The Qualifying Machine

GNATCheck

- In Procedure Unreachable

Test Case
/tests/Improper_Returns_In_Procedure_Unreachable/

The artifact's content has been modified, but is not committed.

Test Result

- contentrst (text/x-rst)
- test.py (text/x-python)
- n_procedure_unreachable.ads (text/x-ada)
- n_procedure_unreachable.adb (text/x-ada)
- test.out (text/x-testoutput)

Artifacts can be opened through
the MDI (Multiple Document Interface) to be easily re-opened.
Each mime-type should be provided with both an editable and a read-only view. These views are used by this preview widget to display the content of the file. If no view is provided, source view is used by default.
Improper Returns

Requirement

/content.rst (text/x-rst)

---------------------

Improper Returns

---------------------

Return statements in procedures shall be flagged. Return statements after the first one in functions shall be flagged as well.

Testing strategy

---------------------

Check return statements in different contexts, including:

- blocks at different nesting levels
- reachable and unreachable return statements (for both procedures and functions)
- generic subprograms

... qml: TestCaseImporter

TestCases

The activity is now locked. No more test case can be created.
Improper Returns
Requirement
/requirements/Rule_Improper_Returns/

```
-----------
Improper Returns
-----------

Return statements in procedures shall be flagged. Return statements after the first one in functions shall be flagged as well.

Testing strategy
-----------

Check return statements in different contexts, including:

- blocks at different nesting levels
- reachable and unreachable return statements (for both procedures and functions)
- generic subprograms

```

Write a test case

- In Deep Nested Blocks
- In Function Reachable
- In Function Unreachable
- In Generic Definition
- In Procedure Reachable
- In Procedure Unreachable

qminic: TestCasesImporter

TestCases
The Qualifying Machine

In Procedure Unreachable
Test Case
/tests/Improper_Returns_In_Procedure_Unreachable/

The artifact's content has been modified, but is not committed.

- content.rst (text/x-rst)
- test.py (text/x-python)
- in_procedure_unreachable.ads
- in_procedure_unreachable.adb
- test.out (text/x-testoutput)

DEPENDENCIES

Requirement Improper_Returns

REPOSITORY /trunk/GNATCheck

OWNER delay

Improper Returns
Return statements in procedures shall be flagged. Return statements after the first one in functions shall be flagged as well.

Testing strategy

Check return statements in different contexts, including:
- blocks at different nesting levels
- reachable and unreachable return statements (for both procedures and functions)
- generic subprograms
- qmlink:: TestCaseImporter

Write a test case

This 'no-artifact-browser' view is also useful when confronting two entries of the MDI.

This is needed when working on a requirement and one of its test cases, or between the test case source file and its output.
The preview offers edition capabilities. This is goog for quick fixes.

For a more comfortable view, one can open the file as a new MDI's tab.
Opening a file as a tab permits the user to confront different types of file/artifact when using the dual view.

For example, one can edit both the body and the spec of an Ada package.

```
package In_Procedure_Unreachable is
  Condition : Boolean := True;

  procedure Simple_No_Params;

  procedure Simple_Params (A : in Boolean; B : out Boolean);

  procedure If_No_Params;

  procedure If_Params (A : in Boolean; B : out Boolean);

  procedure Loop_No_Params;

  procedure Loop_Params (A : in Boolean; B : out Boolean);

  procedure While_No_Params;

  procedure While_Params (A : in Boolean; B : out Boolean);

end In_Procedure_Unreachable;
```
The save button appears when changes occurred.
It is no use to weighten the interface with widget we don't need at a particular moment.
We only show them when we need them.

Both read-only and editable view are using the same modules associated with the mime-type of the file being manipulated than the one used by the preview panels.

... and edit!
Time to validate our project

One can choose between Quick Fixing the current opened artifact. This is useful when you have just fixed a type, and do not want to go through the entire validation process again (since modifying an artifact can invalidate its children).

This option should be available or not depending on the project type. This may be a variable set in the QM model which allows or not quick fixing.

The other option, ‘Validate’ is always available (from the moment a change occurs, and one or more artifacts need to be revaluated).
In Procedure Unreachable
Test Case
/tests/Improper_Returns_In_Procedure_Unreachable/

ARTEFACT LINKS
- Traceability links
- Requirement Improper Returns
- Dependency links
- Coding Standard
- Testing Plan
- Verified By
- Test Result Improper Returns

Requirement Improper Returns
/requirements/Rule_Improper_Returns/

Check unreachable return statements in procedures.

- content.rst (text/x-rst)
- test.py (text/x-python)
- in_procedure_unreachable.ads (text/x-ada)
- test.out (text/x-testoutput)

Displays the current state of the artifact's content.

Provides two different views:
- the current state of the artifact
- a diff view with a previous version

Provides a mechanism to switch between artifacts that need to be validated.

Provides a toggle button to tag the artifact as 'Validated'.

Provides a 'Commit' button available when at least one artifact as been validated.

Provide a view of the input and output links of the artifact.
This can be useful when revalidating an artifact which has not changed, but was invalidated by a modification done in another artifact.
Provides a tree view of artefacts in this validation set.

One can switch between artefact using two different ways:

- Through the left/right arrows, which navigate in the artefacts list which is sorted in a topological way.
- Through the tree view widget, which displays the artefacts respecting the traceability links.
The diff view confront two versions of the artefact contents. This view can confront any previous version of the artefact with the last one.
The comment scratch pad is a quick and easy way to write one's comments during the validation process, iteratively.

At the end, all comments written here are pasted in the final commit message.

This scratch pad is available through the toggle button close to the "commit" button.
Once reviewed, we can tag an artifact as 'Validated' through the toggle button.

From then, at least one artifact is validated, so the commit button becomes available.
A set of artifacts has been validated. One can commit it.

This allows multiple commits on the same set, dividing them into several atomic commits.

The comments are retrieved from the comments scratch pad. They can still be edited and completed before the final commit.
The artifact's content has been modified, but is not committed.

- content.rat (text/x-rat)
- test.py (text/x-python)
- in_procedure_unreachable.ads (text/x-ada)
- in_procedure_unreachable.adb (text/x-ada)
- test.out (text/x-textoutput)

**DEPENDENCIES**
- Requirement Improper_Returns

**REPOSITORY** /trunk/GNATCheck

**OWNER** delay