QA C 7.2 Release Notes

November 2009

Contents

1. OVERVIEW

The principal features of QAC 7.2 include the following:

1.1. Support for Baselining

The GUI interface now has baseline operation built-in to the analysis operation. Once a project is analysed, a menu item will create a baseline from that version of source. For all further code changes or a different version of that project, this baseline can be automatically applied and the recurring diagnostics marked as baseline-suppressed. It will work either against a version of code or with a Version Control System. Contact Programming Research for details of our VCS support. This feature requires a specific licence.

1.2. Windows GUI Improvements

There have been a number of significant GUI improvements in this release.

- Dependency-based analysis is now an application option, where files are re-analysed only if any of their analysis set or configuration personalities were altered.
- CMA analyses now take additional project-specific parameters, so as to generate unique output names in project paths.
- Application and Project-based environment variables can now be defined and used in any custom reports or secondary or CMA analysis events.
- Add-on Reports and Structure101 are now integrated into the GUI menu structure when installed.

1.3. Message Browser Improvements

Baseline operation is built into the Message Browser, with the specific additional capability to review a generated baseline and alter its contents, in a new “Baseline Administration Mode”.

1.4. Help System Improvements

Extensive improvements have been made to the message help system and the text of some messages has been revised to improve clarity.

1.5. MISRA-C:2004

Additional messages have been implemented which provide improved enforcement of the following rules:

Rule 12.6

Rule 17.4

Rule 20.1

Rule 20.2

1.6. Analysis Highlights

@address:bitno

Parsing support has been added to allow the "@address:bitno" syntax to be used in implementations which support objects of 'bit' type.

Anonymous Unions

Anonymous structs and unions (unnamed objects of unnamed struct or union type) are now supported as defined in certain implementations of the C language.

Microsoft integer literal suffixes

The suffixes I64 and UI64 appended to an integer constant are now considered synonymous with LL and ULL.

Inline assembler syntax

Additional support has been provided for inline assembler code specified within a single preprocessing directive of the form *#asm (" … ")*

2. Message Changes

2.1. New messages

Msg	Message Text	CR
0491	*Array subscripting applied to an object of pointer type.*	11316
0492	*Array subscripting applied to a function parameter declared as a pointer.*	11316
0619	*[C] The identifier '%s' has already been defined in the current scope within the ordinary identifier namespace.*	11064
0822	*[Q] Cannot find forceinclude file '%s'.*	12304
1042	*[E] Using I64 or UI64 as an integer constant suffix. This is a language extension.*	12213
1043	*[E] Defining an anonymous union object. This is a language extension.*	11064
1044	*[E] Defining an anonymous struct object. This is a language extension.*	11064
3377	*Operand of a logical && or || operator is a constant expression which is not a 'Boolean' value.*	12402
3659	*Unnamed zero-width bit-field declared with a signed type.*	12710
4115	*Operand of logical && or || operator is not an effectively Boolean expression.*	12402
4116	*Operand of logical ! operator is not an effectively Boolean expression.*	12402
4600	*The macro '%1s' is also defined in '<%2s>'.*	13289
4601	*The macro '%1s' is the name of an identifier in '<%2s>'.*	13289
4602	*The identifier '%1s' is declared as a macro in '<%2s>'.*	13289
4603	*The object/function '%1s'is being defined with the same name as an ordinary identifier defined in '<%2s>'.*	13289
4604	*The object/function '%1s' is being declared with the same name as an ordinary identifier defined in '<%2s>'.*	13289
4605	*The typedef '%1s' is also defined in '<%2s>'.*	13289
4606	*The typedef '%1s' has the same name as another ordinary identifier in '<%2s>'.*	13289
4607	*The enum constant '%1s' has the same name as another ordinary identifier in '<%2s>'.*	13289
4608	*The tag '%1s' is also defined in '<%2s>'.*	13289
4620	*The macro '%1s' may also be defined as a macro in '<%2s>'.*	13290
4621	*The macro '%1s' may also be defined as a typedef in '<%2s>'.*	13290
4622	*The identifier '%1s' may be defined as a macro in '<%2s>'.*	13290
4623	*The typedef '%1s' may also be defined in '<%2s>'.*	13290
4624	*The ordinary identifier '%1s' may be defined as a typedef in '<%2s>'.*	13290
4640	*The macro '%1s' could conflict in the future with the name of a macro in '<%2s>'.*	13291
4641	*The identifier '%1s' could conflict in the future with the name of a macro in '<%2s>'.*	13291
4642	*The macro '%1s' could conflict in the future with the name of a function in '<%2s>'.*	13291
4643	*The identifier '%1s' could conflict in the future with the name of a function in '<%2s>'.*	13291
4644	*The macro '%1s' could conflict in the future with the name of a typedef in '<%2s>'.*	13291
4645	*The identifier '%1s' could conflict in the future with the name of a typedef in '<%2s>'.*	13291

2.2. Messages with modified behaviour

The following table summarises the messages whose behaviour has been modified in QAC 7.2.

3 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
0310	*Casting to different object pointer type.*	F	12379
0338	*[C] Octal or hex escape sequence value is too large for 'unsigned char' or 'wchar_t' type.*	C	12792
0431	*[C] Function argument points to a more heavily qualified type.*	F	13144
0562	*[C] Right operand of assignment points to a more heavily qualified type.*	F	13144
0602	*[U] The identifier '%s' is reserved for use by the library.*	C	11241
0602	*[U] The identifier '%s' is reserved for use by the library.*	F	12781
0660	*[E] Defining an unnamed member in a struct or union.*	F	13129
0673	*[C] Initializer points to a more heavily qualified type.*	F	13144
0685	*[C] Initializer for any object with static storage duration must be a constant expression.*	F	12435
0757	*[C] 'return' expression points to a more heavily qualified type.*	F	13144
0778	*[L] Identifier matches other identifier(s) (e.g. '%s') in first 31 characters - program does not conform strictly to ISO:C90.*	F	12759
0779	*[U] Identifier does not differ from other identifier(s) (e.g. '%s') within the specified number of significant characters.*	F	12759
0823	*[S] Unexpected '#else' or '#elif' directive follows '#else'.*	F	12282
0826	*[S] Unexpected ':' found without a preceding '?' in a '#if' or '#elif' expression.*	F	11973
0827	*[S] Missing ':' after '?' in a '#if' or '#elif' expression.*	F	11973
0850	*[C99] Macro argument is empty.*	F	12354
0877	*[C] '#if' and '#elif' expressions may contain only integral constants.*	F	11973
0879	*[S] Illegal operator in '#if' or '#elif' expression.*	F	11973
0886	*[S] Missing or invalid expression in '#if' or '#elif' directive.*	F	11973
0896	*[S] Missing operand in '#if' or '#elif' expression.*	F	11973
0897	*[S] Missing operator in '#if' or '#elif' expression.*	F	11973
0898	*[S] Unexpected ')' or ':' in '#if' or '#elif' expression.*	F	11973
0899	*[S] Preprocessing directive appears in the middle of a line.*	F	13249
1012	*[E] Use of a C++ reference type ('type &') will be treated as a language extension..*	F	12850
1028	*[E] Use of the sizeof operator in a preprocessing directive is a language extension.*	F	11973
1304	*Old style definition of function '%s()' is not portable to C++.*	F	11477
2003	*The preceding 'switch' clause is not empty and does not end with a 'jump' statement. Execution will fall through.*	F	12719
2003	*The preceding 'switch' clause is not empty and does not end with a 'jump' statement. Execution will fall through.*	F	12790
2020	*Final 'switch' clause does not end with an explicit 'jump' statement.*	F	12789
2467	*Loop control variable, %s, is not modified inside loop.*	F	12197
2469	*Loop control variable in this 'for' statement, %s, is modified in the body of the loop.*	F	12197
3001	*Function has been declared with an empty parameter list.*	F	11477
3002	*Defining '%s()' with an identifier list and separate parameter declarations is an obsolescent feature.*	F	11477
3007	*"void" has been omitted when defining a function with no parameters.*	F	12829
3230	*Address of automatic object assigned to local pointer with static storage duration.*	F	11666
3314	*This controlling expression is an assignment.*	C	13252
3314	*This controlling expression is an assignment.*	C	13254
3322	*Operand of a logical ! operator is a constant expression which is not a 'Boolean' value.*	C	12402
3332	*The macro '%s' used in this '#if' or '#elif' expression is not defined.*	F	11973

3332	*The macro '%s' used in this '#if' or '#elif' expression is not defined.*	F	12544
3333	*A 'break' statement has been used in the middle of a 'switch' 'case'/'default' clause.*	F	12094
3392	*A shift, relational or equality operator has been used more than once. Extra parentheses recommended.*	F	12414
3394	*Binary operator other than + - * / % used with different binary operator of the same precedence. Extra parentheses recommended.*	F	12414
3396	*Extra parentheses recommended. A binary operation is the operand of a conditional operator.*	F	12414
3396	*Extra parentheses recommended. A binary operation is the operand of a conditional operator.*	F	12663
3397	*Binary operator other than + - * / % used with binary operator of different precedence. Extra parentheses recommended.*	F	12414
3399	*Extra parentheses recommended. A unary operation is the operand of a logical && or ||.*	F	12414
3400	*Binary/ternary operator used with && or ||. Extra parentheses recommended.*	F	11855
3401	*Possible precedence confusion: extra parentheses are recommended here.*	F	12414
3406	*Object/function '%s', with external linkage, has been defined in a header file.*	F	12830
3410	*Macro parameter not enclosed in ().*	C	11189
3480	*Object/function '%s', with internal linkage, has been defined in a header file.*	F	12830
3700-3881	*<Implicit conversion messages>*	F	11285 12029
3900-4081	*<Function return implicit conversion messages>*	F	11285 12029

2.3. Messages relocated to a different message group

Some existing messages have been relocated to a different message group.

Msg	Old Group	New Group	Message Text
0271	ISO_ImplDef	ISO_ExpU	*[U] Left shift operation on constant signed expression generating an undefined value.*
0277	ISO_ImplDef	Maj_Ops	*Conversion of a constant negative value to an unsigned type.*
0294	ISO_ImplDef	ISO_ExpU	*[U] Definite signed left shift operation generating an undefined value.*
0295	ISO_ImplDef	ISO_ExpU	*[U] Apparent signed left shift operation generating an undefined value.*
0434	Constraint	Syntax	*[S] The identifier '%s' has not been declared.*
0473	Constraint	Config	*[Q] Result of 'sizeof' operation will not fit in configured type for 'size_t'.*
0541	ISO_ImpU	Constraint	*[C] Argument no. %s does not have object type.*
0547	ISO_ExpU	Constraint	*[C] This declaration of tag '%s' conflicts with a previous declaration.*
0581	Constraint	ISO_ImplDef	*[I] Floating-point constant may be too small to be representable.*
0617	Constraint	C99_ext	*[C99] 'const' qualifier has been duplicated.*
0618	Constraint	C99_ext	*[C99] 'volatile' qualifier has been duplicated.*
0626	ISO_ImpU	ISO_ExpU	*[U] '%s' has different type to previous declaration (which is no longer in scope).*
0652	ISO_ImpU	Min_Func	*Identifiers have been provided for some but not all of the parameters in a function prototype.*
0660	Min_Array	Lang_ext	*[E] Defining an unnamed member in a struct or union. This is a language extension.*
0675	ISO_ExpU	Constraint	*[C] Initializer is not of compatible 'struct'/'union' type.*
0688	Min_Array	Maj_Decl	*Array size determined by number of initializers which include concatenated string literals.*
0689	Maj_Redun	ISO_ImpU	*[u] 'Switch' statement will bypass the initialization of this local variable.*
0706	ISO_ImpU	ISO_ExpU	*[U] Label '%s' is not unique within this function.*
0747	ISO_ImpU	Constraint	*[C] 'return exp;' found in '%s()' whose return type is qualified 'void'.*
0832	Maj_Prepro	Min_Prepro	*Macro substitution in #include preprocessing directive.*
0861	Min_Prepro	Maj_Redun	*This #include <%s> directive is redundant.*
0862	Min_Prepro	Maj_Redun	*This #include "%s" directive is redundant.*
0899	Constraint	Lang_ext	*[E] Unrecognized preprocessing directive has been ignored - assumed to be a language extension.*
1012	Syntax	Lang_ext	*[E] Use of a C++ reference type ('type &') will be treated as a language extension.*
1310	Min_Cpp	Min_Ident	*'%s' is used as a tag and a typedef for the same 'struct' / 'union' / 'enum'.*
1311	Min_Cpp	Obsolete	*'void *' and 'const T *' pointers used as operands to an equality or conditional operator.*
1313	Min_Cpp	Obsolete	*Executing 'goto %s' will cause local initialization to be skipped.*
1318	Min_Enum	Min_Cpp	*Object of enum type is being modified with a compound assignment operator.*
1319	Min_Enum	Min_Cpp	*Object of enum type is being modified with an increment or decrement operator.*
1460	Min_Switch	Maj_Enum	*'Switch' label value, %s, not contained in enum type.*
1461	Min_Enum	Maj_Enum	*Value of constant expression is not in the enum type to which it is being converted, but is bitwise OR of constants in the enum type.*
1474	Min_Enum	Maj_Enum	*Object of enum type is being modified with a bitwise compound assignment operator.*
1477	Min_Enum	Obsolete	*Object of enum type is being implicitly compared against zero in a controlling expression.*
1479	Min_Enum	Maj_Enum	*Object of enum type is being modified with an arithmetic compound assignment operator.*

1481	Min_Enum	Maj_Enum	*Object of enum type is being modified with an increment or decrement operator.*
1482	Min_Enum	Maj_Enum	*Non-constant expression cast to enum type.*
2005	Min_Ctrl	Obsolete	*A 'continue' statement has been used.*
3104	Config	Syntax	*[S] #pragma '%s' has invalid arguments and has been ignored.*
3105	Config	Syntax	*[S] A #pragma block has not been closed with a #pragma '%s'.*
3107	Config	Syntax	*[S] A #pragma '%s' has been found without a matching #pragma block start directive.*
3313	Maj_Decl	Min_Decl	*No definition has been found for structure/union tag '%s'.*
3371	Maj_Ops	Min_Ops	*Definite truncation of bits in an unsigned left shift operation.*
3381	Maj_Ops	Min_Ops	*Apparent truncation of bits in an unsigned left shift operation.*
3419	Min_Ctrl	Maj_Redun	*Initialization expression of 'for' statement has no side effects.*
3420	Min_Ctrl	Maj_Redun	*Increment expression of 'for' statement has no side effects.*
3424	Min_Ops	Maj_Redun	*Statement contains a redundant & or | at top level.*
3435	Min_Prepro	Obsolete	*Parameter '%s' occurs more than once in the replacement list of this macro.*
3623	Min_Func	Min_Array	*Passing a struct/union by value as a function argument.*
3624	Min_Func	Min_Array	*Function returns a struct/union by value.*

2.4. Messages relocated to a different message level

Some existing messages have been relocated to a different message level.

Msg	Old Level	New Level	Message Text
0271	6	7	*[U] Left shift operation on constant signed expression generating an undefined value.*
0277	6	3	*Conversion of a constant negative value to an unsigned type.*
0294	6	7	*[U] Definite signed left shift operation generating an undefined value.*
0295	6	7	*[U] Apparent signed left shift operation generating an undefined value.*
0434	8	9	*[S] The identifier '%s' has not been declared.*
0473	8	9	*[Q] Result of 'sizeof' operation will not fit in configured type for 'size_t'.*
0541	7	8	*[C] Argument no. %s does not have object type.*
0547	7	8	*[C] This declaration of tag '%s' conflicts with a previous declaration.*
0581	8	6	*[I] Floating-point constant may be too small to be representable.*
0617	8	6	*[C99] 'const' qualifier has been duplicated.*
0618	8	6	*[C99] 'volatile' qualifier has been duplicated.*
0652	7	2	*Identifiers have been provided for some but not all of the parameters in a function prototype.*
0660	2	6	*[E] Defining an unnamed member in a struct or union. This is a language extension.*
0675	7	8	*[C] Initializer is not of compatible 'struct'/'union' type.*
0688	2	3	*Array size determined by number of initializers which include concatenated string literals.*
0689	3	7	*[u] 'Switch' statement will bypass the initialization of this local variable.*
0747	7	8	*[C] 'return exp;' found in '%s()' whose return type is qualified 'void'.*
0832	3	2	*Macro substitution in #include preprocessing directive.*
0861	2	3	*This #include <%s> directive is redundant.*
0862	2	3	*This #include "%s" directive is redundant.*
0899	8	6	*[E] Unrecognized preprocessing directive has been ignored - assumed to be a language extension.*
1012	9	6	*[E] Use of a C++ reference type ('type &') will be treated as a language extension.*
1460	2	3	*'Switch' label value, %s, not contained in enum type.*
1461	2	3	*Value of constant expression is not in the enum type to which it is being converted, but is bitwise OR of constants in the enum type.*
1474	2	3	*Object of enum type is being modified with a bitwise compound assignment operator.*
1479	2	3	*Object of enum type is being modified with an arithmetic compound assignment operator.*
1481	2	3	*Object of enum type is being modified with an increment or decrement operator.*
1482	2	3	*Non-constant expression cast to enum type.*
3313	3	2	*No definition has been found for structure/union tag '%s'.*
3371	3	2	*Definite truncation of bits in an unsigned left shift operation.*
3381	3	2	*Apparent truncation of bits in an unsigned left shift operation.*
3419	2	3	*Initialization expression of 'for' statement has no side effects.*
3420	2	3	*Increment expression of 'for' statement has no side effects.*
3424	2	3	*Statement contains a redundant & or | at top level.*

3. CR Summary

The following table lists 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
11064	N	Parsing support has been added for anonymous structs and anonymous unions.
11189	C	*3410: Macro parameter not enclosed in ().* Message 3410 is no longer generated in instances where the macro parameter is the right hand operand of a struct/union member operator.
11241	C	*0602: [U] The identifier '%s' is reserved for use by the library.* Message 0602 is now generated for tag identifiers. See also CR 12781
11285	F	Implicit conversion messages were incorrect or missing following the use of a conditional operator (? :).
11316	N	New messages have been implemented to identify use of the array subscript operator on an operand declared as a pointer. *0491: Array subscripting applied to an object of pointer type rather than array type.* *0492: Array subscripting applied to a function parameter declared as a pointer.*
11477	F	*1304: Old style definition of function '%s()' is not portable to C++.* *3001: Function has been declared with an empty parameter list.* *3002: Defining '%s()' with an identifier list and separate parameter declarations is an obsolescent feature.* Messages 1304, 3001 & 3002 were sometimes not issued as expected when a function was declared inconsistently in both K&R style and function prototype style.
11666	F	*3230: Address of automatic object assigned to local pointer with static storage duration.* Message 3230 was generated incorrectly in certain contexts.
11855	F	*3400: Extra parentheses recommended. A binary operation is the operand of a logical && or ||.* Message 3400 was generated incorrectly when a && or || operator was the operand of a conditional operator (? :).
11911	F	The symbolic type code in <DEFINE> records in the .met file sometimes referred incorrectly to a typedef name rather than the struct tag to which the typedef referred.
11973	F	A number of messages were sometimes not generated when expected or else generated when not required for expressions in #elif directives. *0826: [S] Unexpected ':' found without a preceding '?' in a '#if' or '#elif' expression.* *0827: [S] Missing ':' after '?' in a '#if' or '#elif' expression.* *0877: [C] '#if' and '#elif' expressions may contain only integral constants.* *0879: [S] Illegal operator in '#if' or '#elif' expression.* *0886: [S] Missing or invalid expression in '#if' or '#elif' directive.* *0896: [S] Missing operand in '#if' or '#elif' expression.* *0897: [S] Missing operator in '#if' or '#elif' expression.* *0898: [S] Unexpected ')' or ':' in '#if' or '#elif' expression.* *1028: [E] Use of the sizeof operator in a preprocessing directive is a language extension.* *3332: The macro '%s' used in this '#if' or '#elif' expression is not defined.*
11979	F	Implicit conversion messages were incorrect or missing following the use of a conditional operator (? :).

CR	T	Resolution
12029	F	Implicit conversion messages were incorrect or missing following the use of a conditional operator (? :).
12094	F	*:3333 A 'break' statement has been used in the middle of a 'switch' 'case'/'default' clause.* Message 3333 was not always generated when expected in an "else" clause within a switch statement.
12118	F	In Windows GUI, file-based metrics browser was incorrectly stripping the filename when a left-bracket was encountered in the path.
12197	F	*2467: Loop control variable, %s, is not modified inside loop.* *2469: Loop control variable in this 'for' statement, %s, is modified in the body of the loop.* Messages 2467 and 2469 were issued inconsistently in nested loops.
12213	N	Support has been added for two additional non-standard integer suffixes, "I64" and "UI64". These are treated as equivalent to "LL" and "ULL" respectively. A new message is generated whenever these suffixes are encountered: *1042: [E] Using I64 or UI64 as an integer constant suffix. This is a language extension.*
12282	F	*0823: [S] Unexpected '#else' or '#elif' directive follows '#else'.* Message 0823 was not generated on an unexpected #else or #elif directive within a nested #if construct.
12304	N	A new message is now generated to indicate when a forceinclude file is not found. Message 0822 is a Level 9 configuration error and so QAC now returns a non-zero status value. *0822: [Q] Cannot find forceinclude file '%s'.*
12354	F	*0850: [C99] Macro argument is empty.* Message 0850 was sometimes generated incorrectly when performing macro substitution with a macro defined with an empty replacement list.
12364	F	Unix GUI now handles filenames containing a $ character correctly.
12379	F	*0310: Casting to different object pointer type.* Message 0310 was sometimes generated incorrectly when casting an object pointer declared with a storage class specifier.
12381	F	Windows GUI Metric Browser now properly displays all file-static functions with the same name.
12402	N	New messages have been introduced to identify operands to the logical &&, and || operators which are not explicitly Boolean in nature. *4115: Operand of logical && or || operator is not an effectively Boolean expression.* *4116: Operand of logical ! operator is not an effectively Boolean expression.* A new message has been introduced to identify an operand to a logical &&,or || operator which is a constant expression but is not of integral type or is a value other than 0 or 1. *3377: Operand of a logical && or || operator is a constant expression which is not a 'Boolean' value.* The specification of existing message 3322 has been modified slightly. Previously this message was generated for any operand which was a constant expression but was not an integer constant with value 0 or 1. The message is no longer generated providing the operand is an integral constant expression of value 0 or 1. *3322: Operand of a logical ! operator is a constant expression which is not a 'Boolean' value.*
12414	F	*3396: Extra parentheses recommended. A binary operation is the operand of a conditional operator.* *3399: Extra parentheses recommended. A unary operation is the operand of a logical && or ||.* *3401: Possible precedence confusion: extra parentheses are recommended here.* *M*essages which recommend additional parentheses in expressions, were not always generated correctly when operands were subject to integral promotion.
12435	F	*0685: [C] Initializer for any object with static storage duration must be a constant expression.* Message 0685 was generated incorrectly when casting an expression of array type or function type to an integer type.

CR	T	Resolution
12512	F	The scope flag used in DEFINE records with the .met file was sometimes incorrect: 1. enumeration constants defined at file scope were flagged as having block scope instead of file scope 2. struct/union/enum tags defined within function prototype declarations were flagged as having block scope instead of prototype scope 3. struct/union members and enum constants defined within function prototype declarations were flagged as having block scope instead of prototype scope
12538	F	The value resulting from casting a constant expression to a floating type was sometimes computed incorrectly resulting in generation of incorrect messages.
12544	F	*3332: The macro '%s' used in this '#if' or '#elif' expression is not defined.* Message 3332 was generated incorrectly on the right hand operand of a logical && or || operator in a pre-processing expression. The right hand operand of these operands should not be evaluated if the result can be determined from the left hand operand.
12550	F	Recoverable dataflow analysis problem arising when expressions of different types were operands to a division or multiplication operation in a controlling expression.
12585	F	Recoverable dataflow analysis problem arising when the control variable in a loop was a descending rather than ascending value.
12624	F	Unix GUI now permits deletion of project output files.
12663	F	*3396: Extra parentheses recommended. A binary operation is the operand of a conditional operator.* Message 3396 was generated incorrectly when a postfix increment or decrement operator was applied to the operand of a conditional operator and not then parenthesized.
12668	F	Unix GUI metric browser has corrected behaviour and operation of its sort-by menus.
12669	F	Unix GUI now shows a complete function structure diagram for very large files.
12673	C	Unix GUI will recreate an output path structure when this has been deleted.
12674	C	Unix GUI now checks analysis fail status of each file (level 9 errors) on project load.
12675	N	Unix GUI supports dependency-based analysis on source and header files.
12676	F	Unix GUI had a problem of inconsistent file selection on new/existing projects.
12677	F	Unix GUI had some incorect greying of menu items on file selection.
12678	F	Unix GUI has improved relationship diagram display routines for larger projects.
12692	N	Parsing support has been added for an additional form of inline assembler – directives of the form: #asm (" … ") (No corresponding #endasm is expected)
12694	N	Parsing support has been added for the syntax @address:n – sometimes used to specify the location of a bit-field in environments where bit types are supported.
12710	N	A new message has been implemented to identify an unnamed zero-width bit-field of zero length declared with signed type. (Bit-fields used for padding are required to be unsigned in MISRA Rule 6.5). *3659: Unnamed zero-width bit-field declared with a signed type.*
12719	F	*2003: The preceding 'switch' clause is not empty and does not end with a 'jump' statement. Execution will fall through.* Message 2003 was generated incorrectly when the preceding 'switch' clause was terminated with a 'continue' or 'goto' statement.
12728	F	Unix GUI now allows custom report names to contain spaces.
12732	F	Unix GUI has corrected file analysis status for externally deleted output.
12751	F	In Windows GUI Personality Listing, a warning is now issued before deleting the default personality.
12752	F	In Windows GUI, Secondary Analysis window was not refreshing & updating properly.

CR	T	Resolution
12759	F	*0778: [L] Identifier matches other identifier(s) (e.g. '%s') in first 31 characters - program does not conform strictly to ISO:C90.* *0779: [U] Identifier does not differ from other identifier(s) (e.g. '%s') within the specified number of significant characters.* Message 0778 was not generated if 2 identifiers were different but identical in first 31 characters and one identifier was exactly 31 characters in length. Message 0779 should be generated when 2 identifiers are different but identical within the first N characters where N is specified by the –namelength option. However the message was not generated when one of the identifiers was exactly N characters in length.
12768	N	The text of many messages has been revised in order to improve clarity. The references and help text associated with all messages have been improved and extended.
12769	C	In Windows GUI, user-control permits disable of multi-core analysis in order to revert to single sequence file analysis.
12774	F	Documentation corrections in User Guide.
12778	F	The incidence of noisy messages has been reduced when parsing a preprocessing directive with incorrect usage of the "defined" operator
12781	F	*0602: [U] The identifier '%s' is reserved for use by the library.* Message 0602 was not generated for some identifiers, e.g.: a) macros b) enum constants declared at file scope c) function parameters See also CR 11241
12789	F	*2020: Final 'switch' clause does not end with an explicit 'jump' statement.* Message 2020 was generated incorrectly when the preceding case clause was terminated with a 'continue', 'goto' or 'return' statement.
12790	F	*2003: The preceding 'switch' clause is not empty and does not end with a 'jump' statement. Execution will fall through.* Message 2003 was not always generated when fall through did not occur on every path.
12792		Adjacent string literal tokens were being concatenated before conversion of escape sequences to members of the execution character set – a reversal of translation phases 5 and 6. This resulted in the generation of incorrect escape sequences and generation of message 0338. *0338: [C] Octal or hex escape sequence value is too large for 'unsigned char' or 'wchar_t' type.*
12819	F	An undocumented feature of the #pragma PRQA_MESSAGES_ON/OFF facility previously permitted the list of message numbers to be specified with a macro. This functionality was removed in QAC 7.1 and has now been restored.
12821	F	Windows GUI: corrected the visual confusion in "Add Pers" and "Apply Pers" buttons in Personality Listings.
12823	F	In Windows GUI, Personalities Listing was incorrectly adding the same folder in Personalities listings after repeatedly browsing for a new personality file.
12829	F	*3007: "void" has been omitted when defining a function with no parameters.* Message 3007 was sometimes generated incorrectly when the definition of a function differed from a previous declaration in the use of typedefs.
12830	F	*3406: Object/function '%s', with external linkage, has been defined in a header file.* *3480: Object/function '%s', with internal linkage, has been defined in a header file.* Messages 3406 and 3480 were generated incorrectly when an object was defined in the main source file with a tentative definition (i.e. with no explicit initializer), following a previous declaration in a header file.
12845	C	In Windows GUI, relative path implementation is now extended to Secondary Analysis settings.
12846	F	In Windows GUI, Apply Relative Path window had some incorrect tick-box visual enablement.
12850	F	*1012: [E] Use of a C++ reference type ('type &') will be treated as a language extension..* Message 1012 is no longer a level 9 (hard error) message. It has been reclassified as a level 6 (language extension) message.
12852	F	Unix GUI corrects a problem specifying an editor with kterm.
12853	C	Unix GUI now allows a single -spragma value.

CR	T	Resolution
12855	F	Unix GUI personality selection dialog now operates from the current personality path.
12857	F	Unix GUI was ignoring file selection in the demographic generation step within the metric browser.
12858	F	Unix GUI correctly finds a project entered as parameter to xqac.
12867	F	Metrics values such as STAV1 and STST1 were not calculated following a recoverable dataflow analysis failure.
12895	F	In Windows GUI, secondary analysis execution in multi-processor environment was intermittently failing due to EV/process issues.
12900	F	In Windows GUI, relative path implementation failed in certain circumstances to recreate the full project path correctly.
12905	F	In Windows GUI, manual setting of a Project Configuration File was not being retained.
12907	C	Windows GUI improves folder display & apply-to-project logic in Configuration | Personalities.
12909	F	Windows GUI was not noting alterations made in auto-adjustment when opening a project file.
12910	F	In Windows GUI, the root path (used in Relative path implementation) was not updating correctly when double-clicking to open a project file.
12915	F	In Windows GUI, project auto-create was wrongly testing for existence of a same-named folder.
12918	F	The symbolic type code in <DEFINE> records in the .met file sometimes referred incorrectly to a typedef name rather than the struct tag to which the typedef referred.
13000	F	A variadic macro definition could not be parsed when the ellipsis was followed by trailing space.
13034	F	Unix GUI Message Personality correction to operation of 'Save Message Settings by' setting.
13060	F	References associated with messages have been improved so that they consistently quote the ISO section heading rather than just the sub-section heading.
13066	F	In Windows GUI, relative path implementation was not allowing save of a RP project to a different filename for backup purposes.
13107	C	Windows GUI now eliminates, when possible, the parser command-line handshake with licence server.
13109	N	In Windows GUI, application and project-level environment variables can be defined and used in all GUI operations.
13118	N	In Windows GUI, baseline analysis is now fully automated through menu actions and analysis.
13119	N	Windows GUI now performs dependency checking on source, header files and configuration files for source file analysis.
13120	F	Initialization using a designated initializer sometimes resulted in a parsing failure.
13121	C	In Windows GUI custom reports and CMA analysis, the range of substitution parameters is extended for better per-project management.
13129	F	*0660: [E] Defining an unnamed member in a struct or union.* Message has been moved to level 6 (Language Extensions). Unnamed members in a struct or union are a language extension.
13140	N	Windows GUI automatically populates menu items for PDF Reports and Structure101 add-ons.
13144	F	*:0431 [C] Function argument points to a more heavily qualified type.* *:0562 [C] Right operand of assignment points to a more heavily qualified type.* *:0673 [C] Initializer points to a more heavily qualified type.* *:0757 [C] 'return' expression points to a more heavily qualified type.* Messages 0431, 0562, 0673 and 0757 were not always generated when the pointer addressed a struct/union member of array type.
13170	F	A nested compound literal resulted in a parsing error.
13172	F	Windows GUI was experiencing occasional start-up delays, caused by a visual component initialisation problem.
13203	F	Recoverable dataflow analysis problem occurring in special circumstances within a nested loop construct.
13221	F	Unix GUI Metric Browser had confusing logic in sorting by name and by metric.
13222	F	Unix GUI Metric Browser had an incorrect alphabetic order.

CR	T	Resolution
13252	C	*3314: This controlling expression is an assignment.* Message 3314 is no longer generated for controlling expressions in switch statements. Its purpose is to identify situations where an assignment operator (=) has been used mistakenly instead of an equality operator (==).
13254	C	*3314: This controlling expression is an assignment.* When an assignment operator is deliberately used in a controlling expression, it is conventional in open source environments to surround the assignment expression with an extra (redundant) set of parentheses in order to clarify the intention. Message 3314 is no longer generated in this situation.
13260	F	Recoverable dataflow analysis problem in a for loop construct with an uninitialized loop counter.
13282	C	0434: [S] The identifier '%s' has not been declared. 0473: [Q] Result of 'sizeof' operation will not fit in configured type for 'size_t'. Messages 0434 and 0473 have been reclassified as level 9 hard errors.
13286	F	Unix GUI had confusing logic where project files are removed and later displayed.
13289	N	New messages have been introduced to identify reuse of identifiers which exist in the system library. *4600: The macro '%1s' is also defined in '<%2s>'.* *4601: The macro '%1s' is the name of an identifier in '<%2s>'.* *4602: The identifier '%1s' is declared as a macro in '<%2s>'.* *4603: The object/function '%1s'is being defined with the same name as an ordinary identifier defined in '<%2s>'.* *4604: The object/function '%1s' is being declared with the same name as an ordinary identifier defined in '<%2s>'.* *4605: The typedef '%1s' is also defined in '<%2s>'.* *4606: The typedef '%1s' has the same name as another ordinary identifier in '<%2s>'.* *4607: The enum constant '%1s' has the same name as another ordinary identifier in '<%2s>'.* *4608: The tag '%1s' is also defined in '<%2s>'.*
13290	N	New messages have been introduced to identify use of identifiers which are reserved for implementation defined types and associated macros. *4620: The macro '%1s' may also be defined as a macro in '<%2s>'.* *4621: The macro '%1s' may also be defined as a typedef in '<%2s>'.* *4622: The identifier '%1s' may be defined as a macro in '<%2s>'.* *4623: The typedef '%1s' may also be defined in '<%2s>'.* *4624: The ordinary identifier '%1s' may be defined as a typedef in '<%2s>'.*
13291	N	New messages have been introduced to identify use of identifiers which are reserved for future library use. *4640: The macro '%1s' could conflict in the future with the name of a macro in '<%2s>'.* *4641: The identifier '%1s' could conflict in the future with the name of a macro in '<%2s>'.* *4642: The macro '%1s' could conflict in the future with the name of a function in '<%2s>'.* *4643: The identifier '%1s' could conflict in the future with the name of a function in '<%2s>'.* *4644: The macro '%1s' could conflict in the future with the name of a typedef in '<%2s>'.* *4645: The identifier '%1s' could conflict in the future with the name of a typedef in '<%2s>'.*
13306	F	Windows GUI experienced a rare process failure during analysis of extremely large projects.
13308	F	Dataflow analysis failed to analyse some complex value domains correctly and failed to identify invariant expressions.
13326	C	Unix GUI custom reports extend the range of substitution parameters for better per-project management.
13327	F	Unix GUI was incorrectly calculating scroll amount on certain project file list displays.

CR	T	Resolution
13330	F	*0586: [U] Division by constant zero.* Message 0586 was not always generated when expected in #elif expressions.
13333	F	Unix GUI metric browser was truncating the postcript file, when printing.
13335	F	In Windows GUI, negative numbers could be entered in Analyser Personality, Maximum Line Length field.
13344	F	Recoverable dataflow problem in special circumstances within unstructured code with an infinite loop.
13351	F	Dataflow analysis failed to analyse some complex value domains correctly.
13357	F	Dataflow analysis failed to analyse some complex value domains correctly.
13368	F	A parsing failure resulted from an attempt to redefine the keyword "main" using a macro defined on the command line.
13424	F	Message personality could be resized so as to hide some settings.