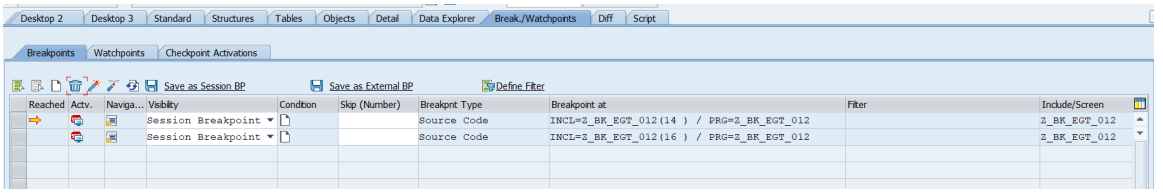
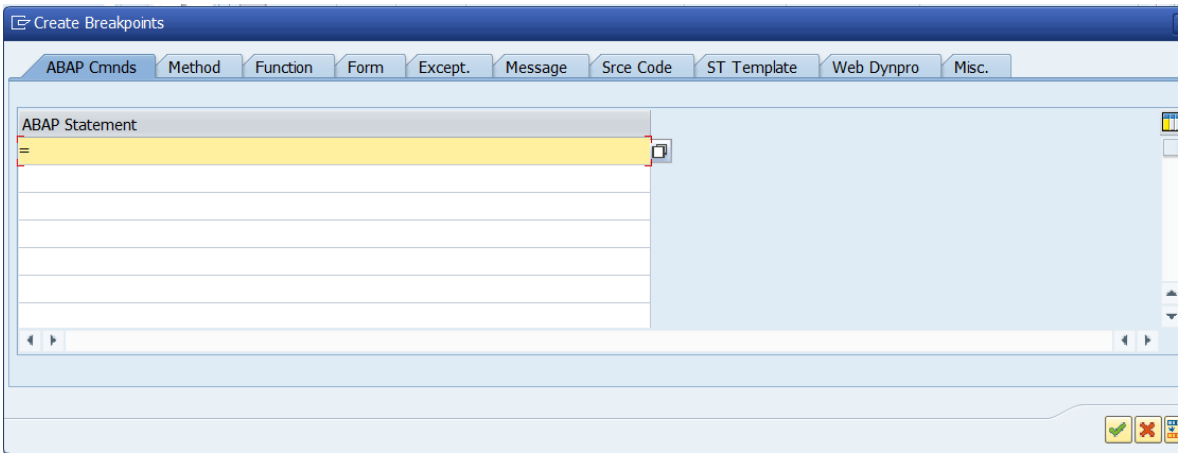
Breakpoint and Watchpoint

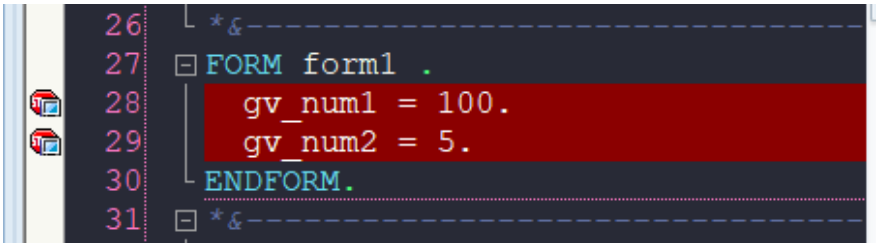
Bu alanda koyduğumuz breakpointleri gösterir.



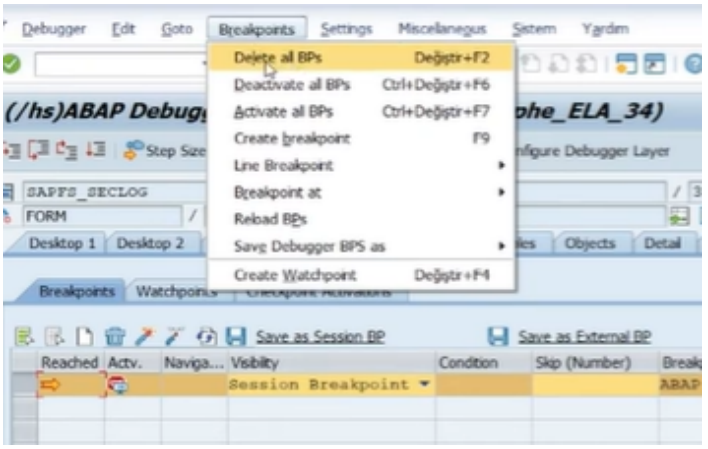
Bu alandan yeni bir breakpointi el ile de atayabiliyoruz direkt sistemden;



örneğin = koyduk ve bu demek ki her = gordugun yerde breakpoint ataması anlamına gelir. Bu şekilde;



Ayrıca ne kadar breakpoint tanımlaması yapıldıysa bile şu şekilde hepsi kaldırılabilir;

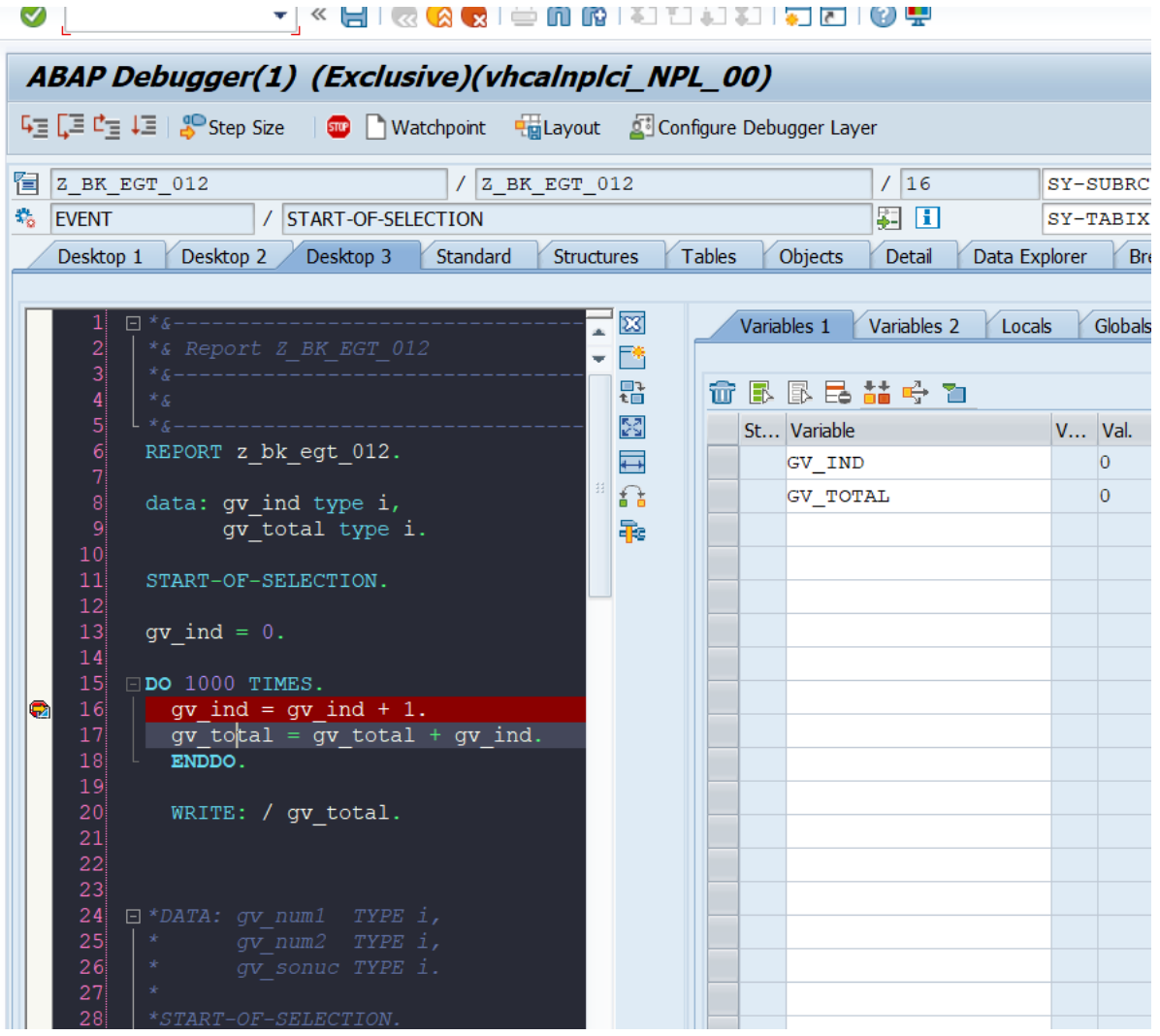


Bu breakpoint alanında genellikle message alanında breakpoint atamak çok elverişlidir. Standart programlarda bir hatanın neden alındığıyla ilgili talepler gelir ve sap'in standart mesajlarını yakalamak açısından bu önemlidir.

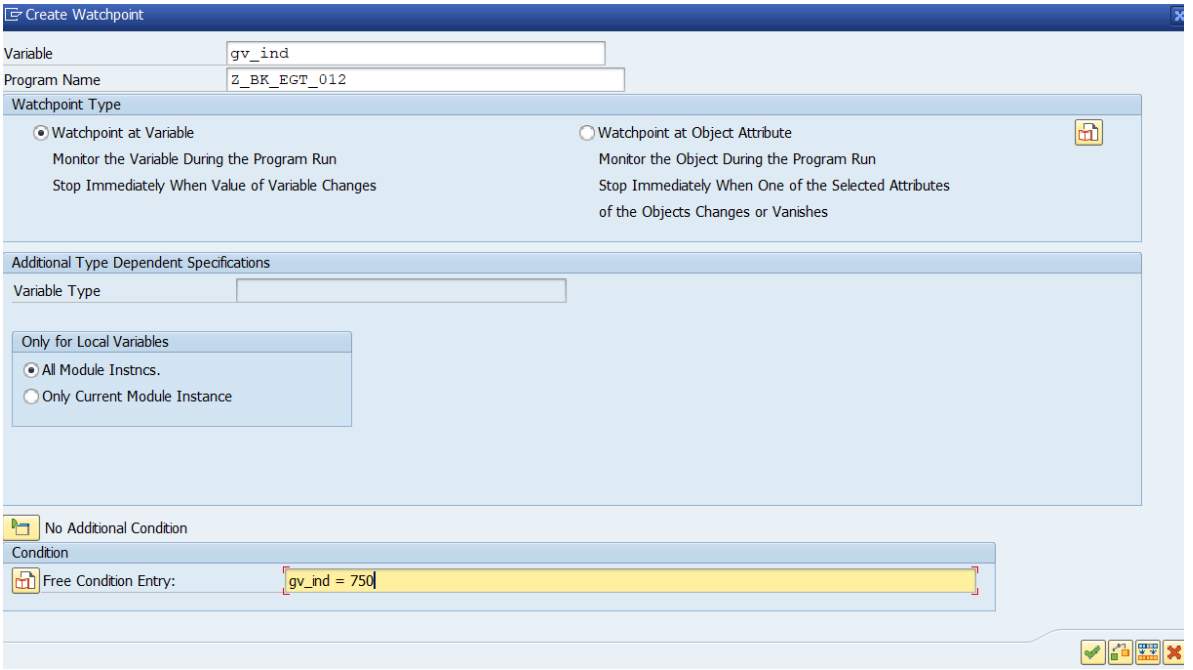
WATCHPOINT

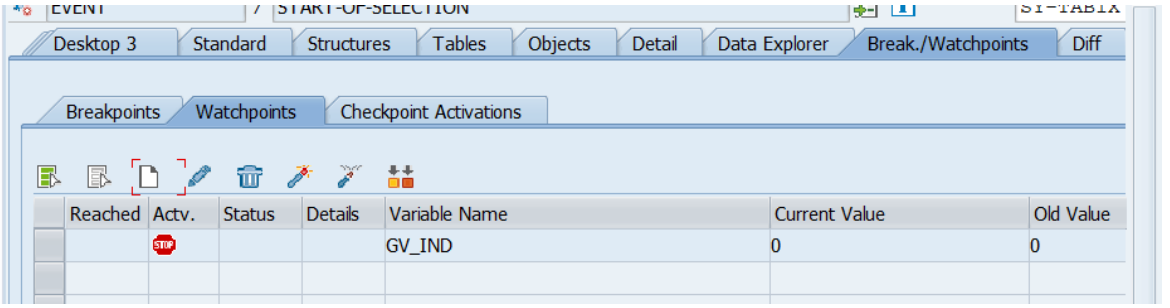
örneğin burada 1000 kere f5'e basinca bitmis olacak. Ve misal 750'deyken ne deger aldigini ogrenmek istiyoruz, bunun için 750 kez f5'e basmamıza gerek yok.

Breakpoint nasıl bir satırda durduruyorsa, Watchpoint ise bir variable'in belirli bir değer aldığında durdurmamızı sağlayan yapıdır.

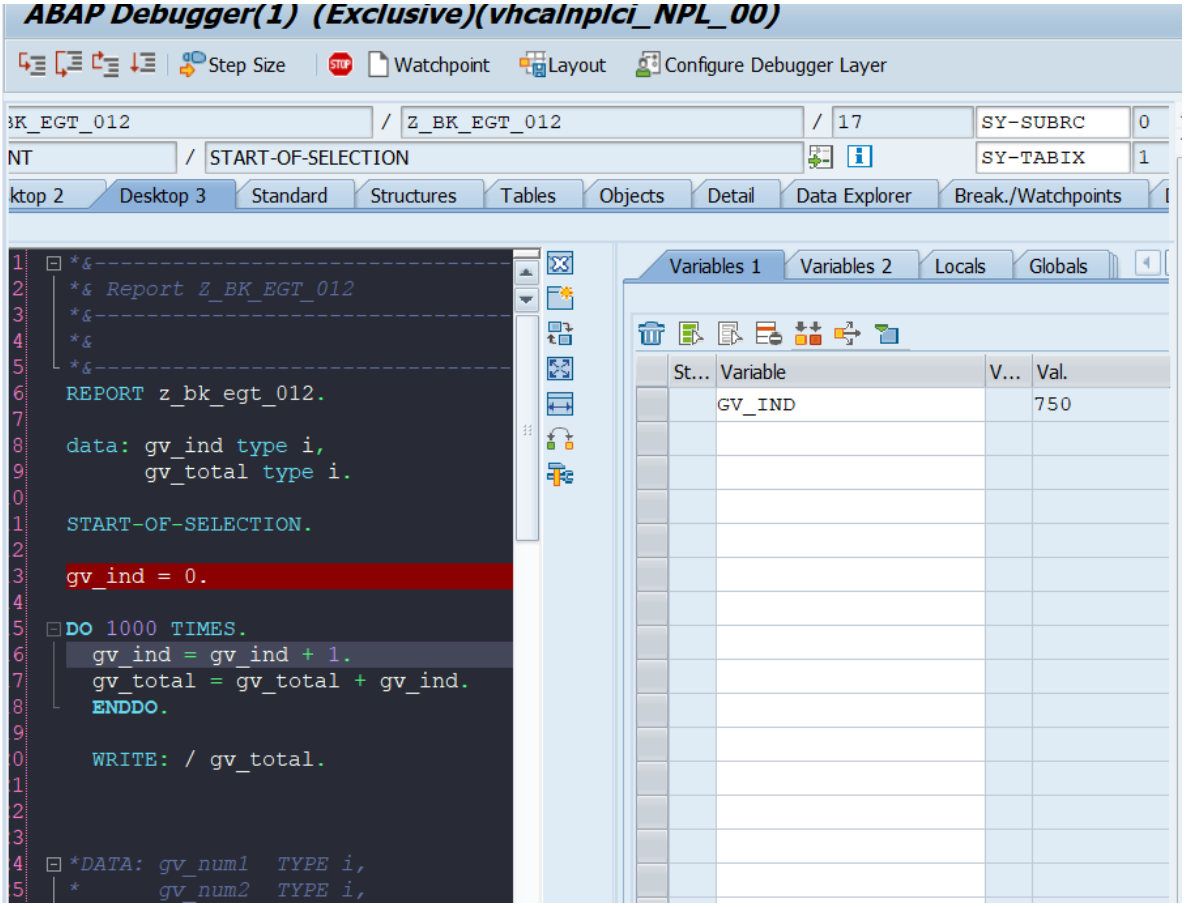


Uygulaması ise şuradaki gibi olacaktır;





Ardından desktop 3'e geldiğimizde ve f8'e bastığımızda 750 kere döngüye girince durdurmuş olup gv_ind değerini bize gösteriyor.


ABAP Debugger(1) (Exclusive)(vhcalnplci_NPL_00)
Step Size
Watchpoint
Layout
Configure Debugger Layer
BK_EGT_012 / Z_BK_EGT_012 / 17 SY-SUBRC 0
NT / START-OF-SELECTION SY-TABIX 1
ktop 2 Desktop 3 Standard Structures Tables Objects Detail Data Explorer Break./Watchpoints
*& Report Z_BK_EGT_012
REPORT z_bk_egt_012.
data: gv_ind type i,
gv_total type i.
START-OF-SELECTION.
gv_ind = 0.
DO 1000 TIMES.
gv_ind = gv_ind + 1.
gv_total = gv_total + gv_ind.
ENDDO.
WRITE: / gv_total.
*DATA: gv_num1 TYPE i,
Variables 1 Variables 2 Locals Globals
St... Variable V... Val.
GV_IND 750