

C and Cpp Analyzer 2.0 - Release Notes

- [2.0.8-funcrel](#)
 - [Resolved Issues](#)
- [2.0.7-funcrel](#)
 - [Resolved Issues](#)
 - [Other Updates](#)
- [2.0.6-funcrel](#)
 - [Resolved Issues](#)
- [2.0.5-funcrel](#)
 - [Updates](#)
 - [Resolved Issues](#)
- [2.0.4-funcrel](#)
 - [Resolved Issues](#)
- [2.0.3-funcrel](#)
 - [Updates](#)
 - [Resolved Issues](#)
- [2.0.2-funcrel](#)
 - [Updates](#)
- [2.0.1-funcrel](#)
 - [Updates](#)
- [2.0.0-beta3](#)
 - [Updates](#)
- [2.0.0-beta2](#)
 - [Updates](#)
- [2.0.0-beta1](#)
 - [Updates](#)
- [2.0.0-alpha3](#)
 - [Updates](#)
- [2.0.0-alpha2](#)
 - [Updates](#)
- [2.0.0-alpha1](#)
 - [Updates](#)

2.0.8-funcrel

Resolved Issues

Customer Ticket Id	Details
23050	com.castsoftware.cpp 2.0.5-funcrel fails at application level with "KeyError"

2.0.7-funcrel

Resolved Issues

Customer Ticket Id	Details
24685	Incorrect links to Cast-Sytem objects leading to false violations
16885	No Warning if not able to evaluate conditional expression
20168	Onboarding CPP analyzer: Missing Method Objects linked to lambda expressions
22784	com.castsoftware.cpp 2.0.5-funcrel fails at application level with "KeyError"
19318	Perm fix: PROBLEM- Objects that do not use the "gets()" function - false violation

Other Updates

Details
False positive for the diag: Avoid setting a pointer to the address of a local variable defined in a lower scope

False positive for QR Do not return a dereferenced pointer allocated inside the function
DELTATCC - CPP - False violation - Avoid unreferenced Methods
DELTATCC - CPP - False violation - Ensure that dynamically allocated memory on local pointer in functions/methods is freed.
DELTATCC - CPP - Missing violation - Ensure that dynamically allocated memory on local pointer in functions/methods is freed.
DELTATCC - CPP - Rational to be corrected - Ensure that dynamically allocated memory on local pointer in functions/methods is freed.

2.0.6-funcrel

Resolved Issues

Internal ID	Call ID	Summary
CPP-225	20462; 20294	Analysis is crashing during the External Link step.
CPP-92		Incorrect devirtualisation link in case of Class::method call

2.0.5-funcrel

Updates

CPP-109: Update connectivity to be able to analyze C++ source code on a PostgreSQL 10, or PostgreSQL 11 server

Resolved Issues

Internal ID	Call ID	Summary
CPP-124	-	Perm fix: ESP - False positive for QR- Avoid redeclaring and overriding Methods with different default parameter values

2.0.4-funcrel

Resolved Issues

Internal ID	Call ID	Summary
CPP-139	-	CASTONCAST: cpp analysis produces many errors "Run-time exception occurred..." and "Major issue, skipping: access violation"
CPP-135	18122	Permanent Fix - CPP Application extension warning- Plugin has encountered the following error: Traceback (most recent call last)
CPP-133	-	CASTONCAST: App Level "virtual call" link creation step is failing once every 2 snapshots

2.0.3-funcrel

Updates

- CPP-125: Added queries objects for Pro*C EXEC SQL

Resolved Issues

Internal ID	Call ID	Summary
CPP-94	15092	CPP Analysis Unit log validation getting the Warning Job execution Syntax error

CPP-117	16562	Permanent fix: Link not created between C Function and Oracle Table
CPP-126	17501	Missing links from C++ function to functions in Pro C files
CPP-120	17207	Permanent fix: Missing headers warnings in log while the header files are present in source
CPP-113	16670	Permanent fix: C analysis crashed with error Cppcmd.xml" exited with code -1073741819

2.0.2-funcrel

Updates

- CPP-102 : removed rule "Avoid use of unvalidated input in array indexing and memory allocation due to huge performances issues"
- CPP-104, CPP-108 : removed crashes
- Solved a performance issue when having an Analysis Unit with lots of files

2.0.1-funcrel

Updates

Following new rules have been added in this release of the extension:

1065008	Avoid use of unvalidated input in array indexing and memory allocation
1065004	Ensure that dynamically allocated memory on local pointer in functions/methods is freed
1065006	Avoid iterator invalidation

2.0.0-beta3

Updates

- Removed false violations on:
 - Avoid local variables that are not initialized at declaration time
 - Ensure you provide a user defined copy constructor or disable copy when a class allocates memory in its constructor

2.0.0-beta2

Updates

- crash corrections
- Removed false violations on:
 - Avoid Constructors not supplying an initial value for all non-static data members
 - Ensure you provide a user-defined copy constructor or disable copy when a class allocates memory in its constructor

2.0.0-beta1

Updates

- enhance identity stability for methods/functions
- support of C++ 14/17 syntaxes
- Removed false violation on:
 - Do not return a dereferenced pointer allocated inside the function
 - Ensure you provide a user-defined copy constructor or disable copy when a class allocates memory in its constructor

2.0.0-alpha3

Updates

- Enhance identity stability for methods/functions
- New rules:
 - "continue" should not be used
 - "enum" members other than the first one should not be explicitly initialized unless all members are explicitly initialized.

2.0.0-alpha2

Updates

- Analyzer is now able to find more include files by itself - therefore providing more resistance to incorrect/incomplete include paths
- Corrects an issue with pre-compiled headers on Visual Studio 2010
- Corrects bookmarks between C/C++ functions and Oracle table objects
- Callback links
- Virtual call links

2.0.0-alpha1

Updates

- Execution units
 - now all Analysis Units are analyzed together in the same run, speeding up the analysis and generating only one log per execution unit