

Release Notes

In this section:

- [Version 2020.2](#)
- [Version 2020.1](#)
- [Version 10.4.3](#)
- [Version 10.4.2](#)
- [Version 10.4.1](#)
- [Version 10.4.0](#)
- [Version 10.3.4](#)

Version 2020.2

This is a general service update.

Version 2020.1

RuleWizard for C/C++

The following properties have been added to the C/C++test dictionary:

- `IsAtomic`; see [IsAtomic](#).
- `Result`, see [Result](#).

The following nodes have been added:

- `alignof`; see [alignof](#).
- `Generic Selection`; see [Generic Selection](#).

Other improvements:

- In the C/C++test dictionary, the `Kind` property for the `String Constant` node contains new values: `char8_t`, `char16_t`, `char32_t`; see [Kind](#).
- In the C++Text dictionary, the `Body` property is no longer available for the `#error` node.

Version 10.4.3

RuleWizard for C/C++

The following properties have been added to the C/C++test dictionary:

- `HasDirectInitializer`; see [HasDirectInitializer](#).
- `HasBracedInitializer`; see [HasBracedInitializer](#).
- `Column-End`; see [Column-End](#).
- `UsesOperatorSyntax`; see [UsesOperatorSyntax](#).
- `IsNoReturn`; see [IsNoReturn](#).

Other improvements:

- The `Line-End` property from the C/C++ dictionary is now supported for all Statements; see [Line-End](#).
- The `StartLine` property from the C/C++ dictionary is now supported for variables and typedefs; see [StartLine](#).
- The `IsCompilerGenerated` property from the C/C++ dictionary now returns false when a(b) is an implicit call of the operator function; see [IsCompilerGenerated](#) and [UsesOperatorSyntax](#).

Version 10.4.2

RuleWizard for C/C++

The following nodes have been added to the C/C++test dictionary:

- `Decltype`; see [Decltype](#).

The following properties have been added to the C/C++test dictionary:

- `IsDefaultValue`; see [IsDefaultValue](#).

- `IsThreadLocal`; see [IsThreadLocal](#).
- `IsValueInitialized`; see [IsValueInitialized](#).
- `HasStaticInitializer`; see [HasStaticInitializer](#).

Other improvements:

- The `Init` property from the C/C++ dictionary is now supported for the `if` and `switch` nodes. See [Init](#) for usage.
- The `Attributes` property from the C/C++ dictionary is now supported for the `statements` nodes. See [Attributes](#) for usage.
- The `OperatorName` property from the C/C++ dictionary is now supported for the `a.b`, `a->b`, `a.*b`, `a->*b` nodes. See [OperatorName](#) for usage.

RuleWizard for Java

The `var` node has been added to the Java dictionary.

The `Module` node category has been added to the Java dictionary. The category includes the following nodes:

- `Module Declaration`
- `Requires Directive`
- `Exports Directive`
- `Opens Directive`
- `Provides Directive`
- `Uses Directive`
- `Opens Entry`
- `Exports Entry`
- `Provides Entry`

The following nodes have been added to the `JavaText` dictionary:

- `->`
- `::`
- `var`
- `module`

Other improvements:

- The `ColonColon` node from the Java dictionary has been renamed as `::`. The previous name is only supported for backward compatibility.

Version 10.4.1

The `IsBuiltin` property is now supported for types. See [IsBuiltin](#) for usage.

Version 10.4.0

The `InitializerInClass` property has been added to the C/C++ dictionary. See [InitializerInClass](#) for usage.

Version 10.3.4

The `IsBraceNotationCast` property has been added to the C/C++ dictionary. See [IsBraceNotationCast](#) for usage.