LDRA Ltd

C/C++ LDRA tool suite RELEASE NOTES

For Platform specific configuration, please refer to the Installation Guide.

New Features for 9.8.6

LDRA Testbed Section

NEW

LDRA_EXCLUDE comments can have tags. For example:

/* LDRA_EXCLUDE 130 S <tag start> Deviation <tag end> */

NEW

Analysis scope report. A notification is added to the report if the Ctrl-z character (ASCII 26) is detected as an end of file marker.

NEW

Sysearch record type 302 added. This has the same action as type 300 (header is excluded from expansion) but is only used in the sysppvar generation phase.

This may be necessary where a header has constructs that prevent sysppvar generation from running.

NEW

Analysis issues found during MC/DC Test Case Planner phase cause a soft fail, allowing other phases to continue. Set MCDC_PLANNER_SOFT_FAIL=FALSE in INI to revert to previous behaviour.

This applies only to single files and system sets where continue_system_analysis is not set

NEW

Command line qualifier -disable_reports added. This disables all end of analysis and analysis phase reports.

The INI flag DISABLE_REPORT_WRITERS can also be used, if both are set the command line takes precedence.

NEW

Command line qualifiers -analyse_as_c and -analyse_as_cpp added. These override the analysis language for new source files, which would normally be determined by the file extension.

NEW

Section for setting default C/C++ language versions added to command_line.pdf.

NEW

INI file entries to set default C/C++ language versions:

C_LANGUAGE= < one of C99 or C11 >

CPP_LANGUAGE= < one of CPP11 or CPP14 or CPP17 >

For example:

C_LANGUAGE=C99

CPP_LANGUAGE=CPP14

These are in the same form as the command line/TCF entries and should be used in place of the older INI entries such as CPP14_DIALECT=TRUE in switching on specific language version dialect elements.

These set the default language version for new files and sets, and may be overridden by command line/TCF settings.

NEW

File specific search paths are incorporated into the $(Includedirs) command macro.

Set ADD_FILE_SPECIFIC_TO_INCLUDEDIRS_MACRO=FALSE to disable.

NEW

File specific macros can be incorporated into the $(Defines) command macro. Set ADD_FILE_SPECIFIC_TO_DEFINES_MACRO=TRUE to enable. This flag is FALSE by default as if sysppvar generation is used it is likely that inappropriate defines would be added to the $(Defines) expansion.

CHG

LCSAJ Test Case Planner report. An alternative faster method is used when calculating possible paths. The new method can in some cases generate different paths.

To restore the previous method set LCTCPLAN_OPTIMISE_PATH_CALC=FALSE in the INI file.

CHG

Dynamic dataflow. Declaration uses in unexecuted procedures are shown as unexecuted.

To revert to previous behaviour set DDFC_DECL_USE_UNEXECUTED_IN_PROC=FALSE

CHG

Dynamic dataflow. Variable uses on unexecutable lines in unexecuted procedures are shown as unexecuted. To revert to previous behaviour set

DDFC_SPECIAL_USE_UNEXECUTED_IN_PROC=FALSE

CHG

Dynamic dataflow. Improvements to reporting of same named variables.

CHG

Dynamic dataflow. Uses of member variables within member functions are reported as 'M' (or Member) type variables instead of 'P' (Parameter). The key to terms no longer refers to Constants.

CHG

Code Review report. Static analysis based rules in the summary table have '-' instead of '0' where analysis has failed. Qualsys rules ('U') have '-' where qualsys is not applicable or has not been run.

FIX

Processing of TCF files removes trailing tab characters from element keys and values.

FIX

Testbed GUI could fail to invoke graphics applications in certain circumstances.

FIX

Reporting of rule numbers for AUTOSAR-C++:19-03/19-11 standard model. (LM ref. 9474)

FIX

Corrected matching of short file names to files in set for TCF/BTF options such as compile_as_language.

FIX

Data Object Analysis. General enhancements to report including improvements to reporting of declaration types and removal of repeated file name entries.

FIX

Data Object Analysis. Improvements to reporting of different variable types with the same name.

TBvision Section

FIX

Enhanced invocation of System Callgraph. (LM ref. 9431)

TBmanager Section

CHG

External Tasks that fail to start execution will now return an exit code of -1. (LM ref. 8808)

FIX

Enhanced display of Unit Test Regression Report. (LM ref. 9304)

FIX

Enhanced import of Custom Attributes from ReqIF files. (LM ref. 9324)

Static Analysis Module

CHG

Enhanced Static Analysis. (LM ref. 9478)

CHG

Enhanced Static Analysis. (LM ref. 8935)

CHG

Enhanced Static Analysis of digraphs in C++. (LM ref. 8819)

CHG

Enhanced Static Analysis of C++11 alternative operators. (LM ref. 9030)

CHG

Enhanced Static Analysis of constexpr functions. (LM ref. 9351, 9353, 9355, 9356)

CHG

Enhanced Static Analysis of templated types in switch case statements. (LM ref. 9354)

CHG

Enhanced detection of 283 S. This may result in fewer reported instances. (LM ref. 9365)

CHG

Enhanced detection of 214 S. This may result in fewer reported instances. (LM ref. 9363, 9364)

CHG

Enhanced Static Analysis of function-try-blocks. (LM ref. 9370, 9372)

CHG

Enhanced detection of 93 S. This may result in fewer reported instances. (LM ref. 9285)

CHG

Enhanced detection of 105 S. This may result in fewer reported instances. (LM ref. 8776)

CHG

Enhanced detection of 120 S. This may result in fewer reported instances. (LM ref. 9376)

CHG

Enhanced Static Analysis. (LM ref. 9396)

CHG

Enhanced Static Analysis of templated return types. (LM ref. 8956)

CHG

Enhanced detection of 63 D. This may result in fewer reported instances. (LM ref. 9390

CHG

Enhanced Static Analysis. (LM ref. 9402)

CHG

Enhanced Static Analysis. (LM ref. 9401)

CHG

Enhanced detection of 47 S. This may result in fewer reported instances. (LM ref. 9410)

CHG

Enhanced Static Analysis. (LM ref. 9434)

CHG

Enhanced Static Analysis of constexpr. (LM ref. 9433)

CHG

Enhanced Static Analysis of initialiser lists. (LM ref. 9345)

CHG

Enhanced Static Analysis. (LM ref. 9145)

CHG

Enhanced Static Analysis of multiple condition statements with interleaved comments. (LM ref. 9221)

CHG

Enhanced Static Analysis. (LM ref. 9451)

CHG

Enhanced Static Analysis. (LM ref. 8936)

CHG

Enhanced Static Analysis of shift operators within a uniform initialisation. (LM ref. 9357)

CHG

Enhanced Static Analysis. (LM ref. 9178)

CHG

Enhanced Static Analysis. (LM ref. 9463)

CHG

Enhanced detection of 139 S. This may result in fewer reported instances. (LM ref. 9101)

CHG

Enhanced Static Analysis of && in a braced initializer. (LM ref. 9453)

CHG

Enhanced Static Analysis. (LM ref. 9291)

CHG

Enhanced Static Analysis of ternary expressions within template declarations. (LM ref. 8180)

CHG

Enhanced Static Analysis of ternary in template formal parameter list. (LM ref. 8179)

CHG

Enhanced Static Analysis. (LM ref. 9352)

CHG

Enhanced detection of 319 S. This may result in fewer reported instances. (LM ref. 9366)

CHG

Enhanced Static Analysis of template parameter lists. (LM ref. 9368)

CHG

Enhanced Static Analysis of flexible array members in C++. (LM ref. 9369)

CHG

Enhanced detection of 488 S. This may result in fewer reported instances. (LM ref. 8104)

CHG

Enhanced detection of 66 S. This may result in fewer reported instances. (LM ref. 8697)

CHG

Enhanced detection of 71 S. This may result in fewer reported instances. (LM ref. 9375)

CHG

Enhanced detection of 9 S. This may result in fewer reported instances. (LM ref. 8701)

CHG

Enhanced Static Analysis of macro expansions within an enumeration. (LM ref. 9391)

CHG

Enhanced Static Analysis. (LM ref. 8941)

CHG

Enhanced Static Analysis. (LM ref. 9423)

CHG

Enhanced Static Analysis of class members initialised within a class body (C++11). (LM ref. 9392)

CHG

Enhanced Static Analysis. (LM ref. 9427)

CHG

Enhanced Static Analysis. (LM ref. 9428)

CHG

Enhanced Static Analysis of calls with multiple parameters in noexcept operator. (LM ref. 9064)

CHG

Enhanced Static Analysis of constexpr. (LM ref. 9412)

CHG

Enhanced detection of 57 S. This may result in fewer reported instances. (LM ref. 5065

CHG

Enhanced Static Analysis. (LM ref. 9440)

CHG

Enhanced Static Analysis of lambda expressions within a uniform initialisation. (LM ref. 9441)

CHG

Enhanced Static Analysis. (LM ref. 9381)

CHG

Enhanced Static Analysis. (LM ref. 9009)

CHG

Enhanced Static Analysis. (LM ref. 8862)

CHG

Enhanced Static Analysis of construction via std::initializer_list. (LM ref. 9237)

CHG

Enhanced Static Analysis. (LM ref. 9358)

CHG

Enhanced Static Analysis. (LM ref. 9359)

CHG

Enhanced detection of 413 S. This may result in fewer reported instances. (LM ref. 8802)

CHG

Enhanced detection of 35 X. This may result in fewer reported instances. (LM ref. 8803)

CHG

Enhanced Static Analysis of ternary expressions. (LM ref. 9273)

CHG

Enhanced Static Analysis. (LM ref. 9063)

CHG

Enhanced detection of 384 S. This may result in fewer reported instances. (LM ref. 8598)

CHG

Enhanced Static Analysis. (LM ref. 8288

Data Flow Module

CHG

Enhanced detection of 128 D. This may result in fewer reported instances. (LM ref. 9411)

CHG

Enhanced detection of 46 D. This may result in fewer reported instances. (LM ref. 9349)

CHG

Enhanced detection of 105 D. This may result in fewer reported instances. (LM ref. 8360)

CHG

Enhanced detection of 45 D. This may result in fewer reported instances. (LM ref. 8700)

CHG

Enhanced Data Flow Analysis. (LM ref. 9344)

C/C++ Penalties and Standards Mappings

NEW

16 V Base copy/move ctor/assign neither protected or delete.

C++ Mapping: AUTOSAR-C++:17-03 A12-8-6

C++ Mapping: AUTOSAR-C++:18-10 A12-8-6

C++ Mapping: AUTOSAR-C++:19-03/19-11 A12-8-6

NEW

17 V Assignment operator not declared with the ref-qualifier &.

C++ Mapping: AUTOSAR-C++:17-03 A12-8-7

C++ Mapping: AUTOSAR-C++:18-10 A12-8-7

C++ Mapping: AUTOSAR-C++:19-03/19-11 A12-8-7

C++ Mapping: HIC++v4 12.5.7

NEW

18 V Derived classes shall use inheriting constructors.

C++ Mapping: AUTOSAR-C++:18-10 A12-1-6

C++ Mapping: AUTOSAR-C++:19-03/19-11 A12-1-6

NEW

19 V The use of the std::auto_ptr type has been deprecated.

C++ Mapping: AUTOSAR-C++:17-03 A18-1-3

C++ Mapping: AUTOSAR-C++:18-10 A18-1-3

C++ Mapping: AUTOSAR-C++:19-03/19-11 A18-1-3

C++ Mapping: HIC++v4 1.3.4

NEW

20 V Comparison operator is a member function.

C++ Mapping: AUTOSAR-C++:18-10 A13-5-5

C++ Mapping: AUTOSAR-C++:19-03/19-11 A13-5-5

NEW

21 V Comparison operator has differing parameter types.

C++ Mapping: AUTOSAR-C++:18-10 A13-5-5

C++ Mapping: AUTOSAR-C++:19-03/19-11 A13-5-5

NEW

22 V Comparison operator declared without noexcept qualifier.

C++ Mapping: AUTOSAR-C++:18-10 A13-5-5

C++ Mapping: AUTOSAR-C++:19-03/19-11 A13-5-5

NEW

23 V A relational operator must return a boolean value.

C++ Mapping: AUTOSAR-C++:17-03 A13-2-3

C++ Mapping: AUTOSAR-C++:18-10 A13-2-3

C++ Mapping: AUTOSAR-C++:19-03/19-11 A13-2-3

C++ Mapping: HIC++v4 13.2.2

NEW

24 V User-declared constructor takes std::initializer_list.

C++ Mapping: AUTOSAR-C++:17-03 A8-5-4

C++ Mapping: AUTOSAR-C++:18-10 A8-5-4

C++ Mapping: AUTOSAR-C++:19-03/19-11 A8-5-4

NEW

25 V Function containing forwarding reference is overloaded.

C++ Mapping: AUTOSAR-C++:17-03 A13-3-1

C++ Mapping: AUTOSAR-C++:18-10 A13-3-1

C++ Mapping: AUTOSAR-C++:19-03/19-11 A13-3-1

C++ Mapping: HIC++v4 13.1.2

NEW

27 V Class derived from multiple non-interface (Autosar) classes.

C++ Mapping: AUTOSAR-C++:17-03 A10-1-1

C++ Mapping: AUTOSAR-C++:18-10 A10-1-1

C++ Mapping: AUTOSAR-C++:19-03/19-11 A10-1-1

NEW

29 V Virtual should contain one of virtual, override or final

C++ Mapping: AUTOSAR-C++:17-03 A10-3-1

C++ Mapping: AUTOSAR-C++:18-10 A10-3-1

C++ Mapping: AUTOSAR-C++:19-03/19-11 A10-3-1

NEW

30 V Overriding virtual fn not declared with override or final.

C++ Mapping: AUTOSAR-C++:17-03 A10-3-2

C++ Mapping: AUTOSAR-C++:18-10 A10-3-2

C++ Mapping: AUTOSAR-C++:19-03/19-11 A10-3-2

NEW

31 V Class derived from multiple non-interface (Pure) classes.

CHG

214 S Member not declared virtual.

C++ Mapping removed: AUTOSAR-C++:17-03 A10-3-1,A10-3-2

C++ Mapping removed: AUTOSAR-C++:18-10 A10-3-1,A10-3-

C++ Mapping removed: AUTOSAR-C++:19-03/19-11 A10-3-1,A10-3-2

CHG

283 S Multiple direct inheritance found.

C++ Mapping removed: AUTOSAR-C++:17-03 A10-1-1

C++ Mapping removed: AUTOSAR-C++:18-10 A10-1-1

C++ Mapping removed: AUTOSAR-C++:19-03/19-11 A10-1-1

CHG

331 S Literal value requires a U suffix.

Vals flag 309 set on for MISRA-AC

Vals flag 309 set on for MISRA-C:2004

CHG

522 S Public assign operator in abstract class.

C++ Mapping removed: AUTOSAR-C++:17-03 A12-8-6

C++ Mapping removed: AUTOSAR-C++:18-10 A12-8-6

C++ Mapping removed: AUTOSAR-C++:19-03/19-11 A12-8-6

CHG

659 S Overriding a virtual function without override specifier

C++ Mapping removed: AUTOSAR-C++:17-03 A10-3-2

C++ Mapping removed: AUTOSAR-C++:18-10 A10-3-

C++ Mapping removed: AUTOSAR-C++:19-03/19-11 A10-3-2

CHG

684 S Member function specified virtual and final.

C++ Mapping removed: AUTOSAR-C++:17-03 A10-3-1

C++ Mapping removed: AUTOSAR-C++:18-10 A10-3-

C++ Mapping removed: AUTOSAR-C++:19-03/19-11 A10-3-

CHG

685 S Member function specified virtual and override.

C++ Mapping removed: AUTOSAR-C++:17-03 A10-3-1

C++ Mapping removed: AUTOSAR-C++:18-10 A10-3-

C++ Mapping removed: AUTOSAR-C++:19-03/19-11 A10-3-1

CHG

686 S Member function specified override and final.

C++ Mapping removed: AUTOSAR-C++:17-03 A10-3-1

C++ Mapping removed: AUTOSAR-C++:18-10 A10-3-1

C++ Mapping removed: AUTOSAR-C++:19-03/19-11 A10-3-1

CHG

13 C Headline updated to "More than *** essential knots."

Utilities Section

NEW

Dialect Generator. Generation of stub tcf files is controlled by INI flag DGEN_STUB_TCF. Set = 0 to disable, or 1 to generate a single stub tcf. The default (-1) generates multiple stub tcf files.

NEW

Dialect Generator. Generates a tcf file containing type 300 header file exclusion records. These are the headers expanded during the analysis when generating the dialect and which should not be expanded when using the dialect data files.

Can be applied using the command line qualifier -tbedtcf=<tcf options file>.

NEW

Added Visual Studio Code as a Source Code Browser option. (LM ref. 9481)

NEW

Added Visual Studio 2019 as a compiler option. (LM ref. 9284, 9286, 9404, 9485)

NEW

TBmakelogparser. Detects any specified compiler language arguments, e.g. /Tp or -x c++ and assigns the appropriate language to the source file(s) in the generated BTF(s).

NEW

TBbuildimport - Merge scripts Start-In directory accepts $(Buildstartdir) macro, substituting the Builds Start-In in its place. This can be used in both the field on the Settings page of the dialog and the INI entry

TBBUILDIMPORT_MERGE_TARGET_FILE_SCRIPT_STARTIN. (LM ref. 9430)

NEW

TBglhapi. Added functions printoutUsingList and printoutUsingInformation to main_interface_examples.

NEW

LDRA Launcher - Added Clear Locks to Configuration Options on the Configuration page. (LM ref. 8335)

CHG

TBmakelogparser. Entries with blank/empty start-in directories are now ignored by default. This may result in the generation of fewer PTFs. To process entries with blank/empty start-in directories set the ini file entry

TBMAKELOGPARSER_IGNORE_EMPTY_STARTIN=FALSE.

CHG

CSV2TCF. Updated to handle alternate naming conventions. (LM ref. 9471)

CHG

Cashregister 6 Example. Updated example and source exclusion annotations. (LM ref. 9479, 9486)

CHG

Cashregister 6 Example. Updated Installation macros. (LM ref. 9209)

CHG

C++ Cashregister 6 Example. Updated example and source exclusion annotations. (LM ref. 9480, 9487)

CHG

Tunnel Examples. Updated Makefile. (LM ref. 9489)

CHG

Safe Utilities Example. Updated Source and TBmanager project configuration. (LM ref. 9432, 9482)

CHG

Installer. Additional exclusions added for installation checks. (LM ref. 9437)

CHG

Optimised compiler options available for point products. (LM ref. 9516)

CHG

Updated Jenkins Plugin for compatibility with Linux and newer versions of Jenkins.

CHG

Updated Command Line Guide. (LM ref. 9399)