

# Inhalt

1	Report ZZ_ME .....	2
1.1	Allgemeine Beschreibung.....	2
1.2	Das Selektionsbild.....	2
1.3	Einträge in Spalte "Variant / Table" .....	4
1.4	GR55 – Berichtsgruppen .....	4
1.5	Verwendung von Ikonen.....	4
1.6	Funktionstasten .....	5
1.7	Eingabe als Variante sichern.....	6
1.8	Variante für anderen User .....	6
1.9	Berechtigung für das Reporting (SA38).....	6
1.10	Implementierung in einem R/3-System.....	7
1.11	Programmcode .....	7

# 1 Report ZZ\_ME

## 1.1 Allgemeine Beschreibung

Mit dem Report ZZ\_ME, für den sinnvollerweise eine namensgleiche Transaktion eingerichtet werden sollte, kann sich jeder SAP R/3-Anwender die Dinge als Favoriten definieren, die dummerweise im SAP Easy Access nicht vorgesehen sind.

In Erster Linie sind dies Reports, mit oder ohne Varianten. Es können aber auch Transaktionen, Tabellen, View Cluster, Funktionsbausteine oder SAPscript-Formulare sein.

Zum einen kann damit das öffnen und ausführen von Varianten stark vereinfacht werden und zum anderen können so bspw. Tabellen, die nicht so häufig benötigt werden, hinterlegt werden. Denn gerade die Name weniger häufig benötigter Tabellen werden gerne vergessen.

Ist das Bildschirmbild ausgefüllt, muss es lediglich als Variante gespeichert werden. Beim nächsten Aufruf, wird die zuvor gespeicherte Variante dann automatisch geöffnet. Dabei spielt es keine Rolle, wie die Variante genannt wurde.

## 1.2 Das Selektionsbild

Description	Show	Run	Report / Transaction	Variant / Table

Zuerst einmal kann in 5 Kommentarzeilen, persönliche Hinweise zu dieser Variante und deren Verwendung abgelegt werden.

Danach stehen 99 Zeilen zur Verfügung, in die mehr oder weniger beliebiges eingetragen werden kann. Alles was nicht als erlaubte Funktion erkannt wird, wird als Kommentar interpretiert.

Die Namen von Reports und Transaktionen sind in die Spalte einzutragen, die mit "Report / Transaktion" betitelt ist (Leerzeichen sind erlaubt).

Die Spalte, mit dem Titel "Variant / Table", ist dafür vorgesehen, um primär Varianten und Tabellennamen einzutragen.

Nach Betätigung der Enter-Taste wird die Eingabe geprüft.

Für gültige Aktionen werden dann Ampeln aktiviert und die im System hinterlegten Beschreibungen in die Spalte "Description" übernommen.

Eingabe:

Description	Open	Run	Report / Transaction	Variant / Table
			showicon	sort by name
			va43	
			se16	mara
			rfitemar	

Ergebnis nach Enter

Description	Open	Run	Report / Transaction	Variant / Table
Ikonen nach Namen sortiert			SHOWICON	SORT BY NAME
Kontrakt anzeigen			VA43	
Allgemeine Materialdaten			SE16	MARA
Debitoren Einzelpostenliste			RFITEMAR	

Zu allen Einträgen in der Spalte "Report / Transaction" zu denen eine unterstützte Funktion gefunden wurde, zeigt die Spalte "Description" den im System hinterlegten Text und in der Spalte "Open" werden gelbe Ampeln gezeigt, mit der die Aktionen gestartet werden können.

Da zu dem Report SHOWICON eine existierende Variante definiert wurde, wird der Test der Variante angezeigt und in der Spalte "Run" wird eine grüne Ampel gezeigt. Beim Drücken der gelben Ampel wird die Variante geöffnet und beim Drücken der grünen Ampel, wird die Variante direkt ausgeführt.

Alles was der Report nicht als unterstützte Aktion erkennt, wird als Kommentar interpretiert.

### 1.3 Einträge in Spalte "Variant / Table"

Im Falle von Reports und Reporttransaktionen kann hier der Name einer Variante eingetragen werden. Großschreibung ist nicht erforderlich.

Da Variantennamen mit Leerzeichen beginnen können, muss ein Variantename exakt so eingegeben werden, so wie dieser beim Sichern der Variante erfasst wurde.

Im Falle von Tabellen (Transaktionen: SE11, SE16, SE16N, SM30, SM31), Viewcluster (SM34), Funktionsbausteinen (SE37), Reports (SA38, SE38); Screenpainter (SE51), kann das betreffende Objekt mit beliebig vielen Leerzeichen erfasst werden.

### 1.4 GR55 – Berichtsgruppen

Bei Aufruf der Transaktion GR55, muss der Anwender immer erst eine Berichtsgruppe auswählen, bevor dann zu dieser eine Variante öffnen kann. Dies ist etwas umständlich und kann mit der Zeit nervig sein.

Zur Vereinfachung kann der Anwender in ZZ\_ME in die Spalte "Report / Transaktion", deshalb die GR55 gefolgt von der Berichtsgruppe eintragen. Damit die Transaktion GR55 erkannt wird, muss zwischen dieser und der Berichtsgruppe ein Trennzeichen ("=", "-", "\_", ".", ":", ">", "/" oder blanc) gesetzt werden.

Description	Open	Run	Report / Transaction	Variant / Table
Kst.stellengruppe CA-NET			GR55 1SHL	CA-NET

### 1.5 Verwendung von Ikonen

Grundsätzlich ist es im SAP-Standard so, dass sämtliche Texte, sofern diese mit dem 2-stelligen internen Schlüssel einer Ikone beginnen, eingeschlossen vom @-Zeichen, in der Anzeige, automatisch das Bild der Ikone zeigen.



Damit Ikonen sichtbar werden, gibt es die Taste Field toggle display/modify

Um ein Feld in den Anzeige-/Pflegermodus zu wechseln, muss dieses markiert und dann die Taste gedrückt werden.

Nach der Erfassung



Description	Open	Run	Report / Transaction	Variant / Table
@1D@ Aktivitäten vor dem Monatswechsel				
@1A@ Aktivitäten nach dem Monatswechsel				

Nach dem Wechsel in den Anzeigemodus

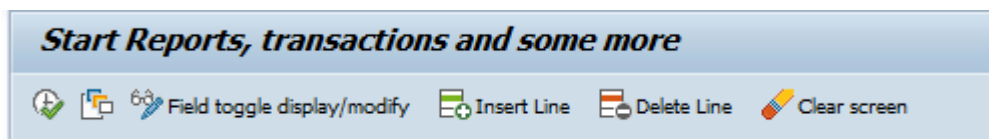
Description	Open	Run	Report / Transaction	Variant / Table
 Aktivitäten vor dem Monatswechsel				
 Aktivitäten nach dem Monatswechsel				

Somit können mit Ikonen Orientierungshilfen geschaffen werden.

Die Kombination von Reports, Transaktionen, Tabellen und Varianten mit Ikonen ist auch möglich!

Description	Open	Run	Report / Transaction	Variant / Table
Anzeigen Debitor (Vertrieb)			 VD03	

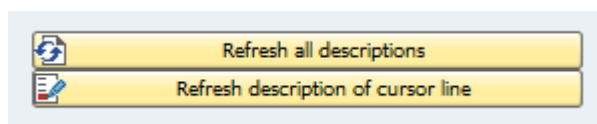
## 1.6 Funktionstasten



Der Wechsel vom Eingabe- in den Anzeigemodus wurde schon behandelt. Darüber hinaus gibt es die Möglichkeit, Leerzeilen einzufügen (Insert Line / Strg+F2), Leerzeilen zu löschen (Delete Line / Strg+F3) oder den Inhalt des gesamten Bildschirmbildes zu löschen (Clear Screen / Strg+F4).

Die Funktion "Clear Screen" ist deshalb sinnvoll, weil ja die vom User erfassten Varianten automatisch gestartet werden, somit alle Zeilen einzeln gelöscht werden müssten.

Innerhalb des Selektionsbilds gibt es noch die Tasten "Refresh all descriptions" und "Refresh description of cursor line".



Mit diesen können einzelne bzw. alle Beschreibungen neu ermittelt werden.

**Achtung:** Bei der Neuermittlung aller Beschreibungen gehen manuelle Ergänzungen verloren. Sofern es sich um Texte handelt, die als Kommentar erkannt wurden, bleiben diese erhalten.

## 1.7 Eingabe als Variante sichern

Nach dem sichern der erfassten Daten als Variante, wird beim nächsten Aufruf diese Variante automatisch geladen.

Hat ein Anwender mehr als eine Variante angelegt, werden diese gelesen um Im Selektionsbild gezeigt.

The screenshot shows the SAP 'Start Reports, transactions and some more' interface. At the top, there are navigation icons and buttons: 'Field toggle display/modify', 'Insert Line', 'Delete Line', and 'Clear screen'. Below this is a 'Comment' field with five input lines. To the right of the comments are two yellow buttons: 'Refresh all descriptions' and 'Refresh description of cursor line'. The main part of the interface is a table with the following columns: 'Description', 'Open', 'Run', 'Report / Transaction', and 'Variant / Table'.

Description	Open	Run	Report / Transaction	Variant / Table
Mtl. Reports am Monatsende			ZZ_ME	HANS MAIER_M01
Mtl. Reports Am Monatsanfang			ZZ_ME	HANS MAIER_M02
Quartalsische Auswertungen			ZZ_ME	HANS MAIER_Q01
wöchentliche Aktivitäten			ZZ_ME	HANS MAIER_W01

Durch das Drücken der gelben Ampeln, kann die gewünschte Variante geöffnet werden.

## 1.8 Variante für anderen User

Möchte beispielsweise eine Supportabteilung für neue Anwender den Report ZZ\_ME benutzen, um diesen die täglich benötigten Funktionen, direkt zur Verfügung zu stellen, Dann muss der Variantename mit "Z:" beginnen und dem Usernamen enden.

Da ein Username nicht mehr als 12 Zeichen lang sein kann und der Variantename eine maximale Länge von 14 hat, kann somit für jeden Usernamen eine Default Variante angelegt werden.

Legt sich ein User später noch eigene Varianten an, dann werden ihm im Startbild immer seine eigenen Varianten gefolgt von der Z:- Variante angezeigt.

Dem User, der für andere User Z:-Varianten angelegt hat, werden diese beim Aufruf von ZZ\_ME nicht gezeigt.

## 1.9 Berechtigung für das Reporting (SA38)

Wenn aus einem Report ein anderer Report gestartet wird, geschieht dies mittels eines SUBMIT. Somit wird dabei die Transaktion SA38 umgangen und auch nicht geprüft, ob der Anwender den Report über die SA38 aufrufen könnte.

Da in vielen Unternehmen inzwischen die meisten User die SA38 nicht mehr ausführen dürfen, würde somit die Berechtigungsprüfung durch ZZ\_ME ausgehebelt werden können.

Um dies zu verhindern, wird geprüft ob der jeweilige User die SA38-Berechtigung hat. Dies geschieht mit dem Befehl:

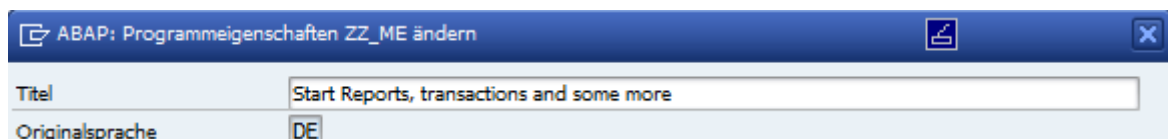
```
AUTHORITY-CHECK OBJECT 'S_TCODE' ID 'TCD' FIELD 'SA38'.
```

Hat der User keine Berechtigung für die SA38, kann er somit nur Reporttransaktionen ausführen.:

### 1.10 Implementierung in einem R/3-System

Ich kann es leider nicht testen, aber mehrfacher Überprüfung und Korrektur sollte der Quellcode auf jedem R/3-System direkt ausführbar sein.

Bis auf den Reporttitel



werden alle anderen Texte direkt im Quellcode zugeordnet. Die aufgerufenen Funktionsbausteine befinden sich alle im SAP-Namensraum und sollten auch auf jedem anderen System verfügbar sein.

Damit der Report ZZ\_ME in SAP Easy Access als Favorit definiert werden kann, sollte diesem mit der Transaktion SE93 eine Transaktion zugeordnet werden.

### 1.11 Programmcode

```
REPORT ZZ_ME.
*=====
*
* Funktion:      Start Transaktion, Submit Reports mit und ohne Varianten
* Beschreibung: Das Selektionsbild dieses Reports wird benutzt, um mit
*               gespeicherten Varianten eine Art von Menue-Liste zu
*               erhalten. Aus dem Selektionsbild können die Varianten
*               geöffnet oder gestartet werden.
*
*               User ohne SA38-Berechtigung können keine Reports ein-
*               tragen. Das Starten von Transaktionen mit Varianten
*               wird auch unterstützt.
*
*               Alles was nicht Transaktion oder Report ist, wird als
*               Kommentar interpretiert.
*
*               Wilfried Schäfer
*               wilfried.schaefer@arcor.de
*=====
* Möglicher Inhalt der Spalte "Variant / Table" abhängig vom Inhalt der
* Spalte "Report/Transaktion"
*
* Report/Transaktion  Variant / Table
```

```

*
*
* SE11 oder
* SE16 oder
* SE16N oder
* SE17 oder
* SE54 oder
* SM30 oder
* SM31 Tabelle
* SM34 View Clusters
* SE37 Funktionsbaustein
* SA38 oder
* SE38 Report
* SE51 Report
* SE71 SAPscript Formular
*
* Reportname Variante
* Transaktionscode Variante
* GR55.Reportgruppe Variante
*
*=====
*
* Aktionstypen: B Berichtsgruppe
* C Kommentar
* H Report mit Variante zur Ausführung im Hintergrund
* h Transaktion mit Variante zur Ausführung im Hintergrund
* R Report
* T Transaktion
* V Report mit Variante / Berichtsgruppe mit Variante
* v Transaktion mit Variante
*
*=====

INCLUDE <LIST>. " List enthält alle icons

TABLES: TRDIR, TRDIRT, " Reports
        TFDIR, " Funktionsbausteine
        TSTC, TSTCT, " SAP-Transaktions-Codes
        TSTCP, " Parameter zu Transaktionen
        DDO2L, DDO2T, " SAP-Tabellen /-Texte
        VARID, VARIT, " Varianten
        SSCRFIELDS, " Felder auf Selektionsbildern
        T803VP, " Report Writer: Generierte Programmversionen
        VCLDIR, VCLDIRT, " Viewcluster.
        STXH, " STXD SAPscript Text-Datei Header
        TFTIT. " Kurztext Funktionsbaustein

*===== Selektionskriterien =====

SELECTION-SCREEN FUNCTION KEY 1.
SELECTION-SCREEN FUNCTION KEY 2.
SELECTION-SCREEN FUNCTION KEY 3.
SELECTION-SCREEN FUNCTION KEY 4.

SELECTION-SCREEN BEGIN OF BLOCK COMMENT_LINES.
* WITH FRAME. " TITLE text_U11.

* 1. Comment line
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN POSITION 01. SELECTION-SCREEN COMMENT (07) TEXT_U11.
SELECTION-SCREEN POSITION 10. PARAMETERS PM_C001(60) LOWER CASE
MODIF ID COM.
SELECTION-SCREEN END OF LINE.

* 2. Comment line
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN POSITION 10. PARAMETERS PM_C002(50) LOWER CASE
MODIF ID COM.
SELECTION-SCREEN POSITION 79.
SELECTION-SCREEN END OF LINE.

```



\* 3. Comment line

```
SELECTION-SCREEN BEGIN OF LINE.
  SELECTION-SCREEN POSITION 10.  PARAMETERS  PM_C003(50) LOWER CASE
                                MODIF ID COM.
SELECTION-SCREEN END OF LINE.
```

\* 4. Comment line

```
SELECTION-SCREEN BEGIN OF LINE.
  SELECTION-SCREEN POSITION 10.  PARAMETERS  PM_C004(50) LOWER CASE
                                MODIF ID COM.

  SELECTION-SCREEN POSITION 68.
  SELECTION-SCREEN PUSHBUTTON (45) PB1_TXT USER-COMMAND UC_REFRESH_COMM
                                MODIF ID PB
                                VISIBLE LENGTH 45.
SELECTION-SCREEN END OF LINE.
```

\* 5. Comment line

```
SELECTION-SCREEN BEGIN OF LINE.
  SELECTION-SCREEN POSITION 10.  PARAMETERS  PM_C005(50) LOWER CASE
                                MODIF ID COM.

SELECTION-SCREEN POSITION 61.
PARAMETERS: P_HIDE01(7) MODIF ID NOD.
SELECTION-SCREEN POSITION 68.
SELECTION-SCREEN PUSHBUTTON (45) PB2_TXT USER-COMMAND
                                UC_REFRESH_LINE_COMM
                                MODIF ID PB
                                VISIBLE LENGTH 45.

SELECTION-SCREEN END OF LINE.
```

```
selection-screen skip.
```

\* Eine Selop, die nicht angezeigt wird, ist notwendig, damit SY-OPTI  
\* abgefragt werden kann.

```
SELECT-OPTIONS PM_VARI FOR VARID-VARIANT MODIF ID NOD.
```

```
SELECTION-SCREEN END OF BLOCK COMMENT_LINES.
```

\*Wenn die Zeile mit "Report / Transaktions" beginnt

```
*SELECTION-SCREEN BEGIN OF LINE.
* SELECTION-SCREEN POSITION 3. SELECTION-SCREEN COMMENT (28) TEXT_U01.
* SELECTION-SCREEN POSITION 34. SELECTION-SCREEN COMMENT (16) TEXT_U02.
* SELECTION-SCREEN POSITION 60. SELECTION-SCREEN COMMENT (04) TEXT_U03.
* SELECTION-SCREEN POSITION 66. SELECTION-SCREEN COMMENT (04) TEXT_U04.
* SELECTION-SCREEN POSITION 72. SELECTION-SCREEN COMMENT (09) TEXT_U12.
* SELECTION-SCREEN POSITION 81.
*SELECTION-SCREEN END OF LINE.
```

\*Wenn die Zeile mit "Description" beginnt

```
SELECTION-SCREEN BEGIN OF LINE.
  SELECTION-SCREEN POSITION 1. SELECTION-SCREEN COMMENT (15) TEXT_U12.
  SELECTION-SCREEN POSITION 46. SELECTION-SCREEN COMMENT (04) TEXT_U03.
  SELECTION-SCREEN POSITION 52. SELECTION-SCREEN COMMENT (04) TEXT_U04.
  SELECTION-SCREEN POSITION 57. SELECTION-SCREEN COMMENT (20) TEXT_U01.
  SELECTION-SCREEN POSITION 83. SELECTION-SCREEN COMMENT (16) TEXT_U02.
SELECTION-SCREEN END OF LINE.
```

\*=====

```
SELECTION-SCREEN BEGIN OF BLOCK ACTION_LINES.
```

```
SELECTION-SCREEN BEGIN OF LINE.
PARAMETERS  P_C_TOGL(10)          MODIF ID ND2.
SELECTION-SCREEN END OF LINE.
```

```
SELECTION-SCREEN BEGIN OF LINE.
PARAMETERS  P_R_TOGL(100)        MODIF ID ND2.
SELECTION-SCREEN END OF LINE.
```

```
SELECTION-SCREEN BEGIN OF LINE.
PARAMETERS  P_V_TOGL(100)        MODIF ID ND2.
```

## ZZ\_ME – Reports und einiges mehr als Favoriten definieren

---

SELECTION-SCREEN END OF LINE.

SELECTION-SCREEN BEGIN OF LINE.  
PARAMETERS P\_I\_TOGL(100)                   MODIF ID ND2.  
SELECTION-SCREEN END OF LINE.

\*=====

TYPES: TEXT40 TYPE C LENGTH 40.

DATA: WK\_P\_V(30),  
      WK\_C40K       LIKE MAKT-MAKTX,  
      WK\_C30K       LIKE DOCCLSST-TEXT30.

\*=====

DEFINE LINE6.  
\*     p\_r\_&1           Report  
\*     p\_v\_&1           Variante  
\*     p\_i\_&1           Information / comment.

\*Wenn die Zeile mit "Report / Transaktions" beginnt  
\*SELECTION-SCREEN BEGIN OF LINE.  
\*SELECTION-SCREEN POSITION 1.   PARAMETERS: P\_R\_&1(40) LOWER CASE  
\*   VISIBLE LENGTH 30  
\*   MODIF ID P\_R.  
\*SELECTION-SCREEN POSITION 32.   PARAMETERS: P\_V\_&1(30) LOWER CASE  
\*   VISIBLE LENGTH 25  
\*   MODIF ID CHK.  
\*SELECTION-SCREEN PUSHBUTTON 58(4) T\_B1&1 USER-COMMAND OPEN\_&1  
\*   MODIF ID OPN  
\*   VISIBLE LENGTH 4.  
\*SELECTION-SCREEN PUSHBUTTON 63(4) T\_B2&1 USER-COMMAND SUBM\_&1  
\*   MODIF ID SUB  
\*   VISIBLE LENGTH 4.  
\*  
\*SELECTION-SCREEN POSITION 70.   PARAMETERS: P\_I\_&1(70) LOWER CASE  
\*   VISIBLE LENGTH 70  
\*   MODIF ID P\_I.  
\*  
\*SELECTION-SCREEN END OF LINE.

\*Wenn die Zeile mit "Description" beginnt  
SELECTION-SCREEN BEGIN OF LINE.  
SELECTION-SCREEN POSITION 1.   PARAMETERS: P\_I\_&1(100) LOWER CASE  
  VISIBLE LENGTH 45  
  MODIF ID P\_I.  
SELECTION-SCREEN PUSHBUTTON 47(4) T\_B1&1 USER-COMMAND OPEN\_&1  
  MODIF ID OPN  
  VISIBLE LENGTH 4.  
SELECTION-SCREEN PUSHBUTTON 52(60) T\_B2&1 USER-COMMAND SUBM\_&1  
  MODIF ID SUB  
  VISIBLE LENGTH 4.  
  
SELECTION-SCREEN POSITION 57.   PARAMETERS: P\_R\_&1(40) LOWER CASE  
  VISIBLE LENGTH 25  
  MODIF ID P\_R.  
SELECTION-SCREEN POSITION 83.   PARAMETERS: P\_V\_&1(30) LOWER CASE  
  VISIBLE LENGTH 30  
  MODIF ID CHK.

SELECTION-SCREEN END OF LINE.

END-OF-DEFINITION.                                   "Line6

\*=====

LINE6 S001. LINE6 S002. LINE6 S003. LINE6 S004. LINE6 S005.  
LINE6 S006. LINE6 S007. LINE6 S008. LINE6 S009. LINE6 S010.  
  
LINE6 S011. LINE6 S012. LINE6 S013. LINE6 S014. LINE6 S015.  
LINE6 S016. LINE6 S017. LINE6 S018. LINE6 S019. LINE6 S020.

## ZZ\_ME – Reports und einiges mehr als Favoriten definieren

---

LINE6 S021. LINE6 S022. LINE6 S023. LINE6 S024. LINE6 S025.  
LINE6 S026. LINE6 S027. LINE6 S028. LINE6 S029. LINE6 S030.  
  
LINE6 S031. LINE6 S032. LINE6 S033. LINE6 S034. LINE6 S035.  
LINE6 S036. LINE6 S037. LINE6 S038. LINE6 S039. LINE6 S040.  
  
LINE6 S041. LINE6 S042. LINE6 S043. LINE6 S044. LINE6 S045.  
LINE6 S046. LINE6 S047. LINE6 S048. LINE6 S049. LINE6 S050.  
  
LINE6 S051. LINE6 S052. LINE6 S053. LINE6 S054. LINE6 S055.  
LINE6 S056. LINE6 S057. LINE6 S058. LINE6 S059. LINE6 S060.  
  
LINE6 S061. LINE6 S062. LINE6 S063. LINE6 S064. LINE6 S065.  
LINE6 S066. LINE6 S067. LINE6 S068. LINE6 S069. LINE6 S070.  
  
LINE6 S071. LINE6 S072. LINE6 S073. LINE6 S074. LINE6 S075.  
LINE6 S076. LINE6 S077. LINE6 S078. LINE6 S079. LINE6 S080.  
  
LINE6 S081. LINE6 S082. LINE6 S083. LINE6 S084. LINE6 S085.  
LINE6 S086. LINE6 S087. LINE6 S088. LINE6 S089. LINE6 S090.  
  
LINE6 S091. LINE6 S092. LINE6 S093. LINE6 S094. LINE6 S095.  
LINE6 S096. LINE6 S097. LINE6 S098. LINE6 S099.

\*=====

DEFINE SINE6.

DATA: S\_R\_&1 TYPE PROGRAMM, "Report  
S\_V\_&1 TYPE VARID-VARIANT, "Variante  
S\_I\_&1(55). "Description

END-OF-DEFINITION.

SINE6 S001. SINE6 S002. SINE6 S003. SINE6 S004. SINE6 S005.  
SINE6 S006. SINE6 S007. SINE6 S008. SINE6 S009. SINE6 S010.  
  
SINE6 S011. SINE6 S012. SINE6 S013. SINE6 S014. SINE6 S015.  
SINE6 S016. SINE6 S017. SINE6 S018. SINE6 S019. SINE6 S020.  
  
SINE6 S021. SINE6 S022. SINE6 S023. SINE6 S024. SINE6 S025.  
SINE6 S026. SINE6 S027. SINE6 S028. SINE6 S029. SINE6 S030.  
  
SINE6 S031. SINE6 S032. SINE6 S033. SINE6 S034. SINE6 S035.  
SINE6 S036. SINE6 S037. SINE6 S038. SINE6 S039. SINE6 S040.  
  
SINE6 S041. SINE6 S042. SINE6 S043. SINE6 S044. SINE6 S045.  
SINE6 S046. SINE6 S047. SINE6 S048. SINE6 S049. SINE6 S050.  
  
SINE6 S051. SINE6 S052. SINE6 S053. SINE6 S054. SINE6 S055.  
SINE6 S056. SINE6 S057. SINE6 S058. SINE6 S059. SINE6 S060.  
  
SINE6 S061. SINE6 S062. SINE6 S063. SINE6 S064. SINE6 S065.  
SINE6 S066. SINE6 S067. SINE6 S068. SINE6 S069. SINE6 S070.  
  
SINE6 S071. SINE6 S072. SINE6 S073. SINE6 S074. SINE6 S075.  
SINE6 S076. SINE6 S077. SINE6 S078. SINE6 S079. SINE6 S080.  
  
SINE6 S081. SINE6 S082. SINE6 S083. SINE6 S084. SINE6 S085.  
SINE6 S086. SINE6 S087. SINE6 S088. SINE6 S089. SINE6 S090.  
  
SINE6 S091. SINE6 S092. SINE6 S093. SINE6 S094. SINE6 S095.  
SINE6 S096. SINE6 S097. SINE6 S098. SINE6 S099.

SELECTION-SCREEN END OF BLOCK ACTION\_LINES.

\*=====

DATA: WK\_FILENAME LIKE RLGRAP-FILENAME,  
WK\_FLENGTH TYPE F,  
WK\_ZAE1(3) TYPE P,  
WK\_DIV\_0 TYPE P,

```
WK_DIV_1          TYPE P,
WK_FIRST_STEP(3) TYPE N  VALUE 1,
WK_LAST_STEP(3)  TYPE N,
WK_MESSAGE(100),
WK_CURSOR_FIELD(20),
WK_FIELD_CONTENT(40),
WK_DYNTXT LIKE SMP_DYNTXT,
WK_MEMCHECK,
WK_ACTIONTYPE TYPE TEXT120,
WK_ZZ_ME        TYPE TEXT120,
WK_VARIANT LIKE VARID-VARIANT,
WK_REFRESH_LINE(3) TYPE N,
WK_NOTHING_TO_DO,
WK_GR55_REPNAME(40),
WK_TSTCP_REPNAME(40),
WK_SA38_ALLOWED,
WK_UCOMM LIKE SY-UCOMM,

WK_ENDE.

DATA: US_REPID LIKE TRDIR-NAME,
      US_VARI  LIKE VARID-VARIANT,
      US_COND,
      US_DATE  LIKE SY-DATUM,
      US_X(2)  TYPE N.

DATA: L_NOD,
      X_OPN TYPE I,
      X_SUB TYPE I,
      FN_REPID(8),           " like trdir-name,
      FN_VARI(8),           " like varid-variant,
      X(3)  TYPE N,

      ENDE.

DATA: BEGIN OF TAB OCCURS 0,
      STEP(2) TYPE N,
      NEXT(3) TYPE N,
      REP LIKE TRDIR-NAME,
      VARI(30),
      COND(1),
      DATE LIKE SY-DATUM,
      INFO(55),
      IGNORE_LINE,
      ERROR_REASON(50),
      END OF TAB.

DATA: BEGIN OF MEMORY_PREREP,
      PREREP LIKE SY-REPID.
DATA: END OF MEMORY_PREREP.

DATA: BEGIN OF MEMORY_INFO,
      STEP_HIERARCHY(40).
DATA: END OF MEMORY_INFO.

DATA: BEGIN OF S_MEM,
      MEMORY(20),
      SERVICE(1),
      2ND_RUN(1),
      EXIT_1(1),
      EXIT_2(1),
      ZVARIANT_DEFAULT LIKE VARID-VARIANT,
      NO_UNAME_SELECT,
      PREV_REP LIKE SY-REPID.
DATA: END OF S_MEM.

DATA: BEGIN OF S_MEM2,
      ZVARIANT_DEFAULT LIKE VARID-VARIANT.
DATA: END OF S_MEM2.

DATA: CDD_SELTAB TYPE TABLE OF RSPARAMS,
```

## ZZ\_ME – Reports und einiges mehr als Favoriten definieren

---

```
CDD_SELTAB_WA LIKE LINE OF CDD_SELTAB.

RANGES R_VARI FOR VARID-VARIANT.

FIELD-SYMBOLS: <FS_1>, <FS_REPID>, <FS_VARI>, <FS_TEXT>, <FS_DATE>.

*===== Aufbereiten Selektionsbild vor 1. Anzeige =====

INITIALIZATION.

if sy-s1set = space.
  PERFORM PRUEF_DEFAULT_VARIANT.
endif.

*Feststellen, ob der User SA38-Berechtigung hat.

AUTHORITY-CHECK OBJECT 'S_TCODE'
                  ID    'TCD' FIELD 'SA38'.
IF SY-SUBRC = 0.
  TEXT_U01 = 'Report / Transaction'.
  WK_SA38_ALLOWED = 'X'.
ELSE.
  TEXT_U01 = 'Transaction'.
ENDIF.

TEXT_U02 = 'Variant / Table'.
TEXT_U03 = 'Open'.
TEXT_U04 = 'Run'.
TEXT_U11 = 'Comment'.
TEXT_U12 = 'Description'.

DATA: LPB1T(8) VALUE 'T_B1S',
      LPB2T(8) VALUE 'T_B2S'.

FIELD-SYMBOLS: <FS_PB1>, <FS_PB2>.

DO 99 TIMES.
  ADD 1 TO X.
  LPB1T+5 = X.
  LPB2T+5 = X.
  ASSIGN (LPB1T) TO <FS_PB1>. <FS_PB1> = '@09@'.
  ASSIGN (LPB2T) TO <FS_PB2>. <FS_PB2> = '@08@'.
ENDDO.

PB1_TXT = '@42@Refresh all descriptions'.
PB2_TXT = '@4K@Refresh description of cursor line'.

CLEAR WK_DYNTXT.
MOVE 'Field toggle display/modify'
  TO WK_DYNTXT-TEXT.
MOVE 'Toggle on cursor field between display/modify mode!'
  TO WK_DYNTXT-QUICKINFO.
MOVE ICON_TOGGLE_DISPLAY_CHANGE
  TO WK_DYNTXT-ICON_ID.
MOVE 'Field toggle display/modify'
  TO WK_DYNTXT-ICON_TEXT.
MOVE WK_DYNTXT TO SSCRFIELDS-FUNCTXT_01.

CLEAR WK_DYNTXT.
MOVE 'Insert Line'
  TO WK_DYNTXT-TEXT.
MOVE 'Insert a line on cursor position'
  TO WK_DYNTXT-QUICKINFO.
MOVE ICON_INSERT_ROW
  TO WK_DYNTXT-ICON_ID.
MOVE 'Insert Line '
  TO WK_DYNTXT-ICON_TEXT.
MOVE WK_DYNTXT TO SSCRFIELDS-FUNCTXT_02.

CLEAR WK_DYNTXT.
MOVE 'Delete Line'
  TO WK_DYNTXT-TEXT.
MOVE 'Delete the line of cursor position'
```

```
    TO WK_DYNTXT-QUICKINFO. "Quickinfo
MOVE  ICON_DELETE_ROW      TO WK_DYNTXT-ICON_ID.
MOVE  'Delete Line '
    TO WK_DYNTXT-ICON_TEXT. "Ikonentext
MOVE  WK_DYNTXT TO SSCRFIELDS-FUNCTXT_03.

CLEAR WK_DYNTXT.
MOVE  'Clear Screen'      TO WK_DYNTXT-TEXT.      "Text
MOVE  'clear Screen '    TO WK_DYNTXT-QUICKINFO. "Quickinfo
MOVE  ICON_ERASE         TO WK_DYNTXT-ICON_ID.
MOVE  'Clear screen '    TO WK_DYNTXT-ICON_TEXT. "Ikonentext
MOVE  WK_DYNTXT TO SSCRFIELDS-FUNCTXT_04.

* LOOP AT SCREEN.
*   IF SCREEN-GROUP1 = 'VIW'.
*       SCREEN-INPUT = 'OFF'.
*       MODIFY SCREEN.
*   ENDIF.
* ENDLOOP.

*=====
FORM PRUEF_DEFAULT_VARIANT.
*=====

IMPORT WK_MEMCHECK      FROM MEMORY ID 'WK_MEMCHECK'.
FREE MEMORY              ID 'WK_MEMCHECK'.
CHECK WK_MEMCHECK = SPACE.

DATA WK_BACK.
IMPORT WK_BACK          FROM MEMORY ID 'WK_BACK'.
IF SY-SLSET <> SPACE.
    FREE MEMORY          ID 'WK_BACK'.
ENDIF.
CHECK WK_BACK = SPACE.

* Gibt es für den User nur eine Variante, dann wird diese gestartet.

DATA: L_COUNT TYPE I,
      L_VARIANT LIKE VARID-VARIANT.

RANGES R_VARIANT FOR VARID-VARIANT.

SELECT * FROM VARID WHERE REPORT = SY-REPID
        AND ENAME = SY-UNAME.

    IF VARID-VARIANT(2) = 'Z:'.
        CHECK VARID-VARIANT+2 = SY-UNAME.
    ENDIF.

    ADD 1 TO L_COUNT.
    L_VARIANT = VARID-VARIANT.

    IF L_COUNT > 1.
        EXIT.
    ENDIF.
ENDSELECT.

*Es kann auch eine Z:USERNAME-Variante geben, die ein anderer
*User angelegt hat.

if l_count = 0.
    concatenate 'Z:' sy-uname into L_VARIANT.
    SELECT single * FROM VARID WHERE REPORT = SY-REPID
        AND variant = l_variant.

    if sy-subrc = 0.
        add 1 to l_count.
    endif.
endif.

IF L_COUNT = 1.
    WK_MEMCHECK = 'X'.

```

## ZZ\_ME – Reports und einiges mehr als Favoriten definieren

---

```
EXPORT WK_MEMCHECK          TO MEMORY ID 'WK_MEMCHECK'.

SUBMIT (SY-REPID) VIA SELECTION-SCREEN
      USING SELECTION-SET L_VARIANT.
ENDIF.

PERFORM CHECK_USER_VARI_EXIST.

ENDFORM.
=====
FORM CHECK_USER_VARI_EXIST.
=====
RANGES R_VARI    FOR VARID-VARIANT.

* Gibt es für diesen Report keine Z:USERNAME-Variante keine
* USERNAME:-Varianten und nicht nur eine Variante, dann wird der
* Sel-Screen mit den vorhandenen Varianten (max. 99) gefüllt.

**DATA: BEGIN OF T_ZVISE OCCURS 0.
**      INCLUDE STRUCTURE ZVISE.
**DATA: END OF T_ZVISE.
**
**DATA: BEGIN OF SELTAB_1 OCCURS 0,
**      TABNAME LIKE ZVISE-TABNAME.
**DATA: END OF SELTAB_1.
**
**DATA: BEGIN OF S_MEM,
**      MEMORY(20),
**      SERVICE(1),
**      2ND_RUN(1),
**      EXIT_1(1),
**      EXIT_2(1),
**      ZVARIANT_DEFAULT LIKE VARID-VARIANT,
**      NO_UNAME_SELECT,
**      PREV_REP         LIKE SY-REPID.
**DATA: END OF S_MEM.

DATA: WK_UNAME LIKE SY-UNAME,
      WK_CLEAR_ENTRY,
      WK_VARI_DEFAULT_LIST.

WK_UNAME = SY-UNAME.

DATA: L_VARI LIKE VARID-VARIANT,
      L_STRLN TYPE I,
      L_COUNT TYPE I,
      l_num3(3) type n,
      l_screen_field_R(20),
      l_screen_field_V(20).

field-symbols <fs_SCREEN_R>.
field-symbols <fs_SCREEN_V>.

CONCATENATE SY-UNAME ':' INTO L_VARI.
L_STRLN = STRLEN( L_VARI ).

CLEAR CDD_SELTAB.
CDD_SELTAB_WA-SELNAME = 'P_REPORT'.
CDD_SELTAB_WA-KIND    = 'S'.
CDD_SELTAB_WA-SIGN    = 'I'.
CDD_SELTAB_WA-OPTION  = 'EQ'.
CDD_SELTAB_WA-LOW     = SY-REPID.

SELECT * FROM VARID WHERE REPORT = SY-REPID
      AND ENAME = SY-UNAME.

IF VARID-VARIANT(2) = 'Z:'.
  CHECK VARID-VARIANT+2 = SY-UNAME.
ENDIF.

ADD 1 TO L_COUNT.
```

\* Hier nun den Sel-Screen mit den vorhandenen Varianten füllen

```
if L_count < 99.
  add 1 to L_num3.
  concatenate 'P_R_S' l_num3 into l_screen_field_R.
  concatenate 'P_V_S' l_num3 into l_screen_field_V.
  assign (l_screen_field_r) to <fs_Screen_R>.
  assign (l_screen_field_V) to <fs_Screen_V>.

  <fs_Screen_R> = varid-report.
  <fs_Screen_V> = varid-variant.
endif.
ENDSELECT.
```

\*Jetzt noch die Z:USERNAME-Variante, die von einem anderen User  
\*angelegt wurde.

```
if l_count < 99.
  concatenate 'Z:' sy-uname into l_vari.

  SELECT single * FROM VARID WHERE REPORT = SY-REPID
                                AND ENAME  <> SY-UNAME
                                and variant = L_vari.

  if sy-subrc = 0.
    ADD 1 TO L_COUNT.
    add 1 to L_num3.
    concatenate 'P_R_S' l_num3 into l_screen_field_R.
    concatenate 'P_V_S' l_num3 into l_screen_field_V.
    assign (l_screen_field_r) to <fs_Screen_R>.
    assign (l_screen_field_V) to <fs_Screen_V>.

    <fs_Screen_R> = varid-report.
    <fs_Screen_V> = varid-variant.
  endif.
endif.

ENDFORM.
```

\*===== Aufbereiten Selektionsbild vor jeder Anzeige =====  
AT SELECTION-SCREEN OUTPUT.

```
*=====
DATA: L_NOD,
      X_OPN TYPE I,
      X_SUB TYPE I,
      FN_REPID(8),           " like trdir-name,
      FN_VARI(8),           " like varid-variant,
      FN_INFO(8),
      X(3) TYPE N,
      L_POS(3) TYPE N,
      L_B1(8) VALUE 'T_B1S000',
      L_B2(8) VALUE 'T_B2S000',
      N3(3) TYPE N,

      ENDE.
```

FIELD-SYMBOLS: <FS\_1>, <FS\_REPID>, <FS\_VARI>, <FS\_TEXT>, <FS\_DATE>,  
<FS\_B1>, <FS\_B2>, <FS\_INFO>.

\*Feststellen, ob der User SA38-Berechtigung hat.

```
AUTHORITY-CHECK OBJECT 'S_TCODE'
                  ID    'TCD' FIELD 'SA38'.
IF SY-SUBRC = 0.
  TEXT_U01 = 'Report / Transaction'.
  WK_SA38_ALLOWED = 'X'.
ELSE.
  TEXT_U01 = 'Transaction'.
ENDIF.

LOOP AT SCREEN.
```



```
IF P_HIDE01 <> 'DISPLAY'.
  IF SCREEN-NAME = 'C2_DIS' OR
     SCREEN-NAME = 'C4_MOD'.
     SCREEN-INPUT = '0'.
     SCREEN-OUTPUT = '0'.
     SCREEN-INVISIBLE = '1'.
     MODIFY SCREEN.
  ENDIF.
ENDIF.

IF SCREEN-NAME BETWEEN 'PM_C001' AND 'PM_C199'.
  L_POS = SCREEN-NAME+5.
  IF P_C_TOGL+L_POS(1) <> ' '.
     SCREEN-INPUT = '0'.
     MODIFY SCREEN.
  ENDIF.

  FN_INFO = SCREEN-NAME. ASSIGN (FN_INFO) TO <FS_INFO>.
  IF <FS_INFO>+0(1) = '@' AND
     <FS_INFO>+3(1) = '@'.
     TRANSLATE <FS_INFO>+1(2) TO UPPER CASE.
  ENDIF.

ENDIF.

IF SCREEN-NAME BETWEEN 'P_R_S001' AND 'P_R_S199'.
  L_POS = SCREEN-NAME+6.
  IF P_R_TOGL+L_POS(1) <> ' '.
     SCREEN-INPUT = '0'.
     MODIFY SCREEN.
  ENDIF.
ENDIF.

IF SCREEN-NAME BETWEEN 'P_V_S001' AND 'P_V_S199'.
  L_POS = SCREEN-NAME+6.
  IF P_V_TOGL+L_POS(1) <> ' '.
     SCREEN-INPUT = '0'.
     MODIFY SCREEN.
  ENDIF.
ENDIF.

IF SCREEN-NAME BETWEEN 'P_I_S001' AND 'P_I_S199'.
  L_POS = SCREEN-NAME+6.
  IF P_I_TOGL+L_POS(1) <> ' '.
     SCREEN-INPUT = '0'.
     MODIFY SCREEN.
  ENDIF.
  FN_INFO = SCREEN-NAME. ASSIGN (FN_INFO) TO <FS_INFO>.
  IF <FS_INFO>+0(1) = '@' AND
     <FS_INFO>+3(1) = '@'.
     TRANSLATE <FS_INFO>+1(2) TO UPPER CASE.
  ENDIF.
ENDIF.

IF SCREEN-GROUP1 = 'CHK'.
  FN_VARI = SCREEN-NAME. ASSIGN (FN_VARI) TO <FS_VARI>.
  FN_REPID = FN_VARI.
  FN_REPID+2(1) = 'R'. ASSIGN (FN_REPID) TO <FS_REPID>.

  IF <FS_VARI>+0(1) = '@' AND
     <FS_VARI>+3(1) = '@'.
     TRANSLATE <FS_VARI>+1(2) TO UPPER CASE.
  ENDIF.

  X = FN_VARI.

  IF <FS_REPID> = SPACE.
     WK_ACTIONTYPE+X(1) = 'C'.
  ELSE.
     PERFORM PRUEF_STEP USING X <FS_REPID> <FS_VARI>
```

```

                                FN_REPID  FN_VARI.
ENDIF.
ENDIF.
* IF SCREEN-GROUP1      = 'VIW'.
*   SCREEN-INPUT       = 'OFF'.
*   MODIFY SCREEN.
* ENDIF.

IF SCREEN-GROUP1      = 'NOD'.
  SCREEN-INVISIBLE    = '1'.
  SCREEN-INPUT       = '0'.
  SCREEN-OUTPUT      = '0'.
  MODIFY SCREEN.
ENDIF.

IF SCREEN-GROUP1      = 'ND2'.
  SCREEN-INVISIBLE    = '1'.
  SCREEN-INPUT       = '0'.
  SCREEN-OUTPUT      = '0'.
  MODIFY SCREEN.
ENDIF.

IF SCREEN-GROUP1      = 'PB3'.
  SCREEN-INVISIBLE    = '1'.
  SCREEN-INPUT       = '0'.
  SCREEN-OUTPUT      = '0'.
  MODIFY SCREEN.
ENDIF.
ENDLOOP.

x_opn = 0.
x_sub = 0.

loop at screen.

  IF SCREEN-GROUP1 = 'CHK'.
    FN_VARI      = SCREEN-NAME.  ASSIGN (FN_VARI) TO <FS_VARI>.
    FN_REPID     = FN_VARI.
    FN_REPID+2(1) = 'R'.        ASSIGN (FN_REPID) TO <FS_REPID>.
  ENDIF.

  IF SCREEN-GROUP1 = 'OPN'.
    ADD 1 TO X_OPN.

    IF WK_ACTIONTYPE+X_OPN(1) = 'C' or
       WK_ACTIONTYPE+X_OPN(1) = ' '.
      SCREEN-INVISIBLE = '1'.
      SCREEN-INPUT     = '0'.
      SCREEN-OUTPUT    = '0'.
      MODIFY SCREEN.
    else.
      N3 = X_opn.
      L_B1+5 = N3.
      ASSIGN (L_B1) TO <FS_B1>.
      <FS_B1> = '@09@'.
      modify screen.
    ENDIF.
  ENDIF.

  IF SCREEN-GROUP1 = 'SUB'.
    ADD 1 TO X_SUB.

    N3 = X_SUB.
    L_B2+5 = N3.
    ASSIGN (L_B2) TO <FS_B2>.

    IF WK_ACTIONTYPE+X_SUB(1) = 'H' OR
       WK_ACTIONTYPE+X_SUB(1) = 'h'.
      <FS_B2> = '@M4@'.
      <FS_B2>+5

```

```
        = 'Variant has flag "Only for Background Processing"'.
    MODIFY SCREEN.
ELSE.
    <FS_B2> = '@08@'.
    MODIFY SCREEN.
ENDIF.

IF WK_ACTIONTYPE+X_SUB(1) = 'v' OR
   WK_ACTIONTYPE+X_SUB(1) = 'V' OR
   WK_ACTIONTYPE+X_SUB(1) = 'H' OR
   WK_ACTIONTYPE+X_SUB(1) = 'h'.
else.
    SCREEN-INVISIBLE = '1'.
    SCREEN-INPUT     = '0'.
    SCREEN-OUTPUT    = '0'.
    MODIFY SCREEN.
ENDIF.

IF wk_zz_me+X_SUB(1) = 'X'.
    SCREEN-INVISIBLE = '1'.
    SCREEN-INPUT     = '0'.
    SCREEN-OUTPUT    = '0'.
    MODIFY SCREEN.
ENDIF.
ENDIF.
endloop.

IF WK_UCOMM = 'UC_ZZ_DIS_MODE' OR
   P_HIDE01 = 'DISPLAY'.
   P_HIDE01 = 'DISPLAY'.
   MODIFY SCREEN.
   PERFORM SWITCH_TO_DISPLAY_MODE.
ENDIF.

IF WK_UCOMM = 'UC_ZZ_MOD_MODE' OR
   P_HIDE01 = 'MODIFY' OR
   P_HIDE01 = SPACE.
   P_HIDE01 = 'MODIFY'.
   MODIFY SCREEN.
   PERFORM SWITCH_TO_MODIFY_MODE.
ENDIF.

IF P_HIDE01 <> 'DISPLAY'.
LOOP AT SCREEN.
    CHECK SCREEN-NAME(4) = 'PM_C'
    OR   SCREEN-NAME(4) = 'P_R_'
    OR   SCREEN-NAME(4) = 'P_V_'
    OR   SCREEN-NAME(4) = 'P_I_'.
    L_POS = SCREEN-NAME+5.

CASE SCREEN-NAME(4).
    WHEN 'PM_C'. IF P_C_TOGL+L_POS(1) = SPACE.
        SCREEN-INPUT = '1'.
        ELSE. SCREEN-INPUT = '0'.
        SCREEN-DISPLAY_3D = '1'.
        ENDIF.
    WHEN 'P_R_'. IF P_R_TOGL+L_POS(1) = SPACE.
        SCREEN-INPUT = '1'.
        ELSE. SCREEN-INPUT = '0'.
        SCREEN-DISPLAY_3D = '1'.
        ENDIF.
    WHEN 'P_V_'. IF P_V_TOGL+L_POS(1) = SPACE.
        SCREEN-INPUT = '1'.
        ELSE. SCREEN-INPUT = '0'.
        SCREEN-DISPLAY_3D = '1'.
        ENDIF.
    WHEN 'P_I_'. IF P_I_TOGL+L_POS(1) = SPACE.
        SCREEN-INPUT = '1'.
        ELSE. SCREEN-INPUT = '0'.
        SCREEN-DISPLAY_3D = '1'.
        ENDIF.
ENDCASE.
ENDLOOP.
ENDIF.
```

```
    ENDCASE.
  MODIFY SCREEN.
ENDLOOP.
ENDIF.

PERFORM GET_COMMENTS.

*=====
FORM SWITCH_TO_DISPLAY_MODE.
*=====

LOOP AT SCREEN.
  IF SCREEN-GROUP1 = 'PBM'.
    SCREEN-INPUT      = '1'.
    MODIFY SCREEN.
  ENDIF.
  CHECK SCREEN-GROUP1 = 'P_R'
    OR SCREEN-GROUP1 = 'P_I'
    OR SCREEN-GROUP1 = 'CHK'
    OR SCREEN-GROUP1 = 'COM'
    OR SCREEN-GROUP1 = 'PBD'
    OR SCREEN-GROUP1 = 'PB'.

  SCREEN-INPUT      = '0'.
  SCREEN-INTENSIFIED = '0'.
  SCREEN-DISPLAY_3D = '1'.
  MODIFY SCREEN.
ENDLOOP.

ENDFORM.
*=====
FORM SWITCH_TO_MODIFY_MODE.
*=====

LOOP AT SCREEN.
  IF SCREEN-GROUP1 = 'PBM'.
    SCREEN-INPUT      = '0'.
    MODIFY SCREEN.
  ENDIF.
  CHECK SCREEN-GROUP1 = 'P_R'
    OR SCREEN-GROUP1 = 'P_I'
    OR SCREEN-GROUP1 = 'CHK'
    OR SCREEN-GROUP1 = 'COM'
    OR SCREEN-GROUP1 = 'PBD'
    OR SCREEN-GROUP1 = 'PB'.
  SCREEN-INPUT      = '1'.
  SCREEN-INTENSIFIED = '0'.
  SCREEN-DISPLAY_3D = '0'.
  MODIFY SCREEN.
ENDLOOP.

ENDFORM.
*=====
FORM GET_COMMENTS.
*=====

DATA: FN_TEXT(8),
      L_REPID      LIKE SY-REPID,
      LS_REPID     TYPE PROGRAM,
      LS_VARI(30),
      L_ACTIONTYPE.

data: begin of t_Varid,
      report like varid-report,
      variant like varid-variant,
      mlangu like varid-mlangu,
      langu like VARIT-langu,
      vtext like varit-vtext.
data  end of t_varid.

data: begin of t_Varid_trans,
      tcode  like tstc-tcode,
```

```
report like varid-report,
variant like varid-variant,
mlangu like varid-mlangu,
langu like VARIT-langu,
vtext like varit-vtext.
data end of t_varid_trans.

data: lt_varid like table of t_varid,
lt_varid_trans like table of t_varid_trans,
ll_varid like t_varid,
ll_varid_trans like t_varid_trans.

CLEAR X.

DO 99 TIMES.
ADD 1 TO X.

CONCATENATE 'P_V_S' X INTO FN_VARI. ASSIGN (FN_VARI) TO <FS_VARI>.
FN_REPID = FN_VARI.
FN_REPID+2(1) = 'R'. ASSIGN (FN_REPID) TO <FS_REPID>.

CHECK <FS_REPID> <> SPACE.

FN_TEXT = FN_VARI.
FN_TEXT+2(1) = 'I'. ASSIGN (FN_TEXT) TO <FS_TEXT>.

IF SY-UCOMM = 'UC_REFRESH_COMM'.
CASE WK_ACTIONTYPE+X(1).
WHEN 'V'
OR 'v'
OR 'R'
OR 'T'
OR 'G'
OR 'H'
OR 'h'. CLEAR <FS_TEXT>.
ENDCASE.
ENDIF.
IF SY-UCOMM = 'UC_REFRESH_LINE_COMM'.
CHECK WK_REFRESH_LINE = X.
CASE WK_ACTIONTYPE+X(1).
WHEN 'V'
OR 'v'
OR 'R'
OR 'T'
OR 'G'
OR 'H'
OR 'h'. CLEAR <FS_TEXT>.
WHEN OTHERS. WK_NOTHING_TO_DO = 'X'.
EXIT.
ENDCASE.
ENDIF.

CHECK <FS_TEXT> = SPACE.

LS_REPID = <FS_REPID>.
IF LS_REPID(5) = 'GR55 '.
LS_REPID(5) = 'GR55.'.
CONDENSE LS_REPID+5 NO-GAPS.
ELSE.
CONDENSE LS_REPID NO-GAPS.
ENDIF.
TRANSLATE LS_REPID TO UPPER CASE.
IF LS_REPID(1) = '@' AND
LS_REPID+3(1) = '@'.
LS_REPID = LS_REPID+4.
ENDIF.

LS_VARI = <FS_VARI>.
TRANSLATE LS_VARI TO UPPER CASE.

CASE WK_ACTIONTYPE+X(1).
```

```
WHEN 'G'. REFRESH LT_VARID.
CLEAR t_VARID.
L_REPID = LS_REPID.
CLEAR WK_TSTCP_REPNAME.
PERFORM GET_TSTCP_REPORTNAME USING 1 LS_REPID LS_VARI
                                     LS_REPID LS_VARI.

L_REPID = WK_TSTCP_REPNAME.

DO 2 TIMES.
  select A~report A~Variant A~mlangu b~langu b~vtext
  appending table lt_varid
  from Varid as A
  inner join VARIT as B on A~report = b~report
                        and a~variant = b~variant
  where a~report = l_repid
        and a~variant = ls_vari.

  IF SY-DBCNT > 1.
    SORT LT_VARID BY MLANGU DESCENDING LANGU.
    LOOP AT LT_VARID INTO t_VARID.
      if sy-langu = t_varid-langu.
        clear t_varid-langu.
        modify lt_varid from t_varid.
      endif.
    ENDLOOP.

    SORT LT_VARID BY langu MLANGU.
  ENDIF.

  LOOP AT LT_VARID INTO t_VARID.
    EXIT.
  ENDLOOP.

  IF t_VARID-VTEXT <> SPACE.
    WRITE t_VARID-VTEXT TO <FS_TEXT>.
    EXIT.
  ENDIF.

  IF LS_VARI+0(1) = '@' AND
     LS_VARI+3(1) = '@'.
    LS_VARI = LS_VARI+4.
  ELSE.
    EXIT.
  ENDIF.
ENDDO.

WHEN 'V'
OR 'H'. REFRESH LT_VARID.
CLEAR t_VARID.
L_REPID = LS_REPID.
IF L_REPID(5) = 'GR55=' OR
   L_REPID(5) = 'GR55-' OR
   L_REPID(5) = 'GR55_' OR
   L_REPID(5) = 'GR55.' OR
   L_REPID(5) = 'GR55,' OR
   L_REPID(5) = 'GR55>' OR
   L_REPID(5) = 'GR55/'.
  CLEAR WK_GR55_REPNAME.
  L_ACTIONTYPE = WK_ACTIONTYPE+X(1).
  PERFORM GET_GR55_REPORTNAME USING 1 LS_REPID LS_VARI
                                     LS_REPID LS_VARI.

  L_REPID = WK_GR55_REPNAME.
  WK_ACTIONTYPE+X(1) = L_ACTIONTYPE.
ENDIF.

DO 2 TIMES.
  select A~report A~Variant A~mlangu b~langu b~vtext
  appending table lt_varid
  from Varid as A
  inner join VARIT as B on A~report = b~report
                        and a~variant = b~variant
```

```

        where a~report = l_repid
              and a~variant = ls_vari.

IF SY-DBCNT > 1.
  SORT LT_VARID BY MLANGU DESCENDING LANGU.
  LOOP AT LT_VARID INTO t_VARID.
    if sy-langu = t_varid-langu.
      clear t_varid-langu.
      modify lt_varid from t_varid.
    endif.
  ENDLOOP.

  SORT LT_VARID BY langu MLANGU.
ENDIF.

LOOP AT LT_VARID INTO t_VARID.
  EXIT.
ENDLOOP.

IF t_VARID-VTEXT <> SPACE.
  WRITE t_VARID-VTEXT TO <FS_TEXT>.
  EXIT.
ENDIF.

IF LS_VARI+0(1) = '@' AND
   LS_VARI+3(1) = '@'.
  LS_VARI = LS_VARI+4.
ELSE.
  EXIT.
ENDIF.
ENDDO.

WHEN 'v'
OR 'h'. REFRESH LT_VARID_TRANS.
L_REPID = LS_REPID.
IF     LS_REPID(3) = '/T/'.   L_REPID = LS_REPID+3.
ELSE.   L_REPID = LS_REPID.
ENDIF.

DO 2 TIMES.
  select t~tcode A~report A~Variant A~mlangu b~langu b~vtext
  appending table lt_varid_trans
  from tstc as t
  inner join varid as a on t~pgmna = a~report
  inner join VARIT as B on A~report = b~report
  and a~variant = b~variant
  where t~tcode = l_repid
  and a~variant = ls_vari.

IF SY-DBCNT > 1.
  SORT LT_VARID_trans BY MLANGU DESCENDING LANGU.
  LOOP AT LT_VARID_trans INTO t_VARID_trans.
    if sy-langu = t_varid_trans-langu.
      clear t_varid_trans-langu.
      modify lt_varid_trans from t_varid_trans.
    endif.
  ENDLOOP.

  SORT LT_VARID_trans BY langu MLANGU.
ENDIF.

LOOP AT LT_VARID_trans INTO t_VARID_trans.
  EXIT.
ENDLOOP.

IF t_VARID_TRANS-VTEXT <> SPACE.
  WRITE t_VARID_TRANS-VTEXT TO <FS_TEXT>.
  EXIT.
ENDIF.

IF LS_VARI+0(1) = '@' AND

```

```

        LS_VARI+3(1) = '@'.
        LS_VARI = LS_VARI+4.
    ELSE.
        EXIT.
    ENDIF.
ENDDO.

WHEN 'R'. IF TRDIRT-NAME <> <FS_REPID>.
    CLEAR TRDIRT.
ENDIF.
IF TRDIRT-TEXT = SPACE.
    SELECT SINGLE * FROM TRDIRT WHERE NAME = LS_REPID
        AND SPRSL = SY-LANGU.

    IF SY-SUBRC <> 0.
        SELECT SINGLE * FROM TRDIRT WHERE NAME = LS_REPID
            AND SPRSL = 'E'.

    ENDIF.
    IF SY-SUBRC <> 0.
        SELECT SINGLE * FROM TRDIRT WHERE NAME = LS_REPID
            AND SPRSL = 'D'.

    ENDIF.
ENDIF.
<FS_TEXT> = TRDIRT-TEXT.

WHEN 'B'.
    "Berichtsgruppe aus GR55
    IF LS_VARI+0(1) = '@' AND
        LS_VARI+3(1) = '@'.
        LS_VARI = LS_VARI+4.
    ENDIF.
    PERFORM GET_GR55_REPORTNAME USING 1 LS_REPID LS_VARI
        LS_REPID LS_VARI.

    IF TRDIRT-NAME <> WK_GR55_REPNAME.
        CLEAR TRDIRT.
    ENDIF.
    IF TRDIRT-TEXT = SPACE.
        SELECT SINGLE * FROM TRDIRT WHERE NAME = WK_GR55_REPNAME
            AND SPRSL = SY-LANGU.

        IF SY-SUBRC <> 0.
            SELECT SINGLE * FROM TRDIRT WHERE NAME =
                WK_GR55_REPNAME
                    AND SPRSL = 'E'.

        ENDIF.
        IF SY-SUBRC <> 0.
            SELECT SINGLE * FROM TRDIRT WHERE NAME =
                WK_GR55_REPNAME
                    AND SPRSL = 'D'.

        ENDIF.
    ENDIF.
    <FS_TEXT> = TRDIRT-TEXT.

WHEN 'T'. IF TSTCT-TCODE <> <FS_REPID>
    OR SY-UCOMM = 'UC_REFRESH_LINE_COMM'
    OR SY-UCOMM = 'UC_REFRESH'.
    CLEAR: TSTCT, DDO2T.
ENDIF.
CLEAR <FS_TEXT>.

IF <FS_VARI> <> SPACE.

    IF LS_VARI+0(1) = '@' AND
        LS_VARI+3(1) = '@'.
        LS_VARI = LS_VARI+4.
    ENDIF.
    CONDENSE LS_VARI NO-GAPS.

    CASE LS_REPID.
        WHEN 'SE11'
        OR 'SE16'
        OR 'SE16N'
        OR 'SE17'

```



```

OR 'SE54'
OR 'SM30'
OR 'SM31'.
SELECT SINGLE * FROM DD02L WHERE TABNAME = LS_VARI.
IF SY-SUBRC = 0.
  SELECT SINGLE * FROM DD02T
    WHERE TABNAME = LS_VARI
    AND DDLANGUAGE = SY-LANGU.
  IF SY-SUBRC <> 0.
    SELECT SINGLE * FROM DD02T
      WHERE TABNAME = LS_VARI
      AND DDLANGUAGE = 'E'.
  ENDIF.
  IF SY-SUBRC <> 0.
    SELECT SINGLE * FROM DD02T
      WHERE TABNAME = LS_VARI
      AND DDLANGUAGE = 'D'.
  ENDIF.
  IF DD02T-DDTEXT <> SPACE.
    <FS_TEXT> = DD02T-DDTEXT.
    TSTCT-TTEXT = DD02T-DDTEXT.
  ENDIF.
ENDIF.

WHEN 'SM34'.
SELECT SINGLE * FROM VCLDIRT
  WHERE VCLNAME = LS_VARI
  AND SPRAS = SY-LANGU.
IF SY-SUBRC = 0.
  <FS_TEXT> = VCLDIRT-TEXT.
  TSTCT-TTEXT = VCLDIRT-TEXT.
ENDIF.

WHEN 'SE71'.
SELECT SINGLE * FROM STXH WHERE TDOBJECT = 'FORM'
  AND TDNAME = LS_VARI
  AND TDID = 'TXT'
  AND TDSPRAS = SY-LANGU.
IF SY-SUBRC = 0.
  <FS_TEXT> = STXH-TDTITLE.
  TSTCT-TTEXT = STXH-TDTITLE.
ENDIF.

WHEN 'SE37'.
SELECT SINGLE * FROM TFTIT
  WHERE SPRAS = SY-LANGU
  AND FUNCNAME = LS_VARI.
IF SY-SUBRC <> 0.
  SELECT SINGLE * FROM TFTIT
    WHERE SPRAS = 'E'
    AND FUNCNAME = LS_VARI.
  ENDIF.
  IF SY-SUBRC <> 0.
    SELECT SINGLE * FROM TFTIT
      WHERE SPRAS = 'D'
      AND FUNCNAME = LS_VARI.
  ENDIF.
  <FS_TEXT> = TFTIT-STEXT.

WHEN 'SE38'
OR 'SA38'
OR 'SE51'
OR 'SE80'.
SELECT SINGLE * FROM TRDIRT WHERE NAME = LS_VARI
  AND SPRSL = SY-LANGU.
IF SY-SUBRC <> 0.
  SELECT SINGLE * FROM TRDIRT
    WHERE NAME = LS_VARI
    AND SPRSL = 'E'.
  ENDIF.
  IF SY-SUBRC <> 0.
    SELECT SINGLE * FROM TRDIRT

```

```

                                WHERE NAME = LS_VARI
                                AND SPRSL = 'D'.
        ENDIF.
        <FS_TEXT> = TRDIRT-TEXT.

        ENDCASE.
    ENDIF.

    CHECK <FS_TEXT> = SPACE.

    IF      LS_REPID(3) = '/T/'.      L_REPID = LS_REPID+3.
    ELSE.      L_REPID = LS_REPID.
    ENDIF.

    IF TSTCT-TTEXT = SPACE.
        SELECT SINGLE * FROM TSTCT WHERE TCODE = L_REPID
            AND SPRSL = SY-LANGU.

        IF SY-SUBRC <> 0.
            SELECT SINGLE * FROM TSTCT WHERE TCODE = L_REPID
                AND SPRSL = 'E'.
        ENDIF.
        IF SY-SUBRC <> 0.
            SELECT SINGLE * FROM TSTCT WHERE TCODE = L_REPID
                AND SPRSL = 'D'.
        ENDIF.
        SELECT SINGLE * FROM TSTC WHERE TCODE = TSTCT-TCODE.
        IF SY-SUBRC = 0 AND
            TSTC-PGMNA(4) = 'MENU'.
            CLEAR TSTCT.
        ENDIF.
    ENDIF.

    <FS_TEXT> = TSTCT-TTEXT.

ENDCASE.

IF SY-UCCOMM = 'UC_REFRESH_LINE_COMM' AND
    WK_REFRESH_LINE = X.
    EXIT.
ENDIF.
ENDDO.

ENDFORM.
=====
AT SELECTION-SCREEN.
=====

IF SY-UCCOMM = 'SPOS'.
    LOOP AT SCREEN.
        IF SCREEN-GROUP1 = 'ND2'.
            SCREEN-INVISIBLE = '0'.
            SCREEN-INPUT      = '1'.
            SCREEN-OUTPUT     = '1'.
            MODIFY SCREEN.
        ENDIF.
    ENDLOOP.
ENDIF.

CLEAR WK_NOTHING_TO_DO.

IF SY-UCCOMM = 'UC_ZZ_DIS_MODE' OR
    SY-UCCOMM = 'UC_ZZ_MOD_MODE' OR
    SY-UCCOMM = 'OPTI'.
    WK_UCCOMM = SY-UCCOMM.
ENDIF.

IF SY-UCCOMM = 'UC_REFRESH_COMM'.
    PERFORM GET_COMMENTS.
    MESSAGE 'If result is OK, do not forget to save the variant!'
        type 'I'.
ENDIF.

```

```
IF SY-UCOMM = 'UC_REFRESH_LINE_COMM'.
  GET CURSOR FIELD WK_CURSOR_FIELD VALUE WK_FIELD_CONTENT.
  IF WK_CURSOR_FIELD+4 BETWEEN 'S001' AND 'S099'.
    WK_REFRESH_LINE = WK_CURSOR_FIELD+5.
    PERFORM GET_COMMENTS.
    IF WK_NOTHING_TO_DO = SPACE.
      MESSAGE 'If result is OK, do not forget to save the variant!'
        type 'I'.
    ELSE.
      MESSAGE 'Nothing to do fort that line!'
        type 'I'.
    ENDIF.
  ELSE.
    MESSAGE 'Position your cursor on a valid line and try again!'
      type 'I'.
  ENDIF.
ENDIF.

*LOOP AT SCREEN.
* IF SCREEN-GROUP1 = 'VIW'.
*   SCREEN-INPUT = 'OFF'.
*   MODIFY SCREEN.
* ENDIF.
*ENDLOOP.

FIELD-SYMBOLS: <FS_REPID>, <FS_VARI>, <FS_COND>, <FS_DATE>,
               <FS_FROMU>, <FS_COMM>,
               <FS_PREPID>, <FS_PVARI>, <FS_PCOND>, <FS_PDATE>,
               <FS_PFROMU>, <FS_PCOMM>.

DATA: FN_REPID(8),           " like trdir-name,
      FN_VARI(8),           " like varid-variant,
      FN_COND(8),
      FN_DATE LIKE SY-DATUM,
      FN_FROMU(8),
      FN_COMM(8),
      X(3) TYPE N,
      ENDE.

IF SY-UCOMM = 'OPTI'.
  SY-UCOMM = 'FC01'.    "Zur Vermeidung unsinniger Meldung

  WK_MESSAGE = 'Open Variant'.
  PERFORM OPEN_VARIANT.
ENDIF.

IF SY-UCOMM(6) = 'OPEN_S'.
  PERFORM OPEN_VARIANT.
ENDIF.

IF SY-UCOMM(6) = 'SUBM_S'.
  PERFORM SUBMIT_VARIANT.
ENDIF.

IF SSCRFIELDS-UCOMM = 'FC01'.
  IF SSCRFIELDS-FUNCTXT_01 <> SPACE.
    PERFORM TOGGLE_FIELD.
    EXIT.
  ENDIF.
ENDIF.

IF SSCRFIELDS-UCOMM = 'FC02'.
  IF SSCRFIELDS-FUNCTXT_02 <> SPACE.
    PERFORM INSERT_LINE.
    EXIT.
  ENDIF.
ENDIF.

IF SSCRFIELDS-UCOMM = 'FC03'.
```

```
IF SSCRFIELDS-FUNCTXT_03 <> SPACE.
  PERFORM DELETE_LINE.
  EXIT.
ENDIF.
ENDIF.

IF SSCRFIELDS-UCOMM = 'FC04'.
  IF SSCRFIELDS-FUNCTXT_04 <> SPACE.
    PERFORM CLEAR_SCREEN.
    EXIT.
  ENDIF.
ENDIF.

X = 0.

DO 99 TIMES.                                "For 99 possible Steps
  ADD 1 TO X.

  CLEAR TAB.
  FN_REPID = 'P_R_S'. FN_REPID+5 = X. ASSIGN (FN_REPID) TO <FS_REPID>.
  FN_VARI = 'P_V_S'. FN_VARI+5 = X. ASSIGN (FN_VARI) TO <FS_VARI>.
  FN_COND = 'P_C_S'. FN_COND+5 = X. ASSIGN (FN_COND) TO <FS_COND>.
  FN_DATE = 'P_D_S'. FN_DATE+5 = X. ASSIGN (FN_DATE) TO <FS_DATE>.

  PERFORM PRUEF_STEP USING X <FS_REPID> <FS_VARI>
                          FN_REPID FN_VARI.
ENDDO.

*=====
FORM PRUEF_STEP USING US_X US_REPID US_VARI
                    FN_REPID FN_VARI.
*=====

DATA: LINE(120),
      LS_REPID TYPE PROGRAM,
      LS_REPIDU TYPE PROGRAM,
      LS_VARIU LIKE WK_C30K,
      LS_VARI LIKE WK_C30K.

CLEAR WK_ACTIONTYPE+US_X(1).
CLEAR WK_ZZ_ME+US_X(1).

CHECK US_REPID <> SPACE.

LS_REPIDU = US_REPID.
TRANSLATE LS_REPIDU TO UPPER CASE.
LS_VARIU = US_VARI.
TRANSLATE LS_VARIU TO UPPER CASE.

LS_REPID = LS_REPIDU.
LS_VARI = LS_VARIU.

if ls_repid(5) = 'ZZ_ME'.
  WK_ZZ_ME+US_X(1) = 'X'.
endif.

IF LS_REPID(5) = 'GR55 '.
  LS_REPID(5) = 'GR55.'.
  CONDENSE LS_REPID+5 NO-GAPS.
ELSE.
  CONDENSE LS_REPID NO-GAPS.
ENDIF.
IF LS_REPID(1) = '@' AND
  LS_REPID+3(1) = '@'.
  LS_REPID = LS_REPID+4.
  TRANSLATE US_REPID+1(2) TO UPPER CASE.
ENDIF.

IF LS_REPID(4) = 'GR55'.
  CASE LS_REPID+4(1).
  WHEN '='
```

```

OR '-'
OR '-'
OR '.'
OR ','
OR '>'
OR '/'
WHEN OTHERS. LS_REPID+5 = LS_REPID+4. CLEAR LS_REPID+4(1).
ENDCASE.
ENDIF.

IF WK_SA38_ALLOWED = 'X'.
  SELECT SINGLE * FROM TRDIR WHERE NAME = LS_REPID.
  IF SY-SUBRC <> 0.
    LINE      = 'Line'.
    LINE+5    = US_X.
    LINE+9    = 'Report:'.
    LINE+18   = LS_REPID.
    LINE+62   = 'does not exist!'.
    CONDENSE LINE.
    CLEAR WK_ACTIONTYPE+US_X(1).
  ELSE.
    WK_ACTIONTYPE+US_X(1) = 'R'.          "Report
  ENDIF.

  IF WK_ACTIONTYPE+US_X(1) = 'R' AND      "Report
     US_VARI <> SPACE.
    DO 2 TIMES.
      SELECT SINGLE * FROM VARID WHERE REPORT = LS_REPID
                                AND VARIANT = LS_VARI.

      IF SY-SUBRC = 0 OR
         LS_VARI+0(1) <> '@'.
        EXIT.
      ENDIF.
      IF LS_VARI+0(1) = '@' AND
         LS_VARI+3(1) = '@'.
        LS_VARI      = LS_VARI+4.
      ENDIF.
    ENDDO.

    IF SY-SUBRC <> 0.
      LINE      = 'Line'.
      LINE+5    = US_X.
      LINE+9    = 'Variant:'.
      LINE+18   = LS_VARI.
      LINE+62   = 'does not exist!'.
      CONDENSE LINE.
    ELSE.
      IF VARID-ENVIRONMNT = 'B'.
        WK_ACTIONTYPE+US_X(1) = 'H'.      "Rep. with vari for background
      ELSE.
        WK_ACTIONTYPE+US_X(1) = 'V'.      "Report with variant
      ENDIF.
      US_VARI = LS_VARIU.
    ENDIF.
  ENDIF.
ENDIF.

IF WK_ACTIONTYPE+US_X(1) <> SPACE.
  US_REPID = LS_REPIDU.
ENDIF.
CHECK WK_ACTIONTYPE+US_X(1) = SPACE.

DATA: L_REPID(40).

SELECT SINGLE * FROM TSTC WHERE TCODE = LS_REPID.
IF SY-SUBRC <> 0.
  IF LS_REPID(3) = '/T/'.    L_REPID = LS_REPID+3.
  ENDIF.
  IF L_REPID <> SPACE.
    SELECT SINGLE * FROM TSTC WHERE TCODE = L_REPID.
  ENDIF.

```

```

ENDIF.

IF SY-SUBRC      = 0      AND
   TSTC-PGMNA(4) <> 'MENU'.
   WK_ACTIONTYPE+US_X(1) = 'T'.           "Transaction
   US_REPID = LS_REPIDU.
ENDIF.

IF WK_ACTIONTYPE+US_X(1) = 'T' AND           "Transaktion
   US_VARI <> SPACE.
   DO 2 TIMES.
     SELECT SINGLE * FROM VARID WHERE REPORT = TSTC-PGMNA
                                   AND VARIANT = LS_VARI.

     IF SY-SUBRC = 0 OR
        LS_VARI+0(1) <> '@'.
       EXIT.
     ENDIF.
     IF LS_VARI+0(1) = '@' AND
        LS_VARI+3(1) = '@'.
       LS_VARI      = LS_VARI+4.
     ENDIF.
   ENDDO.

IF SY-SUBRC <> 0.
   LINE      = 'Line'.
   LINE+5    = US_X.
   LINE+9    = 'Variant:'.
   LINE+18   = US_VARI.
   LINE+62   = 'does not exist!'.
   CONDENSE LINE.
ELSE.
   IF VARID-ENVIRONMNT = 'B'.
     WK_ACTIONTYPE+US_X(1) = 'h'.   "Trans with vari for background
   ELSE.
     WK_ACTIONTYPE+US_X(1) = 'v'.   "Transaction with variant
   ENDIF.
   US_VARI = LS_VARIU.
ENDIF.
ENDIF.

IF WK_ACTIONTYPE+US_X(1) = 'T' AND           "Transaktion
   US_VARI <> SPACE.
   CONDENSE LS_VARI NO-GAPS.

CASE LS_REPID.
  WHEN 'SE11'
    OR 'SE16'
    OR 'SE16N'
    OR 'SE17'
    OR 'SE54'
    OR 'SM30'
    OR 'SM31'. SELECT SINGLE * FROM DD02L WHERE TABNAME = LS_VARI.
               IF SY-SUBRC = 0.
                 US_VARI = LS_VARIU.
               ENDIF.
  WHEN 'SM34'. SELECT SINGLE * FROM VCLDIRT WHERE VCLNAME = LS_VARI
               AND SPRAS = SY-LANGU.
               IF SY-SUBRC = 0.
                 US_VARI = LS_VARIU.
               ENDIF.

  WHEN 'SE71'. SELECT SINGLE * FROM STXH WHERE TDOBJECT = 'FORM'
               AND TDNAME = LS_VARI
               AND TDID = 'TXT'
               AND TDSRAS = SY-LANGU.
               IF SY-SUBRC = 0.
                 US_VARI = LS_VARIU.
               ENDIF.

  WHEN 'SE38'
    OR 'SA38'
    OR 'SE51'

```

```

OR 'SE80'. SELECT SINGLE * FROM TRDIRT WHERE NAME = LS_VARI
                                AND SPRSL = SY-LANGU.
    IF SY-SUBRC <> 0.
        SELECT SINGLE * FROM TRDIRT WHERE NAME = LS_VARI
                                AND SPRSL = 'E'.
    ENDIF.
    IF SY-SUBRC <> 0.
        SELECT SINGLE * FROM TRDIRT WHERE NAME = LS_VARI
                                AND SPRSL = 'D'.
    ENDIF.
    IF SY-SUBRC = 0.
        US_VARI = LS_VARIU.
    ENDIF.
ENDCASE.
ENDIF.

IF WK_ACTIONTYPE+US_X(1) = SPACE .
    IF LS_REPID(4) = 'GR55' AND
        LS_REPID+5 <> SPACE.
        IF LS_REPID(5) = 'GR55=' OR
            LS_REPID(5) = 'GR55-' OR
            LS_REPID(5) = 'GR55_' OR
            LS_REPID(5) = 'GR55.' OR
            LS_REPID(5) = 'GR55,' OR
            LS_REPID(5) = 'GR55>' OR
            LS_REPID(5) = 'GR55/'.

            PERFORM GET_GR55_REPORTNAME USING US_X LS_REPID US_VARI
                                FN_REPID FN_VARI.

        IF WK_ACTIONTYPE+US_X(1) = 'B'.
            US_REPID = LS_REPIDU.
        ENDIF.

        IF WK_ACTIONTYPE+US_X(1) = 'B' AND
            US_VARI <> SPACE AND
            WK_GR55_REPNAME <> SPACE.
            DO 2 TIMES.
                SELECT SINGLE * FROM VARID WHERE REPORT = WK_GR55_REPNAME
                                AND VARIANT = LS_VARI.

                IF SY-SUBRC = 0 OR
                    LS_VARI+0(1) <> '@'.
                    EXIT.
                ENDIF.
                IF LS_VARI+0(1) = '@' AND
                    LS_VARI+3(1) = '@'.
                    LS_VARI = LS_VARI+4.
                ENDIF.
            ENDDO.

        IF SY-SUBRC <> 0.
            LINE = 'Line'.
            LINE+5 = US_X.
            LINE+9 = 'Variant:'.
            LINE+18 = US_VARI.
            LINE+62 = 'does not exist!'.
            CONDENSE LINE.
        ELSE.
            IF VARID-ENVIRONMNT = 'B'.
                WK_ACTIONTYPE+US_X(1) = 'H'. "Rep. with vari for background
            ELSE.
                WK_ACTIONTYPE+US_X(1) = 'V'. "Report with variant
            ENDIF.
            US_VARI = LS_VARIU.
        ENDIF.
    ENDIF.
ENDIF.
ENDIF.
ENDIF.
ENDIF.

```

## ZZ\_ME – Reports und einiges mehr als Favoriten definieren

---

\* Nun kann es sich noch um eine Berichtsgruppen Transaktion handeln  
\* der Reportname muss hier über die Tab. TSTCP ermittelt werden.

```
IF WK_ACTIONTYPE+US_X(1) = 'T' AND          "Transaktion
   US_VARI <> SPACE.
   PERFORM GET_TSTCP_REPORTNAME USING US_X LS_REPID LS_VARIU
                                       FN_REPID FN_VARI.
```

ENDIF.

```
IF WK_ACTIONTYPE+US_X(1) = 'G'.
   US_VARI = LS_VARIU.
ENDIF.
```

```
IF WK_ACTIONTYPE+US_X(1) = SPACE.
   WK_ACTIONTYPE+US_X(1) = 'C'.
ENDIF.
```

ENDFORM.

```
*=====
FORM GET_GR55_REPORTNAME USING US_X US_REPID US_VARI
                             FN_REPID FN_VARI.
```

```
*=====
DATA: L_GENERATED,
      L_E30_JOB(4),
      L_PROG_NAME TYPE PROGNAME,
      L_SUBRC     LIKE SY-SUBRC.
```

CLEAR WK\_GR55\_REPNAME.

```
IF US_REPID(5) = 'GR55=' OR
   US_REPID(5) = 'GR55-' OR
   US_REPID(5) = 'GR55_' OR
   US_REPID(5) = 'GR55.' OR
   US_REPID(5) = 'GR55,' OR
   US_REPID(5) = 'GR55>' OR
   US_REPID(5) = 'GR55/' OR
   US_REPID(5) = 'GR55 ' .
   L_E30_JOB = US_REPID+5.
```

ELSE.

```
   L_E30_JOB = US_REPID+4.
```

ENDIF.

```
SELECT * FROM T803VP WHERE RGJNR      = L_E30_JOB
                        AND PROGCLASS = 'RW_EXECUTION'
                        ORDER BY VERSN DESCENDING.
```

EXIT.

ENDSELECT.

IF SY-SUBRC = 0.

```
   CONCATENATE 'GP' T803VP-UNI_IDC25 SY-MANDT INTO WK_GR55_REPNAME.
```

```
   WK_ACTIONTYPE+US_X(1) = 'B'.
```

EXIT.

ENDIF.

```
CALL FUNCTION 'GRW_JOB_SUBMIT_PREPARE'
  EXPORTING   I_REPORT_GROUP = L_E30_JOB
             I_PROGRAM_TYPE = 'X'
  IMPORTING   E_SUBRC       = L_SUBRC
             E_PROGRAM_NAME = L_PROG_NAME
  EXCEPTIONS OTHERS        = 4.
```

CHECK SY-SUBRC = 0.

```
WK_ACTIONTYPE+US_X(1) = 'B'.
WK_GR55_REPNAME = L_PROG_NAME.
```

ENDFORM.

```
*=====
FORM GET_TSTCP_REPORTNAME USING US_X US_REPID US_VARI
                             FN_REPID FN_VARI.
```

```
*=====
```



```
DATA: L_GENERATED,
      L_E30_JOB(4),
      L_PROG_NAME TYPE PROGNAME,
      L_SUBRC LIKE SY-SUBRC,
      L_TRANS(40),
      L_F1(60),
      L_F2(60),
      L_F3(60).

CLEAR WK_TSTCP_REPNAME.

IF US_REPID(3) = '/T/'. L_TRANS = US_REPID+3.
ELSE. L_TRANS = US_REPID.
ENDIF.

SELECT SINGLE * FROM TSTC WHERE TCODE = L_TRANS.
CHECK SY-SUBRC = 0.
CHECK TSTC-PGMNA = SPACE.

SELECT SINGLE * FROM TSTCP WHERE TCODE = L_TRANS.
CHECK SY-SUBRC = 0.

SPLIT TSTCP-PARAM AT ';' INTO L_F1 L_F2 L_F3.

CHECK L_F1+27 = 'REPORTTYPE=RW'.

L_E30_JOB = L_F2+19.
SELECT * FROM T803VP WHERE RGJNR = L_E30_JOB
      AND PROGCLASS = 'RW_EXECUTION'
      ORDER BY VERSN DESCENDING.

EXIT.
ENDSELECT.

IF SY-SUBRC = 0.
  CONCATENATE 'GP' T803VP-UNI_IDC25 SY-MANDT INTO WK_TSTCP_REPNAME.
  SELECT SINGLE * FROM VARID WHERE REPORT = WK_TSTCP_REPNAME
      AND VARIANT = US_VARI.

  IF SY-SUBRC = 0.
    WK_ACTIONTYPE+US_X(1) = 'G'.
    EXIT.
  ENDIF.
ENDIF.
EXIT.
ENDIF.

CALL FUNCTION 'GRW_JOB_SUBMIT_PREPARE'
  EXPORTING I_REPORT_GROUP = L_E30_JOB
            I_PROGRAM_TYPE = 'X'
  IMPORTING E_SUBRC = L_SUBRC
            E_PROGRAM_NAME = L_PROG_NAME
  EXCEPTIONS OTHERS = 4.

CHECK SY-SUBRC = 0.

WK_TSTCP_REPNAME = L_PROG_NAME.
SELECT SINGLE * FROM VARID WHERE REPORT = WK_TSTCP_REPNAME
      AND VARIANT = US_VARI.

IF SY-SUBRC = 0.
  WK_ACTIONTYPE+US_X(1) = 'G'.
ENDIF.

ENDFORM.
*=====
DATA: T_FIELDS LIKE HELP_VALUE OCCURS 0 WITH HEADER LINE.

DATA: BEGIN OF T_FNAMES OCCURS 0,
      FNAME(21), END OF T_FNAMES.

DATA: BEGIN OF T_ABSTG OCCURS 0,
      KENNZAHN LIKE SY-REPID,
      TEXT LIKE ZVISH-TITEL, END OF T_ABSTG.
```

DATA: WK\_SELECTIONFIELD LIKE HELP\_INFO-FIELDNAME,  
WK\_L\_INDEX LIKE SY-TABIX,  
WK\_VR\_TITEL(40).

DATA: DYNPFIELDS LIKE DYNPREAD OCCURS 0 WITH HEADER LINE,  
LI\_REP LIKE SY-REPID.

DATA: WK\_FF\_INDEX(5) TYPE N,  
WK\_FF\_LINE(150),  
WK\_FF\_LINE\_RESET,  
WK\_FF\_FILTER\_INDEX(5) TYPE N,  
WK\_FF\_NUM5(5) TYPE N,  
WK\_FF\_NODEL\_T\_LOADED\_FILTER,  
WK\_FF\_SCREEN(4),  
WK\_FF\_ANSWER,  
WK\_FF\_MSG(60),  
WK\_FF\_TEXTLINE1(60),  
WK\_FF\_TEXTLINE2(60),  
WK\_FF\_TEXTLINE3(60),  
WK\_FF\_TEXT\_OPTION1(10),  
WK\_FF\_TEXT\_OPTION2(10),  
WK\_FF\_TITEL(25),  
WK\_FF\_UNAME LIKE SY-UNAME,  
WK\_FF\_REPID LIKE SY-REPID,  
WK\_FF\_TOTAL\_LINES(6) TYPE N,  
WK\_FF\_CURRENT\_LINE(6) TYPE N,  
WK\_FF\_REMARK\_NO\_LINES,  
  
WK\_FF\_ENDE2.

DEFINE MACRO\_VALUE\_REQUEST.  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR P\_V\_S&1.  
PERFORM SHOW\_VARIANTS CHANGING P\_V\_S&1.  
END-OF-DEFINITION.

MACRO\_VALUE\_REQUEST 001.  
MACRO\_VALUE\_REQUEST 002.  
MACRO\_VALUE\_REQUEST 003.  
MACRO\_VALUE\_REQUEST 004.  
MACRO\_VALUE\_REQUEST 005.  
MACRO\_VALUE\_REQUEST 006.  
MACRO\_VALUE\_REQUEST 007.  
MACRO\_VALUE\_REQUEST 008.  
MACRO\_VALUE\_REQUEST 009.  
MACRO\_VALUE\_REQUEST 010.  
MACRO\_VALUE\_REQUEST 011.  
MACRO\_VALUE\_REQUEST 012.  
MACRO\_VALUE\_REQUEST 013.  
MACRO\_VALUE\_REQUEST 014.  
MACRO\_VALUE\_REQUEST 015.  
MACRO\_VALUE\_REQUEST 016.  
MACRO\_VALUE\_REQUEST 017.  
MACRO\_VALUE\_REQUEST 018.  
MACRO\_VALUE\_REQUEST 019.  
MACRO\_VALUE\_REQUEST 020.  
MACRO\_VALUE\_REQUEST 021.  
MACRO\_VALUE\_REQUEST 022.  
MACRO\_VALUE\_REQUEST 023.  
MACRO\_VALUE\_REQUEST 024.  
MACRO\_VALUE\_REQUEST 025.  
MACRO\_VALUE\_REQUEST 026.  
MACRO\_VALUE\_REQUEST 027.  
MACRO\_VALUE\_REQUEST 028.  
MACRO\_VALUE\_REQUEST 029.  
MACRO\_VALUE\_REQUEST 030.  
MACRO\_VALUE\_REQUEST 031.  
MACRO\_VALUE\_REQUEST 032.  
MACRO\_VALUE\_REQUEST 033.  
MACRO\_VALUE\_REQUEST 034.  
MACRO\_VALUE\_REQUEST 035.

MACRO\_VALUE\_REQUEST 036.  
MACRO\_VALUE\_REQUEST 037.  
MACRO\_VALUE\_REQUEST 038.  
MACRO\_VALUE\_REQUEST 039.  
MACRO\_VALUE\_REQUEST 040.  
MACRO\_VALUE\_REQUEST 041.  
MACRO\_VALUE\_REQUEST 042.  
MACRO\_VALUE\_REQUEST 043.  
MACRO\_VALUE\_REQUEST 044.  
MACRO\_VALUE\_REQUEST 045.  
MACRO\_VALUE\_REQUEST 046.  
MACRO\_VALUE\_REQUEST 047.  
MACRO\_VALUE\_REQUEST 048.  
MACRO\_VALUE\_REQUEST 049.

MACRO\_VALUE\_REQUEST 050.  
MACRO\_VALUE\_REQUEST 051.  
MACRO\_VALUE\_REQUEST 052.  
MACRO\_VALUE\_REQUEST 053.  
MACRO\_VALUE\_REQUEST 054.  
MACRO\_VALUE\_REQUEST 055.  
MACRO\_VALUE\_REQUEST 056.  
MACRO\_VALUE\_REQUEST 057.  
MACRO\_VALUE\_REQUEST 058.  
MACRO\_VALUE\_REQUEST 059.  
MACRO\_VALUE\_REQUEST 060.  
MACRO\_VALUE\_REQUEST 061.  
MACRO\_VALUE\_REQUEST 062.  
MACRO\_VALUE\_REQUEST 063.  
MACRO\_VALUE\_REQUEST 064.  
MACRO\_VALUE\_REQUEST 065.  
MACRO\_VALUE\_REQUEST 066.  
MACRO\_VALUE\_REQUEST 067.  
MACRO\_VALUE\_REQUEST 068.  
MACRO\_VALUE\_REQUEST 069.  
MACRO\_VALUE\_REQUEST 070.  
MACRO\_VALUE\_REQUEST 071.  
MACRO\_VALUE\_REQUEST 072.  
MACRO\_VALUE\_REQUEST 073.  
MACRO\_VALUE\_REQUEST 074.  
MACRO\_VALUE\_REQUEST 075.  
MACRO\_VALUE\_REQUEST 076.  
MACRO\_VALUE\_REQUEST 077.  
MACRO\_VALUE\_REQUEST 078.  
MACRO\_VALUE\_REQUEST 079.  
MACRO\_VALUE\_REQUEST 080.  
MACRO\_VALUE\_REQUEST 081.  
MACRO\_VALUE\_REQUEST 082.  
MACRO\_VALUE\_REQUEST 083.  
MACRO\_VALUE\_REQUEST 084.  
MACRO\_VALUE\_REQUEST 085.  
MACRO\_VALUE\_REQUEST 086.  
MACRO\_VALUE\_REQUEST 087.  
MACRO\_VALUE\_REQUEST 088.  
MACRO\_VALUE\_REQUEST 089.  
MACRO\_VALUE\_REQUEST 090.  
MACRO\_VALUE\_REQUEST 091.  
MACRO\_VALUE\_REQUEST 092.  
MACRO\_VALUE\_REQUEST 093.  
MACRO\_VALUE\_REQUEST 094.  
MACRO\_VALUE\_REQUEST 095.  
MACRO\_VALUE\_REQUEST 096.  
MACRO\_VALUE\_REQUEST 097.  
MACRO\_VALUE\_REQUEST 098.  
MACRO\_VALUE\_REQUEST 099.

```
*=====
FORM SHOW_VARIANTS CHANGING L_VARI.
*=====
DATA L_CURR_LINE(3) TYPE N.
```

## ZZ\_ME – Reports und einiges mehr als Favoriten definieren

---

```
CLEAR:  T_FIELDS, T_FNAMES, T_ABSTG.
REFRESH: T_FIELDS, T_FNAMES, T_ABSTG.

T_FNAMES-FNAME = 'VARIT-VARIANT'. APPEND T_FNAMES.
T_FNAMES-FNAME = 'ZVISH-TITEL'.  APPEND T_FNAMES.
WK_SELECTIONFIELD = 'KENNZAHL'.

GET CURSOR FIELD WK_CURSOR_FIELD VALUE WK_FIELD_CONTENT.

L_CURR_LINE = WK_CURSOR_FIELD+5.

FN_REPID = 'P_R_S'. FN_REPID+5 = L_CURR_LINE.
ASSIGN (FN_REPID) TO <FS_REPID>.

CLEAR DYNPFIELDS. REFRESH DYNPFIELDS.
DYNPFIELDS-FIELDNAME = FN_REPID.
APPEND DYNPFIELDS.

WK_FF_REPID = SY-REPID.

CALL FUNCTION 'DYNP_VALUES_READ'
  EXPORTING DYNNAME           = WK_FF_REPID
            DYNUMB           = '1000'
  TABLES   DYNPFIELDS      = DYNPFIELDS
  EXCEPTIONS INVALID_ABAPWORKAREA = 1

IF SY-SUBRC <> 0.
  CLEAR LI_REP.
ELSE.
  LI_REP = DYNPFIELDS-FIELDVALUE.
  TRANSLATE LI_REP TO UPPER CASE.
ENDIF.

CASE WK_ACTIONTYPE+L_CURR_LINE(1).
  WHEN 'R'.
    SELECT * FROM VARIT WHERE LANGU   = SY-LANGU
              AND REPORT = LI_REP.
      T_ABSTG-KENNZAHL = VARIT-VARIANT.
      T_ABSTG-TEXT    = VARIT-VTEXT.
      APPEND T_ABSTG.
      CLEAR T_ABSTG.
    ENDSELECT.

    CONCATENATE 'Variants of report' <FS_REPID>
      INTO WK_VR_TITEL
      SEPARATED BY SPACE.
  WHEN 'T'.
    SELECT SINGLE * FROM TSTC WHERE TCODE = LI_REP.
    IF SY-SUBRC = 0.
      SELECT * FROM VARIT WHERE LANGU   = SY-LANGU
                AND REPORT = TSTC-PGMNA.
        T_ABSTG-KENNZAHL = VARIT-VARIANT.
        T_ABSTG-TEXT    = VARIT-VTEXT.
        APPEND T_ABSTG.
        CLEAR T_ABSTG.
      ENDSELECT.

      CONCATENATE 'Variants of transaction' <FS_REPID>
        INTO WK_VR_TITEL
        SEPARATED BY SPACE.
    ENDIF.
ENDCASE.

PERFORM DISPLAY_REQUST_VALUES TABLES T_ABSTG.

CHECK SY-SUBRC = 0.
READ TABLE T_ABSTG INDEX WK_L_INDEX.
IF SY-SUBRC = 0.
  MOVE T_ABSTG-KENNZAHL TO L_VARI.
ENDIF.
```

ENDFORM.

```
*=====
FORM DISPLAY_REQUST_VALUES TABLES FORM_TABLE1.
*=====
```

```
CALL FUNCTION 'TRANSFER_NAMES_TO_FIELDS'
  EXPORTING SELECTFIELD      = WK_SELECTIONFIELD
  TABLES   FIELDS          = T_FIELDS
           NAMELIST         = T_FNAMES
  EXCEPTIONS WRONG_FORMAT_GIVEN = 1
           OTHERS           = 2.
IF SY-SUBRC = 0. ENDIF.
```

```
CALL FUNCTION 'HELP_VALUES_GET_NO_DD_NAME'
  EXPORTING SELECTFIELD      = WK_SELECTIONFIELD
           TITEL            = WK_VR_TITEL
  IMPORTING IND              = WK_L_INDEX
  TABLES   FIELDS          = T_FIELDS
           FULL_TABLE       = FORM_TABLE1
  EXCEPTIONS FULL_TABLE_EMPTY = 1
           NO_TABLESTRUCTURE_GIVEN = 2
           NO_TABLEFIELDS_IN_DICTIONARY = 3
           MORE_THEN_ONE_SELECTFIELD = 4
           NO_SELECTFIELD      = 5
           OTHERS              = 6.
IF SY-SUBRC = 0. ENDIF.
```

ENDFORM.

```
*=====
START-OF-SELECTION.
*=====
```

```
IMPORT MEMORY_PREREP FROM MEMORY ID 'MEMORY_PREREP'.
FREE MEMORY ID 'MEMORY_PREREP'.
```

```
CHECK MEMORY_PREREP-PREREP <> SY-REPID.
```

```
SET PF-STATUS 'LISTE'.
```

```
IF SY-SLSET <> SPACE.
  MEMORY_PREREP-PREREP = SY-REPID.
  EXPORT MEMORY_PREREP TO MEMORY ID 'MEMORY_PREREP'.
  SUBMIT (SY-REPID) VIA SELECTION-SCREEN
    USING SELECTION-SET SY-SLSET AND RETURN.
ENDIF.
```

```
WK_MESSAGE = ' Start of processing'.
```

```
*=====
FORM OPEN_VARIANT.
*=====
```

```
DATA: LS_REPID TYPE PROGRAM,
      LS_VARI(30),
      L_VARI_VALID,
      L_ACTIONTYPE.
```

```
IF SY-UCOMM(6) <> 'OPEN_S'.
  GET CURSOR FIELD WK_CURSOR_FIELD VALUE WK_FIELD_CONTENT.
```

```
  IF WK_CURSOR_FIELD(5) = 'P_R_S'.
    EXIT.
  ENDIF.
```

```
  IF WK_CURSOR_FIELD = SPACE OR
     WK_CURSOR_FIELD(5) <> 'P_V_S'.
    MESSAGE 'Select a Variant in column "Variant"!'
      type 'E'.
  ENDIF.
```

```
ELSE.
  CONCATENATE 'P_V_' SY-UCOMM+5 INTO WK_CURSOR_FIELD.
ENDIF.
```

## ZZ\_ME – Reports und einiges mehr als Favoriten definieren

---

FIELD-SYMBOLS: <FS\_REPID>, <FS\_VARI>, <FS\_COND>, <FS\_DATE>.

```
DATA: FN_REPID(8),           " like trdir-name,
      FN_VARI(8),           " like varid-variant,
      X(3) TYPE N,
```

ENDE.

```
FN_VARI      = WK_CURSOR_FIELD. ASSIGN (FN_VARI) TO <FS_VARI>.
FN_REPID     = FN_VARI.
FN_REPID+2(1) = 'R'.           ASSIGN (FN_REPID) TO <FS_REPID>.
```

```
LS_REPID = <FS_REPID>.
TRANSLATE LS_REPID TO UPPER CASE.
CONDENSE LS_REPID NO-GAPS.
IF LS_REPID(1) = '@' AND
   LS_REPID+3(1) = '@'.
   LS_REPID = LS_REPID+4.
ENDIF.
```

X = FN\_VARI.

```
IF LS_REPID = SPACE.
   MESSAGE 'Report missing in current line'
         type 'E'.
ENDIF.
```

```
IF LS_REPID(4) = 'GR55'.
   CASE LS_REPID+4(1).
     WHEN '='
     OR '-'
     OR '_'
     OR '.'
     OR ','
     OR '>'
     OR '/'
     OR ' '.
     WHEN OTHERS. LS_REPID+5 = LS_REPID+4. CLEAR LS_REPID+4(1).
   ENDCASE.
ENDIF.
```

```
IF LS_REPID(5) = 'GR55=' OR
   LS_REPID(5) = 'GR55-' OR
   LS_REPID(5) = 'GR55_' OR
   LS_REPID(5) = 'GR55.' OR
   LS_REPID(5) = 'GR55,' OR
   LS_REPID(5) = 'GR55>' OR
   LS_REPID(5) = 'GR55/' OR
   LS_REPID(5) = 'GR55 '.
```

```
L_ACTIONTYPE = WK_ACTIONTYPE+X(1).
PERFORM GET_GR55_REPORTNAME USING 1 LS_REPID <FS_VARI>
                                LS_REPID <FS_VARI>.
```

```
TAB-REP = WK_GR55_REPNAME.
WK_ACTIONTYPE+X(1) = L_ACTIONTYPE.
```

```
ELSE.
   PERFORM PRUEF_STEP USING X LS_REPID <FS_VARI>
                           FN_REPID FN_VARI.

   TAB-REP = LS_REPID.
   IF WK_ACTIONTYPE+X(1) = 'G'.
     TAB-REP = WK_TSTCP_REPNAME.
   ENDIF.
ENDIF.
```

```
TAB-VARI = <FS_VARI>.
TRANSLATE TAB-VARI TO UPPER CASE.
```

```
IF WK_ACTIONTYPE+X(1) = 'v'.
   ASSIGN (TSTC-PGMNA) TO <FS_REPID>.
ELSE.
```

## ZZ\_ME – Reports und einiges mehr als Favoriten definieren

---

```
    ASSIGN (TAB-REP)    TO <FS_REPID>.
ENDIF.

IF WK_ACTIONTYPE+X(1) = 'v'.
    ASSIGN (TSTC-PGMNA) TO <FS_REPID>.
ELSE.
    ASSIGN (TAB-REP)    TO <FS_REPID>.
ENDIF.

IF TAB-VARI+0(1) = '@' AND
   TAB-VARI+3(1) = '@'.
    SELECT SINGLE * FROM VARID WHERE REPORT = <FS_REPID>
                                   AND VARIANT = <FS_VARI>.

    IF SY-SUBRC <> 0.
        TAB-VARI = TAB-VARI+4.
    ENDIF.
ENDIF.

CASE WK_ACTIONTYPE+X(1).
  WHEN 'R'.
    OR 'B'.  SUBMIT (TAB-REP)    VIA SELECTION-SCREEN          AND RETURN.
  WHEN 'V'.
    OR 'G'.
    OR 'H'.  SUBMIT (TAB-REP)    VIA SELECTION-SCREEN
                                   USING SELECTION-SET TAB-VARI AND RETURN.
  WHEN 'v'.
    OR 'h'.  SUBMIT (TSTC-PGMNA) VIA SELECTION-SCREEN
                                   USING SELECTION-SET TAB-VARI AND RETURN.
ENDCASE.

LS_VARI = <FS_VARI>.

CASE WK_ACTIONTYPE+X(1).
  WHEN 'T'.
    IF LS_VARI+0(1) = '@' AND
       LS_VARI+3(1) = '@'.
        LS_VARI = LS_VARI+4.
    ENDIF.
    CONDENSE LS_VARI NO-GAPS.
    CASE LS_REPID.
      WHEN 'SE11'
      OR 'SE16'
      OR 'SE16N'
      OR 'SE17'
      OR 'SE54'
      OR 'SM30'
      OR 'SM31'.
        IF <FS_VARI> <> SPACE.
          SELECT SINGLE * FROM DD02L WHERE TABNAME = LS_VARI.
          IF SY-SUBRC = 0.
            L_VARI_VALID = 'X'.
            CASE LS_REPID.
              WHEN 'SE11'
              OR 'SE16'
              OR 'SE16N'
              OR 'SE17'. SET PARAMETER ID 'DTB' FIELD LS_VARI.
              WHEN 'SM30'
              OR 'SM31'
              OR 'SE54'. SET PARAMETER ID 'DVI' FIELD LS_VARI.
            ENDCASE.
          ENDIF.
        ENDIF.
      WHEN 'SM34'. SET PARAMETER ID 'VCL' FIELD LS_VARI.
      WHEN 'SE71'. SET PARAMETER ID 'TXF' FIELD LS_VARI.
      WHEN 'SE37'. SET PARAMETER ID 'LIB' FIELD LS_VARI.
      WHEN 'SA38'
      OR 'SE38'
      OR 'SE80'. SET PARAMETER ID 'RID' FIELD LS_VARI.
        IF LS_REPID = 'SA38'.
          IF SY-SUBRC = 0.
            L_VARI_VALID = 'X'.
          ENDIF.
        ENDIF.
    ENDCASE.
  ENDIF.
ENDCASE.
```

```

                ENDIF.
            ENDIF.
        WHEN 'SE51'. SET PARAMETER ID 'DYP' FIELD LS_VARI.
    ENDCASE.

    DATA L_TRANS LIKE TSTC-TCODE.

    IF     TAB-REP(3) = '/T/'.      L_TRANS = TAB-REP+3.
    ELSE.   L_TRANS = TAB-REP.
    ENDIF.

    SET CURSOR FIELD FN_REPID.
    CALL FUNCTION 'AUTHORITY_CHECK_TCODE'
        EXPORTING TCODE          = L_TRANS
        EXCEPTIONS OK             = 1
                  NOT_OK         = 2
                  OTHERS         = 3
        .

    IF SY-SUBRC > 1.
        SET CURSOR FIELD FN_REPID.
        concatenate 'No authority for transaction: '
                   L_trans
                   into wk_message
                   SEPARATED BY SPACE.
        MESSAGE wk_message
               type 'I'.
    ELSE.
        IF     L_VARI_VALID = 'X'.
            CALL TRANSACTION L_TRANS AND SKIP FIRST SCREEN.
        ELSE. CALL TRANSACTION L_TRANS.
        ENDIF.
    ENDIF.
    EXIT.
ENDCASE.

ENDFORM.
=====
FORM SUBMIT_VARIANT.
=====
DATA: LS_REPID TYPE PROGRAM,
      LS_VARI(30),
      L_VARI_VALID,
      L_ACTIONTYPE.

IF SY-UCOMM(6) <> 'SUBM_S'.
    GET CURSOR FIELD WK_CURSOR_FIELD VALUE WK_FIELD_CONTENT.

    IF WK_CURSOR_FIELD = SPACE      OR
       WK_CURSOR_FIELD(5) <> 'P_V_S'.
        MESSAGE 'Select a Variant in column "Variant"!'
               type 'E'.
    ENDIF.
ELSE.
    CONCATENATE 'P_V_' SY-UCOMM+5 INTO WK_CURSOR_FIELD.
ENDIF.

FIELD-SYMBOLS: <FS_REPID>, <FS_VARI>, <FS_COND>, <FS_DATE>.

DATA: FN_REPID(8),           " like trdir-name,
      FN_VARI(8),           " like varid-variant,
      X(3) TYPE N,
      ENDE.

FN_VARI      = WK_CURSOR_FIELD. ASSIGN (FN_VARI) TO <FS_VARI>.
FN_REPID     = FN_VARI.
FN_REPID+2(1) = 'R'.          ASSIGN (FN_REPID) TO <FS_REPID>.

LS_REPID = <FS_REPID>.
TRANSLATE LS_REPID TO UPPER CASE.
CONDENSE LS_REPID NO-GAPS.

```



```
IF LS_REPID(1) = '@' AND
  LS_REPID+3(1) = '@'.
  LS_REPID = LS_REPID+4.
ENDIF.

LS_VARI = <FS_VARI>.
TRANSLATE LS_VARI TO UPPER CASE.

X = FN_VARI.

IF <FS_REPID> = SPACE.
  MESSAGE 'Report missing in current line'
    type 'I'.
ENDIF.

IF LS_REPID(4) = 'GR55' AND
  LS_REPID+5 <> SPACE.
  L_ACTIONTYPE = WK_ACTIONTYPE+X(1).
  PERFORM GET_GR55_REPORTNAME USING 1 LS_REPID <FS_VARI>
    LS_REPID <FS_VARI>.

  TAB-REP = WK_GR55_REPNAME.
  WK_ACTIONTYPE+X(1) = L_ACTIONTYPE.
ELSE.
  PERFORM PRUEF_STEP USING X LS_REPID LS_VARI
    FN_REPID FN_VARI.

  TAB-REP = LS_REPID.
  IF WK_ACTIONTYPE+X(1) = 'G'.
    TAB-REP = WK_TSTCP_REPNAME.
  ENDIF.
ENDIF.

TAB-VARI = <FS_VARI>.

CASE WK_ACTIONTYPE+X(1).
  WHEN 'T'.
    IF LS_VARI+0(1) = '@' AND
      LS_VARI+3(1) = '@'.
      LS_VARI = LS_VARI+4.
    ENDIF.
  CONDENSE LS_VARI NO-GAPS.
  CASE LS_REPID.
    WHEN 'SE11'
    OR 'SE16'
    OR 'SE16N'
    OR 'SE17'
    OR 'SE54'
    OR 'SM30'
    OR 'SM31'.
      IF <FS_VARI> <> SPACE.
        SELECT SINGLE * FROM DD02L WHERE TABNAME = LS_VARI.
        IF SY-SUBRC = 0.
          L_VARI_VALID = 'X'.
          CASE LS_REPID.
            WHEN 'SE11'
            OR 'SE16'
            OR 'SE16N'
            OR 'SE17'. SET PARAMETER ID 'DTB' FIELD LS_VARI.
            WHEN 'SM30'
            OR 'SM31'
            OR 'SE54'. SET PARAMETER ID 'DVI' FIELD LS_VARI.
          ENDCASE.
        ENDIF.
      ENDIF.
    WHEN 'SM34'. SET PARAMETER ID 'VCL' FIELD LS_VARI.
    WHEN 'SE71'. SET PARAMETER ID 'TXF' FIELD LS_VARI.
    WHEN 'SE37'. SET PARAMETER ID 'LIB' FIELD LS_VARI.
    WHEN 'SA38'
    OR 'SE38'
    OR 'SE80'. SET PARAMETER ID 'RID' FIELD LS_VARI.
      IF LS_REPID = 'SA38'.
```

```

                IF SY-SUBRC = 0.
                    L_VARI_VALID = 'X'.
                ENDIF.
            ENDF.
        WHEN 'SE51'. SET PARAMETER ID 'DYR' FIELD LS_VARI.
    ENDCASE.

DATA L_TRANS LIKE TSTC-TCODE.

IF     TAB-REP(3) = '/T/'.     L_TRANS = TAB-REP+3.
ELSE.   L_TRANS = TAB-REP.
ENDIF.

SET CURSOR FIELD FN_REPID.
CALL FUNCTION 'AUTHORITY_CHECK_TCODE'
    EXPORTING TCODE           = L_TRANS
    EXCEPTIONS OK              = 1
              NOT_OK          = 2
              OTHERS          = 3
    .

IF SY-SUBRC > 1.
    SET CURSOR FIELD FN_REPID.
    concatenate 'No authority for transaction: '
                L_trans
                into wk_message
                SEPARATED BY SPACE.
    MESSAGE wk_message
            type 'I'.
ELSE.
    IF     L_VARI_VALID = 'X'.
        CALL TRANSACTION L_TRANS AND SKIP FIRST SCREEN.
    ELSE. CALL TRANSACTION L_TRANS.
    ENDF.
ENDIF.
EXIT.
ENDCASE.

IF WK_ACTIONTYPE+X(1) = 'v'.
    ASSIGN (TSTC-PGMNA) TO <FS_REPID>.
ELSE.
    ASSIGN (TAB-REP) TO <FS_REPID>.
ENDIF.

IF TAB-VARI+0(1) = '@' AND
    TAB-VARI+3(1) = '@'.
    SELECT SINGLE * FROM VARID WHERE REPORT = <FS_REPID>
        AND VARIANT = <FS_VARI>.

    IF SY-SUBRC <> 0.
        TAB-VARI = TAB-VARI+4.
    ENDF.
ENDIF.

CASE WK_ACTIONTYPE+X(1).
    WHEN 'V'
    OR 'R'
    OR 'G'.
        IF <FS_VARI> = SPACE.
            SUBMIT (TAB-REP) AND RETURN.
        ELSE.
            SUBMIT (TAB-REP) USING SELECTION-SET TAB-VARI AND RETURN.
        ENDF.
    WHEN 'v'.
        SUBMIT (TSTC-PGMNA) USING SELECTION-SET TAB-VARI AND RETURN.
    WHEN 'H'
    OR 'H'.
        concatenate 'Variant >'
                    TAB-VARI
                    '< has flag "Only for Background Processing'
                    into wk_message

```

separated by space.

```
MESSAGE wk_message  
      type 'I'.
```

ENDCASE.

```
*=====
```

```
* Direkt als Job ausführen
```

```
*=====
```

```
*  
* http://www.abapforum.com/forum/viewtopic.php?t=2472 / 27.11.2004  
*  
* Henry fragt zuerst und liefert dann später auch die Antwort  
*  
* Ich möchte eine Report aus der Selektionmaske heraus im Hintergrund  
* starten. Der Ablauf sollte ohne die SAP Standardfunktionalität "F9"  
* erfolgen.  
*  
*falls jemand schon mal das Problem hatte hier eine Lösung die ich  
* herausgefunden habe:  
*  
* 1) Aktuelle Selektionparameter holen mit  
*   FUBA CALL FUNCTION 'RS_REFRESH_FROM_SELECTOPTIONS'  
*  
* 2) Job Öffnen  
*   CALL FUNCTION 'JOB_OPEN'  
*  
* 3) Report für Job definieren  
*   SUBMIT (l_report) VIA SELECTION-SCREEN  
*           VIA JOB l_job NUMBER l_jobnum  
*           WITH SELECTION-TABLE l_selopt  AND RETURN.  
*  
* 4) Job abschließen  
*   CALL FUNCTION 'JOB_CLOSE'  
*  
* Funktioniert einwandfrei und es lassen sich auch noch Option für  
* Jobdatum/-Zeit mitgeben.  
*  
* Allerdings wäre es für andere Einsatzzwecke manchmal sehr hilfreich,  
* trotzdem das aktuelle Selektionbild (u.U. mit verschiedenen Tabstrips)  
* als Variante direkt ohne den SAP Standard-Popup speichern zu können.  
* Also falls hierzu jemand was weiß, wäre ich für Hilfe sehr dankbar.  
*  
* Henry hfahrian  
*  
* ForumUser  
* Beiträge: 33 Registriert: 15.11.2004, 11:10  
* Wohnort: Frankenland  Dank erhalten: 0 mal  
*
```

ENDFORM.

```
*=====
```

```
FORM TOGGLE_FIELD.
```

```
*=====
```

```
DATA: L_POS(3) TYPE N.
```

```
GET CURSOR FIELD WK_CURSOR_FIELD VALUE WK_FIELD_CONTENT.
```

```
CHECK WK_CURSOR_FIELD+4 <> 'TOGL'.
```

```
LOOP AT SCREEN.
```

```
  IF SCREEN-GROUP1 = 'ND2'.  
    SCREEN-INVISIBLE = '0'.  
    SCREEN-INPUT      = '1'.  
    SCREEN-OUTPUT     = '1'.  
    MODIFY SCREEN.
```

```
  ENDIF.  
ENDLOOP.
```

```
L_POS = WK_CURSOR_FIELD+5.
```

```

CASE WK_CURSOR_FIELD(4).
  WHEN 'PM_C'. IF    P_C_TOGL+L_POS(1) = SPACE.
                  P_C_TOGL+L_POS(1) = 'X'.
                  ELSE. P_C_TOGL+L_POS(1) = ' '.
                  ENDIF.
  WHEN 'P_R'. IF    P_R_TOGL+L_POS(1) = SPACE.
                  P_R_TOGL+L_POS(1) = 'X'.
                  ELSE. P_R_TOGL+L_POS(1) = ' '.
                  ENDIF.
  WHEN 'P_V'. IF    P_V_TOGL+L_POS(1) = SPACE.
                  P_V_TOGL+L_POS(1) = 'X'.
                  ELSE. P_V_TOGL+L_POS(1) = ' '.
                  ENDIF.
  WHEN 'P_I'. IF    P_I_TOGL+L_POS(1) = SPACE.
                  P_I_TOGL+L_POS(1) = 'X'.
                  ELSE. P_I_TOGL+L_POS(1) = ' '.
                  ENDIF.
ENDCASE.

LOOP AT SCREEN.
  CHECK SCREEN-NAME = WK_CURSOR_FIELD.
  L_POS = SCREEN-NAME+5.

  CASE WK_CURSOR_FIELD(4).
    WHEN 'PM_C'. IF    P_C_TOGL+L_POS(1) = SPACE.
                  SCREEN-INPUT = '1'.
                  ELSE. SCREEN-INPUT = '0'.
                  ENDIF.
    WHEN 'PM_R'. IF    P_R_TOGL+L_POS(1) = SPACE.
                  SCREEN-INPUT = '1'.
                  ELSE. SCREEN-INPUT = '0'.
                  ENDIF.
    WHEN 'PM_V'. IF    P_V_TOGL+L_POS(1) = SPACE.
                  SCREEN-INPUT = '1'.
                  ELSE. SCREEN-INPUT = '0'.
                  ENDIF.
    WHEN 'PM_I'. IF    P_I_TOGL+L_POS(1) = SPACE.
                  SCREEN-INPUT = '1'.
                  ELSE. SCREEN-INPUT = '0'.
                  ENDIF.

  ENDCASE.
  MODIFY SCREEN.
ENDLOOP.

ENDFORM.
*=====
FORM INSERT_LINE.
*=====
DATA: WK_CURR_LINE(3)   TYPE N,
      WK_LAST_LINE(3)  TYPE N VALUE 99,
      WK_BEFLAST_LINE(3) TYPE N VALUE 98,
      WK_INSERT_POSSIBLE,
      WK_SUCCESSFUL.

FIELD-SYMBOLS: <FS_REPID>, <FS_VARI>, <FS_COND>, <FS_DATE>,
              <FS_FROMU>, <FS_COMM>,

              <FS_PREPID>, <FS_PVARI>, <FS_PCOND>, <FS_PDATE>,
              <FS_PFROMU>, <FS_PCOMM>,

              <FS_NREPID>, <FS_NVARI>, <FS_NCOND>, <FS_NDATE>,
              <FS_NFROMU>, <FS_NCOMM>.

DATA: FN_REPID(8),           " like trdir-name,
      FN_VARI(8),           " like varid-variant,
      FN_FROMU(8),
      FN_COMM(8),

      FN_PREPID(8),         " like trdir-name,
      FN_PVARI(8),         " like varid-variant,

```

```

    FN_PFROMU(8),
    FN_PCOMM(8),

    FN_NREPID(8),           " like trdir-name,
    FN_NVARI(8),           " like varid-variant,
    FN_NFROMU(8),
    FN_NCOMM(8).

GET CURSOR FIELD WK_CURSOR_FIELD VALUE WK_FIELD_CONTENT.

IF WK_CURSOR_FIELD(2) <> 'P_'.
    MESSAGE 'Position your cursor on a STEP line!'
        type 'E'.
ENDIF.

IF WK_CURSOR_FIELD+5 = '099'.
    MESSAGE 'Insert on last line not possible!'
        type 'E'.
ENDIF.

WK_CURR_LINE = WK_CURSOR_FIELD+5.

DO 99 TIMES.
    IF WK_BEFLAST_LINE < WK_CURR_LINE.
        EXIT.
    ENDIF.

    FN_REPID = 'P_R_S'. FN_REPID+5 = WK_LAST_LINE.
        ASSIGN (FN_REPID) TO <FS_REPID>.
    FN_VARI = 'P_V_S'. FN_VARI+5 = WK_LAST_LINE.
        ASSIGN (FN_VARI) TO <FS_VARI>.
    FN_COMM = 'P_I_S'. FN_COMM+5 = WK_LAST_LINE.
        ASSIGN (FN_COMM) TO <FS_COMM>.

    IF <FS_REPID> = SPACE      AND
       <FS_VARI> = SPACE      AND
       <FS_COMM> = SPACE      AND
       P_R_TOGL+WK_LAST_LINE(1) = SPACE AND
       P_V_TOGL+WK_LAST_LINE(1) = SPACE AND
       P_I_TOGL+WK_LAST_LINE(1) = SPACE.
        WK_INSERT_POSSIBLE = 'X'.
    ELSE.
        SUBTRACT 1 FROM WK_LAST_LINE.
        SUBTRACT 1 FROM WK_BEFLAST_LINE.
        CONTINUE.
    ENDIF.

    FN_PREPID = 'P_R_S'. FN_PREPID+5 = WK_BEFLAST_LINE.
        ASSIGN (FN_PREPID) TO <FS_PREPID>.
    FN_PVARI = 'P_V_S'. FN_PVARI+5 = WK_BEFLAST_LINE.
        ASSIGN (FN_PVARI) TO <FS_PVARI>.
    FN_PCOMM = 'P_I_S'. FN_PCOMM+5 = WK_BEFLAST_LINE.
        ASSIGN (FN_PCOMM) TO <FS_PCOMM>.

    IF <FS_PREPID> = SPACE      AND
       <FS_PVARI> = SPACE      AND
       <FS_PCOMM> = SPACE      AND
       P_R_TOGL+WK_BEFLAST_LINE(1) = ' ' AND
       P_V_TOGL+WK_BEFLAST_LINE(1) = ' ' AND
       P_I_TOGL+WK_BEFLAST_LINE(1) = ' '.
        SUBTRACT 1 FROM WK_LAST_LINE.
        SUBTRACT 1 FROM WK_BEFLAST_LINE.
        CONTINUE.
    ENDIF.

    MOVE <FS_PREPID> TO <FS_REPID>. CLEAR <FS_PREPID>.
    MOVE <FS_PVARI> TO <FS_VARI>. CLEAR <FS_PVARI>.
    MOVE <FS_PCOMM> TO <FS_COMM>. CLEAR <FS_PCOMM>.

    MOVE P_R_TOGL+WK_BEFLAST_LINE(1) TO P_R_TOGL+WK_LAST_LINE(1).
    CLEAR P_R_TOGL+WK_BEFLAST_LINE(1).

```

## ZZ\_ME – Reports und einiges mehr als Favoriten definieren

---

```
MOVE P_V_TOGL+WK_BEFLAST_LINE(1) TO P_V_TOGL+WK_LAST_LINE(1).
CLEAR P_V_TOGL+WK_BEFLAST_LINE(1).
MOVE P_I_TOGL+WK_BEFLAST_LINE(1) TO P_I_TOGL+WK_LAST_LINE(1).
CLEAR P_I_TOGL+WK_BEFLAST_LINE(1).
```

```
WK_SUCCESSFUL = 'X'.
```

```
SUBTRACT 1 FROM WK_LAST_LINE.
SUBTRACT 1 FROM WK_BEFLAST_LINE.
```

```
ENDDO.
```

```
ENDFORM.
```

```
*=====
FORM DELETE_LINE.
```

```
*=====
DATA: WK_CURR_LINE(3)   TYPE N,
      WK_LAST_LINE(3)  TYPE N VALUE 99,
      WK_NEXT_LINE(3)  TYPE N VALUE 1,
      WK_BEFLAST_LINE(3) TYPE N VALUE 98,
      WK_DO_LINE(3)   TYPE N,
      WK_INSERT_POSSIBLE,
      WK_SUCCESSFUL.
```

```
FIELD-SYMBOLS: <FS_REPID>, <FS_VARI>, <FS_COND>, <FS_DATE>,
               <FS_FROMU>, <FS_COMM>,

               <FS_PREPID>, <FS_PVARI>, <FS_PCOND>, <FS_PDATE>,
               <FS_PFROMU>, <FS_PCOMM>,

               <FS_NREPID>, <FS_NVARI>, <FS_NCOND>, <FS_NDATE>,
               <FS_NFROMU>, <FS_NCOMM>.
```

```
DATA: FN_REPID(8),           " like trdir-name,
      FN_VARI(8),           " like varid-variant,
      FN_COND(8),
      FN_DATE      LIKE SY-DATUM,
      FN_FROMU(8),
      FN_COMM(8),

      FN_PREPID(8),         " like trdir-name,
      FN_PVARI(8),         " like varid-variant,
      FN_PCOND(8),
      FN_PDATE      LIKE SY-DATUM,
      FN_PFROMU(8),
      FN_PCOMM(8),

      FN_NREPID(8),        " like trdir-name,
      FN_NVARI(8),        " like varid-variant,
      FN_NCOND(8),
      FN_NDATE      LIKE SY-DATUM,
      FN_NFROMU(8),
      FN_NCOMM(8).
```

```
GET CURSOR FIELD WK_CURSOR_FIELD VALUE WK_FIELD_CONTENT.
```

```
IF WK_CURSOR_FIELD(2) <> 'P_'.
  MESSAGE 'Position your cursor on a STEP line!'
    type 'E'.
ENDIF.
```

```
WK_CURR_LINE = WK_CURSOR_FIELD+5.
```

```
DO 99 TIMES.
  ADD 1 TO WK_DO_LINE.
  ADD 1 TO WK_NEXT_LINE.
  IF WK_DO_LINE < WK_CURR_LINE.
    CONTINUE.
  ENDIF.
```

```
IF WK_NEXT_LINE < 100.
```

```
FN_REPID = 'P_R_S'. FN_REPID+5 = WK_NEXT_LINE.
                                ASSIGN (FN_REPID) TO <FS_REPID>.
FN_VARI  = 'P_V_S'. FN_VARI+5  = WK_NEXT_LINE.
                                ASSIGN (FN_VARI) TO <FS_VARI>.
FN_COMM  = 'P_I_S'. FN_COMM+5  = WK_NEXT_LINE.
                                ASSIGN (FN_COMM) TO <FS_COMM>.
ENDIF.

FN_PREPID = 'P_R_S'. FN_PREPID+5 = WK_DO_LINE.
                                ASSIGN (FN_PREPID) TO <FS_PREPID>.
FN_PVARI  = 'P_V_S'. FN_PVARI+5  = WK_DO_LINE.
                                ASSIGN (FN_PVARI) TO <FS_PVARI>.
FN_PCOMM  = 'P_I_S'. FN_PCOMM+5  = WK_DO_LINE.
                                ASSIGN (FN_PCOMM) TO <FS_PCOMM>.

IF WK_DO_LINE < 99.
    MOVE <FS_REPID> TO <FS_PREPID>.
    MOVE <FS_VARI> TO <FS_PVARI>.
    MOVE <FS_COMM> TO <FS_PCOMM>.

    MOVE P_R_TOGL+WK_NEXT_LINE(1) TO P_R_TOGL+WK_DO_LINE(1).
    CLEAR P_R_TOGL+WK_NEXT_LINE(1).
    MOVE P_V_TOGL+WK_NEXT_LINE(1) TO P_V_TOGL+WK_DO_LINE(1).
    CLEAR P_V_TOGL+WK_NEXT_LINE(1).
    MOVE P_I_TOGL+WK_NEXT_LINE(1) TO P_I_TOGL+WK_DO_LINE(1).
    CLEAR P_I_TOGL+WK_NEXT_LINE(1).
ENDIF.

IF WK_NEXT_LINE = 99.
    CLEAR <FS_REPID>.
    CLEAR <FS_VARI>.
    CLEAR <FS_COMM>.
ENDIF.

WK_SUCCESSFUL = 'X'.
ENDDO.

ENDFORM.
*=====
FORM CLEAR_SCREEN.
*=====
FIELD-SYMBOLS: <FS_CLR_SCREEN>.

LOOP AT SCREEN.
    CHECK SCREEN-NAME(4) <> 'TEXT'.
    CHECK SCREEN-INVISIBLE = 0.
    CHECK SCREEN-ACTIVE    = 1.
    CHECK SCREEN-GROUP1(2) <> 'PB'.
    ASSIGN (SCREEN-NAME) TO <FS_CLR_SCREEN>.
    CHECK <FS_CLR_SCREEN> <> SPACE.
    CLEAR <FS_CLR_SCREEN>.
ENDLOOP.

ENDFORM.
```