Title: German Tax Deduction Calculation using BRF+

1. Objective

The objective of this requirement is to design and implement a tax calculation engine in BRF+ that automates the derivation of German tax deductions. The system should calculate VAT, Corporate Tax, Solidarity Surcharge, and Trade Tax based on financial input values.

2. Scope

Automate multi-step tax calculations using BRF+ formulas and rulesets. Provide flexibility to adjust tax rates and formulas without modifying ABAP code. Ensure transparency in tax calculation logic by maintaining all rules in BRF+. Support simulation and testing of tax scenarios for validation.

3. Inputs

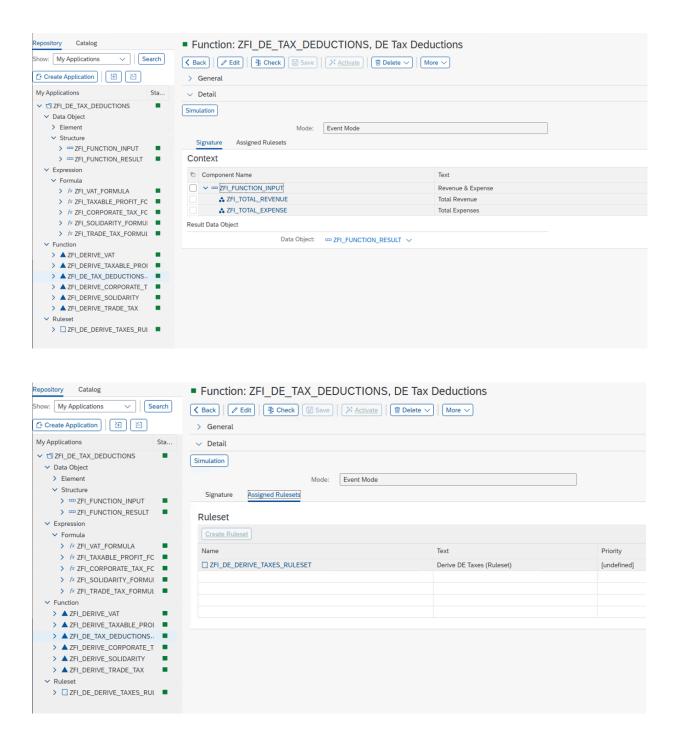
The function will accept the following input parameters (Data Object: `ZFI_FUNCTION_INPUT`):

- Total Revenue ('ZFI_TOTAL_REVENUE')
- Total Expenses (`ZFI_TOTAL_EXPENSE`)

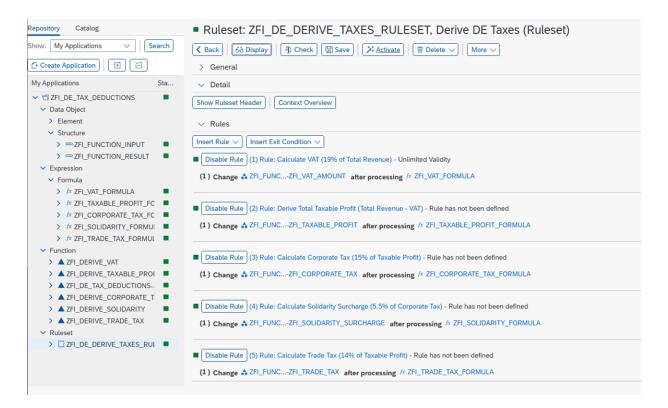
4. Processing Logic (Business Rules)

The system will apply the following rules sequentially through the ruleset ('ZFI_DE_DERIVE_TAXES_RULESET'):

- o Calculate VAT which is 19% of Total Revenue
- o Derive Taxable Profit which is Total Revenue VAT Total Expenses
- o Calculate Corporate Tax which is 15% of Taxable Profit
- Calculate Solidarity Surcharge which is 5.5% of Corporate Tax
- o Calculate Trade Tax which is 14% of Taxable Profit
- 5. Outputs: The function will return results in the output data object:
 - VAT Amount (`ZFI_VAT_AMOUNT`)
 - Taxable Profit (`ZFI_TAXABLE_PROFIT`)
 - Corporate Tax (`ZFI_CORPORATE_TAX`)
 - Solidarity Surcharge (`ZFI_SOLIDARITY_SURCHARGE`)
 - Trade Tax (`ZFI_TRADE_TAX`)



Functions are executed in sequence to derive tax components



How BRF+ are consumed in ABAP programs:

```
Report: ZFI_DE_TAX_CALCULATION
                                                                                                                          REPORT zfi_de_tax_calculation.
                                                                                681
                                                                                                                       ⊕ SELECTION-SCREEN: BEGIN OF BLOCK bl WITH FRAME TITLE TEXT-001.

FARAMETERS: p_revenu TYPE rfi_de_amount,
p_expens TYPE rfi_de_amount.

SELECTION-SCREEN: END OF BLOCK bl.
← ∨ → ∨ | ♥ | Q | 2  ★ ∨ ,

∨ ⑤ Dictionary Objects

∨ ⑤ Structures

                                                                                                                              " Process the Punction and Capture the Result

"Process the Punction and Capture the Result

ONTA(ly_result) = zcl_process brf_app-yee_instance(EXPORTING iy_func_id = 'D0009098958AIFE0A2A90543E4A18A7B')->process(EXPORTING iy_func_id = 'D000998958AIFE0A2A90543E4A18A7B')->process(EXPORTING iy_func_id = 'D000998958AIFE0A2A9B')->process(EXPORTING iy_func_id = 'D000998958AIFE0A2A9B')->process(EXPORTING iy_func_id = 'D000998958AIFE0A2A9B')->process(EXPORTING iy_func_id = 'D000998958AIFE0A2ABB')->process(EXPORTING iy_func_id = 'D000998958AIFE0A2ABB')->process(EXPOR
                           ZFI S FUNCTION INPUT
                           ZFI_S_FUNCTION_RESULT
                  ✓ 🖰 Data Elements
ZFI_DE_AMOUNT
                                                                                                                              WRITE: /10 'German Tax Calculation'.
SKIP.

WRITE: /10 'Total Revenue:', 35 p_revenu,
WRITE: /10 'Total Expense:', 35 p_expens,
             ∨ 🔁 Class Library
                  ✓ Classes

> □ ZCL_PROCESS_BRF_APP
                                                                                                                                                                                                                                                                                                     55 '19% of Total Revenue'.
55 'Total Revenue - VAT Amc
55 '15% of Taxable Profit'.
55 '5.5% of Corporate Tax'.
55 '14% of Taxable Profit'.
            ✓ ☑ Programs

> ☐ ZFI_DE_TAX_CALCULATION
                                         " Function Context Variables are assigned with input values
                                      <fs_total_revenue> = iv_total_revenue.
                                       <fs_total_expense> = iv_total_expense.
 24
                                     ls_name_value-value = lr_zfi_function_input.
 25
 26
                                     cl_fdt_function_process=>move_data_to_data_object( EXPORTING ir_data
                                                                                                                                                                                                                                                                                                                                      = lr_zfi_function_input
 27
                                                                                                                                                                                                                                                                                  iv_function_id = mv_func_id
 28
                                                                                                                                                                                                                                                                                   iv_timestamp = lv_timestamp
 29
                                                                                                                                                                                                                                                                                  iv_data_object = 'DD80909B95BA1FE0A1FF809F07A1CA7B'
                                                                                                                                                                                                                                                                                                                                          = ls_name_value-value ).
 30
                                                                                                                                                                                                                                            IMPORTING er_data
 31
                                      INSERT ls_name_value INTO TABLE lt_name_value.
 33
                                      CLEAR ls_name_value.
```

```
± *$*---
     *$* Process the Function and Capture the Result
37
         38
39
40
41
                                                                        iv_trace_generation = abap_false
                                                                                           = lr_zfi_function_result ).
42
                                                              IMPORTING er_data
         {\tt ASSIGN \ lr\_zfi\_function\_result-} {\tt >* \ TO \ FIELD-SYMBOL} \, ({\tt <fs\_zfi\_function\_result>}) \; .
43
44
45
46
            cl_fdt_function_process=>process( EXPORTING iv_function_id = mv_func_id
                                                                                           " Unique ID of the BRF+ function
                                               iv_timestamp = lv_timestamp

IMPORTING ea_result = <fs_zfi_function_result>
CHANGING ct_name_value = lt_name_value ). " Result Table
47
48
49
50
           CATCH cx_fdt.
51
         ENDTRY.
```

DE Tax Calculations

German Tax Calculation

Total Revenue: 10.000,00 Total Revenue

Total Expense: 5.000,00 Total Expenses

Total VAT: 1.900,00 19% of Total Revenue

Taxable Profit: 3.100,00 Total Revenue - VAT Amount

Corporate Tax: 465,00 15% of Taxable Profit
Solidarity Surcharge: 25,58 5.5% of Corporate Tax
Trade Tax: 434,00 14% of Taxable Profit