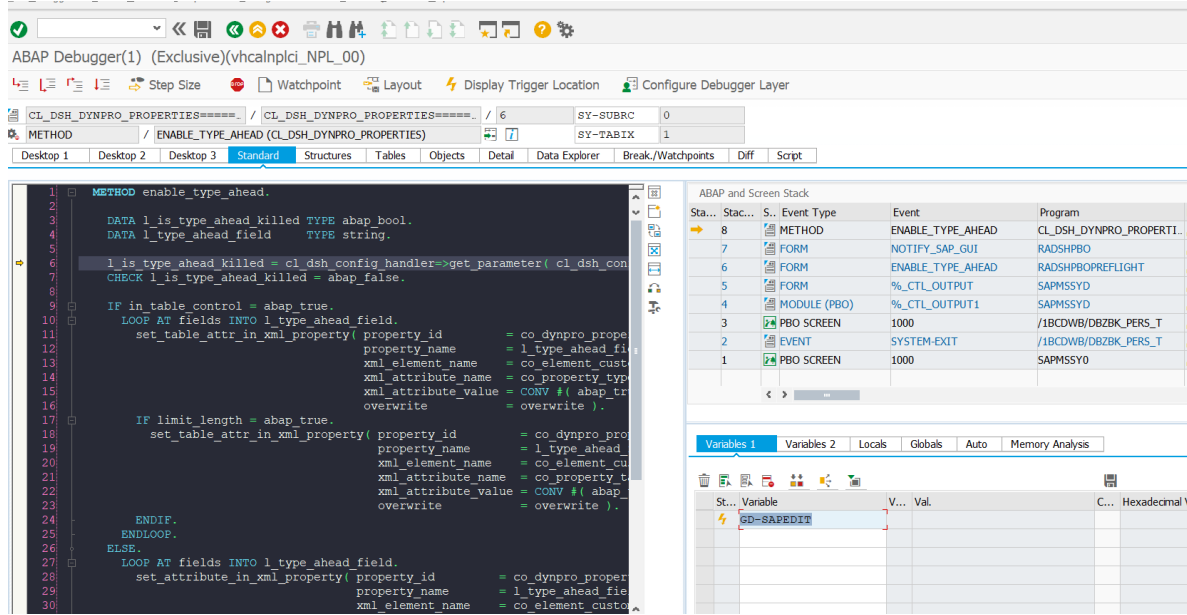


06.03.2023| Gün raporlarına ek



Burada data elementler ve domainlerle beraber bir table oluşturmuş bulunuyorum fakat hiç bilmediğim bir programı baştan aşağı sıfırdan öğrendiğim için birçok hata ile karşılaşmaya devam ediyorum. Örneğin şimdi de tablonun tüm içeriğini düzenledikten sonra se16 ile tekte veri girişi yapabilecek durumda olmam gerekirken bilmediğim bir sorundan dolayı veri girişi yapamıyorum, bunu çözmek için ingilizce kaynaklara bakıyorum fakat halen daha çözüme ulaşamadım. /h ile debugger'a ulaştıktan sonra iki yeni variable ataması yapıp bunların ardından veri eklenebilir hale gelmesi gerekirken şu anda halen olduğum yerde kalmış durumdayım, arayışlarıma devam ediyorum.

16:04 ile gelen son durum:

Konuyu çözmüş bulunmaktayım ve gözümün önünde olan bir sorunu çözmek için bu kadar zaman sarf etmeme inanamıyorum. TCode se16 yaptıktan sonra Data Browser : Initial Screen kısmına gelindiği zaman onun hemen altındaki ikinci kutucuk olan "Create Entries" kısmı ile veri girişi yapılabildiğini görmüş oldum. Böylece verileri girerek tabloyu doldurmaya başlayabildim. Bunun nedeni ise normalde data girişinde se16n kullanılması gerekmekte fakat developer edition'da böyle bir şey mümkün olmadığından dolayı mecburen se16 kullanıyoruz, işte bu nedenden dolayı se16'da direkt olarak seri şekilde veri girişimiz tutorial videoları içerisinde veya makalelerde bulunması biraz zor oluyor sadece deneme yanılma yöntemiyle bazı şeylerin bulunması gerekmektedir. Sonuç olarak sona ulaştım ve sorunu çözüp devam ediyorum.

OPEN SQL

Buradan sonra ise gelen kısım **OPEN SQL** komutları oluyor.

```
REPORT zbk_test.

DATA gv_persid TYPE zbk_persid_de.
DATA gv_persad TYPE zbk_persad_de.
DATA gv_perssoyad TYPE zbk_perssoyad_de.
DATA gv_perscins TYPE zbk_perscins_de.
DATA gs_pers_t TYPE zbk_pers_t.
DATA gt_pers_t TYPE TABLE OF zbk_pers_t.

*SELECT, UPDATE, INSERT, DELETE, MODIFY

☐ *SELECT * FROM ZBK_PERS_T INTO TABLE gt_pers_t WHERE pers_id EQ 1.
| *SELECT SINGLE * FROM ZBK_PERS_T INTO gs_pers_t. "Structure
| *SELECT SINGLE pers_id FROM ZBK_PERS_T INTO gv_persid .

☐ *UPDATE zbk_pers_t SET pers_ad = 'Hakan' WHERE pers_id EQ 1.
| *WRITE: 'Update komutu calistirildi.'.

☐ *gs_pers_t-pers_id = 4.
| *gs_pers_t-pers_ad = 'FURKAN'.
| *gs_pers_t-pers_soyad = 'Söylemez.'.
| *gs_pers_t-pers_cins = 'E'.
| *INSERT zbk_pers_t FROM gs_pers_t.
| *WRITE: 'Insert komutu yururlulukte.'.

☐ *DELETE FROM zbk_pers_t WHERE pers_id EQ 2.
| *WRITE: 'Delete komudu calistirildi.'.

☐ *gs_pers_t-pers_id = 4.
| *gs_pers_t-pers_ad = 'Gulcin'.
| *gs_pers_t-pers_soyad = 'Söylemez.'.
| *gs_pers_t-pers_cins = 'K'.
| *MODIFY zbk_pers_t FROM gs_pers_t.
| *WRITE: 'Modify calisti.'.
```

Onlar da bu şekilde tanımlanıp kullanıldığında işlevselliği sağlanmış oluyor. Tek ss'de **SELECT, UPDATE, INSERT, DELETE VE MODIFY** kullanımı görülmektedir.

KULLANICI VERİ GİRİŞLERİ

Bundan sonra gelecek olan kısım ise "**Kullanıcı Veri Girişleri**" konusu olacak.

Bununla beraber bir program ve table üzerinde dışarıdan kullanıcı veri girişinin nasıl yapılabileceğini de anlamış olacağız.

```

" ***** KULLANICI VERİ GİRİSLERİ ***** "
PARAMETERS: p_num1 TYPE int4,
             p_persad TYPE zbk_persad_de.

DATA: gv_perssoyad TYPE zbk_perssoyad_de.
TABLES: ZBK_PERS_T.
SELECT-OPTIONS: s_perssd for gv_perssoyad. " Burada SELECT-OPTIONS'dan hemen sonra tanımlama 8 karakterden uzun olamaz, bundan dolayı s_perssd verdik.
SELECT-OPTIONS: s_perscn for ZBK_PERS_T-PERS_CINS.

PARAMETERS: p_chox1 AS CHECKBOX.

SELECTION-SCREEN BEGIN OF BLOCK B11 WITH FRAME TITLE text-001.
PARAMETERS: p_rad1 RADIOBUTTON GROUP gr1,
             p_rad2 RADIOBUTTON GROUP gr1,
             p_rad3 RADIOBUTTON GROUP gr1.
SELECTION-SCREEN END OF BLOCK B11.

SELECTION-SCREEN BEGIN OF BLOCK B12 WITH FRAME.
PARAMETERS: p_rad4 RADIOBUTTON GROUP gr2,
             p_rad5 RADIOBUTTON GROUP gr2.
SELECTION-SCREEN END OF BLOCK B12.

```

Şimdi burada parametre girerken örnek olarak PARAMETERS: p_persad verdiğimiz zaman burada görüntü kötülüğü açısından daha güzelleştirmemiz gerekiyor. Çünkü ekranda parametre girişi kısmında yoksa programı kötü gösterecektir. Bunun için üstten **git'e basıyoruz (goto'da yazıyor olabilir.)** ve oradan text elements'e giriş yapıyoruz.

Goto	Utilities	Environment
Attributes	Shift+F9	
Text Elements	F5	
Messages		

Ve buradan giriş yaptıktan sonra ise "**Selection Texts**" kısmından gerekli görünen tanımlamaları yapabiliyoruz.

ABAP Text Elements: Change Selection Texts Language English



Program ZBK_TEST Active

Text Symbols Selection Texts List Headings

Name	Text	DDIC Refere...
P_CBOX1	Üyelik formunu okudum	<input type="checkbox"/>
P_NUM1	Yaşınız:	<input type="checkbox"/>
P_PERSAD	Adınızı giriniz:	<input type="checkbox"/>
P_RAD1	0-18	<input type="checkbox"/>
P_RAD2	18-55	<input type="checkbox"/>
P_RAD3	55+	<input type="checkbox"/>
P_RAD4	ERKEK	<input type="checkbox"/>
P_RAD5	KADIN	<input type="checkbox"/>
S_PERSCN	Cinsiyet	<input type="checkbox"/>
S_PERSSD	Soyad	<input type="checkbox"/>

Görüldüğü gibi bu yanda yazan tanımlamalar ekranda gözükmüş ve kullanıcıyla daha kolay uyumluluk sağlanmış olacaktır.

Bu konuya ek olarak ise;

```
SELECTION-SCREEN BEGIN OF BLOCK B11 WITH FRAME TITLE text-001.  
  PARAMETERS: p_rad1 RADIOBUTTON GROUP gr1,  
              p_rad2 RADIOBUTTON GROUP gr1,  
              p_rad3 RADIOBUTTON GROUP gr1.  
SELECTION-SCREEN END OF BLOCK B11.  
  
SELECTION-SCREEN BEGIN OF BLOCK B12 WITH FRAME TITLE text-002.  
  PARAMETERS: p_rad4 RADIOBUTTON GROUP gr2,  
              p_rad5 RADIOBUTTON GROUP gr2.  
SELECTION-SCREEN END OF BLOCK B12.
```

SELECTION-SCREEN kısmının en sonuna bu gruplandırılmış ve iyi bir estetik olması adına aynı görüntü altında toplanmış duruma bir başlık vermek daha etkili bir seçenek olmuş oluyor. Bu gruba başlığı ise TITLE text-sayıskod şeklinde atama yapabiliyoruz ve bunun yaparken de orada vermiş olduğumuz 001 veya 002 kısmı kendi tercihimize göre vermiş olduğumuz kısımlardan ibaret durumda. Bunları verdikten sonra ise asıl önemli kısım olan yine git/goto kısmından text üzerinden

Text Symbols alanına ulaşarak bu tanımlamış olduğumuz sayısal kodları orada belirtmemiz ve karşılığına da neyin gelmesini istiyorsak bunu yazarak sonuca gitmemiz oluyor.

ABAP Text Elements: Change Text Symbols Language English

Program ZBK_TEST Active

Text Symbols Selection Texts List Headings

Next Free Number

Sym	Text
001	Yaş aralığı
002	Cinsiyet Seçimi

Son hali aşağıdadır;

Yaşınız:

Adınızı giriniz:

Soyad to

Cinsiyet to

☐ Üyelik formunu okudum

Yaş aralığı

☒ 0-18

☐ 18-55

☐ 55+

Cinsiyet Seçimi

☒ ERKEK

☐ KADIN

Ve günün son konusu ise **Event Blocks, Form, Include** konuları;

```
PARAMETERS: p_num TYPE int4.  
  
INITIALIZATION.  
    p_num = 12.
```

P_NUM

Görüldüğü gibi daha bir şey girmeden inputbox'da sabit olarak 12 yazmamızı sağladı.

```
AT SELECTION-SCREEN.  
  p_num = p_num + 1.  
  
END-OF-SELECTION.
```

Bu şekilde yapılıncı ise entera bastıkça sayı artıyor.

Altta görüldüğü şekilde yapıldığı zaman zaten çıkan sonuç görüldüğü gibidir.

```
START-OF-SELECTION.  
  WRITE: 'START OF SELECTION'.  
END-OF-SELECTION.  
  WRITE: 'END OF SELECTION'.
```

START OF SELECTION
END OF SELECTION

FORM - PERFORM KULLANIM ÖRNEKLERİ

```
DATA: gv_num1 TYPE int4.  
START-OF-SELECTION.  
  PERFORM sayiya_bir_ekle.  
  PERFORM sayiya_bir_ekle.  
  PERFORM sayiya_bir_ekle.  
  PERFORM sayiya_bir_ekle.  
  PERFORM sayiya_bir_ekle.  
  WRITE: gv_num1.  
  
END-OF-SELECTION.  
FORM sayiya_bir_ekle.  
  gv_num1 = gv_num1 + 1.  
ENDFORM.
```

Yine örnek bir form kullanım örneği;

```
DATA: gv_num3 TYPE int4.  
DATA: gv_num4 TYPE int4.  
gv_num3 = 22.  
gv_num4 = 36.  
PERFORM sayi_farki USING gv_num3  
                           gv_num4.
```

```
□ *&-----  
  *& Form SAYI_FARKI  
  *&-----  
  *& text  
  *&-----  
  *&      --> GV_NUM3  
  *&      --> GV_NUM4  
  *&-----
```

```
□ FORM sayi_farki USING p_num3  
                           p_num4.  
  DATA: lv_sonuc TYPE int4.  
  lv_sonuc = p_num4 - p_num3.  
  WRITE: 'Sonuc =', lv_sonuc.  
  
  ENDFORM.
```