

Ask a Question Write a Blog Post



#### Personal Insights



#### Lars Hvam

January 5, 2024 | 1 minute read

## Using AI in ABAP development

□ 0 0 0 2 Edit

#### Follow





### My development setup

I typically use vscode for writing ABAP code in a standalone setup, editing just local files and committing those to git, this works on low-end laptops and tablets just fine.

I guess its also possible to use the ABAP remote file system extension from Marcello Urbani, but I've not tested it.

### The beginnings

I first tried GitHub Copilot back in the summer of 2021, it did give some suggestions, but had many syntax errors, so I switched it off again after a few hours of playing around.

```
• zcl_ntlm_md4.clas.abap - abapNTLM - Visual Studio Code
         ≡ zcl_ntlm_md4.clas.abap 1 ●
                                                                                                                 1 Synthesizing 10/10 solutions (Duplicates hidden)
                  DATA: lo_barrel TYPE REF TO lcl_barrel,
                    lo_buffer TYPE REF TO 1cl_buffer,
lo_ff TYPE REF TO 1cl_ff,
lo_gg TYPE REF TO 1cl_fg,
lo_hh TYPE REF TO 1cl_hh.
                                                                                                                    _ff->get_hash( ).

rv_hash = lo_gg->get_hash( ).
                                                                                                                         rv_hash = lo_hh->get_hash( ).
ENDIF.
                  lo_buffer = NEW #( iv_xstr ).
lo_barrel = NEW #( ).
lo_ff = NEW #( io_barrel = lo_barrel io_buffer = lo_buffer ).
                  logg = NEW #( io_barrel = lo_barrel io_buffer = lo_buffer ).
lo_hh = NEW #( io_barrel = lo_barrel io_buffer = lo_buffer ).
                                                                                                                    DO lo_buffer->get_blocks( ) TIMES.
lo_buffer->set_block( sy-index ).
lo_barrel->snapshot( ).
                                                                                                                            rv_hash = lo_hh->get_hash( ).
                    lo_ff->hash( ).
lo_gg->hash( ).
lo_hh->hash( ).
                                                                                                                20 _ff->get_hash().
21 | rv_hash = lo_gg->get_hash().
                 lo_barrel->accumulate( ).
                                                                                                                         rv_hash = lo_hh->get_hash( ).
endif.
                  rv_hash = lo_barrel->get_hash( ).
       145
                 if rv_hash = lo_ff->get_hash( ):
                                                                                                                    _ff->get_hash( ).

rv_hash = lo_gg->get_hash( ).
総
                                                                                                                34 _ff->get_hash( ).
                                                                                                                             ry hash = 10 gg->get hash( )
Ln 145, Col 21 Spaces: 2 UTF-8 LF ABAP abaplint: Ready 🔠 🔊 🚨
           ↔ ⊗ 1 🛆 0 Git Graph 🍪 abap | 🗸 zcl_ntlm_md4.clas.abap
```

### Now, Today!

Now I've been consistently using Copilot for the last month or so. And it has become part of my development flow, helping me to type less.

Its not perfect, but very useful especially when doing scaffolding or code that is very symmetric.

I've collected some of the good examples below,

Working as typeahead intellisense,

```
METHODS varint_one FOR TESTING RAISING cx_static_check.

METHODS decode int32 FOR TESTING RAISING cx_static_check.

Statement

ENDCLASS.
```

```
416
417 while strlem ( lv_bits ) > 0. Statement does

DATA(lv_bit) = lv_bits(1).

IF lv_bit = '1'.

rv_int = rv_int + lv_shift.

ENDIF.

lv_shift = lv_shift * 2.

lv_bits = lv_bits+1.

418
```

And scaffolding unit test methods,

### The process

I think, AI should not save and activate objects directly in the ABAP system. Instead changes should happen locally, so the developer can verify the changes. After verification the changes can be committed to git.

This triggers the normal quality assurance process in the pull request, kicking off static static analysis, automated unit testing and manual peer review.

After the peer review is done, the changes can be pulled to the central ABAP development system.







**Alert Moderator** 

#### Assigned Tags

ABAP Development

#### Similar Blog Posts

Announcing - Microsoft AI SDK for SAP

By Gopal Nair May 12, 2023

Al Powered Invoice Management with SAP RAP and ABAP on Cloud

By Sabarna Chatterjee Jun 16, 2023

#### A Cool use of Open AI in SAP GUI Editor

By Monalisa Biswal May 24, 2023

Related Questions		

#### DIFFERENCE B/W CHECK & IF STATEMENT.

By Former Member Jul 18, 2007

Has The New SAP Community Killed The Community?

By Former Member Oct 27, 2016

# Bookmarks-keybindings in Eclipse & ABAP Development Tools

Privacy

Legal Disclosure

Trademark

Newsletter

By Vlad Ghitulescu Jan 19, 2018

#### Be the first to leave a comment

**Add Comment** 

#### Find us on













Support