

Synergy/DE Quick Migration Guide

Migrating from Version 6 to Version 9.3

Migrating from Version 7 to Version 9.3

Migrating from Version 8 to Version 9.3

Printed: December 2009

The information contained in this document is subject to change without notice and should not be construed as a commitment by Synergex. Synergex assumes no responsibility for any errors that may appear in this document.

The software described in this document is the proprietary property of Synergex and is protected by copyright and trade secret. It is furnished only under license. This manual and the described software may be used only in accordance with the terms and conditions of said license. Use of the described software without proper licensing is illegal and subject to prosecution.

© Copyright 2001–2005, 2007, 2008 by Synergex

Synergex, Synergy, Synergy/DE and all Synergy/DE product names are trademarks of Synergex. Windows and ActiveX are registered trademarks of Microsoft Corporation. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All other product and company names mentioned in this document are trademarks of their respective holders.

MG-02-9301

Synergex
2330 Gold Meadow Way
Gold River, CA 95670 USA

<http://www.synergex.com>

phone 916.635.7300

fax 916.635.6549

Contents

Overview

Migrating from Version 6 to Version 9.3

License Manager	2
Synergy Language	3
UI Toolkit	22
Composer	29
Repository	30
ReportWriter	33
xfServer	34
xfServerPlus	36
xfNetLink Java Edition	38
SQL OpenNet	39
xfODBC	40
SQL Connection	42

Migrating from Version 7 to Version 9.3

License Manager	45
Synergy Language	46
Workbench	67
UI Toolkit	69
Composer	76
Repository	77
ReportWriter	78
xfServer	79
xfServerPlus	80
xfNetLink Java Edition	82
xfNetLink COM Edition	83

xfNetLink .NET Edition	84
SQL OpenNet	86
xfODBC	87
SQL Connection	90

Migrating from Version 8 to Version 9.3

License Manager	93
Synergy Language	94
Workbench	110
UI Toolkit	111
Composer	116
Repository	117
ReportWriter	117
xfServer	118
xfServerPlus	119
xfNetLink Java Edition	121
xfNetLink COM Edition	121
xfNetLink .NET Edition	122
SQL OpenNet	124
xfODBC	125
SQL Connection	126

Overview

This document explains the changes you may need to make to your system and code when you upgrade to Synergy/DE™ version 9.3.

- ▶ If you are migrating from Synergy™ version 5.7.9 or earlier, refer to the [Updating Synergy/DE](#) manual for information about changes in version 5.1 and 5.7. Then, see the version 8.1 *Quick Migration Guide* for additional information; it includes a section on migrating from version 5. (This document is available in the Resource Center on the Synergex™ website at <http://resourcecenter.synergex.com/product/docs/manualv81.aspx>.)
- ▶ If you are migrating from Synergy/DE version 6.1 or 6.3, see “[Migrating from Version 6 to Version 9.3](#)” on page 2.
- ▶ If you are migrating from Synergy/DE version 7.1, 7.3, or 7.5, see “[Migrating from Version 7 to Version 9.3](#)” on page 45.
- ▶ If you are migrating from Synergy/DE version 8.1 or 8.3, see “[Migrating from Version 8 to Version 9.3](#)” on page 93.

Operating System Requirements

For operating system requirements, see the version 9.3 installation instructions, or the “[Requirements and Considerations](#)” chapter of the *Installation Configuration Guide*, or the [Synergy/DE platforms page](#) in the Resource Center on the Synergex web site. For information on patch requirements for your operating system, see the “[Requirements and Considerations](#)” chapter of the *Installation Configuration Guide*.

Additional Resources

- ▶ For additional information on migrating, see the product release notes, the [Updating Synergy/DE](#) manual, or your Synergy/DE account manager.
- ▶ For details on migrating to Windows, see [Migrating Your Application to Windows](#), available on the Online Manuals CD.
- ▶ For details on migrating from VAX DIBOL, see [Migrating from VAX DIBOL to Synergy Language](#), available on the Online Manuals CD.

Questions or Comments

We welcome your comments and suggestions for improving this guide. Send your comments, suggestions, and queries, as well as any errors or omissions you’ve discovered, to doc@synergex.com.

Migrating from Version 6 to Version 9.3

This section is designed to help you migrate from version 6.1 or 6.3 to version 9.3.

License Manager

Compatibility Information

You must reconfigure all products for Synergy/DE version 9.

Required Changes

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
Windows	If you use the Synergy Service manager utility (servbox.exe)...	Use the Synergy Configuration Program instead.	The Synergy Service manager is no longer distributed.
Windows	If you run License Manager from the License Configuration option on the Start menu...	Use the Synergy Configuration Program instead.	The menu option has been removed.

Synergy Language

Compatibility Information

- ▶ Version 9 will run code compiled and linked with Synergy/DE 6.1 and higher; you do not *have* to recompile.
- ▶ You may *want* to recompile, however, to take advantage of new optimization techniques and some fixes in version 9 (as noted in the table below). You may choose to recompile some portions of your application now and other portions later. See also the compatibility information in “UI Toolkit” on [page 22](#).
- ▶ If you recompile, you will need to make changes to your code due to the stricter error checking in the version 9 compiler and the fact that we have strongly prototyped all Synergex-supplied subroutines and functions. In most cases, the version 9 compiler will find errors that earlier versions of the compiler did not. You can turn off most of the stricter error checking by compiling with the **-qrelaxed (/RELAXED** on OpenVMS) switch. For details on this switch and its options, see “[Compiler options](#)” in the “Building and Running Synergy Language Programs” chapter of *Synergy Language Tools*.
- ▶ To debug with version 9, you must recompile, because the version 9 debugger cannot show source line information or provide symbolic access to variables unless the program is recompiled. If you need to debug your code in version 9, but are not yet ready to take advantage of version 9’s stricter error-checking and strong prototyping, recompile with the compiler distributed as **dbl8**. Note that code compiled with **dbl8** can be run only on version 9.
- ▶ The object files, object libraries, and executables generated with version 9 are not backward compatible. However, changes you make to your code to comply with the enhanced error checking and strong prototyping will compile under previous versions of Synergy, unless stated otherwise in this document. In those few instances, you can use the IFDEF compiler directive so that the code can be compiled with either version 9 or a previous version. For example, let’s say you want to overload the subroutine PUTCM, which was defined like this prior to version 9:

```
.subroutine putcm  
data, a
```

Due to strong prototyping, the parameter *data* is defined thus in version 9:

```
in req data, a
```

Since the IN and REQ parameter modifiers are new in version 9, code that includes them will not compile in a prior version. You can handle this situation thus:

```
.ifdef DBLV9  
    subroutine putcm  
        in req data, a  
.else  
    subroutine putcm  
        data, a  
.endc
```

- ▶ (Windows) We changed the default installation directory to C:\Program Files\Synergex\SynergyDE.
- ▶ (Windows, UNIX) Synergy/DE version 9 supports concurrent access to ISAM revisions 2 through 5. All ISAM files created with version 7.1 or higher, whether they contain new features or not, will be created at the REV 4 level by default. (REV 4 was introduced in version 6.3.0d.) See the table below for more information about ISAM files.

- On OpenVMS, due to increased memory requirements, you may need to change the value of some authorize parameters for developers who are using the version 9 compiler. See “[OpenVMS requirements](#)” in the “Requirements and Considerations” chapter of the *Installation Configuration Guide* for details and recommended values.

Required Changes

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If you pass a variable of the wrong data type, such as an alpha variable to a routine that expects a decimal or integer variable...	Correct your code to use the correct data type. Optionally, you may be able to cast (^A , ^D , or ^I) the variable to the correct type.	Passing the incorrect data type causes the compiler to generate a “Type mismatch” error. The version 9 compiler strongly prototypes Synergex-supplied functions and subroutines, which means that your call must exactly match the Synergex-supplied routine in return type, and in number, direction, and type of parameters, as well as whether parameters are optional or required. You can reduce the occurrence of this error by using the -qrelaxed:param compiler option. In addition, the version 9 compiler strongly prototypes all user-defined routines in a compilation unit to ensure that calls within that unit match the routine declaration. For these routines, you can use the -qrelaxed:local compiler option to reduce the occurrence of this error.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If you pass more arguments to a routine than the routine defined...	Correct your code to pass the correct number of arguments. For user-defined routines, you can optionally do one of the following to correct your code: - Change the routine declaration to include the extra arguments. - Change the routine declaration to include the VARARGS modifier.	This situation now causes the compiler to generate a "More argument values than declared parameters in routine" error. See the item above for an explanation of strong prototyping in Synergex-supplied and user-defined routines and for the compiler options that will reduce the occurrence of this error. Note that if you use VARARGS, your code will no longer compile in versions 8.3 and earlier. For more information on VARARGS, see the Synergy Language Reference Manual.
All	If you fail to pass required arguments to a routine...	Correct your code to pass the missing arguments.	Previously, this error would not be caught until runtime. Now, because the version 9 compiler strongly prototypes all Synergex-supplied routines, it causes the compiler to generate a "Missing required parameter <i>name</i> in routine <i>name</i>" error.
All	If your code includes external functions in which the function return type in the function declaration does not match the return type of the function itself...	Correct your code so that they match. In version 9 you can explicitly specify the type of the function return value. If it's not explicitly typed, the return type is determined by the data type of the first FRETURN statement. See FUNCTION-ENDFUNCTION in the "Synergy Language Statements" chapter of the <i>Synergy Language Reference Manual</i> for details.	Because the version 9 compiler strongly prototypes all routines in a compilation unit, this situation now causes the compiler to generate an "External function <i>name</i> does not match prototype" error. You can reduce the occurrence of this error by using the -qrelaxed:extf compiler option.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If you overload any Synergex-supplied function or subroutine...	Verify that your overloaded routine exactly matches the prototype of the Synergex-supplied routine in return type, and in number, direction, and type of parameters, as well as whether parameters are optional or required.	We now strongly prototype Synergex-supplied functions and subroutines. Using the -W4 compiler option (/WARNINGS=4 on OpenVMS) will show the prototype the compiler expects and your overloaded routine definition, so that you can see how they differ. Once you change your routine, your code may no longer compile in versions 8.3 and earlier. See “Compatibility Information” on page 3 for a suggestion on how to handle this situation.
All	If you use the %SYNSOCK routine...	Change your code to use the %SS_ functions instead.	Previously, the Synergy socket API routines were implemented as parameterized macros (%SS_), which invoked the %SYNSOCK routine. In version 9, the compiler treats the %SS_ routines the same as functions in order to support strong prototyping. Therefore, %SYNSOCK is no longer needed and has been removed from the runtime. See the “Synergy Socket API” chapter of the <i>Synergy Language Reference Manual</i>. This change may break your code.
All	If you use END (as opposed to .END) and expect all .DEFINEs to be removed and unavailable to the next module in the source file...	Add .UNDEFINE compiler directives as necessary. For example, a .DEFINE might be used to control the initializing of a global data section: <pre>.ifdef GBL_INIT .undefine GBL_INIT global data section A ,init .else global data section A .endc</pre>	The compiler now treats END and .END the same. This means that .DEFINEs are not undefined when END is encountered and are therefore available to the next module in the source file. This could cause a “Symbol already defined” error or lead to unexpected behavior because something is defined when it should not be. This change may break your code. To retain the previous behavior, in which END and .END were treated differently, use the -qrelaxed:end compiler option.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If you define a field that implicitly extends a record by overlapping another field... For example, <pre>record pbuff ,a80 page_hdr ,a5@pbuff page_text ,a85@pbuff+5</pre>	Correct your code. In this example, you could change the size of pbuff to 90.	This is often the result of a coding error. It is no longer supported, and the compiler now generates a "Cannot extend record by <i>nn</i> bytes with overlay field <i>name</i>" error. Note that a field is permitted to extend a record as long as it does not overlap another field.
All	If you have records that overlay commons or records, or if you have commons that overlay other commons...	Be aware that record and common overlays are now strictly enforced as to the type being overlaid. You may need to correct your code.	This situation causes the compiler to generate an "Overlay not allowed" error. See "Overlaying data" in the "Defining Data" chapter of the <i>Synergy Language Reference Manual</i> for a list of rules that apply to overlaying records.
All	If your code <code>.INCLUDEs</code> named records that have overlay fields, and you get an "Ambiguous symbol" error when compiling...	Correct your code to remove the ambiguity. The easiest way to do this is to use the NORECORD qualifier. For example, instead of this: <pre>.include "rec" repository, record=myRecord</pre> You can do this: <pre>record myRecord .include "rec" repository, norecord</pre> The other option is to remove the ambiguity by renaming the duplicate symbol.	In version 9, we changed the way in which <code>.INCLUDE</code> offsets are specified by the compiler. If the record is <i>named</i>, dot notation using the record, group, and field names is used. If the record is <i>unnamed</i>, a numeric offset relative to the overlay field is used. The use of the named reference could cause the compiler to generate an "Ambiguous symbol" error if you have a duplicate symbol elsewhere in your code. Using the NORECORD qualifier will ensure this does not happen. For more information, see "Including from a Repository" in the Discussion for <code>.INCLUDE</code> in the "Preprocessor and Compiler Directives" chapter of the <i>Synergy Language Reference Manual</i>.
All	If your code includes a data division in which the first item is a structure <i>and</i> there is no MAIN statement...	Add a MAIN statement.	MAIN is optional <i>unless</i> the first item in the data division is a STRUCTURE statement without a corresponding ENDSTRUCTURE. The error generated by the compiler will vary.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If your code contains references to ambiguous field definitions within the data division... For example, <pre>record red ,a3 record joe blue ,a ^size(red) red ,d2</pre> A similar problem occurs when you have a structure defined in a repository, and you reference that structure as an implicit group in another repository structure and name the group the same name as the referenced structure, and then include both structures in a program.	Correct your code to remove the ambiguity. In the first example, the correction required would depend on what the code was attempting to do. For the repository example, you need to either rename the first structure in the .INCLUDE directive to something unique, or change the name of the group within the second structure in the repository.	In version 9, ambiguous field references can be detected within the data division and cause the compiler to generate an "Ambiguous symbol" error. Previously, they could be detected only when referenced within the procedure division. To allow the old usage, use the -qrelaxed:path compiler option.
All	If you use anything other than an a6 for a GETRFA or RFA qualifier variable...	Change your code to use an a6 or set system option #56.	Prior to 9.3, a GETRFA or RFA qualifier variable that exceeded 6 bytes was allowed. Now, with the introduction of GRFAs, if the size of a GETRFA or RFA argument is anything other than 6 or 10 bytes, an "Invalid record's file address" error is generated. Setting system option #56 turns off this behavior and causes all RFAs that exceed 6 bytes to be treated as 6-byte RFAs. Even if you are using an a10, you will need to set system option #56, because a10 now indicates a GRFA and will cause a "Record not same" or "Feature not supported in this version" error when used for an RFA. This change may break your code.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If your code uses i8 data types as function return values on 32-bit systems...	Change your code to use a supported data type.	Using an i8 (or long) data type on a 32-bit platform now causes the compiler to generate a "Type i8 cannot be return value on 32-bit platform" error. Previously, there was no error, and if you passed a value larger than an i4 to an i8 , it was truncated. This change may break your code.
All	If you use the LOCAL, STATIC, or STACK qualifiers with the MAIN statement...	Remove them from your code.	MAIN no longer allows these qualifiers. Previously, although allowed, they were ignored and did not affect how your code operated. Now, the compiler generates an "Invalid MAIN declaration" error.
All	If your code uses FIRST or LAST with the I/O statements FIND, READ, or WRITE...	Use ^FIRST or ^LAST instead.	The use of these qualifiers without the caret is no longer supported and causes the compiler to generate an "xxx not found" error.
All	If your code uses short forms of identifier names, such as NOPREC for NOPRECISION...	Use the full documented identifier names.	We no longer support shortened forms of identifier names. Using them causes a compiler error.
All	If your code includes the START compiler directive without a period at the beginning...	Add the period.	All compiler directives now require a period at the beginning. Without the period, the START directive will be ignored.
All	If your code uses non-numeric variables as subscripts or in ranging operations... For example, <code>clear myvar(avar)</code> Or <code>clear myvar(offset:avar)</code>	Correct your code to use whole numbers instead.	Non-numeric variables are not allowed in subscripts or ranging operations and cause the compiler to generate a "Numeric expression required" error.
All	If your code uses implied decimal variables for channel numbers, subscripts, or in ranging operations...	Correct your code to use whole numbers instead.	Previously, implied decimal values were rounded to the nearest whole number. Now, they cause the compiler to generate a "Decimal or integer expression required" error.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If your code includes decimal fields in which the whole number portion exceeds 18...	Correct your code so that the whole number portion is 18 or less.	Although this was previously allowed, only assignment statements were guaranteed to work, and it was treated as an 18-digit field with leading zeros in the inaccessible part of the field. This now causes the compiler to generate an "invalid data size specification" warning. To allow the old usage, use the -qrelaxed compiler option.
All	If your code includes implied decimal fields, in which the difference between the whole number part of the field and the precision part is greater than 18...	Correct your code such that the difference is 18 or less.	The compiler now generates a BIGNUM error in this condition.
All	If your code includes IF statements that are missing the body statement... For example, <pre>if (condition) then nop else end</pre>	Correct your code to include a body statement.	This syntax is no longer permitted and causes the compiler to generate a "Statement expected" error.
All	If your code includes a statement on the same line as a BEGIN statement...	Move it to the next line.	This coding style is no longer permitted and causes the compiler to generate a "BEGIN should be followed by end of line" error.
All	If your code has function declarations with multiple qualifiers... For example, <pre>.function myFunc, ^val reentrant</pre>	Verify that there is a comma between the qualifiers.	The comma is now required. A missing comma causes the compiler to generate an "Invalid expression" error.
All	If you use .IF with a compile-time expression using ^SIZE or %LEN...	Correct your code.	This is no longer supported and causes the compiler to generate a "Function name is not supported in .IF" error.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If you use the following syntax: <pre>record array , [10]a5 array[] (ix:1) = value</pre>	Change your code. In this example, you can add a GROUP/ENDGROUP pair around the array declaration: <pre>record group array_grp array , [10]a5 endgroup</pre> and then change the code to <pre>array_grp(ix:1) = value</pre>	This syntax is no longer supported and causes the compiler to generate an "Invalid expression" error.
All	If your code uses brackets [] to access the elements of a pseudo array...	Use parentheses () instead.	Brackets can be used only for accessing the elements of real and dynamic arrays. Using them to access the elements of a pseudo array causes the compiler to generate an "Incorrect number of dimensions" error.
All	If your code includes arrayed fields larger than 256 MB...	Change your code.	Arrayed fields larger than 256 MB are no longer supported and cause the compiler to generate a "Size of field <i>name</i> too large" error.
All	If you use parentheses to reference elements of a structure in a ^M reference... For example, <pre>^m(quote_lines(ix), rec)</pre>	Change your code to use brackets instead.	This syntax is no longer supported and causes the compiler to generate an "Indexing not allowed in structures" error (error number 683).
All	If your code includes parameterized macros that have empty parameter lists... For example, <pre>.define myMac() %char(8)</pre>	Remove the empty parentheses.	Empty parentheses are no longer permitted when no parameters are specified in a .DEFINE parameterized macro reference and cause the compiler to generate an "Invalid compiler directive" error.
All	If your code includes nested parameterized macros of the same name...	Change your code.	In version 9, nested use of parameterized macros with the same name will cause the compiler to generate a "<i>name</i> not found" error.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If you have inadvertent unary + operations on alpha operands... For example, <pre>xcall fred ("ABC"+ & +"DEF")</pre> Or <pre>xcall fred ("ABC" &, +"DEF")</pre>	Correct your code to remove the unnecessary unary +.	This operation is mathematically invalid and is most often the result of a coding error. It causes the compiler to generate a "Numeric operand required (unary +)" error. To allow the old usage, use the -qrelaxed compiler option.
All	If your code uses the dollar sign (\$) on function calls...	Change it to a percent sign (%).	The dollar sign is no longer supported and causes the compiler to generate an "Invalid calling convention" error. To allow the old usage, use the -qrelaxed:deprecate compiler option.
All	If your code uses \$A or \$D...	Change the dollar sign to a caret sign (^A, ^D).	
All	If you use .DEFINE to simulate a Boolean operator... For example, <pre>.define either or . . . if var1 .either. var2 nop</pre>	Correct your code.	This syntax is no longer supported and causes the compiler to generate an "Invalid expression" error.
All	If you use an old form of the .PROC compiler directive that specifies the number of I/O buffers on RSTS... For example, <pre>.proc (4)</pre>	Remove the number from your code.	This is no longer supported and causes the compiler to generate an "Invalid record field" error.
All	If your code passes a numeric non-implied literal to a routine that declares it as n and uses ^D, ^F, or ^A on the passed argument without first determining its data type...	Make the argument type n , or use ^DATATYPE to determine the argument type before casting it.	Previously, literals were passed as decimals. Now, they are passed as integers, so the results will not be as expected.
All	If your code passes the <i>line</i> argument to MODNAME and declares it as a ...	Declare it as n instead.	Previously, <i>line</i> could be a or n ; now, only n is permitted.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If your code uses ^PASSED on a local variable...	Change your code.	Previously, this always evaluated to true. Now, it causes the compiler to generate an "Invalid argument" error because ^PASSED is intended for parameters only.
All	If you have files with more than 262,143 lines in them...	Split them into smaller files.	Previously, line numbers in such large files were misreported in errors. Now, the compiler generates an "Exceeded maximum number of lines in file <i>name</i> " error.
All	If you have duplicate global data sections, global commons, or global literals that are not the same size...	Change your code.	In version 9, the linker will generate an error when it finds duplicate symbols that are not the same size. The exact error depends on which symbol is duplicated. (Duplicate symbols that <i>are</i> the same size will cause the linker to generate a warning.)
All	If your code includes decimal and implied decimal variables in a single SET statement...	Correct your code to use separate SET statements for each data type.	Previously, the precision would be truncated when setting the decimal variable, without an error being reported. Now, the compiler generates a "Type mismatch between <i>type</i> and <i>type</i> " error.
All	If you use the DBLTOKENS environment variable...	None.	DBLTOKENS is not used by the version 9 compiler, so this environment variable has been retired.
All	If your code passes a ^VAL parameter to a ^REF argument in order to use the result of a call to a system DLL that returned a pointer...	Access the pointer using %MEM_PROC(DM_REG) instead.	Passing a ^VAL to a ^REF argument is not supported in version 9 and will cause the compiler to generate a "^REF required for argument <i>name</i> " error.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If your code includes a relative range specification that accesses characters preceding the specified field...	Correct your code to range within the bounds of the field.	Attempting to access characters that precede the specified field now causes the compiler to generate an "Invalid range specified" error. Although this type of access has been deprecated, the Synergy Runtime will continue to support it for backward compatibility. We recommend that you compile and run with -qcheck and make the necessary changes to avoid future memory access violations.
All	If you specify invalid MATCH, DIRECTION, POSITION, or LOCK I/O statement qualifiers...	Correct your code.	Invalid MATCH, DIRECTION, POSITION, and LOCK values now signal an E_OUTRNG error. This change may break your code.
All	If you are experiencing incorrect line numbers in the debugger...	Recompile to take advantage of a fix for this problem.	In 8.1.3, we fixed a bug that occurred when a program .INCLUDEd two files, and the first file contained only one line.
All	If you .INCLUDE repository fields that are within nested groups in which all group levels have prefixes...	Modify references to those fields in your code to include the additional prefixes.	Previously, the compiler was appending only the lowest level prefix to the field, making it possible to have non-unique field names. It now concatenates the prefixes from all levels to ensure that all field names are unique.
All	If you have programs that do not have the .PROC statement in the primary source file...	Recompile to take advantage of a fix for this problem.	This situation may cause incorrect line numbers in a traceback.
All	If you do string concatenation on a 32-bit system where the result could be greater than 65,535 bytes...	Modify your code to trap for the error \$ERR_BIGALPHA.	Previously, we did not generate an error in this situation. This change may break your code.
All	If you're passing a numeric argument to %TRIM or %ATRIM...	Pass an alpha argument instead.	An error is now returned if the argument passed to %TRIM or %ATRIM is not alpha.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
Windows	If you pass more than nine arguments with %AX_CALL...	Remove the parentheses that were used to pass the additional arguments.	%AX_CALL is no longer a macro. It now supports an unlimited number of arguments, so the parentheses are no longer required. Using parentheses causes the compiler to generate an "Invalid expression" error.
Windows	If your synergy.ini file is in the Windows directory (the default location) or the dbl directory...	Move it to another directory (outside the synergyde directory tree) and set SFWINIPATH to point to that directory.	A synergy.ini file is no longer created during installation, nor is an existing file updated. A default synergy.ini file (containing entries referencing %SYNERGYDE%) is now distributed in the dbl directory. This file is removed on an uninstall or upgrade. See the notes on page 20 for more information on synergy.ini .
Windows	If you use %WPR_GETDEVICE to obtain the printer name...	Use the <i>extended_pr_name</i> field instead of the <i>printer</i> field.	The <i>extended_pr_name</i> field supports 256 character printer names, as required for Windows 2000 and higher.
Windows	If you use the Windows printing API or the LPQUE statement...	Expect \$ERR_NOTAVL instead of \$ERR_FNF when the printer device is incorrect.	We changed the error value returned.
Windows	If you use the Windows printing API, do not specify an extension for the metafile, and then reference the metafile later...	Use an .emf extension when referencing the metafile.	Previously, when no extension was specified, .spf or .ddf was added. Now, .emf is the default extension. This change may break your code.
Windows	If your code uses the DWP_ADDFILE function and your application provides the preview function...	We recommend passing the enhanced_preview flag to DWP_BEGINJOB.	Truncation of long lines may occur in the previewer if this flag is not set.
Windows	If you use the PRINT_METHOD environment variable (PRINT_METHOD= escape) and changed the printer driver default values...	To continue using the changes to the default values, you must code them yourself within the data.	We now use the correct Windows API to send data directly to the printer. Changes you make to the printer driver default values will be ignored.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
Windows	If you expect a return value of 0 from the %TTSTS routine to indicate the channel is not open to a terminal or input is redirected...	Modify your code to expect a return value of 1.	The %TTSTS routine now returns 1 as documented in this situation. This change may break your code.
Windows	If you have programs that depend on AX_INPUT never timing out...	Specify D_FOREVER for the new wait_time argument.	The wait_time argument sets the time-out value for an ActiveX® control. For more information, see AX_INPUT in the “ActiveX Routines” chapter of the <i>UI Toolkit Reference Manual</i> . This change may break your code.
Windows	If you use the Synergy Language profiler and want to calculate elapsed CPU time using the processor time...	Set the environment variable PROFILE_PROCESSOR_TIME.	By default, the profiler now calculates elapsed CPU time using the high-granularity system clock. Set the aforementioned environment variable to revert to the previous behavior. See PROFILE_PROCESSOR_TIME in the “Environment Variables” chapter of <i>Environment Variables and System Options</i> for more information.
Windows	If you rely on the default Synergy runtime color definitions...	To maintain the same color scheme, define the specific colors you want the application to use in your synergy.ini file.	We removed the Synergy color and palette definitions from the distributed synergy.ini file in version 8.3. The new default colors are slightly different than they were previously. Consequently, you may notice a change in the appearance of applications that rely on the default colors.
Windows	If your code uses i1 variables to store the color values returned by the WI_PALET subfunction to W_INFO...	Change these variables to at least an i2 or d3 . (We recommend using an i4 .)	To accommodate the increased number of Synergy colors in version 8.3, the WI_PALET subfunction to W_INFO may now return values that are too large for i1 variables. This change may break your code.
Windows	If you use the PARSE routine to parse UNC path specifications...	Modify your code to reflect that the \\server\share_name portion of the UNC path specification is returned in the device argument.	Previously, this information was returned in the directory argument. This change may break your code.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
Windows UNIX	If you rebuild the Synergy runtime...	Create a DLL or shared object containing your C routines and access it via the Synergy DLL API.	The makedbr file is no longer distributed. The DLL API was enhanced in version 8.1 to assist you in creating C DLLs and shared objects that interact with the runtime's C interface.
Windows UNIX	If you use %DLL_SUBR...	Change your code to use %DLL_CALL instead.	%DLL_SUBR has been removed. This change may break your code.
Windows UNIX	If you're using REV 2 or REV 3 ISAM files...	Convert your ISAM files to the REV 4 level or patch them to the REV 5 level.	Although REV 2 and REV 3 files are still supported, extra overhead is required to maintain them. See "Converting to ISAM REV 4 or Patching to REV 5" on page 21.
Windows UNIX	If you need to create files that will be recognized by Synergy/DE versions prior to 6.3.0d...	Set the environment variable ISAMC_REV to 3 for each Synergy user who may be creating a file.	Versions 6.3.0d and higher support REV 4, which became the default value for ISAMC_REV starting with version 7.1. For more information, see ISAMC_REV in the "Environment Variables" chapter of <i>Environment Variables and System Options</i> .
Windows UNIX	If you use ISAM files that are named with a trailing period and no extension (e.g., myfile.)...	Rename the index file to filename.ism and the data file to filename.is1 . You may also need to change the code that generated the file to remove the trailing "." or explicitly add an extension.	Opening an ISAM file with no extension now generates an E_FILSPEC error. This change may break your code.
Windows UNIX	If the input file, output file, or temporary file for client/server sorting reside on different machines...	Move these files as necessary so that they all reside on the same machine.	Synergy sort for client/server now handles all I/O on the target machine. All file specifications must include the same host name or omit the host name. This change may break your code.
Windows UNIX	If you have any source modules that contain a global data section that doesn't have an ".INIT" but does contain an .ALIGN directive, and the object file is linked into an ELB...	Recompile to take advantage of a fix for this problem.	The described scenario caused unpredictable results because the compiler erroneously emitted data over other pieces of compiler information.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
Windows UNIX	If you have duplicate global data sections, global commons, or global literals that are not the same size...	Change your code.	In version 9, the runtime will detect duplicate symbols when ELBs are loaded and generate a BADELb error if the size is different. The error text tells you which symbols are duplicated. (Duplicates that <i>are</i> the same size will cause a warning in debug output.) This change may break your code.
Windows UNIX	If you have duplicate static records that are not the same size...	Recompile and relink.	
Windows UNIX	If you use READ or READS from a relative file when the file record size is greater than the size of the data area...	Correct your code or set system option #27.	This situation now generates an "Input data size exceeds destination size" error, as documented in the <i>Synergy Language Reference Manual</i> . This error can also occur when a file is opened in the default mode with RECSIZ set. This change may break your code.
Windows UNIX	If you call the COPY routine and specify a destination file that another process might have open...	Modify your code to handle a possible "File in use" error.	A "File in use" error is now returned if a COPY is performed on an open file. This change may break your code.
Windows UNIX	If you use %DATETIME and the resolution on your system for milliseconds is not six digits...	Review your code to see if it needs to be modified.	%DATETIME has been changed to fill with zeroes instead of spaces if the system resolution for milliseconds is not six digits.
UNIX	If you use the FORK subroutine...	Change your code to reopen any files needed by the child process.	As a result of a change we made to fix random I/O failures, the FORK subroutine now close all channels opened by the parent in the forked child process. This change may break your code.
UNIX	If you pass Windows-style commands that contain "\\\" to the UNIX shell...	Use "\\\" instead.	The SHELL subroutine now correctly handles shell environment variables (\$BBLDIR). This change may break your code.
UNIX	If you use %TNMBR...	Examine your code to ensure the value returned by %TNMBR is not being used for static data.	On some UNIX systems, %TNMBR was returning the wrong value. This change may break your code , as it may change the terminal numbers returned on a particular system.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
UNIX	If you call the RENAM or DELET routine or the SORT statement on a file that another process might have open...	Modify your code to handle a possible "File in use" error.	A "File in use" error is now returned if a RENAM or DELETE is performed on an open file or if SORT is performed when the sort output file is in use.
UNIX	If your application takes advantage of the fact that file locking between channels within the same process is not enforced...	Set the INTRAFILELOCKS environment variable to 0 (zero).	We changed the default behavior for file locks within a process; it is now the same as the behavior for file locks between processes. This change may break your code. You can revert to the previous behavior by setting INTRAFILELOCKS to 0. We do not recommend this, although it does permit you to continue executing version 8.1.7 and earlier applications that violate these new locking rules. For more information, see INTRAFILELOCKS in the "Environment Variables" chapter of <i>Environment Variables and System Options</i>.
UNIX OpenVMS	If you need to use 8-bit characters (128-255)...	You must define a LOCALE or set the LANG environment variable for your operating system.	The correct uppercase and lowercase characters in the 8-bit range (international characters) are now read from the system LOCALE. Should you want to revert to the old behavior, where A-Z and a-z with the 8th bit set were considered characters (though they rarely matched actual international characters), set the SYNERGY_NOLOCALE environment variable. For more information, see SYNERGY_NOLOCALE in the "Environment Variables" chapter of <i>Environment Variables and System Options</i>.
OpenVMS	If you're calling READS after calling WRITE, or using relative files...	Modify your code to be compatible with ANSI standards.	The next record after a previous READ/READS combination is now read, instead of the next record after the WRITE. This is the correct behavior according to ANSI standards.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
OpenVMS	If the audit server is not running on your machine...	Start the audit server and leave it running.	If the audit server is not running, you'll see the internal C++ runtime error "Create_Exclusive_Lock_1" when you compile. This error occurs because the version 9 compiler uses the C++ runtime. Even though the message looks serious, it does not affect the compilation of your code.
OpenVMS	If your code uses a lowercase logical to define the location of a repository in a .INCLUDE statement...	Change the logical to uppercase in the .INCLUDE statement or add a second define in which the logical is lowercase and quoted (which preserves the case). For example, \$ define MYRPS "<path>" \$ define "myrps" "<path>"	Prior to version 9, the compiler automatically uppercased the logical used for the repository location in a .INCLUDE statement. This ensured that it matched the logical definition, since DCL uppercases the command line by default. In version 9, the compiler does not uppercase the logical in the .INCLUDE statement. Consequently, if this logical is lowercase in your code, the compiler generates a "Logical is not defined" error. This change was made in anticipation of future support for ODS-5 case-sensitive filenames.
OpenVMS	If your code has embedded characters (e.g., CR, LF, ESC, NULL) in a quoted literal... For example, record crlf ,a2 , "<cr><lf>"	Use string arithmetic and %CHAR to pass the value that corresponds to the ASCII character. For example, record crlf ,a2 ,%char(13)+ & %char(10)	Embedded non-printing characters are not supported by the version 9 compiler. The results of using them may be unpredictable, such as getting a space instead of the desired character.

Notes:

(Windows) Synergy/DE now supports a "system-specific" initialization file (**synergy.ini**) and a "user-specific" initialization file (**synuser.ini**). Settings in **synuser.ini** override settings in **synergy.ini**. See ["Synergy initialization files"](#) in the "Environment Variables" chapter of *Environment Variables and System Options* for more information.

Converting to ISAM REV 4 or Patching to REV 5

Converting version 2 or 3 ISAM files to REV 4 or patching them to REV 5 will improve performance. We recommend using **fconvert** to convert the files. However, conversion can be a lengthy process. It is much quicker to patch the files with **isutl**. A version 2 or 3 ISAM file that has been patched to version 5 can be used with Synergy/DE version 7.5 and higher. However, you will not be able to make additional parameter changes to a patched file.

Using fconvert to convert to REV 4

Fconvert enables you to convert your existing ISAM files and add on some of the new ISAM characteristics (such as density and index block size).

1. Either unset the environment variable ISAMC_REV or set it to 4. (**Fconvert** creates files that are the current REV level or the REV level defined by ISAMC_REV.)
2. Generate a par file from your existing ISAM file.
3. Add the new options to the par file, as documented in [bldism](#) in the “Synergy DBMS” chapter of *Synergy Language Tools*.
4. Run **fconvert** on the file using the modified par file. For details, see [fconvert](#) in the “Synergy DBMS” chapter of *Synergy Language Tools*.

Using isutl to patch to REV 5

Run **isutl** with the **-p** option on the desired files. For details, see [isutl](#) in the “Synergy DBMS” chapter of *Synergy Language Tools*.

Upcoming Changes

A future version of Synergy/DE will support Microsoft .NET. As a result, some current features will not be supported in that version. These include deprecated data types, syntax, APIs, compiler options, and alignment warnings. You can see how compatible your code will be with .NET by compiling with the **-qnet** option. It will output warnings to standard error for items that will not be supported.

UI Toolkit

Compatibility Information

- ▶ If you include the Toolkit `gblctl.def` or `lstprv.def` file in your programs, you must recompile. See [“Upcoming Changes” on page 28](#) for more information on these files.
- ▶ If you want to modify a field’s range using `I_FLDMOD` or want to take advantage of using range for date and time fields, you must rescript.

Required Changes

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If you overload any Synergex-supplied function or subroutine...	Verify that your overloaded routine exactly matches the prototype of the Synergex-supplied routine in return type, and in number, direction, and type of parameters, as well as whether parameters are optional or required.	We now strongly prototype all Synergex-supplied functions and subroutines. Using the -W4 compiler option (/WARNINGS=4 on OpenVMS) will show the prototype the compiler expects and your overloaded routine definition, so that you can see how they differ. Once you change your routine, your code may no longer compile in versions 8.3 and earlier. See “Compatibility Information” on page 3 for a suggestion on how to handle this situation.
All	If you undefine <code>TOOLS_INIT</code> in your code...	Remove the <code>.UNDEFINE</code> .	<code>TOOLS_INIT</code> is now undefined in tools.def . If you also undefine it in your code, the compiler generates an “Identifier not defined” error. Once you remove the <code>.UNDEFINE</code> , your code may not work properly in versions 8.3 and earlier. See “Compatibility Information” on page 3 for a suggestion on how to handle this situation.
All	If you pass the <i>field_id</i> and <i>selection_id</i> arguments to <code>IB_END</code> ...	Remove these arguments.	These arguments have not been supported (and have had no affect on the operation of your code) since Toolkit version 3.3. Their use now causes the compiler to generate a “More argument values than declared parameters in routine” error.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If you link against tklib.olb ...	Link against tklib.elb (Windows and UNIX) or tklib_sh.exe (OpenVMS) instead.	We no longer distribute tklib.olb in version 8.3. We also no longer distribute the files chninf.dbl , chninf.def , and fkey.dbl because they were used only with tklib.olb . This change may break your code.
All	If you use chninf.def to find the highest channel number that Toolkit can use...	Change your code to use the new D_CHANNEL_MAX subfunction to %E_INFO.	We ceased distribution of chninf.def in version 8.3. This change may break your code.
All	If an input field specification in a window script or call to IB_FIELD is too wide for the window...	Revise your input field specification by shortening the field to fit within the window or specifying a view length.	Toolkit now generates an error when the display length is too long. Previously, the display was truncated. This change may break your code.
All	If you use IB_FIELD with D_OFF followed by D_FLD_PROMPT or use D_OFF followed by D_FLD_PROMPTLEN...	Be aware that both text and runtime prompts will be canceled.	This makes the behavior of IB_FIELD consistent with windows scripts, where "noprompt" is used to cancel either type of prompt. Previously, D_OFF followed by D_FLD_PROMPT canceled only text prompts, and D_OFF followed by D_FLD_PROMPTLEN canceled only runtime prompts. This change may break your code.
All	If an input field specification in a window script or call to IB_FIELD has more than nine decimal places...	Revise your input field specification to use no more than nine decimal places.	Toolkit supports only nine decimal places in implied-decimal fields. When a number greater than nine is specified, Toolkit will now generate an error message. Previously, it may have displayed the field incorrectly. This change may break your code.
All	If you use the left or right square bracket character as a field delimiter in the routines I_DISABLE, I_DISPLAY, I_ENABLE, I_INIT, or I_SETDEL...	Use a different delimiter character.	Toolkit now permits you to create fields that reference specific elements in repository arrayed fields. To do this, you must use square brackets; consequently, these routines no longer recognize square brackets as delimiters in field lists. This change may break your code.
All	If you use I_FLDMOD to modify the range of an input field...	You must recompile your window script.	Otherwise, I_FLDMOD will fail if the number of significant digits of the new minimum or maximum value is longer than that of the original value.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If you use the D_FLD_SIZE qualifier for I_FLDMOD to specify the input length for alpha or decimal data types...	Use D_FLD_INPLEN instead.	Previously, this qualifier affected the input length for all data types except integer, for which it affected the storage length. Now, D_FLD_SIZE is for the storage length, D_FLD_INPLEN is for the input length, and D_FLD_DSPLEN is for the display length. This change may break your code. For more information, see I_FLDMOD in the “Input Routines” chapter of the <i>UI Toolkit Reference Manual</i> .
All	If you pass a null string for the I_FORCE <i>new_data</i> argument, and you depend on there being a space in the field...	Change your code to pass a space instead.	Previously, a null string in the <i>new_data</i> argument was interpreted as a space (" "); now, it is interpreted as a null string (""). This change results from a feature that enables you to clear the force buffer and cancel a pending I_FORCE call by passing a null string. This change may break your code.
All	If your Toolkit application includes a customized version of any of the user-replaceable list subroutines (these are the subroutines that begin with USR_L)...	Remove any references to those routines and to the lstprv.def file.	We ceased distribution of the user-replaceable list subroutines in version 8.3. If your application includes a customized version of any of these routines, Toolkit will simply ignore it and call the corresponding default routine (e.g., it will call L_DATA instead of USR_LDATA). We also no longer distribute the files lstusr.dbl , lstprv.dbl , and lstbuf.def . This change may break your code. See also “ Upcoming Changes ” on page 28.
All	If your code passes a value other than 0 or 1 for the <i>no_place</i> argument to MB_COLUMN...	Change your code to pass a value of 1.	The name of this argument has changed from <i>no_place</i> to <i>option</i> , and it now is used to determine both column loading and placement options. Previously, any value greater than 0 would cause the column to be loaded without placing it; now, you must pass 1 to get that behavior. This change may break your code.
All	If your code creates blank menu entry names...	Correct your code so that all menu entries are named.	If you pass a blank menu entry to MB_ENTRY, a fatal error will occur. This change may break your code.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If your application exceeds the maximum number of entry lists or menu column rows in the MB_routines...	Correct your code.	Previously, the maximums (9 for lists and 20 for rows) were not enforced. Exceeding the maximums could cause unpredictable results, but there was no error. Now a "Too many lists" or "Too many rows" error is returned. This change may break your code.
All	If you have code that calls U_SELDFLT...	Update your code to call S_SELDFLT instead.	We removed the U_SELDFLT routine, which was renamed to S_SELDFLT in Toolkit version 3.0. This change may break your code.
All	If your code uses the <i>ctrl_record</i> argument to U_REND...	Change the variable definition in your code to an a425 .	We increased the size of this variable. Previously, it was an a404 .
All	If you use the g_werr record in gblctl.def ...	Define your own record to use with the W_RETERR function.	We removed g_werr. This change may break your code. For information on W_RETERR, see "Error Processing in the Windowing System" in the "Synergy Windowing API" chapter of the <i>Synergy Language Reference Manual</i> . See also "Upcoming Changes" on page 28 for more information about gblctl.def .
All	If your application accesses the gs_dec member of fldinf.def ...	Recompile your code.	To support a change in the maximum size of decimal and implied-decimal fields in Toolkit, we changed gs_dec from a d1 to an i1 .
All	If you are using DTKDBG to enable Toolkit bounds checking...	Set the new initialization setting DTK_BOUNDS.	Setting DTKDBG to 1 no longer enables Toolkit bounds checking.
Windows	If you specify color palette entry 16 for a list header or footer...	Specify color palette entry 1 to retain the current look.	Previously, color palette entry 16 did not work properly, which caused the default (1) to be used instead. We fixed entry 16, which could cause a change in the appearance of your application.
Windows	If you use the ActiveX list control and have non-static headers or footers...	Adjust the position of the text as necessary.	Text is now centered in the column by default. Previously, it was left-justified.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
Windows	In an ActiveX control, if you use %M_SIGNAL in an event method triggered by WM_KEYDOWN for a key that will be translated into a character...	User WM_CHAR to trigger the event method instead.	Using %M_SIGNAL in response to a keydown event causes the keypress to remain in the message queue, which could cause duplicate characters.
Windows	If you depend on the U_EDITREND, U_SAVELOG, or U_SAVESETTINGS routines or on the D_SAVEFONT subfunction to %U_WNDFONT to write to synergy.ini ...	You must now explicitly specify that you want to write to synergy.ini using the new options for these routines.	Synergy/DE now supports a “system-specific” initialization file (synergy.ini) and a “user-specific” initialization file (synuser.ini). Settings in synuser.ini override settings in synergy.ini . The default behavior for these routines is to write to the synuser.ini file. This change may break your code. See “ Synergy initialization files ” in the “Environment Variables” chapter of <i>Environment Variables and System Options</i> for more information.
Windows	If you use the %U_PRINTQUERY function...	Use the <i>extprinterid</i> field instead of the <i>printerid</i> field.	The <i>extprinterid</i> field supports 256 character printer names, as required for Windows 2000 and higher.
Windows	If you specify a height for a selection window associated with a set of radio buttons...	Be aware that the appearance of your radio buttons may change. You may need to clear the height setting in order to restore them to their previous appearance.	Previously, the height setting was ignored. Now, it is used to determine the arrangement of radio buttons.
Windows	If you depend on an input method being applied to the subsequent field or button when a menu entry is signaled from a method during I_INPUT processing and the user action moves context to the subsequent field...	Change your code. Or, to revert to the previous behavior, set the RETAIN_CONTEXT_CHANGE_ON_SIGNAL environment variable or the g_retaincontext global variable.	We corrected a problem that caused a method to be applied to the following field when the user moved context to that field. If your application depended on this behavior, this change may break your code.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
Windows	If your code unconditionally signals a menu (via M_SIGNAL) in a click event for an ActiveX list...	Test whether the D_LSTCLK_ONLINE flag is set.	The D_LSTCLK_ONLINE flag tells you that the user clicked on the line used to resize the columns. If you don't test for this, the drag resize of the column will be ignored. This change may break your code. For more information, see %UWNDEVENTS_METHOD in the "Utility Routines" chapter of the <i>UI Toolkit Reference Manual</i> .
Windows	If you rely on Toolkit to change the cursor from an hourglass back to a normal cursor...	Add code to change the cursor.	Setting the cursor to an hourglass will cause it to remain an hourglass until it is reset.
Windows	If you override cell color and font for an ActiveX list control and then call L_RESTART...	Reapply the overrides after the call to L_RESTART, if you want to retain them.	Cell color and font overrides are now reset to the default after a call to L_RESTART.
Windows	If you rely on the default Synergy runtime color definitions...	To maintain the same color scheme, define the specific colors you want the application to use in your synergy.ini file.	We removed the Synergy color and palette definitions from the distributed synergy.ini file in version 8.3. The new default colors are slightly different than they were previously. Consequently, you may notice a change in the appearance of applications that rely on the default colors.
Windows	If your code uses i1 variables to store the color values returned by U_REND and the WI_PALET subfunction to W_INFO...	Change these variables to at least an i2 or d3 . (We recommend using an i4 .)	To accommodate the increased number of Synergy colors in version 8.3, U_REND and the WI_PALET subfunction to W_INFO may now return values that are too large for i1 variables. This change may break your code.
Windows	If your application uses tabbed dialogs...	You may want to use the g_tabfacecolor global variable in tkctl.def to control which color palette entry will be applied to a window when it is added to a tabbed dialog.	Due to changes we made to the color palette in 8.3, on a Windows system with themes enabled, you may notice a change in the appearance of your application when a window is added to a tabbed dialog.
Windows	If your application has tab sets with the tabs on the bottom or side...	Move the tabs to the top of the tab set.	Tabs are supported only at the top of the tab set on systems that support Windows themes. This change may break your code.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
UNIX OpenVMS	If your application has an input window with more than 31 fields, and the 32nd field has an arrive method...	Recompile the window script to take advantage of a fix for this problem.	We corrected a problem that caused the arrive method on the thirty-second field to sometimes fail to be invoked.
UNIX OpenVMS	If you have an application with a check box with field-level renditions specified...	Set the new D_INPPRCRND state for E_STATE to D_OFF if you want to retain the current look of the prompt.	In 9.3, we added the D_INPPRCRND state to E_STATE. By default, this state is set to D_ON on UNIX and OpenVMS, which causes Toolkit to use field-level renditions for check box prompts. Previously Toolkit used renditions for the part of the window that the prompt occupied.

Upcoming Changes

In an upcoming major version of Synergy/DE, we plan to remove the following files from the distribution:

- **gblctl.def.** This file contains an internal structure that gets updated on each release, thereby requiring that you recompile.
- **lstprv.def.** This file contains defines for lists. These defines were required for the user-defined list subroutines, which are no longer a part of Toolkit. You should modify your code to remove references to this file.

For more information, see the “Future Revisions” section of the Toolkit release notes (**REL_DTK.TXT**).

Composer

Required Changes

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
Windows	If you use Composer to write colors or fonts to the synergy.ini file...	Be aware that colors and fonts will now be written to the [COLORS] and [FONTS] sections of synuser.ini —not synergy.ini .	Synergy/DE now supports a “system-specific” initialization file (synergy.ini) and a “user-specific” initialization file (synuser.ini). Settings in synuser.ini override settings in synergy.ini . Colors and fonts are treated as user-specific settings. See “ Synergy initialization files ” in the “Environment Variables” chapter of <i>Environment Variables and System Options</i> for more information.
Windows	If you have a composer.ini file from version 6 that you want to continue using...	Run the synckini utility to determine the location of synuser.ini . Then, move your composer.ini file to the same directory.	Composer treats composer.ini as a user-specific initialization file and searches for it in the Synergex subdirectory of your local application data directory. (Note that Composer no longer uses SFWINIPATH to locate composer.ini .)

Repository

Compatibility Information

- ▶ All version 6 repositories must be converted to version 7 format. Version 7 format is used by Repository versions 7, 8, and 9. See [“Using the Repository Conversion Program” on page 31](#).
- ▶ Repositories created with version 6.1 must be patched to correct a problem with missing format strings. See [“Patching Repositories Created with Version 6.1.0” on page 31](#).
- ▶ All distributed Repository ISAM files are in REV 4 ISAM format. They are not accessible by versions of Synergy/DE prior to version 6.3.0d. These files include **RPSDAT:rpsmain.ism**, **RPSDAT:rpsstext.ism**, and **RPSDAT:rpsctl.ism**.
- ▶ If you have customized Repository’s window library, you must reapply your customizations and rescript it.
- ▶ If your code uses the Repository Subroutine Library, you must recompile and relink against either **RPSLIB:ddlib.olb** or **RPSLIB:ddlib.elb**. We recommend that you link against **ddlib.elb**. See [“Upcoming Changes” on page 32](#) for additional information
- ▶ SDL (Synergy Data Language) files do not need to be converted.

Required Changes

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If you have version 6 repositories...	Convert them with the Repository conversion program (rpscvt.dbr) before using them with version 9.	New Repository. See “Using the Repository Conversion Program” on page 31 .
All	If you have your own copies of any of the record definitions contained within ddinfo.def ...	Check ddinfo.def for details of changes made and make the corresponding changes to your own copies.	We have modified the ddinfo.def file used by the Repository Subroutine Library. This change may break your code.
All	If you use rpsutl.dbr and test the status of .new files to determine if the load succeeded...	Test the exit status from rpsutl.dbr instead.	If an error occurs during the load process, rpsutl.dbr will now delete any repository files it created. This change may break your code.
All	If you use templates...	Generate your repository to a schema and then reload it.	This is required to ensure that all current template attribute values have been properly propagated to their descendants.

Using the Repository Conversion Program

Follow the instructions below to convert version 6.1.x or 6.3.x definitions to version 7 format. Version 7 format is used by Repository versions 7, 8, and 9.

1. Change to the RPS directory and enter
`dbx rpscnavt`
2. Press RETURN to continue after reading the program messages.
3. Enter the filenames of the old (version 6.1 or 6.3) repository main and text files that you want to convert. The filenames can include a full path or logical.

4. Enter filenames for the new (version 7) repository main and text files to be created. They must be different than the version 6 filenames. The filenames can include a full path or logical.

An error message will display on the information line if any file open errors occur.

While the conversion program is running, definition names are listed on the screen, and a log file, **RPSCNAV.LOG**, is created to log the conversion activities.

5. Repeat steps 1 through 4 for each version 6 repository you want to convert. If you want to save the conversion log file, **RPSCNAV.LOG**, rename the file each time after you run **rpscnavt**.

Patching Repositories Created with Version 6.1.0

If any of the repositories that you converted to version 7 format were *created with version 6.1.0*, follow the instructions in this section. You do not need to follow these instructions if the repository was created with a pre-6.1.0 version and then *converted to 6.1.0* with **ddcnvt.dbr**.

In repositories created with version 6.1.0, fields and templates of Date or Time type that reference a pre-defined format are missing their format strings. This affects ReportWriter reports and input windows that reference those fields.

If you're not sure whether a particular repository needs to be patched, do this:

1. In Repository version 9, go to Utilities > Set Current Repository and set the current repository to point to the repository in question.
2. Create a template with a type of Date.
3. Select Template Functions > Edit Display Information.
4. Put your cursor in the Format name field and press F7.
5. When the list of pre-defined Date formats is displayed, select one.
6. Look at the read-only field displayed next to the Format name field. If it contains the bracketed format string, this repository is OK. If it is blank, *this repository must be patched*.

To patch a repository, do this:

1. Using version 9, run the Generate Repository Schema utility on the repository to be patched.
2. Load the schema into a *new* repository using the Load Repository Schema utility. *Do not merge it into an existing repository.*

Upcoming Changes

In an upcoming version of Synergy/DE, we are planning to remove the Repository object library (OLB) file **ddlib.olb**. It has been replaced with the executable library (ELB) file **ddlib.elb**. (The OLB file will be available on the Synergex web site.)

You should be linking against the ELB file instead of the OLB file for the following reasons:

- ▶ An ELB will reduce the size of your application if you link multiple programs against the library.
- ▶ Linking against an ELB is more reliable than rebuilding the OLB and making a custom ELB.
- ▶ An ELB provides more compatibility when multiple Synergy applications from different companies coexist on one system.
- ▶ In most cases, you can upgrade to newer versions of Synergy/DE without recompiling and relinking your applications.

ReportWriter

Compatibility Information

Repository files used by ReportWriter must be converted. All version 6 repositories must be converted to version 7 format. Version 7 format is used by Synergy/DE versions 7, 8, and 9. See [“Using the Repository Conversion Program” on page 31](#). No other conversion or rescripting is required.

xfServer

Compatibility Information

- xfServer 9 supports clients running Synergy/DE version 6.1 and higher.
- (Windows) We changed the default installation directory to C:\Program Files\Synergex\SynergyDE.
- (Windows) If you have previously had xfServer, and you install version 9 to a different directory, you must remove and re-add your xfServer services.

Required Changes

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If you access xfServer data on the xfServerPlus port...	Reconfigure one of these services to run on a different port.	xfServer data access on the xfServerPlus port is no longer supported. This change may break your code.
All	If you reference one remote file specification from another when the first file specification is on Windows...	Cease to do so.	This is no longer permitted because of the potential for data corruption.
Windows	If you use the Synergy Service manager utility (servbox.exe)...	Use the Synergy Configuration Program instead.	The Synergy Service manager is no longer distributed. The Synergy Configuration Program provides a graphical user interface for configuring and starting xfServer.
Windows	If you create files without specifying a path...	Set the environment variable RSFILPATH to specify a location for files without a path.	If RSFILPATH is not set, the files are put in the directory that contains documents common to all users on the specific operating system. For example, on Windows XP, they are put in C:\Documents and Settings\ All Users\Shared Documents. See RSFILPATH in the “Environment Variables” chapter of <i>Environment Variables and System Options</i> .

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
Windows	If the RUSER environment variable is set in both the registry and the environment...	Be aware that RUSER set in the environment now has precedence over RUSER set in the registry.	When RUSER is set in the Synrc node in the registry, as is required to configure xfServerPlus for remote data access, it is read into the environment when rsynd starts up, and it is therefore considered to be set in the environment. This change may break your code.
Windows	If RUSER is set to the user name "SSPI" and no password is specified...	Change the user name.	To support Windows authentication security, RUSER recognizes "SSPI" as a special value.

xfServerPlus

Compatibility Information

- ▶ xfServerPlus version 9.3 can receive routine calls from version 7.1 and higher xfNetLink clients.
- ▶ Synergy Method Catalog (SMC) files must be converted for use with xfServerPlus and Workbench 9.3 and with version 9.3 utilities such as the Method Definition Utility (MDU) and `smc_elb.exe`. See the “SMC conversion utility” section of the xfServerPlus release notes (**REL_XFPL.TXT**) for instructions.
- ▶ (Windows) We changed the default installation directory to C:\Program Files\Synergex\SynergyDE.
- ▶ (Windows) If you have previously had xfServerPlus, and you install version 9 to a different directory, you must remove and re-add your xfServerPlus services.

Required Changes

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If you access xfServer data on the xfServerPlus port...	Reconfigure one of these services to run on a different port.	xfServer data access on the xfServerPlus port is no longer supported. This change may break your code.
All	If you use the default port 2330...	You must explicitly specify 2330 as the port, or use the new default, 2356.	We changed the default port from 2330 to 2356 so that it would not conflict with the default port for xfServer.
Windows	If you set SFWINIPATH on your xfServerPlus machine...	Be aware that db s will now read the synergy.ini file.	The service runtime (db s.exe) will use the synergy.ini file when SFWINIPATH is set. If SFWINIPATH is not set, db s ignores synergy.ini . This change may break your code.
Windows	If the RUSER environment variable is set in both the registry and the environment...	Be aware that RUSER set in the environment now has precedence over RUSER set in the registry.	When RUSER is set in the Synrc node in the registry, as is required to configure xfServerPlus for remote data access, it is read into the environment when rsynd starts up, and it is therefore considered to be set in the environment. This change may break your code.
Windows UNIX	If you use the xfServerPlus log file...	Expect to see multiple log files.	By default, xfServerPlus now generates a separate log file for each session. You can set the XFPL_SINGLELOGFILE option to ON in the xfpl.ini file to continue producing a single log file.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
UNIX	If you start <i>xfServerPlus</i> with a password...	Encode the -u password with setruser .	To improve <i>xfServerPlus</i> security, a clear text password is no longer permitted when starting <i>xfServerPlus</i> . This change may break your code if you start <i>xfServerPlus</i> with a script.
UNIX	If you start <i>xfServerPlus</i> without a password...	The user starting <i>xfServerPlus</i> must be root or signed on as the user specified with -u .	This is the documented behavior, but it was not previously enforced. This change may break your code if you start <i>xfServerPlus</i> with a script.
OpenVMS	If you run <i>xfServerPlus</i> on OpenVMS...	The <i>xfServerPlus</i> account must have the SHARE privilege.	Without this privilege, <i>xfServerPlus</i> will not run. This change may break your code.

xfNetLink Java Edition

Compatibility Information

- ▶ (Windows) We changed the default xfNetLink Java installation directory from C:\xfNetLnk to C:\Program Files\Synergex\xfNLJava.
- ▶ (UNIX and OpenVMS) We changed the name of the default xfNetLink Java installation directory from xfNetLnk to xfNLJava.
- ▶ xfNetLink Java version 6.3 is not compatible with xfServerPlus version 9.

Required Changes

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If you have your classpath set to point to the xfNetLink Java classes...	Update your class path to include the Xerces parser files (xercesImpl.jar and xml-apis.jar) and the directory where xfNetLink Java was installed.	The xfNetLink classes are now distributed in a .jar file. Additional functionality added in versions 7.5 and 8.1.7 requires that you include the Xerces parser files and the installation directory in the classpath. See " Setting the Classpath " in the "Creating Java Class Wrappers" chapter of the <i>Developing Distributed Synergy Applications</i> manual.

Notes:

Starting with version 7.5, xfNetLink Java includes utilities that enable you to create Java class wrappers from your SMC and Repository definitions, and then use the class wrappers to build a Java JAR file. Rather than use the xfNetLink Java classes directly, we recommend that you use the class wrappers feature to take advantage of all the latest xfNetLink Java enhancements.

Upcoming Changes

The port number range feature (minport and maxport) will be deprecated in a future version. We recommend that you quit using it.

SQL OpenNet

Compatibility Information

All Connectivity Series products for client and server must be upgraded to version 9.3. Otherwise, a version mismatch error occurs.

Required Changes

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
Windows	If you rely on the installation program to register and start SQL OpenNet...	Use the Synergy Configuration Program instead to register, start, and stop SQL OpenNet.	The SynSQL service is not registered and started by the installation program.

Notes:

The Vortex service is no longer used. A new service, Synergy/DE OpenNet Server (**SynSQL**), is used instead.

xfODBC

Compatibility Information

- ▶ In version 7.3, xfODBC Driver was replaced with a new product, xfODBC.
- ▶ All Connectivity Series products for client and server must be upgraded to version 9.3. Otherwise, a version mismatch error occurs.
- ▶ You must re-create your system catalogs for use with version 9.3 by running the **dbcreate** utility with the **-c** option (or by using the “Clear and re-create” option in the DBA program).

Required Changes

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If you use SQL_REAL for the CAST and CONVERT functions...	Use SQL_FLOAT instead.	xfODBC no longer supports SQL_REAL.
All	If you use xfODBC Driver DSNs...	Create new DSNs for use with for xfODBC.	xfODBC cannot use DSNs created for xfODBC Driver. This change may break your code. For details on creating new DSNs, see “Setting Up Access with DSNs” in the “Configuring Data Access” chapter of the <i>xfODBC User’s Guide</i> .
All	If you use the Appended to Connect String field in the xfODBC Setup window for the network connect string...	Use the other fields in the xfODBC Setup window to specify that information instead.	xfODBC creates the network string from the information in the xfODBC Setup window. See the “Configuring Data Access” chapter of the <i>xfODBC User’s Guide</i> for more information.
All	If you use the SODBC_NOROLL or VORTEX_ODBC_DATETIME environment variables...	Cease to do so.	These xfODBC Driver environment variables are not used by xfODBC.
All	If you use multidimensional arrays...	Rewrite your queries to use the current array name format.	We changed the way multidimensional arrays are named. This change may break your code.
All	If you use the Repository Converter (sodbccnv)...	Use dbcreate instead.	The sodbccnv utility has been replaced with dbcreate .
All	If you rely on the default display formats for date/time, dates, or times...	Modify your code to use the new defaults or use the TO_CHAR scalar function to change the default format. (See SSQL.TXT for details.)	The default formats have changed. The new defaults are as follows: YYYY-MM-DD HH:MI:SS (date/time) YYYY-MM-DD (dates) HH:MI:SS (times)

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If you want the column names in the system catalog to consist solely of the field names...	Set the environment variable SODBC_NOGROUPNAME.	By default if the field is part of a group in the repository, the column name created in the system catalog will include the group name as a prefix to the field name.
All	If your data includes unsigned fields that contain negative values...	Change the fields to signed fields by selecting the "Negative allowed" option in Repository. Or, you can set the SODBC_NOUNSIGNED environment variable to revert to the previous behavior.	The catalog creation utility (dbcreate) and xfODBC driver now distinguish between signed and unsigned fields. This prevents negative values from being stored in non-negative fields, which enables xfODBC to work better with ADO.NET and Microsoft Access. Refer to the "xfODBC" section of the Connectivity Series release notes (REL_CONN.TXT) for more detailed information on this topic.
Windows	If you use DBA for an owner name for a table such as DBA.TABLENAME...	Specify DBA in double quotes, such as "DBA".TABLENAME.	Using DBA as a name for a table is no longer supported because DBA is now a reserved name.

Notes:

Refer to the [xfODBC User's Guide](#) for additional information about the new xfODBC and for a tutorial to help you learn how to set up and use xfODBC.

SQL Connection

Compatibility Information

- All Connectivity Series products for client and server must be upgraded to version 9.3. Otherwise, a version mismatch error occurs.
- If you use SQL Connection with SQL Server, we recommend that you install Service Pack 4 for SQL Server 2000. This service pack contains several Synergy/DE-specific fixes.

Required Changes

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If you use the %SSC_BLOB routine...	Use the new routine %SSC_LARGECOL instead.	The new routine and option improve SQL Connection's ability to get and put large binary column data and large character column data. See %SSC_LARGECOL in the "Database Functions" chapter of the <i>SQL Connection Reference Manual</i> for details.
All	If you use the SSQL_BLOBCOL option for %SSC_OPEN...	Instead, use the new SSQL_LARGECOL option for %SSC_OPEN for gets and the SSQL_LARGECOL option for %SSC_EXECUTE for puts.	
All	If you don't want to use encryption...	Remove the -k option from the vtxnetd or vtxnet2 command line and remove the key_connect setting from the net.ini file.	By default, the database user name and password and the host user name and password are encrypted. For more information, see "The vtxnetd and vtxnet2 Programs" in the "Configuring Connectivity Series" chapter of the <i>Installation Configuration Guide</i> and "Setting connect string defaults and encryption in net.ini" in the "Preparing Your Environment" chapter of the <i>SQL Connection Reference Manual</i> .
All	If your program .INCLUDEs the ssql.def file...	Update your program to .INCLUDE ssql.def from the connect directory using CONNECTDIR.	We moved ssql.def from the synsqlx directory to the connect directory.
All	If you have created a net.ini file...	Add the settings from your net.ini file to the distributed net.ini file.	We now distribute a net.ini file in the connect\synodbc\lib directory. If you have a net.ini file in that same directory, it will be overwritten when version 9 is installed.

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
All	If you use numeric fields for dates...	Change your code to use ^A to access the fields as alpha fields. Or, in your database, overlay the fields with alpha fields, and then access data with the overlay fields. Or, change your code to handle Julian day values.	When numeric date fields are bound or defined, they are treated as Julian days, which are calculated using a default time base of -1721378. (Formerly, they were treated as alpha fields with YYYYMMDD formats.) This change may break your code. Note that you can use SSQL_OLD_ZONEDATE, an %SSC_CMD option, and SSQL_OUTDATE, an %SSC_EXECIO option, to disable the new date behavior for <i>retrieved</i> dates. These options do not, however, affect dates written to the database. In addition, using SSQL_OLD_ZONEDATE can cause increased CPU time and memory use.
All	If you rely on the default format for dates returned into defined alpha variables...	Change your code to handle the new default format or use SET OPTION DATETIME to change the default format.	The format for dates returned into defined alpha variables has changed from DD-MON-RR to DD-MON-YYYY. This change may break your code. See the SSQL.TXT file for information on SET OPTION DATETIME.
Windows	If you select the database driver during installation...	Use the Synergy Configuration Program to select the active database.	During installation, a default driver is selected for Oracle and SQL Server. You can change the default from the Synergy Configuration Program.
Windows	If you use vtxnet2 because you use VTX12_2000 or rely on child processes rather than threads...	Change the start-up command line in the opennet.srv file to specify vtxnet2 .	We changed the default SQL OpenNet server from vtxnet2 to vtxnetd . This change may break your code.
Windows	If you use VTX12 or the sqlserver keyword in your connect string...	Update your program to use VTX12_ODBC or change the default driver in the Synergy Configuration Program to continue using a DB-Library-based driver.	We changed the default database driver for SQL Server from a DB-Library-based driver to an ODBC-based driver (VTX12_ODBC). This is the driver that is used when a connect string specifies VTX12 or sqlserver. This change may break your code. For more information, see "Using your program with different drivers and databases" in the "Creating SQL Connection Programs" chapter of the <i>SQL Connection Reference Manual</i> .

Platform	Version 6 usage	Change required for 9.3	Explanation/Comments
UNIX	If you use connect strings that include a UNIX SQL OpenNet Server...	Alter the connect string to include the driver's program name.	See the “Configuring Connectivity Series” chapter of the <i>Installation Configuration Guide</i> and the <i>SQL Connection Reference Manual</i> for details on connect strings.
UNIX	If you execute sqlunixbld and makessqlsrv to add client support...	Cease to do so.	Client support is now pre-built.
OpenVMS	If you use connect strings...	Change the <i>driver_name</i> argument to a valid driver.	The <i>driver_name</i> argument for connect strings is no longer just a placeholder. Now, it must be set to the name of the database driver. For a list of database driver names, see “Connect string syntax” in the <i>“Creating SQL Connection Programs”</i> chapter of the <i>SQL Connection Reference Manual</i> .

Migrating from Version 7 to Version 9.3

This section is designed to help you migrate from version 7.1, 7.3, or 7.5 to version 9.3.

Some items in the Version 7 usage columns in the Required Changes tables include a version number. This indicates the version to which that item pertains. That is, if the version number is 7.1, users migrating from 7.1 to 9.3 will need to make the required change. However, users migrating from version 7.3 or 7.5 to 9.3, will not need to make the change. If no version is specified, the item applies to all version 7 users.

License Manager

Compatibility Information

You must reconfigure all products for Synergy/DE version 9.

Required Changes

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
Windows	If you run License Manager from the License Configuration option on the Start menu... (7.1)	Use the Synergy Configuration Program instead.	The menu option has been removed.

Synergy Language

Compatibility Information

- ▶ Version 9 will run code compiled and linked with Synergy/DE 6.1 and higher; you do not *have* to recompile.
- ▶ You may *want* to recompile, however, to take advantage of new optimization techniques and some fixes in version 9 (as noted in the table below). You may choose to recompile some portions of your application now and other portions later. See also the compatibility information in “UI Toolkit” on [page 69](#).
- ▶ If you recompile, you will need to make changes to your code due to the stricter error checking in the version 9 compiler and the fact that we have strongly prototyped all Synergex-supplied subroutines and functions. In most cases, the version 9 compiler will find errors that earlier versions of the compiler did not. You can turn off most of the stricter error checking by compiling with the **-qrelaxed (/RELAXED** on OpenVMS) switch. For details on this switch and its options, see “[Compiler options](#)” in the “Building and Running Synergy Language Programs” chapter of *Synergy Language Tools*.
- ▶ To debug with version 9, you must recompile, because the version 9 debugger cannot show source line information or provide symbolic access to variables unless the program is recompiled. If you need to debug your code in version 9, but are not yet ready to take advantage of version 9’s stricter error-checking and strong prototyping, recompile with the compiler distributed as **dbl8**. Note that code compiled with **dbl8** can be run only on version 9.
- ▶ The object files, object libraries, and executables generated with version 9 are not backward compatible. However, changes you make to your code to comply with the enhanced error checking and strong prototyping will compile under previous versions of Synergy, unless stated otherwise in this document. In those few instances, you can use the IFDEF compiler directive so that the code can be compiled with either version 9 or a previous version. For example, let’s say you want to overload the subroutine PUTCM, which was defined like this prior to version 9:

```
.subroutine putcm  
data, a
```

Due to strong prototyping, the parameter *data* is defined thus in version 9:

```
in req data, a
```

Since the IN and REQ parameter modifiers are new in version 9, code that includes them will not compile in a prior version. You can handle this situation thus:

```
.ifdef DBLV9  
    subroutine putcm  
        in req data, a  
.else  
    subroutine putcm  
        data, a  
.endc
```

- ▶ (Windows) We changed the default installation directory to C:\Program Files\Synergex\SynergyDE. The new directory will be used for new installations. If you are upgrading, the installation will default to your current directory.
- ▶ On OpenVMS, due to increase memory requirements, you may need to change the value of some authorize parameters for developers who are using the version 9 compiler. See “[OpenVMS requirements](#)” in the “Requirements and Considerations” chapter of the *Installation Configuration Guide* for details and recommended values.

Required Changes

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you pass a variable of the wrong data type, such as an alpha variable to a routine that expects a decimal or integer variable...	Correct your code to use the correct data type. Optionally, you may be able to cast (^A , ^D , or ^I) the variable to the correct type.	Passing the incorrect data type causes the compiler to generate a "Type mismatch" error. The version 9 compiler strongly prototypes Synergex-supplied functions and subroutines, which means that your call must exactly match the Synergex-supplied routine in return type, and in number, direction, and type of parameters, as well as whether parameters are optional or required. You can reduce the occurrence of this error by using the -qrelaxed:param compiler option. In addition, the version 9 compiler strongly prototypes all user-defined routines in a compilation unit to ensure that calls within that unit match the routine declaration. For these routines, you can use the -qrelaxed:local compiler option to reduce the occurrence of this error.
All	If you pass more arguments to a routine than the routine defined...	Correct your code to pass the correct number of arguments. For user-defined routines, you can optionally do one of the following to correct your code: - Change the routine declaration to include the extra arguments. - Change the routine declaration to include the VARARGS modifier.	This situation now causes the compiler to generate a "More argument values than declared parameters in routine" error. See the item above for an explanation of strong prototyping in Synergex-supplied and user-defined routines and for the compiler options that will reduce the occurrence of this error. Note that if you use VARARGS, your code will no longer compile in versions 8.3 and earlier. For more information on VARARGS, see the Synergy Language Reference Manual.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you fail to pass required arguments to a routine...	Correct your code to pass the missing arguments.	Previously, this error would not be caught until runtime. Now, because the version 9 compiler strongly prototypes all Synergex-supplied routines, it causes the compiler to generate a "Missing required parameter <i>name</i> in routine <i>name</i> " error.
All	If your code includes external functions in which the function return type in the function declaration does not match the return type of the function itself...	Correct your code so that they match. In version 9 you can explicitly specify the type of the function return value. If it's not explicitly typed, the return type is determined by the data type of the first FRETURN statement. See FUNCTION-ENDFUNCTION in the "Synergy Language Statements" chapter of the <i>Synergy Language Reference Manual</i> for details.	Because the version 9 compiler strongly prototypes all routines in a compilation unit, this situation now causes the compiler to generate an "External function <i>name</i> does not match prototype" error. You can reduce the occurrence of this error by using the -qrelaxed:extf compiler option.
All	If you overload any Synergex-supplied function or subroutine...	Verify that your overloaded routine exactly matches the prototype of the Synergex-supplied routine in return type, and in number, direction, and type of parameters, as well as whether parameters are optional or required.	We now strongly prototype Synergex-supplied functions and subroutines. Using the -W4 compiler option (/WARNINGS=4 on OpenVMS) will show the prototype the compiler expects and your overloaded routine definition, so that you can see how they differ. Once you change your routine, your code may no longer compile in versions 8.3 and earlier. See "Compatibility Information" on page 46 for a suggestion on how to handle this situation.
All	If you use %ISINFO to return alpha data...	Use the new %ISINFOA routine instead or use ^A to cast the return type.	%ISINFO now returns only numeric data because strong prototyping requires that there be only one return type. If you use the new %ISINFOA routine, your code will no longer compile in versions 8.3 and earlier. For backward compiler compatibility, use ^A.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you use the %SYNSOCK routine...	Change your code to use the %SS_ functions instead.	Previously, the Synergy socket API routines were implemented as parameterized macros (%SS_), which invoked the %SYNSOCK routine. In version 9, the compiler treats the %SS_ routines the same as functions in order to support strong prototyping. Therefore, %SYNSOCK is no longer needed and has been removed from the runtime. See the “Synergy Socket API” chapter of the <i>Synergy Language Reference Manual</i> . This change may break your code.
All	If you use END (as opposed to .END) and expect all .DEFINES to be removed and unavailable to the next module in the source file...	Add .UNDEFINE compiler directives as necessary. For example, a .DEFINE might be used to control the initializing of a global data section: <pre>.ifdef GBL_INIT .undefine GBL_INIT global data section A ,init .else global data section A .endc</pre>	The compiler now treats END and .END the same. This means that .DEFINES are not undefined when END is encountered and are therefore available to the next module in the source file. This could cause a “Symbol already defined” error or lead to unexpected behavior because something is defined when it should not be. This change may break your code. To retain the previous behavior, in which END and .END were treated differently, use the -qrelaxed:end compiler option.
All	If you define a field that implicitly extends a record by overlapping another field... For example, <pre>record pbuff ,a80 page_hdr ,a5@pbuff page_text ,a85@pbuff+5</pre>	Correct your code. In this example, you could change the size of pbuff to 90.	This is often the result of a coding error. It is no longer supported, and the compiler now generates a “Cannot extend record by <i>nn</i> bytes with overlay field <i>name</i> ” error. Note that a field is permitted to extend a record as long as it does not overlap another field.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you have records that overlay commons or records, or if you have commons that overlay other commons...	Be aware that record and common overlays are now strictly enforced as to the type being overlaid. You may need to correct your code.	This situation causes the compiler to generate an "Overlay not allowed" error. See "Overlaying data" in the "Defining Data" chapter of the <i>Synergy Language Reference Manual</i> for a list of rules that apply to overlaying records.
All	If your code <code>.INCLUDEs</code> named records that have overlay fields, and you get an "Ambiguous symbol" error when compiling...	Correct your code to remove the ambiguity. The easiest way to do this is to use the NORECORD qualifier. For example, instead of this: <pre>.include "rec" repository, record=myRecord</pre> You can do this: <pre>record myRecord .include "rec" repository, norecord</pre> The other option is to remove the ambiguity by renaming the duplicate symbol.	In version 9, we changed the way in which <code>.INCLUDE</code> offsets are specified by the compiler. If the record is <i>named</i> , dot notation using the record, group, and field names is used. If the record is <i>unnamed</i> , a numeric offset relative to the overlay field is used. The use of the named reference could cause the compiler to generate an "Ambiguous symbol" error if you have a duplicate symbol elsewhere in your code. Using the NORECORD qualifier will ensure this does not happen. For more information, see "Including from a Repository" in the Discussion for <code>.INCLUDE</code> in the "Preprocessor and Compiler Directives" chapter of the <i>Synergy Language Reference Manual</i> .
All	If your code includes a data division in which the first item is a structure <i>and</i> there is no MAIN statement...	Add a MAIN statement.	MAIN is optional <i>unless</i> the first item in the data division is a STRUCTURE statement without a corresponding ENDSTRUCTURE. The error generated by the compiler will vary.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If your code contains references to ambiguous field definitions within the data division... For example, <pre>record red ,a3 record joe blue ,a ^size(red) red ,d2</pre> A similar problem occurs when you have a structure defined in a repository, and you reference that structure as an implicit group in another repository structure and name the group the same name as the referenced structure, and then include both structures in a program.	Correct your code to remove the ambiguity. In the first example, the correction required would depend on what the code was attempting to do. For the repository example, you need to either rename the first structure in the .INCLUDE directive to something unique, or change the name of the group within the second structure in the repository.	In version 9, ambiguous field references can be detected within the data division and cause the compiler to generate an "Ambiguous symbol" error. Previously, they could be detected only when referenced within the procedure division. To allow the old usage, use the -qrelaxed:path compiler option.
All	If you use anything other than an a6 for a GETRFA or RFA qualifier variable...	Change your code to use an a6 or set system option #56.	Prior to 9.3, a GETRFA or RFA qualifier variable that exceeded 6 bytes was allowed. Now, with the introduction of GRFAs, if the size of a GETRFA or RFA argument is anything other than 6 or 10 bytes, an "Invalid record's file address" error is generated. Setting system option #56 turns off this behavior and causes all RFAs that exceed 6 bytes to be treated as 6-byte RFAs. Even if you are using an a10, you will need to set system option #56, because a10 now indicates a GRFA and will cause a "Record not same" or "Feature not supported in this version" error when used for an RFA. This change may break your code.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If your code uses i8 data types as function return values on 32-bit systems...	Change your code to use a supported data type.	Using an i8 (or long) data type on a 32-bit platform now causes the compiler to generate a "Type i8 cannot be return value on 32-bit platform" error. Previously, there was no error, and if you passed a value larger than an i4 to an i8 , it was truncated. This change may break your code.
All	If you use the LOCAL, STATIC, or STACK qualifiers with the MAIN statement...	Remove them from your code.	MAIN no longer allows these qualifiers. Previously, although allowed, they were ignored and did not affect how your code operated. Now, the compiler generates an "Invalid MAIN declaration" error.
All	If your code uses FIRST or LAST with the I/O statements FIND, READ, or WRITE...	Use ^FIRST or ^LAST instead.	The use of these qualifiers without the caret is no longer supported and causes the compiler to generate an "xxx not found" error.
All	If your code uses short forms of identifier names, such as NOPREC for NOPRECISION...	Use the full documented identifier names.	We no longer support shortened forms of identifier names. Using them causes a compiler error.
All	If your code includes the START compiler directive without a period at the beginning...	Add the period.	All compiler directives now require a period at the beginning. Without the period, the START directive will be ignored.
All	If your code uses non-numeric variables as subscripts or in ranging operations... For example, <code>clear myvar(avar)</code> Or <code>clear myvar(offset:avar)</code>	Correct your code to use whole numbers instead.	Non-numeric variables are not allowed in subscripts or ranging operations and cause the compiler to generate a "Numeric expression required" error.
All	If your code uses implied decimal variables for channel numbers, subscripts, or in ranging operations...	Correct your code to use whole numbers instead.	Previously, implied decimal values were rounded to the nearest whole number. Now, they cause the compiler to generate a "Decimal or integer expression required" error.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If your code includes decimal fields in which the whole number portion exceeds 18...	Correct your code so that the whole number portion is 18 or less.	Although this was previously allowed, only assignment statements were guaranteed to work, and it was treated as an 18-digit field with leading zeros in the inaccessible part of the field. This now causes the compiler to generate an "invalid data size specification" warning. To allow the old usage, use the -qrelaxed compiler option.
All	If your code includes implied decimal fields, in which the difference between the whole number part of the field and the precision part is greater than 18...	Correct your code such that the difference is 18 or less.	The compiler now generates a BIGNUM error in this condition.
All	If your code includes IF statements that are missing the body statement... For example, <pre>if (condition) then nop else end</pre>	Correct your code to include a body statement.	This syntax is no longer permitted and causes the compiler to generate a "Statement expected" error.
All	If your code includes a statement on the same line as a BEGIN statement...	Move it to the next line.	This coding style is no longer permitted and causes the compiler to generate a "BEGIN should be followed by end of line" error.
All	If your code has function declarations with multiple qualifiers... For example, <pre>.function myFunc, ^val reentrant</pre>	Verify that there is a comma between the qualifiers.	The comma is now required. A missing comma causes the compiler to generate an "Invalid expression" error.
All	If you use .IF with a compile-time expression using ^SIZE or %LEN...	Correct your code.	This is no longer supported and causes the compiler to generate a "Function name is not supported in .IF" error.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you use the following syntax: <pre>record array , [10]a5 array[] (ix:1) = value</pre>	Change your code. In this example, you can add a GROUP/ENDGROUP pair around the array declaration: <pre>record group array_grp array , [10]a5 endgroup</pre> and then change the code to <pre>array_grp(ix:1) = value</pre>	This syntax is no longer supported and causes the compiler to generate an "Invalid expression" error.
All	If your code uses brackets [] to access the elements of a pseudo array...	Use parentheses () instead.	Brackets can be used only for accessing the elements of real and dynamic arrays. Using them to access the elements of a pseudo array causes the compiler to generate an "Incorrect number of dimensions" error.
All	If your code includes arrayed fields larger than 256 MB...	Change your code.	Arrayed fields larger than 256 MB are no longer supported and cause the compiler to generate a "Size of field <i>name</i> too large" error.
All	If you use parentheses to reference elements of a structure in a ^M reference... For example, <pre>^m(quote_lines(ix), rec)</pre>	Change your code to use brackets instead.	This syntax is no longer supported and causes the compiler to generate an "Indexing not allowed in structures" error (error number 683).
All	If your code includes parameterized macros that have empty parameter lists... For example, <pre>.define myMac() %char(8)</pre>	Remove the empty parentheses.	Empty parentheses are no longer permitted when no parameters are specified in a .DEFINE parameterized macro reference and cause the compiler to generate an "Invalid compiler directive" error.
All	If your code includes nested parameterized macros of the same name...	Change your code.	In version 9, nested use of parameterized macros with the same name will cause the compiler to generate a "<i>name</i> not found" error.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you have inadvertent unary + operations on alpha operands... For example, <pre>xcall fred ("ABC"+ & +"DEF")</pre> Or <pre>xcall fred ("ABC" &, +"DEF")</pre>	Correct your code to remove the unnecessary unary +.	This operation is mathematically invalid and is most often the result of a coding error. It causes the compiler to generate a "Numeric operand required (unary +)" error. To allow the old usage, use the -qrelaxed compiler option.
All	If your code uses the dollar sign (\$) on function calls...	Change it to a percent sign (%).	The dollar sign is no longer supported and causes the compiler to generate an "Invalid calling convention" error. To allow the old usage, use the -qrelaxed:deprecate compiler option.
All	If your code uses \$A or \$D...	Change the dollar sign to a caret sign (^A, ^D).	
All	If you use .DEFINE to simulate a Boolean operator... For example, <pre>.define either or . . . if var1 .either. var2 nop</pre>	Correct your code.	This syntax is no longer supported and causes the compiler to generate an "Invalid expression" error.
All	If you use an old form of the .PROC compiler directive that specifies the number of I/O buffers on RSTS... For example, <pre>.proc (4)</pre>	Remove the number from your code.	This is no longer supported and causes the compiler to generate an "Invalid record field" error.
All	If your code passes a numeric non-implied literal to a routine that declares it as n and uses ^D, ^F, or ^A on the passed argument without first determining its data type...	Make the argument type n . or use ^DATATYPE to determine the argument type before casting it.	Previously, literals were passed as decimals. Now, they are passed as integers, so the results will not be as expected.
All	If your code passes the <i>line</i> argument to MODNAME and declares it as a ...	Declare it as n instead.	Previously, <i>line</i> could be a or n ; now, only n is permitted.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If your code uses ^PASSED on a local variable...	Change your code.	Previously, this always evaluated to true. Now, it causes the compiler to generate an "Invalid argument" error because ^PASSED is intended for parameters only.
All	If you have files with more than 262,143 lines in them...	Split them into smaller files.	Previously, line numbers in such large files were misreported in errors. Now, the compiler generates an "Exceeded maximum number of lines in file <i>name</i> " error.
All	If you have duplicate global data sections, global commons, or global literals that are not the same size...	Change your code.	In version 9, the linker will generate an error when it finds duplicate symbols that are not the same size. The exact error depends on which symbol is duplicated. (Duplicate symbols that <i>are</i> the same size will cause the linker to generate a warning.)
All	If your code includes decimal and implied decimal variables in a single SET statement...	Correct your code to use separate SET statements for each data type.	Previously, the precision would be truncated when setting the decimal variable, without an error being reported. Now, the compiler generates a "Type mismatch between <i>type</i> and <i>type</i> " error.
All	If you use the DBLTOKENS environment variable...	None.	DBLTOKENS is not used by the version 9 compiler, so this environment variable has been retired.
All	If your code passes a ^VAL parameter to a ^REF argument in order to use the result of a call to a system DLL that returned a pointer...	Access the pointer using %MEM_PROC(DM_REG) instead.	Passing a ^VAL to a ^REF argument is not supported in version 9 and will cause the compiler to generate a "^REF required for argument <i>name</i> " error.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If your code includes a relative range specification that accesses characters preceding the specified field...	Correct your code to range within the bounds of the field.	Attempting to access characters that precede the specified field now causes the compiler to generate an "Invalid range specified" error. Although this type of access has been deprecated, the Synergy Runtime will continue to support it for backward compatibility. We recommend that you compile and run with -qcheck and make the necessary changes to avoid future memory access violations.
All	If you specify invalid MATCH, DIRECTION, POSITION, or LOCK I/O statement qualifiers...	Correct your code.	Invalid MATCH, DIRECTION, POSITION, and LOCK values now signal an E_OUTRNG error. This change may break your code.
All	If you .INCLUDE repository fields that are within nested groups in which all group levels have prefixes...	Modify references to those fields in your code to include the additional prefixes.	Previously, the compiler was appending only the lowest level prefix to the field, making it possible to have non-unique field names. It now concatenates the prefixes from all levels to ensure that all field names are unique.
All	If you are experiencing incorrect line numbers in the debugger...	Recompile to take advantage of a fix for this problem.	In 8.1.3, we fixed a bug that occurred when a program .INCLUDEd two files, and the first file contained only one line.
All	If you have programs that do not have the .PROC statement in the primary source file...	Recompile to take advantage of a fix for this problem.	This situation may cause incorrect line numbers in a traceback.
All	If you do string concatenation on a 32-bit system where the result could be greater than 65,535 bytes...	Modify your code to trap for the error \$ERR_BIGALPHA.	Previously, we did not throw an error in this situation. This change may break your code.
All	If you use any of the definitions in object.def and were acquiring them by .INCLUDEing activex.def ... (7.5)	Explicitly .INCLUDE object.def .	We no longer .INCLUDE object.def within activex.def because it is not needed for the ActiveX interface.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you use hierarchies in the Synergy namespace API...	Modify your code to either not pass the <i>parent_id</i> argument to %NSPC_OPEN or pass it with a value of 0.	Namespace hierarchies are no longer supported. This change may break your code. In version 9, the "Synergy namespace API" was renamed the "Synergy symbol table API" to avoid confusion with namespace support for objects.
All	If you use the Synergy HTTP document transport API... (7.5)	Recompile and relink with version 9.	We moved the HTTP document transport API into the Synergy runtime. This means that you no longer need to link with synxml.elb to use this API. In making this change, we modified all HTTP API routines to be macros that invoke the F_SYNHTTP runtime routine.
All	If you use the Synergy HTTP document transport API, and you have implemented your own escape mechanism for special characters... (7.5)	Set the HTTP_NOESCAPE environment variable to any value to continue using your own escape mechanism.	The HTTP document transport API now automatically escapes certain characters in URIs. If you have implemented your own escape mechanism, this change may break your code.
Windows	If you pass more than nine arguments with %AX_CALL...	Remove the characters < and > that were used to pass the additional arguments.	%AX_CALL is no longer a macro. It now supports an unlimited number of arguments, so the <> characters are no longer required. Using <> causes the compiler to generate an "Invalid expression" error.
Windows	If your synergy.ini file is in the Windows directory (the default location) or the dbl directory...	Move it to another directory (outside the synergyde directory tree) and set SFWINIPATH to point to that directory.	A synergy.ini file is no longer created during installation, nor is an existing file updated. A default synergy.ini file (containing entries referencing %SYNERGYDE%) is now distributed in the dbl directory. This file is removed on an uninstall or upgrade.
Windows	If you have previously installed and uninstalled Synergy/DE or Synergy/DE Client...	Edit your PATH to remove any extraneous references to the dbl\bin and dbl\connect directories.	Previous versions did not always remove PATH modifications during uninstallation.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
Windows	If you have installed the “Files for shared installation” and have customized the setup.ini file in the synergyde\client directory... (7.5)	Make a copy of the command line (CmdLine=) in your setup.ini file before upgrading the shared machine, and then merge your changes into the new setup.ini file.	Customized setup.ini files in the synergyde\client directory will be lost when upgrading from version 7.5.
Windows	If you are using a local synergy.ini file on your Synergy/DE Clients... (7.5)	Remove the system-specific settings from the local synergy.ini file and place them in a synergy.ini file on the shared machine. Set SFWINIPATH on each client to point to the synergy.ini file on the shared machine. For each client, save the user-specific settings from that client’s synergy.ini file in a file named synuser.ini and place it in the local application data directory on the client machine.	Synergy/DE now supports a “system-specific” initialization file (synergy.ini) and a “user-specific” initialization file (synuser.ini). Settings in synuser.ini override settings in synergy.ini . See “ Synergy initialization files ” in the “Environment Variables” chapter of <i>Environment Variables and System Options</i> for more information.
Windows	If you use the Windows printing API or the LPQUE statement... (7.1, 7.3)	Expect \$ERR_NOTAVL instead of \$ERR_FNF when the printer device is incorrect.	We changed the error value returned.
Windows	If you use the Windows printing API, do not specify an extension for the metafile, and then reference the metafile later... (7.1, 7.3)	Use an .emf extension when referencing the metafile.	Previously, when no extension was specified, .spf or .ddf was added. Now, .emf is the default extension. This change may break your code.
Windows	If your code uses the DWP_ADDFILE function and your application provides the preview function...	We recommend passing the enhanced_preview flag to DWP_BEGINJOB.	Truncation of long lines may occur in the previewer if this flag is not set.
Windows	If you use the PRINT_METHOD environment variable (PRINT_METHOD= escape) and change the printer driver default values... (7.1, 7.3)	To continue using the changes to the default values, you must code them yourself within the data.	We now use the correct Windows API to send data directly to the printer. Changes you make to the printer driver default values will be ignored.
Windows	If you use %WPR_GETDEVICE to obtain the printer name... (7.1, 7.3)	Use the <i>extended_pr_name</i> field instead of the <i>printer</i> field.	The <i>extended_pr_name</i> field supports 256 character printer names, as required for Windows 2000 and higher.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
Windows	If you expect a return value of 0 from the %TTSTS routine to indicate the channel is not open to a terminal or input is redirected...	Modify your code to expect a return value of 1.	The %TTSTS routine now returns 1 as documented in this situation. This change may break your code.
Windows	If you use the PARSE routine to parse UNC path specifications...	Modify your code to reflect that the \\server\share_name portion of the UNC path specification is returned in the device argument.	Previously, this information was returned in the <i>directory</i> argument. This change may break your code.
Windows	If you have programs that depend on AX_INPUT never timing out... (7.1)	Specify D_FOREVER for the new wait_time argument.	The <i>wait_time</i> argument sets the time-out value for an ActiveX control. This change may break your code. For more information, see AX_INPUT in the “ActiveX Routines” chapter of the <i>UI Toolkit Reference Manual</i> .
Windows	If you use the Synergy Language profiler and want to calculate elapsed CPU time using the processor time...	Set the environment variable PROFILE_PROCESSOR_TIME.	By default, the profiler now calculates elapsed CPU time using the high-granularity system clock. Set the aforementioned environment variable to revert to the previous behavior. For more information, see PROFILE_PROCESSOR_TIME in the “Environment Variables” chapter of <i>Environment Variables and System Options</i> .
Windows	If you rely on the default Synergy runtime color definitions...	To maintain the same color scheme, define the specific colors you want the application to use in your synergy.ini file.	We removed the Synergy color and palette definitions from the distributed synergy.ini file in version 8.3. The new default colors are slightly different than they were previously. Consequently, you may notice a change in the appearance of applications that rely on the default colors.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
Windows	If your code uses i1 variables to store the color values returned by the WI_PALET subfunction to W_INFO...	Change these variables to at least an i2 or d3 . (We recommend using an i4 .)	To accommodate the increased number of Synergy colors in version 8.3, the WI_PALET subfunction to W_INFO may now return values that are too large for i1 variables. This change may break your code.
Windows UNIX	If you rebuild the Synergy runtime...	Create a DLL or shared object containing your C routines and access it via the Synergy DLL API.	The makedbr and makedbs files are no longer distributed. The DLL API was enhanced in version 8.1 to assist you in creating C DLLs and shared objects that interact with the runtime's C interface.
Windows UNIX	If the input file, output file, or temporary file for client/server sorting reside on different machines... (7.1)	Move these files as necessary so that they all reside on the same machine.	Synergy sort for client/server now handles all I/O on the target machine. All file specifications must include the same host name or omit the host name. This change may break your code.
Windows UNIX	If you're using REV 2 or REV 3 ISAM files...	Convert your ISAM files to the REV 4 level or patch them to the REV 5 level.	Although REV 2 and REV 3 files are still supported, extra overhead is required to maintain them. See "Converting to ISAM REV 4 or Patching to REV 5" on page 66 .
Windows UNIX	If you need to create files that will be recognized by Synergy/DE versions prior to 6.3.0d...	Set the environment variable ISAMC_REV to 3 for each Synergy user who may be creating a file.	Versions 6.3.0d and higher support REV 4, which became the default value for ISAMC_REV starting with version 7.1. For more information, see ISAMC_REV in the "Environment Variables" chapter of <i>Environment Variables and System Options</i> .
Windows UNIX	If you use ISAM files that are named with a trailing period and no extension (e.g., myfile.)... (7.1, 7.3)	Rename the index file to filename.ism and the data file to filename.is1 . You may also need to change the code that generated the file to remove the trailing "." or explicitly add an extension.	Opening an ISAM file with no extension now generates an E_FILSPEC error. This change may break your code.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
Windows UNIX	If you have any source modules that contain a global data section that doesn't have an ",INIT" but does contain an .ALIGN directive, and the object file is linked into an ELB...	Recompile to take advantage of a fix for this problem.	The described scenario caused unpredictable results because the compiler erroneously emitted data over other pieces of compiler information.
Windows UNIX	If you have duplicate global data sections, global commons, or global literals that are not the same size...	Change your code.	In version 9, the runtime will detect duplicate symbols when ELBs are loaded and generate a BADEL error if the size is different. The error text tells you which symbols are duplicated. (Duplicates that <i>are</i> the same size will cause a warning in debug output.) This change may break your code.
Windows UNIX	If you have duplicate static records that are not the same size...	Recompile and relink.	
Windows UNIX	If you use READ or READS from a relative file when the file record size is greater than the size of the data area...	Correct your code or set system option #27.	This situation now generates an "Input data size exceeds destination size" error, as documented in the <i>Synergy Language Reference Manual</i> . This error can also occur when a file is opened in the default mode with RECSIZ set. This change may break your code.
Windows UNIX	If you call the COPY routine and specify a destination file that another process might have open...	Modify your code to handle a possible "File in use" error.	A "File in use" error is now returned if a COPY is performed on an open file. This change may break your code.
Windows UNIX	If you use %DLL_SUBR...	Change your code to use %DLL_CALL instead.	%DLL_SUBR has been removed. This change may break your code.
Windows UNIX	If you use %DATETIME and the resolution on your system for milliseconds is not six digits...	Review your code to see if it needs to be modified.	%DATETIME has been changed to fill with zeroes instead of spaces if the system resolution for milliseconds is not six digits.
Windows UNIX	If you use the FILENAME option of the SDMS_ISINFO routine in the Synergy Database API...	Recompile your program with the new sdms.h header file.	Failure to recompile will result in a DE_INVOPT error. This change may break your code.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
UNIX	If you use the FORK subroutine or the optional <i>fork</i> argument of the %HTTP_SERVER_RECEIVE function...	Change your code to reopen any files needed by the child process.	As a result of a change we made to fix random I/O failures, the FORK subroutine and the <i>fork</i> argument to the %HTTP_SERVER_RECEIVE function now close all channels opened by the parent in the forked child process. This change may break your code.
UNIX	If you pass Windows-style commands that contain "\\" to the UNIX shell...	Use "\\\\" instead.	The SHELL subroutine now correctly handles shell environment variables (\$BLDIR). This change may break your code.
UNIX	If you use %TNMBR...	Examine your code to ensure the value returned by %TNMBR is not being used for static data.	On some UNIX systems, %TNMBR was returning the wrong value. This change may break your code , as it may change the terminal numbers returned on a particular system.
UNIX	If you use the JBWAIT environment variable...	None.	XCALL RUNJB no longer waits 2 seconds (or the value of JBWAIT). Hence, the JBWAIT environment variable is no longer necessary and has therefore been retired.
UNIX	If your application takes advantage of the fact that file locking between channels within the same process is not enforced...	Set the INTRAFILELOCKS environment variable to 0 (zero).	We changed the default behavior for file locks within a process; it is now the same as the behavior for file locks between processes. This change may break your code. You can revert to the previous behavior by setting INTRAFILELOCKS to 0. We do not recommend this, although it does permit you to continue executing version 8.1.7 and earlier applications that violate these new locking rules. For more information, see INTRAFILELOCKS in the "Environment Variables" chapter of <i>Environment Variables and System Options</i> .

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
UNIX OpenVMS	If you need to use 8-bit characters (128-255)... (7.1, 7.3)	Define a LOCALE or set the LANG environment variable for your operating system.	The correct uppercase and lowercase characters in the 8-bit range (international characters) are now read from the system LOCALE. Should you want to revert to the previous behavior, where A-Z and a-z with the 8th bit set were considered characters (though they rarely matched actual international characters), set the SYNERGY_NOLOCALE environment variable. For more information, see SYNERGY_NOLOCALE in the "Environment Variables" chapter of <i>Environment Variables and System Options</i> .
OpenVMS	If the audit server is not running on your machine...	Start the audit server and leave it running.	If the audit server is not running, you'll see the internal C++ runtime error "Create_Exclusive_Lock_1" when you compile. This error occurs because the version 9 compiler uses the C++ runtime. Even though the message looks serious, it does not affect the compilation of your code.
OpenVMS	If your code uses a lowercase logical to define the location of a repository in a .INCLUDE statement...	Change the logical to uppercase in the .INCLUDE statement or add a second define in which the logical is lowercase and quoted (which preserves the case). For example, <pre>\$ define MYRPS "<path>" \$ define "myrps" "<path>"</pre>	Prior to version 9, the compiler automatically uppercased the logical used for the repository location in a .INCLUDE statement. This ensured that it matched the logical definition, since DCL uppercases the command line by default. In version 9, the compiler does not uppercase the logical in the .INCLUDE statement. Consequently, if this logical is lowercase in your code, the compiler generates a "Logical is not defined" error. This change was made in anticipation of future support for ODS-5 case-sensitive filenames.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
OpenVMS	If your code has embedded characters (e.g., CR, LF, ESC, NULL) in a quoted literal... For example, <pre>record crlf ,a2 , "<cr><lf>"</pre>	Use string arithmetic and %CHAR to pass the value that corresponds to the ASCII character. For example, <pre>record crlf ,a2 , %char(13) + & %char(10)</pre>	Embedded non-printing characters are not supported by the version 9 compiler. The results of using them may be unpredictable, such as getting a space instead of the desired character.

Converting to ISAM REV 4 or Patching to REV 5

Converting version 2 or 3 ISAM files to REV 4 or patching them to REV 5 will improve performance. We recommend using **fconvert** to convert the files. Conversion, however, can be a lengthy process. It is much quicker to patch the files with **isutl**. A version 2 or 3 ISAM file that has been patched to version 5 can be used with Synergy/DE version 7.5 and higher. However, you will not be able to make additional parameter changes to a patched file.

Using fconvert to convert to REV 4

Fconvert enables you to convert your existing ISAM files and add on some of the new ISAM characteristics (such as density and index block size).

1. Either unset the environment variable ISAMC_REV or set it to 4. (**Fconvert** creates files that are the current REV level or the REV level defined by ISAMC_REV.)
2. Generate a par file from your existing ISAM file.
3. Add the new options to the par file, as documented in [bldism](#) in the “Synergy DBMS” chapter of *Synergy Language Tools*.
4. Run **fconvert** on the file using the modified par file. For details see [fconvert](#) in the “Synergy DBMS” chapter of *Synergy Language Tools*.

Using isutl to patch to REV 5

Run **isutl** with the **-p** option on the desired files. For details see [isutl](#) in the “Synergy DBMS” chapter of *Synergy Language Tools*.

Upcoming Changes

A future version of Synergy/DE will support Microsoft .NET. As a result, some current features will not be supported in that version. These include deprecated data types, syntax, APIs, compiler options, and alignment warnings. You can see how compatible your code will be with .NET by compiling with the **-qnet** option. It will output warnings to standard error for items that will not be supported.

Workbench

Compatibility Information

- We changed the default installation directory to C:\Program Files\Synergex\SynergyDE. The new directory will be used for new installations. If you are upgrading, the installation will default to your current directory.
- If you customized Workbench version 7, refer to “Before you install Workbench 9” in the Workbench release notes (**REL_WB.TXT**) for instructions on how to retain your customizations when you upgrade.

Required Changes

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
Windows	If you have version 7 Synergy projects...	Convert the projects to version 9.3 by opening them in Workbench.	Workbench version 9.3 uses SlickEdit version 14, and therefore requires that all existing Synergy projects be converted. Most projects are converted automatically when they are opened. (See next item.)
Windows	If you have any of the following project types with the specified configurations that were created in version 7.3.1: Synergy/DE Java;Release Synergy/DE COM;Release Synergy/DE COM;Global	Recreate the projects in version 9.3.	Certain projects from version 7.3.1 cannot be converted. If you open them in Workbench 9.3, they will not be recognized as Synergy projects.
Windows	If you have customized project templates created in version 7.1...	Recreate the templates in version 9.3.	Version 7.1 project templates cannot be converted.
Windows	If you have written your own macros...	Reload them into version 9.3. For each macro, at the Workbench command line enter the following: <code>load macro_name.e</code>	Macros cannot be converted.
Windows	If you use the format %VAR% to refer to an environment variable in a project open command...	Use the format %(VAR) instead.	SlickEdit now requires a different format.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
Windows	If you rely on SlickEdit to create user1 and user2 menu commands for all projects when they are created...	Create the user-defined tool entries manually by entering a new tool name of "user 1" and "user 2" in the Tool name field of the Project Properties dialog.	SlickEdit no longer performs this function automatically.
Windows	If you open Workbench projects in Composer...	Instead, launch Composer from Workbench to modify a specific script file, then return to Workbench.	Composer no longer supports Workbench projects. However, Workbench can pass Composer the name of a script file to open.

UI Toolkit

Compatibility Information

If you include the Toolkit `gblctl.def` or `lstprv.def` file in your programs, you must recompile. See [“Upcoming Changes” on page 75](#) for more information on these files.

Required Changes

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you overload any Synergex-supplied function or subroutine...	Verify that your overloaded routine exactly matches the prototype of the Synergex-supplied routine in return type, and in number, direction, and type of parameters, as well as whether parameters are optional or required.	We now strongly prototype all Synergex-supplied functions and subroutines. Using the -W4 compiler option (/WARNINGS=4 on OpenVMS) will show the prototype the compiler expects and your overloaded routine definition, so that you can see how they differ. Once you change your routine, your code may no longer compile in versions 8.3 and earlier. See “Compatibility Information” on page 46 for a suggestion on how to handle this situation.
All	If you undefine <code>TOOLS_INIT</code> in your code...	Remove the <code>.UNDEFINE</code> .	<code>TOOLS_INIT</code> is now undefined in tools.def . If you also undefine it in your code, the compiler generates an “Identifier not defined” error. Once you remove the <code>.UNDEFINE</code> , your code may not work properly in versions 8.3 and earlier. See “Compatibility Information” on page 46 for a suggestion on how to handle this situation.
All	If you pass the <i>field_id</i> and <i>selection_id</i> arguments to <code>IB_END</code> ...	Remove these arguments.	These arguments have not been supported (and have had no affect on the operation of your code) since Toolkit version 3.3. Their use now causes the compiler to generate a “More argument values than declared parameters in routine” error.
All	If you link against tklib.olb ...	Link against tklib.elb (Windows and UNIX) or tklib_sh.exe (OpenVMS) instead.	We ceased distribution of tklib.olb in version 8.3. We also no longer distribute the files chninf.dbl , chninf.def , and fkey.dbl because they were used only with tklib.olb . This change may break your code.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you use chninf.def to find the highest channel number that Toolkit can use...	Change your code to use the new D_CHANNEL_MAX subfunction to %E_INFO.	We ceased distribution of chninf.def in version 8.3. This change may break your code.
All	If an input field specification in a window script or call to IB_FIELD is too wide for the window...	Revise your input field specification by shortening the field to fit within the window or specifying a view length.	Toolkit generates an error when the display length is too long. Previously, the display was truncated. This change may break your code.
All	If you use IB_FIELD with D_OFF followed by D_FLD_PROMPT or use D_OFF followed by D_FLD_PROMPTLEN...	Be aware that both text and runtime prompts will be canceled.	This makes the behavior of IB_FIELD consistent with windows scripts, where "noprompt" is used to cancel either type of prompt. Previously, D_OFF followed by D_FLD_PROMPT canceled only text prompts, and D_OFF followed by D_FLD_PROMPTLEN canceled only runtime prompts. This change may break your code.
All	If an input field specification in a window script or call to IB_FIELD has more than nine decimal places...	Revise your input field specification to use no more than nine decimal places.	Toolkit supports only nine decimal places in implied-decimal fields. When a number greater than nine is specified, Toolkit generates an error message. Previously, it may have displayed the field incorrectly. This change may break your code.
All	If you use the left or right square bracket character as a field delimiter in the routines I_DISABLE, I_DISPLAY, I_ENABLE, I_INIT, or I_SETDEL...	Use a different delimiter character.	Toolkit now permits you to create fields that reference specific elements in repository arrayed fields. To do this, you must use square brackets; consequently, these routines no longer recognize square brackets as delimiters in field lists. This change may break your code.
All	If you are using the D_FLD_SIZE qualifier for I_FLDMOD to specify the input length for alpha or decimal data types... (7.1)	Use D_FLD_INPLEN instead.	Previously, this qualifier affected the input length for all data types except integer, for which it affected the storage length. Now, D_FLD_SIZE is for the storage length, D_FLD_INPLEN is for the input length, and D_FLD_DSPLEN is for the display length. This change may break your code. For more information, see I_FLDMOD in the "Input Routines" chapter of the <i>UI Toolkit Reference Manual</i> .

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you pass a null string for the <code>I_FORCE new_data</code> argument, and you depend on there being a space in the field...	Change your code to pass a space instead.	Previously, a null string in the <code>new_data</code> argument was interpreted as a space (" "); now, it is interpreted as a null string (""). This change results from a feature that enables you to clear the force buffer and cancel a pending <code>I_FORCE</code> call by passing a null string. This change may break your code.
All	If your Toolkit application includes a customized version of any of the user-replaceable list subroutines (these are the subroutines that begin with <code>USR_L</code>)...	Remove any references to those routines and to the lstprv.def file.	We ceased distribution of the user-replaceable list subroutines in version 8.3. If your application includes a customized version of any of these routines, Toolkit will simply ignore it and call the corresponding default routine (e.g., it will call <code>L_DATA</code> instead of <code>USR_LDATA</code>). We also no longer distribute the files lstusr.dbl , lstprv.dbl , and lstbuf.def . This change may break your code. See also "Upcoming Changes" on page 75 .
All	If your application exceeds the maximum number of entry lists or menu column rows in the <code>MB_</code> routines...	Correct your code.	Previously, the maximums (9 for lists and 20 for rows) were not enforced. Exceeding the maximums could cause unpredictable results, but there was no error. Now a "Too many lists" or "Too many rows" error is returned. This change may break your code.
All	If your code passes a value other than 0 or 1 for the <code>no_place</code> argument to <code>MB_COLUMN</code> ...	Change your code to pass a value of 1.	The name of this argument has changed from <code>no_place</code> to <code>option</code> , and it now is used to determine both column loading and placement options. Previously, any value greater than 0 would cause the column to be loaded without placing it; now, you must pass 1 to get that behavior. This change may break your code.
All	If your code creates blank menu entry names...	Correct your code so that all menu entries are named.	Passing a blank menu entry to <code>MB_ENTRY</code> causes a fatal error.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you use the <code>g_werr</code> record in <code>gblctl.def...</code>	Define your own record to use with the <code>W_RETERR</code> function.	We removed <code>g_werr</code> . This change may break your code. For information on <code>W_RETERR</code> , see "Error Processing in the Windowing System" in the <i>Synergy Windowing API</i> chapter of the <i>Synergy Language Reference Manual</i> . See also "Upcoming Changes" on page 75 for more information about <code>gblctl.def</code> .
All	If your application accesses the <code>gs_dec</code> member of <code>fldinf.def...</code>	Recompile your code.	To support a change in the maximum size of decimal and implied-decimal fields in Toolkit, we changed <code>gs_dec</code> from a d1 to an i1 .
All	If you have code that calls <code>U_SELDFLT...</code>	Update your code to call <code>S_SELDFLT</code> instead.	We removed the <code>U_SELDFLT</code> routine, which was renamed to <code>S_SELDFLT</code> in Toolkit version 3.0. This change may break your code.
All	If your code uses the <code>ctrl_record</code> argument to <code>U_REND...</code> (7.1, 7.3)	Change the variable definition in your code to an a425 .	We increased the size of this variable. Previously, it was an a404 .
Windows	If you specify color palette entry 16 for a list header or footer...	Specify color palette entry 1 to retain the current look.	Previously, color palette entry 16 did not work properly, which caused the default (1) to be used instead. We fixed entry 16, which could cause a change in the appearance of your application.
Windows	If you use the ActiveX list control and have non-static headers or footers...	Adjust the position of the text as necessary.	Text is now centered in the column by default. Previously, it was left-justified.
Windows	In an ActiveX control, if you use <code>%M_SIGNAL</code> in an event method triggered by <code>WM_KEYDOWN</code> for a key that will be translated into a character...	User <code>WM_CHAR</code> to trigger the event method instead.	Using <code>%M_SIGNAL</code> in response to a keydown event causes the keypress to remain in the message queue, which could cause duplicate characters.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
Windows	If you depend on the U_EDITREND, U_SAVELOG, or U_SAVESETTINGS routines or on the D_SAVEFONT subfunction to %U_WNDFONT to write to synergy.ini ...	You must now explicitly specify that you want to write to synergy.ini using the new options for these routines.	Synergy/DE now supports a “system-specific” initialization file (synergy.ini) and a “user-specific” initialization file (synuser.ini). Settings in synuser.ini override settings in synergy.ini . The default behavior for these routines is to write to the synuser.ini file. This change may break your code. See “ Synergy initialization files ” in the “Environment Variables” chapter of <i>Environment Variables and System Options</i> for more information.
Windows	If you use the %U_PRINTQUERY function... (7.1, 7.3)	Use the <i>extprinterid</i> field instead of the <i>printerid</i> field.	The <i>extprinterid</i> field supports 256 character printer names, as required for Windows 2000 and higher.
Windows	If you specify a height for a selection window associated with a set of radio buttons...	Be aware that the appearance of your radio buttons may change. You may need to clear the height setting in order to restore them to their previous appearance.	Previously, the height setting was ignored. Now, it is used to determine the arrangement of radio buttons.
Windows	If you depend on an input method being applied to the subsequent field or button when a menu entry is signaled from a method during I_INPUT processing and the user action moves context to the subsequent field...	Change your code. Or, to revert to the previous behavior, set the RETAIN_CONTEXT_CHANGE_ON_SIGNAL environment variable or the g_retaincontext global variable.	We corrected a problem that caused a method to be applied to the following field when the user moved context to that field. If your application depended on this behavior, this change may break your code.
Windows	If your code unconditionally signals a menu (via M_SIGNAL) in a click event for an ActiveX list... (7.1)	Test whether the D_LSTCLK_ONLINE flag is set.	The D_LSTCLK_ONLINE flag tells you that the user clicked on the line used to resize the columns. If you don’t test for this, the drag resize of the column will be ignored. This change may break your code. For more information, see %UWNDEVENTS_METHOD in the “Utility Routines” chapter of the <i>UI Toolkit Reference Manual</i> .
Windows	If you rely on Toolkit to change the cursor from an hourglass back to a normal cursor... (7.1, 7.3)	Add code to change the cursor.	Setting the cursor to an hourglass will cause it to remain an hourglass until it is reset.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
Windows	If you override cell color and font for an ActiveX list control and then call L_RESTART...	Reapply the overrides after the call to L_RESTART, if you want to retain them.	Cell color and font overrides are now reset to the default after a call to L_RESTART.
Windows	If you rely on the default Synergy runtime color definitions...	To maintain the same color scheme, define the specific colors you want the application to use in your synergy.ini file.	We removed the Synergy color and palette definitions from the distributed synergy.ini file in version 8.3. The new default colors are slightly different than they were previously. Consequently, you may notice a change in the appearance of applications that rely on the default colors.
Windows	If your code uses i1 variables to store the color values returned by U_REND and the WI_PALET subfunction to W_INFO...	Change these variables to at least an i2 or d3 . (We recommend using an i4 .)	To accommodate the increased number of Synergy colors in version 8.3, U_REND and the WI_PALET subfunction to W_INFO may now return values that are too large for i1 variables. This change may break your code.
Windows	If your application uses tabbed dialogs...	You may want to use the g_tabfacecolor global variable in tkctl.def to control which color palette entry will be applied to a window when it is added to a tabbed dialog.	Due to changes we made to the color palette in 8.3, on a Windows system with themes enabled, you may notice a change in the appearance of your application when a window is added to a tabbed dialog.
Windows	If your application has tab sets with the tabs on the bottom or side...	Move the tabs to the top of the tab set.	Tabs are supported only at the top of the tab set on systems that support Windows themes. This change may break your code.
UNIX OpenVMS	If your application has an input window with more than 31 fields, and the 32nd field has an arrive method...	Recompile the window script to take advantage of a fix for this problem.	We corrected a problem that caused the arrive method on the thirty-second field to sometimes fail to be invoked.
UNIX OpenVMS	If you have an application with a check box with field-level renditions specified...	Set the new D_INPPRCRND state for E_STATE to D_OFF if you want to retain the current look of the prompt.	In 9.3, we added the D_INPPRCRND state to E_STATE. By default, this state is set to D_ON on UNIX and OpenVMS, which causes Toolkit to use field-level renditions for check box prompts. Previously Toolkit used renditions for the part of the window that the prompt occupied.

Upcoming Changes

In an upcoming major version of Synergy/DE, we plan to remove the following files from the distribution:

- ▶ **gblctl.def**. This file contains an internal structure that gets updated on each release, thereby requiring that you recompile.
- ▶ **lstprv.def**. This file contains defines for lists. These defines were required for the user-defined list subroutines, which are no longer a part of Toolkit. You should modify your code to remove references to this file.

For more information, see the “Future Revisions” section of the Toolkit release notes (**REL_DTK.TXT**).

Composer

Required Changes

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
Windows	If you use Composer to write colors or fonts to the synergy.ini file...	Be aware that colors and fonts will now be written to the [COLORS] and [FONTS] sections of synuser.ini —not synergy.ini .	Synergy/DE now supports a “system-specific” initialization file (synergy.ini) and a “user-specific” initialization file (synuser.ini). Settings in synuser.ini override settings in synergy.ini . Colors and fonts are treated as user-specific settings. See “Synergy initialization files” in the “Environment Variables and System Options” chapter of <i>Environment Variables and System Options</i> for more information.
Windows	If you have a composer.ini file from version 7 that you want to continue using...	Run the synckini utility to determine the location of synuser.ini . Then, move your composer.ini file to the same directory.	Composer treats composer.ini as a user-specific initialization file and searches for it in the Synergex subdirectory of your local application data directory. (Note that Composer no longer uses SFWINIPATH to locate composer.ini .)
Windows	If you open Workbench projects in Composer...	Instead, launch Composer from Workbench to modify a specific script file, and then return to Workbench.	Composer no longer supports Workbench projects. However, Workbench can pass Composer the name of a script file to open.

Repository

Compatibility Information

- ▶ Repositories created in version 7 do not need to be converted to be used with version 9. Version 9 Synergy/DE tools can open repositories created with versions 7, 8, and 9.
- ▶ SDL (Synergy Data Language) files do not need to be converted.
- ▶ If you have customized Repository's window library, you must reapply your customizations and rescript it.

Required Changes

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you have your own copies of any of the record definitions contained within ddinfo.def ...	Check ddinfo.def for details of changes made and make the corresponding changes to your own copies.	We have modified the ddinfo.def file used by the Repository Subroutine Library.
All	If you use rpsutl.dbr and test the status of .new files to determine if the load succeeded... (7.1)	Test the exit status from rpsutl.dbr instead.	If an error occurs during the load process, rpsutl.dbr will now delete any repository files it created. This change may break your code.
All	If you use templates... (7.1, 7.3)	Generate your repository to a schema and then reload it.	This is required to ensure that all current template attribute values have been properly propagated to their descendants.

Upcoming Changes

In an upcoming version of Synergy/DE, we are planning to remove the Repository object library (OLB) file **ddlib.olb**. It has been replaced with the executable library (ELB) file **ddlib.elb**. (The OLB file will be available on the Synergex web site.)

You should be linking against the ELB file instead of the OLB file for the following reasons:

- ▶ An ELB will reduce the size of your application if you link multiple programs against the library.
- ▶ Linking against an ELB is more reliable than rebuilding the OLB and making a custom ELB.
- ▶ An ELB provides more compatibility when multiple Synergy applications from different companies coexist on one system.
- ▶ In most cases, you can upgrade to newer versions of Synergy/DE without recompiling and relinking your applications.

ReportWriter

Compatibility Information

No conversion or rescripting is required.

Required Changes

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
OpenVMS	If you are migrating from a version between 7.1.1 and 7.1.3b...	Relink any application that is linked against SYNRPT.EXE .	In versions 7.1.1 to 7.1.3b, the SYNRPT.EXE shared image did not link against a vector table. This change may break your code.

xfServer

Compatibility Information

- ▶ xfServer 9 supports clients running Synergy/DE version 6.1 and higher.
- ▶ (Windows) We changed the default installation directory to C:\Program Files\Synergex\SynergyDE. The new directory will be used for new installations. If you are upgrading, the installation will default to your current directory.
- ▶ (Windows) If you have previously had xfServer, and you install version 9 to a different directory, you must remove and re-add your xfServer services.

Required Changes

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you access xfServer data on the xfServerPlus port...	Reconfigure one of these services to run on a different port.	xfServer data access on the xfServerPlus port is no longer supported. This change may break your code.
All	If you reference one remote file specification from another when the first file specification is on Windows...	Cease to do so.	This is no longer permitted because of the potential for data corruption.
Windows	If you create files without specifying a path...	Set the environment variable RSFILPATH to specify a location for files without a path.	If RSFILPATH is not set, the files are put in the directory that contains documents common to all users on the specific operating system. For example, on Windows XP, they are put in C:\Documents and Settings\ All Users\Shared Documents. See RSFILPATH in the “Environment Variables” chapter of <i>Environment Variables and System Options</i> .
Windows	If the RUSER environment variable is set in both the registry and the environment...	Be aware that RUSER set in the environment now has precedence over RUSER set in the registry.	Note that when RUSER is set in the Synrc node in the registry, as is required to configure xfServerPlus for remote data access, it is read into the environment when rsynd starts up, and it is therefore considered to be set in the environment. This change may break your code.
Windows	If RUSER is set to the user name “SSPI” and no password is specified...	Change the user name.	To support Windows authentication security, RUSER recognizes “SSPI” as a special value.

xfServerPlus

Compatibility Information

- ▶ xfServerPlus version 9.3 can receive routine calls from version 7.1 and higher xfNetLink clients.
- ▶ Synergy Method Catalog (SMC) files must be converted for use with xfServerPlus and Workbench 9.3 and with version 9.3 utilities such as the Method Definition Utility (MDU) and **smc_elb.exe**. See the “SMC conversion utility” section of the xfServerPlus release notes (**REL_XFPL.TXT**) for instructions.
- ▶ (Windows) We changed the default installation directory to C:\Program Files\Synergex\SynergyDE. The new directory will be used for new installations. If you are upgrading, the installation will default to your current directory.
- ▶ (Windows) If you have previously had xfServerPlus, and you install version 9 to a different directory, you must remove and re-add your xfServerPlus services.

Required Changes

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you use the “Add new interfaces” option in the Import function of the Method Definition Utility...(7.5)	Use the “Update interfaces” option instead and then choose the new interfaces you want to add.	The MDU now has an Update interfaces option, which updates existing interfaces, in addition to adding new interfaces.
All	If you access xfServer data on the xfServerPlus port...	Reconfigure one of these services to run on a different port.	xfServer data access on the xfServerPlus port is no longer supported. This change may break your code.
All	If you use the default port 2330... (7.1)	You must explicitly specify 2330 as the port, or use the new default, 2356.	We changed the default port from 2330 to 2356 so that it would not conflict with the default port for xfServer.
All	If you use the Reports feature in the Method Definition Utility...	Run the Export Methods option to generate an XML file, and then view the file in an XML viewer.	We removed the Reports feature.
Windows	If you set SFWINIPATH on your xfServerPlus machine... (7.1, 7.3)	Be aware that db s will now read the synergy.ini file.	The service runtime (db s.exe) will use the synergy.ini file when SFWINIPATH is set. If SFWINIPATH is not set, db s ignores synergy.ini . This change may break your code.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
Windows	If the RUSER environment variable is set in both the registry and the environment...	Be aware that RUSER set in the environment now has precedence over RUSER set in the registry.	When RUSER is set in the Synrc node in the registry, as is required to configure xfServerPlus for remote data access, it is read into the environment when rsynd starts up, and it is therefore considered to be set in the environment. This change may break your code.
Windows UNIX	If you use the xfServerPlus log file...	Expect to see multiple log files.	By default, xfServerPlus now generates a separate log file for each session. You can set the XFPL_SINGLELOGFILE option to ON in the xfpl.ini file to continue producing a single file.
UNIX	If you start xfServerPlus with a password...	Encode the -u password with setruser .	To improve xfServerPlus security, a clear text password is no longer permitted when starting xfServerPlus. This change may break your code if you start xfServerPlus with a script.
UNIX	If you start xfServerPlus without a password...	The user starting xfServerPlus must be root or signed on as the user specified with -u .	This is the documented behavior, but it was not previously enforced. This change may break your code if you start xfServerPlus with a script.
OpenVMS	If you run xfServerPlus on OpenVMS...	The xfServerPlus account must have the SHARE privilege.	Without this privilege, xfServerPlus will not run. This change may break your code.

xfNetLink Java Edition

Compatibility Information

- ▶ (Windows) We changed the default xfNetLink Java installation directory from C:\xfNetLnk to C:\Program Files\Synergex\xfNLJava. The new directory will be used for new installations. If you are upgrading, the installation will default to your current directory.
- ▶ (UNIX, OpenVMS) We changed the name of the default xfNetLink Java installation directory from xfNetLnk to xfNLJava.

Required Changes

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you do not include the Xerces parser files (xercesImpl.jar and xml-apis.jar) and the directory where xfNetLink Java was installed in your classpath...	Update your classpath to include them.	Additional functionality added in versions 7.5 and 8.1.7 requires that you include the Xerces parser files and the installation directory in the classpath. See "Setting the Classpath" in the "Creating Java Class Wrappers" chapter of the <i>Developing Distributed Synergy Applications</i> manual.

Notes:

Starting with version 7.5, xfNetLink Java includes utilities that enable you to create Java class wrappers from your SMC and Repository definitions, and then use the class wrappers to build a Java JAR file. Rather than use the xfNetLink Java classes directly, we recommend that you use the class wrappers feature to take advantage of all the latest xfNetLink Java enhancements.

Upcoming Changes

The port number range feature (minport and maxport) will be deprecated in a future version. We recommend that you quit using it.

xfNetLink COM Edition

Compatibility Information

- ▶ Version 7.1 client code will not run in version 9.
- ▶ We changed the default xfNetLink COM installation directory from C:\xfNLCOM to C:\Program Files\Synergex\xfNLCOM. The new directory will be used for new installations. If you are upgrading, the installation will default to your current directory.
- ▶ If you have previously had xfNetLink COM, and you install version 9 to a different directory, you must unregister and reregister your type libraries.

Required Changes

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
Windows	If you're using the xfNetLink COM 7.1 API...	Use the component generation tools included with xfNetLink COM to generate a Synergy type library.	We removed xfnlcom.dll from the distribution. (xfnlcom.dll was the original version 7.1 implementation for xfNetLink COM.) Code written in version 7.1 will not run in 9.3. This change may break your code.
Windows	If you use multiple interfaces with the same name... (7.3)	Be aware that a number (starting with 1 and incrementing) is added to the end of each duplicate interface.	Previously, we added underscores. This change may break your code.
Windows	If you register Synergy type libraries... (7.3)	Unregister and reregister them.	Starting with version 7.5, xfrrouter.dll , which is the implementation for Synergy TLBs, is installed in the xfNLCOM directory. (Previously, it was installed in SynergyDE\xfutil.) This change may break your code. See the “Creating Synergy COM Components” chapter of the <i>Developing Distributed Synergy Applications</i> manual for instructions on registering and unregistering TLBs.
Windows	If you would like to pool xfNetLink COM components... (7.3)	Add an entry for the XFPL_REGCLEANUP routine to your SMC.	Support for COM pooling was added in version 7.5. (New installations of xfNetLink COM will include this routine in the SMC.)

Upcoming Changes

The port number range feature (minport and maxport) will be deprecated in a future version. We recommend that you quit using it.

xfNetLink .NET Edition

Compatibility Information

Version 7.5.3 assemblies must be rebuilt to be used with version 9. See the table below.

Required Changes

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
Windows	If your application passes structures that contain fields defined as binary data type in the repository...	To continue to have these fields converted to strings in the generated C# classes, run gencs with the -nb option; else, they will be converted to byte arrays.	A field in a structure that is defined as a binary data type in the repository is now converted to a byte array in the generated C# code instead of a string. This change may break your code if you regenerate classes for an existing application.
Windows	If you use the port number range feature (minport and maxport)...	Cease to do so.	This feature has been deprecated. Minport and maxport are ignored when set in the configuration file or passed with the connect() method.
Windows	If you have created an application using a Synergy assembly built with version 7.5.3...	Rerun the batch file (created when you ran gencs) to rebuild the assembly. Then, rebuild your application to use the rebuilt assembly.	Rebuilding is required because of the way Microsoft .NET handles assembly versioning. This change may break your code. See the “Creating Synergy .NET Assemblies” chapter of the <i>Developing Distributed Synergy Applications</i> manual for more information on running the batch file.
Windows	If you use version 1.0 or 1.1 of the .NET Framework...	Upgrade to version 2.0.	.NET Framework 2.0 is now the minimum supported version.
Windows	If you use version 1.0 or 1.1 of the .NET Framework...	Before regenerating classes for an existing assembly, edit the AssemblyInfo.cs file to remove the attributes AssemblyDelaySign and AssemblyKeyFile. Or, you can delete the AssemblyInfo file; xfNetLink will create a new one the next time you run gencs .	In .NET Framework versions 1.0 and 1.1, these two attributes were included in the AssemblyInfo file. In version 2, they are specified on the compiler command line. Because the AssemblyInfo file is no longer overwritten in version 9.1.3, you must either delete these attributes from the file or delete the file so that it is recreated.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
Windows	If you have fields in Repository that are flagged as read-only, and you generate structure members as properties, and you need to be able to set those properties in your client code...	Clear the read-only fields in Repository before you generate classes.	Starting with version 9.1.3, fields flagged as read-only in Repository are generated as read-only in the C# classes when structure members are generated as properties. These properties will have a "get" method, but no "set" method. This change may break your code if you regenerate classes for an existing application. Note that the read-only flag is not honored when you generate structure members as public fields.
Windows	If you use client-side logging...	Expect to see multiple log files.	By default, xfNetLink .NET now generates a separate log file for each class that instantiates a connection to xfServerPlus. You can set "single log file" to "on" in the application configuration file to continue producing a single file.
Windows	If your Synergy assembly includes methods that pass structures as parameters...	Add a reference to xfnlnet.dll in your Visual Studio .NET project.	This requirement results from two new features added in 8.3, the gencs -i option and the generation of a C# interface for every generated class.

SQL OpenNet

Compatibility Information

- ▶ All Connectivity Series products for client and server must be upgraded to version 9.3. Otherwise, a version mismatch error occurs.
- ▶ (Windows) If you have previously had Connectivity Series, and you install version 9.3 to a different directory, you must remove and re-add your SQL OpenNet service.

Required Changes

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
Windows	If you rely on the installation program to register and start SQL OpenNet... (7.1, 7.3)	Use the Synergy Configuration Program instead to register, start, and stop SQL OpenNet.	The SynSQL service is no longer registered and started by the installation program.
Windows	If you use vtxnet2 ...	Change the start-up command line in the opennet.srv file to specify vtxnet2 .	We changed the default SQL OpenNet server from vtxnet2 to vtxnetd . This change may break your code.

xfODBC

Compatibility Information

- ▶ All Connectivity Series products for client and server must be upgraded to version 9.3. Otherwise, a version mismatch error occurs.
- ▶ You must re-create your system catalogs for use with version 9.3 by running the **dbcreate** utility with the **-c** option (or by using the “Clear and re-create” option in the DBA program).

Required Changes

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you set SODBC_NONULL... (7.3.3, 7.5)	Review “ Preventing null updates and interpreting spaces, zeros, and null values ” in the “Preliminary Steps” chapter of the <i>xfODBC User’s Guide</i> . Then set “Null allowed” values in Repository and the SODBC_NONULL environment variable as necessary.	The dbcreate utility now uses the new Repository “Null allowed” field to determine how to set the “Null allowed” property for system catalog columns. When generating a system catalog, the SODBC_NONULL environment variable setting is used only for repository fields with “Null allowed” set to Default. We changed the default for SODBC_NONULL to 4, and we also changed the behavior for some SODBC_NONULL settings.
All	If you do not set SODBC_NONULL... (7.3.3, 7.5)	Set SODBC_NONULL to 0 to maintain your current “Null allowed” settings for system catalog columns.	
All	If you use SQL_REAL for the CAST and CONVERT functions...	Use SQL_FLOAT instead.	xfODBC no longer supports SQL_REAL.
All	If you use xfODBC Driver DSNs... (7.1)	Create new DSNs for use with for xfODBC.	xfODBC cannot use DSNs created for xfODBC Driver. This change may break your code. For details on creating new DSNs, see “ Setting Up Access with DSNs ” in the “Configuring Data Access” chapter of the <i>xfODBC User’s Guide</i> .
All	If you use the Appended to Connect String field in the xfODBC Setup window for the network connect string... (7.1)	Use the other fields in the xfODBC Setup window to specify that information instead.	xfODBC creates the network string from the information in the xfODBC Setup window. See the “ Configuring Data Access ” chapter of the <i>xfODBC User’s Guide</i> for more information.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you use the SODBC_NOROLL or VORTEX_ODBC_DATETIME environment variables... (7.1)	Cease to do so.	These xfODBC Driver environment variables are not used by xfODBC.
All	If you use multidimensional arrays... (7.1, 7.3)	Rewrite your queries to use the current array name format.	We changed the way multidimensional arrays are named. This change may break your code.
All	If you use the Repository Converter (sodbccnv)... (7.1)	Use the dbcreate utility instead.	The sodbccnv utility has been replaced with dbcreate .
All	If you rely on the default display formats for date/time, dates, or times... (7.1)	Modify your code to use the new defaults or use the TO_CHAR scalar function to change the default format. (See SSQL.TXT for details.)	The default formats have changed. The new defaults are as follows: YYYY-MM-DD HH:MI:SS (date/time) YYYY-MM-DD (dates) HH:MI:SS (times)
All	If you want the column names in the system catalog to consist solely of the field names... (7.1, 7.3)	Set the environment variable SODBC_NOGROUPNAME.	By default, if the field is part of a group in the repository, the column name created in the system catalog will include the group name as a prefix to the field name.
All	If your data includes unsigned fields that contain negative values...	Change the fields to signed fields by selecting the "Negative allowed" option in Repository. Or, you can set the SODBC_NOUNSIGNED environment variable to revert to the previous behavior.	The dbcreate utility and xfODBC driver now distinguish between signed and unsigned fields. This prevents negative values from being stored in non-negative fields, which enables xfODBC to work better with ADO.NET and Microsoft Access. Refer to the "xfODBC" section of the Connectivity Series release notes (REL_CONN.TXT) for more detailed information on this topic.
Windows	If you use the ODBC API to call SQLDriverConnect to open a connection from a heavily threaded environment...	Use the SQL_DRIVER_NOPROMPT option.	Using SQL_DRIVER_NOPROMPT prevents a login box from appearing, which may cause vtx4.dll to hang.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
UNIX	If you use the CONNECT environment variable...	Use CONNECTDIR instead.	We replaced CONNECT and CONNECTEXE with CONNECTDIR so that all platforms use the same setting. This change may break your code.
OpenVMS	If you use the CONNECTEXE logical...		
OpenVMS	If you have created your own XFODBCUSR.EXE file... (7.3, 7.5)	Rename the file XFODBCUSR_SO.EXE and set the logical XFODBCUSR_SO (instead of XFODBCUSR) to point to it.	We changed the name of the XFODBCUSR.EXE file (and the corresponding logical) to XFODBCUSR_SO.EXE . This change may break your code.

Notes:

Refer to the [xfODBC User's Guide](#) for a tutorial to help you learn how to set up and use xfODBC.

SQL Connection

Compatibility Information

- All Connectivity Series products for client and server must be upgraded to version 9.3. Otherwise, a version mismatch error occurs.
- If you use SQL Connection with SQL Server, we recommend that you install Service Pack 4 for SQL Server 2000. This service pack contains several Synergy/DE-specific fixes.

Required Changes

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you use the %SSC_BLOB routine...	Use the new routine %SSC_LARGECOL instead.	The new routine and option improve SQL Connection's ability to get and put large binary column data and large character column data. See %SSC_LARGECOL in the "Database Functions" chapter of the <i>SQL Connection Reference Manual</i> for details.
All	If you use the SSQL_BLOBCOL option for %SSC_OPEN...	Instead, use the new SSQL_LARGECOL option for %SSC_OPEN for gets and the SSQL_LARGECOL option for %SSC_EXECUTE for puts.	
All	If you don't want to use encryption...	Remove the -k option from the vtxnetd or vtxnet2 command line and remove the key_connect setting from the net.ini file.	By default, the database user name and password and the host user name and password are encrypted. For more information, see "The vtxnetd and vtxnet2 Programs" in the "Configuring Connectivity Series" chapter of the <i>Installation Configuration Guide</i> and "Setting connect string defaults and encryption in net.ini" in the "Preparing Your Environment" chapter of the <i>SQL Connection Reference Manual</i> .
All	If your program .INCLUDEs the ssql.def file...	Update your program to .INCLUDE ssql.def from the connect directory using CONNECTDIR.	We moved ssql.def from the synsqlx directory to the connect directory.

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
All	If you use numeric fields for dates... (7.1, 7.3)	Change your code to use ^A to access the fields as alpha fields. Or, in your database, overlay the fields with alpha fields, and then access data with the overlay fields. Or, change your code to handle Julian day values.	When numeric date fields are bound or defined, they are treated as Julian days, which are calculated using a default time base of -1721378. (Formerly, they were treated as alpha fields with YYYYMMDD formats.) This change may break your code. Note that you can use SSQL_OLD_ZONEDATE, an %SSC_CMD option, and SSQL_OUTDATE, an %SSC_EXECIO option, to disable the new date behavior for <i>retrieved</i> dates. These options do not, however, affect dates written to the database. In addition, using SSQL_OLD_ZONEDATE can cause increased CPU time and memory use.
All	If you rely on the default format for dates returned into defined alpha variables... (7.1)	Change your code to handle the new default format or use SET OPTION DATETIME to change the default format.	The format for dates returned into defined alpha variables has changed from DD-MON-RR to DD-MON-YYYY. This change may break your code. See the SSQL.TXT file for information on SET OPTION DATETIME.
All	If you have created a net.ini file...	Add the settings from your net.ini file to the distributed net.ini file.	We now distribute a net.ini file in the connect\synodbc\lib directory. If you have a net.ini file in that same directory, it will be overwritten when version 9 is installed.
Windows	If you select the database driver during installation... (7.1, 7.3)	Use the Synergy Configuration Program to select the active database.	During installation, a default driver is selected for Oracle and SQL Server. You can change the default from the Synergy Configuration Program.
Windows	If you use VTX12 or the sqlserver keyword in your connect string...	Update your program to use VTX12_ODBC or change the default driver in the Synergy Configuration Program to continue using a DB-Library-based driver.	We changed the default database driver for SQL Server from a DB-Library-based driver to an ODBC-based driver (VTX12_ODBC). This is the driver that is used when a connect string specifies VTX12 or sqlserver. This change may break your code. For more information, see "Using your program with different drivers and databases" in the "Creating SQL Connection Programs" chapter of the <i>SQL Connection Reference Manual</i> .

Platform	Version 7 usage	Change required for 9.3	Explanation/Comments
Windows	If you are running Connectivity Series in a shared installation and want to take advantage of encryption without reinstalling on all your client machines... (7.5)	Manually set the VORTEX_HOME environment variable to the same location as GENESIS_HOME on each Synergy/DE Client machine.	Although it is possible to set VORTEX_HOME by updating the Synergy/DE Client installations, we recommend that you simply set the environment variable manually.
UNIX	If you use connect strings that include a UNIX SQL OpenNet Server... (7.1)	Alter the connect string to include the driver's program name.	See the "Configuring Connectivity Series" chapter in the <i>Installation Configuration Guide</i> and the SQL Connection Reference Manual for details on connect strings.
UNIX	If you execute sqlunixbld and makesqlsrv to add client support... (7.1)	Cease to do so.	Client support is now pre-built.
UNIX	If you use the CONNECT environment variable...	Use CONNECTDIR instead.	We replaced CONNECT and CONNEXTE with CONNECTDIR so that all platforms use the same setting. This change may break your code.
OpenVMS	If you use the CONNEXTE logical...		
OpenVMS	If you use connect strings... (7.1, 7.3)	Change the <i>driver_name</i> argument to a valid driver.	The <i>driver_name</i> argument for connect strings is no longer just a placeholder. Now, it must be set to the name of the database driver. For a list of database driver names, see "Connect string syntax" in the <i>"Creating SQL Connection Programs"</i> chapter of the <i>SQL Connection Reference Manual</i> .
OpenVMS	If you use Oracle or Oracle Rdb natively...	Rebuild your database driver using BUILD_SSQL_DB.COM .	We have moved to a shared image model for database drivers on OpenVMS. This change may break your code. For instructions on building a database driver, see "Configuring and Building SQL OpenNet for SQL Connection" in the VMS section of the <i>"Configuring Connectivity Series"</i> chapter of the <i>Installation Configuration Guide</i> .

Migrating from Version 8 to Version 9.3

This section is designed to help you migrate from version 8.1 or 8.3 to version 9.3.

Some items in the Version 8 usage columns in the Required Changes tables include a version number. This indicates the version to which that item pertains. That is, if the version number is 8.1, users migrating from 8.1 to 9.3 will need to make the required change. However, users migrating from version 8.3 to 9.3 will not need to make the change. If no version is specified, the item applies to all version 8 users.

License Manager

Compatibility Information

You must reconfigure all products for Synergy/DE version 9.

Required Changes

No changes are required.

Synergy Language

Compatibility Information

- ▶ Version 9 will run code compiled and linked with Synergy/DE 6.1 and higher; you do not *have* to recompile.
- ▶ You may *want* to recompile, however, to take advantage of new optimization techniques and some fixes in version 9 (as noted in the table below). You may choose to recompile some portions of your application now and other portions later. See also the compatibility information in “UI Toolkit” on [page 111](#).
- ▶ If you recompile, you will need to make changes to your code due to the stricter error checking in the version 9 compiler and the fact that we have strongly prototyped all Synergex-supplied subroutines and functions. In most cases, the version 9 compiler will find errors that earlier versions of the compiler did not. You can turn off most of the stricter error checking by compiling with the **-qrelaxed (/RELAXED** on OpenVMS) switch. For details on this switch and its options, see “[Compiler options](#)” in the “Building and Running Synergy Language Programs” chapter of *Synergy Language Tools*.
- ▶ To debug with version 9, you must recompile, because the version 9 debugger cannot show source line information or provide symbolic access to variables unless the program is recompiled. If you need to debug your code in version 9, but are not yet ready to take advantage of version 9’s stricter error-checking and strong prototyping, recompile with the compiler distributed as **dbl8**. Note that code compiled with **dbl8** can be run only on version 9.
- ▶ The object files, object libraries, and executables generated with version 9 are not backward compatible. However, changes you make to your code to comply with the enhanced error checking and strong prototyping will compile under previous versions of Synergy, unless stated otherwise in this document. In those few instances, you can use the IFDEF compiler directive so that the code can be compiled with either version 9 or a previous version. For example, let’s say you want to overload the subroutine PUTCM, which was defined like this prior to version 9:

```
.subroutine putcm  
data, a
```

Due to strong prototyping, the parameter *data* is defined thus in version 9:

```
in req data, a
```

Since the IN and REQ parameter modifiers are new in version 9, code that includes them will not compile in a prior version. You can handle this situation thus:

```
.ifdef DBLV9  
  subroutine putcm  
  in req data, a  
.else  
  subroutine putcm  
  data, a  
.endc
```

- ▶ On OpenVMS, due to increased memory requirements, you may need to change the value of some authorize parameters for developers who are using the version 9 compiler. See “[OpenVMS requirements](#)” in the “Requirements and Considerations” chapter of the *Installation Configuration Guide* for details and recommended values.

Required Changes

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If you pass a variable of the wrong data type, such as an alpha variable to a routine that expects a decimal or integer variable...	Correct your code to use the correct data type. Optionally, you may be able to cast (^ A , ^ D , or ^ I) the variable to the correct type.	Passing the incorrect data type causes the compiler to generate a "Type mismatch" error. The version 9 compiler strongly prototypes Synergex-supplied functions and subroutines, which means that your call must exactly match the Synergex-supplied routine in return type, and in number, direction, and type of parameters, as well as whether parameters are optional or required. You can reduce the occurrence of this error by using the -qrelaxed:param compiler option. In addition, the version 9 compiler strongly prototypes all user-defined routines in a compilation unit to ensure that calls within that unit match the routine declaration. For these routines, you can use the -qrelaxed:local compiler option to reduce the occurrence of this error.
All	If you pass more arguments to a routine than the routine defined...	Correct your code to pass the correct number of arguments. For user-defined routines, you can optionally do one of the following to correct your code: ■ Change the routine declaration to include the extra arguments. ■ Change the routine declaration to include the VARARGS modifier.	This situation now causes the compiler to generate a "More argument values than declared parameters in routine" error. See the item above for an explanation of strong prototyping in Synergex-supplied and user-defined routines and for the compiler options that will reduce the occurrence of this error. Note that if you use VARARGS, your code will no longer compile in versions 8.3 and earlier. For more information on VARARGS, see the Synergy Language Reference Manual.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If you fail to pass required arguments to a routine...	Correct your code to pass the missing arguments.	Previously, this error would not be caught until runtime. Now, because the version 9 compiler strongly prototypes all Synergex-supplied routines, it causes the compiler to generate a "Missing required parameter <i>name</i> in routine <i>name</i> " error.
All	If your code includes external functions in which the function return type in the function declaration does not match the return type of the function itself...	Correct your code so that they match. In version 9 you can explicitly specify the type of the function return value. If it's not explicitly typed, the return type is determined by the data type of the first FRETURN statement. See FUNCTION-ENDFUNCTION in the "Synergy Language Statements" chapter of the <i>Synergy Language Reference Manual</i> for details.	Because the version 9 compiler strongly prototypes all routines in a compilation unit, this situation now causes the compiler to generate an "External function <i>name</i> does not match prototype" error. You can reduce the occurrence of this error by using the -qrelaxed:extf compiler option.
All	If you overload any Synergex-supplied function or subroutine...	Verify that your overloaded routine exactly matches the prototype of the Synergex-supplied routine in return type, and in number, direction, and type of parameters, as well as whether parameters are optional or required.	We now strongly prototype Synergex-supplied functions and subroutines. Using the -W4 compiler option (/WARNINGS=4 on OpenVMS) will show the prototype the compiler expects and your overloaded routine definition, so that you can see how they differ. Once you change your routine, your code may no longer compile in versions 8.3 and earlier. See "Compatibility Information" on page 94 for a suggestion on how to handle this situation.
All	If you use %ISINFO to return alpha data...	Use the new %ISINFOA routine instead or use ^A to cast the return type.	%ISINFO now returns only numeric data because strong prototyping requires that there be only one return type. If you use the new %ISINFOA routine, your code will no longer compile in versions 8.3 and earlier. For backward compiler compatibility, use ^A.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If you use the %SYNSOCK routine...	Change your code to use the %SS_ functions instead.	Previously, the Synergy socket API routines were implemented as parameterized macros (%SS_), which invoked the %SYNSOCK routine. In version 9, the compiler treats the %SS_ routines the same as functions in order to support strong prototyping. Therefore, %SYNSOCK is no longer needed and has been removed from the runtime. See the “Synergy Socket API” chapter of the <i>Synergy Language Reference Manual</i> . This change may break your code.
All	If you use END (as opposed to .END) and expect all .DEFINES to be removed and unavailable to the next module in the source file...	Add .UNDEFINE compiler directives as necessary. For example, a .DEFINE might be used to control the initializing of a global data section: <pre>.ifdef GBL_INIT .undefine GBL_INIT global data section A ,init .else global data section A .endc</pre>	The compiler now treats END and .END the same. This means that .DEFINES are not undefined when END is encountered and are therefore available to the next module in the source file. This could cause a “Symbol already defined” error or lead to unexpected behavior because something is defined when it should not be. This change may break your code. To retain the previous behavior, in which END and .END were treated differently, use the -qrelaxed:end compiler option.
All	If you define a field that implicitly extends a record by overlapping another field... For example, <pre>record pbuff ,a80 page_hdr ,a5@pbuff page_text ,a85@pbuff+5</pre>	Correct your code. In this example, you could change the size of pbuff to 90.	This is often the result of a coding error. It is no longer supported, and the compiler now generates a “Cannot extend record by <i>nn</i> bytes with overlay field <i>name</i> ” error. Note that a field is permitted to extend a record as long as it does not overlap another field.
All	If you have records that overlay commons or records, or if you have commons that overlay other commons...	Be aware that record and common overlays are now strictly enforced as to the type being overlaid. You may need to correct your code.	This situation causes the compiler to generate an “Overlay not allowed” error. See “Overlaying data” in the “Defining Data” chapter of the <i>Synergy Language Reference Manual</i> for a list of rules that apply to overlaying records.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If your code <code>.INCLUDEs</code> named records that have overlay fields, and you get an “Ambiguous symbol” error when compiling...	Correct your code to remove the ambiguity. The easiest way to do this is to use the <code>NORECORD</code> qualifier. For example, instead of this: <pre>.include "rec" repository, record=myRecord</pre> You can do this: <pre>record myRecord .include "rec" repository, norecord</pre> The other option is to remove the ambiguity by renaming the duplicate symbol.	In version 9, we changed the way in which <code>.INCLUDE</code> offsets are specified by the compiler. If the record is <i>named</i>, dot notation using the record, group, and field names is used. If the record is <i>unnamed</i>, a numeric offset relative to the overlay field is used. The use of the named reference could cause the compiler to generate an “Ambiguous symbol” error if you have a duplicate symbol elsewhere in your code. Using the <code>NORECORD</code> qualifier will ensure this does not happen. For more information, see “Including from a Repository” in the Discussion for <code>.INCLUDE</code> in the “Preprocessor and Compiler Directives” chapter of the <i>Synergy Language Reference Manual</i>.
All	If your code includes a data division in which the first item is a structure <i>and</i> there is no <code>MAIN</code> statement...	Add a <code>MAIN</code> statement.	<code>MAIN</code> is optional <i>unless</i> the first item in the data division is a <code>STRUCTURE</code> statement without a corresponding <code>ENDSTRUCTURE</code>. The error generated by the compiler will vary.
All	If your code contains references to ambiguous field definitions within the data division... For example, <pre>record red ,a3 record joe blue ,a ^size(red) red ,d2</pre> A similar problem occurs when you have a structure defined in a repository, and you reference that structure as an implicit group in another repository structure and name the group the same name as the referenced structure, and then include both structures in a program.	Correct your code to remove the ambiguity. In the first example, the correction required would depend on what the code was attempting to do. For the repository example, you need to either rename the first structure in the <code>.INCLUDE</code> directive to something unique, or change the name of the group within the second structure in the repository.	In version 9, ambiguous field references can be detected within the data division and cause the compiler to generate an “Ambiguous symbol” error. Previously, they could be detected only when referenced within the procedure division. To allow the old usage, use the -qrelaxed:path compiler option.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If your code uses i8 data types as function return values on 32-bit systems...	Change your code to use a supported data type.	Using an i8 (or long) data type on a 32-bit platform now causes the compiler to generate a "Type i8 cannot be return value on 32-bit platform" error. Previously, there was no error, and if you passed a value larger than an i4 to an i8 , it was truncated. This change may break your code.
All	If you use anything other than an a6 for a GETRFA or RFA qualifier variable...	Change your code to use an a6 or set system option #56.	Prior to 9.3, a GETRFA or RFA qualifier variable that exceeded 6 bytes was allowed. Now, with the introduction of GRFAs, if the size of a GETRFA or RFA argument is anything other than 6 or 10 bytes, an "Invalid record's file address" error is generated. Setting system option #56 turns off this behavior and causes all RFAs that exceed 6 bytes to be treated as 6-byte RFAs. Even if you are using an a10 , you will need to set system option #56, because a10 now indicates a GRFA and will cause a "Record not same" or "Feature not supported in this version" error when used for an RFA. This change may break your code.
All	If you use the LOCAL, STATIC, or STACK qualifiers with the MAIN statement...	Remove them from your code.	MAIN no longer allows these qualifiers. Previously, although allowed, they were ignored and did not affect how your code operated. Now, the compiler generates an "Invalid MAIN declaration" error.
All	If your code uses FIRST or LAST with the I/O statements FIND, READ, or WRITE...	Use ^FIRST or ^LAST instead.	The use of these qualifiers without the caret is no longer supported and causes the compiler to generate an "xxx not found" error.
All	If your code uses short forms of identifier names, such as NOPREC for NOPRECISION...	Use the full documented identifier names.	We no longer support shortened forms of identifier names. Using them causes a compiler error.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If your code includes the START compiler directive without a period at the beginning...	Add the period.	All compiler directives now require a period at the beginning. Without the period, the START directive will be ignored.
All	If your code uses non-numeric variables as subscripts or in ranging operations... For example, <code>clear myvar(aavar)</code> Or <code>clear myvar(offset:aavar)</code>	Correct your code to use whole numbers instead.	Non-numeric variables are not allowed in subscripts or ranging operations and cause the compiler to generate a "Numeric expression required" error.
All	If your code uses implied decimal variables for channel numbers, subscripts, or in ranging operations...	Correct your code to use whole numbers instead.	Previously, implied decimal values were rounded to the nearest whole number. Now, they cause the compiler to generate a "Decimal or integer expression required" error.
All	If your code includes decimal fields in which the whole number portion exceeds 18...	Correct your code so that the whole number portion is 18 or less.	Although this was previously allowed, only assignment statements were guaranteed to work, and it was treated as an 18-digit field with leading zeros in the inaccessible part of the field. This now causes the compiler to generate an "invalid data size specification" warning. To allow the old usage, use the -qrelaxed compiler option.
All	If your code includes implied decimal fields, in which the difference between the whole number part of the field and the precision part is greater than 18... (8.1)	Correct your code such that the difference is 18 or less.	The compiler now generates a BIGNUM error in this condition.
All	If your code includes IF statements that are missing the body statement... For example, <code>if (condition) then nop else end</code>	Correct your code to include a body statement.	This syntax is no longer permitted and causes the compiler to generate a "Statement expected" error.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If your code includes a statement on the same line as a BEGIN statement... For example, <pre>begin some_command end</pre> Or <pre>begin & some_command end</pre>	Move it to the next line. Or, in the case of the second example, just remove the continuation character.	This coding style is no longer permitted and causes the compiler to generate a "BEGIN should be followed by end of line" error.
All	If your code has function declarations with multiple qualifiers... For example, <pre>.function myFunc, ^val reentrant</pre>	Verify that there is a comma between the qualifiers.	The comma is now required. A missing comma causes the compiler to generate an "Invalid expression" error.
All	If you use .IF with a compile-time expression using ^SIZE or %LEN...	Correct your code.	This is no longer supported and causes the compiler to generate a "Function <i>name</i> is not supported in .IF" error.
All	If you use the following syntax: <pre>record array , [10]a5</pre> <pre>array[] (ix:1) = value</pre>	Change your code. In this example, you can add a GROUP/ENDGROUP pair around the array declaration: <pre>record group array_grp array , [10]a5 endgroup</pre> and then change the code to <pre>array_grp (ix:1) = value</pre>	This syntax is no longer supported and causes the compiler to generate an "Invalid expression" error.
All	If your code uses brackets [] to access the elements of a pseudo array...	Use parentheses () instead.	Brackets can be used only for accessing the elements of real and dynamic arrays. Using them to access the elements of a pseudo array causes the compiler to generate an "Incorrect number of dimensions" error.
All	If your code includes arrayed fields larger than 256 MB...	Change your code.	Arrayed fields larger than 256 MB are no longer supported and cause the compiler to generate a "Size of field <i>name</i> too large" error.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If you use parentheses to reference elements of a structure in a ^M reference... For example, <code>^m(quote_lines(ix), rec)</code>	Change your code to use brackets instead.	This syntax is no longer supported and causes the compiler to generate an "Indexing not allowed in structures" error (error number 683).
All	If your code includes parameterized macros that have empty parameter lists... For example, <code>.define myMac() %char(8)</code>	Remove the empty parentheses.	Empty parentheses are no longer permitted when no parameters are specified in a .DEFINE parameterized macro reference and cause the compiler to generate an "Invalid compiler directive" error.
All	If your code includes nested parameterized macros of the same name...	Change your code.	In version 9, nested use of parameterized macros with the same name will cause the compiler to generate a "name not found" error.
All	If you have inadvertent unary + operations on alpha operands... For example, <code>xcall fred ("ABC"+ & + "DEF")</code> Or <code>xcall fred ("ABC" &, + "DEF")</code>	Correct your code to remove the unnecessary unary +.	This operation is mathematically invalid and is most often the result of a coding error. It causes the compiler to generate a "Numeric operand required (unary +)" error. To allow the old usage, use the -qrelaxed compiler option.
All	If your code uses the dollar sign (\$) on function calls...	Change it to a percent sign (%).	The dollar sign is no longer supported and causes the compiler to generate an "Invalid calling convention" error. To allow the old usage, use the -qrelaxed:deprecate compiler option.
All	If your code uses \$A or \$D...	Change the dollar sign to a caret sign (^A, ^D).	
All	If you use .DEFINE to simulate a Boolean operator... For example, <code>.define either or . . . if var1 .either. var2 nop</code>	Correct your code.	This syntax is no longer supported and causes the compiler to generate an "Invalid expression" error.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If you use an old form of the .PROC compiler directive that specifies the number of I/O buffers on RSTS... For example, <code>.proc (4)</code>	Remove the number from your code.	This is no longer supported and causes the compiler to generate an "Invalid record field" error.
All	If your code passes a numeric non-implied literal to a routine that declares it as n and uses ^D , ^F , or ^A on the passed argument without first determining its data type...	Make the argument type n . or use ^DATATYPE to determine the argument type before casting it.	Previously, literals were passed as decimals. Now, they are passed as integers, so the results will not be as expected.
All	If your code passes the <i>line</i> argument to MODNAME and declares it as a ...	Declare it as n instead.	Previously, <i>line</i> could be a or n ; now, only n is permitted.
All	If your code uses ^PASSED on a local variable...	Change your code.	Previously, this always evaluated to true. Now, it causes the compiler to generate an "Invalid argument" error because ^PASSED is intended for parameters only.
All	If you have files with more than 262,143 lines in them...	Split them into smaller files.	Previously, line numbers in such large files were misreported in errors. Now, the compiler generates an "Exceeded maximum number of lines in file <i>name</i> " error.
All	If you have duplicate global data sections, global commons, or global literals that are not the same size...	Change your code.	In version 9, the linker will generate an error when it finds duplicate symbols that are not the same size. The exact error depends on which symbol is duplicated. (Duplicate symbols that <i>are</i> the same size will cause the linker to generate a warning.)
All	If your code includes decimal and implied decimal variables in a single SET statement...	Correct your code to use separate SET statements for each data type.	Previously, the precision would be truncated when setting the decimal variable, without an error being reported. Now, the compiler generates a "Type mismatch between <i>type</i> and <i>type</i> " error.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If you use the DBLTOKENS environment variable...	None.	DBLTOKENS is not used by the version 9 compiler, so this environment variable has been retired.
All	If your code passes a ^VAL parameter to a ^REF argument in order to use the result of a call to a system DLL that returned a pointer...	Access the pointer using %MEM_PROC(DM_REG) instead.	Passing a ^VAL to a ^REF argument is not supported in version 9 and will cause the compiler to generate a “^REF required for argument <i>name</i> ” error.
All	If your code includes a relative range specification that accesses characters preceding the specified field...	Correct your code to range within the bounds of the field.	Attempting to access characters that precede the specified field now causes the compiler to generate an “Invalid range specified” error. Although this type of access has been deprecated, the Synergy Runtime will continue to support it for backward compatibility. We recommend that you compile and run with -qcheck and make the necessary changes to avoid future memory access violations.
All	If your program is built with version 8.3.1 through 8.3.1b and uses integer FOR loops using literal range values (FROM-THRU- STEP)... (8.3)	Recompile to take advantage of a fix for this problem.	In versions 8.3.1 through 8.3.1b, the compiler did not properly optimize FOR loops that used literal range values, which resulted in a significant performance degradation for those loops.
All	If you are experiencing incorrect line numbers in the debugger... (8.1)	Recompile to take advantage of a fix for this problem.	In 8.1.3, we fixed a bug that occurred when a program .INCLUDEd two files, and the first file contained only one line.
All	If you have programs that do not have the .PROC statement in the primary source file... (8.1)	Recompile to take advantage of a fix for this problem.	This situation may cause incorrect line numbers in a traceback.
All	If you do string concatenation on a 32-bit system where the result could be greater than 65,535 bytes... (8.1)	Modify your code to trap for the error \$ERR_BIGALPHA.	Previously, we did not generate an error in this situation. This change may break your code.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If you use the Synergy HTTP document transport API, and you have implemented your own escape mechanism for special characters... (8.1)	Set the HTTP_NOESCAPE environment variable to any value to continue using your own escape mechanism.	The HTTP document transport API now automatically escapes certain characters in URLs. If you have implemented your own escape mechanism, this change may break your code.
Windows	If you pass more than nine arguments with %AX_CALL...	Remove the characters < and > that were used to pass the additional arguments.	%AX_CALL is no longer a macro. It now supports an unlimited number of arguments, so the <> characters are no longer required. Using <> causes the compiler to generate an "Invalid expression" error.
Windows	If your code uses the DWP_ADDFILE function and your application provides the preview function... (8.1)	We recommend passing the enhanced_preview flag to DWP_BEGINJOB.	Truncation of long lines may occur in the previewer if this flag is not set.
Windows	If you rely on the default Synergy runtime color definitions... (8.1)	To maintain the same color scheme, define the specific colors you want the application to use in your synergy.ini file.	We removed the Synergy color and palette definitions from the distributed synergy.ini file in version 8.3. The new default colors are slightly different than they were previously. Consequently, you may notice a change in the appearance of applications that rely on the default colors.
Windows	If your code uses i1 variables to store the color values returned by the WI_PALET subfunction to W_INFO... (8.1)	Change these variables to at least an i2 or a d3 . (We recommend using an i4 .)	To accommodate the increased number of Synergy colors that were added in version 8.3, the WI_PALET subfunction to W_INFO may now return values that are too large for i1 variables. This change may break your code.
Windows UNIX	If you rebuild the Synergy runtime...	Create a DLL or shared object containing your C routines and access it via the Synergy DLL API.	The makedbr and makedbs files are no longer distributed. The DLL API was enhanced in version 8.1 to assist you in creating C DLLs and shared objects that interact with the runtime's C interface.
Windows UNIX	If you're using REV 2 or REV 3 ISAM files...	Convert your ISAM files to the REV 4 level or patch them to the REV 5 level.	Although REV 2 and REV 3 files are still supported, extra overhead is required to maintain them. See "Converting to ISAM REV 4 or Patching to REV 5" on page 109.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
Windows UNIX	If you need to create files that will be recognized by Synergy/DE versions prior to 6.3.0d...	Set the environment variable ISAMC_REV to 3 for each Synergy user who may be creating a file.	Versions 6.3.0d and higher support REV 4, which became the default value for ISAMC_REV starting with version 7.1. For more information, see ISAMC_REV in the “Environment Variables” chapter of <i>Environment Variables and System Options</i> .
Windows UNIX	If you have any source modules that contain a global data section that doesn’t have an “,INIT” but does contain an .ALIGN directive, and the object file is linked into an ELB... (8.1)	Recompile to take advantage of a fix for this problem.	The described scenario caused unpredictable results because the compiler erroneously emitted data over other pieces of compiler information.
Windows UNIX	If you have duplicate global data sections, global commons, or global literals that are not the same size...	Change your code.	In version 9, the runtime will detect duplicate symbols when ELBs are loaded and generate a BADEL error if the size is different. The error text tells you which symbols are duplicated. (Duplicates that <i>are</i> the same size will cause a warning in debug output.) This change may break your code.
Windows UNIX	If you have duplicate static records that are not the same size...	Recompile and relink.	
Windows UNIX	If you use READ or READS from a relative file when the file record size is greater than the size of the data area...	Correct your code or set system option #27.	This situation now generates an “Input data size exceeds destination size” error, as documented in the <i>Synergy Language Reference Manual</i> . This error can also occur when a file is opened in the default mode with RECSIZ set. This change may break your code.
Windows UNIX	If you call the COPY routine and specify a destination file that another process might have open... (8.1)	Modify your code to handle a possible “File in use” error.	A “File in use” error is now returned if a COPY is performed on an open file. This change may break your code.
Windows UNIX	If you use %DLL_SUBR... (8.1)	Change your code to use %DLL_CALL instead.	%DLL_SUBR has been removed. This change may break your code.
Windows UNIX	If you use %DATETIME and the resolution on your system for milliseconds is not six digits... (8.1)	Review your code to see if it needs to be modified.	%DATETIME has been changed to fill with zeroes instead of spaces if the system resolution for milliseconds is not six digits.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
UNIX	If you use the FORK subroutine or the optional <i>fork</i> argument of the %HTTP_SERVER_RECEIVE function...	Change your code to reopen any files needed by the child process.	As a result of a change we made to fix random I/O failures, the FORK subroutine and the <i>fork</i> argument to the %HTTP_SERVER_RECEIVE function now close all channels opened by the parent in the forked child process. This change may break your code.
UNIX	If you pass Windows-style commands that contain "\\\" to the UNIX shell... (8.1)	Use "\\\" instead.	The SHELL subroutine now correctly handles shell environment variables (\$DBLDIR). This change may break your code.
UNIX	If you use %TNMBR... (8.1)	Examine your code to ensure the value returned by %TNMBR is not being used for static data.	On some UNIX systems, %TNMBR was returning the wrong value. This change may break your code , as it may change the terminal numbers returned on a particular system.
UNIX	If you use the JBWAIT environment variable... (8.1)	None.	XCALL RUNJB no longer waits 2 seconds (or the value of JBWAIT). Hence, the JBWAIT environment variable is no longer necessary and has therefore been retired.
UNIX	If your application takes advantage of the fact that file locking between channels within the same process is not enforced... (8.1)	Set the INTRAFILELOCKS environment variable to 0 (zero).	We changed the default behavior for file locks within a process; it is now the same as the behavior for file locks between processes. This change may break your code. You can revert to the previous behavior by setting INTRAFILELOCKS to 0. We do not recommend this, although it does permit you to continue executing version 8.1.7 and earlier applications that violate these new locking rules. For more information, see INTRAFILELOCKS in the "Environment Variables" chapter of <i>Environment Variables and System Options</i> .

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
OpenVMS	If the audit server is not running on your machine...	Start the audit server and leave it running.	If the audit server is not running, you'll see the internal C++ runtime error "Create_Exclusive_Lock_1" when you compile. This error occurs because the version 9 compiler uses the C++ runtime. Even though the message looks serious, it does not affect the compilation of your code.
OpenVMS	If your code uses a lowercase logical to define the location of a repository in a .INCLUDE statement...	Change the logical to uppercase in the .INCLUDE statement or add a second define in which the logical is lowercase and quoted (which preserves the case). For example, \$ define MYRPS "<path>" \$ define "myrps" "<path>"	Prior to version 9, the compiler automatically uppercased the logical used for the repository location in a .INCLUDE statement. This ensured that it matched the logical definition, since DCL uppercases the command line by default. In version 9, the compiler does not uppercase the logical in the .INCLUDE statement. Consequently, if this logical is lowercase in your code, the compiler generates a "Logical is not defined" error. This change was made in anticipation of future support for ODS-5 case-sensitive filenames.
OpenVMS	If your code has embedded characters (e.g., CR, LF, ESC, NULL) in a quoted literal... For example, record crlf ,a2 , "<cr><lf>"	Use string arithmetic and %CHAR to pass the value that corresponds to the ASCII character. For example, record crlf ,a2 , %char(13)+ & %char(10)	Embedded non-printing characters are not supported by the version 9 compiler. The results of using them may be unpredictable, such as getting a space instead of the desired character.

Converting to ISAM REV 4 or Patching to REV 5

Converting version 2 or 3 ISAM files to REV 4 or patching them to REV 5 will improve performance. We recommend using **fconvert** to convert the files. Conversion, however, can be a lengthy process. It is much quicker to patch the files with **isutl**. A version 2 or 3 ISAM file that has been patched to version 5 can be used with Synergy/DE version 7.5 and higher. However, you will not be able to make additional parameter changes to a patched file.

Using fconvert to convert to REV 4

Fconvert enables you to convert your existing ISAM files and add on some of the new ISAM characteristics (such as density and index block size).

1. Either unset the environment variable ISAMC_REV or set it to 4. (**Fconvert** creates files that are the current REV level or the REV level defined by ISAMC_REV.)
2. Generate a par file from your existing ISAM file.
3. Add the new options to the par file, as documented in [bldism](#) in the “Synergy DBMS” chapter of *Synergy Language Tools*.
4. Run **fconvert** on the file using the modified par file. For details see [fconvert](#) in the “Synergy DBMS” chapter of *Synergy Language Tools*.

Using isutl to patch to REV 5

Run **isutl** with the **-p** option on the desired files. For details see [isutl](#) in the “Synergy DBMS” chapter of *Synergy Language Tools*.

Upcoming Changes

A future version of Synergy/DE will support Microsoft .NET. As a result, some current features will not be supported in that version. These include deprecated data types, syntax, APIs, compiler options, and alignment warnings. You can see how compatible your code will be with .NET by compiling with the **-qnet** option. It will output warnings to standard error for items that will not be supported.

Workbench

Compatibility Information

If you customized Workbench version 8, you must upgrade your existing installation (rather than uninstall) to retain those customizations. See “Before you install Workbench 9.3” in the Workbench release notes (REL_WB.TXT) for instructions on how to retain your customizations when you upgrade.

Required Changes

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
Windows	If you have version 8 Synergy projects...	Convert the projects to version 9.3 by opening them in Workbench.	Workbench version 9.3 uses SlickEdit version 14, and therefore requires that all existing Synergy projects be converted. Projects are converted automatically when they are opened.
Windows	If you have written your own macros...	After installing, reload them into Workbench. For each macro, at the Workbench command line enter the following: <code>load macro_name.e</code>	Macros cannot be converted.
Windows	If you have Synergy .NET Component projects built with version 1.1 of the .NET Framework...	Install .NET Framework version 2.0.	Version 2.0 of the .NET Framework is now the minimum supported version. When you open a project created with 1.1, you'll be prompted to select the correct environment batch file and the target Framework version.
Windows	If you use the format %VAR% to refer to an environment variable in a project open command... (8.1)	Use the format %(VAR) instead.	SlickEdit now requires a different format.
Windows	If you rely on SlickEdit to create user1 and user2 menu commands for all projects when they are created... (8.1)	Create the user-defined tool entries manually by entering a new tool name of “user 1” and “user 2” in the Tool name field of the Project Properties dialog.	SlickEdit no longer performs this function automatically.
Windows	If you open Workbench projects in Composer... (8.1)	Instead, launch Composer from Workbench to modify a specific script file, then return to Workbench.	Composer no longer supports Workbench projects. However, Workbench can pass Composer the name of a script file to open.

UI Toolkit

Compatibility Information

If you include the Toolkit `gblctl.def` or `lstprv.def` file in your programs, you must recompile. See [“Upcoming Changes” on page 115](#) for more information on these files.

Required Changes

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If you overload any Synergex-supplied function or subroutine...	Verify that your overloaded routine exactly matches the prototype of the Synergex-supplied routine in return type, and in number, direction, and type of parameters, as well as whether parameters are optional or required.	We now strongly prototype all Synergex-supplied functions and subroutines. Using the -W4 compiler option (/WARNINGS=4 on OpenVMS) will show the prototype the compiler expects and your overloaded routine definition, so that you can see how they differ. Once you change your routine, your code may no longer compile in versions 8.3 and earlier. See “Compatibility Information” on page 94 for a suggestion on how to handle this situation.
All	If you undefine <code>TOOLS_INIT</code> in your code...	Remove the <code>.UNDEFINE</code> .	<code>TOOLS_INIT</code> is now undefined in tools.def . If you also undefine it in your code, the compiler generates an “Identifier not defined” error. Once you remove the <code>.UNDEFINE</code> , your code may not work properly in versions 8.3 and earlier. See “Compatibility Information” on page 94 for a suggestion on how to handle this situation.
All	If you pass the <i>field_id</i> and <i>selection_id</i> arguments to <code>IB_END</code> ...	Remove these arguments.	These arguments have not been supported (and have had no affect on the operation of your code) since Toolkit version 3.3. Their use now causes the compiler to generate a “More argument values than declared parameters in routine” error.
All	If you link against tklib.olb ... (8.1)	Link against tklib.elb (Windows and UNIX) or tklib_sh.exe (OpenVMS) instead.	We ceased distribution of tklib.olb in version 8.3. We also no longer distribute the files chninf.dbl , chninf.def , and fkey.dbl because they were used only with tklib.olb . This change may break your code.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If you use chninf.def to find the highest channel number that Toolkit can use... (8.1)	Change your code to use the new D_CHANNEL_MAX subfunction to %E_INFO.	We ceased distribution of chninf.def in version 8.3. This change may break your code.
All	If you use the left or right square bracket character as a field delimiter in the routines I_DISABLE, I_DISPLAY, I_ENABLE, I_INIT, or I_SETDEL... (8.1)	Use a different delimiter character.	Toolkit now permits you to create fields that reference specific elements in repository arrayed fields. To do this, you must use square brackets; consequently, these routines no longer recognize square brackets as delimiters in field lists. This change may break your code.
All	If you pass a null string for the I_FORCE <i>new_data</i> argument, and you depend on there being a space in the field... (8.1)	Change your code to pass a space instead.	Previously, a null string in the <i>new_data</i> argument was interpreted as a space (" "); now, it is interpreted as a null string (""). This change results from a feature that enables you to clear the force buffer and cancel a pending I_FORCE call by passing a null string. This change may break your code.
All	If your Toolkit application includes a customized version of any of the user-replaceable list subroutines (these are the subroutines that begin with USR_L)... (8.1)	Remove any references to those routines and to the lstprv.def file.	We ceased distribution of the user-replaceable list subroutines in version 8.3. If your application includes a customized version of any of these routines, Toolkit will simply ignore it and call the corresponding default routine (e.g., it will call L_DATA instead of USR_LDATA). We also no longer distribute the files lstusr.dbl , lstprv.dbl , and lstbuf.def . This change may break your code. See also "Upcoming Changes" on page 115 .
All	If your application exceeds the maximum number of entry lists or menu column rows in the MB_ routines...	Correct your code.	Previously, the maximums (9 for lists and 20 for rows) were not enforced. Exceeding the maximums could cause unpredictable results, but there was no error. Now a "Too many lists" or "Too many rows" error is returned. This change may break your code.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If your code passes a value other than 0 or 1 for the <i>no_place</i> argument to <i>MB_COLUMN...</i> (8.1)	Change your code to pass a value of 1.	The name of this argument has changed from <i>no_place</i> to <i>option</i> , and it now is used to determine both column loading and placement options. Previously, any value greater than 0 would cause the column to be loaded without placing it; now, you must pass 1 to get that behavior. This change may break your code.
All	If your code creates blank menu entry names... (8.1)	Correct your code so that all menu entries are named.	Passing a blank menu entry to <i>MB_ENTRY</i> causes a fatal error. This change may break your code.
All	If you have code that calls <i>U_SELDFLT...</i> (8.1)	Update your code to call <i>S_SELDFLT</i> instead.	We removed the <i>U_SELDFLT</i> routine, which was renamed to <i>S_SELDFLT</i> in Toolkit version 3.0. This change may break your code.
All	If you use the <i>g_werr</i> record in gblctl.def...	Define your own record to use with the <i>W_RETERR</i> function.	We removed <i>g_werr</i> . This change may break your code. For information on <i>W_RETERR</i> , see “Error Processing in the Windowing System” in the “Synergy Windowing API” chapter of the <i>Synergy Language Reference Manual</i> . See also “Upcoming Changes” on page 115 for more information about gblctl.def .
All	If your application accesses the <i>gs_dec</i> member of fldinf.def...	Recompile your code.	To support a change in the maximum size of decimal and implied-decimal fields in Toolkit, we changed <i>gs_dec</i> from a d1 to an i1 .
Windows	If you specify color palette entry 16 for a list header or footer...	Specify color palette entry 1 to retain the current look.	Previously, color palette entry 16 did not work properly, which caused the default (1) to be used instead. We fixed entry 16, which could cause a change in the appearance of your application.
Windows	If you use the ActiveX list control and have non-static headers or footers...	Adjust the position of the text as necessary.	Text is now centered in the column by default. Previously, it was left-justified.
Windows	In an ActiveX control, if you use <i>%M_SIGNAL</i> in an event method triggered by <i>WM_KEYDOWN</i> for a key that will be translated into a character... (8.1)	User <i>WM_CHAR</i> to trigger the event method instead.	Using <i>%M_SIGNAL</i> in response to a keydown event causes the keypress to remain in the message queue, which could cause duplicate characters.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
Windows	If you depend on an input method being applied to the subsequent field or button when a menu entry is signaled from a method during L_INPUT processing and the user action moves context to the subsequent field... (8.1)	Change your code. Or, to revert to the previous behavior, set the RETAIN_CONTEXT_CHANGE_ON_SIGNAL environment variable or the g_retaincontext global variable.	We corrected a problem that caused a method to be applied to the following field when the user moved context to that field. If your application depended on this behavior, this change may break your code.
Windows	If you override cell color and font for an ActiveX list control and then call L_RESTART... (8.1)	Reapply the overrides after the call to L_RESTART, if you want to retain them.	Cell color and font overrides are now reset to the default after a call to L_RESTART.
Windows	If you rely on the default Synergy runtime color definitions... (8.1)	To maintain the same color scheme, define the specific colors you want the application to use in your synergy.ini file.	We removed the Synergy color and palette definitions from the distributed synergy.ini file in version 8.3. The new default colors are slightly different than they were previously. Consequently, you may notice a change in the appearance of applications that rely on the default colors.
Windows	If your code uses i1 variables to store the color values returned by U_REND and the W_PALET subfunction to W_INFO... (8.1)	Change these variables to at least an i2 or d3 . (We recommend using an i4 .)	To accommodate the increased number of Synergy colors in version 8.3, U_REND and the W_PALET subfunction to W_INFO may now return values that are too large for i1 variables. This change may break your code.
Windows	If your application uses tabbed dialogs... (8.1)	You may want to use the g_tabfacecolor global variable in tkctl.def to control which color palette entry will be applied to a window when it is added to a tabbed dialog.	Due to changes we made to the color palette in 8.3, on a Windows system with themes enabled, you may notice a change in the appearance of your application when a window is added to a tabbed dialog.
Windows	If your application has tab sets with the tabs on the bottom or side... (8.1)	Move the tabs to the top of the tab set.	Tabs are supported only at the top of the tab set on systems that support Windows themes. This change may break your code.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
UNIX OpenVMS	If your application has an input window with more than 31 fields, and the 32nd field has an arrive method... (8.1)	Recompile the window script to take advantage of a fix for this problem.	We corrected a problem that caused the arrive method on the thirty-second field to sometimes fail to be invoked.
UNIX OpenVMS	If you have an application with a check box with field-level renditions specified...	Set the new D_INPPRCRND state for E_STATE to D_OFF if you want to retain the current look of the prompt.	In 9.3, we added the D_INPPRCRND state to E_STATE. By default, this state is set to D_ON on UNIX and OpenVMS, which causes Toolkit to use field-level renditions for check box prompts. Previously Toolkit used renditions for the part of the window that the prompt occupied.

Upcoming Changes

In an upcoming major version of Synergy/DE, we plan to remove the following files from the distribution:

- **gblctl.def.** This file contains an internal structure that gets updated on each release, thereby requiring that you recompile.
- **lstprv.def.** This file contains defines for lists. These defines were required for the user-defined list subroutines, which are no longer a part of Toolkit. You should modify your code to remove references to this file.

For more information, see the “Future Revisions” section of the Toolkit release notes (**REL_DTK.TXT**).

Composer

Required Changes

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
Windows	If you open Workbench projects in Composer... (8.1)	Instead, launch Composer from Workbench to modify a specific script file, and then return to Workbench.	Composer no longer supports Workbench projects. However, Workbench can pass Composer the name of a script file to open.

Repository

Compatibility Information

- ▶ Repositories created in version 8 do not need to be converted to be used with version 9. Version 9 Synergy/DE tools can open repositories created with versions 7, 8, and 9.
- ▶ SDL (Synergy Data Language) files do not need to be converted.
- ▶ If you have customized Repository's window library, you must reapply your customizations and rescript it.

Required Changes

No changes are required.

Upcoming Changes

In an upcoming version of Synergy/DE, we are planning to remove the Repository object library (OLB) file **ddlib.olb**. It has been replaced with the executable library (ELB) file **ddlib.elb**. (The OLB file will be available on the Synergex web site.)

You should be linking against the ELB file instead of the OLB file for the following reasons:

- ▶ An ELB will reduce the size of your application if you link multiple programs against the library.
- ▶ Linking against an ELB is more reliable than rebuilding the OLB and making a custom ELB.
- ▶ An ELB provides more compatibility when multiple Synergy applications from different companies coexist on one system.
- ▶ In most cases, you can upgrade to newer versions of Synergy/DE without recompiling and relinking your applications.

ReportWriter

Compatibility Information

No conversion or rescripting is required.

Required Changes

No changes are required.

xfServer

Compatibility Information

- xfServer 9 supports clients running Synergy/DE version 6.1 and higher.
- (Windows) If you have previously had xfServer, and you install version 9 to a different directory, you must remove and re-add your xfServer services.

Required Changes

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If you reference one remote file specification from another when the first file specification is on Windows... (8.1)	Cease to do so.	This is no longer permitted because of the potential for data corruption.
Windows	If the RUSER environment variable is set in both the registry and the environment... (8.1)	Be aware that RUSER set in the environment now has precedence over RUSER set in the registry.	Note that when RUSER is set in the Synrc node in the registry, as is required to configure xfServerPlus for remote data access, it is read into the environment when rsynd starts up, and it is therefore considered to be set in the environment. This change may break your code.
Windows	If RUSER is set to the user name "SSPI" and no password is specified... (8.1)	Change the user name.	To support Windows authentication security, RUSER recognizes "SSPI" as a special value.

xfServerPlus

Compatibility Information

- xfServerPlus version 9.3 can receive routine calls from version 7.1 and higher xfNetLink clients.
- Synergy Method Catalog (SMC) files must be converted for use with xfServerPlus and Workbench 9.3 and with version 9.3 utilities such as the Method Definition Utility (MDU) and `smc_elb.exe`. See the “SMC conversion utility” section of the xfServerPlus release notes (**REL_XFPL.TXT**) for instructions.
- (Windows) If you have previously had xfServerPlus, and you install version 9 to a different directory, you must remove and re-add your xfServerPlus services.

Required Changes

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If you use the “Add new interfaces” option in the Import function of the Method Definition Utility...	Use the “Update interfaces” option instead and then choose the new interfaces you want to add.	The MDU now has an Update interfaces option, which updates existing interfaces, in addition to adding new interfaces.
All	If you use the Reports feature in the Method Definition Utility... (8.1)	Run the Export Methods option to generate an XML file, and then view the file in an XML viewer.	We removed the Reports feature.
Windows	If the RUSER environment variable is set in both the registry and the environment... (8.1)	Be aware that RUSER set in the environment now has precedence over RUSER set in the registry.	When RUSER is set in the Synrc node in the registry, as is required to configure xfServerPlus for remote data access, it is read into the environment when rsynd starts up, and it is therefore considered to be set in the environment. This change may break your code.
Windows UNIX	If you use the xfServerPlus log file... (8.1)	Expect to see multiple log files.	By default, xfServerPlus now generates a separate log file for each session. You can set the XFPL_SINGLELOGFILE option to ON in the xfpl.ini file to continue producing a single log file.
UNIX	If you start xfServerPlus with a password... (8.1)	Encode the -u password with setruser .	To improve xfServerPlus security, a clear text password is no longer permitted when starting xfServerPlus. This change may break your code if you start xfServerPlus with a script.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
UNIX	If you start <i>xfServerPlus</i> without a password... (8.1)	The user starting <i>xfServerPlus</i> must be root or signed on as the user specified with -u .	This is the documented behavior, but it was not previously enforced. This change may break your code if you start <i>xfServerPlus</i> with a script.
OpenVMS	If you run <i>xfServerPlus</i> on OpenVMS... (8.1)	The <i>xfServerPlus</i> account must have the SHARE privilege.	Without this privilege, <i>xfServerPlus</i> will not run. This change may break your code.

xfNetLink Java Edition

Required Changes

No changes are required.

Upcoming Changes

The port number range feature (minport and maxport) will be deprecated in a future version. We recommend that you quit using it.

xfNetLink COM Edition

Compatibility Information

If you have previously had xfNetLink COM, and you install version 9 to a different directory, you must unregister and reregister your type libraries.

Required Changes

No changes are required.

Upcoming Changes

The port number range feature (minport and maxport) will be deprecated in a future version. We recommend that you quit using it.

xfNetLink .NET Edition

Compatibility Information

Version 8 assemblies (32-bit only) can be used in version 9.3 without being rebuilt. However, if you want to take advantage of the new data types, methods, and other features available in 9.3, you must regenerate classes and rebuild your assembly. See the xfNetLink .NET release notes file, **REL_XFNN.TXT**, for details.

Required Changes

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
Windows	If your application passes structures that contain fields defined as binary data type in the repository...	To continue to have these fields converted to strings in the generated C# classes, run gencs with the -nb option; else, they will be converted to byte arrays.	A field in a structure that is defined as a binary data type in the repository is now converted to a byte array in the generated C# code instead of a string. This change may break your code if you regenerate classes for an existing application.
Windows	If you use the port number range feature (minport and maxport)...	Cease to do so.	This feature has been deprecated. Minport and maxport are ignored when set in the configuration file or passed with the connect() method.
Windows	If you use version 1.0 or 1.1 of the .NET Framework...	Upgrade to version 2.0.	.NET Framework 2.0 is now the minimum supported version.
Windows	If you use version 1.0 or 1.1 of the .NET Framework...	Before regenerating classes for an existing assembly, edit the AssemblyInfo.cs file to remove the attributes AssemblyDelaySign and AssemblyKeyFile. Or, you can delete the AssemblyInfo file; xfNetLink will create a new one the next time you run gencs .	In .NET Framework versions 1.0 and 1.1, these two attributes were included in the AssemblyInfo file. In version 2, they are specified on the compiler command line. Because the AssemblyInfo file is no longer overwritten, you must either delete these attributes from the file or delete the file so that it is recreated.
Windows	If you have fields in Repository that are flagged as read-only, and you generate structure members as properties, and you need to be able to set those properties in your client code...	Clear the read-only fields in Repository before you generate classes.	Fields flagged as read-only in Repository are generated as read-only in the C# classes when structure members are generated as properties. These properties will have a "get" method, but no "set" method. This change may break your code if you regenerate classes for an existing application. Note that the read-only flag is not honored when you generate structure members as public fields.

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
Windows	If you use client-side logging... (8.1)	Expect to see multiple log files.	By default, <i>xfNetLink .NET</i> now generates a separate log file for each class that instantiates a connection to <i>xfServerPlus</i> . You can set "single log file" to "on" in the application configuration file to continue producing a single file.
Windows	If your Synergy assembly includes methods that pass structures as parameters... (8.1)	Add a reference to xfnlnet.dll in your Visual Studio .NET project.	This requirement results from two new features added in 8.3, the gens -i option and the generation of a C# interface for every generated class.

SQL OpenNet

Compatibility Information

- ▶ All Connectivity Series products for client and server must be upgraded to version 9.3. Otherwise, a version mismatch error occurs.
- ▶ (Windows) If you have previously had Connectivity Series, and you install version 9.3 to a different directory, you must remove and re-add your SQL OpenNet service.

Required Changes

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
Windows	If you use vtxnet2... (8.1)	Change the start-up command line in the opennet.srv file to specify vtxnet2 .	We changed the default SQL OpenNet server from vtxnet2 to vtxnetd . This change may break your code.

xfODBC

Compatibility Information

- All Connectivity Series products for client and server must be upgraded to version 9.3. Otherwise, a version mismatch error occurs.
- You must re-create your system catalogs for use with version 9.3 by running the **dbcreate** utility with the **-x** option (or by using the “Add new tables and update existing tables” option in the DBA program).

Required Changes

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If you set SODBC_NONULL...	Review “ Preventing null updates and interpreting spaces, zeros, and null values ” in the “Preliminary Steps” chapter of the <i>xfODBC User’s Guide</i> . Then set “Null allowed” values in Repository and the SODBC_NONULL environment variable as necessary.	The dbcreate utility now uses the new Repository “Null allowed” field to determine how to set the “Null allowed” property for system catalog columns. When generating a system catalog, the SODBC_NONULL environment variable setting is used only for repository fields with “Null allowed” set to Default. We changed the default for SODBC_NONULL to 4, and we also changed the behavior for some SODBC_NONULL settings.
All	If you do not set SODBC_NONULL...	Set SODBC_NONULL to 0 to maintain your current “Null allowed” settings for system catalog columns.	
All	If you use SQL_REAL for the CAST and CONVERT functions...	Use SQL_FLOAT instead.	xfODBC no longer supports SQL_REAL.
All	If your data includes unsigned fields that contain negative values... (8.1)	Change the fields to signed fields by selecting the “Negative allowed” option in Repository. Or, you can set the SODBC_NOUNSIGNED environment variable to revert to the previous behavior.	The dbcreate utility and xfODBC driver now distinguish between signed and unsigned fields. This prevents negative values from being stored in non-negative fields, which enables xfODBC to work better with ADO.NET and Microsoft Access.
OpenVMS	If you have created your own XFODBCUSR.EXE file... (8.1)	Rename the file XFODBCUSR_SO.EXE and set the logical XFODBCUSR_SO (instead of XFODBCUSR) to point to it.	We changed the name of the XFODBCUSR.EXE file (and the corresponding logical) to XFODBCUSR_SO.EXE . This change may break your code.

SQL Connection

Compatibility Information

- All Connectivity Series products for client and server must be upgraded to version 9.3. Otherwise, a version mismatch error occurs.
- If you use SQL Connection with SQL Server, we recommend that you install Service Pack 4 for SQL Server 2000. This service pack contains several Synergy/DE-specific fixes.

Required Changes

Platform	Version 8 usage	Change required for 9.3	Explanation/Comments
All	If you use the %SSC_BLOB routine...	Use the new routine %SSC_LARGECOL instead.	The new routine and option improve SQL Connection's ability to get and put large binary column data and large character column data. See %SSC_LARGECOL in the "Database Functions" chapter of the <i>SQL Connection Reference Manual</i> for details.
All	If you use the SSQL_BLOBCOL option for %SSC_OPEN...	Instead, use the new SSQL_LARGECOL option for %SSC_OPEN for gets and the SSQL_LARGECOL option for %SSC_EXECUTE for puts.	
Windows	If you use VTX12 or the sqlserver keyword in your connect string... (8.1)	Update your program to use VTX12_ODBC or change the default driver in the Synergy Configuration Program to continue using a DB-Library-based driver.	We changed the default database driver for SQL Server from a DB-Library-based driver to an ODBC-based driver (VTX12_ODBC). This is the driver that is used when a connect string specifies VTX12 or sqlserver. This change may break your code. For more information, see "Using your program with different drivers and databases" in the "Creating SQL Connection Programs" chapter of the <i>SQL Connection Reference Manual</i> .
OpenVMS	If you use Oracle or Oracle Rdb natively... (8.1)	Rebuild your database driver using BUILD_SSQL_DB.COM .	We have moved to a shared image model for database drivers on OpenVMS. This change may break your code. For instructions on building a database driver, see "Configuring and Building SQL OpenNet for SQL Connection" in the VMS section of the "Configuring Connectivity Series" chapter of the <i>Installation Configuration Guide</i> .