# Release Notes T3Q v2.1.0

This is a major new release of T3Q adopting a modern infrastructure based on Eclipse EMF and Xtext. While effort has been invested in migrating all the existing functionality and maintaining the existing interfaces to the user, there may be some changes and pending issues worth noting.

## Changed in T3Q v2.1.0b2

- Increased version number, build 3 for v2.1.0
- Added support for TTCN-3 v4.10.1 (including features from v4.8.1 and v4.9.1, as well as select features from newer versions, e.g. open type and automatic type)
- Changed warning label for **checkNoUninitializedFieldsInTemplates**, now correctly shows **checkNoUninitializedFieldsInTemplates** instead of **checkUnreferencedFields**
- Improved the following new features
    - ♦ Check for uninitialised fields in templates
        - ◊ consider fields from base template in the case of modified templates (one level - directly modified template only)
        - ◊ optionally consider fields from multiple levels of modified templates (indirectly modified templates as well)

## Changed in T3Q v2.1.0b2

- Increased version number, build 2 for v2.1.0
- Added support for TTCN-3 v4.7.1 (+ select features from newer versions, e.g. open type and automatic type)
- Added the following new features
    - ♦ Check for AnyValueOrNone assignment in "record of" list
    - ♦ Check for uninitialised fields in templates
        - ◊ currently only for types defined in TTCN-3 (a solution for types imported from ASN.1 is being explored)
        - ◊ currently only for declared templates (template variables and other cases may be added as needed)
- Added the following fixes and improvements:
    - ♦ Improved reference resolution

## Changed in T3Q v2.1.0b1

- Increased version number, first build for v2.1.0
- Updated to support Java 21 (**requires Java 21**)
- Updated dependencies (Xtext, EMF, etc.) and addressed compatibility issues
- Initiated transition of development to GitLab, project to be made public once complete
- Added the following new features
    - Check for worthless use of valueof
    - Support for @fuzzy modifier (updates to more recent versions of TTCN-3 underway)
    - Support for int2enum and other predefined functions outside of OpCalls/assignments
- Added the following fixes and improvements:
    - Improved reference resolution
    - Improved handling of reference chains in recursive checkNoOverSpecificRunsOn
    - Fixed problems with checkLocalDefinitionsComeFirst
    - Fixed deadlock with checkNoOverSpecificRunsOn
    - Fixed incorrect results with checkNoUnusedFormalParameters
    - Fixed unreliable information messages for cyclic calls in recursive checkNoOverSpecificRunsOn, also causing errors
    - Build automation with Maven (enables faster releases, better testing)
    - Continuous integration with GitLab (enables faster releases, better testing)
    - Batch reference resolver (disabled by default)
    - Single core mode (disabled by default)
    - Reduced dependencies and JAR size
    - Related refactorings
- Known issues
    - Processing times may increase by 5-10% (on average, based on measurements during testing), due to new / more detailed checks, deeper analysis as a result of improved reference resolution, and upstream changes, some optimisations already added, further optimisations will be added in future releases.

# Release Notes T3Q v2.0.0

This is preview for a major new release of T3Q adopting a modern infrastructure based on Eclipse EMF and Xtext. While effort has been invested in migrating all the existing functionality and maintaining the existing interfaces to the user, there are some changes and pending issues worth noting.

## Changed in T3Q v2.0.0b30

- Increased version number
- Updated dependencies (Xtext, EMF, etc.)
- Improved reporting of unused local definitions
- Added the following fixes and improvements:
    - #79 Update XStream library
    - #80 Fixed processing of parameters in data flow validator
    - #81 Fixed updating of variable status for template variables
    - #82 Fixed processing of parameters in checkNoUnusedFormalParameters
    - Related refactorings and renames

## Changed in T3Q v2.0.0b29

- Increased version number
- Rolled back:
    - #77 Ignore assignments when checking for unused local definitions (needs further clarification)

## Changed in T3Q v2.0.0b28

- Increased version number
- Added the following fixes and improvements:
    - #78 Fix parsing issues for return ifpresent
    - #77 Ignore assignments when checking for unused local definitions

## Changed in T3Q v2.0.0b27

- Increased version number
- Added the following fixes and improvements:
    - #76 Change extendsInNoOverSpecificRunsOn to aliasInNoOverSpecificRunsOn

## Changed in T3Q v2.0.0b26

- Increased version number
- Added the following fixes and improvements:
    - #75 Add support for checking inherited component members in no over specific runs on
    - #74 Statistics mismatch when types are excluded in checkNoUninitialisedVariables

## Changed in T3Q v2.0.0b25

- Increased version number
- Added the following fixes and improvements:
    - #73 Add support for filtering checkNoUninitialisedVariables warnings by type
    - #72 Superfluous warning for uninitialised variables
    - #71 Improve reporting upon parsing issues
    - #70 Parsing issues with length (...)
    - #69 Wrong warnings raised as checkNoUnusedFormalParameters and checkNoUnusedLocalDefinitions
    - #68 Wrong warnings raised as checkNoUnusedImports
    - #67 Fix parsing issues for select union

## Changed in T3Q v2.0.0b24

- Increased version number
- Improved the following features:
    - #64 Distinguish between in and out parameters when checking whether parameters passed to a function are initialised

## Changed in T3Q v2.0.0b23

- Increased version number
- Added the following features:
    - #65 Migrate support for exporting low-level dependencies
    - #64 Distinguish between in and out parameters when checking whether parameters passed to a function are initialised
    - #63 Unify naming for checkNoUnitialisedVariables

## Changed in T3Q v2.0.0b22

- Increased version number
- Added the following feature:
    - #62 Add support for the identification of uninitialised variables

## Changed in T3Q v2.0.0b21

- Increased version number
- Fixed the following issues:
    - #61 Missing warnings on unused local definitions for components
    - #59 Investigate differences between output for unused imports in v1 and v2 (refined)

## Changed in T3Q v2.0.0b20

- Increased version number
- Fixed the following issues:
    - #60 Superfluous warning for unused formal timer parameters
    - #59 Investigate differences between output for unused imports in v1 and v2
    - #58 Over specific runs on erroneously triggered in certain cases
    - #57 Over specific runs on not triggered in obvious cases
    - #55 Unresolved function definitions cause a deadlock for over specific runs on check

## Changed in T3Q v2.0.0b19

- Increased version number
- Fixed the following issues:
    - #54 Superfluous warning for unused timer formal parameters
    - #55 Unresolved function definitions cause a deadlock for over specific runs on check
    - #56 Superfluous warning for over-specific runs on clauses

## Changed in T3Q v2.0.0b18

- Increased version number
- Fixed bug in validation rule checkNoOverSpecificRunsOnClauses

## Changed in T3Q v2.0.0b17

- Updated version number
- Fixed validation rule checkNoTabs

## Changed in T3Q v2.0.0b16

- Updated version number
- Improved validation rule checkNoUnusedFormalParameters: Unexpected warning in case of referenced timers in timeout statements is fixed

## Changed in T3Q v2.0.0b15

- Updated version number
- Updated Xtext to version 2.10.0
- Improved validation rule checkNoUnusedLocalDefinitions: Unexpected warnings in case of definitions within a "type component" are fixed
- Changed validation rule checkNoUnusedImports: Information massages about not resolved types are disabled (due to not parsed asn1 types)
- Changed Naming Conventions: Information massages about not resolved types are disabled (due to not parsed asn1 types)
- Validation rules checkTypesAndValuesModuleImportsLibNames and checkTestcasesModuleImportsLibCommon_Sync are now classified as test suite modularization

## Changed in T3Q v2.0.0b14

- Updated version number
- Fixed configuration issue concerning information messages about not resolved type declarations of naming conventions

## Changed in T3Q v2.0.0b13

- Updated version number
- Improved naming conventions check: The information message about not resolved type declarations is now printed
- Improved validation rule checkLogStatementFormat: information messages about not resolved functions are now printed
- Improved validation rule checkNoDuplicatedModuleDefinitionIdentifiers: checks for missing types are implemented, adjusted output to match the old version?s output
- Improved validation rule checkNoUnusedLocalDefinitions: fixed wrong warnings when a template is defined and is used as input parameter afterwards, show warnings for all not referenced ports in one module
- Improved validation rule checkInlineTemplates: reduced the number of messages

## Changed in T3Q v2.0.0b12

- Improved validation rule checkNoUnusedImports: Imported modules with language ASN are no longer considered
- Updated version number

## Changed in T3Q v2.0.0b11

- Updated Xtext to version 2.9.2
- Updated version number
- Improved object resolution. This results in a significant improvement of the following validation rules:
    - checkTemplatesModuleContainmentCheck
    - checkNoUnusedLocalDefinitions
    - checkNoUnusedFormalParameters
    - checkNoUnusedImports
    - checkZeroReferencedModuleDefinitions
- Fixed validation rule: checkNoOverSpecificRunsOnClauses
- Due to some changes Java Runtime Environment (JRE) version 1.8 or newer is now required

## Changed in T3Q v2.0.0b10

- Fixed feature ignoredResourceRegExp

## Changed in T3Q v2.0.0b9

- Fixed validation rule checkPortMessageGrouping
- Fixed validation rule checkLocalDefinitionsComeFirst
- Fixed validation rule checkNoOverSpecificRunsOnClauses
- Improved validation rule checkZeroReferencedModuleDefinitions
- Implemented new validation rule checkInlineTemplates (if this check is enabled it raises a lot of warnings)
- Improved performance
- Improved finding referenced variables
- Adjusted some printed line numbers according to the old version?s output
- Implemented feature to ignore desired resources according to configurable regular expressions
- Changes to the XML configuration:
    - Added new parameter ignoredResourceRegExp
    - Updated version number to 2.0.0.b9

- Changed internal processing order, therefore first the files will be parsed, then analyzed, and in the end the output will be printed. This means that the steps analyzing and printing are separated now.

## Changed in T3Q v2.0.0b8

- Added overall processing timer
- Added missing grammar rules to support TTCN-3 version 4.6.1 completely
- Improved checks for naming conventions
- Some minor improvements concerning the output
- Added new checks:
    - checkLevelOfNestedCalls: generates a warning when the level of nested calls (functions or templates) exceeds a given value
    - checkListedVariableDeclarations: generates a warning for listed variable declarations
- Added new configuration parameters:
    - boolean checkLevelOfNestedCalls: turns the corresponding check on or off
    - integer maxLevelOfNestedCalls: number of the maximum allowed nesting level
    - booelan checkListedVariableDeclarations: turns the corresponding check on or off
    - boolean checkInlineTemplates: turns the corresponding check, that will be implemented in future versions, on or off
- Updated version number

## Changed in T3Q v2.0.0b7

- Made it executable with Java version 1.7 again
- Updated version number

## Changed in T3Q v2.0.0b6

- Improved type system and resolution of type references
- Fixed some issues checking naming conventions
- Added new basic implementation of the validation rule checkNoUnusedImports because the old one did not work. Currently the following checks are working correctly:
    - If a imported module cannot be resolved
    - If no definitions of an imported module are ever used

  This validation rule needs further improvements and missing functionality will be implemented in future releases.
- Updated Xtext to version 2.8.4
- Minor improvements on modularization checks

- Improved performance finding modules that are importing an observed one. This speeded up all rules that need to know if one variable or type is referenced elsewhere, including checkNoUnusedImports.
- Disabled unresolved reference warnings because most of them are caused by the not supported (and resolvable) ASN1 types

## Changed in T3Q v2.0.0b5

- Improved performance
- Improved recognition of variable references (including the search in modules that are importing the module where the variable is defined in)
- Added XMI generation for testing purposes (XML configuration option *generateXMI*)
- Removed the following configuration options
    - *resourceExtensionsRegExp*, *projectExtension*: both options are removed because the new Xtext based system needs these information at build time
    - *ignoredResourceRegExp*: this option is currently not used in the new version and all resources in the input path will be parsed and analyzed (with the exception of using a single file as input, than only this file is processed)
    - *settingRecursiveProcessing*: this option is currently not used in the new version, thus all files in sub folders are processed

## Changed in T3Q v2.0.0b4

- Reduced number of not resolved objects
- Updated Xtext to version 2.7.3
- Removed slashes at the beginning of file names if option *showFullPath* is selected
- Fixed scoping issues including wrong warnings for not used parameters and variables
- Improved log statement validation

## Fixed in T3Q v2.0.0b3

- Issue with handling absolute paths

## Fixed in T3Q v2.0.0b2

- Illegal arguments startup issue

## Known Issues

- Some references may not be resolved correctly due to unresolved issues with upstream components
    - As a consequence some superfluous warnings may be produced (message class Universal)
- Some variable references may not be resolved correctly due to unresolved issues with upstream components
    - The following checks are affected when used with large TTCN-3 code-bases:
        - "There Must Be No Unused Definition On the Module Level"
        - "There Must Be No Unused Formal Parameters"
        - "There Must Be No Unused Local Definitions"
    - This also has an impact on the "There Must Be No Unused Imports" check, which is currently not enabled as a consequence
- Performance and processing time may vary between different platforms and configurations, as well as based on structural features of the TTCN-3 code-base

Potential solutions to these issues are currently being evaluated.

## Missing Features

- Support for automated code formatting with T3Q has not been migrated yet
- Support for "There Must Be No Unused Imports" check is not enabled yet due to unresolved issues
- The installation wizard has not been migrated yet

Support for these issues will be enabled and/or migrated once related issues have been resolved.

## Requirements, Installation, and Usage

- Java Runtime Environment (JRE), version 1.8 or newer shall be installed on the running system. A 64bit version is recommended, although the builds also run on 32bit systems and 32bit JRE.
- Please make sure that the path to the Java binaries is registered in the system PATH variable (see also ?https://www.java.com/en/download/help/path.xml).
- Download and unpack the `t3q-v2.0.0b19.zip` file
- Start T3Q by using the supplied start scripts `t3q.bat` for Windows-based systems and `t3q` for UNIX-based systems
- The configuration system and input interfaces have been is ported to T3Q v2.0.0b19, thus T3Q can be used as usual.
    - Note that the following options related to formatting and filtering are currently not supported: --format, --verbosity, --output-path.

- Default maximum memory settings have been set to 3GB
    - On systems with less available memory, this setting may need to be lowered in the `t3q.bat` and `t3q` files (changing `-Xmx3g` to e.g `-Xmx2g`