

❖ EU DSGVO im HCM: Verarbeitung von kundeneigenen Zeitbezügen analog zu HCM_TERMN_DATE



Ausgangssituation und Problemstellung:

Bei der Verwendung/Verarbeitung des Zeitbezuges HCM_TERMN_DATE (Austrittsdatum) im SAP Standard findet eine harte Prüfung auf diesen Zeitbezug statt. Die Intention dieser Prüfung ist, auch die Highdate-Datensätze (31.12.9999) verarbeiten und loggen zu können.

Wenn ein kundeneigener Zeitbezug definiert wird, werden diese Highdate-Datensätze jedoch nicht korrekt verarbeitet/gelogggt. Wir haben für Sie in einem konkreten Beispiel einen kundeneigenen Zeitbezug zum Renteneintritt angelegt. Dieser liefert das Datum des Renteneintritts des Mitarbeiters. Da allerdings nur hart auf das Austrittsdatum (HCM_TERMN_DATE) geprüft wird, konnten zu dem kundeneigenen Zeitbezug die Highdate-Datensätze nicht korrekt verarbeitet/gelogggt werden.

Lösung:

Da es kein passendes BAdI hierfür gibt wurde für die Methode „MODIFY_EXCEEDS_LIMIT_FLAG“ eine Erweiterungsimplementierung angelegt in welcher analog zum HCM_TERMN_DATE nun auch das Z_TERMN_R2 geprüft wird.

Art	Parameter	Typisierung	Beschreibung
▢	EV_TERMINATION_EXISTS	TYPE BOOLE_D	Datenelement zur Domäne BOOLE: TRUE (= 'X') und FALSE (= '')
▢	IS_RET_LOCATION	TYPE IF_LRM_RTP_TYPS=>TY_S_RETENTION_LOCATION	
▢	CV_EXCEEDS_LIMIT_DATE	TYPE BOOLE_D	Datenelement zur Domäne BOOLE: TRUE (= 'X') und FALSE (= '')


```

Methode: MODIFY_EXCEEDS_LIMIT_FLAG aktiv
1  METHOD MODIFY_EXCEEDS_LIMIT_FLAG.
2
3  CLEAR ev_termination_exists.
4
5  IF is_ret_location-s_routing_parameter_time-v_routing_field_name = 'HCM_TERMN_DATE'.
6    CLEAR cv_exceeds_limit_date.
7    ev_termination_exists = 'X'.
8  ENDIF.
9
10 *****
11 *$$$-Start: (1)-----"SE:(1) Klasse CL
12 ENHANCEMENT 1 Z_HR1_ILM_TERMN_R2. "active version-----$$$*
13 *
14 IF is_ret_location-s_routing_parameter_time-v_routing_field_name = 'Z_TERMN_R2'.
15   CLEAR cv_exceeds_limit_date.
16   ev_termination_exists = 'X'.
17 ENDIF.
18 ENHENHANCEMENT.
19 *$$$-End: (1)-----$$$*
20 ENDMETHOD.
    
```

❖ EU DSGVO im HCM: Verarbeitung von kundeneigenen Zeitbezügen analog zu HCM_TERMN_DATE



Betrifft:

Klasse: CL_HRARCH_DESTRUCTION_LOG

Methoden: WRITE_DESTRUCTION_LOG und MODIFY_EXCEEDS_LIMIT_FLAG

Derzeitige Verarbeitung:

In der Klasse CL_HRARCH_DESTRUCTION_LOG Methode MODIFY_EXCEEDS_LIMIT_FLAG wird hart auf HCM_TERMN_DATE geprüft und das Flag von cv exceeds_limit_date entfernt. Das Flag steht auf X wenn es sich um einen Highend Datensatz handelt.

Art	Parameter	Typisierung	Beschreibung
▢	EV_TERMINATION_EXISTS	TYPE BOOLE_D	Datenelement zur Domäne BO
▢	IS_RET_LOCATION	TYPE IF_LRM_RTP_TYSES=>TY_S_RETENTION_LOCATION	
▢	CV_EXCEEDS_LIMIT_DATE	TYPE BOOLE_D	Datenelement zur Domäne BO

Methode	MODIFY_EXCEEDS_LIMIT_FLAG	aktiv
1	METHOD MODIFY_EXCEEDS_LIMIT_FLAG.	
2		
3	CLEAR ev_termination_exists.	
4		
5	IF is_ret_location-s_routing_parameter_time-v_routing_field_name = 'HCM_TERMN_DATE'.	
6	CLEAR cv_exceeds_limit_date.	
7	ev_termination_exists = 'X'.	
8	ENDIF.	
9		
10	ENDMETHOD.	
11		

❖ EU DSGVO im HCM:
Verarbeitung von kundeneigenen Zeitbezügen analog zu HCM_TERMN_DATE



Das Vernichtungslog wird pro Datensatz über Methode WRITE_DESTRUCTION_LOG aufgerufen. Hier findet eine generelle Prüfung auf die Vernichtung (GET_RETENTION_RULES) und die Prüfung auf das Austrittsdatum (MODIFY_EXCEEDS_LIMIT_FLAG) statt.

Art	Parameter	Typisierung	Beschreibung
▶	IV_ENDDA_ADJUSTED	TYPE BOOLE_D	Datenelement zur Domäne BOOLE: TRUE ('X') und FALSE ('')
▶	IV_OBJID	TYPE STRING	Objekt-ID
▶	IV_PROTOCOL	TYPE HRARCH_PROT	Detailliertes Protokoll für die Datenvernichtung
▢	EV_ERROR_TXT	TYPE STRING	Fehlermeldung
▢	EV_REJECT_PERNR	TYPE BOOLE_D	Datenelement zur Domäne BOOLE: TRUE ('X') und FALSE ('')

```

Methode  WRITE_DESTRUCTION_LOG  aktiv
24  CLEAR: ev_error_txt, ev_reject_pernr.
25  lv_endda_adjusted = iv_endda_adjusted.
26
27  TRY.
28  *   Get the Retention Rules.
29  *   This method is common for both preprocessing and write program
30  CALL METHOD me->get_retention_rules
31  IMPORTING
32     es_retention_location = ls_ret_location
33     ev_destructible       = lv_destructible.
34
35  *   Below lines of code were added to reset the endda adjusted flag when
36  *   HCM_TERMN_DATE is used as one of the start time. Reset the endda adjusted
37  *   for starting time termination_date.
38  CALL METHOD modify_exceeds_limit_flag
39  EXPORTING
40     is_ret_location      = ls_ret_location
41  IMPORTING
42     ev_termination_exists = lv_termination_exists
43  CHANGING
44     cv_exceeds_limit_date = lv_endda_adjusted.

```

❖ EU DSGVO im HCM:
Verarbeitung von
kundeneigenen
Zeitbezügen analog zu
HCM_TERMN_DATE



Nur wenn beide Variablen (lv_destructible, lv_endda_adjusted) leer sind wird ein korrekter Logeintrag geschrieben.

```
Methode WRITE_DESTRUCTION_LOG aktiv
135
136 IF lv_destructible = abap_false AND
137    lv_endda_adjusted = abap_false.
138 MESSAGE e039(hrpa_arch) INTO ev_error_txt.
139 CALL METHOD cl_hrarch_protocol_handler=>archive_protocol_collect_gen
140 EXPORTING
141     iv_object = iv_objid
142     iv_msgtype = '2'
143     iv_msgid = sy-msgid
144     iv_msgno = sy-msgno.
145
146 * Destruction criterion not satisfied
147 ELSEIF lv_destructible = abap_false AND
148     lv_endda_adjusted = abap_true.
```