Synergy DBL
Version 10.3.3e
Release Notes
February 2018

These release notes describe changes that have been made in Synergy DBL in version 10.3.

CONTENTS

IMPORTANT NOTES AND WARNINGS

-- If you are upgrading from a previous version of Synergy, we strongly recommend that you review the Quick Migration Guide and the Updating Synergy/DE manual for important upgrade information.

-- You should always check the Synergex secured web site (http://www.synergex.com) for any patches that may be available for this version. Log into the Resource Center and then navigate to the Downloads page.

Compatibility

-- Based on recommendations from the NIST (National Institute of Standards and Technology) Computer Security Division, we enhanced the DATA_ENCRYPT and DATA_DECRYPT routines for better security. We recommend that everyone using these routines upgrade to a minimum version of 10.1.1a for improved security. All prior versions of Synergy/DE (9.3.1, 9.5.1, 9.5.3, and 10.1.1) that access data encrypted with 10.1.1a or later will require a hot fix from Synergex. THIS CHANGE MAY BREAK YOUR CODE IF USED ALONGSIDE OLDER VERSIONS. [tr#34363]

-- Version 10 will run traditional code compiled and linked with Synergy/DE 6.1 and higher. You do not have to recompile unless you want to take advantage of performance improvements (or to debug if you’re coming from a version prior to 9).

-- If you have code compiled with a version prior to version 9, you must recompile in order to debug.

-- If you are using non-Synergex-provided classes and you rebuild any object module using a class, YOU MUST REBUILD all object modules using that class.

-- (.NET) If your application uses global commons, global data sections, or public class fields that are accessed across assemblies, any time one of those elements changes, YOU MUST RECOMPILE all projects that reference the assembly containing them. As an example, in version 10, the UI Toolkit globals changed, so therefore any application that uses Synergex.SynergyDE.tklib.dll must be recompiled. To ensure that this occurs, we have changed the assembly version of the tklib.dll to force a recompile. We recommend that you use assembly versioning on your dependent projects as well. (Traditional Synergy does not have this problem because names are resolved at runtime, not at build time.)

Version 10.3.3e notes and warnings

-- (Windows) Due to new Authenticode signing procedures from Microsoft, we no longer support Windows Server 2008 or Windows Vista. With the installation of Microsoft KB2763674, software may still run, though unsupported. [tr#37659]

Version 10.3.3d notes and warnings

-- Subroutines and functions in traditional Synergy are now automatically added to the global namespace. As a result, the SYNDEFNS environment variable is no longer used to set a default namespace, and its value must now be a semicolon-delimited list of namespaces to implicitly import. THIS CHANGE MAY BREAK BUILDS FOR TRADITIONAL SYNERGY PROGRAMS AND LIBRARIES. If subroutines and functions have been added only to a default namespace, an import that specifies that namespace will now fail, because there will be no classes in the namespace to import. To correct this problem, add a single class to the namespace or remove the IMPORT statement. All .dbp files should be regenerated for this change. [tr#37211]

-- The runtime now correctly throws an error if an object is passed to a non-object parameter without an explicit cast/conversion first. THIS MAY BREAK YOUR CODE AT RUNTIME if it doesn’t cast or convert. In addition, passing a string to an unprototyped alpha now converts the string to alpha. [tr#37277]

-- To avoid the possibility of not matching records that contain leading blanks instead of zeros, optimization was turned off in 10.3.3d for a Select with a Where expression that matches an alpha typed or untyped key. Previously, a record that contained alpha key data where a decimal number should be (for example, "   1" instead of "00001") could be missed when optimized. As a result of this change, large Selects may run slower. In addition, Join (On) expressions that don’t match the ISAM file key type now throw an $ERR_INVOPER exception ("Join (On) key type does not match key type in file"). If you’re using the Synergy Select Join feature, THIS MAY BREAK YOUR CODE AT RUNTIME. To avoid these issues, we recommend that you update your ISAM file to include the correct type. However, as a temporary workaround, if you can’t immediately make the changes that will permanently correct the situation, you can set SYNSEL_NUMALPHA_KEYS=1 to ignore the key type mismatch check and thus keep your application from slowing down or failing with an %ERR_INVOPER exception. Warnings about key mismatches are logged if DBG_SELECT=2 is set. Additionally, to locate Selects that no longer optimize, you can set DBG_SELECT_INTERNAL=64 to force an exception to occur in those cases. DBG_SELECT_INTERNAL=128 will force an exception on all Select key mismatches, including those that still optimize to a secondary key. [tr#37356]

-- A regression introduced in 10.3.1b caused isutl -r to introduce index errors. This has been fixed. To ensure isutl didn’t introduce errors to your files prior to this resolution, run isutl -v <file> on suspect files. If an index error is present, re-index using the new isutl (isutl -r <file>). If you don’t have the new isutl yet, you can try to re-index with your existing isutl, and follow these instructions: If isutl -v continues to show index errors, try re-indexing with a different page size. If you’re using default settings, page sizes are page=1024 for REV4 and page=4096 for REV6. To try a different page size, specify -qfile=page=#, where # is the new page size. Page sizes are powers of 2 starting at 512. Using a different page size will have no adverse effects on your files. We recommend a page size of at least the default or higher, for example, isutl -r -qfile=page=2048 <filename>. See fcompare and the dv option to validate your data. [tr#37385]

Version 10.3.3a notes and warnings

-- Use of the conditional operator could cause an "Invalid class handle" error. This has been fixed. When using -qrntcompat to target a version earlier than 10.3.3a, a PLATSUPT error is now reported at compile time in situations that previously would have generated the "Invalid class handle" runtime error. A runtime change was made for 10.3.3a to allow this to work properly. If you want this to work, you must target at least 10.3.3a (-qrntcompat=10030301). [tr#36557]

Version 10.3.3 notes and warnings

-- Dblproto now creates a single .dbp file containing prototypes for all subroutines, functions, and classes rather than a file for each routine/class. THIS CHANGE MAY BREAK YOUR CODE, as well as highlight coding errors with your imports and cause TOKUDF errors due to duplicate prototypes. See tr#36331 under "Compiler" in the "REVISION 10.3.3" section below for more information. [tr#36331]

-- The IMPORT statement no longer supports a DIRECTORY specification. You should use -qimpdir or SYNIMPDIR instead. THIS CHANGE MAY BREAK YOUR CODE. See tr#36331 under "Compiler" in the "REVISION 10.3.3" section below for more information. [tr#36331]

-- We changed the way namespaces are imported within a compilation unit. Previously, any namespace defined in a compilation unit was available to all source files in that compilation unit. In addition, a namespace that was defined outside the compilation unit and then imported with the IMPORT statement was available to all subsequent source files on the dbl command line. Now, a namespace defined in a file is available only to that file unless it is explicitly imported into other files in the compilation unit (with IMPORT) or implicitly imported with SYNDEFNS. In addition, a namespace imported with the IMPORT statement applies only to the file into which it is imported. THIS CHANGE MAY BREAK YOUR CODE. If your code relies on the old behavior, you will now get compilation errors (NFND, NVTF, NVTP, or NVTPR). To correct this, you must add IMPORT statements to your code (or redefine SYNDEFNS). We made this change so that Synergy behavior matches other .NET languages, and it enables the compiler to resolve symbols that would otherwise be ambiguous and would prevent IntelliSense and QuickWatch features in Visual Studio from working. [tr#36395]

-- Using dblproto to generate prototype files (.dbp) from multiple source files may cause duplicate symbols to be resolved differently than in past versions. THIS CHANGE MAY BREAK YOUR CODE. Previously, by default dblproto generated multiple .dbp files from each source file. If these files were output to the same directory, duplicates were simply overwritten, and the last file processed "won." Now, dblproto generates a single file from multiple source files, and the order in which files are processed means it is unlikely that the same duplicate definition will prevail as in prior versions.

It is possible to encounter a similar situation at compile time if you have prototype files with duplicate definitions and the new qrelaxed:allowdup option is set. You are most likely to encounter this issue if you run dblproto on files generated by multiple runs of gennet40 (which is NOT recommended). When you do this, the generated files include partial classes (for internal use) with the same names as fully implemented classes. The compiler might now use prototypes for partial classes where previously the prototypes for fully implemented classes were used, resulting in prototype mismatch, method missing, or property missing errors. You can avoid this by generating code for all the types you need with a single run of gennet40.

-- (OpenVMS) The minimum supported version of OpenSSL for the HTTP document transport API and DATA_ENCRYPT and DATA_DECRYPT routines is now SSL 1.4. If your OS version is 8.4 (either Alpha or I64), the default version is now the more secure SSL1 1.0-2C. For earlier versions of the OS, which do not support SSL1 1.0-2C, SSL 1.4 will be installed as the default. THIS CHANGE MAY BREAK YOUR CODE. See "OpenSSL Requirements" in the "Requirements and Considerations" chapter of the Installation Configuration Guide and Synergex KnowledgeBase article 100001979. [tr#36282]

-- Previously, if the last data field in a section was not a handle, the compiler failed to report a LBLSCOPE error ("Label %s out of scope"). This has been fixed. THIS CHANGE MAY BREAK YOUR CODE. See LBLSCOPE in the Error Messages chapter of the Tools manual for more information about what causes this error. [tr#36429]

Version 10.3.1b notes and warnings

-- An "Invalid operation" error (INVOPER) is now reported in every case where a READS, FIND(MATCH:Q_SEQ), or WRITES follows a Select, unless context has been reestablished with a READ, FIND, or WRITE statement. (Until READ, FIND, or WRITE is called in this situation, sequential context is undefined.) Previously, there was one case where this error was not reported: after using Dispose with a Select in a FOREACH statement. This has been corrected, but note that THIS CORRECTION MAY BREAK YOUR CODE. Previously, the runtime would allow your program to call READS, FIND(MATCH:Q_SEQ), or WRITES without re-establishing context after a FOREACH Select was Disposed (even though the results were undefined). Now it will report the error. [tr#35907]

-- (Windows 10) When a machine is upgraded to Windows 10, Windows firewall is automatically enabled. If you upgrade a machine that has Synergy installed and is a license server, you may need to reboot after running the license server once to reset the firewall rules. If Windows firewall was not in use before the Windows 10 upgrade, we recommend you disable it after upgrading. [tr#35881]

Version 10.3.1a notes and warnings

-- When using the Synergy HTTP document transport API, note that SSL 2 and SSL 3 protocols are not secure, and TLS 1.0 has known vulnerabilities. We recommend using TLS 1.1 or TLS 1.2 for secure sites. For more information, see Section 3.1 in nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r1.pdf

TLS 1.1 and TLS 1.2 are available on OpenSSL 1.0.1.

Version 10.3.1 notes and warnings

-- The following versions are the minimums supported in 10.3.1:

AIX 6.1
Solaris 10
HP-UX 11i v3

[tr#34993, tr#34994]

-- (.NET) In rare cases issuing a STOP statement in a WinForms or low-level Windows application caused a segmentation fault. This has been fixed. You must recompile your code with 10.3.1, and the "Target Synergy runtime" option (on the Build page of Visual Studio Project Designer) must also be set to "10.3.1." If it’s set to an earlier version, you won’t get this fix. [tr#35491]

-- (Windows, UNIX) We now support a minimum revision of OpenSSL 1.0.1j on Windows and 1.x on all UNIX platforms except HP-UX, where it is 0.9.8. See Synergex KnowledgeBase article 100001979 for detailed information on OpenSSL requirements and installation. THIS CHANGE MAY BREAK YOUR CODE. [tr#33997]

-- (Windows, UNIX) Synergy/DE 10.3 on Linux and Windows 32-bit systems requires a processor with at least SSE2 instructions (i.e., AMD Athlon 64 or higher and Pentium IV or higher). For Windows, the Sysinternals Coreinfo tool can indicate this. Programs like LMU will fail to operate and the installation will fail without this instruction support. [tr#35398]

-- The performance of simple groups has been optimized. THIS MAY BREAK YOUR CODE, as it affects groups in global commons, global data sections, global literals, and global non-CLS structures, and it is not compatible with assemblies built with older versions of the compiler. As a result, all code that exposes groups in global entities across an assembly boundary needs to be recompiled. [tr#31997]

-- The minimum Framework supported is now CLR 4.0.30319. This version or higher is required in order to install Synergy. [tr#34961, tr#34965]

-- Synergy installations require at least Service Pack 1 on Windows 7 and Windows Server 2008 R2. [tr#34849]

-- We now strictly enforce that the xfServer version be at least as high as the client version.

-- The minimum supported Linux version requires a minimum of GCC4.4 compatible libstdc++ and libtinfo.so.5 and OpenSSL 1.03 to use Synergy/DE 10.3. This support was from 2010 onwards, but lower Linux versions may be able to download the required OpenSSL and libncurses (containing libtinfo) packages to enable runtime-only support. [tr#34992]

-- Because prototypes are built differently in the traditional compiler in 10.3.1 and higher, your existing prototypes must be rebuilt, or a PRTLOAD error will occur. [tr#34308]

Version 10.1.1 notes and warnings

-- Doing a FOREACH over a new instantiation of a class that implements IDisposable now causes the class’s Dispose method to be called when the FOREACH loop is exited. THIS CHANGE MAY BREAK YOUR CODE. [tr#34212]

-- (Windows) Synergy/DE is now always installed to the default Program Files\Synergex location, and this path cannot be changed during installation. Existing installations of Synergy/DE can be upgraded to version 10 and will remain in their current location. (If you need to install Synergy/DE to a non-default location, there is an undocumented way to do this via the command line. Contact Synergy/DE Developer Support for more information.)

Note that because Synergy/DE is now always installed under Program Files, the distributed sample repository files, method catalog, system catalog, and synergy.ini files may need to be copied to a writable location outside Program Files and located via the pertinent environment variables (RPSMFIL, GENESIS_HOME, SFWINIPATH, etc.) [tr#33517, tr#33646]

-- (Windows) Visual C++ 2012 libraries are now installed by Synergy/DE Client. If you are currently running a shared installation and upgrading from version 8.1 through 9.5.3, you must also upgrade the clients. If you do not wish to upgrade (or uninstall/reinstall) the clients, and your shared installation has not changed location, you can instead download the appropriate Visual Studio 2012 redistributables (vcredist) from the Microsoft website and install them on each client machine. See the version 10.1 installation instructions for more information. [tr#33272, tr#33243]

-- (Windows, UNIX) ISAMC_REV now defaults to REV 6. All new files created will be revision 6 unless the environment variable ISAMC_REV is set to a lower revision. Also, setting ISAMC_REV to 2, 3, or 5 will no longer allow creation of files at those revisions. [tr#33556]

-- In 9.5.1 and 9.5.3, the // operator was incorrectly evaluated before * and /. This has been fixed, and these three operators now have the same precedence and are evaluated from left to right. THIS CHANGE MAY BREAK YOUR CODE if you rely on // evaluating before * and /. You may need to add parentheses to explicitly specify the evaluation order. For example, if you have

```
var = a * b // c
```

you would need to change it to this to get the same result:

```
var = a * (b // c)
```

[tr#34101]

-- (Windows, UNIX) The functionality for the OPEN options ALLOC and NUMREC has been disabled. The functionality for system option #15 has been removed. [tr#33176]

-- To match behavior of .NET, an unhandled exception now executes all remaining FINALLY blocks in the call stack (including in the same frame). Also, a STOP or CHAIN no longer executes any remaining FINALLY blocks in the call stack (including in the same frame, a FINALLY on the TRY from which the STOP was issued). Note: This parenthetical case is a change in behavior and MAY BREAK YOUR CODE. [tr#33048]

-- For version 10, we have reviewed areas in our source code and published APIs that use version strings, evaluating them to ensure that the buffers used are long enough to store a two-byte version and identifying places where string comparisons are done against the version. (Our version strings are stored left-justified.) To ensure that string comparisons will continue to work in your deployed applications, some existing version strings now use alphabetic letters to represent versions 10 and higher. For example, the SYN_VER define (dbl.def) is "A.1.1" in version 10 so that existing string comparisons against earlier versions will continue to evaluate as you expect. (Otherwise, comparing "10.1.1" to "9.5.3" would result in "9.5.3" being evaluated as higher.) Because you may be passing SYN_VER to some Synergy routines, we have modified those routines in version 10 to handle both "A" and "10".

If you currently use SYN_VER, when you start developing with version 10, we recommend that you modify your code to use the new SYN_VERSION define (in dbl.def), which is "10.0.3". We also recommend that you review all version string logic in your application for version 10.

-- %VERSN returns the runtime version in a text string (e.g., "DBL/MS Windows Version v.r.m "). If you have code that parses the version number out of this string and then uses it in a string comparison against an earlier version, we recommend you review your code’s string comparison logic before you run version 10.

CHANGE HISTORY

REVISION 10.3.3e

Compilers

For compiler fixes that are specific to the Synergy .NET compiler, see rel_sdi.

-- We added the MAX_SEGBIG environment variable. If MAX_SEGBIG is set to any value less than 65535, the traditional compiler will report a level 3 W_SEGBIG warning when any routine’s code size exceeds that limit. (Code of larger than 65535 bytes still results in a fatal F_SEGBIG error.) [tr#37631]

-- The traditional compiler now reports an INVSTX error when doing a NEW of a dynamic array without specifing the array size or an initial value list. Previously this situation caused the compiler to crash. [tr#35354]

-- Using a hyphen in the output filename to dblproto caused an unexpected dblproto "Invalid namespace" error. Because the .dbp file was not generated, no IntelliSense appeared in Visual Studio. This has been fixed. [tr#37495]

-- The compiler previously reported an IVLNG error on an incorrect line when one of the initial values for a local arrayed field was too long (and there was more than one initial value). This has been fixed. [tr#37499]

-- Providing an initial value list for a new array of multi-dimensional dynamic arrays previously caused runtime errors. Both the traditional and .NET compilers now report a NOTSUPT error when an initial value list is applied to a multi-dimensional, dynamic array. [tr#37502]

-- Line information was incorrect for debugger and runtime errors in certain combinations of IF statements (END ELSE on the same line and nested IFs). This has been fixed. [tr#37583]

-- The traditional compiler no longer crashes when emitting code for an ^M operation that combines range and dimensioned access. [tr#37505]

-- The compiler misreported a DIMMISMCH warning (as an error when compiling with -WE) on routines that had multiple dimensioned parameters. This has been fixed, and we also prevent a check from occurring if the warning level is not at least 4. [tr#37568]

-- Using a new class() inside of an array initializer before the class declaration has been encountered caused a segmentation fault in the traditional compiler. This has been fixed. [tr#37507]

-- In the traditional compiler, we fixed the -WD option to correctly disable the INVOPT warning for -qrelaxed:interop and debug. [tr#37531]

-- The traditional compiler incorrectly reported an E_PLATSUPT ("Static constructor not supported") error on OpenVMS for the default implicit constructor. This has been fixed. [tr#37562]

-- A namespace/class defined locally in the last source could cause paths in previous sources that didn’t contain imports to that namespace to resolve unexpectedly in a previous source, which sometimes caused a runtime error later. This has been fixed. [tr#37592]

-- Two local fields with the same name pointing at different classes previously could cause unexpected NFND errors or incorrect resolution when used in the middle of a path. This has been fixed. [tr#37613]

-- When a project built in debug mode used a class from another project that had an unresolved parameter type (i.e., whose type was defined in another project) in an unused method, the traditional compiler reported an unexpected E_PRTLOAD error for the unresolved parameter type. This has been fixed when -qrntcompat is 10.3.3 or greater. [tr#37634]

-- Under specific code conditions, the traditional Synergy debugger could be off by one line due to compiler line emissions. This has been fixed. [tr#37510]

-- Beginning in 10.3.3, a missing type on a simple property followed by a method with at least one parameter caused the compiler to crash. This has been fixed. [tr#37654]

-- Declaring a local field using inferred typing where the expression was declared later and evaluated to a handle sometimes caused an OHNDCPY error at runtime when the local variable was being used. This has been fixed. [tr#37656]

-- In certain code situations, the compiler crashed after an

F_SEGBIG error. This has been fixed. [tr#37682]

-- Overlay records in traditional class data overwrote the intial value (or default) of the field they overlay. This has been fixed. [tr#37683]

Debugger

-- Under Visual Studio, group argument access in the debugger sometimes failed, ranging in severity from displaying incorrect data to crashing the runtime. This has been fixed. [tr#37542]

-- (OpenVMS) Break points in methods in a shared image can now be set by specifying the mangled name. [tr#37589]

-- Command lines longer than 126 characters no longer hang the Visual Studio debugger. [tr#37590]

-- The Visual Studio debugger couldn’t find a break point on OpenVMS when the version number wasn’t specified. This has been fixed. [tr#37604]

-- More than 64 fields in an unnamed record and a large debugger locals window caused a debugger error: {"Error":"Internal error: CTX pool overflow {64}"}. This has been fixed. [tr#37653]

Linker and librarian

-- We removed an artificial 65535 limitation on the global common name table size. [tr#37643]

Runtime

-- (Windows, UNIX) When using -qcheck in prior versions, and in 10.3.3d with or without -qcheck, if segment reclamation occurred during certain operations (dynamic memory allocation, routine call block API calls, and debugger operations), a segmentation fault could occur. This has been fixed. [tr#37663]

-- (Windows) Attempting to open a TBYTE ISAM file from a user account with low privileges on a network share no longer generates an $ERR_FNF ("File not found") error, "Terabyte files not supported on this platform." (Note: ISAM access is unsupported on network shares.) [tr#37508]

-- In rare cases, the runtime got a segmentation fault while displaying the 14-day copyright message. This has been fixed. [tr#37512]

-- (Windows) The WAIT routine no longer hangs the runtime in certain circumstances (for example, dragging around a title bar). [tr#37526]

-- (Windows, UNIX) The runtime no longer crashes if a record larger than the declared record size is stored to an ISAM file with change tracking enabled. [tr#37574]

-- (OpenVMS Alpha) Executing the I/O hooks constructor no longer causes a crash with an "Improperly handled condition, image exit forced by last chance handler" exception. [tr#37578]

-- (OpenVMS) Previously, virtual methods used for method overloads contained in shared images could not be found, including methods used for I/O hooks. This has been fixed. [tr#37588]

-- (.NET) Version 4.7.1 of the .NET Framework made the MODNAME routine very slow. Although we don’t recomend using MODNAME in high-performance areas (for example, on every READ statement, as it will slow down the read), we have improved performance. [tr#37595]

-- When using the default OPEN statement channel allocator (chn=0), if there aren’t any more channel slots, a CHNEXC error is now correctly generated instead of CHNUSE. [tr#37605]

-- (OpenVMS) An OPEN with chan=0 after a failed call to ISCLR generated an ILLCHN error. This has been fixed. [tr#37625]

-- (Windows, UNIX) A runtime error in an xfServer client application on an OpenVMS server was followed by multiple FILWAS informational errors. This has been fixed. [tr#37456]

-- The .NET runtime accessed incorrect data when a negative relative range was specified. This has been fixed. [tr#37639]

Synergy DBMS

-- We’ve corrected a change made to fconvert in 10.3.3d (tr#37301) that caused an abort when incoming data was zero, null, or blank for an auto sequence key and -ak was not explicitly specified. The new corrected behavior is as follows: If -ak is not specified and the first record of an input file contains zeros, nulls, or blanks at the location of a sequence key in the destination file, the sequence key of that and all subsequent records will automatically be reset to the next sequence as if the -ak option were specified. This same behavior occurs with the isutl utility when specifying the -f fastload option. [tr#37538]

Synergy .NET API

-- An implied-decimal being passed through the .NET interop layer caused narrowing of the whole number part if there were more than 28 digits in total (for example, 31.28 containing 127.1234567890123456789012345678). This has been fixed. [tr#37599]

Synergy routine call block API

-- In Synergy/DE 10.3.1 and higher, in rare cases, an RCB_INSARG or other routine call block API call that required memory and caused memory reclamation could seg fault when bounds checking was enabled. This has been fixed. [tr#37608]

Utilities

-- In Synergy/DE 10.3.3, the dbl2xml utility began outputting XML in which implied-decimal parameters could have a size of 56.28. This caused the Method Definition Utility (MDU) to generate warnings when it imported the XML. The MDU has been updated to allow an implied-decimal parameter length of 56. [tr#37587]

REVISION 10.3.3d

Compiler

For compiler fixes that are specific to the Synergy .NET compiler, see rel_sdi.

-- We added the %DECIMAL function, which converts an integer (i) or implied decimal (.d) value to a decimal (d). For instance, if you have an integer value that you want to pass to an argument marked as IN, you can now use %DECIMAL to convert the integer to decimal, as required by the argument. For example,

```
   xcall mysubr(%decimal(<orig_arg>))
```

[tr#37287]

-- (Windows) We added the -qrelaxed:optval compiler option to turn off the error that occurs if OPTIONAL is specified on a ^VAL parameter. [tr#37194]

-- We added the -WE=<err_num> compiler option (/ERRORWARNINGS=<err_num> on OpenVMS), which turns the specified warnings into errors. You can specify multiple warnings, separated by commas. If both -WD and -WE are specified, -WD takes priority. -WE ignores levels specified by the -W option. For example, even if only level 3 warnings were displayed, -WE would turn a specified level 4 warning into an error.

-- The compiler now reports a level 3 or 4 PASSUR warning when an n (an ambiguous numeric descriptor type) is passed to a d or i parameter, or when a string is passed to an alpha parameter with unspecified direction. The compiler also reports a level 4 W_ADDED warning when it adds a conversion for input parameters, or a level 4 W_ADDED2 warning when the conversion is for n or n. being passed to i. W_PASSIMPL, which occurred when passing d. to d, was changed from a level 4 warning to a level 3 PASSUR warning. See Synergex KnowledgeBase article 100002319 (https://resourcecenter.synergex.com/devres/kb-details.aspx?id=3121) for more information, including techniques for resolving these warnings. [tr#37203, tr#37279]

-- We added two level 3 warnings for truncation. If a literal is truncated and its sign changes, the compiler now reports a LITSIGN warning. If the sign does not change, the compiler reports a LITTRUNC warning. Additionally, the compiler now reports a NARROWING warning for a ^M (mapped) field. [tr#37292]

-- Subroutines and functions in traditional Synergy are now automatically added to the global namespace. As a result, the SYNDEFNS environment variable is no longer used to set a default namespace, and its value must now be a semicolon-delimited list of namespaces to implicitly import. THIS CHANGE MAY BREAK BUILDS FOR TRADITIONAL SYNERGY PROGRAMS AND LIBRARIES. If subroutines and functions have been added only to a default namespace, an import that specifies that namespace will now fail, because there will be no classes in the namespace to import. To correct this problem, add a single class to the namespace or remove the IMPORT statement. All .dbp files should be regenerated for this change. [tr#37211]

-- We improved the compiler so that it now has prototypes for DD_FILESPEC, %RW_METHOD, and RPS_METHOD. [tr#37265]

-- The compiler now allows a boolean value to be passed into an OUT or INOUT numeric parameter. [tr#37256]

-- A missing END statement in a routine caused .NODEBUG not to be applied. This has been fixed. Also, .NODEBUG now applies to every routine from where it was specified to the end of the source file (rather than just to the end of the routine). [tr#37392]

-- When selecting from overloaded methods with the same name and that differ only in optional parameters, the compiler now chooses an accessible method over an inaccessible method. Previously, it reported an ACCESS error. [tr#37066]

-- The compiler now correctly reports an NFND error when [] is used with a structure type declaration. [tr#37124]

-- The message for a NOCONS error (generated when no match can be found for a parameter in a constructor call) now lists the statement that called the constructor. [tr#37150]

-- In versions 10.3.3 through 10.3.3c, if a continuation line for a define started with a single quotation mark, an extra single quotation mark was added to the resulting string. This has been fixed. [tr#37152]

-- Using INIT with a local enumeration data field caused the traditional Synergy compiler to crash. This has been fixed [tr#37174]

-- The compiler no longer incorrectly reports an IVBAD error on fields with initial values in a main routine when the -qnet option is used. [tr#37181]

-- The compiler now reports a REFTYPMSMC error when compiling with -qnet (or a level 3 warning when not using -qnet) when a passed value does not match the argument type exactly and the parameter is BYREF, OUT, or INOUT or has an unspecified direction. [tr#37182]

-- The compiler now consistently reports a NOTIMPLE error when INIT is used with a local field that is an array. [tr#37183]

-- The compiler failed to report a BADACCESS error for a public class field whose type was a private structure or class, and it mistakenly reported a BADACCESS error when ^SIZE was used for a global structure defined later in the compile. These have been fixed. [tr#37201, tr#37235]

-- Dblproto no longer stops with a READER error when it is unable to find a file. It now reports a "File not found" error (FNF) and continues processing. [tr#37210]

-- We fixed several issues that caused the compiler to mistakenly report BSTMTCH, NFND, OUTPARM, and PROTOMISMCH errors. [tr#37224, tr#37227, tr#37254, tr#37257, tr#37259, tr#37242, tr#37252]

-- If an undefined argument number is used as the position for a data reference operation (e.g., ^ARG) that is called in a non-VARARGS routine, the compiler now reports an INVARG warning for traditional Synergy. For example, the following now results in an INVARG warning (or an error in Synergy .NET):

```
    subroutine mysubroutine
        fixedarg,  d
    proc
        if ^arg(4) nop
    xreturn
    end
```

To eliminate these warnings, add the VARARGS modifier to the routine declaration. [tr#37241]

-- The compiler now reports an ACCESS error if a private structure is accessed from another class if that other class inherits from the class that contains the structure. [tr#37243]

-- Previously, the compiler reported INVCAST errors for cast statements that should have resulted in BSTMTCH errors. This has been fixed. [tr#37251]

-- Previously, the compiler reported an NFND error if a MISMATCH n was passed to a method with overloads that accepted n and alpha. The compiler now reports an AMBSYM (Ambiguous symbol) error indicating that the MISMATCH n needs to be cast to an overload type. [tr#37267]

-- Using MISMATCH for a parameter group of type a no longer causes an unexpected INVMOD error. [tr#37271]

-- The compiler now reports a warning when both -qdebug (or -d) and -qrelaxed:interop are specified. (The :interop option prevents a project from being built in debug mode.) [tr#37278]

-- If a structure was duplicated and contained an unnamed field, the compiler mistakenly reported an INTCMPERR error if the program was prototyped or compiled with -qrelaxed:allowdup. This has been fixed. [tr#37294]

-- Documentation comments are now correctly emitted on enumeration values in dblproto. [tr#37325]

-- Nested documentation tags (<para>, <see>, or <seealso>) within documentation comments were being escaped improperly, causing invalid XML. Also, methods were losing their signature. These have been fixed. [tr#37337]

-- Dblproto now reports an XMLDOC warning when a cref path within documentation comments cannot be resolved. [tr#37340]

-- When an implicit main began with a repository .INCLUDE followed by another .INCLUDE file, line numbers in the debugger were off by the number of lines in the second .INCLUDE file. As a result, the debugger reported a "line is not a code line" error when trying to set a breakpoint on a known line number. This has been fixed. [tr#37440]

-- Previously, the compiler reported a TYPPARM error when a type that was declared more than once was used in a method call. The compiler has been improved to report a DUPSYM error on a declaration for a type that’s been declared more than once with both declarations prototyped. You can turn this error off by setting -qrelaxed:allowdup. [tr#37402]

Debugger

-- In some cases, performing a GO/EXIT on a subroutine compiled with -r (or /refresh) caused a segmentation fault. This has been fixed. [tr#37179]

-- Previously, if a descendent class object was passed to a base-class-defined argument when using Visual Studio, base members of the class could not be seen. This has been fixed. [tr#37108]

-- In rare cases, when the runtime failed on initial program startup (e.g., with an "ELB not found" error), the Visual Studio output window didn’t state the reason. This has been fixed. [tr#37146]

-- Previously, an access violation could occur in the debugger when using Visual Studio. This has been fixed. [tr#37148]

-- The type and size of a structfield element under Visual Studio is no longer that of the parent structfield. [tr#37175]

-- Under Visual Studio, the type for boxed data was sometimes incorrect. This has been fixed. [tr#37176]

-- All of the instance fields within an object’s class hierarchy are now displayed. Previously, there was no way to examine the base class variables from an inherited class instance. [tr#37185]

-- Previously, canceling a breakpoint from within Visual Studio failed if the source file contained a logical. This has been fixed. [tr#37186]

-- We now truncate data sent to the Visual Studio debugger at 8192 bytes to avoid making the debugger hang. [tr#37200]

-- When examining the children of an object under Visual Studio, under certain unusual circumstances, the runtime previously crashed. This has been fixed. [tr#37240]

-- The debugger for Synergy in Visual Studio no longer breaks on STOP chains. [tr#37284]

-- The 32-bit debugger got an unexpected HSIZERR error when examining a string larger than 65535 bytes. This has been fixed to truncate at 65535 bytes. [tr#37331]

-- Under certain circumstances, an ArrayList element could incorrectly get a lookup failure. This has been fixed. [tr#37398]

-- In 9.1.1 and higher, the debugger was off by one line under specific code conditions due to compiler line emissions. This has been fixed. [tr#37410]

-- Under certain circumstances, accessing the implied "this" field could cause an access violation. This has been fixed. [tr#37422]

-- (UNIX) In 10.3.3c, running a program with an error (such as file not found on a linked ELB) in debug mode crashed. This has been fixed. [tr#37151]

-- Using a group argument of type n could cause a variety of problems within the debugger, from incorrect data to a full program abort with an access violation. This has been fixed. [tr#37370]

-- When setting a break point using the syntax "break routine line#", if the source file wasn’t directly linked into the routine calling it, the debugger threw a LINE error. This has been fixed. [tr#37133]

-- The debugger no longer gets a "Failed to locate symbol" error on entry to a routine. [tr#37442]

Linker and librarian

-- If ELBs were specified on the command line when linking a .dbr file with no external routine references, the ELBs were not added to the ELB list in the .dbr file. This has been fixed. [tr#37396]

Runtime

-- We removed the pre-reads I/O hook (reads_pre_operation_hook) from being activated on a Select MoveNext method due to lack of functionality. We also deprecated the post-reads I/O hook (reads_post_operation_hook) activated after a Select MoveNext in favor of a post-read I/O hook (read_post_operation_hook) activated on the Current get property (or FOREACH iteration). To avoid breaking existing code, the reads_post_operation_hook will override the new read_post_operation_hook if both are implemented and part of the I/O hooks creation mask. [tr#37232]

-- As a replacement for the deprecated post-reads I/O hook (reads_post_operation_hook) that was previously activated on a MoveNext() method, a new post-read I/O hook (read_post_operation_hook) is now activated when calling the Select Current get property (or a FOREACH iteration on a Select object that calls the Current property). The deprecated post-reads I/O hook will override the new post-read I/O hook, so to change to the new hook you need to disable or remove the post-reads I/O hook (reads_post_operation_hook). [tr#37233]

-- The reads I/O hooks (reads_pre_operation_hook and reads_post_operation_hook) on a JoinSelect MoveNext method have been removed due to inadequate functionality. Instead, the post-read I/O hook (read_post_operation_hook) on a JoinSelect is now activated on the Fill() method or RowEnumerator Indexer method once for each table for which data was transferred. The table index is available in the post-read hook’s keynum argument. [tr#37234]

-- (Windows, UNIX) In version 10.3.3c, specifying RECSIZ=0 on a SORT generated an out-of-memory error. This has been fixed. [tr#37164]

-- When using Select with I/O hooks, the AlphaEnumerator method SparseUpdate() now activates the write_pre_operation_hook() and write_post_operation_hook() when enabled. In addition, sparse records are now ignored or disabled when using I/O hooks. To re-enable Sparse records, disable or remove I/O hooks from the Select. [tr#37202]

-- (Windows) On a stack overflow, the 64-bit runtime now reports a STKOVER error instead of crashing. [tr#37226]

-- Attempting to hook either pre-delete or post-delete I/O hooks and issuing a Select Delete() will now generate an "Invalid operation: Delete allowed on I/O hook channel" error, because

Select Delete() is a remote procedure call and cannot be hooked. [tr#37237]

-- The runtime now correctly throws an error if an object is passed to a non-object parameter without an explicit cast/conversion first. THIS MAY BREAK YOUR CODE AT RUNTIME if it doesn’t cast or convert. In addition, passing a string to an unprototyped alpha now converts the string to alpha. [tr#37277]

-- (Windows, UNIX) In 10.3.3c, a bound chain with debugging enabled could crash the runtime. This has been fixed. [tr#37289]

-- If the command line to a service combined with the path to dbssvc.exe exceeded 260 characters, a segmentation fault could occur on registration. This has been fixed to generate an error if the combined length exceeds 2048 characters. If a dbssvc service is removed with the -x option, a "Message service does not exist" error no longer occurs. [tr#37336]

-- To avoid the possibility of not matching records that contain leading blanks instead of zeros, optimization was turned off in 10.3.3d for a Select with a Where expression that matches an alpha typed or untyped key. Previously, a record that contained alpha key data where a decimal number should be (for example, "   1" instead of "00001") could be missed when optimized. As a result of this change, large Selects may run slower. In addition, Join (On) expressions that don’t match the ISAM file key type now throw an $ERR_INVOPER exception ("Join (On) key type does not match key type in file"). If you’re using the Synergy Select Join feature, THIS MAY BREAK YOUR CODE AT RUNTIME. To avoid these issues, we recommend that you update your ISAM file to include the correct type. However, as a temporary workaround, if you can’t immediately make the changes that will permanently correct the situation, you can set SYNSEL_NUMALPHA_KEYS=1 to ignore the key type mismatch check and thus keep your application from slowing down or failing with an %ERR_INVOPER exception. Warnings about key mismatches are logged if DBG_SELECT=2 is set. Additionally, to locate Selects that no longer optimize, you can set DBG_SELECT_INTERNAL=64 to force an exception to occur in those cases. DBG_SELECT_INTERNAL=128 will force an exception on all Select key mismatches, including those that still optimize to a secondary key. [tr#37356]

-- A condition that caused the runtime to hang on startup due to SYNERGYDE64 being set wrong has been corrected. Also, if SYNERGYDE32 or SYNERGYDE64 is set incorrectly, any attempt to open a file (including the Synergy error file) now displays a BACKUPMODE error ("Backup mode error or SYNERGY64/32 improperly set"). [tr#37358]

-- (Windows, UNIX) Performing a WRITES of a string or large alpha greater than 66000 bytes on .NET or 64-bit platforms caused a segmentation fault. This has been fixed. [tr#37382]

-- Attempting to call a static method in a class that contains a static constructor prior to the class being referenced no longer causes a segmentation fault. [tr#37390]

-- We have avoided a memory fault that could occur if the runtime freed memory for a routine that was no longer in the call stack due to a memory allocation that required using more than MAXMEM memory, and the memory was bound to an RCB block or SQL Connection statement. [tr#37400]

-- (OpenVMS) In 10.3.3 through 10.3.3c, when using SORT on OpenVMS 7.3, an IDENT mismatch error occurred with the SORTSHR shared image. This has been fixed. [tr#37401]

-- During a Join, the isChanged flag on the Rows class method RowInfo() now correctly reports the state of a null row immediately following another null row as being "not changed" or false, and the state of a null row immediately following a joined row as being "changed" or true. [tr#37431]

-- Synckusr.exe and setruser.exe now check if RUSER is set in synergy.ini or synuser.ini, which is not recommended. If RUSER is set in either .ini file, synckusr -r will display its value with a warning message, and setruser will generate a failure message indicating that RUSER is set in one of the .ini files and must be removed before running setruser. [tr#37333]

-- We fixed the following issues with versions 10.3.3 through 10.3.3c of the Synergy .NET runtime:

- SYN_GETSTATE sometimes caused a segmentation fault.

- The runtime crashed if an I/0 hook object was instantiated on the wrong thread after a call to S_SERVER_THREAD_INIT.

- The runtime crashed on shutdown if channels opened by S_SERVER_THREAD_INIT were not closed before thread termination.

- Multiple calls to S_SERVER_THREAD_INIT on the same thread sometimes caused a different routine/thread to get a WRONGTHREAD error.

[tr#37190, tr#37230, tr#37231, tr#37403]

-- We corrected an issue that caused the Synergy .NET runtime to hang on startup due to an incorrect SYNERGYDE64 setting. Now if either SYNERGYDE32 or SYNERGYDE64 is set incorrectly, any attempt to open a Synergy file (including the Synergy error file) will cause an exception ("SynIOException: Backup mode error or SYNERGY64/32 improperly set"). [tr#37359]

Synergy Configuration Program (SynConfig)

-- Changing the license server name using SynConfig from an SDE client when the name of the shared installation server contains a hyphen (-) now successfully changes the license server name. [tr#36437]

Synergy DBMS

-- (Windows, UNIX) Attempting to create a file with more than one autokey of the same type now generates an error. If you run fconvert with the -ak option and there is more than one input key, you must also specify the input file option -k <krf> to define the order for the autokey. The -ak option now resets only sequence autokeys, because resetting a timestamp key to the current time may be undesirable, especially when keys of both types are present in the file. We added a new environment variable, FCNV_RESET_TIMEKEY, to enable you to override this new behavior if desired. To allow timestamp keys to reset to the current time when fconvert is run with -ak, set FCNV_RESET_TIMEKEY=1. Regardless of preservation state, the value on input of a timekey that is 0, null, or blank causes the new timekey value to be reset to the current time. Likewise, the preservation of an invalid incoming sequence key (with the value of zero) now causes fconvert to abort with an error. [tr#37301]

-- Previously, bldism failed to recognize a file passed with the -k option when the runtime launched with the full path or a Windows environment variable. This has been fixed by increasing the command line buffer to 4096. [tr#37272]

-- Running isutl -ra on an intermediate 5r6/4r6 file no longer causes the file to report an "Index too deep" index error. [tr#37355]

-- A regression introduced in 10.3.1b caused isutl -r to introduce index errors. This has been fixed. To ensure isutl didn’t introduce errors to your files prior to this resolution, run isutl -v <file> on suspect files. If an index error is present, re-index using the new isutl (isutl -r <file>). If you don’t have the new isutl yet, you can try to re-index with your existing isutl, and follow these instructions: If isutl -v continues to show index errors, try re-indexing with a different page size. If you’re using default settings, page sizes are page=1024 for REV4 and page=4096 for REV6. To try a different page size, specify -qfile=page=#, where # is the new page size. Page sizes are powers of 2 starting at 512. Using a different page size will have no adverse effects on your files. We recommend a page size of at least the default or higher, for example, isutl -r -qfile=page=2048 <filename>. See fcompare and the dv option to validate your data. [tr#37385]

-- (Linux) Starting in 10.3.3, running chklock could cause an application accessing the same file to either hang or receive a "Record is locked" error until chklock completed. This has been fixed. [tr#37246]

Synergy .NET API

-- We added a .NODEBUG directive at the beginning of the generated file from gennet40 to avoid STEP INTO in the debugger. [tr#37264]

Utilities

-- (UNIX) We updated the Synergy DBL Profiler to more accurately track CPU time and page faults. In addition, child process time is no longer accumulated for the profiled .dbr process.

REVISION 10.3.3c

Compiler

-- We added support for simple (automatically implemented) properties, which are similar to auto-implemented properties for C#. When you declare a simple property, the compiler creates a private anonymous field and get/set methods behind the scenes. The field can be accessed only through the get and set methods. Use the following syntax to declare a simple property:

```
[<access>] [<property_mod> ...] <simple_mod> PROPERTY <name>,
&  <property_type> [,<initial_value>]
```

where <simple_mod> is one of the following simple property modifiers:

READONLY Generate public get method only
READWRITE Generate public get and set methods
SETPROTECTED Generate public get and protected set methods
SETPRIVATE Generate public get and private set methods
SETINTERNAL Generate public get and internal set methods
SETPROTECTEDINTERNAL
  Generate public get and protected internal set methods

See PROPERTY-ENDPROPERTY in the Synergy Language Reference manual for more information and a description of the other arguments. [tr#36250]

-- We added the %SYN_DEBUG routine for traditional Synergy. If a traditional Synergy program is running in debug mode, %SYN_DEBUG returns 1. (Otherwise, it returns 0.) Additionally, it has an optional parameter that is set to 1 if the current routine was compiled in debug mode or 0 if it wasn’t compiled in debug mode. For example, the following code returns 1 (in run_mode) if the program is running in debug mode, and compile_mode is set to 1 if the current routine was compiled in debug mode:

```
   run_mode = %SYN_DEBUG(compile_mode)
```

[tr#32981]

-- We improved the performance of dblproto. It is now up to six times faster on large projects. [tr#36905]

-- We added the RUNTIME_TARGET define. This define’s value is set at runtime to a program’s runtime compatibility setting. This enables you to programmatically determine the runtime target used when the program was built. [tr#36888]

-- We added a level 3 warning, W_USEWARN, that occurs when ^D or ^F is used on an integer variable. [tr#37041]

-- The compiler for traditional Synergy will now report a level 3 DIMMISMCH warning when a pseudo array argument is passed to a real array and -qstrict is used. [tr#37016]

-- In 10.3.3b, accessing a protected member from within a nested class caused an unexpected ACCERR warning. This has been fixed. [tr#36992]

-- If a DATA statement had a complex initial value that required runtime evaluation, the traditional Synergy debugger generated incorrect line numbers. This has been fixed. [tr#36279]

-- In some situations, the compiler did not give the expected NOWRITE error if you used a function, subroutine, or method name as the destination of a WRITE statement. This has been fixed. [tr#36361]

-- In certain cases, the parent source name was not output (or caused an HSIZERR error) when compiling with the -qvar_review option. This has been fixed. [tr#36874]

-- In some rare cases, the compiler crashed when processing undimensioned class fields. This has been fixed. [tr#36900]

-- The compiler no longer crashes if a program that calls XFPL_LOG is compiled more than once during a session (using redirected output). [tr#36909]

-- The compiler for traditional Synergy no longer reports a TYPMISMCH error for prototyped code that passes a scalar alpha argument to a fixed array parameter ([*]a). [tr#36988]

-- Object files generated from traditional Synergy projects with multiple source files previously contained unnecessary line information. This has been fixed. [tr#36934]

-- When a variable defined in a DATA statement was defined without a type and assigned the value of an existing array that was explicitly cast as an array, attempting to access an element of the new array caused the compiler to report an INVNUMDIM error. This has been fixed. [tr#36931]

-- When accessing an outer class from within a nested private class, the compiler mistakenly reported an ACCESS error. This has been fixed. [tr#36991]

-- The compiler no longer mistakenly reports a NOTSUPT error ("Conditional operator is not supported here due to temps") when a ternary operator is used as an argument to an instance method call or an instance property set method. [tr#37001]

-- The compiler previously reported an unexpected HIDEERR warning on a constructor for a child class whose name was the same as its base class. This has been fixed. [tr#37049]

-- The compiler no longer mistakenly reports an ACCESS error when calling a protected method from a class that is nested within an inheriting class. [tr#37071]

-- When compiling source that had a local external function (as well as the fully defined function) using -qrelaxed:param and -qreqproto, the compiler reported an unexpected NFND error on the function call. This has been fixed. [tr#37086]

Debugger

-- We added a SET TRAP IGNORE <error_list> command to allow a break on all errors and exceptions except those listed. [tr#36718]

-- We enhanced syntax checking for the SET DBGSRC command. [tr#37005]

-- SET TRAP ON did not work in callback routines like the Synergy .NET assembly API. In addition, the program didn’t stop at the point of exception when an error was trapped or when an exception was thrown where there was no error trapping or CATCH, making debugging difficult. This has been fixed. [tr#36953]

-- If a file and line number existed within the DBR file and/or one or more ELBs currently attached to an executing program, clicking on that file/line within Visual Studio caused a separate breakpoint to be set/cleared for each of those occurrences. This has been fixed. [tr#36768]

-- We enhanced the examination of structures, structure fields, and structure groups. An error is now reported if a structure field is accessed without a field, a class, or ^M specifying the base address. Previously the debugger indicated that the address was invalid. [tr#36871]

-- Debugging a callback routine from the .NET assembly API sometimes caused the runtime to quit unexpectedly. This has been fixed. [tr#36954]

-- When debugging a traditional Synergy program, the Visual Studio debugger Go command was used when the Step Over command was selected for an OPENELB statement. This has been fixed. [tr#37012]

-- (Windows) In some cases, runtime messages in debug mode were lost in Visual Studio debugging. This has been fixed. [tr#36949]

-- We now provide debug output when a developer uses ^D or ^ID on an integer descriptor. [tr#37035]

-- We added extra informational messages and call stack to the Break on Exception dialog in Visual Studio. [tr#37045]

-- (UNIX) Visual Studio now supports remote telnet debug to a program on a UNIX system. [tr#36720]

-- When a "@*" variable is examined, instead of "<system.object>", an implied cast to the class of the assigned object is performed and the output is now based on that class. This occurs regardless of whether or not the referenced class is declared within the current routine. [tr#36597]

-- We fixed an issue that prevented users from effectively setting variable watches in the traditional debugger. [tr#37091]

Installation

-- We resolved an issue that caused an SDE32 and SDE64 uninstall operation to remove the GENESIS_HOME system-level environment variable if a user had overwritten the value. This fix also prevents the installation from overwriting an existing value on install, allowing a user to correctly configure GENESIS_HOME in advance or when upgrading. [tr#37026]

Runtime

-- The SORT statement has been updated to make better use of memory. SORT now determines optimal memory usage to avoid excessive intermediate I/O with an upper ceiling limit of 4 MB on UNIX and 64 MB on Windows. You can adjust the upper ceiling by setting the new environment variable SORTMEMMAX=<kilobytes> (the number of kilobytes that SORT should not exceed). A significant performance improvement may occcur when sorting large files (with very large and/or many records, including tens of millions). You must unset the existing SORTMEM environment variable to enable this feature; if SORTMEM is set, it will override the optimal memory feature. The Select class, which uses SORT on large selections using Orderby.Ascending and Orderby.Descending, will also use this feature, as will remote SORT over xfServer. [tr#37060]

-- When an error occurs in the windowing routines (for example, W_AREA) and it’s called from a method, the error reporting now unmangles the method name instead of reporting the mangled method name. We added an optional parameter to the ERRMOD routine that (when it’s non-zero and the error routine is a method) causes the method name to return unmangled. If the value is 0 or the parameter is not passed, the mangled name is returned. This also applies to %DBL$ERRMOD. [tr#34310]

-- (OpenVMS) When a filename with an explicit version number was passed to ISAMC and FLAG #3 (supercede) was not set, a new file with the next version number was created. Now the existing file is deleted and the file with that explicit version number is created. [tr#35088]

-- (Windows, UNIX) Structures did not expand in locals, watch, or hover over in the Visual Studio debugger. This has been fixed. [tr#36864]

-- (UNIX) In 10.3.3 through 10.3.3b, a segmentation fault could occur when doing a Select or a Join to an xfServer process from a big-endian client. This has been fixed. [tr#36922]

-- (Windows, UNIX) In 10.3.3b, a Join using a tag field did not correctly match against the tag when the joined files were accessed via xfServer. In addition, using a Sparse object caused xfServer to crash. These have been fixed. [tr#36924]

-- Logging out a process that also used .NET API to Synergy .NET didn’t always log out the runtime license. In addition, segmentation faults could occur when closing RemoteApp sessions. This has been fixed. [tr#36944]

-- (Windows) In rare cases, shutting down a DBR program using a WinForm could cause a "Program has stopped responding" error. This has been fixed. [tr#36960]

-- Explicit and implicit Select AND optimization now occurs when ORs are present but not optimized for OR (like the keys used by the ORs are different or non-keys) or Where.In() on a non-key. Previously, no optimization occurred even though a key on the AND side of the tree could be optimized. For AND optimization, Where.Keynum() is now only required to override OR/In optimization with AND optimization of the specified key. Also, AND must be on top of the expression tree with only ANDs present on one side for AND optimization to occur. (See tr#36601.) Setting DBG_SELECT now displays an "OR+AND (OR on top) - optimization cancelled" message when AND is not at the top and the entire expression cannot be optimized. [tr#36989]

-- In 9.5.1 through 10.3.3b, LPQUE with DBLOPT #22 set failed to spawn dblpq.bat when run under dbr.exe. This has been fixed. [tr#37010]

-- (Windows) The %SYNMSW_GETFILENAME filters parameter was documented as optional, but omitting it generated an "Argument missing" error at runtime. In addition, the compiler treated the title parameter as optional, but it is required. These have been fixed. [tr#37038]

-- (Windows) In some cases after a fatal runtime failure, a Windows system error indicating that there was an invalid throw could occur after the error was acknowledged. This has been fixed. In rare cases, a segmentation fault occurred in the runtime in areas where DTK_BOUNDS=2 would indicate too many rows. The segmentation fault has been corrected to truncate the same way DTK_BOUNDS=2 does to avoid the segmentation fault. [tr#37074]

-- On a READ or FIND where no positioning argument was specified (key specification, POSITION:, ^FIRST, ^LAST, RFA:, etc.), a READ still occurred and a "Key not same" error was generated. A new, more appropriate $ERR_INVRD error ("Invalid READ, %s") is now generated. [tr#32446]

Synergy DBMS

-- The isutl and fconvert utilities take advantage of some new memory enhancements made to SORT. This will improve the performance of indexing during reload and verify operations, as well as data reorganization, especially on very large files. We’ve seen speeds increase up to 10 times faster on very large files. There is an upper ceiling limit of 64 MB for both UNIX and Windows. You can adjust this limit by setting the new environment variable IU_SORTMEMMAX=<kilobytes> (the number of kilobytes that isutl and fconvert should not exceed). The existing SORTMEM environment variable is no longer recognized by these utilities. (The new SORTMEMMAX environment variable is ignored as well.) [tr#37061]

Synergy .NET assembly API

-- The runtime crashed when the assembly type parameter to DotNetDelegate was null or "". This has been fixed. [tr#36520]

-- Code that threw a FileLoadException from .NET into Synergy in the .NET assembly API previously caused the process to crash. Now a DotNetException occurs. [tr#36939]

-- Passing bad data to a .NetAPI routine that takes a decimal argument previously caused a fatal runtime exception. This has been fixed. [tr#37078]

Synergy socket API

-- In 10.3.3 through 10.3.3b, %SS_SENDTO reversed the port and IP addresses. This has been fixed. [tr#36884]

Utilities

-- The setruser utility now accepts passwords longer than 15 characters. [tr#36928]

-- (Windows, UNIX) The listdbr and listelb utilities have a new command line switch (-f), which shows for each routine the MDB flags, the compilation version, and all routines that are called. The link version, endian type, and bit size of the DBR or ELB file is also displayed. [tr#37007]

REVISION 10.3.3b

Compiler

-- We added the compiler option -qreqproto (/REQPROTO on OpenVMS), which enforces that either a prototype or locally defined routine must be available for subroutine or function calls. If only an external function or external subroutine is available, the compiler will return an E_NFND error on the call. This switch causes -X and external function declarations to be ignored. [tr#36831]

-- We added type checking for the Select class's Where/ NoCaseWhere/On expressions. [tr#34383]

-- The compiler didn't report an error when incompatible types were used in the AS clause of FOREACH statements. It now reports errors on .NET, and warnings are reported in traditional Synergy if -qnet is specified. [tr#35973]

-- An unexpected TYPMISMCH error was reported when assigning a ^NULL to a field initializer. This has been fixed. [tr#36554]

-- An invalid E_INVTYP error was reported when passing an .IS. expression as an argument to a boolean parameter in a method call. This has been fixed. [tr#36581]

-- In 10.3.3 through 10.3.3a, an extra quotation mark was added to the resulting string when a continuation line in a .DEFINE led with a quote. This has been fixed. [tr#36634]

-- Defining an interface qualified method name caused a crash in the traditional compiler. This has been fixed. [tr#36591]

-- (Windows, UNIX) When using an input file to compile multiple sources, the error and warning counts were put in subsequent compiles. This has been fixed. [tr#36610]

-- Using a .NOT. operator (!) on an unresolved path caused a segmentation fault. This has been fixed. [tr#36646]

-- Previously, the compiler crashed when doing an INIT of a local variable whose type was a fixed array that had an initial value. Now a NOTIMPLE error is reported. [tr#36653]

-- In 10.3.3 and 10.3.3a, import files with the same name as other already imported files were inappropriately filtered, resulting in undefined errors. This has been fixed. [tr#36655]

-- The compiler incorrectly allowed access to enum values within a private enum if the enum itself was not accessible. This has been fixed. [tr#36656]

-- (Windows) We fixed miscellaneous dblproto errors to output a %DBLPROTO-E- format to better work with Visual Studio. [tr#36668]

-- The compiler now reports an E_NOTSUPT error when the conditional operator is used in a situation that would cause a runtime error. [tr#36679]

-- The backing field for simple properties has been added to the debug tables. [tr#36682]

-- Simple properties generate a get method, set method, and a backing field that are based off the name of the simple property. If that simple property name is greater than 26, or if the name of the backing field exceeds 30, the compiler now reports an E_BIGIDEN error on the property. [tr#36701]

-- Casting any Synergy descriptor type to an object didn't allow the object's method to be called. For example, casting a Synergy alpha variable to string and trying to call TRIM() on it caused an E_NOTSTAT error to be reported. This has been fixed. [tr#36702]

-- An instance property with a static backing field previously caused a segmentation fault. This has been fixed. [tr#36706]

-- Previously, dblproto caused a segmentation fault when trying to demand-load an unresolved type that was part of a prototype file. This has been fixed. [tr#36742]

-- Passing the result of ^M(struct, hnd) as a parameter to a non-CLS structure argument previously reported an unexpected "Type mismatch" error. This has been fixed by adding an implicit [1] to access the first element of the returned alpha array. [tr#36746]

-- Runtime corruption occurred when using a simple property when the simple property was the last declaration in the class. This has been fixed. [tr#36766]

-- We added a W_INNSPACE warning when a subroutine or function is declared in a namespace. This is a level 3 warning in traditional Synergy. [tr#36828]

-- Pseudo arrays of structure types previously caused runtime errors. An E_NOPSEUDO error is now reported when a pseudo array of type structfield is defined. [tr#36814]

-- The compiler was allowing access to PROTECTED and PROTECTED INTERNAL members in violation of .NET’s rules. This is now reported as an E_ACCESS error on .NET and a W_ACCERR warning in the traditional compiler. [tr#36839]

Debugger

-- (Windows, UNIX) Setting a break at the last source line in a routine not compiled with -d previously generated a "Breakpoint line not in module" error. In particular, it failed if compiled with the new -qdebug=1 qualifier. This has been fixed. [tr#36590]

-- (Windows, UNIX) The debugger has been enhanced to allow ^REF arguments to be accessed as a single-dimensioned array. [tr#36778]

-- (Windows) The stepping state in all routines in the current calling chain is now turned off on a GO command. [tr#36731]

-- (OpenVMS) BREAK <routine> <line#> worked incorrectly if <routine> was not current. This has been fixed so you can set a breakpoint in a routine that is not currently resident. (This tracker was fixed on Windows and UNIX in version 10.3.3a.) [tr#36532]

-- Breakpoint memory is now obtained on an executing DBR basis and is discarded when a new routine execution is started. As a result, breakpoints no longer persist across a STOP chain. [tr#36730]

-- (Windows, UNIX) EXAMINE of a ^REF argument failed with an "Undefined variable" error. This has been fixed. [tr#36751]

Linker and librarian

-- Dblink now adds unresolved global data section references to the ELB being created. When an ELB contains a reference to a global data section and that ELB is specified on the dblink command line, a REFBIG warning is now issued. This requires the ELB to be linked with the updated linker. Older linkers and runtimes will ignore the new section in the ELB. [tr#35991]

-- We removed the 65,535 limit on the number of modules in an ELB. [tr#36650]

-- The name of the ELB causing the error has been added to the dblink error message for %DBLINK-F-ELBNAM. [tr#36677]

Runtime

-- SORT/MERGE did not generate an error when one of the key fields did not belong to the specified record. This has been fixed to generate a BADKEY error. [tr#34813]

-- (Windows, UNIX) GetCTInfo from an AlphaEnumerator class no longer returns an invalid CTInfo object if the Select's Where does not specify change tracking. Now a ^NULL object is returned. [tr#35534]

-- (OpenVMS) When a pipe command specified on the OPEN statement contained a comma, the part of the command that followed the comma was ignored. This has been fixed. [tr#35746]

-- A Where expression in a Select statement containing both AND and OR operators that contains a non-conditional OR on any other expression (via the AND) no longer causes the expression to partially optimize, which resulted in fewer records selected. [tr#36594]

-- A Where expression containing ORs such as "Where.In(K1,A,B).AND.K2==C" or "(K1==A.OR.K1==B).AND.K2==C" where an AND operator is at the top of the expression tree can now be optimized using key optimization by explicitly specifying a key on the opposite side of the ORs: "Where.Keynum(K2).AND.(K1==A.OR.K1==B).AND.K2==C". [tr#36601]

-- Where.Keynum() or Where.Portable() specified in a Where expression by itself when used on InnerJoin or LeftJoin no longer causes an "Inner table requires key reference" error ($ERR_JOINKEYREQ) or a segmentation violation if DBG_SELECT is set. [tr#36603]

-- Attempting to reuse From objects on a Join after they were used with a previous Join no longer causes the Join to hang. [tr#36604]

-- (Windows) In 10.3.3 and 10.3.3a, a RENAME or DELETE could hang for 50 seconds instead of 5 seconds on a file in use. This has been fixed. [tr#36606]

-- When using Change Tracking, exceeding the maximum number of concurrent snapshots (using ctutl -s or the ChangeTracking method ApplySnapshot()) now always generates an error indicating the maximum number of snapshots (255) has been exceeded. In addition, issuing ApplySnapshot() no longer generates a segmentation fault when attempting to generate an error. [tr#36614]

-- The Synergy DBL profiler did not log time taken for the main routine. This has been fixed. [tr#36636]

-- Passing an argument that was not passed by the calling routine to a group argument and then using a field of the group as an argument to %STRING caused a segmentation fault. It now returns an empty string. [tr#36657]

-- (Windows, UNIX) Reading and writing file text in an ISAM file has now been coordinated between processes. Changes by one process are now immediately available to another process. [tr#36667]

-- The ISCONFIG subroutine now correctly writes file text to a remote ISAM file across xfServer. [tr#36754]

-- (Windows, UNIX) A JoinSelect using remote files over xfServer now returns the correct data for all joined rows when SCSPREFETCH is set. [tr#36783]

-- LeftJoins in a multi-table join that evaluate to null no longer terminate the further evaluation of additional joins on the same JoinSelect. [tr#36804]

-- Using Contains within an On expression no longer causes an $ERR_INVOPER error ("CONTAINS operator left operand requires field reference"). [tr#36816]

-- A Reset method is now available in the RowEnumerator class for restarting Joins. (Setting -qrntcompat=10030302 or higher is required at compile and link time.) [tr#36820]

-- A segmentation fault no longer occurs on a Select that includes a Where expression containing OR operators referencing the same key field or a Where.In. Typically, this condition was occurring when the Where.In (or the set of OR operations) occurred along with a minimum of two additional AND operations. [tr#36825]

-- S_BLD failed with a BIGNUM error when using an i8 with a large number with a %d format string on a 32-bit platform. This has been fixed. [tr#36794]

-- A Select Where expression with optimized .OR. operations (all .OR. operations are against the same key) or a Where.In (when optimized) now returns selected records when the first (leftmost) condition is false based on the existing data. [tr#36841]

-- A segmentation violation no longer occurs when logging with DBG_SELECT while running a Select with a large Where expression (greater than about 28 operations). In addition, advanced logging with Where.In optimization no longer gets a segmentation fault on .NET. [tr#36842]

-- In rare cases, using a large Select Where expression (greater than about 70 operations) failed by not returning all records or gave other unexpected results. This has been fixed. We also fixed the use of large object trees, specifically when very large Where expressions are used (1000+ operations in a single Where object). [tr#36843]

Synergy DBMS

-- (UNIX) When the ISAM file passed to isutl had an extension other than .ism, .ism was appended to the end of the filename reported in the isutl log file, after the existing extension. This has been fixed. [tr#35350]

-- (Windows) When the ISAM file passed to isutl was specified using a logical, the logical was included in the filename reported in the isutl log file. This has been fixed. [tr#35351]

-- Isutl -rc (to compress ISAM data records) no longer fails with a segmentation violation when the file revision is 4r6 and the file is already PAGE=4096 and STATIC_RFA, Also, when the file revision is 4r6, isutl -ro now upgrades the file to rev 6 as indicated. [tr#36749]

Synergy HTTP document transport API

-- In 10.3.3 and 10.3.3a, when a proxy was specified, an unknown host error was returned. This has been fixed. [tr#36593]

Synergy windowing API

-- (OpenVMS) In 10.1.1a and later, when using DBLOPT #39 in conjunction with the W_ windowing routines, the W_ routines used a different channel for I/O than the one on which TT: was opened. As a result, when a user pressed the Enter key on a field in a Toolkit input window, the cursor didn’t advance to the next field in the input set. This has been fixed. [tr#36687]

Synergy XML API

-- Certain badly formed XML documents could cause an $ERR_HSIZERR error ("Map outside bounds of field or handle") to occur in the XML parser. This has been fixed. [tr#36613]

Utilities

-- (UNIX) When synbackup creates a memory segment but the backup mode is set to off, the error message when the file is being used by another instance of fconvert, irecovr, and isutl has been changed from "Operation not allowed due to backup mode" to "File in use by another user." [tr#25195]

-- (Windows) In 10.3.3a, the use of traditional Synergy tools by the SPAWN routine from a program started with DBR got an INVOPT error. This has been fixed. [tr#36631]

-- In 10.3.3a, when using either listdbr or listelb with the -i switch to display the version number, the display was incorrect. This has been fixed. [tr#36638]

-- In 10.3.3 and 10.3.3a, dbl2xml incorrectly used the WEB VIEW flag analyzing code for intermediate XML file generation. This has been fixed to honor the LANGUAGE VIEW in the analysis phase. [tr#36834]

REVISION 10.3.3a

Analysis engine

-- An incorrectly formed SET method caused the compiler to crash. This has been fixed. [tr#36525]

-- An incorrectly formed nested BEGIN caused the compiler to crash. This has been fixed. [tr#36526]

-- Dividing by zero in a compile-time expression (for example a1/0 as a type specification) crashed the compiler. This has been fixed. [tr#36527]

Compiler

-- We added support for -qdecargs to dblproto. [tr#36517]

-- Dblproto now honors the DBLMAXERR environment variable. If DBLMAXERR isn’t set, the number of errors that dblproto will generate before abandoning prototype file generation defaults to 20. [tr#36497]

-- Dblproto previously crashed when a field’s type was a%size(field2) and field2’s type was declared as a%size(field3), if field3 was not defined. This has been fixed. [tr#36501]

-- The compiler reported an unexpected TYPPARM warning when using -qrelaxed:paramad on an overloaded routine where one of the overloads exactly matched the call. This has been fixed. [tr#36502]

-- In 10.3.3, the compiler reported an unexpected NFND error when calling a subroutine without an XCALL when SYNDEFNS was set. In addition, default conversions for subroutines in the same source file were sometimes missing (for example, default string to alpha argument conversions). This has been fixed. [tr#36508]

-- Passing an object type, like a string, into an unresolved XCALL or function call could cause unpredictable results at runtime, as no conversions were done on the arguments to match the routine parameter types. The compiler now reports a level 3 W_URCALL warning when objects are passed as argument types to unprototyped routines. Note that this warning won’t appear when -qrelaxed:param is set, as this allows resolution to local external functions. [tr#36515]

-- When displaying the source line for an error from a repository include, the name of the file was not displayed when -qexport (or SLKRUNS=1) was set. This has been fixed. [tr#36519]

-- When a function call had an unresolved cast in front, a BSTMTCH error occurred. The compiler now correctly reports an INVCAST error. [tr#36523]

-- The line number was off by many lines on certain duplicate repository includes. This has been fixed. [tr#36524]

-- We now allow the use of an alpha variable with a format string when qrelaxed:paramad is specified, giving a W_NUMREQ warning. [tr#36530]

-- An INIT of an overlay record seemed to clear out the record after the INIT, giving incorrect results. The compiler now reports a NOTALLOWED error in this case. [tr#36544]

-- Passing a literal integer value to a method that takes a boxed integer no longer gives a runtime error. [tr#36550]

-- (UNIX) The compiler was not importing prototype files with uppercase file extensions. This has been fixed. [tr#36536]

-- In 10.3.3, the compiler reported an unexpected AMBSYM error when SYNDEFNS was set and a source that contained a subroutine and its call (after the routine was prototyped) was compiled. This has been fixed. [tr#36567]

-- In 10.3.3, using a struct field in a conditional operator that targeted a string did not give an error when targeting a version less than 10.3.3a. Now the compiler reports a PLATSUPT error in this situation. [tr#36558]

-- Use of the conditional operator could cause an "Invalid class handle" error. This has been fixed. When using -qrntcompat to target a version earlier than 10.3.3a, a PLATSUPT error is now reported at compile time in situations that previously would have generated the "Invalid class handle" runtime error. A runtime change was made for 10.3.3a to allow this to work properly. If you want this to work, you must target at least 10.3.3a (-qrntcompat=10030301). [tr#36557]

-- Building a traditional Synergy project in Visual Studio with a blank source file correctly reported a NOPROSEC error, but the name of the file causing the error was not included, making it difficult to locate. The name of the first source file is now reported. [tr#36516]

____
Debugger

-- We fixed the debugger to break at a breakpoint set on a FOREACH statement. As a consequence, an extra STEP will occur on BEGIN statements that have DATA statement(s) with object handles. [tr#36529]

-- The debugger now steps to the correct line with SET STEP OVER and a trapped error. In addition, the error number is output in the message when stepping and an error is trapped. [tr#36533]

-- (Windows, UNIX) BREAK <routine> <line#> worked incorrectly if <routine> was not current. This has been fixed so you can set a breakpoint in a routine that is not currently resident. [tr#36532]

____
Runtime

-- Using a tag key on a Join now correctly returns all possible matches when the On expression is not fully optimized. [tr#36492]

-- An RCBSTACK error previously could cause a stack overflow with TRY-CATCH or $ERR_CATCH. This has been fixed. [tr#36499]

-- When attempting to join one or more remote tables using the Select Join, and the leftmost (driving table) file is not an ISAM file, the creation of the Join object no longer seems to hang. [tr#36545]

-- When using automatic channel allocation, we now start the channel scan at the size of the variable (e.g., 999 for d3) to avoid CHNUSE errors. [tr#36534]

-- (.NET) In 10.3.3, a .NET runtime license did not share a seat with a traditional runtime license. This has been fixed. [tr#36568]

-- (.NET) In 10.3.3, setting SCSPREFETCH while running a .NET client to xfServer hung with a Select or READS. This has been fixed. [tr#36543]

-- (.NET) In 10.3.3, a Synergy .NET app running on the desktop concurrently with a traditional Synergy app consumed one additional RUN license. This has been fixed. [tr#36511]

-- (Linux) In 10.3.3, the SSL linkage was broken for httpslib.so, synssllib.so, and VTXSSL.so. This has been fixed. [tr#36569]

-- (SuSE Linux) The Redhat version of the files httpslib.so, synssllib.so, and VTXSSL.so was being installed on SuSE. This has been fixed. [tr#36570]

____

Synergy DBMS

-- Isutl no longer reports a "Bad RFA vector" or "snapshot may be corrupted" error when verifying a file with change tracking when a deleted record is compressed with a time key present, or a record is not compressed but follows a free snapshot (isutl -f), and the record was updated and deleted during the same snapshot prior to being freed. [tr#36431]

____

xfNetLink Synergy

-- In 10.3.3, xfNetLink Synergy failed to connect to an xfServerPlus server using %RX_START_REMOTE with error code RX_NOXFSPL when IPv6 protocol was not available. This has been fixed. [tr#36507]

____

REVISION 10.3.3

____

Compiler

-- We added the -qdefns and -quserdef options in the dbl compiler and dblproto to override SYNDEFNS and SYNUSERDEF, respectively. We also added -qimpdir and now allow both -qrelaxed:<option> and -qrelaxed=<option> syntax in dblproto. [tr#36338]

-- We added the -qrelaxed:paramad option, which allows passing an alpha argument to a d parameter whose direction is in or unspecified. [tr#36328]

-- We added the -qrelaxed:paramst option, which allows a non-CLS structure (without objects) to be passed as an argument to a parameter whose type is another non-CLS structure of the same size. [tr#36394]

-- We modified the compiler switch -qdebug=level, where level is the debug level. Level=1 reduces the debug emission output, while level=2 leaves it unchanged. The -d switch now defaults to level=1. With level=1, field references are only forced for those in the scope of a referenced field’s ancestry. That is, if a referenced field is within group G1 within record R1, all of the fields within the scope of R1 will be referenced (i.e., all the R1 fields, all the fields within the R1 groups, all the fields in THOSE groups, and so forth). Finally, with level=1, the source lines in the data division and its include files are no longer accessible via the debug LIST and VIEW commands. All other source emissions are unchanged. [tr#36412]

-- (Windows, UNIX) The dbl compiler, the dblink linker, and dblproto now support the -qrntcompat=<value> switch, added for compiling to a previous version of the compiler. Valid values are 90501 for 9.5.1, 90503 for 9.5.3, 100101 for 10.1.1, 10010101 for 10.1.1a, 10030100 for 10.3.1, 10030101 for 10.3.1a, 10030102 for 10.3.1b, and 10030103 for 10.3.1c. The linker -qrntcompat option ensures that the version of the object files is not greater than the version specified and that it is compatible with the targeted runtime. Note that object files created prior to this release cannot have their version verified. [tr#34357]

-- (Windows, UNIX) The dbl compiler and dblproto now support the -platform=<type> option, which was added to allow specifying 32-bit or 64-bit object file creation. <Type> can be one of the following values:

x86	32-bit. Use a size of 4 for D_ADDR.
x64	64-bit. Use a size of 8 for D_ADDR.

If -platform is not specified, the size of the D_ADDR remains at its current values (i.e., 8 on 64-bit platforms and 4 on 32-bit platforms). [tr#34357, tr#36244]

-- (Windows, UNIX) The -qdefine option from the Synergy .NET compiler is now also supported by the traditional compiler, dblproto, and dbl2xml. [tr#36249]

-- The -qrelaxed:allowdup option from the Synergy .NET compiler is now also supported by the traditional compiler. If -qrelaxed:allowdup is set, the compiler now prevents TOKUDF errors for duplicate subroutines or functions from prototypes. [tr#36314, tr#36252]

-- Dblproto now sets the following -qrelaxed options under the hood: allowdup, deprecate, interop, local, param, path. Additionally, you can set the -qrelaxed options end and extf on the dblproto command line. (Note that setting -qrelaxed=end is the same as the dblproto -qrelaxedend option.) [tr#36312]

-- You can now specify more than one namespace in SYNDEFNS (separated by semicolons) whose prototypes will be implicitly imported. In addition, you can import nested namespaces by adding a wildcard character (*) after the namespace (for example, SET SYNDEFNS=synglobal;mydefaultnamespace.*). We recommend using this format rather than SYNUSERDEF, which has been deprecated. [tr#36264]

-- Dblproto now creates a single .dbp file containing prototypes for all subroutines, functions, and classes rather than a file for each routine/class. Although it improves and greatly simplifies prototyping, THIS CHANGE MAY BREAK YOUR CODE, as well as highlight coding errors with your imports and cause TOKUDF errors due to duplicate prototypes. You must regenerate existing prototypes or you will get an error. Be sure to delete the old .dbp files first.

- The dblproto -out option now specifies a filename rather than a namespace. If not specified, the filename defaults to the first source file name on the command line. If the filename does not end in .dbp, a .dbp extension will be added.

- The dblproto -single option has been deprecated and should not be used. We recommend that you use wildcards or a complete list of source files as input to dblproto. Some customers used the -single option to improve processing time and decrease memory usage, especially on 32-bit machines. You can now use the -platform=type option (see tr#34357 and tr#36244) to target 32-bit from the 64-bit dblproto, making this use of the -single option unnecessary. If you feel you still have a need for -single, please contact Synergex Developer Support.

- Due to corrected error reporting, you may find some files now correctly report an IMPORT error ("Namespace myNamespace not found"). This either means that there’s a prototype import or generation ordering issue that needs to be fixed or the import is actually incorrect.

- Dblproto and dbl compiler performance has been significantly improved when using prototypes in conjunction with wildcard input files to dblproto.

[tr#36331]

-- The IMPORT statement no longer supports a DIRECTORY specification. If included, it will be ignored and generate a warning. Instead, specify the location of prototypes with -qimpdir when you compile or by setting the SYNIMPDIR environment variable. THIS CHANGE MAY BREAK YOUR CODE. This change was a result of a shift in the function of the IMPORT statement. Now it is used solely to enable you to use members of a namespace without having to fully qualify the namespace. IMPORT no longer causes checking of prototypes. Prototype checking now takes place automatically if you have generated prototypes and specified their location with -qimpdir or SYNIMPDIR.

As a result of this change, the -impdir option for the gennet40 utility has been deprecated.

[tr#36331]

-- (OpenVMS Itanium) Dblproto now puts private members into the .dbp file to prevent the I64/OpenVMS linker from generating an INVOVRINI error ("Incompatible multiple initializations for overlaid section"). This was a regression introduced in 10.3.1. [tr#36327]

-- Previously dblproto would not generate any prototypes for the compilation unit if an error was encountered. Instead it now reports errors but goes ahead and generates the corresponding prototypes. Note that if errors are reported, you must fix them, or you will potentially face CRC mismatch errors when linking. [tr#36271]

-- (Windows) TOKUDF errors now provide the location of the original definition of a duplicate symbol as an informational message following the error message. [tr#36073]

-- Unexpected AMBSYM errors previously occurred from imports due to concatenated source file functionality. Therefore, the compiler no longer concatenates files by default. To provide the old concatenating functionality, we added a -qconcat command line option (/CONCAT on VMS). [tr#36389]

-- A string variable passed for a non-string, writable parameter will now always generate an INVPARM error. This replaces a similar error that was previously only issued for .NET compatible compilations. Also, if a runtime-generated, read-only parameter is passed for one of these parameters, a WRTLIT error will be generated on any modification attempt. [tr#36289]

-- If an array used in an INIT has initial values, the first element is no longer simply replicated, and the entire array is directly loaded. In this case, the maximum size of the referenced array is 32767. A NOTALLOWED compiler error will be generated for the INIT if this is not the case. [tr#36008]

-- Previously the compiler crashed when it was running out of disk space while writing to the .dbo file. This has been fixed. [tr#36176]

-- Comparing a boxed boolean variable to an alpha using the .IS. operator incorrectly returned "true." The .IS. operator has been fixed to map boolean to integer. [tr#36201]

-- Prototyping a file with a duplicate structure with one or more unnamed fields caused an unexpected PROTOMISMC compilation error later. This has been fixed. [tr#36318]

-- In 10.3.1, public class records containing fields marked private would not prototype correctly. This has been fixed. [tr#36323]

-- The compiler previously reported an unexpected NFND error on an XCALL of a subroutine (not defined locally) within the nested scope of an overloaded method. This has been fixed. [tr#36367]

-- The compiler no longer reports an unexpected AMBSYM error for structure resolution. [tr#36386]

-- We changed the way namespaces are imported within a compilation unit. Previously, any namespace defined in a compilation unit was available to all source files in that compilation unit. In addition, a namespace that was defined outside the compilation unit and then imported with the IMPORT statement was available to all subsequent source files on the dbl command line. Now, a namespace defined in a file is available only to that file unless it is explicitly imported into other files in the compilation unit (with IMPORT) or implicitly imported with SYNDEFNS. In addition, a namespace imported with the IMPORT statement applies only to the file into which it is imported. THIS CHANGE MAY BREAK YOUR CODE. If your code relies on the old behavior, you will now get compilation errors (NFND, NVTF, NVTP, or NVTPR). To correct this, you must add IMPORT statements to your code (or redefine SYNDEFNS). We made this change so that Synergy behavior matches other .NET languages, and it enables the compiler to resolve symbols that would otherwise be ambiguous and would prevent IntelliSense and QuickWatch features in Visual Studio from working. [tr#36395]

-- (Windows) You can now use .IFDEF and .IFNDEF for conditional compilation based on a global enumeration with a particular name. [tr#36072]

-- (Windows) For TOKUDF errors, the location of the original definition of the duplicate symbol is now provided in the information that follows the error message. [tr#36073]

-- Previously, the compiler incorrectly reported an RFAERR when an arrayed parameter was used as an RFA value and when it encountered ^M and function calls. These issues have been fixed. [tr#36174]

-- You can now use duplicated structures when prototyping or when -qrelaxed:allowdup is specified in the "Other options" field on the Compile page of project designer. [tr#36254]

-- Declaring a field of a structure type whose name is the same as a local group no longer causes an AMBSYM error. [tr#36258]

-- Previously, if a field defined using ^SIZE referenced another field in the same structure, the compiler reported an unexpected AMBSYM error when that structure field name was repeated within another structure in the same scope. This has been fixed. The nearest scoped field is now chosen in this situation. [tr#36260]

-- The compiler now handles implicit closures correctly (e.g., for an "ENDSUBROUTINE expected" warning) when it encounters IMPORT, START, and global STRUCTURE statements. [tr#36326]

-- The compiler no longer crashes if a function is used on the left side of an assignment. The compiler now generates an error for this situation. [tr#36388]

-- (Windows, UNIX) -qimpdir="quoted string" did not work from a redirected input file. This has been fixed. If you have multiple paths, either quote the entire string or use multiple -qimpdir commands. [tr#36401]

-- Previously, using a type from the same namespace within another source reported an unexpected NVTF error ("Cannot resolve type for field"). Now, for namespaces declared locally within a source, the compiler automatically imports prototypes from that namespace. [tr#36411]

-- The compiler now reports a W_TYPPARM level 3 warning when passing an argument is satisfied using either the -qrelaxed:paramad or -qrelaxed:paramst option. [tr#36417]

-- Previously, dynamic [#] arrays of an enum type were incorrectly set to i1 rather than i4 values. This has been fixed. [tr#36419]

-- The compiler no longer reports an unexpected NOTCEXP error ("Not a compile-time expression") when using an enumeration value from another source compiled later as the initial value of an enumeration field in an earlier source. [tr#36415]

-- Dblproto no longer crashes if a problem such as an invalid name or a protection issues occurs when attempting to rename the temporary file to the final .dbp filename. [tr#36421]

-- Previously, a runtime error occurred when a simple property was a member of a class. This has been fixed. [tr#36420]

-- An error on the first compiled line following uncompiled lines in an .INCLUDE file was previously reported on the wrong line. This has been fixed. [tr#36375]

-- Previously, if the last data field in a section was not a handle, the compiler failed to report a LBLSCOPE error ("Label %s out of scope"). This has been fixed. THIS CHANGE MAY BREAK YOUR CODE. See LBLSCOPE in the Error Messages chapter of the Tools manual for more information about what causes this error.

[tr#36429]

-- Previously, using .NOPROTO around a non-private method within a class caused an unexpected PROTOMISMCH error when compiling source against its prototype. This has been fixed. [tr#36430]

-- The -qrelaxed:EXTF option did not work with prototypes. This has been fixed so that an error is no longer reported. [tr#36391]

-- An unexpected PROTOMISMCH error previously occurred when using a prototype of a class whose member used a dynamic array of type short that was preceded by a fixed array of type short. This has been fixed; dblproto no longer generates an incorrect prototype. [tr#36444]

-- Related classes and strings could report unexpected "Type mismatch between %s and %s on each side of the ':' in a ternary operator" errors (TERNMSMCH). This has been fixed. [tr#36459]

-- An unexpected NOTCALLABLE error occurred when calling a string function without function call syntax. This has been fixed. [tr#36315]

-- A define with a value wasn't getting applied from -qdefine when the define was second or later in the list. For example, -qdefine=DOFLD1,F2TYPE=a10,DOFLD3 didn't apply F2TYPE as a10. This has been fixed. [tr#36467]

-- Assigning a boxed alpha to a boxed alpha field within a dynamic array of structures returned from a property or method previously caused a segmentation fault in the traditional compiler. The compiler now reports an "Invalid use of type" error (INVTYP) for properties and methods that return an array of a structure that has handle in it. [tr#36486]

-- A structure with a boxed alpha private field previously caused an unexpected PROTOMISMCH error when the source from which the prototype was derived was compiled. This has been fixed. [tr#36487]

-- Previously, a local data field whose type was a fixed array of structures did not apply the structure's initial values. This has been fixed. [tr#36489]

Debugger

-- After setting a break point in the calling routine, a GO command will now break at the break point when it is reached. [tr#36481]

Installation

-- (Windows 10) After uninstalling xfNetLink .NET and xfNetLink Java, environment variable settings lingered until the user logged out and logged back in. This has been fixed: environment variables are now "refreshed" after installation and uninstallation. [tr#36074]

Runtime

-- We made several enhancements to Select and related classes to enable you to create simple joins in a way that is similar to

SQL92 join syntax. See "Joining data from multiple sources" in the Discussion for Synergex.SynergyDE.Select.Select in your Synergy Language Reference Manual for an overview, as well as the individual syntax sections for the following new classes and methods:

Synergex.SynergyDE.Select.Sparse
Synergex.SynergyDE.Select.On
Synergex.SynergyDE.Select.JoinSelect
Synergex.SynergyDE.Select.RowEnumerator
Synergex.SynergyDE.Select.Rows classes
Synergex.SynergyDE.Select.Select (constructors and Join method)
Synergex.SynergyDE.Select.From (InnerJoin and LeftJoin methods)

[tr#35559]

-- Join allows the use of one or more "tag" fields on the join of an inner table within the On expression. This tag field is a field of any type or size (compared against a value that doesn’t come from the outer table) that when combined with one or more join fields (compared against a value that comes from the outer table) matches a complete key or contiguous part/segment (starting from the beginning of the key) defined in the inner table. In addition to tag fields, multiple join fields specified in the On expression are now combined (if they match a defined key on the inner table), thus optimizing the effectiveness of the join. [tr#36468]

-- We improved the performance of concurrent read operations when the same ISAM file is accessed across threads and processes. In testing, we have observed between 10% and 50% improvement with some configurations, with even greater improvements when using SSD drives. [tr#36276]

-- (.NET) We now support .NET native release configurations for UWP. [tr#36259]

-- To support large XML documents, we added %HANDLE_FROM_STRING to enable you to create a memory handle from a String object:

```
   <hnd> = %handle_from_string(<str> [, DM_STATIC])
```

and %STRING_FROM_HANDLE to enable you to create a String object from a memory handle:

```
   <str> = %string_from_handle(<hnd>)
```

[tr#34311]

-- (.NET) We now default to an xfServer connection per AppDomain without requiring S_SERVER_THREAD_INIT. We also added the new S_SERVER_THREAD_SHUTDOWN routine. If a program uses S_SERVER_THREAD_INIT, it should now explicitly call S_SERVER_THREAD_SHUTDOWN() to ensure the xfServer connection is closed. [tr#34798]

-- In some cases, the Select class caused a segmentation fault or returned partial results when a Where expression contained one or more OR expressions. This has been fixed. [tr#36127]

-- We corrected an issue that caused the index for a Rev 6 ISAM file to become corrupt after a DELETE or WRITE. A file would be susceptible to this if it had been configured with TBYTE explicitly or implicitly (e.g., with change tracking) and it had a PAGE setting smaller than 2048. [tr#36130]

-- Parentheses are now used correctly in Select.QueryString values. [tr#36143]

-- Attempting to create an ISAM file from an XDL file (using bldism or fconvert) when specifying a key as an autokey (SEQUENCE, TIMESTAMP, or CREATESTAMP) caused a BADKEY error ("AUTOKEYS cannot be segmented") with bldism or an "Invalid XDL/FDL file" error with fconvert if the last key definition was not the autokey but was segmented. This has been fixed. [tr#36144]

-- (Windows, UNIX) Previously, EOF was not trapped for an ACCEPT statement with an I/O error list. This has been fixed. [tr#36157]

-- (Windows, UNIX) Calling READS after opening a pipe with the O mode for OPEN no longer causes a "signal trap/segmentation violation" error. The FILOPT error is now generated in this situation. [tr#36168]

-- Changing the Select locking behavior by using Q_MANUAL_LOCK in the From constructor no longer causes non-matching records to remain locked. [tr#36197]

-- Previously, a Select with Where.In(key1, val1, val2, ...), where the key and the values were alpha, failed to return matches for values with fewer than the maximum number of digits for the key (e.g., "   123"). This has been fixed. [tr#36227]

-- (Windows, UNIX) MERGE now correctly combines remote files. Previously, the first file specified for a MERGE was used, but other remote files for the MERGE were omitted. (For each of these other files, the MERGE added another copy of the first file to the output.) [tr#36291]

-- If the SQL Connection function %SSC_MOVE was called for a cursor in an error state, a segmentation fault sometimes occurred. This has been fixed. An E_TOOFEWVAR error is now reported. [tr#36346]

-- In versions 10.1.1 through 10.3.1b, the runtime leaked memory if an OPEN statement failed with certain errors but the file existed. This has been fixed. [tr#36424]

-- In rare cases, opening an ELB in a STOP-chained program caused a segmentation fault. This has been fixed. [tr#36438]

-- The use of a structfield in local data (DATA statements following BEGIN) where the structure contains one or more object handles no longer unintentionally overwrites data, which sometimes resulted in a segmentation violation at runtime. [tr#36488]

Synergy DBMS

-- We added a Sparse class to the Synergex.SynergyDE.Select namespace that takes a variable number of field arguments and whose objects can be dynamically combined using the .AND. operator. Also, the Select constructors now have an additional overload that includes the Sparse class object as its first argument. This is functionally equivalent to, and an alternative to, the Select.SparseRecord() method, which remains for backwards compatibility. [tr#35021]

-- The fconvert -v option (when <count> is not specified) no longer defaults to 1 but now automatically throttles the display count by a factor of 10 up to 100,000 records. Due to terminal I/O speed, this will significantly improve performance, especially on large files, when using the -v option. [tr#36214]

-- The fconvert utility no longer ignores NOTBYTE, NOROLLBACK, SGRFA, and ERASE_ON_DELETE file options when specified in a PAR file (using -d PAR file). In addition, specifying an option that takes an argument as the last file option in a PAR file no longer causes a Segmentation Violation. Also, TRACK_CHANGES and SGRFA no longer expect an argument. [tr#36131]

-- (UNIX) The fconvert utility no longer generates an "Invalid par file - Bad file option" error when loading an ISAM file using a valid PAR file copied from Windows with CR/LF line terminators. [tr#36453]

-- Read operations (READ/READS) no longer wait for the completion of isutl -v. [tr#36275]

-- (Windows, Linux) The performance of the chklock utility has been improved when accessing large files. [tr#36341]

-- Fcompare crashed when a structure lacked access keys, and the data file had errors. This has been fixed. [tr#36380]

Synergy HTTP document transport API

-- We added support for IPv6. %HTTP_SERVER_CREATE has a new optional argument, <IP_flag>, that determines which IP versions are allowed. Pass 0 to allow IPv4 only; pass any nonzero value to allow IPv4 or IPv6. If <IP_flag> is not passed, it’s equivalent to passing 0. [tr#35565]

-- (OpenVMS) The minimum supported version of OpenSSL for the HTTP document transport API and DATA_ENCRYPT and DATA_DECRYPT routines is now SSL 1.4. If your OS version is 8.4 (either Alpha or I64), the default version is now the more secure SSL1 1.0-2C. For earlier versions of the OS, which do not support SSL1 1.0-2C, SSL 1.4 will be installed as the default. THIS CHANGE MAY BREAK YOUR CODE. See "OpenSSL Requirements" in the "Requirements and Considerations" chapter of the Installation Configuration Guide and Synergex KnowledgeBase article 100001979. [tr#36282]

Synergy socket API

-- We added new routines to the API to support IPv6. If your application will be connecting to an IPv6 socket, use the new SS2_ routines instead of the corresponding SS_ routines. See the "Synergy Socket API" chapter of the Synergy DBL Language Reference Manual for details. [tr#35565]

Utilities

-- We added two new options, -v4 and -v6, to synxfpng for specifying an IPv4 or IPv6 address. By default, synxfpng looks first for an IPv4-formatted address and then for IPv6. Use the command line options to restrict look-up to only one or the other.

[tr#35565]

-- We fixed a problem that was causing synxfpng to crash when the host name was invalid. Now you’ll get a "host name not found" error. [tr#36233]

____

xfNetLink Synergy

-- We added support for IPv6. You do not need to make any changes to your Synergy client for IPv6 unless you use the %RX_DEBUG_INIT routine (for running in debug mode), in which case you will need to pass the <listen_ip> argument as an a16. [tr#35565]

____

REVISION 10.3.1b

____

Compiler

-- When trying to make a structure into a typed box, an INVCAST error was previously reported. Boxing structures as alpha types now works correctly. [tr#34560]

-- The compiler now supports INIT on local data fields. Previously, using INIT on a variable defined with the DATA statement caused the compiler to crash. [tr#35957]

-- In 10.3.1 and 10.3.1a, the compiler reported an unexpected PROTOMISM error on a prototyped class with a private constructor. This has been corrected. [tr#36023]

-- If the accessibility of an overridden property method differs from the accessibility of the base method, the compiler now reports an E_BADACCESS error. Previously, the compiler produced code that resulted in a runtime exception for .NET code. [tr#36058]

-- We fixed an issue that caused the compiler to report an invalid E_IVLNG error for a prototyped decimal or implied-decimal field with a negative initial value. [tr#36068]

____

Installation

-- Previously the links in relnotes.html did not work in Firefox. This has been corrected. [tr#35935]

____

Runtime

-- We added support for Windows 10. XCALL ENVRN will return 28 (OS_WINDOWS10) when a program is running on Windows 10. [tr#36089]

-- (Windows, UNIX) We added an optional runtime option, /scl=<level>, to the OPEN statement OPTIONS qualifier to allow an xfServer client to specify the security compliance level, which defines which protocols will be used when encryption is enabled. Valid values for <level> are

0 = use the current Synergy default
1 = use protocols TLS 1.0, TLS 1.1, TLS 1.2 (current default)
2 = use protocols TLS 1.1 and TLS 1.2

SSLv3 and lower protocols are no longer supported.

Additionally, the information returned from GETFA("SLC") now includes the full cipher description: <ciphername> <protocol version> Kx=<key exchange> Au=<authentication> Enc=<symmetric encryption method> Mac=<message authentication code>.

(Note: Confusingly, in this output that comes from OpenSSL, TLS 1.0 ciphers are marked as being SSLv3 protocol, but they are really TLS 1.0.) [tr#35922]

-- (Windows, UNIX) We added an optional timeout parameter to the %SS_CONNECT socket routine that specifies the number of seconds to wait for a connection before the routine returns with an SS_ETIMEDOUT error. If the timeout parameter is not specified, the routine will wait indefinitely for either a successful connection or another error to occur. [tr#35931]

-- (Windows, UNIX) When running in debug mode or when BOUNDS_CHECK_FATAL is set, we now check for global symbol references (either common or global data section) that range past the end of the containing psect. If found, a GBLSIZ fatal error occurs. [tr#35992]

-- (Windows, UNIX) We fixed an issue that could prevent a Select from returning all matching records when there was a type mismatch between a numeric field type specified in the Where condition and the defined key type in the file. As a result of this fix, field/key type mismatches can no longer be optimized unless the runtime can determine from the search criteria that no negative values will be returned. This may result in a noticeable decrease in performance unless the type mismatch is corrected or the Where expression is changed. For example, if key1 is a decimal field and it matches an alpha key, the Where expression (key1 < val) will not be optimized. However, if the Where expression is changed to (key1 > 0 .and. Key1 < val), it will be optimized if val is greater than 0. Correctly typing key1 as decimal in the file will also result in optimization. [tr#35651]

-- Stack memory attached to RCB arguments or an SQL bind now gets an RCBSTACK or SQLSTACK error on a THROW/CATCH when compiled with -qcheck. Also, if a routine is built with -qrefresh, an RCBREL or SQLREL error will be signaled. [tr#35966]

-- (Windows) We improved the way the winerrpost.log file is generated and updated. Previously, if an error that was logged to the file was encountered in a loop, the file could grow to an enormous size if the error didn’t terminate the process. Furthermore, the information logged in the file wasn’t sufficient to help diagnose errors. These issues have been corrected. [tr#35882]

-- (Windows, UNIX) In versions 9.5.1 through 10.3.1a, a READS on a channel opened in input mode performed poorly if updates or deletes simultaneously occurred on another channel or by another process. This has been corrected. [tr#35936]

-- We made several improvements to the CurrentRFA method for the Select.AlphaEnumerator and Select.RestrictedAlphaEnumerator classes. This method now returns the correct RFA from a remote server with SCSPREFETCH, and it now returns the correct GRFA when a From class record is smaller than the actual ISAM file record.

Additionally, if the "Stored GRFA" attribute of an ISAM file is configured, CurrentRFA now uses this attribute, which reduces CRC calculation overhead for each enumeration. [tr#35915]

-- (Windows, UNIX) The informational error "%DBR-I-OPWFND, Operation was FIND" was reported multiple times for an untrapped I/O error traceback on a Select channel. This has been corrected so that this informational error is no longer reported in this situation. [tr#35895]

-- (Windows, UNIX) In versions 10.1.1 to 10.3.1a, a Select with a Where expression that specified multiple adjacent segments failed to return all records that matched the criteria if it used EQ or EQS for the first segment and GE, GES, LE, or LES for the second and third segments. This has been corrected. [tr#35874]

-- Select now optimizes Where expressions regardless of the way comparisons are ordered. Previously, (Where)(0 < key) would not be optimized, even though (Where)(key > 0) was optimized. Additionally, the order of GE, GES, LE, and LES comparisons would sometimes prevent optimization for segmented keys when multiple adjacent segments were compared. These issues have been corrected. [tr#35871, tr#35623]

-- An "Invalid operation" error (INVOPER) is now reported in every case where a READS, FIND(MATCH:Q_SEQ), or WRITES follows a Select, unless context has been reestablished with a READ, FIND, or WRITE statement. (Until READ, FIND, or WRITE is called in this situation, sequential context is undefined.) Previously, there was one case where this error was not reported: after using Dispose with a Select in a FOREACH statement. This has been corrected, but note that THIS CORRECTION MAY BREAK YOUR CODE. Previously, the runtime would allow your program to call READS, FIND(MATCH:Q_SEQ), or WRITES without reestablishing context after a FOREACH Select was Disposed (even though the results were undefined). Now it will report the error. [tr#35907]

-- (Windows, UNIX) Long exception messages caught by TRY-CATCH are now displayed in full, rather than truncated with the message "*** Warning: Message replacement too long ***." [tr#36045]

-- An exception explicitly thrown or rethrown that is unhandled or covered by an $ERR_CATCH when there’s a TRY with a FINALLY somewhere within scope of the calling chain no longer generates a Signal Trap. [tr#36044]

-- (Linux) In 10.3.1 and 10.3.1a, secure connections failed with "OPENSSL1_0_0 not found" on Redhat and Fedora Linux. This has been corrected. [tr#36066]

Synergy Configuration Program (SynConfig)

-- SynConfig now supports setting the Security Compliance Level. [tr#35953]

Synergy DBMS

-- The isload CLEAR option now supports records larger than 20K. [tr#35970]

-- All output from fconvert (informational and error messages) now goes to STDOUT instead of STDERR. This allows direct use of the fconvert command through an OPEN pipe. [tr#36060]

-- We improved the performance of isutl -f (fastload) and fconvert -ai (append) when loading an existing ISAM file that is empty. [tr#35877]

-- Various issues with ISAM files configured with multiple fixed-length records (i.e., record type "multiple") have been resolved. Isutl -rs (add static RFA) no longer fails, and isutl -rc (add compression) is no longer allowed, as that configuration is not permitted. Also, change tracking is not allowed with multiple fixed-length files and now generates an error if attempted. [tr#36015]

Synergy HTTP document transport API

-- The default protocol is now set to TLS1.0/TLS1.1/TLS1.2 for OpenSSL 1.0.1 and above, and TLS1.0 for OpenSSL versions below 1.0.1. [tr#35823]

-- We added a new encrypting rule set, HTTPDATA_RULES, to %SYN_ESCAPE_HANDLE and %SYN_UNESCAPE_HANDLE that encrypts data according to HTTP RFC 3986. This enables you to encrypt data parts of the URI without affecting unreserved characters in the URI. [tr#35990]

-- Previously, once HTTP_NOESCAPE was set it couldn’t be unset, which prevented automatic escaping from being turned back on. This has been corrected. [tr#35951]

-- In version 10.1.1 on Windows or 10.3.1 and 10.3.1a on UNIX and OpenVMS, running against OpenSSL version 1.0.1n or greater (or a functionally equivalent patch for operating systems that provides OpenSSL) caused the HTTP document transport API to give a "dh key too small" error when using default Diffie-Hellman keys. This has been corrected.

Synergy .NET assembly API

-- Large strings were previously truncated at 65536 bytes on 32-bit systems when returned from the .NET assembly API. This has been corrected. [tr#36052]

Utilities

-- An open error previously occurred with listelb and listdbr when the ELB or DBR file had linked ELBs and the translated environment variables exceeded 32 characters. This has been corrected. [tr#35968]

xfNetLink Synergy

-- We made a number of enhancements to encryption. We added an optional parameter, SCL, to %RX_START_REMOTE and %RX_DEBUG_START to allow you to override the default security compliance level, which determines which protocols will be used for encrypted communication. Valid values for the SCL parameter are

0 = use the current Synergy default

1 = use protocols TLS 1.0, TLS 1.1, TLS 1.2 (current default)
2 = use protocols TLS 1.1 and TLS 1.2

Note that setting 0 means that the supported protocols could change when you upgrade to a new version of Synergy.

SSLv3 is no longer supported.

The security compliance level is also set on the xfServerPlus machine. The setting there should match the one on the xfNetLink machine.

In addition, when there is a protocol handshaking error and secure communication cannot be established between xfNetLink and xfServerPlus, the server closes the socket, logs an error, and RX_SSLCONNECTERR (12323) is returned on the client. Previously, this caused the client to hang. [tr#35923]

REVISION 10.3.1a

Compiler

-- A new global define, _DEBUG, is now supported for compiling with the debug option. This enables you to conditionally include or exclude code based on whether or not the program is compiled in debug mode. [tr#35724]

-- Compiling two sources with separate command lines that both used a built-in enumeration (such as Select’s CTState) caused a SegVio. This has been fixed. [tr#35688]

-- Previously, an INVOPT error occurred when dbl was used in a .proj file with msbuild. This has been fixed. [tr#35421]

-- We made some minor corrections to the compiler when using -qnet to track changes made to the .NET compiler. [tr#35703]

-- A globally declared enumeration followed by an implicit main did not correctly recognize the implicit main. This has been fixed. [tr#35756]

-- A field initializer incorrectly reported an INVINITMBR error if the assignment statement that contained it was the first line in the routine. This has been fixed. [tr#35807]

Installation

-- The Synergy/DE Windows installations now include VC++ merge modules. These merge modules were not included in Synergy/DE 10.3.1 distributions, which prevented Synergy projects from opening in Visual Studio on 64-bit systems that had only Visual Studio 2012 installed. Note: To use Synergy DBL Integration on a 64-bit system that has Synergy/DE 10.3.1 and Visual Studio 2012 only, you must either upgrade 64-bit Synergy/DE to 10.3.1a or download vcredist_x64.exe from http://www.microsoft.com/en-us/download/details.aspx?id=40784 and run it on the 64-bit system. [tr#35780]

Linker and librarian

-- Previously, an INVOPT error occurred when dblink and dblibr were used in a .proj file with msbuild. This has been fixed. [tr#35421]

Runtime

-- We added an %ATRIMTOSTRING() function to reduce overhead when creating a string from trimmed alpha variables. [tr#35609]

-- (.NET) We added support for thread-specific xfServer connections for Mono platforms. [tr#35591]

-- When buffering records stored to an ISAM file (using the BUFSTORE OPEN option) and also doing a sequential READS on the same channel, a certain condition sometimes caused a record to be returned out of sequence. Specifying LOCK:Q_NO_TLOCK or LOCK:Q_NO_LOCK increased the likelihood that this would occur. This has been fixed. Also, SCSPREFETCH (input caching) is no longer enabled implicitly on a channel opened with /BUFSTORE. [tr#35568]

-- When a program used IO_HOOKS and was compiled prior to 10.3.1 but run with 10.3.1 or later, a memory leak occurred. This has been fixed. [tr#35579]

-- In 10.3.1, a Select didn’t return any records when the Where expression contained an OR operator with terms that referenced a field contained within a defined key, but the offset of the field didn’t start at the beginning of the key. This has been fixed. [tr#35595]

-- In versions 10.1.1b through 10.3.1, a RTNNF error ("Cannot access external routine") occurred (with no filename) when loading a method when the MAXMEM value was exceeded and the routine being loaded referenced a crushed method name. This has been fixed. [tr#35641]

-- Converting from descriptor types to objects is now prohibited in conditional operators in traditional Synergy, as this could cause an invalid class handle error at runtime. A PLATSUPT compiler error is now reported in this situation. [tr#35679]

-- When using output I/O buffering (/BUFSTORE or OUTPUT sequential WRITES) with xfServer Connection Recovery, and the STORE or WRITES operation was retried with a successful reconnection to the original context using either reconnect_hook() or a retry of the STORE/WRITES on an $ERR_SRVCONRTY error, some records that should have been written to the file were not, due to the cache buffer not being flushed to disk. This has been fixed. [tr#35644]

-- A Select on a non-ISAM file using Orderby.Ascending(fld) or Orderby.Descending(fld) previously skipped the first record when the first record in the ordered selection was also the first record in the file. This has been fixed. [tr#35738]

-- On a big-endian machine, when passing an argument by value to a routine where the ^VAL argument has a specified size of less than the size of an address on that platform, the correct value (instead of 0) is now accessed. [tr#35801]

-- (.NET) We fixed a small memory leak caused by the Select class. [tr#35764]

-- (.NET) We fixed an issue that caused a System.OverflowException ("Arithmetic operand exceeds maximum size") to occur when storing an i8 value greater than 2147483647 on a 32-bit platform. [tr#35767]

-- (.NET) Converting a decimal 18446744073709551615 (2^64-1) to an unsigned long no longer causes a System.OverflowException. [tr#35772]

-- (.NET) We improved the performance of DM_ALLOC for both %MEM_PROC and the use of temporary descriptor types by eliminating the initial value (spaces) for these operations. Consequently, DM_ALLOC is now explicitly uninitialized, as documented. (In previous versions it defaulted to DM_BLANK.) [tr#35782]

-- (.NET) We fixed several issues that had the potential to cause threading problems for Select.OrderBy, record locking for the WAIT I/O statement qualifier, and concurrent opens using the secure sequential or caching open options. [tr#35652]

-- (.NET) Previously, the Select class returned the incorrect number of records in certain cases because the system garbage collector prematurely released Where condition data. This has been fixed. [tr#35695]

-- (.NET) Previously, if an application used multi-threaded access to xfServer with connection recovery but without S_SERVER_THREAD_INIT, an incorrect error code was sometimes returned or the application could lock up. This has been fixed. [tr#35636]

-- (.NET) We fixed a memory leak that occurred if you created an AlphaEnumerator, used From (for the Select class) on the same channel, and then closed the channel. [tr#35663]

-- (.NET) Operations that assign unsigned integers (ushort or uint) to decimals now result in positive numbers. [tr#35658]

-- (.NET) If an ISAM file reaches a limit set with SIZE_LIMIT or RECORD_LIMIT, FileFullException is now thrown. (Previously, SynException was thrown.) [tr#35637]

-- (.NET) Previously, the negative sign (-) was not shown for negative values if you used ToString or an implicit string cast from an implied decimal with 0 decimal places. This has been fixed. [tr#35606]

-- (.NET) Previously, an unhandled exception would in rare cases cause the runtime to hang on shutdown. For example, this could happen if you pressed CTRL+C while xfServer was reconnecting. This has been fixed. [tr#35583]

-- (.NET) Previously, a thread shutdown would sometimes cause a .NET Framework segmentation fault after an I/O operation where the channel that was last used had been closed on another thread. This has been fixed. [tr#35575]

-- (.NET) Characters in the unsigned byte range (128-255) now work on Mono platforms when converting to/from a string. [tr#35570]

-- (.NET) XCALL SETDFN did not support PATHs/logicals or work with XCALL ISAMC. This has been fixed. [tr#35664]

-- (.NET) If you used the Select class for a segmented key where the order of the segments differed, some matching records would not be returned. This has been fixed. [tr#35612]

-- (.NET) Previously, no records would be returned if you used the Select class with an order that was different than the specified key and the comparison spanned more than one segment. This has been fixed. [tr#35573]

-- (.NET) We removed support for MonoTouch, which is no longer supported by Apple. [tr#35667]

-- (.NET) On Android and 32-bit iOS, using %DATE with an i8 variable caused an arithmetic overflow exception. This has been fixed. [tr#35654]

Synergy DBMS

-- We added a new file creation option to the ISAMC routine, bldism, and XDL keyword files. If ERASE_ON_DELETE is specified at file creation, when a record is deleted (or moved internally by ISAM), it will be erased (i.e., nulled out). Without this flag enabled, records that are deleted are only marked deleted; their contents remain in the file until the space is reused, which has been the behavior until now. ERASE_ON_DELETE cannot be used with change tracking. [tr#35481]

-- We removed an unsupported system call to allow the Synergy runtime to run with limited support on Red Hat Enterprise Linux 5. [tr#35712]

-- When isutl -f (fastload) is used without the -k option (counted input file), the operation is aborted if the ISAM file is configured with one or more integer or auto keys, and a message is displayed to indicate that a counted file must be used. [tr#35616]

-- When running isutl -f (fastload) to a file containing autokeys, the autokeys values were not being preserved (regardless of whether or not the -ak option was specified). This has been fixed. [tr#35616]

-- By default (if the -r "input record size" option is NOT specified), when using fconvert with a relative file for input, the OPEN scans the first record to determine the record size (see OPEN option RECSIZ=-1). If the first record contains binary data, it’s possible the open may encounter a premature line terminator or EOF condition. The -r "input record size" specification now correctly overrides the OPEN RECSIZ scan and ignores binary data in the first record. [tr#35805]

Synergy HTTP document transport API

-- OpenSSL version 1.0.1 added support for protocols TLS 1.1 and TLS 1.2, and by default, the HTTP document transport API supports them as well. We’ve also added support to turn off these two new protocols, if desired, by not specifying them as desired protocols in the API. In addition, we added two new defines, SSLVER_TLS1_1 and SSLVER_TLS1_2, to SSLVER_ALL. [tr#35794]

-- The %HTTP_GET, %HTTP_PUT, %HTTP_POST, and %HTTP_DELETE routines previously did not return headers when an error occurred. This has been fixed. [tr#35656]

Utilities

-- Previously, an INVOPT error occurred when dblproto and dbl2xml were used in a .proj file with msbuild. This has been fixed. [tr#35421]

REVISION 10.3.1

Compiler

-- We added support for simple properties. [tr#34689]

-- We added support for STATIC classes. [tr#34691]

-- We added support for READONLY STATIC class fields. [tr#34690]

-- The following operators are now supported for enumerations: bitwise OR (.BXOR.), bitwise NOT AND (.BNAND.), and bitwise NOT (.BNOT.). [tr#34441]

-- We reduced prototype dependencies of private members. Prototypes now don’t include private methods, private properties, and private classes. Due to runtime restrictions, private data members must be loaded (records, groups, fields, structs); however, if a field is marked as private and its type is a handle or array of handles, it is represented as object or array of object in the prototype (to reduce dependencies). The version number and format of the prototype files require that prototypes be rebuilt or a PRTLOAD error will occur. [tr#34308]

-- If the maximum number of files is exceeded when using a command file to redirect input, a MAXIF error is generated. [tr#34444]

-- If a local data field or catch variable hides another local field or catch variable, a level 3 W_HIDEW2 warning is now reported. A level 4 warning was reported in previous versions. [tr#35012]

-- Using an implied-decimal initial value of length 33 or greater could previously crash the compiler. This has been fixed. [tr#35485]

-- Initial values on class fields of type float, decimal, and implied decimal coming from a prototype file were not loaded correctly, causing PROTOMISMCH errors. This has been fixed. [tr#35282]

-- (UNIX) You can now unbox an object variable to a Synergy fixed array ([*]a). Previously, the compiler reported an invalid INVEXPR error. [tr#34627]

-- (UNIX) An "invalid INVNUMDIM" error was reported if ^M was used with an array that had no element number and that was part of a += statement. This has been fixed. [tr#35064]

-- (UNIX) The compiler no longer crashes if NEW is used with a type whose name differs only in case from the name of a local parameter of that type. [tr#35145]

-- (UNIX) In version 10.1.1c, the compiler did not report an

INITVERR error as expected when it encountered a mismatched data type for an initial DATA statement value. This has been fixed. [tr#35184]

-- (UNIX) Previously, the compiler did not ensure that required parameters were passed when a system method was called without qualifiers. This has been fixed. [tr#35185]

-- (UNIX) The length method of a property that returned an array of strings gave an incorrect runtime result. This has been fixed. [tr#35229]

Installation

-- Framework CLR 4.0.30319 or higher is required in order to install Synergy. [tr#34965]

-- Synergy installations require at least Service Pack 1 on Windows 7 and Windows Server 2008 R2. [tr#34849]

-- We added experimental support for Mono on 64-bit Linux systems. If a system has Mono, you should add synrnt.dll (included with your distribution) to the global assembly cache. [tr#35292]

-- Dbs services created by dbssvc.exe are now affected by the installation package. On uninstall, all dbssvc services will be removed. On upgrade, all dbssvc services are shut down and restarted after the upgrade is complete. [tr#25302]

-- (UNIX) During the execution of install.sde, choosing to skip a chosen but non-existent installation package such as lm.gz by typing "s" at the prompt has been fixed to properly skip the package and complete the installation of the other selected packages. [tr#34502]

-- Previously, the installation could fail if the license server was selected but no licensee name was provided. This has been fixed. [tr#35106]

Runtime

-- (Windows, UNIX) We added a new ChangeTracking class to retrieve the current change tracking information from an SDMS ISAM file. The class also contains the ChangeTrackingEnabled property, which returns whether or not change tracking is enabled on the file, and the ApplySnapshot method, which applies another snapshot to the file. [tr#34234, tr#34553]

-- We added the String.Replace(str,str) method. [tr#34320]

-- (.NET) We added initial iOS runtime support. [tr#34709]

-- (.NET) The debugger’s SHOW CHANNELS command now shows the open mode. [tr#34636]

-- (Windows, UNIX) We added ISCONFIG, a subroutine that enables you to modify the user-defined text in an ISAM file’s data header. This routine has the following syntax where <channel> is the channel on which the file is open, "TEXT" indicates that the change is for the file’s user-defined text (this is the only option currently available for this argument), and <text_string> is the new text for the file:

```
   ISCONFIG(<channel>, "TEXT", <text_string>)
```

Note that if <text_string> is longer than the space allocated for user-defined data header text, the string will be truncated. [tr#33686]

-- You can now use AS in traditional Synergy to indicate the type of members in a collection. Note that this feature does not yet result in type conversions, so essentially it is useful only for unboxing. [tr#35020]

-- The Where.In selections for the Select statement are now optimized when an ISAM key field is specified by translating to a set of OR conditions. Since IN(F,A,B) results in the equivalent (Where)(F.eq.A.or.F.eq.B), doing both head and tail optimization of each individual part (which would normally not occur because of the .OR.) has greatly improved performance. Also, if the expression contains an additional AND condition (such as IN(F1,A,B).and.(Where)F2.eq.C), each comparison of A and B includes the AND part of the expression, making separate passes for each IN parameter specified. [tr#34453]

-- (.NET) We now support %DLL_CALL on .NET. [tr#35011]

-- (.NET) We changed the default font for W_ in .NET to Consolas to scale better, especially with SYN_RESIZE_SCALE. If a user runs on an unsupported Windows XP or Server 2003 machine, the Consolas font will need to be installed from Microsoft. [tr#34510]

-- We now provide a notification when an invalid service name is specified on the dbssvc unregister command line. [tr#35467]

-- (.NET) Using an environment variable with a trailing backslash (\) and XSUBR in .NET incorrectly generated an "index out of range" exception. This has been fixed. [tr#35429]

-- (.NET) The concurrent use of XCALL GETLOG/SETLOG on multiple threads previously corrupted the environment or caused a segmentation fault. This has been fixed. [tr#35456]

-- (.NET) If %MEM_PROC(DM_RESIZ+DM_STATIC) was used on a local handle, an INVHLD error occurred (which did not happen in traditional Synergy) and memory was leaked. Although this is invalid code, the compatibility should have been the same. This has been fixed. [tr#35457]

-- (.NET) Sometimes a container was too small when SYN_RESIZE_SCALE was set. This has been fixed. [tr#35461]

-- (.NET) Using a System.String type function on a STOP statement is now allowed. A STOP statement cannot be used with the -portable compiler option; an E_NOTSUPT error is reported in this situation. [tr#35492]

-- (.NET) Pressing CTRL+P in Print Preview mode caused a segmentation fault if main() was not marked with the attribute {STAThread}. This has been fixed to report an E_ARGMIS error if the attribute is missing and CTRL+P is pressed. [tr#35500]

-- String.EndsWith and String.StartsWith reported false positives when compared with a longer string. This has been fixed. [tr#35437]

-- Creating an ISAM file with the maximum record size with compression no longer generates an ISAM file where any record stored generates either "Internal ISAM error - Data segment size mismatch with leaf" (in 10.1) or a segmentation fault (prior to 10.1). [tr#35459]

-- (.NET) In rare cases issuing a STOP statement in a WinForms or low-level Windows application caused a segmentation fault. This has been fixed. You must recompile your code with 10.3.1, and the "Target Synergy runtime" option (on the Build page of Visual Studio Project Designer) must also be set to "10.3.1." If it’s set to an earlier version, you won’t get this fix. [tr#35491]

-- (UNIX) The synsslx check for libcrypto.so was wrong for modern 64-bit Linux like Unbuntu and caused SSL failures. This has been fixed. [tr#35168]

-- (UNIX) Using rsynd on 32-bit UNIX with LDAP is unsupported but no longer causes a core dump. Using XCALL GTPPN on 32-bit UNIX with LDAP returns zero for the group and member instead of doing a core dump. [tr#35174]

-- (UNIX) If %DLL_OPEN could not open a file via its full path or find the target file in LD_LIBRARY_PATH, the code appended DBLDIR to the beginning of the path and looked there. Then the runtime reported this updated path in the error message rather than the original path. Additionally, there was no way to get the error text from dlerror() to find out why it failed. These have been fixed, and the <reserved2> argument from XCALL ERROR shows the resulting system error text. [tr#35288]

-- (Windows, UNIX) When storing records to a non-TBYTE ISAM file where the index file (.ism) reached the maximum 2 GB before the data file (.is1) and an $ERR_FILFUL error was generated, it was observed that after running isutl -v on the file, an Index error was generated, indicating an error in the index freelist and unreferenced index blocks. This has been fixed. [tr#35359]

-- (.NET) S_SERVER_THREAD_INIT no longer generates an error in the primary AppDomain. [tr#35396]

-- (.NET) XCALL SETLOG no longer causes unhandled exceptions if two threads use either XCALL SETLOG concurrently or XCALL GETLOG at the same time as XCALL SETLOG. [tr#35396]

-- A signal trap occurred on OPEN when using a search path that contained a remote filename specification. This has been fixed. [tr#35148]

-- (Windows, Unix) Previously, creating an ISAM file from an XDL file would result in a "file not found" error if the OPEN specified "" for the file. This has been fixed. [tr#35146]

-- We resolved an ISAM data corruption issue caused during file update with a non-TBYTE file that resulted in a file exhibiting a data file at or exceeding 4 GB (>=4294967296) in size. [tr#35179]

-- The GETFA subroutine with one of the keywords ALQ, FFB, or HBK now returns the correct size value when the specified file exceeds 2 GB in size. As before, if the size of the field passed to return the size is too small, the value will be truncated to the size of the field. However, if the return field is an integer and it is too small, we now return the largest positive value that will fit in the return field instead of a negative value. So, if the size of the return field is an i2, any file that is greater than 32767 bytes will return the value of 32767 (which is the largest positive value allowed by the i2). [tr#34951]

-- (Windows) Doing a STOP chain to a command file in the form STOP '@cmdfile' no longer causes the RUN license and any other license currently logged in by that application to remain logged in. [tr#35131]

-- (.NET) In addition to setting the environment, the logicals RUSER SCSCOMPR and ISAMC_REV are supposed to update the current execution environment. On .NET this did not occur, and only the initial environment was updated. This has been fixed. [tr#34752]

-- (.NET) When CTRL+C was disabled with XCALL FLAGS, ALT+F4 still shut down the app when using W_ routines. This has been fixed. [tr#35071]

-- (.NET) The correct versions of the OpenSSL libraries were not always found, especially when using ClickOnce. This has been fixed. This change affects the HTTP document transport API, xfNetLink Synergy, xfServer client with encryption, and the encryption routines (DATA_ENCRYPT and DATA_DECRYPT). For ClickOnce deployments using these capabilities, OpenSSL must already be installed in DBLDIR on the target machine; you cannot install the OpenSSL libraries via ClickOnce. In addition, other non-.NET windows products no longer look in path and now expect the DLLs to be copied to dbl\bin. [tr#35129]

-- (Windows, UNIX) The parsing of URIs for HTTP functions incorrectly stripped the query from the URI when the URI did not contain a path. This has been fixed. (Note: This was released on OpenVMS in 10.1.1d.) [tr#35327]

-- (UNIX) In versions 9.3.1 through 10.1.1c, the %ABS function returned 0 for user-defined functions. This has been fixed. [tr#34996]

-- (UNIX) Previously, a call to SYN_UNESCAPE_HANDLE in rare cases corrupted the output string and resulted in duplicate characters. This has been fixed. [tr#35176]

-- (UNIX) With versions 9.3.1 through 10.1.1c, variables were not always cleared as expected when running a pre-9.3 .dbr that used INIT. This has been fixed. [tr#35199]

-- (UNIX) %DATETIME_TO_I8 now appropriately adjusts to daylight savings time, and when used in a keyed READ with a timestamp key, it reads the expected record. [tr#35261]

-- (UNIX) If the message argument to the SEND statement has a length of 0, an ARGSIZ error is now signaled. [tr#35289]

-- (.NET) The Select.CTinfo.SnapshotTime property now returns the correct snapshot DateTime. [tr#35533]

Synergy Configuration Program (SynConfig)

-- The Synergy Configuration Program now supports enabling the xfServer Connection Recovery feature. [tr#35404]

-- Adding or modifying an xfServer or xfServerPlus service with a

Service name or Display name that is not unique across all services now generates an appropriate message box instead of an empty one. [tr#35026]

____

Synergy DBMS

-- We added a new file creation option, NOROLLBACK (",[NO]ROLLBACK" or XDL FILE keyword ROLLBACK yes|no), which prohibits the rollback function when change tracking is enabled. Also, the ctutl utility has a new change tracking command option, -n, that prohibits rollbacks on a file. It can be used with or without the -a command. [tr#34450]

-- We added a new file creation option, SIZE_LIMIT (",SIZE_LIMIT=<value>" or XDL FILE keyword "SIZE_LIMIT <value>", where <value> is the maximum number of megabytes that the data .is1 file is allowed to grow). If a STORE or WRITE operation will cause the file to exceed this size, a FILFUL error will be generated and the record will not be stored. [tr#34132]

-- We added a new file creation option, RECORD_LIMIT (",RECORD_LIMIT=value" or XDL FILE keyword "RECORD_LIMIT value", where value is the maximum number of active records allowed in a file). If a STORE or WRITE operation will exceed this number, a RECLIMIT error will be generated and the record will not be stored. [tr#34474]

-- Bldism now recognizes the NETWORK_ENCRYPT option and can create files with network encryption. [tr#35430]

-- We enhanced isutl to better detect and correct the kind of ISAM data corruption with non-TBYTE files (shown in tr#35179) where the data file appears to be at or exceeding 4 GB (greater than 4294967296). However, due to the severity of the corruption, the file header may become damaged, in which case isutl will recommend using fconvert to extract the data to a new file. [tr#35187]

-- ISAM records that exceed the old 64K limit are now allowed when a file is created with ",variable" and 0 for the maximum record size. A file of this type can be created or accessed in 64-bit traditional Synergy or 32- or 64-bit Synergy .NET. [tr#33307]

-- We added error checking when writing the "extend" area of the primary file (for anything not merged at the end of the file) and also when writing the OrderBy intermediate (temporary) file with information for the Select when OrderBy is specified. [tr#35008]

-- Fconvert no longer creates an ISAM file when an exception file exists. [tr#21439]

-- Fconvert now records the number of exception records written when optimized. [tr#34842]

-- Running isutl -r when an old .257 file existed caused unexplained "Sort work file in use" errors. A .257 file results if a failure occurs during the data sort phase of isutl (typically in FILFUL situations). The error situation was possible if the .257 file was not removed by either re-running isutl or physically removing it from the file system. We’ve resolved the problem by automatically removing old .257 files. [tr#34376]

-- (UNIX) An "Internal ISAM error - File may be corrupt" error with the informational text "Unable to locate bucket" no longer occurs on a DELETE operation in very large variable-length records or compressed record files (greater than 4 GB) [tr#35123]

-- (UNIX) With previous version 10 releases, some ISAM files created with version 9.5.3b or earlier and configured with TBYTE could falsely exhibit file corruption. Running isutl -v on such a file reported "Bad reset PAGE offsets." This has been fixed, and the -v option for isutl has been updated to report this situation and suggest using isutl -r to correct it. The fastest way to correct this condition is to patch your file to REV 6 using isutl -p. [tr#35181]

-- (UNIX) Isutl -p previously caused a memory fault error if it patched a pre-REV 6 file that was compressed but contained data that didn’t compress well (i.e., records without much white space or many zeros). This has been fixed. [tr#35212]

-- (UNIX) Running isutl -r or -v no longer fails with "I/O error: No disk space" (when plenty of disk space is available) on a non-TBYTE file when the index file is greater than 2,147,483,647. (A workaround to this is to delete the index file and re-index. If you can add TBYTE, do so now. Otherwise, a higher packing density may suffice.) [tr#35361]

-- (Windows, UNIX) The isutl utility no longer allows an index file to exceed 2,147,483,647 bytes when re-indexing a file that is not configured with TBYTE. It now fails with the error "Index or data exceeds maximum size" and a message indicating that only the data file (.is1) remains, and suggests re-indexing with the addition of TBYTE (-qfile=tbyte). [tr#35361]

-- (Windows, UNIX) The error $ERR_SIZLIMIT (rather than$ERR_FILTUL) is now generated when either part of an ISAM file (.ism or .is1) will exceed 2 GB when updating a file that was not configured with the TBYTE file option. [tr#35361]

-- The index and data headers of an ISAM file pair are now compared on OPEN and will generate an $ERR_NOTISM error ("Not an ISAM file") if they are different. [tr#35360]

-- In version 10.1.1, -ln and -le needed to be at the end of the isutl command string. This has been fixed. [tr#35035]

Synergy HTTP document transport API

-- We increased the maximum HTTP header size to 2048 to better accommodate QAUTH packets. [tr#35080]

-- We updated routines in the Synergy HTTP document transport API so that a variable used for an error argument will now be cleared if the routine completes without error. Variables used for <out_headers> arguments are now also cleared. [tr#34641]

-- The HTTP_ routines caused an "Unhandled exception" error in the .NET runtime when the log file parameter was a zero-length string. This has been fixed to ignore the string. [tr#35117]

-- We changed the stack allocations to use ALLOCA instead of stack allocations for header buffers. [tr#35126]

Synergy .NET API

-- The minimum Framework supported is now CLR 4.0.30319. We no longer distribute gennet.exe, which supports 2.0. [tr#34961, tr#34966]

-- Property initializers used within lambdas reported INVINITMBR errors. This has been fixed. [tr#35257]

Synergy routine call block API

-- (UNIX) A recursive error condition was fixed when doing argument scope checking using the routine call block API and programs built with -qcheck. [tr#34974]

-- (UNIX) When using RCB_SETARG, RCB_SETARGS, or RCB_INSARG with a temporary literal value, the value specified will no longer change after being set. In addition, using literals to RCB_SETARG will no longer cause an F_RCBREL error on segment reclamation of the routine where the literal was set if the RCB handle is still active. [tr#34978]

Utilities

-- (UNIX) Constructor arguments no longer cause an error in dbl2xml. [tr#34968]

-- (UNIX) Functions without an explicitly declared return value where the first FRETURN was another unresolved function call caused dblproto to crash. This has been fixed. [tr#35252]

-- (UNIX) Using dblproto on a parameter whose type was a handle to an unresolved type marked the parameter as optional. When compiled and the type was resolved to class, the parameter was marked as required. Because the prototype didn’t match the declaration, an unexpected PROTOMISMC error was reported. Dblproto now correctly marks the parameter as required when the type is a handle. [tr#35259]

xfNetLink Synergy

-- We added support for the following xfNetLink Synergy routines to the .NET runtime: %RX_START_REMOTE, RX_SHUTDOWN_REMOTE, %RXSUBR, %RX_RMT_TIMOUT, %RX_RMT_SYSINFO, %RX_CONTINUE, RX_GET_ERRINFO, RX_GET_HALTINFO. For more information, see the xfNetLink & xfServerPlus User’s Guide. [tr#34825]

RESTRICTIONS

The following restrictions have been identified in version 10.3. See the RESTRICT.TXT file for restrictions identified in earlier versions.

Runtime

-- (.NET) The .NET DrawString function doesn’t support the ATTR_UNDR Windowing attribute underlining an entire untrimmed buffer. [tr#35287]

-- (.NET) On Windows Phone and Windows Store, the ERRMOD routine always returns a line value of 0. [tr#35599]

Synergy DBMS

-- Running xdlchk on a file with an invalid numeric overlay reports the XDL file is valid. [tr#36551]

Synergy Windows printing API

-- Lines drawn by the Synergy Windows printing API’s DWP_LINE function are reduced in thickness after the first page when the pen is not deleted and recreated for each page. [tr#33894]

OPEN INCIDENTS

Compiler

-- Linking in a partial class with a private method causes an unexpected CLSCRC error and a potential dblproto error. The workaround is to combine partial classes in the same source. [tr#36141]

-- Casting ^M of a structure on a handle to @a gives an E_INVCAST error. The workaround is to add an intermediate (a) cast. [tr#34560]

-- (Windows, UNIX) An expected NFND error does not occur when a local variable matches the leading edge of the search path. For example, a method call of variable e of type joe that does not have the method defined matches an e in a higher scope of a different class that does have the method defined. [tr#35469]

-- When building a project, a CLSCRC error can occur when linking an ELB if the base classes aren’t first in the prototyping order. [tr#35752]

-- When compiling -qnet identifiers longer than 30 characters, the identifiers are truncated to 30 characters without a warning. [tr#36740]

-- In the traditional compiler, -qrelaxed:allowdup allows duplicate subroutines to load without a TOKUDF error even if their arguments are different. [tr#37581]

Debugger

-- (OpenVMS) A second connection to remote debug port is not producing an error. You cannot have two sessions on the same port. [tr#31408]

-- The debugger does not support multi-level casting of objects. [tr#32571]

-- When debugging a program that includes more than 255 source files, the debugger is not able to correctly find labels in source files whose source number is above 255. Attempting to set a breakpoint on such a label may result in a %DBG-E-Breakpoint error, or may appear to succeed without actually setting the breakpoint in the routine. A workaround is to set a break line number. [tr#36582]

-- The traditional debugger has never supported property evaluation (or running arbitrary code). The user experience would be improved in the VS debugger if it at least listed properties in scope, even if it didn’t evaluate them. [tr#36715]

-- Setting a breakpoint in the debugger does not set it in all source files in all ELBs. [tr#36768]

-- When debugging in 64-bit, the line number provided by the client log and the SHOW TRACE command is one number off from what the Visual Studio Editor shows. [tr#37149]

-- Examining a child object of an arrayed field may get an invalid path. This is one specific case of the generic issue of not being able to access object fields based on the actual runtime class instance instead of the declared type. As the most basic example, if handle HND is declared as "@*", class CLS has a field FLD, and HND has an instance of CLS, the debugger cannot currently reference HND.FLD because it doesn’t know that the type of HND’s object is CLS; it only knows it’s "@*". [tr#37187]

-- Breakpoints cannot be set in procedure division code that is .INCLUDEd. [tr#37441]

-- A postmortem stack trace in Visual Studio may show one line too many. [tr#37163]

-- Duplicate variables are displayed in Visual Studio for nested classes with overridden variables for the same name and a different type. Only the current class variable should be displayed. [tr#37198]

-- When using Visual Studio remote debugging with bound chain programs on OpenVMS, the debugger does not break on chaining. [tr#37675]

Installation

-- (Windows 10) On a 64-bit SDE Client installation, Synergy Control Panel is missing its links. [tr#36436]

-- 64-bit SDE does not properly detect existing licensing if the license server is specified using an IP address. [tr#36413]

-- 64-bit SDE Client is not installing environment variables on Windows 10 correctly unless you log out and restart after installation. [tr#36469]

-- Upgrading a machine with running dbssvc services causes License Manager service to be disabled. [tr#36321]

Runtime

-- (Windows, UNIX) When a stop is called either explicitly or implicitly, open channels in output or append mode are not purged as expected. [tr#36118]

-- Issuing the Count property on a Select that contains an OrderBy.Ascending or OrderBy.Descending takes almost twice as long as iterating through the same Select (with FOREACH or MoveNext). [tr#36148]

-- A Reset() is supposed to set the enumerator to its initial position before the first element in the selection, but when iterating on a Select with Orderby.Ascending or Orderby.Descending, instead of the first row being returned, the next row is being returned. [tr#36149]

-- Using Where.In within a Select where one or more fields contain the same value results in the same matching record being returned more than once when optimized. [tr#36226]

-- (UNIX) When you execute an LPQUE command, the underlying shell is emitting escape sequences to reset terminal wrap, turn on the cursor, and reset all attributes. However, the shell should not emit any escape sequences for LPQUE. Note that this only occurs when running a UI Toolkit or low-level windowing application. [tr#23243]

-- Certain XDL keywords like (NETWORK_ENCRYPT, TRACK_CHANGES, DUPLICATES, etc.) that have yes/no responses are not validated against invalid responses. [tr#34554]

-- Enabling a new network adapter or moving a mobile device that activates a new adapter (e.g., wired to wireless) causes xfServer access to fail with a network error when xfServer is running on the same machine as the client application. The workaround is to change the metric on the adapter in the IPv4 config settings. [tr#34498]

-- (.NET) An OPENELB of an assembly whose name has trailing spaces fails. You can use %ATRIM as a workaround. [tr#34392]

-- (.NET) A dynamic array of Synergy structure types causes a non-blittable error when it contains a boolean type. [tr#31494]

-- (Windows, UNIX) Using a remote spec (@server) in a search list logical doesn’t work. "DAT=@myserver,dat" works only if the file exists on myserver; if the file is only in the dat directory (or any other directories following dat in the search list logical), it won’t be found. "DAT:dat,@myserver" works only if the file exists in the dat directory (or any other directories preceding the remote server spec in the search list logical). [tr#35395]

-- Incorrect instantiation of the AlphaEnumerator, RestrictedAlphaEnumerator, and CTInfo classes should generate an error but doesn’t. [tr#36006]

-- (.NET) Using an integer in a (NoCaseWhere) expression causes an InvalidCastException of Select.Where to Select.NoCaseWhere. [tr#36475]

-- (OpenVMS) The UNLOCK statement releases both manual and automatic locks. On Windows and UNIX, UNLOCK leaves the manual locks locked. A request was made for UNLOCK to behave the same on all systems by only releasing the automatic locks. [tr#36673]

-- (OpenVMS) Finding BOF on a stream file does not position correctly. When a Select is performed on a stream file, the first record(s) are missed. [tr#37432]

-- (Windows) In the DISPLAY statement, the decimal values 7, 8, 9, and 10 do not display the crossing, right tee, lower-left corner and bottom tee characters, respectively. [tr#37622]

Synergy DBMS

-- The isutl utility does not detect .is1 header corruption. [tr#23168, tr#23169]

-- When copying a file using fconvert (without a description file) that contains portable integers (file options ",i=n:m"), the portable integer definitions aren't transferred to the new file. If a description file is used (-d) that contains the portable integer definitions, they are transferred correctly. [tr#32314]

-- When specifying a remote description file that is an FDL file, fconvert generates an "%FCONVERT: Invalid par file - Bad file option FDL_VERSION" error and fails. [tr#35868]

-- (Windows) Fcompare incorrectly reports data errors when a record contains a group field. [tr#36443]

-- Executing ipar with -g or -gx and specifying the file with a logical or complete path will result in either a 0-byte file with the name of the logical being created in the current directory, a PAR or XDL file named <logical>:<filename> created in the current directory, or the PAR or XDL file created in the directory in which the file is located. [tr#36495]

Synergy .NET assembly API

-- When using the Synergy .NET assembly API in conjunction with DevExpress 10.1.4, the embedded DevExpress undo/redo controls do not change to an enabled state when the text in the textbox changes. [tr#31554]

-- When an event is based on a delegate whose name is 30 characters or more, gennet40.exe will not include that event in the generated code. [tr#36490]

Synergy XML API

-- When an invalid XML file is placed into an XML string and this string is placed into a handle and then passed to %XML_PARSER_PARSESTRING, a "Map outside bounds of field or handle" error will occur, but only if the XML has all carriage return/line feeds removed when placed into the dynamic memory handle. [tr#36465]