempts to suspend ActiveX events whenever the application is not processing an ACCEPT statement. To suspend and resume ActiveX events, the runtime calls the ActiveX function `IOleControl::FreezeEvents()`.

You can disable the `FreezeEvents()` logic by setting a new `FREEZE_AX_EVENTS` runtime configuration variable. For more information, please refer to Appendix H in *ACUCOBOL-GT Appendices*.

## IO\_FLUSH\_COUNT

Use this variable to specify how often the runtime should flush pending screen output during file operations. When set to a positive value, the variable indicates the number of file operations to perform between each screen flush. By default, `IO_FLUSH_COUNT` is set to “20”.

For optimal performance, set `IO_FLUSH_COUNT` to zero (“0”). When `IO_FLUSH_COUNT` is set to zero, COBOL file verbs do not automatically flush pending screen output. To reduce unexpected screen behavior, however, leave this variable at its default setting. The overhead at the default setting is small.

## NT\_OPP\_LOCK\_STATUS

This configuration variable controls how files on a shared drive are opened if you are working in the Windows opportunistic locking mode. `NT_OPP_LOCK_STATUS` can take one of four values: `CREATEFILE`, `SAFE`, `GETFILETYPE`, or `FAST`.

The default value, `SAFE`, is a synonym for `CREATEFILE`. `FAST` is a synonym for `GETFILETYPE`. When the variable is set to either of the latter values, the runtime uses the fast method of opening files.

Note that a setting of `FAST` (or `GETFILETYPE`) may cause file corruption if your Windows installations are not completely up to date with all available patches from Microsoft, particularly those related to opportunistic locking.

## SORT\_FILES and SORT\_MEMORY

In Version 7.2, the default value of the SORT\_FILES configuration variable is changed from “4” to “8”, and the maximum allowable value is increased to “64”. The minimum allowable value remains “4”.

The default value of SORT\_MEMORY is changed from “4” (256KB) to “32” (2MB), and the maximum allowable value is increased to “16384” (1GB). The minimum allowable value remains “1” (64KB).

## V71\_FONT\_WIDTHS

Windows has a function called GetTextMetrics that returns information about a font. Experimentation has shown that the “maximum character width” data returned is often inaccurate, sometimes by very large margins (for example, getting back “42” pixels when the correct answer is “15”). The runtime now validates the data returned by the Windows function and corrects it when it is too large.

Because this change can affect previously deployed programs, the change takes effect only when the programs are recompiled or the change is specifically enabled. The precise rules depend on the setting of a new V71\_FONT\_WIDTHS configuration variable:

- 1 (default) The change is enabled for programs using Version 7.2 or later semantics. In other words, the program has been compiled with Version 7.2 or later and the command line does not contain a compiler option for pre-7.2 semantics.
- 0 The change is enabled.
- 1 The change is disabled and the Version 7.1 and earlier font-measuring code is used.

## New Runtime Command Line Options

Version 7.2 introduces two new command line options:

- |               |   |
|---------------|---|
| <b>-p0</b>    | To reduce the file size and processing time for the “acumon.xml” file, the runtime profiler no longer writes out records for paragraphs that have zero execution count and zero execution time. Use the new “-p0” runtime flag to collect zero count paragraphs and take advantage of this performance improvement. |
| <b>--time</b> | This option causes the runtime to keep track of the real time spent executing and, at shutdown, writes the total to its error output file.<br><br>More information about the “--time” option can be found in section 2.2 in Book 1, <i>ACUCOBOL-GT User's Guide</i> .   |

## AcuBench

Refer to the following sections for details on these AcuBench enhancements:

- Context-sensitive Help
- Command Line Interface
- Autoload Combo Boxes, Grids, and List Boxes

## Context-sensitive Help

The help facility within AcuBench has been expanded to provide context-sensitive information. When you press **F1** from within the integrated development environment (IDE), AcuBench opens the help files to show information relevant to your current context. If you select a verb in the Code Editor, a control in the Screen Designer, or a configuration variable in the Configuration File Editor, for example, and press **F1**, AcuBench opens the chapter and section of the four-book ACUCOBOL-GT manual set or the section in the AcuBench manual that addresses the selected item.

Note that you must have the HTML documentation set installed to take advantage of context-sensitive help.

For more information, see “Accessing online documentation” in section 2.2 of the *AcuBench User’s Guide*.

## Command Line Interface

You can now build an AcuBench workspace or program from the Windows command line (using the 32-bit **cmd** shell). This makes it possible for you to create Windows batch (script) files to build multiple workspaces, or build a selection of programs taken from multiple workspaces, automating the build process. Build information is automatically appended to a log file, with date and time information included, to help administrators monitor the build process. The command line interface also makes it easier to manage applications developed in multiple AcuBench workspaces, reducing the need to create very large workspaces.

For more information, see “The command line interface to AcuBench” in section 3.5 of the *AcuBench User’s Guide*.

## Autoload Combo Boxes, Grids, and List Boxes

Using the Drag and Drop interface, you can now create combo boxes, grids, and list boxes and have AcuBench generate the code to populate those controls at runtime. When you use Drag and Drop to create any of these controls (including paged-style grids and list boxes), you can choose the data used to populate the control, the way that data is read into the control, and other details. AcuBench then draws the control on the screen form and generates the code necessary to load the control with the data that you have specified.

For more information, see Chapter 11, Section 11.6.3, “Autoload Controls”, in the *AcuBench User’s Guide*.

Note that when you use Drag and Drop to create paged grids and list boxes, AcuBench generates code in a new manner. Code for these controls is generated into the Event paragraph, allowing users to modify and maintain

the code directly. Please consult Chapter 11, Section 11.6.3.2, “Creating a paged autoload control”, in the *AcuBench User’s Guide* before creating one of these paged controls.

## AcuConnect

The AcuConnect client/server technology for Version 7.2 includes the enhancements described in the following paragraphs. AcuConnect users benefit from features that enhance the server’s capabilities, including new system logging and security features. Thin client users can take advantage of the new automatic update feature and several performance enhancements.

### The AcuConnect Server (acurcl)

The following sections outline new features for the **acurcl** executable.

#### Writing to the system log

If your operating system (OS) supports it, the UNIX version of AcuConnect can now write to the **syslog** facility when informational or error messages need to be reported. If your OS does not support **syslog**, AcuConnect writes this information to the console.

The Windows version of AcuConnect can now write to the event log of other machines in the network. A new WINNT\_EVENTLOG\_DOMAIN configuration variable contains the Universal Naming Convention (UNC) name of the computer to which all event log messages should be sent. This must be set in the configuration file and is not changeable.

For details, refer to section 8.5, “Event Logging,” in the *AcuConnect User’s Guide*.

#### Using the Windows NT and UNIX native logon facilities

AcuConnect can now use the native logon facility on Windows NT and on UNIX. Users can use their regular passwords when they connect to servers on these machines, instead of having a different password in the AcuAccess

file (or remembering to keep the password in the AcuAccess file coordinated with the native password). This ability has been available for Windows servers for many versions and is new to UNIX. This feature also works for UNIX servers that use shadow passwords, but does not use the newer pluggable authentication module (PAM) libraries.

A new SECURITY\_METHOD configuration variable replaces the NT\_SECURITY variable (which has been removed). Detailed information about this variable can be found in section 3.4.9, “SECURITY\_METHOD,” in the *AcuConnect User’s Guide*. Note that this feature is also available with AcuServer and AcuODBC.

Some UNIX machines have the ability to restrict access to the machine based on various parameters. If you want to include those restrictions in AcuConnect, set the new USE\_SYSTEM\_RESTRICTIONS configuration variable to the value “TRUE”. This configuration variable is available only on UNIX servers, and is used only if SECURITY\_METHOD is “LOGON”. See your UNIX administration manuals for information on this restriction feature.

## Setting automatic refresh on the control panel

An automatic refresh function is now available on the Info tab of the AcuConnect graphical control panel. When you connect to a server for a query, a new Auto Refresh check box appears. Setting this check box enables an entry field in which you can enter the number of seconds between each automatic refresh action.

In automatic refresh mode, a text message appears in the lower right corner of the Info tab, displaying the time to the next automatic refresh. The time is updated every second.

## Receiving start-up status codes on UNIX

On UNIX machines, AcuConnect now returns its status to the operating system on a start-up attempt. AcuConnect returns a status code of “0” after a successful start. When **acurcl** fails to start, it returns one of the status codes described in section 3.6.1, “Starting acurcl on a UNIX Server,” in the *AcuConnect User’s Guide*.

The way to get this status depends on your shell. When you use the Bourne shell (or a compatible shell), echo \$? to see the return status of any program executed from the shell.

## Debugging in a distributed processing environment

You can now debug remote programs in a number of ways: with a TTY, an xterm, or the thin client. Two new configuration variables control remote program debugging: ACUCONNECT\_DEBUG\_METHOD and ACUCONNECT\_DEBUG\_METHOD\_STRING.

ACUCONNECT\_DEBUG\_METHOD may take one of the following values:

XTERM  
TERMINAL  
THIN or THINCLIENT

These ACUCONNECT\_DEBUG\_METHOD settings determine the values that the ACUCONNECT\_DEBUG\_METHOD\_STRING configuration variable can take. Detailed information about how to set these two configuration variables can be found in section 4.2.2.1, “ACUCONNECT\_DEBUG\_METHOD,” and section 4.2.2.2, “ACUCONNECT\_DEBUG\_METHOD\_STRING,” in the *AcuConnect User’s Guide*.

## Thin Client

The following paragraphs describe new features for Acucorp’s Thin Client technology. In addition to these features, several performance improvements are included in Version 7.2.

## Using the automatic update feature

The ACUCOBOL-GT Thin Client has an automatic update feature that determines whether the **acuthin** executable on the client is compatible with the server components (the AcuConnect executable **acurcl** and the ACUCOBOL-GT runtime executable **runcbl**) and triggers an automatic update of the client piece if it is not. With this feature, the thin client can

automatically download and install a new version of the client piece that matches the server version. It can then restart the user's application using the newly installed thin client.

For detailed information about configuring and using the automatic update feature, please refer to section 6.7, "Thin Client Automatic Update," in the *AcuConnect User's Guide*.

### Adding a trace command line file variable

A new atc-trace variable can be set in the thin client command line file. The variable must be followed with the desired trace level number. This variable is equivalent to the "acuthin -t" command line option.

## AcuODBC and AcuODBC Server

Refer to the following sections for details on AcuODBC enhancements:

- Using the Windows NT and UNIX Native Logon Facilities
- System Log Facility
- Unrestricted Number of Key Columns

### Using the Windows NT and UNIX Native Logon Facilities

AcuODBC Server can now use the native logon facility on both Windows NT and UNIX machines. This means that users can use their usual passwords when connecting to servers on these machines, instead of having a different password in the AcuAccess file (or remembering to keep the password in the AcuAccess file coordinated with the native password).

This ability, long available on Windows servers, is now also available on UNIX servers that use shadow passwords. UNIX servers that use the newer PAM libraries are not supported.

To support this enhancement, a new SECURITY\_METHOD configuration variable replaces NT\_SECURITY (which has been removed). This means that current Windows users of AcuODBC need to update their configuration files for the new version.

As an enhancement for Windows users, if you are using the LOGON security method, a new WIN32\_LOGON\_DOMAIN configuration variable allows you to specify a domain. Set the variable to the name of a Windows domain to cause the server to log the user into the specified domain.

Some UNIX machines can restrict access based on various parameters. The new USE\_SYSTEM\_RESTRICTIONS configuration variable allows you to include those restrictions in AcuODBC Server when you use the LOGON security method on UNIX.

For more information, see “SECURITY\_METHOD” in section 8.4.1 of the *AcuODBC User's Guide*.

## System Log Facility

The UNIX version of AcuODBC Server now writes informational or error messages to the **syslog** facility (if your operating system supports it) or to /dev/console (otherwise). The Windows version of AcuODBC Server can now write to the event log of other machines in the network. Messages that used to go only to the event log on the local machine are now available on all operating systems.

See section 8.1 of the *AcuODBC User's Guide* for more information.

## Unrestricted Number of Key Columns

Earlier versions of AcuODBC included a restriction in the number of key columns allowed in an XFD file. Files with more than 16 columns in a key could not be parsed. This restriction has been lifted.

## AcuServer

Refer to the following sections for details on enhancements to AcuServer:

- Security Enhancements
- Improved Reporting
- Windows Graphical Control Panel Enhancements

### Security Enhancements

AcuServer Version 7.2 includes the following security enhancements:

- Security options for UNIX and Windows. AcuServer has always allowed Windows users to choose between running AcuServer security and operating system (Windows NT) security. As of Version 7.2, it gives UNIX users the same option. The configuration variable that you use to enable native security, NT\_SECURITY, has been renamed SECURITY\_METHOD as a result.

What this means is that users can use their usual passwords when connecting to servers on these machines, instead of having a different password in the AcuAccess file or remembering to keep the password in the AcuAccess file coordinated with the native password. This works for UNIX servers that use shadow passwords. This feature does not use the newer PAM libraries.

---

**Caution:** If you are upgrading a Windows server, you must change your configuration file, replacing the NT\_SECURITY variable with the new SECURITY\_METHOD variable.

---

- Ability to include UNIX machine restrictions. Some UNIX machines have the ability to restrict access to the machine based on various parameters. If you want to include those restrictions in AcuServer, set the new USE\_SYSTEM\_RESTRICTIONS configuration variable to “TRUE”. This configuration variable is applied only on UNIX servers, and is used only if SECURITY\_METHOD is set to “LOGON”.

- Ability to specify Windows logon domain. A new WINNT\_LOGON\_DOMAIN configuration variable allows you to specify the domain onto which to log users. In the past, users were always logged onto the NULL domain. This variable is applied only on Windows servers and is used only if SECURITY\_METHOD is set to "LOGON".

These security improvements are designed to protect your corporate data from unauthorized access and to simplify security administration. Refer to section 4.2.2 of the *AcuServer User's Guide* for more information on any of these new configuration variables.

## Improved Reporting

To simplify system management, AcuServer Version 7.2 includes the following new reporting features:

- Error reporting on startup. AcuServer now returns a status to the operating system when it starts or fails to start. This eases administration and troubleshooting. Refer to section 8.2.3 of the *AcuServer User's Guide* for a description of the return codes.
- New UNIX **syslog** capability. If your OS supports it, the UNIX version of AcuServer now writes to the **syslog** facility when informational or error messages need to be reported. If your OS does not support **syslog**, AcuServer writes these messages to /dev/console instead.
- New event log capabilities. The Windows version of AcuServer now has the ability to write to the event log of other machines in the network. On Windows, there is a new WINNT\_EVENTLOG\_DOMAIN configuration variable for this purpose.

In addition, messages that used to go only to the local machine event log are now available on all operating systems.

Refer to sections 4.2.2 and 8.2.2 of the *AcuServer User's Guide* for more information on **syslog** and the event log.

## Windows Graphical Control Panel Enhancements

The AcuServer graphical control panel has been enhanced in the following ways:

- Statistics for master and slave machines. The control panel now includes statistics for servers that are controlled by the server that was queried in a master/slave relationship.
- Automatic refresh. The control panel has a new automatic refresh feature on the Info tab.

These enhancements make the control panel more useful and informative. Refer to section 5.6 of the *AcuServer User's Guide* for more information.

## AcuSQL

Refer to the following sections for details on enhancements to AcuSQL:

- Support for MySQL
- Support for Stored Procedures
- Performance Option

### Support for MySQL

With Version 7.2, AcuSQL supports the popular open source database, MySQL, as well as DB2 and Microsoft SQL Server. You access MySQL from your ACUCOBOL-GT application by embedding ISO/ANSI SQL 92 compliant SQL statements into your COBOL program and pre-compiling it with the AcuSQL pre-compiler as usual.

## Support for Stored Procedures

Starting with Version 7.2, you can call stored procedures using embedded SQL. A stored procedure is a pre-compiled program (or procedure) that is physically stored within a database. The advantage of a stored procedure is that it is run directly by the database engine, which reduces the processing burden on the client and improves performance. AcuSQL supports two types of stored procedures: those that return no result sets, and those that return a single result set. AcuSQL supports both output parameters and return code values. For information on calling stored procedures, refer to section 2.5.2 of the *AcuSQL User's Guide*.

## Performance Option

The use of SCROLL cursors in your application can sometimes affect performance when connecting to Microsoft SQL Server. You can now turn off the use of the SCROLL phrase using a new `ASQL_SCROLL_CURSOR` configuration variable. Refer to section 4.2.1 of the *AcuSQL User's Guide* for more information on this variable.

## Acu4GL

Refer to the following sections for details on enhancements to Acu4GL:

- Number of Key Columns Raised
- Support for Mixed-case Identifiers

## Number of Key Columns Raised

The maximum number of key columns that can be parsed from an XFD file has been increased from 16 to unlimited. Note that many databases have their own restrictions about the number of columns allowed in a primary key. This does not supersede those restrictions; rather, Acu4GL now supports those databases that have higher or no restrictions.

## Support for Mixed-case Identifiers

Identifiers in SQL-92 are not case-sensitive; however, database drivers that do not conform strictly to SQL-92 may require mixed-case identifiers. Starting with Version 7.2, Acu4GL supports mixed-case identifiers, broadening its database support even further. If your driver enforces a mixed case, refer to section D.5 of the *Acu4GL User's Guide*.

## Technical Support

In the U.S.A., you can reach Acucorp Technical Support Monday through Friday from 6:00 a.m. to 5:00 p.m. Pacific time, excluding holidays. Additional numbers are provided for obtaining technical support around the world.

### In the United States:

Phone: +1 858.689.4502  
Phone: 800 399 7220 (in the USA and Canada)  
Fax: +1 858.689.4552  
E-mail: [support@acucorp.com](mailto:support@acucorp.com)

### In Benelux:

Phone: +31 (0) 30.272.7050  
Fax: +31 (0) 30.272.7041  
E-mail: [nlsupport@acucorp.com](mailto:nlsupport@acucorp.com)

### In France:

Phone: +33 (0) 1.5334.9005  
Fax: +33 (0) 1.5334.9001  
E-mail: [frsupport@acucorp.com](mailto:frsupport@acucorp.com)

**In Germany:**

Phone: +49 (0) 89.455.659.55  
Fax: +49 (0) 89.455.659.50  
E-mail: [desupport@acucorp.com](mailto:desupport@acucorp.com)

**In Italy:**

Phone: 39 0523.071230  
Fax: 39 0523.623155  
E-mail: [itsupport@acucorp.com](mailto:itsupport@acucorp.com)

**In the United Kingdom:**

Phone: +44 (0) 20.8843.7101  
Fax: +44 (0) 20.8744.9401  
E-mail: [uksupport@acucorp.com](mailto:uksupport@acucorp.com)

If your country is not listed, visit “[www.acucorp.com/company/locations](http://www.acucorp.com/company/locations)” to find an Acucorp office or distributor near you.