

1

Overview of *extend*® Version 6.1

Key Topics

Introduction	1-2
ACUCOBOL-GT Compiler	1-4
ACUCOBOL-GT Runtime.....	1-13
AcuConnect.....	1-15
AcuODBC and AcuODBC Server	1-19
AcuServer.....	1-20
AcuSQL.....	1-21
AcuBench.....	1-22
Acu4GL.....	1-22
Technical Support.....	1-22

Introduction

Acucorp is pleased to offer Version 6.1 of our *extend*® family of technologies. This release continues Acucorp's interoperability series, extending ACUCOBOL-GT® support for XML. In addition, AcuODBC® now supports more complicated outer joins, thus enhancing the interaction between ODBC-compliant programs and the Vision file system.

In addition to the many enhancements made to the established members of the *extend* family, Version 6.1 introduces the following new features:

- Enhanced support for XML
- Additional native code options
- Enhanced support for controls
- Enhanced HP COBOL compatibility
- Enhanced support for outer joins in AcuODBC
- Improved support for transactions in AcuSQL®
- Enhanced character mapping facilities in both the compiler and in AcuConnect

This *Release Overview* contains highlights of new features in Version 6.1. Please refer to the respective product manuals for specifics on new features.

This overview provides information on new Version 6.1 features for the following technologies:

- ACUCOBOL-GT Compiler
- ACUCOBOL-GT Runtime
- AcuConnect
- AcuODBC and AcuODBC Server
- AcuServer™
- AcuSQL®

- Acu4GL®
- AcuBench®

Enhancements to the Version 6.1 *extend6* Documentation Set

The Version 6.1 online document set has been enhanced in several ways. Most notably, the new documentation set includes:

- a JavaScript™-based presentation
- a full-text search facility (JavaScript provides integral support to the search facility)

For the best results, we recommend you use a recent release of Internet Explorer or Netscape®; however any browser that supports JavaScript should work. In our testing, we noted that Mozilla 1.0.2 running under a recent release of Red Hat™ Linux® provided excellent results.

The “Books” Button Has Moved

In Version 6.0, the Contents/Index navigation pane included a Books button, which when clicked, loaded a page from which you could select any book in the *extend6* documentation set. In the Beta release of Version 6.1, the Books button has been moved to the top of the topic (main) pane.

Opening Online Documentation from a Help Button

When the online documentation is opened via a Help button in an application such as AcuBench or the AcuServer Control Panel, the associated file is opened in a window that does not initially include the Contents/Index/Search navigation pane. To add the navigation pane to the window, click on either the Show Navigation button in the upper left corner of the window, or the Show in Contents button in the upper right.

The “Show In Contents” Button

There is a new button in the upper right hand corner of the topic pane called the Show In Contents button. When you select **Show In Contents**, the help system reloads the navigation pane with the table of contents, repositioning

the table so that the current topic is visible in the list. If the topic is not included in the table of contents, the browser displays a message stating that the heading could not be found.

The ACUCOBOL-GT Combined Index

The four-volume ACUCOBOL-GT documentation set uses a single, consolidated index. The index is *very* large. On a fast PC that isn't heavily loaded, the index could take 10 or even 20 seconds to load. On heavily loaded or older systems, the index can take one or more minutes to load. Once loaded, however, you can switch back and forth between the Contents, Search, and Index navigation areas and the index will reload quickly during your help session.

ACUCOBOL-GT Compiler

Expanded Native Code Options

At Version 6.1, Acucorp introduces the ability to create native instructions for the following platforms:

- 32- and 64-bit PowerPC® for IBM® pSeries™ machines running under AIX®
- 32- and 64-bit PA-RISC machines running under HP-UX

New options to the compiler and **cblutil** support these new platforms. See the section titled “New Compiler Switches” in this *Release Overview*, and Chapters 2 and 3 of Book 1, *ACUCOBOL-GT User's Guide*, for more information.

Improved CALL Statement Performance

CALL statements to COBOL subprograms are now substantially faster when executed more than once. This is particularly noticeable when there are several COBOL objects loaded into memory at one time.

Each CALL statement now tracks its last resolution (target object, entry point, and owning thread). When these are the same in a subsequent execution of the statement, the resolution of the call uses the saved information and is much faster as a result.

A new configuration option, `CALL_HASH_SIZE`, controls this feature. The setting for `CALL_HASH_SIZE` controls the size of the hash table that tracks the call information. See Appendix H in Book 4, *ACUCOBOL-GT Appendices*, for additional information on `CALL_HASH_SIZE`.

Faster Reference Modification Code

The compiler now generates noticeably faster code for most cases involving reference modification if the offset is a simple variable and the length is a constant.

Note that compiling with the “-Za” option inhibits the optimization.

Optimized Alignment of Program Literals in Memory

The compiler uses a new algorithm for aligning program literals in memory. The alignment is the smaller of the alignment specified by the “-Da” option, which has a default value of “4”, or the “natural” alignment for the literal (the largest power of 2 that is less than or equal to the literal’s size).

This slightly improves the performance of non-native object files. For native object files, smaller and faster objects are usually produced. The data space of most programs will grow slightly due to this change.

For more information on data storage in ACUCOBOL-GT, see section 2.1.9, “Data Storage Options,” in Book 1.

Enhanced Support for XML

New Routine for Parsing XML Files

There is a new standard runtime library routine, C\$XML, that gives you low-level control over the parsing of XML data. This provides an alternative to using AcuXML.

Where AcuXML is a file system interface that transparently maps XML data to COBOL based on eXtended File Descriptors (XFDs) created at compile time, the C\$XML routine lets you define precisely which elements or attributes of the data to parse. For instance, with C\$XML, you can request the handle of the entire XML tree, the first child element of the handle passed, the next sibling element, the number of attributes in an element, the name and value of the attributes, and so on. C\$XML gives you much lower-level control over the parsing of XML data compared to AcuXML, but it requires more programming effort and XML knowledge.

See Appendix I of Book 4 for complete information on the C\$XML routine. See section 6.11 in Book 1 for more information on AcuXML.

Mapping XML Files to Internet URLs

Both AcuXML and C\$XML can now be used to read files from the Internet using standard URL syntax. For more information on using AcuXML this way, see sections 2.7.1 and 6.11.4 in Book 1. For information on using URL syntax with C\$XML, refer to Appendix I of Book 4.

More Schema Output Options for AcuXML

With AcuXML Version 6.1, you have more options for XML output that includes schemas.

For example, you now have the ability to name a schema file in XML output rather than creating a new schema file. This prevents schema files from being overwritten. This is accomplished by setting a new configuration variable, AXML_CREATE_SCHEMA to “false”.

In addition, you can now control the namespace declaration of the schema using another new variable, `AXML_SCHEMA_NAMESPACE_DATA`. This lets you control the information shown in the XML file as the schema namespace.

Both of these variables are described in Appendix H of Book 4.

International Character Mapping

In addition to using the default US-ASCII character set, you may now map the XML output created by ACUCOBOL-GT to a specific character encoding system. This is accomplished with the `AXML_ENCODING` configuration variable, described in Appendix H of Book 4.

Enhanced Support for Controls

GRID Control

A new GRID control property, `FINISH-REASON`, tracks the reason why the grid user left entry mode and entered navigation mode. The value of `FINISH-REASON` is set by the control immediately before generation of a `MSG-FINISH-ENTRY` or a `MSG-CANCEL-ENTRY` event. The value can be inquired within the event procedures for those events to fetch the reason why the user left the field she or he was entering. For more information, see section 5.10.2 in Book 2, *ACUCOBOL-GT User Interface Programming*.

WEB-BROWSER Control

In Version 6.1, the ACUCOBOL-GT `WEB-BROWSER` control has been enhanced to give your COBOL program Windows print, file, and clipboard capabilities.

There are five new special properties relating to the Windows “Print” function. They are designed to let users print the contents of the `WEB-BROWSER` control (i.e., the Web page that is being displayed). In addition, there are several new properties relating to the Windows file and clipboard operations. These properties are defined in section 5.19.2 of Book 2. Additional usability information is provided in Appendix B of the *Programmer's Guide to the Internet*.

ActiveX License Key Enhancement

ACUCOBOL-GT now supports the use of WCHAR (also known as UNICODE) license strings for ActiveX controls. For more information, refer to the ActiveX LICENSE-KEY special property described in section 5.3.2 of Book 2.

Enhanced HP COBOL Compatibility

OPEN OUTPUT Creates MPE files

When a program is compiled with the “-Cp” switch and run on an HP e3000, all OPEN OUTPUT statements create MPE files. Such programs have the following MPE-specific behaviors and restrictions:

- Files referenced by OPEN OUTPUT statements must have MPE filenames.
- OPEN OUTPUT statements create MPE format files.
- MPE file locking is used.

Global Data in Recursively Called Programs

The runtime can now be configured to share the same data as the original instance of a recursively called program. (Normally ACUCOBOL-GT allocates a fresh copy of data to each instance of a recursively called program.) The primary reason for configuring the runtime to share data is if you are migrating code that relies on this behavior from another COBOL system, such as HP COBOL.

A new configuration variable, `RECURSION_DATA_GLOBAL`, facilitates the sharing of data. Set this variable to “1” (on, true, yes) to share data in recursively called programs. You must also set the `RECURSION` configuration option to “1” (on, true, yes) to allow recursion in the first place (recursion is not allowed under ANSI-85 COBOL rules). See Appendix H in Book 4 for additional information on both these variables.

Printing from the MPE/iX POSIX Shell

In Version 6.1, the method for printing from the MPE/iX POSIX shell using the “LP” program has been enhanced. This program is located in the “SYS/HPBIN/LP” directory. This directory should be located in your MPE/iX shell PATH, for example:

```
shell/iX> export PATH="$PATH:/SYS/HPBIN/LP"
```

The usage for the LP program is as follows:

```
LP [-d dest] [-n number] [-p priority] [-f] [-s] [-t title]
[files...]
```

where:

-d <i>dest</i>	Destination or output device (the default is “LP”)
-n <i>number</i>	Number of copies to print (the default is “1” on MPE)
-p <i>priority</i>	Print priority (the default is “8”)
-s	Suppress messages
-t <i>title</i>	Print title on the banner page of the output

We provide a sample configuration file (“cblconfi”) that contains sample settings for assigning the PRINTER device name to the MPE/iX system spooler. For example:

```
PRINTER -P /SYS/HPBIN/LP -s
```

This example runs the LP program on the print file. If your COBOL application writes to a file assigned to “PRINTER”, the first words after the “-P” are treated as the program to run.

New and Enhanced Library Routines

Note: All ACUCOBOL-GT runtime library routines are described in complete detail in Appendix I of Book 4.

Accessing the Windows Registry: DISPLAY_REG_*

A new set of library routines provides parallel functionality to the REG_* routines used to access the Windows registry. You should use the DISPLAY_REG_* routines when you want to work with the Windows registry on the display host. The display host is the thin client when the application is deployed with the ACUCOBOL-GT Thin Client, and is the local host when the application is run with a standard runtime. You should use the REG_* name when you want to work with the registry on the server host (the application host when the application is run with the thin client). For more information, see the section titled, “Routines to Handle the Windows Registry,” in Appendix I of Book 4.

C\$COPY: Special Remote Name Notation

The C\$COPY library routine has been enhanced to enable the transfer of files between the application host (server) and display host (client) when running with thin client. A special remote name notation, “@[DISPLAY]:”, is used to indicate that a file is located on the display host. For more information, see the AcuConnect section of this *Release Overview*.

C\$EXCEPINFO Returns Object Handle and Control ID

C\$EXCEPINFO has two new optional output parameters that hold the identity of the object or control that generated an exception. ERR-OBJECT-HANDLE gives the handle of the COM object or ActiveX control that generated the exception. ERR-CONTROL-ID gives the ID of the ActiveX control that generated the exception.

C\$SYSTEM

The C\$SYSTEM routine has a new option for the FLAGS parameter called CSYS-INHERIT-HANDLES (value 256). If you specify this flag, each inheritable handle in the calling process is inherited by the called process.

C\$XML

C\$XML gives you low-level control over the parsing of XML data. For more information, see the section titled “Enhanced Support for XML” in this *Release Overview*.

WIN\$PRINTER

The WINPRINT-SET-CURSOR operation can now return the current position of the write cursor in pixels, cells, inches or centimeters. In previous versions, the operation returned this information in pixel units only.

The WINPRINT-PRINT-BITMAP operation now allows you to set the coordinates of a bitmap in cells, pixels, inches, or centimeters. In previous versions, the coordinates could be specified cell units only.

The WINPRINT-PRINT-BITMAP, WINPRINT-SET-CURSOR, and WINPRINT-GRAPH-DRAW operations now test for a pending form feed before sending output to the printer.

General Compiler Enhancements

Web Runtime Security File Limit Raised

In Version 6.1, the limit on the “acuauth.txt” path has been raised to 4096 characters to match the limit on the length of the FILE_PREFIX variable. More information on the Web runtime security file can be found in section 5.10.4 in the *Programmer's Guide to the Internet*.

Support for MBP COBOL Sign Encoding

The compiler flag, “-Dcb”, turns on support for MBP COBOL's sign encoding conventions. For more information, see section 2.1.9 in Book 1.

DESTROY Statement Recognizes More Handle Types

The Format 1 DESTROY statement has been enhanced to recognize and operate on most handle types. In prior versions, a Format 1 DESTROY statement recognized a limited set of handle types (control, window, thread, font, menu, and layout manager). As in prior versions, if a handle-type is not recognized, the statement has no effect.

New Compiler Switches

There are several new compiler command line options. For complete information, see the subsections of section 2.1 in Book 1.

-C60	Causes the compiler to maintain source compatibility with the Version 6.0 compiler.
-Dcb	Turns on the MBP COBOL sign storage convention.
--ia-32	A synonym for "--intel".
--pa_risc, or --pa	Causes the compiler to produce 32-bit code for PA-RISC Version 1.0 processors running the HP-UX operating system.
--pa_risc2.0, or --pa2	Causes the compiler to produce 64-bit code for PA-RISC Version 2.0 processors running the HP-UX operating system.
--powerpc, or --ppc	Causes the compiler to produce 32-bit PowerPC native code for pSeries processors running AIX.
--powerpc_64, or --ppc64	Causes the compiler to produce 64-bit PowerPC native code for pSeries processors running AIX.
--noAlignLit	Inhibits use of the default algorithm for aligning literals in memory. See "-Da" in section 2.1.9 of Book 1.
-Zm	Causes the compiler to generate code that tells the runtime the size of a data item specified in a Format 7 SET statement.
-Z60	Creates object code that can be run with a Version 6.0 runtime.

ACUCOBOL-GT Runtime

Enhanced Handling of Files Specified with “-P <name>” Under Windows

In previous versions, under Windows, when you assigned a sequential file to “-P <name>”, the file indicated by “<name>” could only be “SPOOLER” or “SPOOLER-DIRECT”. Starting in Version 6.1, this function has been extended under Windows to support pseudo pipes as well as a connection to the Windows spooler. This means that “<name>” can be a sequence of space-separated words where the first word is the name of a file or “%TMP%” to specify a temporary file. The remaining words consist of a command to run. For input files, the command is run until it completes and then the named file is opened and processed normally. For output files, the named file is written and the command is run after the file is closed.

Also, when you use “-P <name>” or the SYSTEM library routine to execute an operating system command, you may append an ampersand (&) character to the command line to run the program asynchronously. This should not be done with programs that provide input files, but is often useful with programs that process output files.

For more information, see section 2.8.1 of Book 1.

New and Enhanced Configuration Variables

Note: All ACUCOBOL-GT runtime configuration variables are described in complete detail Appendix H of Book 4.

A_OPERATING_SYSTEM

This variable helps provide backward compatibility with pre-Version 5.0 programs that rely on the UNIX-V and UNIX-4 values in the OPERATING-SYSTEM field of the SYSTEM-INFORMATION data item.

AXML_CREATE_SCHEMA

This variable allows you to specify the type of XML output that ACUCOBOL-GT should generate when it creates XML files.

AXML_ENCODING

This variable allows you to specify a character encoding method for the XML files that ACUCOBOL-GT creates.

AXML_SCHEMA_NAMESPACE_DATA

This variable allows you to define the name of the schema file that ACUCOBOL-GT writes, if any, when it creates an XML file.

CALL_HASH_SIZE

This variable controls the size of the hash table used to track CALL statements to COBOL subprograms.

CONTROL_CREATION_EVENTS

This variable applies to those using ActiveX controls in their ACUCOBOL-GT programs. Set it if you want the runtime to allow events during the creation of an ActiveX control.

RECURSION_DATA_GLOBAL

This variable allows you to configure the runtime so that each instance of a recursively called program shares the same data as the original instance of the program.

REL_LOCK_READ_THROUGH

In Windows environments, if this variable is set by all participating processes, programs are allowed to “read through” relative file record locks. This means that READs that do not assert a lock are allowed to READ a relative file record even if it is locked by another process. This variable must be set by all processes accessing the relative file.

TC_CHECK_ALIVE_INTERVAL
TC_QUIT_MODE
TC_SERVER_TIMEOUT

These variables allow for greater control over the timing of thin client activity. For more information, see to the description in the AcuConnect section of this *Release Overview*.

TC_RESTRICT_AX_EVENTS

This variable allows you to control whether the application ignores all ActiveX events between the termination of one ACCEPT statement and the beginning of another. For more information, see the description in the AcuConnect section of this *Release Overview*.

WIN_SPOOLER_PORT

This variable supports the ability to send printer output directly to a file under Windows. It can be set in the configuration file, in the environment, or in the program. For example:

```
WIN_SPOOLER_PORT c:\mydir\myprint.prn
```

or

```
SET ENVIRONMENT "WIN_SPOOLER_PORT" TO  
"c:\mydir\myprint.prn".
```

When this variable is set, it affects all print jobs performed by the current instance of the runtime. The resulting files will be in binary encoding. Any graphics operations (such as winprint-print-bitmap) are preserved, but may cause the file to be very large.

AcuConnect

AcuConnect Version 6.1 contains the enhancements described in the following paragraphs. Many of these enhancements improve the AcuConnect client/server technology by expanding thin client functionality and optimizing system performance.

Unless otherwise indicated, section references are to the on-line *AcuConnect User's Guide*. Please refer to the ECN list for other AcuConnect corrections.

Detecting Lost Connections

In previous versions, when the ACUCOBOL-GT runtime lost its connection to a remote server for any reason, you needed to call the C\$DISCONNECT library routine to close the connection and then re-establish it. With Version 6.1, the runtime automatically checks the socket connection before sending a server request. If the socket connection is open, the runtime sends the request. If it is not, the runtime closes the original socket and attempts to open a new one to the desired server.

Calling DLLs

In a thin client environment, you can call subroutines in DLLs on the display host, or client, as well as on a Windows application host. Calling DLLs on the display host involves the use of a new “@[DISPLAY]:” notation and some other special considerations, which are described in section 6.6.3, “Calling Dynamic Link Libraries (DLLs).”

New acuthin Command Options

Two new **acuthin** command options let you send a username (“--user”) and a password (“--password”) in the thin client command line. These new options allow an AcuConnect distributed processing runtime to start a new thin client connection. See section 7.3.1, “The acuthin Command,” for more information. These options are also available with new “atc-user” and “atc-password” thin client command line file variables. Refer to section 6.7.1, “Thin Client Command Line Files.”

Character Mapping

Client/server character code mapping is now supported in thin client deployments. This enhancement lets you exchange data between client and server machines that use different character code sets. This feature is particularly helpful for international character handling. As with the AcuConnect distributed processing environment, you create a simple text file that maps the characters that are different on the client and on the server. For information about the contents and format of this map file, refer to section 2.5, “International Character Handling.” After you create your map file, you need to point to it using a new runtime configuration variable called `TC_MAP_FILE`. See section 6.3.1, “Configuring the Remote Application Runtime,” for more information.

Library Routines

New library routine support in thin client environments optimizes thin client performance and lets you take advantage of capabilities not previously available for the thin client.

The `C$OPENSVEBOX` library routine is now supported in the thin client. This routine, which allows you to browse the client machine’s drives and select a directory or file, is used in conjunction with other operations that have access to the client machine’s drives, such as `C$COPY`, `C$SYSTEM`, DLL calls, and COM or ActiveX components. See section 5.1.3, “When Other Application Work Is Necessary,” for additional information.

The `WIN$PLAYSOUND` library routine is now supported for thin client. Wave files (“`.wav`”) are downloaded from the server to a temporary cache directory on the client. This support is available even when the application is running on a UNIX server. Information about `WIN$PLAYSOUND` can be found in *ACUCOBOL-GT Appendices*, Appendix I.

Bitmap download performance is improved for thin client environments via the `W$BITMAP` library routine `W$BITMAP-LOAD` process. Setting the new `W$BITMAP-LOAD` parameter `FLAGS` to `W$BITMAP-NO-DOWNLOAD` optimizes bitmap handling. For details, refer to section 8.6.1, “Tuning System Performance” in the *AcuConnect User’s Guide*.

The C\$COPY library routine allows the transfer of files between the application host and the display host in thin client. A special notation of “@[DISPLAY]:” indicates that a particular file involved in a transfer is located on the display host, or client. Information about the use of C\$COPY in thin client environments can be found in section 6.6.1, “Copying Files Between the Client and Server.”

You can now access and modify the Windows registry on the thin client display host via a set of library routines. The names of the routines all begin with “DISPLAY” to distinguish them from similar routines that operate on the server’s registry if the server is Windows. For example, you use DISPLAY_REG_CREATE_KEY or DISPLAY_REG_CREATE_KEY_EX to create a new registry key on the display host. Refer to section 5.1.3 in the *AcuConnect User’s Guide* or to *ACUCOBOL-GT Appendices*, Appendix I, in the section titled “Routines to Handle the Windows Registry,” for more information.

Runtime Configuration Variables

New configuration variables contribute to more efficient thin client operation.

Some ActiveX controls do not support the ability to suspend and resume events when an application is not processing an ACCEPT statement. With a new TC_RESTRICT_AX_EVENTS thin client configuration variable, you can control whether your application ignores all ActiveX events between the termination of one ACCEPT statement and the beginning of another. For more information, see section 6.3.1, “Configuring the Remote Application Runtime.”

Three new configuration variables for thin client help you control how your COBOL application shuts down if either the client or server machine is unresponsive for a user-defined period of time. TC_QUIT_MODE, TC_CHECK_ALIVE_INTERVAL, and TC_SERVER_TIMEOUT provide various options for determining when and how your program terminates in these instances. Information can be found in section 6.3.1, “Configuring the Remote Application Runtime.”

Miscellaneous Enhancements

In Version 6.1, the ACUCOBOL-GT Thin Client can display its own splash screen on startup. You can also use a custom splash screen of your own design. For details about the thin client splash screen, refer to section 7.2.4, “Thin Client Splash Screen.”

Paged grid control performance is enhanced for thin client environments to reduce client/server network exchanges. The generation of MSG-PAGED-NEXT or MSG-PAGED-PREV events for a page of data occurs all on the server side, reducing paged grid initial load and paging times.

You can now access help from the **acurcl** command graphical control panel by pressing the F1 key, as well as by clicking the Help push button.

AcuODBC and AcuODBC Server

At Version 6.1, AcuODBC includes several useful enhancements.

Outer Joins

Some SQL query generators, such as Crystal Reports®, generated outer joins that were not supported by previous versions of AcuODBC. In Version 6.1, AcuODBC has been enhanced to support more complex outer joins, such as those merging three separate tables.

Date, Time, and Timestamp Values

AcuODBC has been enhanced to recognize nearly any valid date, time, and timestamp value. The COBOL programmer must create types that are meaningful for the application that will use them. For example, some applications, including Microsoft Access 2000, do not recognize timestamp variables or time variables that include fractions. If you are using an application with this limitation, modify any time fields containing fractions to be purely numeric.

Duplicate Keys

In prior versions, if two keys differed only in whether they allowed duplicates, AcuODBC considered them distinct, even though the runtime considered them the same. With Version 6.1, this has changed. Now AcuODBC considers these keys the same.

Press <F1> for Help

The AcuODBC Control Panel has a help system, which is accessed using the **Help** button. Now you can also access the help system by pressing the **F1** key.

AcuServer

Enhanced Reporting

When you specify “acuserve -info” with the “-f” option, the resulting report now includes the full machine name and user name, up to 170 characters.

For Windows users, the full names now appear by default in the AcuServer Control Panel.

Press <F1> for Help

The AcuServer Control Panel has a help system, which is accessed using the **Help** button. Now you can also access the help system by pressing the **F1** key.

AcuSQL

Enhanced performance and greater flexibility in the handling of transactions mark Version 6.1 of AcuSQL.

Enhanced Performance

Performance of AcuSQL run against SQL Server has been enhanced. A new configuration variable, `ASQL_BUFFER_SIZE`, can be set to any positive number to cause AcuSQL to cache that many rows from a cursor when the cursor is read. This can result in much less traffic across the network, and therefore, better performance. See section 4.2.1, “Runtime Configuration Variables,” in the *AcuSQL User's Guide* for more information on `ASQL_BUFFER_SIZE`.

Transactions in SQL Server

In Version 6.1, it is now possible to mix explicit transactions and embedded transactions. You can now execute a `BEGIN TRANSACTION` and a `COMMIT WORK` together, and the transaction is committed.

It is now possible to nest transactions, as supported by SQL Server. When you use nested transactions, note how many transactions you `BEGIN`, because you must `COMMIT` the same number of transactions for the transaction to be considered complete. For example, if you execute three `BEGIN TRANSACTION` statements, you must later execute three `COMMIT TRANSACTION` or `COMMIT WORK` statements to commit the work you have done.

Note, however, that a single `ROLLBACK` rolls back all nested transactions. This behavior is a feature of SQL Server, and is not controlled by the AcuSQL interface.

When a transaction is closed, the AcuSQL interface completes or abandons any active transactions (depending on the value of the `ASQL_DISCONNECT_ACTION` configuration variable). See section 4.2.1, “Runtime Configuration Variables,” in the *AcuSQL User's Guide* for more information on `ASQL_DISCONNECT_ACTION`.

Binding Group Items

The AcuSQL precompiler can now bind multi-level group items. In previous versions, it was able to bind the columns of a group item that had only elementary items.

AcuBench

Version 6.1 of AcuBench includes some minor corrections. Please refer to the ECN list for details.

Acu4GL

Version 6.1 of Acu4GL includes some minor corrections. Please refer to the ECN list for details.

Technical Support

If we can help you in any way, please contact Acucorp Technical Support. We can be reached at the following numbers:

Acucorp Technical Support in the United States:

Phone: +1 (858) 689-4500
1-800-262-6585 (within the U.S.)

Fax: +1 (858) 689-4552

E-mail support@acucorp.com

Customers outside of the United States, please contact your regional Technical Support office or your local distributor:

In Benelux:

Phone: +31 (0)73 623 01 95
Fax: +31 (0)73 623 01 99
E-mail nlsupport@acucorp.com

In France:

Phone: +33 (0)1 53 34 9005
Fax: +33 (0)1 53 34 9001
E-mail: frsupport@acucorp.com

In Germany:

Phone: +49 89 455 659 55
Fax: +49 89 455 659 50
E-mail: desupport@acucorp.com

In Scandinavia and the United Kingdom:

Phone: +44 (0)20 8843 7101
Fax: +44 (0)20 8744 9401
E-mail: uksupport@acucorp.com

