QA C 8.0 Release Notes

Contents

1. Introduction

The 8.0 release of QAC features a number of incremental improvements including new analysis messages and bug-fixes; however its main emphasis is a new dataflow analysis engine which provides a greatly improved analysis capability coupled with a much reduced incidence of false positive messages.

2. DF^2 – Deep Flow Dataflow Analysis

This release introduces a comprehensively revised dataflow analysis engine based on a Satisfiable Modulo Theories (SMT) solver which incorporates significant advantages over previous dataflow implementations.

Problem Identification:

The introduction of solver technology has improved the effectiveness of dataflow analysis in identifying real problems.

False Positives:

The incidence of false positives has been dramatically reduced.

Messages:

The set of messages associated with dataflow analysis has been completely superseded by a larger set containing new message numbers; the original messages may still be generated, but their functionality is sometimes a little different. Usage of the old messages is now deprecated and they may be withdrawn in future releases.

A full description of the new dataflow message system is provided in the Dataflow User Guide.

Sub-messages:

Dataflow messages are now supported by sub-messages in order to supply key information about the context in which a dataflow condition is identified.

Pointer Tracking

The attributes of pointers are tracked enabling identification of undefined behaviour such as array bounds violations and subtraction or comparison of pointers which address different objects.

Loop Modelling

The tracking of values within loop constructs has been substantially improved.

Standard Library Call Checking

Semantic checking is now conducted on the arguments to functions defined in the C Standard Library and key attributes of returned values are recognised in subsequent analysis.

Analysis Modes

It is now possible to analyse either with or without dataflow. Analysis without dataflow is generally quicker than in previous versions of QAC, but if dataflow is enabled, analysis times will be substantially longer.

Options

A number of new configuration options are provided to control the operation of dataflow.

Full documentation describing all these features as well as guidance about configuration options and transitioning is provided in the Dataflow User Guide.

3. Message generation

The visibility of messages at view time in QAC is controlled by a number of options including:

-n	-Nomsg
-o	-Only
-hdr	-HDRsuppress
-su	-SUppresslvl

In past versions of QAC, all diagnostics were output to the .err file during the code analysis phase regardless of the settings of these display options. It was therefore possible to change the setting of these options at viewing time and diagnostics which had previously been invisible would be made visible. The one exception to this approach was the –q option:

- -q -Quiet

The effect of the –q option has always been to suppress the generation of diagnostics at the analysis stage so that they are not output to the .err file and can never be viewed, regardless of the setting of other options.

Changes in the behaviour of the –o and –n options

In QAC 8.0 a fundamental change has been implemented in the way that message diagnostics are generated.

The –o and –n options now function in a similar manner to the –q option. If a message is suppressed by the –o or –n options, no diagnostics will be output to the .err file and those messages can never be visible at view time without performing a new analysis.

The functionality of options –hdr and -su remains unchanged; they continue to be view-time options which can only affect the visibility of diagnostics which have been output to the .err file.

This functional change has been implemented with 2 objectives in mind:

a) A reduction in the size of the .err file

b) Improved performance delivered by eliminating unnecessary steps in dataflow analysis.

Dataflow messages

One other important aspect of message generation is the introduction of the –ed option.

- -ed -EnableDataflow

As noted above in the discussion of the new dataflow engine, a capability now exists to analyse source code either with or without dataflow analysis. Dataflow analysis will not occur and dataflow message diagnostics will not be generated unless the –ed+ option is applied.

4. New Messages

The following table lists new messages which have been implemented in QAC 8.0. New dataflow messages are not listed; full details are provided in the Dataflow Users Guide.

Msg	Message Text	CR
0314	*[I] Cast from a pointer to object type to a pointer to void.*	12775
0315	*[I] Implicit conversion from a pointer to object type to a pointer to void.*	12775
0316	*[I] Cast from a pointer to void to a pointer to object type.*	12775
0317	*[I] Implicit conversion from a pointer to void to a pointer to object type.*	12775
0914	*[U] Source file does not end with a newline character.*	11547
0915	*[U] Source file ends with a backslash character followed by a newline.*	11547
0974	*[I] Cast of integer constant expression to a signed integer type which cannot represent the value.*	13786
0977	*Cast of a constant negative value to an unsigned type.*	13786
0980	*Cast of constant negative value which assumes a two's complement representation of signed values.*	13786
2021	*This tentative definition is interpreted as a declaration. Is this intended ?*	13324
2022	*A tentative definition is being used. Is it appropriate to include an explicit initializer ?*	13324
3260	*Typedef defined with more than 2 levels of indirection*	13465
3261	*Member of struct/union defined with more than 2 levels of indirection.*	13465
3262	*Object defined or declared with more than 2 levels of indirection.*	13465
3263	*Function defined or declared with a return type which has more than 2 levels of indirection.*	13465
3290	*Truncation of positive constant integer value during cast to a smaller unsigned type.*	13786
3329	*This 'if' controlling expression is a constant expression and its value is 'false'.*	13686
4240	*The value of an 'effectively Boolean' expression is being assigned to an object of type 'char'.*	11447
4241	*The value of an 'effectively Boolean' expression is being passed to a function parameter of type 'char'.*	11447
4242	*The value of an 'effectively Boolean' expression is being returned from a function with a return type of 'char'.*	11447
4243	*The value of an 'effectively Boolean' expression is being assigned to an object of floating type.*	11447
4244	*The value of an 'effectively Boolean' expression is being passed to a function parameter of floating type.*	11447
4245	*The value of an 'effectively Boolean' expression is being returned from a function with a floating return type.*	11447

5. Messages with modified behaviour

The following table summarises the messages whose behaviour has been modified.

2 categories of modification are recognized:

C – A significant change has been implemented to existing behaviour.

F – A fix of a bug or problem feature.

Msg	Message Text	T	CR
0274	*[I] Conversion of integer constant expression to a signed integer type which cannot represent the value.*	F	12560 13786
0277	*[I] An integer constant expression with negative value is being converted to an unsigned type.*	F	11215 11418 12560 13786
0280	*Use of constant negative value which assumes a two's complement representation of signed values.*	F	12560 13370 13786
0400	*[U] '%s' is modified more than once between sequence points - evaluation order undefined.*	F	1790
0401	*[U] '%s' may be modified more than once between sequence points - evaluation order undefined.*	F	1790
0402	*[U] '%s' is modified and accessed between sequence points - evaluation order undefined.*	F	1790
0403	*[U] '%s' may be modified and accessed between sequence points - evaluation order undefined.*	F	1790
0547	*[C] This declaration of tag '%s' conflicts with a previous declaration.*	F	13699
0672	*[U] The initializer for a 'struct', 'union' or array is not enclosed in braces.*	F	13269
0693	*Struct initializer is missing the optional {.*	F	13092
0694	*Array initializer is missing the optional {.*	F	13092
1271	*Using a non-int expression does not alter the type of the enum constant.*	F	11984
1335	*Parameter identifiers missing in function prototype declaration.*	F	12898
1336	*Parameter identifiers missing in declaration of a function type.*	F	12898
2547	*This declaration of tag '%s' hides a more global declaration.*	F	13747
3120	*Hard-coded 'magic' integer constant '%s'.*	F	13656
3121	*Hard-coded 'magic' floating constant '%s'.*	F	13656
3122	*Hard-coded 'magic' string literal, %s.*	F	12516
3122	*Hard-coded 'magic' string literal, %s.*	F	13656
3123	*Hard coded 'magic' character constant %s.*	F	13656
3306	*Implicit unsigned conversion on positive integer constant expression will not preserve value.*	F	11215 13370
3325	*This 'while' or 'for' loop controlling expression is a constant expression and its value is 'false'. The loop will not be entered.*	C	13686
3345	*Statement contains more than one access to objects that are volatile.*	F	13642
3673	*The object addressed by the pointer parameter '%s' is not modified and so the pointer could be of type 'pointer to const'.*	F	12815
3680	*[U] Indexing array with constant value that is out of bounds.*	F	12422 13478

6. Messages which have been removed

Msg	Message Text	T	CR
0679	*Redundant braces found in initializer.*		13715
2711	*'register' is only a hint - the compiler may well be able to do a better job of optimising - avoid!*		13683
3682	*Index may take a value greater than number of elements.*		13690
3683	*Use of constant address which points to one element beyond the end of the array.*		13690
3688	*Definite use of address which points to one element beyond the end of the array.*		13690
3692	*Apparent use of address which points to one element beyond the end of the array.*		13690

7. CR Summary

The following table summarises the change requests (CRs) that have been implemented in QAC 7.2.

CRs have been categorised into 3 types (column "T"):

C – A significant change has been implemented to existing behaviour.

F – A fix of a bug or problem feature.

N – New functionality has been introduced.

Some CRs are associated with more than one category and, in some cases, the distinction between categories is a little indistinct. The classification should therefore be treated as a guide only.

CR	T	Resolution
1790	F	*0400: [U] '%s' is modified more than once between sequence points - evaluation order undefined.* *0401: [U] '%s' may be modified more than once between sequence points - evaluation order undefined.* *0402: [U] '%s' is modified and accessed between sequence points - evaluation order undefined.* *0403: [U] '%s' may be modified and accessed between sequence points - evaluation order undefined.* Improvements have been made in the implementation of messages 0400-0403 to identify evaluation order problems which were previously unrecognised.
11122	F	A number of messages previously generated in a misleading column position are now located more intuitively. For example: *0602: [U] The identifier '%s' is reserved for use by the library.* *1414: Enum object is being compared with a constant, non-enum expression using a relational operator.* *1476: Range of possible enum values suggests this test is always false.* *3110: The left-hand operand of this ',' has no side effects.* *3344: Controlling expression is not an 'effectively Boolean' expression.* *3415: Right hand operand of '&&' or '||' is an expression with possible side effects.* *3416: Logical operation performed on expression with possible side effects.* *3440: Using the value resulting from a ++ or -- operation.* *4115: Operand of logical && or || operator is not an 'effectively Boolean' expression.*
11215	F	*0277: [I] An integer constant expression with negative value is being converted to an unsigned type.* *3306: Implicit unsigned conversion on positive integer constant expression will not preserve value.* Incorrect instances of messages 0277 and 3306 have been silenced in compound assignment operations
11368	F	The result of right shift operations was sometimes failing to respect the setting of the –arithrsh option in the evaluation of integer constant expressions.
11418	F	*0277: An integer constant expression with negative value is being converted to an unsigned type.* Message 0277 was not generated when assigning a negative value to an unsigned bit-field

CR	T	Resolution
11447	F	*4240: The value of an 'effectively Boolean' expression is being assigned to an object of type 'char'.* *4241: The value of an 'effectively Boolean' expression is being passed to a function parameter of type 'char'.* *4242: The value of an 'effectively Boolean' expression is being returned from a function with a return type of 'char'.* *4243: The value of an 'effectively Boolean' expression is being assigned to an object of floating type.* *4244: The value of an 'effectively Boolean' expression is being passed to a function parameter of floating type.* *4245: The value of an 'effectively Boolean' expression is being returned from a function with a floating return type.* 1. New messages are generated when the value of an effectively Boolean expression is assigned to an object of floating type or type char. 2. Implicit conversion messages (e.g. int to unsigned char) are no longer generated when assigning the value of an effectively Boolean expression to an object of signed or unsigned integer type.
11547	N	New messages have been implemented to identify when a source file doesn't end with a newline character. *0914: [U] Source file does not end with a newline character.* *0915: [U] Source file ends with a backslash character followed by a newline.*
11955	F	The 'pc', 'ansipc' and 'all' language extension options no longer result in mapping of filenames in #include directives to lower case.
11982	F	The size of an array with size determined from an initializer consisting of an empty string was incorrectly computed as "2" rather than "1".
11984	F	1271: Using a non-int expression does not alter the type of the enum constant. Message 1271 was not generated when declaring an enum constant with unsigned type.
12323	F	A syntax error message was located in a misleading location.
12422	F	*3680: [U] Indexing array with constant value that is out of bounds.* Message 3680 was not generated correctly for arrays with more than one dimension if the computed array index remained within the bounds of the combined dimensions.
12516	F	*3122: Hard-coded 'magic' string literal, %s.* Message 3122 is no longer generated for an empty string literal.
12560	F	*0274: [I] Conversion of integer constant expression to a signed integer type which cannot represent the value.* *0277: Conversion of a constant negative value to an unsigned type.* *0280: Use of constant negative value which assumes a two's complement representation of signed values.* *3306: Truncation of positive constant integer value during implicit conversion to a smaller unsigned type.* Incorrect messages were generated when assigning out of range values to unsigned bit-fields or bit-fields defined as type *int*. The bit-field was incorrectly assumed to be of signed type – regardless of the setting of the –bitsigned option.
12660	F	The calculation of the STPTH metric has been modified in a few situations particularly in relation to the way in which null statements are interpreted – for example in switch statements.
12721	F	Messages were not generated to identify evaluation order problems when a volatile was accessed more than once between sequence points.
12775	N	New messages have been implemented to identify implicit conversions to and from void pointers. *0314: [I] Cast from a pointer to object type to a pointer to void.* *0315: [I] Implicit conversion from a pointer to object type to a pointer to void.* *0316: [I] Cast from a pointer to void to a pointer to object type.* *0317: [I] Implicit conversion from a pointer to void to a pointer to object type.*

CR	T	Resolution
12803	N	*0838: File '%1s' has already been included directly from within file '%2s'.* *0839: File '%1s' has already been included indirectly from within file '%2s'.* Messages 0838 and 0839 are now parameterized in order to identify which included files are being referred to.
12815	F	*3673: The object addressed by the pointer parameter '%s' is not modified and so the pointer could be of type 'pointer to const'.* Message 3673 was sometimes generated incorrectly.
12871	F	The value of a character constant was misinterpreted when 'char' was configured as a 'signed' type.
12898	F	*1335: Parameter identifiers missing in function prototype declaration.* *1336: Parameter identifiers missing in declaration of a function type.* False positive instances of messages 1335 and 1336 are no longer generated when the parameter list in a function definition or declaration consists of a typedef representing 'void'.
13081	N	Message 3122 now displays the contents of the string literal to which it refers.
13092	F	*0693: Struct initializer is missing the optional {.* *0694: Array initializer is missing the optional {.* Noisy instances of message 0693 have been eliminated in situations when the initializer for a struct or array has been specified with no braces – and is therefore identified with message 0672.
13267	F	*0694: Array initializer is missing the optional {.* Noisy instances of message 0694 have been eliminated in situations when the initializer for a struct has been specified with no braces – and is therefore identified with message 0672.
13269	F	*0672: [U] The initializer for a 'struct', 'union' or array is not enclosed in braces.* Message 0672 was not generated when initializing a struct with an initializer consisting of an unbraced '0'.
13324	N	New messages have been implemented to identify tentative definitions. *2021: This tentative definition is interpreted as a declaration. Is this intended ?* *2022: This tentative definition is interpreted as a definition. Would it be appropriate to include an explicit initializer ?*
13370	F	*0280: Use of constant negative value which assumes a two's complement representation of signed values.* *3306: Truncation of positive constant integer value during implicit conversion to a smaller unsigned type.* Messages 0280 and 3306 were generated incorrectly when assigning a negative value to an object of unsigned type.
13371	F	*0277: Conversion of a constant negative value to an unsigned type.* Message 0277 was not generated when assigning a negative value to an unsigned bit-field
13465	N	*3260: Typedef defined with more than 2 levels of indirection* *3261: Member of struct/union defined with more than 2 levels of indirection.* *3262: Object defined or declared with more than 2 levels of indirection.* *3263: Function defined or declared with a return type which has more than 2 levels of indirection.* New messages have been implemented to identify the use of more than 2 levels of pointer indirection.
13466	N	The behaviour of the –o and –n options has been modified. If a message is suppressed by these options, no diagnostics will be output to the .err file during analysis and none will be visible at view time when using utilities such as the message browser, even if the message is reenabled.
13478	F	*3680: [U] Access outside bounds of array using a constant array subscript.* Message 3680 was generated incorrectly when using array subscript syntax to access the null byte terminating a string literal.
13502	F	In a very few cases, the value of metrics with non-integer values may be displayed differently in the 2nd decimal place as a result of a modified rounding algorithm.
13515	N	The symbolic type code generated for type (plain) 'char' in the .met file is now '_c' rather than 'sc' or 'uc'.

CR	T	Resolution
13542	F	GUI bug when entering '-file' into custom report configuration parameters area.
13599	C	Edit menu item 'Collapse Folders' only enabled for non-leaf folders.
13560	F	The file based metrics STPRT and STMOB are no longer supported because the theoretical basis on which they were computed is no longer considered as sufficiently rational.
13564	F	Integral promotion messages are no longer generated when using the logical negation operator '!'.
13580	F	Messages which identify a cast operator were incorrectly located on the first character of the type specifier of the cast rather than the left-hand parenthesis which introduces the cast.
13581	F	Message which identify a division operator were incorrectly located one character to the right of the '/' character.
13588	F	The use of very long pathnames could sometimes result in program failure.
13602	F	Incorrect symbolic type codes were sometimes generated in the met file.
13607	C	All cores of a multi-core system are enabled for file analysis operations in the Windows GUI.
13642	F	*3345: Statement contains more than one access to objects that are volatile.* Message 3345 was issued incorrectly when accessing structs and unions.
13647	F	Correction to a rare out-of-bounds error on the 'Find Listings' functionality in Relationships, Metrics and Structure diagrams.
13656	F	*3120: Hard-coded 'magic' integer constant '%s'.* *3121: Hard-coded 'magic' floating constant '%s'.* *3122: Hard-coded 'magic' string literal, "%s".* *3123: Hard coded 'magic' character constant '%s'.* Magic constant messages were generated incorrectly when initialising 'const' qualified arrays.
13683	N	*2711: 'register' is only a hint - the compiler may well be able to do a better job of optimising - avoid!* Message 2711 has been renumbered as message 2011. This change was implemented in order to release the old message number for future use within the range of dataflow analysis messages.
13686	N	*3325: This controlling expression has a constant 'false' value.* Message 3325 has been split into 2 messages. In future message 3325 is generated for controlling expressions in 'while' and 'for' statements. Message 3329 is now generated for controlling expressions in 'if' statements. *3325: This 'while' or 'for' loop controlling expression is a constant expression and its value is 'false'. The loop will not be entered.* *3329: This 'if' controlling expression is a constant expression and its value is 'false'.*
13690	N	*3682: Index may take a value greater than number of elements.* Message 3682 has previously been classified as 'obsolete' and has now been removed. Its functionality has been superseded by message 2841. *3683: Use of constant address which points to one element beyond the end of the array.* *3688: Definite use of address which points to one element beyond the end of the array.* *3692: Apparent use of address which points to one element beyond the end of the array.* These messages identified the computation of an address just one element beyond the end of an array – an operation which is quite permissible so long as the address is not dereferenced. The messages have now been removed (to avoid confusion) because their function has been effectively superseded by new dataflow messages. Messages 2840-2842 identify invalid dereferencing operations – including a dereference of the element just one beyond the end of an array. Messages 2930-2933 identify computation of an invalid address – i.e. an address anywhere outside the bounds of an array object (including the element just one beyond the end of the array).
13691	F	Some illegal code constructs caused analysis failure when a preprocessing directive was encountered within the list of arguments to a macro.
13694	F	Wrong location was being found for identifier declaration in source view from Windows GUI relationships diagram.

CR	T	Resolution
13699	F	*0547: [C] This declaration of tag '%s' conflicts with a previous declaration.* Message 547 was generated incorrectly when a struct or union tag was redeclared following an earlier definition of the same tag.
13715	F	*0679: Redundant braces found in initializer.* Message 0679 has been removed, because its implementation was found to be unreliable and potentially confusing.
13746	C	Removal of deprecated and incompatible Warning Listing report component (prjdsp.exe).
13747	F	*2547: This declaration of tag '%s' hides a more global declaration.* Message 2547 was not generated when a struct or union tag was defined at an inner scope following an earlier declaration of the same tag at an outer scope.
13761	C	The full cause for Reprise license failures is presented in any failure dialog.
13779	F	#pragma PRQA_NO_RETURN failed to function correctly when encountering a call to a function through a function pointer preceded by the indirection operator ('*').
13783	N	A new screen presents Dataflow settings, and controls their saving into Project Configuration Files as well as application default settings.
13786	N	*0274: [I] Conversion of integer constant expression to a signed integer type which cannot represent the value.* *0277: Conversion of a constant negative value to an unsigned type.* *0280: Use of constant negative value which assumes a two's complement representation of signed values.* *3306: Truncation of positive constant integer value during implicit conversion to a smaller unsigned type.* 1. Messages 0274, 0277 and 0280 were previously generated for all type conversions, whether implicit or explicit (cast). The functionality of each of these messages has been split. They are now generated only for implicit conversions; explicit conversions are identified with 3 new messages, 0974, 0977 and 0280. 2. Message 3306 continues to be generated only for implicit conversions; however a new message 3290 has been implemented which identifies the corresponding situation in an explicit conversion. The new message set is as follows: 0274: [I] Implicit conversion of integer constant expression to a signed integer type which cannot represent the value. 0277: Implicit conversion of a constant negative value to an unsigned type. 0280: Implicit conversion of constant negative value which assumes a two's complement representation of signed values. 3306: Truncation of positive constant integer value during implicit conversion to a smaller unsigned type. 0974: [I] Cast of integer constant expression to a signed integer type which cannot represent the value. 0977: Cast of a constant negative value to an unsigned type. 0980: Cast of constant negative value which assumes a two's complement representation of signed values. 3290: Truncation of positive constant integer value during cast to a smaller unsigned type. All these messages are, however, classified as obsolete. Their functionality is superseded by the new dataflow messages 2850,2890,2900,2940,2855,2895,2905 and 2945.
13832	C	The timeout function to release the GUI license for system idleness is now made configurable, rather than being a fixed 30 minutes.
13884		Analysis dependency check (for determining whether to re-analyse) now includes date-based check on all configuration files (personalities and project configuration file) and baseline suppression file.