

Liant v12.12 Supplement

Copyright © 2016 Micro Focus. All rights reserved.

This document describes, for each component of the Liant product set, the changes made in the 12.12 version of the product. In most cases, the changes were the result of a reported product incident (RPI) and in those cases the RPI number is provided for reference.

RM/COBOL 64-bit Windows Runtime System

Beginning with this release, a native 64-bit Windows version of the RM/COBOL runtime system is available for 64-bit Windows deployments.

The 64-bit runtime system is installed separately from the 32-bit runtime, and will co-exist with the 32-bit runtime and 32-bit development system. All Liant-supplied runtime programs and extensions, like RECOVER1, RMCONFIG, and the XML Extensions, are also available as 64-bit programs but the development tools (RM/COBOL compiler and CodeWatch IDE) are still 32-bit programs, although CodeWatch can now also run 64-bit COBOL programs (see *CodeWatch for Windows*, below). Xcentrinity® Business Information Server is currently only available as a 32-bit program.

No special compilation options are required to build 64-bit COBOL programs – compiled .COB files are fully compatible with either 32-bit or 64-bit **runcobol**. However, non-COBOL programs not supplied by Micro Focus (specifically, DLLs) must be rebuilt as 64-bit programs to be loaded by the 64-bit runtime.

Installation Locations

On 64-bit Windows, the 64-bit runtime is installed in “**C:\Program Files\Liant**”, while the 32-bit runtime is installed into “**C:\Program Files (x86)\Liant**”. Shared 64-bit components are installed into “**C:\Program Files\Common Files\Liant Shared**”, while 32-bit shared components are installed into “**C:\Program Files (x86)\Common Files\Liant Shared**”.

If these programs are installed into custom locations, it’s critical that 32-bit components and 64-bit components are separated and not installed in the same directory.

Windows Registry and Properties

The *Windows Registry* section of chapter 3 of the *RM/COBOL User’s Guide, Second Edition* describes *runcobol* properties that may be set by the RMCONFIG utility.

- Properties set for “**This User**” affect both the 32-bit and 64-bit runtime. Either version of RMCONFIG may be used to set these options.
- Properties set for “**All Users**” are different. The Win32 version of RMCONFIG sets properties that only affect Win32 programs. The Win64 version of RMCONFIG sets properties that only affect the Win64 version of the runtime system. This is only a concern if both Win32 and Win64 runtime systems are installed.

When configuring for “**All Users**”, it’s best to use the 32-bit RMCONFIG utility to configure 32-bit runtime systems, and the development system; use the 64-bit RMCONFIG utility only to configure the 64-bit runtime.

Either RMCONFIG utility may be used to configure Windows for “**This User**”.

Note that it is possible to configure programs by selecting “Properties” from the system menu in the runtime window. The above rules apply to that as well. The executing runtime can always configure itself, but only changes made for “This User” will affect the runtime with the opposite “bitness”.

File Associations

Windows can run RM/COBOL programs from the shell – for example, by processing double-clicks on a COBOL program (.COB file) in a Windows Explorer window. Because the Windows shell can only associate file extensions with one program at a time, the last-installed runtime system will be invoked when a .COB file is double-clicked. Please note that we do not recommend applications be launched in this way except on dedicated machines like point-of-sale terminals, as an application from another vendor can take over ownership of .COB files at any time.

Liant Runtime Registration

The Liant runtime system is registered during the installation. However, some customer deployment scenarios do not use the installer and in those cases, manual registration is required. This section only applies to these deployments.

Registration can now be performed only for the current user instead of per-machine basis. This type of registration does not require the user have administrative privileges. “**RUNCOBOL /REGSERVER**” silently registers the runtime only for the current user if per-machine registration fails due to a lack of permissions. However, current user registration could not be forced if the current user is an administrator. This can now be explicitly controlled.

To register the runtime only for the current user, use this Windows system command:

```
regsvr32 /n /i:user path\RMCL12R.DLL
```

Specify **/u /n** to unregister for the current user (this also does not require administrative privileges).

Omit **/n** to register both for the machine and the user. This does require the user have administrative privileges.

Omit **/n /i:user** to register for the machine, and if not possible, then just for the current user. This is also what **RUNCOBOL /REGSERVER** does.

Note that, if both the 32-bit and 64-bit runtime systems are installed, both must be registered independently.

Relativity Data Manager

RPI **582134** — The installation guide has been updated to describe how to run the 32-bit ODBC Administrator on a 64-bit Windows operating system. The 32-bit ODBC Administrator is located at %windir%\SysWOW64\odbcad32.exe.

Relativity Designer

RPI **610205** — A problem in the VB designer has been corrected. In the RST import step, the "Select 01's" dialog allows a subset of the FD and 01-level items to be chosen. These settings are remembered for each item; however, an attempt to re-examine the values caused a runtime error. This has been corrected and the values may now be adjusted more than once, if so desired.

RM/COBOL Compiler

RPI **615505** — The RM/COBOL compiler will now display the full pathname of the file in the 'Open error for file "xxx".' message (xxx is replaced with the full pathname of the file that could not be opened).

RPI **1099924** — The RM/COBOL compiler ignores reference modification when setting the length field of a pointer data item from a referenced modified variable in a SET statement. This is by design. The length field was intended as a way to prevent overrunning the memory that a pointer refers to and is not a general solution to managing lengths.

RM/COBOL Runtime

RPI **609192** — A possible crash while accessing RM/COBOL properties has been fixed. A Windows registry key was sometimes being closed twice.

RPI **617399** — The compiler (rmlibcmp.dll), runtime (rmlibrun.dll), and BIS service engine (xbisse.dll) DLLs in previous releases were excluded from being digitally signed. This has been fixed and all released executables on Windows are digitally signed with a Micro Focus certificate.

RPI **1099943** — A problem with print dialog settings not being maintained when accessing a dynamic printer has been fixed. After being initially set, printer properties such as orientation, number of copies, and other such properties are now doubled-checked and re-set to the desired values if necessary.

RPI **1103326** — The **C\$OSLockInfo** subprogram has been modified to return a value of zero if the process id value will not fit in the number of digits defined by the COBOL data item. In Appendix F of the RM/COBOL User's Guide in the section for **C\$OSLockInfo** the *processid* argument is described as "a four-byte, unsigned COMP-4 numeric item". We recommend that *processid* be defined as **PIC 9 (10) BINARY (4)**. That will be a four-byte (32-bit) numeric item of ten decimal digits. We further recommend that the COBOL program test for a returned value of zero to indicate that a valid process id could not be returned, probably because the COBOL data item has too few digits to contain the value. Note that prior versions of RM/COBOL would store only the low-order digits of the process id if it did not fit in the item, for example, an item described as **PIC 9 (4)** would contain the value 2345 when the actual process id was 12345. The COBOL program would then be referring to the wrong process, a bad situation. Beginning with version 12.12 the value zero will be stored in the case where the item has too few digits to contain the entire process id.

RMSETNCS Utility on Windows

The Windows version of RM/COBOL has a utility named `rmsetncs.exe` for setting the native character set of the compiler and runtime to ANSI or OEM, setting whether a GUI or Console compiler should be used by default and the CodeWatch INI file in the Windows directory. With the advent of User Access Control (UAC) in Windows Vista and later, it should be noted that this utility modifies files in the protected Windows directory and in the installation directory, which is normally a protected directory such as “Program Files” or “Program Files (x86)”. Thus, the utility must be run as administrator with highest privileges when UAC is on (the default state). The utility modifies the `runcobol.exe` file in the same directory as the utility itself. The 64-bit and 32-bit runtimes can be modified separately and need not have the same default character set, but if both exist on a system they should have the same default character set for consistency.

InstantSQL for RM/COBOL

RPI **1101636** — A problem with an incorrect error code being reported when inserting a duplicate key using 64-bit InstantSQL on Linux has been fixed. In InstantSQL, SQL DESCRIBE ERROR has been changed to return the correct error codes on 64-bit machines. When a valid ODBC error used InstantSQL, rather than returning the correct error, SQL DESCRIBE ERROR would instead return 2301, which is an internal error.

WOW Extensions for RM/COBOL

RPI **1098617** — A problem with the Properties Edit Dialog in the WOW Designer has been fixed. Newer Microsoft compilers defaulted to Vista style file dialog boxes which prevented custom templates from working correctly. The use of custom templates has now been corrected.

Xcentricity® Business Information Server (BIS)

IPv6 is now supported in BIS for UNIX.

RPI **615660** — In BIS for UNIX, the tag `{{Value(HTTP_URL, GETQUERYSTRING)}}` now works the same as it does on Windows. The HTTP_URL VALUE now returns the query string and GETQUERYSTRING now returns the leading ?.

XML Extensions for RM/COBOL

XML Extensions has been fixed so that a second reference to an undefined stylesheet or schema file name will not crash. Previously, the first failed load of the stylesheet or schema document was cached even though the load was unsuccessful. A subsequent reference to the same file name tried to use the NULL document pointer stored in the cache and a crash would happen. A failed load now removes the failed document file name from the cache.

XML Extensions has been improved to include more information with errors when the XML GET STATUS-TEXT is used after the error happens. Information about the element name and content, model data name, document name and so forth is provided in the status text provided on separate lines. The operation in which the error happened, such as export, import or transform is also provided. This additional information will likely reduce the need to turn on tracing (with the XML TRACE statement) to determine where a problem happened in a program that uses XML Extensions.

XML Extensions has a new runtime error code for when the trace file name specified in the XML TRACE statement cannot be opened. This resolves an issue where XML Extensions would crash if the trace file could not be opened.

73 Error - trace file open failed

The trace file name specified in an XML TRACE statement could not be opened. The most common cause of this error is that a directory name that does not exist is specified as part of the file name. Other errors that could cause this are access denied (because the file is read-only or the user does not have permissions to the file) or the program does not have permission to write to the directory specified. The program can continue, but tracing will be disabled. This error is only returned when the XML TRACE statement opens the trace file for the first time. Subsequent trace file open failures while the program is running other XML Extensions statements cause tracing to be suppressed.

CodeWatch for Windows

CodeWatch has been enhanced to support the debugging of an RM/COBOL program with the 64-bit runtime system on Windows.

- There is now a “64-Bit” checkbox next to the “Program to Run” field in the Workspace Wizard and the Workspace Properties “Program” page:

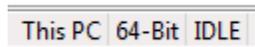


This checkbox is only available if both the 32-bit and 64-bit RM/COBOL runtime system are installed. The selection may be changed at any time and takes effect the next time the runtime system is launched.

If only one system is installed, the 64-Bit checkbox is disabled but is checked if only the 64-bit runtime is present, and not checked if only the 32-bit runtime is present.

If both runtime systems are available, you may choose which will be used for debugging by checking or unchecking the box. The most recent selection is saved in the workspace file. If that workspace file is moved to another system with only one runtime, the saved option is ignored and the installed runtime system will be used.

- There is an indicator on the status bar that indicates the selected runtime system: **32-Bit** or **64-Bit**:



(Note that “My Computer” now says “This PC” for non-BIS debugging.)

Additional Notes:

- If a 64-bit runtime is selected, it is run as a separate process by CodeWatch – specifically, in the DLLHOST.EXE surrogate host process. Note that the 32-bit runtime is executed in-process as before.

Because of the different execution environment, there may be some minor differences in the way that 64-bit COBOL programs function under CodeWatch. One example: runtime-created Windows can appear behind the CodeWatch window. In some cases, it's possible that the program may have displayed a message box or may be waiting for an ACCEPT and that will cause CodeWatch to appear suspended. If, while debugging, you don't see the runtime window, please move the CodeWatch window or use Alt-Tab or the task bar to bring the runtime window to the front. Note that running CodeWatch on a second monitor avoids this situation completely and is highly recommended.

- When running on Windows 8 and later, 64-bit programs being debugged will be loaded into memory using large base addresses (that is, the program and data will reside above the 4GB line and have a base address that cannot be stored in 32-bits). If your program calls 3rd-party non-COBOL 64-bit programs, these will also be loaded at large base addresses.

Because some 64-bit non-COBOL programs are converted 32-bit programs and may still store pointers in 32-bit integers, such 64-bit programs will not tolerate being loaded into the upper reaches of memory. This may cause them to fail when running under CodeWatch, but may run normally or fail intermittently when not running under CodeWatch. To debug such a 3rd-party program, start CodeWatch, run to the point where the 3rd-party program is loaded, and then attach Visual Studio to the DLLHOST.EXE process (not the CodeWatch RMCW.EXE process) and set breakpoints in the code.

Liant v12.11 Supplement

Copyright © 2015 Micro Focus. All rights reserved.

This document describes, for each component of the Liant product set, the changes made in the 12.11 version of the product. In most cases, the changes were the result of a reported product incident (RPI) and in those cases the RPI number is provided for reference.

Liant Installation

On Windows, the Start Menu items installed by Micro Focus Liant products are no longer installed under a single top level Start Menu item named "Liant". There are now up to six top level Start Menu items starting with the word "Liant" and incorporating the product name from the second level menu from previous releases:

"Liant BIS v12"

"Liant Relativity v12"

"Liant RMCOBOL v12"

"Liant RMInfoExpress v12"

"Liant WOW Designer v12"

"Liant WOW Thin Client v12".

Relativity Data Server

RPI **1097632** – A problem was corrected with starting the Data Server service when there is a "C:\Program" file on the machine. The Data Server now tolerates the existence of the "C:\Program" file when it is started. (Note that Windows recommends against having the "C:\Program" file on a machine and prompts the user to rename it to "C:\Program1" when Windows is started.)

Relativity Data Client

UNIX Driver Manager installation removed

The binaries and installation of the unixODBC and iODBC Driver Managers is no longer available on the Relativity installation media. Relativity now requires a pre-installed Driver Manager, either unixODBC or iODBC. The Data Client installation will ask for the directory name where the Driver Manager has been installed.

RM/COBOL Compiler

RPI **1099805** — A compiler issue with numeric temporary data items created by the compiler in nested programs was corrected. The error could cause an incorrect data overflow termination of the compiler and might have caused incorrect re-use of numeric temporaries in the nested program.

RM/COBOL Runtime

RPI **1099261** — A file manager issue causing long running programs to occasionally report 98,27 errors on indexed files was corrected.

A new field has been added to the RMInformation group returned by the **C\$GetRMInfo** routine. The new field is named RM-Bitness and is defined in the **rminfo.cpy** file supplied with RM/COBOL. The value is either 32 or 64 and specifies the bitness of the runcobol executable running the COBOL program. The RMINFO utility program supplied with RM/COBOL has been updated to report the RM-Bitness field. The following is an excerpt from rminfo.cpy showing the new RM-Bitness definition:

```

02  RM-BISValue                Picture X.
    88  RM-BISIsPresent        Value 'Y'
                                     When False 'N'.
02  RM-ThinClientValue        Picture X.
    88  RM-ThinClientIsPresent Value 'Y'
                                     When False 'N'.
02  RM-Bitness                 Picture 999 Binary(1).
    88  RM-COBOL-32-BIT        Value 32.
    88  RM-COBOL-64-BIT        Value 64.
02  RM-Reserved                Picture X(18).
02  RM-LicenseProduct          Picture X(80).
02  RM-LicenseIssuedBy         Picture X(80).
02  RM-LicenseIssuedTo         Picture X(80).
02  RM-LicenseType             Picture X(20).
02  RM-LicenseValidThru        Picture X(20).

```

RMNet

RMNet v12.11 uses libcurl version 7.42.1 and OpenSSL version 1.0.1m. RMNet v12.11 includes a new function, **HttpSetSSLVersion**, which sets CURLOPT_SSLVERSION that specifies the TLS or SSL security protocol version. See **rmnet.pdf** for complete details.

RPI **1098827** – RMNet will no longer report "SSL Connect Error" on some websites on SCO Open Server 6. The issue was that in previous versions not all allowed ciphers were supported on SCO Open Server 6, librmnet.so now supports the same set of ciphers as other Unix platforms and Windows.

RPI **1099015** – RMNet includes a new function, **HttpSetSSLVersion**, which allows the SSL version to be set, for example, to SSLv3. Some older servers do not correctly handle TLS negotiation and cause a

failed connection. Calling `HttpSetSSLVersion` to set SSLv3 before calling `HttpPost` will cause the connection to use SSLv3 rather than any TLS version and will allow the connection to succeed.

Xcentricity® Business Information Server (BIS)

Previous versions of BIS required that, for IIS 7 and later, "IIS 6 Metabase Compatibility" be enabled when IIS is installed. This is because the installer and the **BISMkDir** program used the *Active Directory Services Interface (ADSI)* to configure IIS and create virtual directories.

Starting with v12.11, BIS includes a new program called **BISMkApp**, which natively creates IIS applications without using ADSI. This means that "IIS 6 Metabase Compatibility" is no longer required and corrects other problems that were caused by the IIS 6 metabase translation layer, such as the IIS configuration files becoming larger with each installation or configuration.

Both **BISMkApp** and **BISMkDir** are installed with BIS, provide similar functionality, and either may be used if IIS 6 metabase compatibility is installed on IIS 7 and later. However, **BISMkDir** is deprecated and will be removed in a future release.

Note that IIS 7 was introduced with Windows Vista and Windows Server 2008.

RPI 601316 – BIS for UNIX now works properly when it is installed in a version of Apache that uses a multithreaded MPM. The Apache "child" processes run a Multi-Processing Module (MPM) which is normally selected when the copy of Apache is built. In Apache 2.0 and 2.2, the default MPM was always **prefork**, which processes a single request at a time in each MPM process. In Apache 2.4, the default MPM is either **worker** or **event**, depending on the capability of the operating system. Both of these use threads to increase performance. Prior versions of BIS required the **prefork** MPM. This requirement is now eliminated.

RPI 609456 – In BIS for UNIX, a change has been made in how BIS processes "sequenced" requests -- that is, requests for pages that use "`{{FormActionTarget}}`" as a target, along with the automatically-generated "`_xmlexch=abcd`" query parameter in the URL.

Starting with version 12.11, if a request that includes a non-empty `_xmlexch` query parameter is issued for a page that results in the creation of a new session (either because the current session has expired, or no session ever existed), then service is denied with an error code of 500 ("Internal Server Error").

This change addresses a problem encountered by users of BIS Express: there may be workstations that send refresh requests in the background every minute to keep sessions open. However, if the server is reset, those refresh requests (which include the last known sequence) will create new sessions and start programs that are not in a state where processing can resume, wasting server resources. Those sessions then persist until they time out, but some unattended clients create such new sessions every minute, potentially swamping the server. With this change, such sessions are not created and the refresh requests for defunct sessions from unattended machines are effectively ignored. This change modifies behavior in an error case should not impact normal use of XBIS.

There is not a configuration option to cause BIS to revert to the previous behavior.

XML Extensions for RM/COBOL

RPI **611430** – XML Extensions run time errors 68 to 71 were added in v12.06 as part of eliminating temporary files during export operations, but were not documented. Here is the documentation for those errors:

68 Error in function: AddAttribute.

An attribute node could not be added during export of a document. Attributes are only added by XML Extensions when the XML ENABLE ATTRIBUTES statement has been executed. This error will not normally happen. One cause could be that the XML engine did not have enough memory to add the attribute node to the document.

69 Error in function: AddElement.

An element node could not be added during export of a document. This error will not normally happen. One cause could be that the XML engine did not have enough memory to add the element node to the document.

70 Error in function: AddText.

A text node could not be added during export of a document. This error will not normally happen. One cause could be that the XML engine did not have enough memory to add the text node to the document. The error can also happen if incorrect parameters are passed to the XML EXPORT FILE or XML EXPORT TEXT statement.

71 Error in function: ValidateDOMDocument.

An attempt to validate a document failed. This normally means that the document is not valid, but could also mean the supplied schema is not a valid schema or there was not enough memory to validate the document. Additional error lines are produced to indicate why the validation failed.

Liant v12.10 Supplement

Copyright © 2015 Micro Focus. All rights reserved.

This document describes, for each component of the Liant product set, the changes made in the 12.10 version of the product. In most cases, the changes were the result of a reported product incident (RPI) and in those cases the RPI number is provided for reference.

CodeWatch for RM/COBOL

RPI 602896 – A problem that displayed an error message box when attempting to save a workspace file has been fixed.

RPI 603544 – If CodeWatch was launched with a relative path to the Workspace (.CWF) file from the working directory, that path was being interpreted relative to the directory that contained the workspace file instead of the working directory. This prevented CodeWatch from starting properly and the problem has been corrected.

RPI 603951 – In previous versions, if compilation failed with an initialization error (for example, if an invalid listing or object directory is specified), CodeWatch reported a "[-1] Internal Error". Now, a "250 Initialization Error" is reported, and a descriptive message like "Cannot open object file" appears in the output window.

RPI 604388 – A problem in CodeWatch was resolved: if you directly edit the command line in the workspace properties, the "base directory" was reset to execution directory, and if that directory was relative to the base directory, it was incorrectly being expanded. Starting with this version, editing the command line does not affect the execution or base directory.

RPI 607264 – A problem with directory browsing and selection on Windows Vista and later has been fixed. In 12.09, a file selection instead of a directory selection dialog would sometimes appear, although you could always manually type in the name of the folder.

RPI 609288 – If the CodeWatch main window or the workspace was closed while a debugging session was active, an error was reported and the COBOL program was not being allowed to terminate. This has been corrected and CodeWatch will now shut down properly.

RPI 609679 – A problem with CodeWatch hanging when terminated more than once with a persistent runtime window open has been fixed.

RPI 1094427 – A problem that corrupted the "current working directory" in CodeWatch workspace files when the workspace properties were examined or changed has been fixed. This also includes fixes to the dropdown in the property sheet. If the "Working Directory" field is left blank, the directory that contains the workspace .CWF file will be used as the working directory. A path relative to the .CWF file may also be entered, or an absolute path may be entered; in the latter case, CodeWatch will attempt to

store the path relative to the .CWF file, if possible, to make the file as portable as possible.

Recover1

New Command Line Option P On Windows

The Windows recover1 utility has a new command line option, P, which may be used to specify the Persistent property. The Persistent property controls whether the recover1 window persists after the utility terminates (see page 84 in Chapter 3 of the RM/COBOL User's Guide Second Edition for a full description of the Persistent property). Unlike the new runtime P option which allows a value of Y or N the recover1 P option allows no value; it is used only to turn Persistent on. The recover1 Q option may be specified to suppress all output. It is usually very important to view the results displayed on the final recover1 screen. Specifying the P option on the recover1 command line may be more convenient than always setting Persistent to True in the Windows Registry via the Properties dialog box.

Relativity Data Client

RPI **1093632** – The SQLGetFunctions API call with a FunctionId of SQL_API_SQLFOREIGNKEYS, now returns the proper value of FALSE. Previously, SQL_API_SQLFOREIGNKEYS returned TRUE and SQLForeignKeys always returned Driver not capable. However, this behavior confused some ODBC applications. It is better to say that Relativity doesn't support SQLForeignKeys, which it doesn't.

UNIX Driver Manager sources removed

The source for the unixODBC and iODBC Driver Managers is no longer available on the Relativity installation media. The modified source for the unixODBC Driver Manager is available on the Internet in the location documented in the file License_Notice_File.txt on your installation media. The modified source of the iODBC Driver Manager is no longer available from Micro Focus.

Relativity Data Manager

RPI **609158** – There is no longer a problem retrieving a number into a SQL_C_NUMERIC when the SELECT contains an ORDER BY or GROUP BY.

RPI **609159** – The SQLColAttribute function in Relativity now returns correct information for each attribute; it no longer returns errors for some attributes.

RPI **1095556** – The Relativity Data Manager installation would terminate with error 1603 in load data source if a system had both user and system Relativity data sources created on it from a previous installation of Relativity. The Relativity Data Manager installation was changed to tolerate this situation.

RPI **1098139** – Changed the Relativity Data Source setup utility to be able to use Fileshare catalogs again. Also fixed using RM/InfoExpress catalogs.

Relativity Data Server

RPI **601967** – The names of the exported symbols in the partially-linked Relativity Data Server for AIX have been corrected.

RPI **107772** – Changed the Relativity Data Source setup utility to be able to use RM/InfoExpress catalogs again.

RPI **1096117** – The problem importing file definitions from a .COB file in the Relativity Designer has been fixed in the Relativity Data Server.

Relativity Designer

RPI **1075713** – The initial dialog no longer shows the Liant logo and a link to www.liant.com.

RM/COBOL Compiler

RPI **608784** — Fixed a compiler issue with an incorrect compilation error 0171 for an ORGANIZATION clause that specified LINE or BINARY following a RECORD DELIMITER clause that specified a corresponding LINE-SEQUENTIAL or BINARY-SEQUENTIAL record delimiting technique. The order of the clauses should not have caused the error when the two clauses agree. The error was not generated if the ORGANIZATION clause preceded the RECORD DELIMITER clause. The RECORD DELIMITER clause produces an error and is ignored when it disagrees with the ORGANIZATION clause.

RPI **1096042** — RM/COBOL Windows EXE files no longer have the "Terminal Server Aware" flag set in their image header. This should allow them to work better in a Terminal Services environment. This change has no effect on non-terminal-services environments.

RM/COBOL Runtime

RPI **1098295** – Fixed a 98,27 error that occurs when a file has been modified a very large number of times.

New command line option P on Windows

The Windows runtime has a new command line option, P, which may be used to specify the Persistent property. The Persistent property controls whether the runtime window persists after the COBOL program terminates (see page 84 in Chapter 3 of the RM/COBOL User's Guide Second Edition for a full description of the Persistent property). The P option may be specified as P=Y to turn Persistent on or as P=N to turn Persistent off. Specifying just the option letter P without a value is equivalent to specifying P=Y. Specifying the P option on the runtime command line is equivalent to calling C\$GUICFG to set the Persistent property at the beginning of your COBOL program (see page 589 in Appendix F of the RM/COBOL User's Guide Second Edition for information about the C\$GUICFG subprogram). Thus, the P command line option value overrides the Persistent property value stored in the registry but only for the current invocation of the runtime. If your COBOL program calls C\$GUICFG to set the Persistent property

then that value will be used when the program terminates. For the Unix runtime the P option continues to be invalid (see page 202 in Chapter 7 of the RM/COBOL User's Guide Second Edition for a table of runtime command line options).

RMNet

RMNet v12.10 for Windows requires normaliz.dll

RMNet v12.10 uses Libcurl version 7.38.0 and OpenSSL version 1.0.1i (see RPI 1091213 below). Libcurl 7.38.0 for Windows requires DLL **normaliz.dll**. This DLL is part of Windows but may not be installed with a fresh install of Windows XP. If **runcobol** fails to load **rmnet.dll** then check your Windows system32 directory (normally C:\Windows\System32) to see whether **normaliz.dll** is missing. If it is missing then run Windows Update and ensure that you have SP3 (service pack 3) and all of its updates. Another way to install **normaliz.dll** is to upgrade Internet Explorer to version 8 or higher.

RPI **1091213** – RMNet will no longer report "SSL Connect Error" on some websites. This was an issue in the OpenSSL used by RMNet. RMNet for RM/COBOL v12.10 now uses OpenSSL version 1.0.1i. It also uses Libcurl version 7.38.0.

RPI **1097466** – Three new functions have been added to RMNet in version 12.10: `HttpSetUsername`, `HttpSetPassword`, and `HttpSetLoginOptions`. These are used to set `CURLOPT_USERNAME`, `CURLOPT_PASSWORD`, and `CURLOPT_LOGIN_OPTIONS` respectively. See `rmnet.pdf` supplied with RM/COBOL and online libcurl documentation for further details.

RM/InfoExpress

RPI **605280** – In prior releases the Unix Infox server read the `rmixsrvr.ini` configuration file only in the current directory. The Infox server, `rmservertcp`, and the `rmixsrvr.ini` file are normally installed together in the `/usr/rminfox` directory. If the Infox server was then started from another directory by entering the command `/usr/rminfox/rmservertcp` then the user would be unaware that the installed `rmixsrvr.ini` was not read and thus the server was running with its default configuration. Beginning in this release, when the Infox server is started it first attempts to read `rmixsrvr.ini` in the current directory. If `rmixsrvr.ini` does not exist in the current directory then it reads `rmixsrvr.ini` in the same directory from which `rmservertcp` was executed, normally the installation directory. This allows easier automatic Infox server startup at boot time and should reduce confusion if the Infox server is started manually.

Xcentricity® Business Information Server (BIS)

RPI **596213** – Starting with this release, invalid or malformed XML requests no longer pass to the service engine. Instead, an informative SOAP fault generates. Note that such rejected requests are not traced, so invalid SOAP cannot be used to deny service (the rejection of invalid XML is very fast).

RPI **601298** – BIS installation on Unix now displays an invalid input message and waits for the user to try again when asking most questions. It also always sets Apache's envvars before running the httpd executable to determine the Apache version and prefork status and then displays better messages when skipping Apaches that cannot be used with BIS.

RPI **601315** – BIS installation now correctly installs its startup script in /sbin/init.d and /sbin/rc*.d on HP-UX systems.

RPI **601422** – A problem that could cause XML responses that contain multi-byte character sequences has been fixed. Beginning with this version, XML content is converted from the host machine's OEM or ANSI character set into UTF-8 (which can represent all characters) and is preserved until the payload is delivered to the user agent. Previously, some multi-byte character sequences could have been altered during output processing.

RPI **601437** – The BIS User's Guide now shows the correct command to stop BIS on AIX:

```
stopsrc -g xbis
```

RPI **601468** – Starting with version 12.10, the command line specified in the {{StartService}} tag may be prefaced with either /CS_ANSI or /CS_OEM, separated from the main program name by either spaces or a comma. For more information on these options, see "RM/COBOL for ANSI Codepage on Windows" in the RM/COBOL User's Guide. If neither option is specified, the default is /CS_OEM, which results in unchanged behavior.

RPI **607664** – A problem with the HTTP result code returned with SOAP faults has been addressed. Previously, SOAP faults were returned with a result code of 200 (OK). The standard, however, requires that SOAP faults set the HTTP result code to 500 ("Server Error").

Starting with version 12.10, SOAP faults that are generated by BIS will set the result code to 500. If this new behavior is undesirable, set the following registry entry like this:

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Liant Software Corporation\BIS\Options]
"SoapFaultResultCode"=dword:000000C8
```

(C8 is 200 in hexadecimal). This should not be required, but is available as an option.

RPI **1095576** – In BIS for IIS, a change has been made in how BIS processes "sequenced" requests -- that is, requests for pages that use "{{FormActionTarget}}" as a target, along with the automatically-generated "__xmlexch=abcd" query parameter in the URL.

Starting with version 12.10, if a request that includes a non-empty __xmlexch query parameter is issued for a page that results in the creation of a new session (either because the current session has expired, or no session ever existed), then service is denied with an error code of 500,10 ("Server Error" with a subcode of "Unexpected").

This change addresses a problem encountered by users of BIS Express: there may be workstations that send refresh requests in the background every minute to keep sessions open. However, if the server is reset, those refresh requests (which include the last known sequence) will create new sessions and start programs that are not in a state where processing can resume, wasting server resources. Those sessions then persist until they time out, but some unattended clients create such new sessions every minute, potentially swamping the server. With this change, such sessions are not created and the refresh requests for defunct sessions from unattended machines are effectively ignored. This change modifies behavior in an error case should not impact normal use of XBIS. However, an option can be set in the registry to cause XBIS to use the previous behavior.

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Liant Software Corporation\BIS\Options]
"CreateNewSequencedSessions"=dword:00000001
```

If this registry value is omitted or set to any value other than 1, then the new behavior is in effect.

RPI 1095854 – BIS stylesheets have been updated to use <xsl:stylesheet> instead of <xsl:transform>.

RPI 1095855 – BIS stylesheets no longer refer to the <root> element. For example, raw_payload.xsl now refers to a parameter named Document_Element which has a default value of 'root'.

RPI 1098067 – A problem with the B\$SetInactivityTimeout and B\$SetServiceTimeout calls that was introduced in version 12.09 has been corrected and these functions work properly again.

XML Extensions for RM/COBOL

RPI **604118** – Error 12 improvements. Replace the complete description of error 12 with the following:

The model data-name specified for an import or export (xmlif library) or for the data-name parameter (slicexsy utility) was not found in any symbol table available to the search. When searching a model template file symbol table, the model data-name was not defined in the symbol table contained by the template file. When searching object files, the object files either did not contain an XML Extensions symbol table or the model data-name was not defined in any of the symbol tables found in the object files searched. Note that object files might not contain an XML Extensions symbol table because they were either suppressed when the object files were created or the compiler used to create the object files was not licensed for XML Extensions development. The compiler produces the “Options in effect” listing line “XML compiler (licensed for use with XML Extensions)” when the compiler is licensed for XML Extensions development; if this line is missing from the “Options in effect”, the object files produced by the compiler do not contain XML Extensions symbol tables. When searching object files for a symbol table, the xmlif library counts the number of unlicensed objects that are examined and if this count is greater than zero and the model data-name was not found, an additional status text line similar to the following:

[12] 1 of 1 object files not licensed for XML Extensions

is produced to aid in diagnosing this issue; the “[12]” associates this status text line with the previously reported error 12; the “*n of m*” indicates the number (*n*) of object files examined and the number (*m*) of object files not licensed for XML Extensions development. (See the XML GET STATUS-TEXT statement description for how to get additional status text.)

RPI **607575** – The description of whitespace handling in XML Extensions was not complete. Add the following explanation to the topic “Handling Spaces and Whitespace in XML”:

XML defines whitespace as space, line feed, carriage return and tab characters. XML Extensions has various rules for handling of the individual whitespace characters:

- Space characters are preserved by XML Extensions during import and export, except for leading and trailing spaces as described in the first paragraph of this topic.
- During import all characters received from the parser for XML text nodes for import are preserved. (Note: From about 2009 to 2014, characters less than space, which include line feed, carriage return and tab characters, were unconditionally removed on import when using revision 2.0 model template files; model templates obtained from the object file are considered revision 2.0. Revision 1.0 model template files from version 11 of XML Extensions resulted in the import of all characters received from the parser.) Version 12.10 of XML Extensions provides additional control of how whitespace characters are handled for import; see RPI 1098192 in this document for additional information on this option.

- During export, line feed, carriage return and tab characters present in the exported COBOL data item are preserved in the exported text node. Export to a file may cause LF characters to be translated to a CR/LF sequence on some operating systems (for example, Windows).

The value, “preserve” or “collapse” of the XML reserved attribute XML-space is ignored by XML Extensions unless the parser acts on this attribute.

RPI **1085397** – Error 11: Replace the complete description of error 11 (eliminating the note about error 11 being valid only for versions of XML Extensions prior to version 12) with the following:

The data-name specified for an export (xmlif library) or for the data-name parameter (slicexsy utility) was found more than once in the COBOL object file or library. Additional qualification needs to be provided to unambiguously identify the data-name in the symbol table that is to be exported or sliced.

RPI **1095851** – Fixed a problem where XML Extensions could import data into an area outside the target data area provided. This was likely caused by duplicate names in the symbol table obtained for the program, either from the object file or a supplied model template file combined with insufficient qualification supplied by the imported XML document. Another likely cause was an out-of-date template file that needs to be regenerated with the xmlslice utility using the current revision of the COBOL program. Because of this issue, new warnings were added to alert the user to ambiguous import names and ambiguous model data names. Attempts to store outside the target data area, should they occur, are prevented and an error produced to alert the user.

While fixing this RPI, it was found that the lixmldef.cpy copybook was out-of-date in version 12.09. An updated copybook is supplied with version 12.10. More recent error and warning information, in addition to the new errors and warnings added in the fix for this RPI, are defined in the updated copybook. If the lixmldef.cpy copybook has been deployed to different locations than the installation directory, users are advised to update those copies from the 12.10 installation.

The new errors and warnings documentation, as an addition to Table 1, Summary of Error Messages, Appendix C: XML Extensions Error Messages, is as follows:

-06 Warning -- ambiguous import name.

An element name in an imported XML document is defined more than once in the symbol table determined by the *ModelFileName#DataFileName* parameter of the XML IMPORT statement. Further, the element name is defined more than once subordinate to the model data name. XML Extensions assumes the first of the duplicates found, which is the lexically last defined duplicate, should be used and continues. If that name already has an imported value for the current import, the next duplicate is assumed, that is, a lexically previous duplicate definition. If there is no duplicate name without a previously imported value, a -02 extraneous element warning is produced and the imported text value is discarded.

This warning is not produced if the XML document does not have any text content, after ignoring whitespace, for the element. This warning is also not produced if the XML document contains sufficient qualification, that is, parent element names, to make the reference unique. When a duplicate name is not subordinate to the model data name, this warning is not produced unless all duplicate names are not subordinate to the model data name.

-05 Warning -- ambiguous import target name (model dataname).

The model data name determined by the *ModelFileName#DataFileName* parameter of the XML IMPORT statement is defined more than once in the symbol table determined by that parameter (either in the object program or from a model template file). Additional qualification is necessary to uniquely define the model data name, which is the description of the target data structure for the import. (Note that in the case of nested programs, the program name can be used as a qualifier and might be necessary to differentiate duplicate data-names defined in different programs.)

72 Error -- import offset outside target data structure.

An attempt to store outside the target data structure determined by the *DataItem* parameter of the XML IMPORT statement has been detected. This can happen because of a model template file that is out-of-date with the executing COBOL program; in this case, the model template file needs to be regenerated from the revised COBOL program. Another cause can be duplicate data names in the symbol table and insufficient qualification (parent elements) in the imported XML document to choose the correct data item target for the import. The attempted import is suppressed along with any further import operations for the affected XML IMPORT statement.

Additional text messages produced for warnings and errors have been improved to include the warning or error number that is associated with the additional text. This is helpful when an error terminates the XML Extensions statement, but there are clarifying warnings. For example, an error 72 might terminate an XML IMPORT and there might also be ambiguous name warnings -06 as follows:

```
Error: 72[0] - import offset outside target data structure
[72] Called from line 823 in TestProgram(C:\tests\xml\code\TESTPROGRAM.COB), compiled
2014/11/07 11:27:36.
[-06] Ambiguous import name: ambiguous-data-name Content: N
[72] Element name: ambiguous-data-name Content: N
```

In the example above, a 72 error terminated an XML IMPORT FILE statement; the -06 warning may have been the cause. Note that the additional text frequently provides both the element name and its text content to help resolve the error by reviewing the imported XML document against the provided information.

RPI 1095853 – There was insufficient documentation of how a form or “flat” XML document was imported. Form input can be generated using the form_to_cobol.xsl stylesheet provided with Xcentristy Business Information Server (XBIS) in the samples/common directory. A form document is usually just a list of field-names (element names in the XML document) subordinate to a document element, with no other nesting of element names. The default document element is normally ‘root’, but in 12.10 XBIS, the form_post_to_cobol.xsl stylesheet has a parameter that can be set by the COBOL program to another desired value for the document element name; this document element name usually is the same as the model data name declared in the COBOL program for use as the target data item for the import. A simple example of a form or “flat” XML document with ‘root’ as the document element name follows:

```
<root>
  <field-name-01>Field 01 content</field-name-01>
  <field-name-02>Field 02 content</field-name-02>
</root>
```

Prior to 12.10, XML Extensions failed to validate that the element name in the input XML was subordinate to the model data name when the document element name was 'root'. The 'root' element as the document element was treated as the assumed parent of the field-names, even when the model data name was some other name. The name matching terminated when the 'root' parent element was found and exited without verifying that the field-name was subordinate to the model data name. This was correct only when there were not duplicate names in the COBOL program. When there were two different target areas each having definitions of the same names in the program, XML Extensions could choose the wrong name that wasn't in the target data name for a specific XML IMPORT statement. XML Extensions always chose the lexically last declaration of a duplicate name, which might not be in the target area for the import. In version 12.10, this has been fixed so that even when terminating on the 'root' parent element, XML Extensions verifies that the field-name is defined subordinate to the model data name specified in the XML IMPORT statement. If there are two or more duplicate names subordinate to the model data name, the "-06 Warning -- ambiguous import name" will be produced unless there is no text content, other than whitespace, for the element being imported. Duplicate names subordinate to the model data name should be avoided when using XML Extensions to import data from form or "flat" XML documents.

RPI **1098192** – As noted for RPI 607575 in this document, line feed, carriage return and tab characters were not imported when using revision 2.0 model template files or the XML Symbol table in the object file during import. Version 12.10 of XML Extensions preserves such characters when received from the parser for import. Please note that XML parsers that follow the W3C Recommendation for Extensible Markup Language (XML) translate the two-character sequence CR/LF and any CR that is not followed by a LF to a single LF character. Thus, CR characters are not normally present in imported content. During export to a file, any LF character may be translated to a CR/LF sequence depending on the operating system convention where the parser is running (for example, on Windows).

Because of this change in behavior, additional flag settings have been added to provide compatibility with prior behavior in the unlikely event that the change does cause a customer problem. The flags also provide new whitespace handling capabilities. The flags can be set with the XML SET WHITESPACE-FLAGS statement and their current state can be obtained with XML GET WHITESPACE-FLAGS statement. The lixmldef.cpy file now defines the following whitespace flags constant-names, which can be combined in various ways:

WHITESPACE-DEFAULT-FLAGS (value 0) – No flags set, which is the default when XML Extensions is first initialized and after termination (in case another initialization is done). This constant-name can be used to set the flags back to their default value.

WHITESPACE-STRIP-CONTROL (value 1) – On import, strips all characters less than space as in the 2009 – 2014 XML Extensions implementation, unless one of the preserve flags is set.

WHITESPACE-PRESERVE-TAB (value 16) – When stripping control characters, preserve any TAB characters for import.

WHITESPACE-PRESERVE-LF (value 32) – When stripping control characters, preserve any LF characters for import.

WHITESPACE-PRESERVE-CR (value 64) – When stripping control characters, preserve any CR characters for import. (Note that XML parsers normally translate any CR/LF sequences and any CR not followed by LF to a single LF; thus, CR characters are not normally present and cannot be preserved by this flag setting.)

WHITESPACE-NORMALIZE (value 65536) – On import, collapse any whitespace character sequences (space, LF, TAB or CR) to a single space character. The WHITESPACE-STRIP-CONTROL flag, if set, takes precedence over this flag and this flag, if set, will be ignored in that case.

An example of a meaningful combination of these flags would be 49 as defined by:

78 WHITESPACE-PRESERVE-TAB-LF value WHITESPACE-STRIP-CONTROL +
WHITESPACE-PRESERVE-TAB + WHITESPACE-PRESERVE-LF.

Please note that these flags do not apply when using revision 1.0 model template files from version 11 XML Extensions; for those model template files, only the default flag setting is applicable.

Liant v12 Update 5 (v12.09) Hotfix 2 Supplement

Copyright © 2014 Micro Focus. All rights reserved.

This document describes, for each component of the Liant product set, the changes made in the v12 Update 5 (or v12.09) Hotfix 2 release of the product. In most cases, the changes were the result of a reported product incident (RPI) and in those cases the RPI number is provided for reference.

Relativity Data Manager

RPI **1093632** – Fixed the SQLGetFunctions API call with a FunctionId of SQL_API_SQLFOREIGNKEYS to return a value of FALSE. Previously, SQL_API_SQLFOREIGNKEYS returned TRUE and SQLForeignKeys always returned Driver not capable. That behavior confused some ODBC applications, so it is better to report that Relativity does not support SQLForeignKeys.

Relativity Data Server

RPI **1093632** – Fixed the SQLGetFunctions API call with a FunctionId of SQL_API_SQLFOREIGNKEYS to return a value of FALSE. Previously, SQL_API_SQLFOREIGNKEYS returned TRUE and SQLForeignKeys always returned Driver not capable. That behavior confused some ODBC applications, so it is better to report that Relativity does not support SQLForeignKeys.

RPI **1096117** – Fixed the value length problem that caused the Designer to fail with the message “String length exceeds column length Parameter #4 error. Data truncated.” while importing an FD.

RM/COBOL

RPI **1095017** – Added a new configuration file option to force RM/COBOL to ignore the value of the BLOCK CONTAINS clause, just like Visual COBOL does. Normally, RM/COBOL validates the user-specified block size in the BLOCK CONTAINS clause as a fixed file attribute of an indexed file. But when the indexed file was created by Visual COBOL, which ignores BLOCK CONTAINS, then there is no value to validate and RM/COBOL reports a 39,05 error on the OPEN. This new option allows RM/COBOL and Visual COBOL programs to inter-operate on the same RM indexed files without RM/COBOL reporting a 39,05 error. To specify this new option add the following record to your RM/COBOL runtime configuration file:

```
RUN-INDEX-FILES IGNORE-BLOCK-CONTAINS=YES
```

RM/InfoExpress

RPI **1095017** – Added a new configuration file option to force the RM/InfoExpress Server to ignore the value of the BLOCK CONTAINS clause, just like Visual COBOL does. See the discussion of this RPI in the RM/COBOL section above. To specify this new option for the RM/InfoExpress Server add the following to the Options section of your **rmixsrvr.ini** configuration file:

```
[Options]  
IgnoreBlockContains=Yes
```

Liant v12 Update 5 (v12.09) Hotfix 1 Supplement

Copyright © 2014 Micro Focus. All rights reserved.

This document describes, for each component of the Liant product set, the changes made in the v12 Update 5 (or v12.09) Hotfix 1 release of the product. In most cases, the changes were the result of a reported product incident (RPI) and in those cases the RPI number is provided for reference.

CodeWatch for Windows

RPI **1094427** – Fixed a problem that corrupted the "current working directory" in CodeWatch workspace files when the workspace properties were examined or changed. This also includes fixes to the dropdown in the property sheet. If the "Working Directory" field is left blank, the directory that contains the workspace .CWF file will be used as the working directory. A path relative to the .CWF file may also be entered, or an absolute path may be entered: in the latter case, CodeWatch will now attempt to store the path relative to the .CWF file, if possible, to make the file as portable as possible.

RPI **603544** – Fixed a problem that occurred when CodeWatch was launched with a relative path to the workspace .CWF file. The path was being interpreted relative to the directory that contained the .CWF workspace file instead of the current working directory. This prevented CodeWatch from starting properly in this instance.

RPI **602896** – Fixed a problem that occurred when **File→Workspace→Save As** was selected. The workspace was saved but a spurious message box stating "Attempted an unsupported operation" was displayed.

Liant v12 Update 5 (v12.09) Supplement

Copyright © 2014 Micro Focus. All rights reserved.

This document describes, for each component of the Liant product set, the changes made in the v12 Update 5 (or v12.09) of the product. In most cases, the changes were the result of a reported product incident (RPI) and in those cases the RPI number is provided for reference.

Windows 8.1 and Windows Server 2012R2 Support

Version 12 Update5 (v12.09) has been successfully tested on Windows 8.1 and Windows Server 2012R2.

Windows Development Change

On Windows the various products are now built with Visual Studio 2010 instead of Visual Studio 2005. For the most part, this is just an improvement in the development process. It may have an impact on users that supply their own DLLs developed to interface with RM/COBOL either manually or using CodeBridge. Those customers need to be aware that RM/COBOL now uses the Microsoft Visual C 2010 runtime instead of the Microsoft Visual C 2005 runtime. It would be best, but in most cases not entirely necessary, that the customer supplied DLLs be rebuilt using the Microsoft Visual C 2010 runtime.

Installation

RPI **581766** – Fixed an improper error message that happened during Liant Setup (LiantInstall.exe) when two instances of installation were done at the same time by one Windows user. The error message indicated that the log file could not be written. Now LiantInstall.exe detects a second instance and issues an appropriate error message. Each Windows user can now run only one LiantInstall.exe at a time.

RPI **596671** – Fixed the install.sh script to detect and allow installation of Relativity on HP-UX 11.31.

RPI **600978** – The installation script install.sh for RM/COBOL Plus and RM/COBOL Development system on Unix has been fixed to generate a “#!/bin/sh” line as the first line in the generated rpcstart script.

RPI **1086956** – Fixed the install.sh script to detect and allow installation of RM/COBOL on SunOS 5.11 (Solaris 11).

RPI **1087790** – Fixed license verification (licverifyall) so that it works correctly on Solaris for both Intel and SPARC machine architectures.

Relativity Data Client

RPI **595088** – A problem with a hang when accessing a Relativity Data Server from Unix SVR4 was determined to have been fixed by a previous hot fix to the Relativity Data Server for RPI 1088456.

RPI **1088624** – Relativity Data Manager and Data Client/Data Server now correctly handle the Date data type for an ODBC 2 application.

RPI **1089038** – Relativity has been changed with respect to reporting writable columns. The SQLColAttributes ODBC function now reports whether the underlying columns are writable, rather than reporting all columns as read-only.

RPI **1089202** – Relativity now correctly supports the SQL_C_NUMERIC data type.

Relativity Data Manager

RPI **1088624** – Relativity Data Manager and Data Client/Data Server now correctly handle the Date data type for an ODBC 2 application.

Relativity Data Server

RPI **595923** – The Relativity Data Server installation now asks you whether you want to use the default port 1583 anyway even if it is already assigned in /etc/services, which should be to simbaexpress. If you answer Y or y for yes then the installation continues and uses port 1583. (On newer Linux systems, port 1583 is pre-assigned to simbaexpress in /etc/services, which causes the Data Server installation to fail without this fix. Being able to force the override for port 1583, since in most cases customers do not have simbaexpress, allows those customers to take advantage of the default port number when configuring Data Clients.)

RPI **596671** – Fixed the install.sh script to detect and allow installation of Relativity on HPUX 11.31.

RPI **1087607** – The Connect.log file created by Relativity Server for Windows is now placed in the shared ProgramData folder, rather than the installation folder (Program Files).

RPI **1088456** – The Relativity Data Server no longer hangs or returns an invalid return code when retrieving a large result set. (A hot fix was issued for this problem.)

RPI **1088624** – Relativity Data Manager and Data Client/Data Server now correctly handle the Date data type for an ODBC 2 application.

RM/COBOL

RPI **599883** – Fixed an error in the RM Btrieve adapter that truncated a 16-bit flag value in an 8-bit field.

RPI **1086956** – Fixed the install.sh script to detect and allow installation of RM/COBOL on SunOS 5.11 (Solaris 11).

RPI **1087403** – On Windows 7 and above, when an icon file such as runcobol.ico is used, the taskbar icon is not changed to the desired icon when the taskbar buttons are combined. To workaroud this you can set the taskbar buttons to “Never combine”. To change this setting: on Windows 7 right click on the Start button, select Properties, select Taskbar, and use the Taskbar buttons drop down menu; on Windows 8 and 8.1 right click on the taskbar and then follow the Windows 7 steps above.

RPI 1091695 – The RM Properties Dialog has been changed such that you must be running elevated in order to modify properties for all users. If you are not running elevated then the "All Users" button in the "Scope" box on the "Select File" tab will be disabled (grayed out). In this case the "This User" button will always be selected and the user can only modify his own properties stored in HKEY_CURRENT_USER in the Windows Registry. If you are running elevated then the "All Users" button will be available and the user may modify the properties for all users that are stored in HKEY_LOCAL_MACHINE in the Windows Registry. RMCONFIG.EXE has been changed to include a manifest to request execution at the "highestAvailable" level. If your account is a member of the Administrators group or you right-click on the Configuration shortcut and select "Run As Administrator" then the "All Users" button will be available. (This change to RM/COBOL was required because of a Windows 8 and Windows Server 2012 behavior change.)

RMNet

Two new functions have been added to RMNet, HttpSetConnectTimeout and HttpSetTimeout. HttpSetConnectTimeout specifies the maximum number of seconds that a request waits while attempting to connect to the server, it sets CURLOPT_CONNECTTIMEOUT. HttpSetTimeout specifies the maximum number of seconds that a request is allowed to complete, it sets CURLOPT_TIMEOUT. If a timeout expires then the outstanding request is terminated with an error. See rmnet.pdf in the RMNet subdirectory installed with the product for details.

The RMNet samples have been updated. The calculator sample has been replaced by the rmnettest sample.

RPI 1085433 – The RMNet calculator sample previously attempted to communicate with the non-existent <http://xcentris.org> site. The sample has been replaced as noted above to avoid this issue.

WOW Extensions for RM/COBOL

RPI 600978 – The installation script install.sh for RM/COBOL Plus and RM/COBOL Development system on Unix has been fixed to generate a "#!/bin/sh" line as the first line in the generated rpcstart script.

RPI 1091959 – The WOW Thin Client server, libtclnt.so (loaded by runcobol), now works correctly on 64-bit Unix platforms (that is, it no longer crashes with a segmentation fault).

Xcentrisity® Business Information Server (BIS)

Xcentrisity BIS v12.09 on Unix now supports Apache 2.4 except on SCO OS/5. Support for Apache 2.2 continues. Support for Apache 2.0 is deprecated and will likely be removed in the next release. Support for Apache 2.0 on Solaris (SunOS) has been dropped already in v12.09. The Apache 2.4 support is limited to an Apache 2.4 linked with the Prefork MPM.

RPI 593222 – Starting in Xcentricity Business Information Server (BIS) v12.09, the Unix BIS logging service is disabled if no log directory is specified in the Service Engine configuration file, or if the specified log directory is "none" (without quotes). The log directory is indicated by the directive "LogDir" in the file /etc/xbis.conf. (BIS for Unix writes log files, one per day, detailing all activity, including session start/end, service start/end, request from Request Handler to Service Engine, response from Service Engine. With this change, if the directory name for these files is omitted from the configuration file or if it is "none", BIS will not write the log files.)

RPI 595337 – The Tutorial2Dictionary.xml file is now installed with the BIS samples.

RPI 595414 – A problem with XML request payloads larger than 49,152 bytes being presented to the COBOL program as an empty request has been corrected.

RPI 595416 – A problem with stack overflow for large requests has been eliminated.

RPI 1082148 – In BIS Sample 4, corrected service program to properly report a namespace error with a SOAP fault.

RPI 1091214 – The cobol_to_wsdl.xsl stylesheet has been fixed to support three or more dimensional arrays (COBOL tables defined with nested OCCURS clauses) when generating a SOAP binding (WSDL). Before this fix, the stylesheet had been ignoring tables of three or more dimensions and thus the generated WSDL was incomplete if the COBOL described such tables.

RPI 1090391 – BIS no longer fails when request queueing is enabled and a session is randomly assigned one of the high internal session numbers. (NB: Request queueing can be turned off by adding "BISQueueRequests Off" to the mod_xbis.conf file.)

RPI 1090979 – BISMKDIR.exe no longer crashes on Windows 8/x64 or Windows Server 2012 x64 when testing for Vista or later. (A hot fix was issued for this problem.)

XML Extensions for RM/COBOL

RPI 598144 – The cleanup script for the examples has been fixed so that it now works on Linux.

RPI 601284 – The ICONV support for character encoding transformations on Unix was fixed for various issues on some Unix platforms.

Liant v12 WS4 (v12.08) Supplement

Copyright © 2013 Micro Focus. All rights reserved.

This document describes, for each component of the Liant product set, the changes made in the v12 Web Sync 4 (WS4 or v12.08) of the product. In most cases, the changes were the result of a reported product incident (RPI) and in those cases the RPI number is provided for reference.

Windows 8 and Windows Server 2012 Support

Version 12 WS4 (v12.08) has been successfully tested on Windows 8 and Windows Server 2012.

Relativity Data Client

ODBC 3.5 Support:

The Microsoft Open Database Connectivity (ODBC) interface is a C programming language interface that allows ODBC-enabled applications to obtain access to data from a variety of database management systems (DBMSs). ODBC is a low-level, high-performance interface that is designed specifically for relational data stores. The 3.5 version of the API will permit more ODBC-enabled applications to use Relativity, which previously implemented the 2.5 API version with certain 3.x additions.

In order to be able to access the ODBC 3.5 API, a 12.08 Data Client is required as well as a 12.08 Data Server. A 12.08 Data Server will support Data Clients prior to 12.08, but only at ODBC 2.5 functionality. A 12.08 Data Client will **only** work properly with a 12.08 Data Server. Thus, customers who are upgrading to 12.08 must upgrade the Data Server before upgrading the Data Clients.

Relativity Java Client

RPIs:

1085071 – The getFetchSize and setFetchSize methods of the RelativityStatement class no longer unconditionally throw a SQLException. The setFetchSize method throws an exception only if passed an invalid row count argument.

Relativity Data Manager

ODBC 3.5 Support:

The Microsoft Open Database Connectivity (ODBC) interface is a C programming language interface that allows ODBC-enabled applications to obtain access to data from a variety of database management systems (DBMSs). ODBC is a low-level, high-performance interface that is designed specifically for relational data stores. The 3.5 version of the API will permit more ODBC-enabled applications to use Relativity, which previously implemented the 2.5 API version with certain 3.x additions.

RPIs:

1074370 — When Relativity processes a query for which the WHERE clause does not specify the value of the first column of any index defined on the table, the first index was used by default; this was sometimes a poor choice. With these changes, Relativity attempts to use information from the WHERE clause, resulting in better performance for some queries.

1082951 — When Relativity creates an index that includes a nullable column that is defined on an item (other than the first) in a set of redundant data items, the index would be defined with a type that was not generally usable. In addition, the index was always marked nonunique, even if it could be unique. These problems have been corrected.

Relativity Data Server

ODBC 3.5 Support:

The Microsoft Open Database Connectivity (ODBC) interface is a C programming language interface that allows ODBC-enabled applications to obtain access to data from a variety of database management systems (DBMSs). ODBC is a low-level, high-performance interface that is designed specifically for relational data stores. The 3.5 version of the API will permit more ODBC-enabled applications to use Relativity, which previously implemented the 2.5 API version with certain 3.x additions.

In order to be able to access the ODBC 3.5 API, a 12.08 Data Client is required as well as a 12.08 Data Server. A 12.08 Data Server will support Data Clients prior to 12.08, but only at ODBC 2.5 functionality. A 12.08 Data Client will **only** work properly with a 12.08 Data Server. Thus, customers who are upgrading to 12.08 must upgrade the Data Server before upgrading the Data Clients.

RPIs:

1083045 — The new PreventWriteStarvation option was not being properly communicated to the RM/COBOL file manager.

Relativity DBA

No changes.

Relativity Designer

No changes.

InstantSQL for RM/COBOL

No changes.

RMNet

No changes.

RM/COBOL

RPIs:

589303 — PDFlib support has been eliminated. This has been deprecated since WS2 (v12.06).

1086913 — Memory is no longer "leaked" when a user program repeatedly calls C\$SETENV to set the same environment variable.

1087442 — The *analysis* utility program for runtime instrumentation (-i option) has been updated to support v11 and later source formats for longer lines (both fixed, with an identification area, and variable, without an identification area). Comments were added to the source, which is supplied, indicating how compiler configuration of the listing can affect instrumentation analysis. Changes were also made to fix file-name case (important for Unix), allow file-names up to 30 characters instead of 8 to match the maximum program-name length and properly scan the listing when the source program contains in-line comments. The analysis program now supports the compiler default of 1024-character source records; because the source is supplied, this can be adjusted shorter or longer if needed because of different compiler configuration settings. In addition, the runtime instrumentation implementation was changed to preserve up to 30 characters for program-names instead of truncating them to 8 characters, which previously resulted in overwriting the output verb count (*.CNT) files if program-names were not unique in the first 8 characters.

XML Extensions for RM/COBOL

RPIs:

588771 — XML Extensions now extracts an object file symbol table to memory instead of a file during export and import operations. This has several benefits:

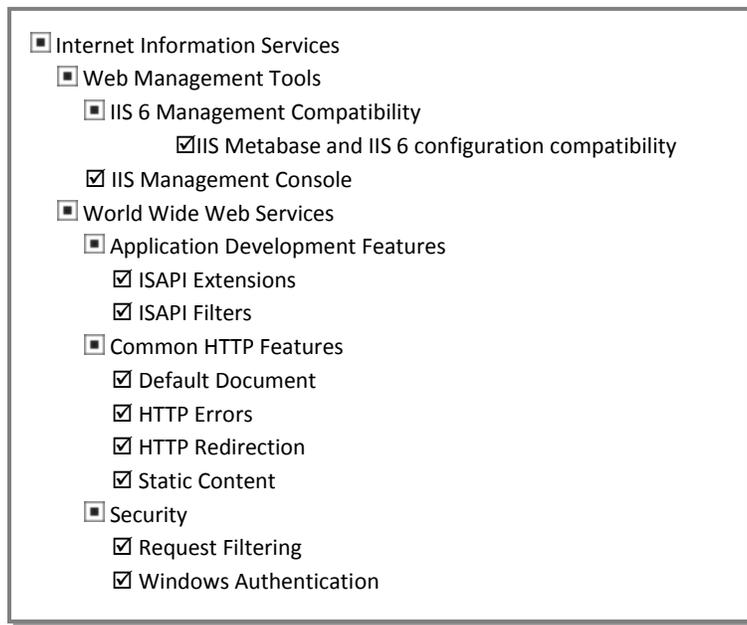
1. Eliminates an unnecessary sequence of write, read and delete file operations, including the permissions problems that can result from these. Not writing a file also eliminates possible file conflicts when multiple processes are run from the same COBOL object file directory simultaneously.
2. Avoids exposing program data layouts in a disk file during execution.
3. Eliminates a bug where a null document pointer could crash XML Extensions during import.
4. Eliminates a bug in file path name manipulations that caused a problem when directory names in the path contained extensions.
5. Eliminates a bug that caused the temporary disk file to be written in the directory containing the directory containing the object file instead of, as documented, the directory containing the object file.
6. Slight performance improvements.

Xcentrity® Business Information Server (BIS)

Installation on Microsoft Windows 8:

This operating system uses a new version of Internet Information Server (IIS): version 8. To configure IIS on Windows 8,

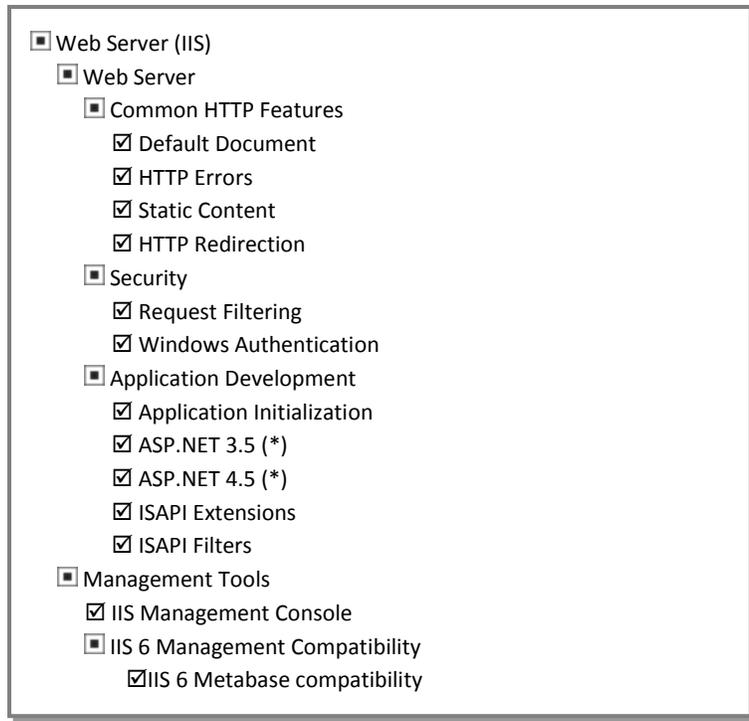
- 1) Open the **Control Panel** and select **Programs and Features**. Or, press the **Windows Key+X**, then select **Programs and Features** from the menu.
- 2) Select **“Turn Windows Features On and Off”** from the side of the window. A list of features will appear.
- 3) Ensure that the items below are checked. Other items may also be checked, but the following checked items are required:



Installation on Microsoft Windows Server 2012:

This operating system uses a new version of Internet Information Server (IIS): version 8. To configure IIS on Windows 2012 Server:

- 1) Launch the **Server Manager**, and click **“Dashboard”**.
- 2) Under **“Select Installation Type”**, click **“Role-Based or feature-based Installation”**.
- 3) Click **Next**, select your server, then click **Next** again.
- 4) Under **“Server Roles”**, select **“Web Server (IIS)”** and
- 5) Select **“Turn Windows Features On and Off”** from the side of the window. A list of features will appear. Ensure that the items below are checked. Other items may also be checked, but the following checked items are required:



* Note: BIS does not require ASP.NET, but installing ASP.NET ensures that the BISMKDIR program can properly configure IIS version 8.

6) Click **Next** and then click **Install**.

BIS Logging:

Starting in Xcentrity Business Information Server (BIS) v12.08, the Windows BIS logging service is disabled by default. It is enabled if a system-wide environment variable is set.

BIS_LOG=[OFF | ON | *directory*]

Values:

OFF disables logging (the same as if **BIS_LOG** is not specified or is blank)

ON enables logging and directs the log files into the default location, which is guaranteed to be writable.

directory enables logging and directs the log files into the specified directory. The user must ensure that it is writable by the BIS request handler.

The directory must be an absolute path or network path. If the specified directory does not exist, BIS will attempt to create it. Note that the containing directory must exist.

The **BIS_LOG** variable is only examined when the BIS application pool is started or recycled. Be sure to restart IIS after changing this variable so it takes effect.

Starting in Xcentricity Business Information Server (BIS) v12.08, the Unix BIS logging service is disabled if no log directory is specified in the Service Engine configuration file, or if the specified log directory is "none" (without quotes). The log directory is indicated by the directive "LogDir" in the file /etc/xbis.conf.

XSLT Stylesheets

The XSLT stylesheets that provide 'out of the box' SOAP web service capability have been enhanced to allow the use of a dictionary document to modify the spellings of method names and parameter names. This same dictionary may also be used to specify bidirectional date and time data conversions between typical COBOL representations and standard XML representations.

The dictionary document is specified using an optional additional XSL parameter named 'spellingDictionary'. If used, this XSL parameter must be specified for XML EXPORTs of the WSDL and the SOAP response, as well as the XML IMPORT of the SOAP request. The tutorial supplied with Business Information Server has been updated to use this feature, and the tutorial documentation describes this capability in more detail.

RPis:

1085819 — The Unix BIS service engine no longer hangs when a large number of requests are received in rapid succession.

1077519 — The Unix BIS request handler has been enhanced to queue requests to a session that is already processing a request, instead of returning an error to the client. This behavior is now the default behavior, but an Apache configuration option of "BISQueueRequests Off" will revert to the original behavior.

RM/InfoExpress Server

No changes.

WOW Extensions for RM/COBOL

No changes.

Liant v12 WS3 (v12.07) Supplement

Copyright © 2012 Micro Focus. All rights reserved.

This document describes, for each component of the Liant product set, the changes made in the v12 Web Sync 3 (WS3 or v12.07) of the product. In most cases, the changes were the result of a reported product incident (RPI) and in those cases the RPI number is provided for reference.

Relativity Data Client

No changes.

Relativity Data Manager

RPIs:

581990 — Fixed a problem with the Relativity Configuration applet not loading on XP.

Relativity Data Server

RPIs:

1078614 — Correct the problem caused when "push-down filter" optimization is enabled, if the filter included columns in the selected table index, it was possible that the query might fail with the error, "The key is completely made.".

1080924 — RelativityConnectionPoolDataSource.java has the following new properties:

- `databaseName` – the name of the database (the server data source)
- `description` – the description of this data source
- `portNumber` – the server port number, as a short
- `serverName` – the name or IP address of the server
- `arrayFetchBuffer` – the array fetch buffer size, as a short
- `user` – the user ID
- `password` – the password

The addition of the properties allow WebSphere to connect to a Relativity data source.

1082683 — Fixed a problem in the Relativity Data Client setup control panel applet where the server port number was constrained to less than 32767, when the limit is actually 65535.

1082646 — A new configuration parameter, `PreventWriteStarvation`, is added. Relativity should be configured with the same logical value, 1 to mean YES or 0 to indicate NO, as the associated application's RM/COBOL *RUN-INDEX-FILES* configuration record (see **1082646** below for additional information). This configuration parameter is also added to the Server Administrator dialog.

Relativity DBA

No changes.

Relativity Designer

No changes.

InstantSQL for RM/COBOL

No changes.

RMNet

No changes.

RM/COBOL

RPIs:

589916 — A problem with debugging (command-line or CodeWatch) of external index-names that was introduced in v12 has been fixed.

1078521 — The install.sh script is now correctly invoked from a CD or any other read-only file system. The system attempts to place the log file first into the /var/tmp directory, then /tmp, then /var. The name of the log file is .rmc_install.log.

1078677 — In the compiler, an incorrect compilation error for an INSPECT statement with the CONVERTING phrase that follows an INSPECT statement with the TRAILING phrase was fixed. This fix also fixed other possible incorrect errors in INSPECT statements.

1078804 — In the CodeWatch editor, restore the capability to use F3 to repeat a search.

1080913 — If the Windows Firewall is disabled, the installation of RM/COBOL failed with the message "Error 1722". Now, installation proceeds with no error message if the Windows Firewall is disabled.

1081523 — Fixed a problem when using multiple record locks in a file.

1082646 — Fixed a problem of 'write starvation' on high performance multiple-CPU systems. Write starvation is exhibited when there are enough long-running read processes on a file that they together effectively lock out file modification operations, including any type of OPEN. A new keyword, PREVENT-WRITE-STARVATION, is added to the RUN-INDEX-FILES configuration record. If the value is set to YES, a file locking method that eliminates write starvation by imposing an additional lock on both the read and modification processes will be used. The default value is NO.

XML Extensions for RM/COBOL

RPIs:

585977 — The RM_ENCODING environment variable now supports the values rmlatin1, mflatin1, xmlextlatin1, builtinlatin1, rmlatin9, mflatin9, xmlextlatin9, builtinlatin9, where all the values with the suffix latin1 are as documented for rmlatin1 and all the values with the suffix latin9 are as documented for rmlatin9.

1078725 — Fixed an XML Extensions defect that caused an XML EXPORT FILE or XML EXPORT TEXT statement to fail when applying a literal result element as stylesheet. Now such exports work correctly.

1079580 — Fixed an XML Extensions defect where an XML EXPORT fails if the data structure contains an OCCURS... DEPENDING ON and XML ATTRIBUTES are enabled.

1080699 — Fixed an XML Extensions defect where if an OCCURS data item is nested within another data item that OCCURS, some, occurrences of the nested OCCURS are not output with XML EXPORT.

Xcentricity® Business Information Server (BIS)

RPIs:

1078595 — Restore the Enable CodeWatch Debugging checkbox to the BISMKDIR dialog.

RM/InfoExpress Server

No changes.

WOW Extensions for RM/COBOL

No changes.

Liant v12 WS2 (v12.06) Supplement

Copyright © 2011 Micro Focus (IP) Limited. All rights reserved.

This document describes, for each component of the Liant product set, the changes made in the v12 Web Sync 2 (WS2 or v12.06) of the product. In most cases, the changes were the result of a reported product incident (RPI) and in those cases the RPI number is provided for reference.

BIS+WOW Thin Client Accept/Display

No changes.

Windows 7 Support

Version 12 WS2 (v12.06) has been successfully tested on Windows 7. Some Windows 7 related issues remaining in the v12.05 product have been fixed in v12.06.

InstantSQL for RM/COBOL

RPIs:

1072711 — Fixed an InstantSQL defect that caused a runtime crash when using bound columns or parameters. The defect used the wrong CodeBridge conversion handle (conversion buffer pointer) for conversion of bound columns and parameters. The defect was masked by the v11 and earlier implementation of CodeBridge, but was revealed by v12 changes to the CodeBridge implementation.

1073475 — Fixed an InstantSQL defect that sometimes caused a runtime crash when using the SQL BIND COLUMN or SQL BIND PARAMETER statements with a large number of columns or parameters. The implementation failed to provide a GIVING or RETURNING argument in a generated argument list. When the generated list was allocated at the beginning of a memory page, a memory access violation could occur when CodeBridge tried to save the non-existent GIVING/RETURNING argument.

Relativity Data Client

RPIs:

573539 — Fixed a problem with the 64-bit data client installation where the data server defined during installation was not available to the server administration control panel applet, thus requiring the data server to be defined again after installation.

581779 — Fixed an installation problem for the Relativity Data Client when the installation is invoked in maintenance mode.

Relativity Data Manager

RPIs:

559726 — Fixed a problem with the global configuration control panel applet on Windows Vista and later versions of Windows that caused the "Program Compatibility Assistant" to ask "Did this control panel work correctly?" The question is no longer asked because a manifest has been added to the control panel applet.

573730 — Fixed an installation problem with the global configuration control panel applet on 32-bit Windows Vista (and later versions of 32-bit Windows) where the applet was not available in v12.05.

1069809 — Fixed a problem in computing the record length when updating a row based on a variable length record. The incorrect record length would cause a 44,7 error or would silently change the record length on relative or indexed files to a potentially incorrect value. The error has been eliminated by correctly calculating the length for the rewrite of the record connected to the updated row.

1075580 — Fixed a byte-order issue that caused big-endian machines (non-Intel memory byte order, LSB in low-order byte) to handle the IF scalar function incorrectly. This affected such big-endian memory architecture platforms as AIX and HP; Intel memory architecture machines (LSB in high-order byte) were not affected by this defect.

1075794 — Fixed a defect that caused an INNER JOIN of two tables, both of which are connected to sequential files, to return a single row when more rows should be returned by the query. Now all the expected rows in this case are returned.

1075939 — Fixed a defect that caused incorrect results from a GROUP BY clause in version 12.05 where version 12.04 obtained the correct results. An oversight in the fix for RPI 1067089 in WS1, another GROUP BY defect, caused this regression.

Relativity Data Server

RPIs:

573352 — The Relativity Data Server does not support using a driver name other than "Relativity Server", but the installation incorrectly allowed the user to change it. There is no evidence that customers actually want this feature, so the Data Server installation was changed to prevent the situation.

581770 — Fixed an installation problem that caused "New Feature" to be displayed where "Query Plan Viewer" should have been displayed. The correct feature name is now displayed.

1069809 — Fixed a problem in computing the record length when updating a row based on a variable length record. The incorrect record length would cause a 44,7 error or would silently change the record length on relative or indexed files to a potentially incorrect value. The error has been eliminated by correctly calculating the length for the rewrite of the record connected to the updated row.

1075580 — Fixed a byte-order issue that caused big-endian machines (non-Intel memory byte order, LSB in low-order byte) to handle the IF scalar function incorrectly. This affected such big-endian memory architecture platforms as AIX and HP; Intel memory architecture machines (LSB in high-order byte) were not affected by this defect.

Relativity DBA

No changes.

Relativity Designer

No changes.

RMNet

Updated RMNet to use more recent cURL and OpenSSL libraries. The cURL library is now 7.21.2 and OpenSSL is 0.9.8o on all platforms on which RMNet is supported.

The TempConvert.cbl example program for RMNet has been slightly enhanced to be a better example.

RPIs:

1077493 — Added two new functions, NetSetSSLKey and NetSetKeyPassword, to RMNet.

RM/COBOL

Language Reference Manual errata:

On page 204, Arithmetic Expressions, Table 20, the first row is incorrect as shown. The first row should contain hash, solid, hash, hash, solid. That is, the second symbol after an operand cannot be an operand, can be an arithmetic operator, cannot be a unary + or -, cannot be an opening parenthesis and can be a closing parenthesis.

RPIs:

578683 — Fixed a license management issue that caused out-dated updates to be left in the license vault when a new update is added to a base license. The license management code was changed to remove existing matching UPDATE amendments from the license vault before inserting new ones.

1067087 — Fixed a CodeWatch editor problem when using “Replace In” set to “Selection”. Text on the remainder of a line at the end of the selection is deleted when text in the selection is replaced. This was a regression from v11.01 and is now fixed.

1072668 — CodeWatch editor has been fixed for a problem with the ‘cut’ operation when characters are selected past the last actual character in the line.

1072678 — Fixed a CodeWatch editor problem that caused incorrect coloring of literals in the COBOL source program.

1072930 — Fixed a CodeWatch problem that made opening copy files inconvenient because of the File Tab jumping to the expanded Program Files node for a program that uses the copy file and then selecting the copybook in that location. Now the File Tab will remain in the “Copy Files” area for the double clicked copy file name as it did in version 11. In addition, the Source Files and Copy Files lists have also been moved to precede, instead of follow, the Program Files and Library Files lists to reduce shifting of the window under most circumstances.

1074065 — Fixed a problem where correct data item value queries either from CodeWatch or the built-in debugger, erroneously resulted in error messages for some identifiers.

1074083 — Fixed an installation problem that caused an error when the Windows firewall was turned off in Windows Vista or later. For Windows Vista or later a new error code was returned in this case and was not properly handled by the installation process. The error code is now interpreted properly and the installation continues instead of failing.

1074438 — Fixed a problem that caused a compiler core dump when an eight-bit (non ASCII) character was used as the currency symbol. An assertion failed in this case when scanning a PICTURE character-string where the leading character was a ‘B’ followed by the currency symbol

character. The test in the assertion has been fixed to handle the case of eight-bit characters.

1075726 — Fixed incorrect use counting on Windows XP Home, Windows Vista Home and Windows 7 Professional. The runtime counted uses in these operating systems as if they were multi-user server operating systems instead of the single-user workstations they are. In this case, uses were over counted, which caused legitimate (per the license) uses to be incorrectly denied.

1075652 — Fixed a CodeWatch problem that caused the Character Set setting not to be saved when choosing Apply or OK and thus not persisting between sessions. The Character Set setting is now saved and persists between CodeWatch sessions for the same workspace.

1076103 — Improved the behavior of two writers writing to the same file opened EXTEND by reducing the time delay before checking to see if the EOF has been moved by another writer. Since the RM/COBOL file manager is not part of the operating system, multiple writers to the same file opened EXTEND need to cooperate somewhat by not writing records as fast as possible, which effectively locks out other writers because the EOF keeps moving. The COBOL programs need to use C\$DELAY after every write, or after a small number of writes, to the file used by other writers to give the other writers a chance to do their writes.

1076502 — Fixed an installation problem that caused an error when the Windows firewall was turned off in Windows Vista or later. Duplicate of RPI **1074083**.

1077514 — Improved the Administrative installation option of LiantInstall to pre-create a LiantInstall.ini configuration file in the target directory. This new configuration file pre-initializes the components to install, disables retrieving of activation codes and disables the License Certificate, Product Selection and Stale Certificates dialogs during subsequent Network installations.

1077559 — Fixed a compiler defect that caused incorrect code generation for an INITIALIZE statement when the subject of the statement was a group that contained two or more variable occurrence data items. The incorrect code would cause a data reference error 104 at runtime when the INITIALIZE statement was executed. The correct code is now generated in this case.

Compiler — Fixed a defect that could cause an erroneous data overflow error when nested OCCURS DEPENDING ON clauses are specified in the COBOL source program. The defect could also have caused an error in the calculation of the minimum record size for a variable length record containing nested OCCURS DEPENDING ON clauses in its data description entry.

CodeBridge — Fixed a defect in the CobolWindowsHandle data conversion function that treated the Windows handle as a signed quantity. In rare cases where the Windows handle value

included a high order bit, an incorrect SIZE ERROR condition would happen in CobolWindowsHandle.

PDFlib support has been migrated to support for PDFlib 8 (version 8.0.1). Support for PDFlib 6 and PDFlib 7 has been dropped, since licenses for those older versions are no longer available from PDFlib GmbH. On Windows, a version 8 pdflib.dll and license must be obtained from PDFlib GmbH; an existing license for PDFlib 7 and a version 7 pdflib.dll will still work on Windows using rmpdflib8.dll. On Unix, PDFlib 8 is statically linked into the installed rmpdflib8.so in v12.06, but a license must still be acquired from PDFlib GmbH to use the full capabilities of PDFlib. There are about 18 new functions in PDFlib 8 that were not available in PDFlib 7; these new functions are not available when using rmpdflib8.dll or rmpdflib8.so. The rmclock.cbl example program has also been slightly enhanced in v12.06. PDFlib support is deprecated and will be eliminated from future releases.

RM/InfoExpress Server

No changes.

WOW Extensions for RM/COBOL

The WOW Designer was fixed to eliminate truncation of some structured comments after column 72 even when the “Generate Short (72 column) Lines” checkbox in Options/Preferences/Code tab was not selected. The designer was failing to check this setting when generating some structured comments and thus always truncated them after column 72. Now the comments will be truncated after column 72 only when the short lines option is selected. Note that the maximum line is 240 characters and thus structured comments longer than 240 characters will be truncated.

RPIs:

1066776 — The incorrect statement in the WOW Extensions User’s Guide that said the TEXT-METRIC structure is defined in **windows.cpy** when it is actually defined in **txtmetric.cpy** was noted in the supplement document for Web Sync 1.

XML Extensions for RM/COBOL

XML Extensions previously wrote temporary files during the execution of the XML EXPORT TEXT, XML IMPORT TEXT, XML TEST WELLFORMED-TEXT and XML VALIDATE TEXT statements. These temporary files are no longer written and the operation of these statements is entirely in memory (except for the external stylesheet input file, if one is specified).

XML Extensions has been enhanced with an XML TRANSFORM TEXT statement that transforms a document in memory to another document in memory. The three parameters for this statement are the input document pointer, the XSLT stylesheet file name and the output document pointer, in that order (the **lixmldef.cpy** file briefly documents this new statement in comments). No temporary files are written during the operation of the XML TRANSFORM TEXT statement. The memory area for the output document is allocated by the XML TRANSFORM TEXT statement and must be freed by using the XML FREE TEXT statement.

XML Extensions no longer supports raw output of XML special characters from FILLER data items; XML special characters in FILLER are now escaped as character references. This was a necessary compromise in the elimination of a temporary file for the XML EXPORT TEXT statement. XSLT stylesheets are a better method of inserting markup text into a result document. The verify sample provided with BIS has been modified from using XML special characters in FILLER to using a stylesheet and is instructive in showing the improved power of using a stylesheet over using XML special characters in FILLER. Associated with this removal of raw XML output, the output of a document prefix feature in XML EXPORT FILE and XML EXPORT TEXT was also removed.

Some examples of XML Extensions, in addition to the previously supplied samples, have been added to the installation. The examples are accompanied by a PDF document describing the examples.

A defect in detecting a “literal result element as stylesheet” (term defined in XML standards) on Windows was fixed that could have caused errors in transcoding the transformed output. The defect assumed that only stylesheets that began with an “xsl:stylesheet” or “xsl:transform” were not literal result element stylesheets. This was the wrong assumption because “xsl” is not required as the namespace prefix for stylesheets. If a different namespace prefix were used as allowed, the wrong decision would be made about the encoding of the result. MSXML6 provides literal result stylesheet transform results in UCS-2 encoding, but normal stylesheet results in the encoding specified by the output element of the stylesheet (a literal result stylesheet cannot have an output element).

A defect in transcoding characters between the Latin 9 local encoding and UTF-8 was fixed. The local character code point 0xA8 was transcribed as if it were Latin Capital U with Macron (code point 0x016a in UCS-2) when it should be Latin Small Letter S with Caron (code point 0x0161 in UCS-2). The defect was symmetric for round trips, but would give the wrong result when a round trip is not done.

RPIs:

578144, 578145 – Fixed problems with an internal data structure size growing wildly large when the COBOL program defined large arrays (COBOL tables), particularly when OCCURS clauses are nested in a data structure. The memory used for the XML Extensions internal data structure in this case has been greatly reduced. The reduction was done in such a way that performance of XML Extensions should also be improved in many cases.

1073539 —Fixed a problem when exporting arrays (COBOL tables) that caused one empty occurrence to be exported when the COBOL table is empty and attributes are enabled. XML Extensions v11 did not export any elements for an empty array when attributes are enabled. The fix restored v11 behavior. Environment variable RM_XML_EXPORT_EMPTY_ARRAY was also introduced to normalize the export behavior for empty arrays:

- When set to the value 'Y', 'y' or '1', an empty array is considered non-optional in the context of the exported document and the empty first element is exported, independently of whether attributes are enabled or not enabled.
- When set to any other single character value, an empty array is considered optional in the context of the exported document the empty array is not exported at all, independently of whether attributes are enabled or not enabled.
- When not set or set to a multiple character value, the v11 behavior is implemented; that is, when attributes are not enabled, the array is non-optional and the first occurrence is exported even if empty; otherwise, when attributes are enabled, the array is optional and an empty first occurrence is not exported.

Regardless of the above, when all occurrences are enabled, all occurrences are exported whether they are empty or not empty and whether attributes are enabled or not enabled.

1074419 – Fixed a defect in XML Extensions that could cause an incorrect error 44, “wrong cobtoxml revision”, when XML INITIALIZE and XML TERMINATE pairs are repeated three or more times in a loop within one run unit.

1074421 – Fixed a V12 regression in XML Extensions where a sparsely populated non-empty, group in a COBOL table would be considered empty and thus not exported unless exporting all occurrences. XML Extensions has been corrected to properly check groups to make sure they are empty before suppressing export of an empty group.

Xcentricity® Business Information Server (BIS)

XML Extensions on Windows writes temporary files to the temporary directory defined by the GetTempPath Windows function. GetTempPath checks for the existence of environment variables in the following order and uses the first path found:

1. The path specified by the TMP environment variable.
2. The path specified by the TEMP environment variable.
3. The path specified by the USERPROFILE environment variable.
4. The Windows directory.

Note that the GetTempPath function does not verify that the path exists. If the path points to a symbolic link, the temp path name maintains any symbolic links.

XML Extensions on Unix or LINUX writes temporary files to the temporary directory defined by the **TMPDIR** environment variable and must have file creation, write and delete permissions for this directory. If the **TMPDIR** environment variable is not defined, the temporary directory defaults to `\tmp` or `\var\tmp` depending on the specific operating system.

XML Extensions is frequently (almost always) used by COBOL service programs run by Xcentrity[®] Business Information Server (BIS). Thus problems with using BIS might be the result of permissions issues with the temporary directory used by XML Extensions.

Error messages produced when running BIS applications are frequently from XML Extensions and are documented in the XML Extensions User's Guide rather than the Xcentrity[®] Business Information Server User's Guide. Also, some problems in using BIS are actually problems with XML Extensions use of temporary files.

BIS has been changed to take advantage of the fact that in WS2, XML Extensions does not write temporary files when the XML EXPORT TEXT and XML IMPORT TEXT statements are used. The BIS request and response documents were previously passed in a file called the "exchange file". BIS now allows the request and response documents to be passed in in-memory strings to avoid writing sensitive information to disk, such as credit card numbers or other personal information. The XML EXPORT TEXT statement can be used to produce a response document in the COBOL service program in memory; note that the COBOL program is responsible for freeing the memory allocated by an XML EXPORT TEXT statement by using a XML FREE TEXT statement, after the response is written to BIS with B\$WriteResponse. The XML IMPORT TEXT statement can be used to process an in-memory request document in the COBOL program; BIS is responsible for freeing the memory used by a request document. Passing the BIS response and request documents in memory is accomplished by changes to the B\$WriteResponse and B\$ReadRequest statements, respectively. A single COBOL pointer data item can be used, much like the single exchange file, but two appropriately named pointers make the program easier to read and maintain.

The B\$WriteResponse BIS service function supports in-memory exchange of the response document by specifying an optional identifier of a COBOL pointer data item in the USING list of the CALL statement. The new calling sequence is as follows:

```
CALL "B$WriteResponse"  
  [ USING ProgramDisposition [ ResponseDocumentPointer ] ]  
  GIVING BIS-Status
```

When the ResponseDocumentPointer argument is omitted, B\$WriteResponse assumes the response document is in the exchange file, as before. When the ResponseDocumentPointer argument is specified, the response document is assumed to be in the memory area referred to by the pointer value. The pointer value is normally obtained before the B\$WriteResponse call by using an XML EXPORT TEXT statement to create the response document from a COBOL data structure in the COBOL service program; in that case, it is the COBOL program's responsibility to free the response memory area by using an XML FREE TEXT statement that specifies the ResponseDocumentPointer argument after

B\$WriteResponse returns to the calling COBOL program. When specifying the ResponseDocumentPointer for the default program disposition, the ProgramDisposition argument may be specified as OMITTED since the ResponseDocumentPointer argument must be the second USING argument to B\$WriteResponse.

The B\$ReadRequest BIS service function supports in-memory exchange of the request document by specifying an optional identifier of a COBOL pointer data item in the USING list of the CALL statement. The new calling sequence is as follows:

```
CALL "B$ReadRequest"
  [ USING TimeoutInSeconds [ RequestDocumentPointer ] ]
  GIVING BIS-Status
```

When the RequestDocumentPointer argument is omitted, B\$ReadRequest writes the request document to the exchange file, as before. When the RequestDocumentPointer is specified, B\$ReadRequest puts the request document in memory and the RequestDocumentPointer value is set to point to that memory, replacing whatever value the pointer data item had before the call. The pointer value is then normally used in an XML IMPORT TEXT statement to obtain the request information into a COBOL data structure for further processing by the COBOL service program. The memory area referred to by RequestDocumentPointer after the call to B\$ReadRequest belongs to BIS and the COBOL program must not attempt to free that memory. When specifying the RequestDocumentPointer for the default timeout value, the TimeoutInSeconds argument may be specified as OMITTED since the RequestDocumentPointer argument must be the second USING argument to B\$ReadRequest.

Note: The B\$Exchange BIS service function has been deprecated and has not been enhanced to support in-memory document exchanges. Calls to B\$Exchange can be replaced by a call to B\$WriteResponse followed by a call to B\$ReadRequest. B\$Exchange can still be used, but only when using an exchange file containing the response and request documents. B\$Exchange will be removed in a future release.

The BIS installation now (as of WS2) includes a tutorial and accompanying examples that are instructive in how BIS can be used. It is strongly recommended that users new to BIS review the tutorial, but even experienced BIS users will find the tutorial helpful and give them new insights into BIS.

The BIS samples have been modified to demonstrate the in-memory exchange of response and request documents. Further, the verify sample has been changed to demonstrate choosing the display language based on the browser preferred language setting (only English and Spanish are demonstrated to keep the sample brief enough for easy understanding). The ASP sample client for driving sample 4, a web services example, has been enhanced to have better error processing and provide an improved user interface experience, particularly when the session is terminated.

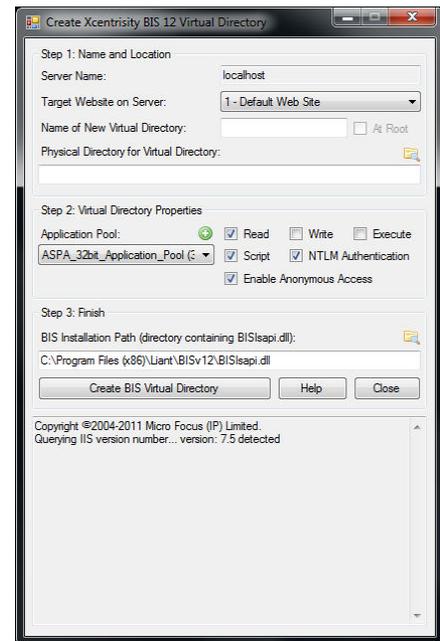
The stylesheets provided for web services have been changed to use the more popular Document/literal wrapped style of SOAP, where they previously used the RPC/encoded style of SOAP. The RPC/encoded stylesheets are still provided, but the names of those stylesheets have been changed to include “_rpcencoded” in the name. The stylesheets have also been simplified through the use of XSLT

parameters that can be set from the COBOL service program because of the v12 addition of the XML SET XSL-PARAMETERS statement in XML Extensions. The BIS web services samples (samples 3 and 4) have been modified to work with the new stylesheets.

Some examples of BIS usage, in addition to the previously supplied BIS samples, have been added to the installation of XML Extensions for RM/COBOL, including some purely XML Extensions examples. The examples are accompanied by a PDF document describing the examples.

RPIs (also see RPIs for XML Extensions):

580776 — The version 12.06 BISMKDIR now properly configures application pool identities in IIS v7.5 (the IIS version delivered on Windows 7 Professional and Windows 2008 R2). BISMKDIR now has a new option for setting the application pool name: `/POOL32 poolname`. If "-" is specified as the *poolname*, then "BISv12-Application-Pool" is used as the application pool name. For IIS 7.5, credentials for a user named "AppPool\poolname" are added to the list of users that are permitted to run BIS applications and access the BIS files. Previously, only IIS_IUSERS and IUSR were granted these permissions. A new button to add application pools is now on the "Create Xcentrinity BIS 12 Virtual Directory" dialog box; an existing application pool may be selected from the drop down list. The dialog varies depending on the version of IIS on the system where BIS is installed.



1077423 -- Fixed a problem in the `cobol_to_wsdl.xsl` and `cobol_to_wsdl_rpcencoded.xsl` stylesheets that caused the generated WSDL to contain "--name" items as output parameters of a method that should not be included. The naming convention intended such "--name" items only as a way to re-name items.

Liant v12 WS1 (v12.05) Supplement

This document describes, for each component of the Liant product set, the changes made in the v12 Web Sync 1 (WS1 or v12.05) release of the product. In most cases, the changes were the result of a reported product incident (RPI) and in those cases the RPI number is provided for reference.

Windows 7 Support

Version 12 WS1 (v12.05) has been successfully tested on Windows 7.

BIS+WOW Thin Client Accept/Display

RPIs:

1069650 —Version 12 WS1 (v12.05) BIS+WOW has removed several dlls from the sample cabinets due to problems with missing Microsoft “c” libraries used by those dlls, which implement Thin Client Accept/Display. The BISplus_HOWTO.doc/pdf contains details of this change.

InstantSQL for RM/COBOL

RPIs:

1069429 —Fixed a defect in InstantSQL, when using a MySQL database on Linux, that caused “glibc detected free(): invalid pointer: 0x08086f2e”. There was a conflict between external symbol names in the RM/COBOL runtime executable and the myodbc shared object. The problem has been resolved by changing the conflicting symbols in the RM/COBOL runtime to be local (static) symbols.

1069445 — Added the InstantSQL documentation, a ZIP file containing the HTML help, to the RM/COBOL development system InstantSQL directory. The file can be unzipped and then browsed by double clicking the [index.htm](#) file in the directory where the zip file is expanded. (NB: The zip file is also available at, and can be downloaded from, the following URL: <http://downloads.microfocus.com/liant/download/isqlhelp.zip>).

1072711 — Fixed an InstantSQL defect that caused a runtime crash when using bound columns or parameters. The InstantSQL defect used the wrong CodeBridge conversion buffer pointer for conversion of bound columns and parameters. The defect was masked by the v11 and earlier implementations of CodeBridge, but was revealed by v12 changes to the CodeBridge implementation to fix memory leaks caused by CodeBridge.

Relativity Data Client

RPIs:

559654 — (not fixed yet; Relativity Data Client segmentation fault when called from InstantSQL on HP-UX PA-RISC systems. On systems other than HP-UX PA-RISC, RPI **1066803** fixed problems with using the Relativity Data Client from InstantSQL).

559679 — Fixed the installation of the Relativity Data Client on Windows to have digitally signed executables.

559727 — Fixed a problem with the server configuration control panel applet on Windows Vista and later versions of Windows that caused the "Program Compatibility Assistant" to ask "Did this control panel work correctly?" The question is no longer asked because a manifest has been added to the control panel applet.

559728 — Created a 64-bit version of the server configuration control panel applet for 64-bit versions of Windows. A 64-bit version is necessary to create the correct registry entries for the 64-bit Data Client introduced in this release.

1066803 — Fixed **SQLDriverConnect** in the Relativity Data Client on some Unix and Linux platforms to not return "success" without doing any work unless **SQL_DRIVER_NOPROMPT** is specified. The data client would crash when a later operation tried to use the uninitialized connection. (NB: SQLConnect worked in the data client, but InstantSQL uses **SQLDriverConnect** with a default of **SQL_DRIVER_COMPLETE** and thus demonstrated the data client failure on the platforms where the problem existed.)

1066963 — Created a 64-bit version of the Relativity Data Client for Windows, which can then be used with 64-bit versions of other database software, such as 64-bit SQLServer, on 64-bit versions of Windows. (The Relativity Data Server remains a 32-bit application and thus no additional database capacity is gained by using the 64-bit client to talk to the server.)

1072464 — Fixed a license certificate issue with Unix Data Server licenses. The issue incorrectly prevented using LiantInstall to install the Relativity Data Client on Windows. The fix has been available in licenses issued since January 5, 2010, when the certificate on the license server was corrected. For Unix licenses issued prior to January, 5, 2010, the Relativity Data Client could be installed on Windows simply by opening the MSI for the data client.

Relativity Data Manager

RPIs:

559675 — Fixed the installation of the Relativity Data Manager on Windows to have digitally signed executables.

559726 — Fixed a problem with the global configuration control panel applet on Windows Vista and later versions of Windows that caused the "Program Compatibility Assistant" to ask "Did this control panel work correctly?" The question is no longer asked because a manifest has been added to the control panel applet.

559729 — The Relativity control panel icon for the DBMS configuration control panel applet is hidden on 64-bit versions of Windows XP, Windows Vista and Windows Server 2008 in a 32-bit Control Panel. On 64-bit XP, the 32-bit Control Panel is opened from the 64-bit Control Panel with the "View x86 Control Panel Icons" icon. On 64-bit Vista and Windows Server 2008, the 32-bit Control Panel is opened from the 64-bit Control Panel with the "View 32-bit Control Panel Items" icon. This release has no 64-bit Data Manager and thus no 64-bit DBMS configuration control panel applet. (NB: See Relativity Data Client in this document for information about the 64-bit Data Client and 64-bit Server Administrator control panel applet.) On 64-bit versions of Windows 7, the 32-bit control panel items are no longer hidden and are shown in the Control Panel with the designation "(32-bit)" appended to the name.

1067065 — Fixed a defect from both v11 and v12 where repeating a certain query multiple times could crash the data manager (or the data server).

1067089 — Fixed a defect with the implementation of the SQL statement **SELECT ... GROUP BY** when applied to a Relativity table based upon multiple COBOL files where an incorrect, and sometimes infinite, result set was returned for the query.

1070016 — Fixed a defect in installation of the data manager that caused the installation to fail when the Micro Focus file manager library (DLL) and its default version could not be found. The error is now simply logged and the installation continues. If the user then attempts to enable Micro Focus file support during installation, the user can locate the library manually with the browse capability on the enablement page.

Relativity Data Server

RPIs:

559039 — Fixed the installation to create registry value named **SvcConfigDir** instead of the misspelled registry value **SvnConfigDir**.

559676 — Fixed the installation of the Relativity Data Server on Windows to have digitally signed executables.

559726 — See discussion regarding this RPI under Relativity Data Manager.

567488 — Fixed server administrator to accept a network name containing a hyphen.

1067065 — Fixed a defect from both v11 and v12 where repeating a query multiple times could crash the data server (or the data manager).

1067089 — Fixed a defect with the implementation of the SQL statement **SELECT ... GROUP BY** when applied to a Relativity table based upon multiple COBOL files where an incorrect, and sometimes infinite, result set was returned for the query.

1070016 — Fixed a defect in installation of the data server that caused the installation to fail when the Micro Focus file manager library (DLL) and its default version could not be found. The error is now simply logged and the installation continues. If the user then attempts to enable Micro Focus file support during installation, the user can locate the library manually with the browse capability on the enablement page.

Relativity DBA

RPIs:

559677 — Fixed the installation of the Relativity DBA on Windows to have digitally signed executables.

Relativity Designer

RPIs:

559678 — Fixed the installation of the Relativity Designer on Windows to have digitally signed executables.

Relativity JDBC Client

RPIs:

3288 (Tracker) — Fixed a problem in the Relativity JDBC Client where the PreparedStatement.setObject method would generate the error “Dynamic parameter #1 type mismatch”.

RMNet

RPIs:

1066670 — Added a programatic way to configure proxy server settings for RMNet. See the description of the NetSetProxyServer function in the rmnet.pdf documentation for further information. Also added a function to disable (and enable) SSL peer certificate verification. See the description of the NetSSLVerifyPeer function in the rmnet.pdf documentation for further information.

1069427 — Included the RMNet documentation in the product as rmnet.pdf. Previously, the documentation was only available on the Micro Focus web site.

RM/COBOL

RPIs:

568496 — Fixed a problem in the implementation of the RUN-FILES-ATTR configuration record USE-PROCEDURE-RECORD-LOCK-TIMEOUT keyword that could cause a file manager hang when a non-zero value is specified for the value of the keyword.

1065891 — Fixed a CodeWatch defect in the Workspace Properties dialog. When an Overriding or Supplemental Configuration file option specified a file path with a space in the name, CodeWatch could not find the specified configuration file.

1066104 — Fixed a CodeWatch defect that caused an unhandled exception crash when Print Preview was used.

1066606 — Fixed a CodeWatch defect introduced in version 12 where a workspace with more than 511 file entries could not be opened.

1066611 — Fixed a COBOL runtime assertion failure when a LIKE relation condition specified the **CASE-INSENSITIVE** option and the regular expression contained a counted character match, that is, an $\{m,n\}$ quantifier on a character match.

1066705 — Fixed a Linux installation defect where cpio returns "Premature end of file." on some Linux distributions.

1066788 — Fixed a **LicVerifyAll** failure when merging a runtime license into a development system installed license vault on HP/UX Itanium systems. Prior to the fix, the message *LicVerifyAll: License install of 'RM/COBOL Runtime System' failed 0xc1000011: The serial number in the license does not match the serial number of the original license.* could be

incorrectly returned.

1066808 — Fixed a COBOL runtime defect when detecting **SIZE ERROR** condition on binary data items where the **PICTURE** character-string specified more digits than supported by the allocation specified by the **USAGE**, as in **COMP-1** data items or other binary data items with a **USAGE** that specified an allocation override. The **SIZE ERROR** condition could be raised when it did not apply or not raised when it did apply for arithmetic results stored in such binary data items. This defect has existed since v7.

1066815 — Fixed a defect on Unix and Linux systems that prevented loading the termcap version of the COBOL runtime because of a spelling error in the function name **RM_TCSETATTRNOW**.

1066909 — Fixed a COBOL compiler problem in the **REPLACE** statement and the **REPLACING** phrase of the **COPY** statement when replacing a single character (for example, an equal sign) that could cause incorrect replacement actions, including replacing that character at the end of a comment line. This also fixed a problem with multiple replacements on the same line, where a latter replacement on the line might not be done when it should, and a problem with incorrect handling of a period following **END-COPY** or **END-REPLACE** (the period is not part of the **COPY** or **REPLACE** statement when the **END-COPY** or **END-REPLACE** phrase is specified, but rather is a period following the statement; v12.02 incorrectly treated the period as part of the statement).

1066914 — Fixed a defect in the administrative install for RM/COBOL to allow installation from a Windows client onto a mapped network drive.

1072048 — Fixed a compiler defect that produced an incorrect warning (0029: W DATA RECORDS data-name not defined for file: <file-name>) about the record-name specified in the DATA RECORDS clause of a file description entry. The problem was evident when the file is properly described in two or more nested programs (for example, by using COPY statements to copy the same file control and file description entries into the two or more nested programs). There was a compiler error in resolving the file-name that implicitly qualifies the data record-name when the file-name exists in two or more nested programs. The error caused the incorrect compilation warning for all but the first file with the same DATA-RECORDS clause.

RM/InfoExpress Server

RPIs:

1066624 — Fixed a problem with I/O errors 30,21 and 90,07 being returned inappropriately when using two InfoExpress servers on the same local area network.

WOW Extensions for RM/COBOL

RPIs:

1065933 — Fixed **WowGetFocus** to allow the user to obtain the handle of an ActiveX child control.

1066530 — Fixed the WOW runtime to work on Windows 2000 systems.

1066812 — Fixed a WOW runtime defect where the scroll bars and **BackColor** on controls were not redrawn for a WOW application when the whole screen was minimized and restored.

1067022 — Fixed a WOW designer defect introduced in version 12. The CONTROL key could cause selected controls to grow larger and repeated use could cause the designer to crash.

1067027 — Fixed a WOW runtime defect with modal forms created with the click event for an edit box or static label not being modal.

1067053 — Fixed a WOW designer defect in the editor that prevented **CTRL+L** and **CTRL+D** from working in version 12 when they previously worked in version 11.

Documentation Correction — The GetTextMetrics documentation states that the TEXT-METRIC structure is defined in windows.cpy. This is inaccurate, it is defined in txtmetric.cpy.

XML Extensions for RM/COBOL

XML Extensions on Unix or Linux writes temporary files to the temporary directory defined by the **TMPDIR** environment variable and must have file creation, write and delete permissions for this directory. If the **TMPDIR** environment variable is not defined, the temporary directory defaults to **\tmp** or **\var\tmp** depending on the system. Note that XML Extensions is used by Xcentricity® Business Information Server (BIS). Thus problems with BIS may result from permissions issues with the temporary directory.

XML Extensions statements **XML GET TEXT** and **XML PUT TEXT** were fixed to allow binary data on Windows. Previously, binary data could possibly be truncated, for example, at a 0x1a character in the “text”). Unix and Linux systems already allowed binary data without problems.

RPIs:

1067093 —Fixed a problem with XML EXPORT attempting to allocate more than 2 GB of memory and crashing the runtime. The data structures in XML Extensions were revised to significantly reduce the amount of memory allocated when large occurrence tables (arrays) in the COBOL program are exported.

1069607 — Fixed output of arrays to suppress empty occurrences when attributes are enabled and all occurrences are not enabled. Version 11 suppressed empty occurrences correctly, but version 12 incorrectly output all occurrences whenever attributes were enabled.

1069720 – Fixed problems with algorithms and data structures that caused excessive memory use and slow execution for both XML EXPORT and XML IMPORT when large occurrence COBOL tables (arrays) were exported or imported. The problem was particularly evident when nested tables (tables containing tables) were used in the COBOL program.

Xcentrity® Business Information Server (BIS)

Xcentrity® Business Information Server (BIS) makes extensive use of XML Extensions for RM/COBOL. Error messages produced when running BIS applications are frequently from XML Extensions and are documented in the XML Extensions User's Guide rather than the Xcentrity® Business Information Server User's Guide. Also, some problems with BIS are actually problems with XML Extensions use of temporary files, as explained in the XML Extensions for RM/COBOL topic of this document.

RPIs (also see RPIs for XML Extensions):

559705 — Fixed a request handler crash that occurred if the **IUSR** security principal is selected to serve anonymous requests, but does not have access to the directory containing the service engine programs. The crash is fixed, and the **IUSR** identity is now granted access to the service engine programs during installation and registration. In addition, other enhancements have been made to clearly explain when the **IUSR** identity should and should not be selected during installation or registration.

559768 — Fixed an installation problem on 64-bit systems running IIS 7 that prevented the application pool from being usable after the installation of BIS on those systems. IIS 7 required the application pool to be marked as allowing 32-bit applications on 64-bit systems. The installation now marks the application pool appropriately when necessary.

567867 — Fixed a problem that caused BIS to hang if a malformed IP is specified in the {{Trace}} tag of an SRF file.

1066093 — Fixed an intermittent problem where after a few days the Unix or Linux BIS request handler would refuse incoming connections and a restart was required to allow new incoming connections to be processed.

1066143 — Fixed an “access denied” issue when executing BIS applications on non-English Windows systems such as Windows Vista and Windows Server 2003 where the “**AUTHENTICATED USERS**” security ID is translated to the locale of the system (BIS installation incorrectly used the untranslated name “**AUTHENTICATED USERS**”).

1066570 — Fixed the Windows utility **BISMKDIR** to support IIS web site identifiers with a value outside the range 1 – 99. This was a prior restriction, but this release allows site ID values in the range 1 – 2,147,483,647 (= (2 ** 31) – 1).

1066938 — Fixed the BIS Windows installer to create the **XBIS12-Application-Pool** as a 32-bit application pool on 64-bit versions of Windows Vista and Windows Server 2008/2008R2 systems, which allows BIS to run successfully on these systems. The Microsoft KB article <http://support.microsoft.com/?id=895976> has additional information about the issue resolved by this fix.

1067102 — Fixed the Windows utility **BISMKDIR** to properly create virtual directories for a 32-bit compatible application on 64-bit Windows systems that run IIS version 7.

1071467 — Modified xbisctl utility to provide information for BIS error 29 on a LAN (too many IP address blocks). The extra information reflects the kind of license (LAN versus WAN) and, for LAN licenses, the IP address range for the server to aid administrators in diagnosing the error 29.

1072047 — Modified the bis/samples/common/soap_request_to_cobol.xsl stylesheet to correctly support web service array input for Java and PHP. Previously, the stylesheet supported only the Microsoft ASP method of array input for web services as demonstrated in Sample4 of the BIS samples.