

Dynamisches Selektionsbild

In diesem Artikel zeigen wir Ihnen, wie Sie dynamisch ein Selektionsbild erzeugen können und die eingegebenen Werte zur Selektion der Daten nutzen

Beschreibung

In diesem Beispielprogramm wird ein Selektionsbild zur Tabelle VBAK erzeugt und als Popup angezeigt. Der Benutzer kann nach belieben andere Felder ein- oder ausblenden. Aus den Selektionsoptionen zu den Feldern wird eine WHERE-Klausel für den Select-Befehl erzeugt, so dass die Daten einfach gelesen werden können.

Werte übergeben

Soll mehr als ein Wert übergeben werden, so muss die LT_EXPRESSIONS-Tabelle so aufgebaut werden, dass in der ersten Zeile steht

LOGOP = ‚AND‘.

ARITY = Anzahl der zu übergebenden Felder

Danach kommen dann die Feldwerte, die vorgeblendet werden sollen.

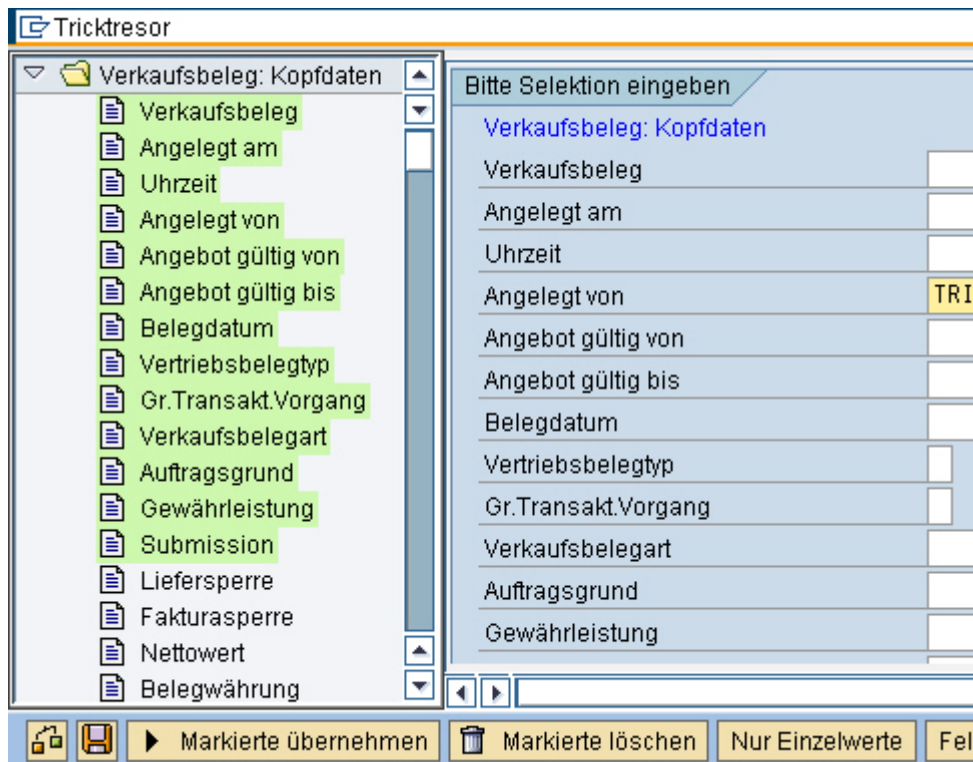
<u>LOGOP</u>	<u>ARITY</u>	<u>FIELDNAME</u>	<u>OPTION</u>	<u>LOW</u>	<u>HIGH</u>
AND	3				
VBELN	BT	1000		2000	
ERDAT	EQ	20070401			
ERNAM	EQ	XK0945			

Hinweis

Zur Erstellung einer WHERE-Bedingung aus RANGES-Tabellen bitte [hier](#) schauen.

Screenshot

Dynamisches Selektionsbild mit Einblendung der Felder



Free Selections Dialog

Coding

REPORT zz_dynsel.Coding

```
*** Type-Pool für "Freie Abgrenzungen"
TYPE-POOLS rsds.
```

DATA:

```
*** Datentabelle
lt_vbak          TYPE STANDARD TABLE OF vbak,
*** Datenstruktur
ls_vbak          TYPE vbak,
*** Selection-ID zur Unterscheidung mehrerer Sets
lv_selection_id LIKE rsdynsel-selid,
*** Anzahl der Felder
lv_number_of_fields TYPE i,
*** Tabellename
lv_tablename     TYPE tablename,
*** Init: Tabellen
lt_tables        TYPE STANDARD TABLE OF rsdstabs,
*** Init: Felder
ls_fields        TYPE                      rsdsfields,
lt_fields        TYPE STANDARD TABLE OF rsdsfields,
*** Daten für Where-Tabelle
lt_where_clauses TYPE rsds_twhere,
ls_where_clause  TYPE rsds_where,
*** Expressions
lt_expressions   TYPE rsds_texpr,
```

```

ls_expression      TYPE rsds_expr,
lt_expr            TYPE rsds_expr_tab,
ls_expr           TYPE rsdsexpr,
*** Ranges
lt_field_ranges   TYPE rsds_trange,
ls_field_ranges   TYPE rsds_range,
lt_range          TYPE rsds_frange_t,
ls_range          TYPE rsds_frange,
lt_selopt        TYPE rsds_selopt_t,
ls_selopt        LIKE rsds_selopt.

```

PARAMETERS:

```

*** Anzeige der Felddauswahl im Tree?
  p_tree AS CHECKBOX DEFAULT 'X',
*** Vorgabe eines Selektionswertes?
  p_vorg AS CHECKBOX DEFAULT 'X'.

```

START-OF-SELECTION.

```

*** Tabellennamen vorgeben
  lv_tablename = 'VBAK'.
*** Tabelle im Dialog bekannt machen:
  APPEND lv_tablename TO lt_tables.
*** einzublendende Felder übergeben
  ls_fields-tablename = lv_tablename.
*** ...Dazu nehmen wir die ersten Felder der Tabelle
  SELECT fieldname FROM dd03l INTO ls_fields-fieldname
     WHERE tabname = lv_tablename
        AND as4local = 'A'
        AND as4vers = 0
        AND position < 15
        AND datatype <> 'CLNT'.
  APPEND ls_fields TO lt_fields.
ENDSELECT.

```

```

IF p_vorg <> space.
*** Name des Erfassers mit SY-UNAME vorbelegen
  ls_expr-fieldname = 'ERNAM'.
  ls_expr-option    = 'EQ'.
  ls_expr-low       = sy-uname.
  APPEND ls_expr TO lt_expr.

  ls_expression-tablename = lv_tablename.
  ls_expression-expr_tab = lt_expr.
  APPEND ls_expression TO lt_expressions.
ENDIF.

```

```

*** "Freie Abgrenzungen" initialisieren
  CALL FUNCTION 'FREE_SELECTIONS_INIT'
    EXPORTING
      kind      = 'T'

```

```

    expressions = lt_expressions
IMPORTING
    selection_id = lv_selection_id
    field_ranges = lt_field_ranges
TABLES
    tables_tab = lt_tables
    fields_tab = lt_fields
EXCEPTIONS
    OTHERS = 20.
IF sy-subrc <> 0.
    WRITE: / sy-subrc.
ELSE.

*** Dialog mit "freien Abgrenzungen" aufrufen
CALL FUNCTION 'FREE_SELECTIONS_DIALOG'
    EXPORTING
        selection_id = lv_selection_id
        title = 'Tricktresor'
        frame_text = 'Bitte Selektion eingeben'
        as_window = 'X'
        start_row = 3
        start_col = 3
        tree_visible = p_tree
        no_frame = 'X'
    IMPORTING
        where_clauses = lt_where_clauses
        expressions = lt_expressions
        field_ranges = lt_field_ranges
        number_of_active_fields = lv_number_of_fields
    TABLES
        fields_tab = lt_fields
    EXCEPTIONS
        internal_error = 1
        no_action = 2
        selid_not_found = 3
        illegal_status = 4
        OTHERS = 5.
IF sy-subrc > 0.
    IF sy-subrc = 2.
*** Meldung "Auswahl wurde abgebrochen"
        MESSAGE s000(00) WITH 'Auswahl wurde abgebrochen'.
    ELSE.
*** Meldung "Fehler bei Funktionsbausteinaufruf"
        MESSAGE i000(00) DISPLAY LIKE 'E'
            WITH 'Fehler bei FREE_SELECTIONS_DIALOG' sy-subrc.
    ENDIF.
ELSE.
*** vom Baustein generierte WHERE-Klausel zur Selektion lesen
    READ TABLE lt_where_clauses INTO ls_where_clause
        WITH KEY tablename = lv_tablename.
*** Selektion durchführen

```

```
SELECT * FROM (lv_tablename) INTO TABLE lt_vbak
  WHERE (ls_where_clause-where_tab).
IF sy-subrc > 0.
  MESSAGE s000(00) WITH 'Keine Daten zur Selektion vorhanden'.
ELSE.
*** ...und ausgeben
  LOOP AT lt_vbak INTO ls_vbak.
    WRITE: / ls_vbak-vbeln.
  ENDLOOP.
ENDIF.
ENDIF.
ENDIF.
```