

SAP ABAP - Technical notes and limitations

- [Code considered as external/generated](#)
- [Syntax support](#)
- [Links to ABAP Macros](#)
- [Generic transaction support](#)
- [ABAP files ignored during an analysis](#)
- [SQL ABAP Metrics](#)
- [Structural Rule limitations](#)
- [Empty or syntactically erroneous files](#)
- [Line count of comments inside macros](#)
- [Dynamic references](#)
- [Unreachable code](#)



Summary: This section provides more detail about the support for SAP ABAP technologies and the way in which they are supported.

Code considered as external/generated

The SAP ABAP Analyzer will consider any component as **external/generated** when the last modification author for a given component is set to "SAP" in **PG_DESCR_*.xml**. When a component is considered as external/generated, it will not contribute to quality/quantity measurement results (lines of code, quality rules, transactions, etc.). Any external/generated component is considered to be third-party code which is not part of the analysis scope.

Syntax support

The following syntax (which is permitted in ABAP source code) is supported:

- "ENVIRONMENT TIME FORMAT" or "ENVIRONMENT FORMAT" (options for WRITE statements)
- "EXEC SQL" with comments between EXEC and SQL
- ~ character now supported in SORT clauses
- Preprocessing when ":" is inside parentheses
- A full expression in the FROM clause is now supported
- When an integer is present in a FROM clause
- When a macro is called with another macro name as a parameter
- "RISK"
- "CONNECTION"
- "CLASS RANDOM DEFINITION"
- RAISE EXCEPTION TYPE...MESSAGE
- Keywords: '*', '&&', '(', 'DATA', ',' and '['.

Links to ABAP Macros

While the ABAP Analyzer will create ABAP Macros as individual objects (which will be visible in CAST Enlighten), no links will be created from other ABAP objects that use the macros (i.e. the ABAP Macro object will always be "isolated").

Generic transaction support

The ABAP analyzer supports the following generic transactions and creates links from the initial transaction to the object that is passed to the generic transaction via a parameter:

- START_REPORT which allows programs to be run from custom transactions via parameters
- SE16

ABAP files ignored during an analysis

The following files will be ignored during an analysis:

- *CP.abap
- *CT.abap
- LSVIM* format .abap files

SQL ABAP Metrics

Metrics are only calculated on Open SQL. No metrics are calculated on ABAP Native SQL. However, ABAP Native SQL is detected for quality purposes.

Structural Rule limitations

7526 - Avoid using hardcoded paths

The Quality Rule "7526 - Avoid using hardcoded paths" will return false positive violations when a custom ABAP object (surrounded by single quote marks) is contained in a custom ABAP namespace, for example: `'names/a_namespace/a_name'`. This is a known limitation of CAST AIP and the recommendation is to exclude these violations from the next snapshot. Note that when single quote marks are not used to surround the custom ABAP object, no false violation will be reported.

Empty or syntactically erroneous files

- Even if a "Program" or "Include" is empty, i.e. it does not contain any code, a corresponding object is still created in the CAST Analysis Service.
- If the analysis of a file yields a syntax error, an object of type "Include" is created in the CAST Analysis Service. The object's name is the filename without extension. This object has no children as the analysis of the file's contents failed due to the syntax error.

Line count of comments inside macros

If an object calls a macro and the called macro contains comments, as expected, the macro's comments are not counted as comments of its calling object. However, if a macro contains a literal string and there is a comment inside the literal string, the comment inside the string is not removed before the macro is expanded in the object calling the macro. This comment is counted as many times as the macro is called by the object. Example of a macro with a string containing a comment:

```
DEFINE MyMacro.  
    MyString = 'Start of string  
                * any comment  
                End of string'.  
END-OF-DEFINITION.
```

Dynamic references

Dynamic references based on variables are not resolved by the analyzer - a link to an object named UNRESOLVED is created instead. A dynamic reference is a reference to an object whose name does not appear statically in the code but is the value of a variable. In this case, the variable name is placed between parentheses. Example: `Select * from (varTable)` where `varTable` is the name of a variable containing a table name. Such dynamic references often involve SAP tables and functions.

Unreachable code

In a program, the lines of code which are not attached to a sub-object and which are located after the first sub-object, are treated as unreachable. The lines of code are counted in the new artifact but they are not attached to it. Note that lines of code not attached to a sub-object and which are located before the first sub-object ARE taken into account by the ABAP analyzer via the creation of an ABAP File Level Code sub-object.