



QA C 7.2.2 Release Notes

October 2010

Contents

| | | |
|-----------|---|-----------|
| 1. | OVERVIEW | 2 |
| 1.1. | Support for Baselineing | 2 |
| 1.2. | Windows GUI Improvements | 2 |
| 1.3. | Message Browser Improvements | 2 |
| 1.4. | Help System Improvements | 2 |
| 1.5. | MISRA-C:2004 | 2 |
| 1.6. | Analysis Highlights | 2 |
| 2. | MESSAGE CHANGES | 4 |
| 2.1. | New messages | 4 |
| 2.2. | Messages with modified behaviour | 5 |
| 2.3. | Messages relocated to a different message group | 7 |
| 2.4. | Messages relocated to a different message level | 9 |
| 3. | CR SUMMARY | 10 |

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 `l64` and `ul64` 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 | <i>Array subscripting applied to an object of pointer type.</i> | 11316 |
| 0492 | <i>Array subscripting applied to a function parameter declared as a pointer.</i> | 11316 |
| 0619 | <i>[C] The identifier '%s' has already been defined in the current scope within the ordinary identifier namespace.</i> | 11064 |
| 0822 | <i>[Q] Cannot find forceinclude file '%s'.</i> | 12304 |
| 1042 | <i>[E] Using I64 or UI64 as an integer constant suffix. This is a language extension.</i> | 12213 |
| 1043 | <i>[E] Defining an anonymous union object. This is a language extension.</i> | 11064 |
| 1044 | <i>[E] Defining an anonymous struct object. This is a language extension.</i> | 11064 |
| 3377 | <i>Operand of a logical && or operator is a constant expression which is not a 'Boolean' value.</i> | 12402 |
| 3659 | <i>Unnamed zero-width bit-field declared with a signed type.</i> | 12710 |
| 4115 | <i>Operand of logical && or operator is not an effectively Boolean expression.</i> | 12402 |
| 4116 | <i>Operand of logical ! operator is not an effectively Boolean expression.</i> | 12402 |
| 4600 | <i>The macro '%1s' is also defined in '<%2s>'.</i> | 13289 |
| 4601 | <i>The macro '%1s' is the name of an identifier in '<%2s>'.</i> | 13289 |
| 4602 | <i>The identifier '%1s' is declared as a macro in '<%2s>'.</i> | 13289 |
| 4603 | <i>The object/function '%1s' is being defined with the same name as an ordinary identifier defined in '<%2s>'.</i> | 13289 |
| 4604 | <i>The object/function '%1s' is being declared with the same name as an ordinary identifier defined in '<%2s>'.</i> | 13289 |
| 4605 | <i>The typedef '%1s' is also defined in '<%2s>'.</i> | 13289 |
| 4606 | <i>The typedef '%1s' has the same name as another ordinary identifier in '<%2s>'.</i> | 13289 |
| 4607 | <i>The enum constant '%1s' has the same name as another ordinary identifier in '<%2s>'.</i> | 13289 |
| 4608 | <i>The tag '%1s' is also defined in '<%2s>'.</i> | 13289 |
| 4620 | <i>The macro '%1s' may also be defined as a macro in '<%2s>'.</i> | 13290 |
| 4621 | <i>The macro '%1s' may also be defined as a typedef in '<%2s>'.</i> | 13290 |
| 4622 | <i>The identifier '%1s' may be defined as a macro in '<%2s>'.</i> | 13290 |
| 4623 | <i>The typedef '%1s' may also be defined in '<%2s>'.</i> | 13290 |
| 4624 | <i>The ordinary identifier '%1s' may be defined as a typedef in '<%2s>'.</i> | 13290 |
| 4640 | <i>The macro '%1s' could conflict in the future with the name of a macro in '<%2s>'.</i> | 13291 |
| 4641 | <i>The identifier '%1s' could conflict in the future with the name of a macro in '<%2s>'.</i> | 13291 |
| 4642 | <i>The macro '%1s' could conflict in the future with the name of a function in '<%2s>'.</i> | 13291 |
| 4643 | <i>The identifier '%1s' could conflict in the future with the name of a function in '<%2s>'.</i> | 13291 |
| 4644 | <i>The macro '%1s' could conflict in the future with the name of a typedef in '<%2s>'.</i> | 13291 |
| 4645 | <i>The identifier '%1s' could conflict in the future with the name of a typedef in '<%2s>'.</i> | 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 | <i>Casting to different object pointer type.</i> | F | 12379 |
| 0338 | <i>[C] Octal or hex escape sequence value is too large for 'unsigned char' or 'wchar_t' type.</i> | C | 12792 |
| 0431 | <i>[C] Function argument points to a more heavily qualified type.</i> | F | 13144 |
| 0562 | <i>[C] Right operand of assignment points to a more heavily qualified type.</i> | F | 13144 |
| 0602 | <i>[U] The identifier '%s' is reserved for use by the library.</i> | C | 11241 |
| 0602 | <i>[U] The identifier '%s' is reserved for use by the library.</i> | F | 12781 |
| 0660 | <i>[E] Defining an unnamed member in a struct or union.</i> | F | 13129 |
| 0673 | <i>[C] Initializer points to a more heavily qualified type.</i> | F | 13144 |
| 0685 | <i>[C] Initializer for any object with static storage duration must be a constant expression.</i> | F | 12435 |
| 0757 | <i>[C] 'return' expression points to a more heavily qualified type.</i> | F | 13144 |
| 0778 | <i>[L] Identifier matches other identifier(s) (e.g. '%s') in first 31 characters - program does not conform strictly to ISO:C90.</i> | F | 12759 |
| 0779 | <i>[U] Identifier does not differ from other identifier(s) (e.g. '%s') within the specified number of significant characters.</i> | F | 12759 |
| 0823 | <i>[S] Unexpected '#else' or '#elif' directive follows '#else'.</i> | F | 12282 |
| 0826 | <i>[S] Unexpected ':' found without a preceding '?' in a '#if' or '#elif' expression.</i> | F | 11973 |
| 0827 | <i>[S] Missing ':' after '?' in a '#if' or '#elif' expression.</i> | F | 11973 |
| 0850 | <i>[C99] Macro argument is empty.</i> | F | 12354 |
| 0877 | <i>[C] '#if' and '#elif' expressions may contain only integral constants.</i> | F | 11973 |
| 0879 | <i>[S] Illegal operator in '#if' or '#elif' expression.</i> | F | 11973 |
| 0886 | <i>[S] Missing or invalid expression in '#if' or '#elif' directive.</i> | F | 11973 |
| 0896 | <i>[S] Missing operand in '#if' or '#elif' expression.</i> | F | 11973 |
| 0897 | <i>[S] Missing operator in '#if' or '#elif' expression.</i> | F | 11973 |
| 0898 | <i>[S] Unexpected ')' or ':' in '#if' or '#elif' expression.</i> | F | 11973 |
| 0899 | <i>[S] Preprocessing directive appears in the middle of a line.</i> | F | 13249 |
| 1012 | <i>[E] Use of a C++ reference type ('type &') will be treated as a language extension..</i> | F | 12850 |
| 1028 | <i>[E] Use of the sizeof operator in a preprocessing directive is a language extension.</i> | F | 11973 |
| 1304 | <i>Old style definition of function '%s()' is not portable to C++.</i> | F | 11477 |
| 2003 | <i>The preceding 'switch' clause is not empty and does not end with a 'jump' statement. Execution will fall through.</i> | F | 12719 |
| 2003 | <i>The preceding 'switch' clause is not empty and does not end with a 'jump' statement. Execution will fall through.</i> | F | 12790 |
| 2020 | <i>Final 'switch' clause does not end with an explicit 'jump' statement.</i> | F | 12789 |
| 2467 | <i>Loop control variable, %s, is not modified inside loop.</i> | F | 12197 |
| 2469 | <i>Loop control variable in this 'for' statement, %s, is modified in the body of the loop.</i> | F | 12197 |
| 3001 | <i>Function has been declared with an empty parameter list.</i> | F | 11477 |
| 3002 | <i>Defining '%s()' with an identifier list and separate parameter declarations is an obsolescent feature.</i> | F | 11477 |
| 3007 | <i>"void" has been omitted when defining a function with no parameters.</i> | F | 12829 |
| 3230 | <i>Address of automatic object assigned to local pointer with static storage duration.</i> | F | 11666 |
| 3314 | <i>This controlling expression is an assignment.</i> | C | 13252 |
| 3314 | <i>This controlling expression is an assignment.</i> | C | 13254 |
| 3322 | <i>Operand of a logical ! operator is a constant expression which is not a 'Boolean' value.</i> | C | 12402 |
| 3332 | <i>The macro '%s' used in this '#if' or '#elif' expression is not defined.</i> | F | 11973 |

| | | | |
|-----------|--|---|----------------|
| 3332 | <i>The macro '%s' used in this '#if' or '#elif' expression is not defined.</i> | F | 12544 |
| 3333 | <i>A 'break' statement has been used in the middle of a 'switch' 'case'/'default' clause.</i> | F | 12094 |
| 3392 | <i>A shift, relational or equality operator has been used more than once. Extra parentheses recommended.</i> | F | 12414 |
| 3394 | <i>Binary operator other than + - * / % used with different binary operator of the same precedence. Extra parentheses recommended.</i> | F | 12414 |
| 3396 | <i>Extra parentheses recommended. A binary operation is the operand of a conditional operator.</i> | F | 12414 |
| 3396 | <i>Extra parentheses recommended. A binary operation is the operand of a conditional operator.</i> | F | 12663 |
| 3397 | <i>Binary operator other than + - * / % used with binary operator of different precedence. Extra parentheses recommended.</i> | F | 12414 |
| 3399 | <i>Extra parentheses recommended. A unary operation is the operand of a logical && or .</i> | F | 12414 |
| 3400 | <i>Binary/ternary operator used with && or . Extra parentheses recommended.</i> | F | 11855 |
| 3401 | <i>Possible precedence confusion: extra parentheses are recommended here.</i> | F | 12414 |
| 3406 | <i>Object/function '%s', with external linkage, has been defined in a header file.</i> | F | 12830 |
| 3410 | <i>Macro parameter not enclosed in ().</i> | C | 11189 |
| 3480 | <i>Object/function '%s', with internal linkage, has been defined in a header file.</i> | F | 12830 |
| 3700-3881 | <i><Implicit conversion messages></i> | F | 11285 12029 |
| 3900-4081 | <i><Function return implicit conversion messages></i> | 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 | <i>[U] Left shift operation on constant signed expression generating an undefined value.</i> |
| 0277 | ISO_ImplDef | Maj_Ops | <i>Conversion of a constant negative value to an unsigned type.</i> |
| 0294 | ISO_ImplDef | ISO_ExpU | <i>[U] Definite signed left shift operation generating an undefined value.</i> |
| 0295 | ISO_ImplDef | ISO_ExpU | <i>[U] Apparent signed left shift operation generating an undefined value.</i> |
| 0434 | Constraint | Syntax | <i>[S] The identifier '%s' has not been declared.</i> |
| 0473 | Constraint | Config | <i>[Q] Result of 'sizeof' operation will not fit in configured type for 'size_t'.</i> |
| 0541 | ISO_ImpU | Constraint | <i>[C] Argument no. %s does not have object type.</i> |
| 0547 | ISO_ExpU | Constraint | <i>[C] This declaration of tag '%s' conflicts with a previous declaration.</i> |
| 0581 | Constraint | ISO_ImplDef | <i>[I] Floating-point constant may be too small to be representable.</i> |
| 0617 | Constraint | C99_ext | <i>[C99] 'const' qualifier has been duplicated.</i> |
| 0618 | Constraint | C99_ext | <i>[C99] 'volatile' qualifier has been duplicated.</i> |
| 0626 | ISO_ImpU | ISO_ExpU | <i>[U] '%s' has different type to previous declaration (which is no longer in scope).</i> |
| 0652 | ISO_ImpU | Min_Func | <i>Identifiers have been provided for some but not all of the parameters in a function prototype.</i> |
| 0660 | Min_Array | Lang_ext | <i>[E] Defining an unnamed member in a struct or union. This is a language extension.</i> |
| 0675 | ISO_ExpU | Constraint | <i>[C] Initializer is not of compatible 'struct'/'union' type.</i> |
| 0688 | Min_Array | Maj_Decl | <i>Array size determined by number of initializers which include concatenated string literals.</i> |
| 0689 | Maj_Redun | ISO_ImpU | <i>[u] 'Switch' statement will bypass the initialization of this local variable.</i> |
| 0706 | ISO_ImpU | ISO_ExpU | <i>[U] Label '%s' is not unique within this function.</i> |
| 0747 | ISO_ImpU | Constraint | <i>[C] 'return exp;' found in '%s()' whose return type is qualified 'void'.</i> |
| 0832 | Maj_Prepro | Min_Prepro | <i>Macro substitution in #include preprocessing directive.</i> |
| 0861 | Min_Prepro | Maj_Redun | <i>This #include <%s> directive is redundant.</i> |
| 0862 | Min_Prepro | Maj_Redun | <i>This #include "%s" directive is redundant.</i> |
| 0899 | Constraint | Lang_ext | <i>[E] Unrecognized preprocessing directive has been ignored - assumed to be a language extension.</i> |
| 1012 | Syntax | Lang_ext | <i>[E] Use of a C++ reference type ('type &') will be treated as a language extension.</i> |
| 1310 | Min_Cpp | Min_Ident | <i>'%s' is used as a tag and a typedef for the same 'struct' / 'union' / 'enum'.</i> |
| 1311 | Min_Cpp | Obsolete | <i>'void *' and 'const T *' pointers used as operands to an equality or conditional operator.</i> |
| 1313 | Min_Cpp | Obsolete | <i>Executing 'goto %s' will cause local initialization to be skipped.</i> |
| 1318 | Min_Enum | Min_Cpp | <i>Object of enum type is being modified with a compound assignment operator.</i> |
| 1319 | Min_Enum | Min_Cpp | <i>Object of enum type is being modified with an increment or decrement operator.</i> |
| 1460 | Min_Switch | Maj_Enum | <i>'Switch' label value, %s, not contained in enum type.</i> |
| 1461 | Min_Enum | Maj_Enum | <i>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.</i> |
| 1474 | Min_Enum | Maj_Enum | <i>Object of enum type is being modified with a bitwise compound assignment operator.</i> |
| 1477 | Min_Enum | Obsolete | <i>Object of enum type is being implicitly compared against zero in a controlling expression.</i> |
| 1479 | Min_Enum | Maj_Enum | <i>Object of enum type is being modified with an arithmetic compound assignment operator.</i> |

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

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

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: <i>Macro parameter not enclosed in ()</i> . 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: <i>[U] The identifier '%s' is reserved for use by the library</i> . 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: <i>Array subscripting applied to an object of pointer type rather than array type</i> . 0492: <i>Array subscripting applied to a function parameter declared as a pointer</i> . |
| 11477 | F | 1304: <i>Old style definition of function '%s()' is not portable to C++</i> . 3001: <i>Function has been declared with an empty parameter list</i> . 3002: <i>Defining '%s()' with an identifier list and separate parameter declarations is an obsolescent feature</i> . 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: <i>Address of automatic object assigned to local pointer with static storage duration</i> . Message 3230 was generated incorrectly in certain contexts. |
| 11855 | F | 3400: <i>Extra parentheses recommended. A binary operation is the operand of a logical && or </i> . 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: <i>[S] Unexpected ':' found without a preceding '?' in a '#if' or '#elif' expression</i> . 0827: <i>[S] Missing ':' after '?' in a '#if' or '#elif' expression</i> . 0877: <i>[C] '#if' and '#elif' expressions may contain only integral constants</i> . 0879: <i>[S] Illegal operator in '#if' or '#elif' expression</i> . 0886: <i>[S] Missing or invalid expression in '#if' or '#elif' directive</i> . 0896: <i>[S] Missing operand in '#if' or '#elif' expression</i> . 0897: <i>[S] Missing operator in '#if' or '#elif' expression</i> . 0898: <i>[S] Unexpected ')' or ':' in '#if' or '#elif' expression</i> . 1028: <i>[E] Use of the sizeof operator in a preprocessing directive is a language extension</i> . 3332: <i>The macro '%s' used in this '#if' or '#elif' expression is not defined</i> . |
| 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, "l64" and "Ul64". These are treated as equivalent to "LL" and "ULL" respectively. A new message is generated whenever these suffixes are encountered: 1042: [E] Using l64 or Ul64 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. Messages 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: <i>The macro '%s' used in this '#if' or '#elif' expression is not defined.</i> 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: <i>Extra parentheses recommended. A binary operation is the operand of a conditional operator.</i> 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 incorrect 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: <i>Unnamed zero-width bit-field declared with a signed type.</i> |
| 12719 | F | 2003: <i>The preceding 'switch' clause is not empty and does not end with a 'jump' statement. Execution will fall through.</i> 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 | <p>0778: [L] Identifier matches other identifier(s) (e.g. '%s') in first 31 characters - program does not conform strictly to ISO:C90.</p> <p>0779: [U] Identifier does not differ from other identifier(s) (e.g. '%s') within the specified number of significant characters.</p> <p>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.</p> <p>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.</p> |
| 12768 | N | <p>The text of many messages has been revised in order to improve clarity.</p> <p>The references and help text associated with all messages have been improved and extended.</p> |
| 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 | <p>0602: [U] The identifier '%s' is reserved for use by the library.</p> <p>Message 0602 was not generated for some identifiers, e.g.:</p> <ul style="list-style-type: none"> a) macros b) enum constants declared at file scope c) function parameters <p>See also CR 11241</p> |
| 12789 | F | <p>2020: Final 'switch' clause does not end with an explicit 'jump' statement.</p> <p>Message 2020 was generated incorrectly when the preceding case clause was terminated with a 'continue', 'goto' or 'return' statement.</p> |
| 12790 | F | <p>2003: The preceding 'switch' clause is not empty and does not end with a 'jump' statement.</p> <p>Execution will fall through.</p> <p>Message 2003 was not always generated when fall through did not occur on every path.</p> |
| 12792 | | <p>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.</p> <p>0338: [C] Octal or hex escape sequence value is too large for 'unsigned char' or 'wchar_t' type.</p> |
| 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 | <p>3007: "void" has been omitted when defining a function with no parameters.</p> <p>Message 3007 was sometimes generated incorrectly when the definition of a function differed from a previous declaration in the use of typedefs.</p> |
| 12830 | F | <p>3406: Object/function '%s', with external linkage, has been defined in a header file.</p> <p>3480: Object/function '%s', with internal linkage, has been defined in a header file.</p> <p>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.</p> |
| 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 | <p>1012: [E] Use of a C++ reference type ('type &') will be treated as a language extension..</p> <p>Message 1012 is no longer a level 9 (hard error) message. It has been reclassified as a level 6 (language extension) message.</p> |
| 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: <i>This controlling expression is an assignment.</i> 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: <i>This controlling expression is an assignment.</i> 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: <i>The macro '%1s' is also defined in '<%2s>'.</i> 4601: <i>The macro '%1s' is the name of an identifier in '<%2s>'.</i> 4602: <i>The identifier '%1s' is declared as a macro in '<%2s>'.</i> 4603: <i>The object/function '%1s' is being defined with the same name as an ordinary identifier defined in '<%2s>'.</i> 4604: <i>The object/function '%1s' is being declared with the same name as an ordinary identifier defined in '<%2s>'.</i> 4605: <i>The typedef '%1s' is also defined in '<%2s>'.</i> 4606: <i>The typedef '%1s' has the same name as another ordinary identifier in '<%2s>'.</i> 4607: <i>The enum constant '%1s' has the same name as another ordinary identifier in '<%2s>'.</i> 4608: <i>The tag '%1s' is also defined in '<%2s>'.</i> |
| 13290 | N | New messages have been introduced to identify use of identifiers which are reserved for implementation defined types and associated macros. 4620: <i>The macro '%1s' may also be defined as a macro in '<%2s>'.</i> 4621: <i>The macro '%1s' may also be defined as a typedef in '<%2s>'.</i> 4622: <i>The identifier '%1s' may be defined as a macro in '<%2s>'.</i> 4623: <i>The typedef '%1s' may also be defined in '<%2s>'.</i> 4624: <i>The ordinary identifier '%1s' may be defined as a typedef in '<%2s>'.</i> |
| 13291 | N | New messages have been introduced to identify use of identifiers which are reserved for future library use. 4640: <i>The macro '%1s' could conflict in the future with the name of a macro in '<%2s>'.</i> 4641: <i>The identifier '%1s' could conflict in the future with the name of a macro in '<%2s>'.</i> 4642: <i>The macro '%1s' could conflict in the future with the name of a function in '<%2s>'.</i> 4643: <i>The identifier '%1s' could conflict in the future with the name of a function in '<%2s>'.</i> 4644: <i>The macro '%1s' could conflict in the future with the name of a typedef in '<%2s>'.</i> 4645: <i>The identifier '%1s' could conflict in the future with the name of a typedef in '<%2s>'.</i> |
| 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 postscript 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. |
| 13602 | F | <p>A bug was introduced in QAC 7.2. The precise circumstances under which it occurred are complex but the most common conditions were when a typedef was defined as an enum, struct or union without a tag. The following symptoms have been observed.</p> <ol style="list-style-type: none"> 1) An incorrect type is output in "met" file <DEFINE> records. This gave rise to problems in secondary analysis and cross module analysis (CMA). 2) Message 3448 was not generated as expected. 3) Minor discrepancies were evident in the evaluation of the metrics STPRT and STMOB. |