TABLES/AS

Reference Manual

Part 2

DISCLAIMER

The information presented is subject to change. Every effort has been taken to ensure that this manual is accurate and up to date. However, Specialized Software International, Inc. disclaims any liability for any loss or damage resulting from the use of this manual.

May 2005
This edition applies to Release 5.1 of **TABLES/AS**

TABLES/AS Generalized Online Maintenance

Table of Contents

3. Miscellaneous functions

4. Screen processing

Section 3

TABLES/AS

Miscellaneous Functions

This section of **TABLES/AS** Generalized Online Maintenance covers the options from the **TABLES/AS** Primary Option menu that fall outside the Screen Generation and the Screen Processing. These options are:

5- PREPARE SCREEN
6- TEST FACILITY
7- UTILITIES
8- TABLES/AP

Primary Option Menu
Option ===> 5 - Prepare Screen

To "Prepare Screen" means to gather information from numerous control tables containing the screen layout, field edits, custom messages, relational edits, discrete values, maps and processing options. The results of "Prepare Screen" are stored in the SCREENDB control table.

Prior to testing or using a screen with the Screen Processor, it must be prepared. This can occur by selecting Option 9 - Screen Processing with the screen name entered and pressing ENTER. The screen is prepared and then it is displayed.

You may also use Option 5 - Prepare Screen. If you create or change two or more screens that you want to test with the Screen Processor, you could prepare screens 2-X with Option 5 and screen 1 with Option 9. You could also prepare all the screens with Option 5 and then choose Option 9 without a screen name.

If you create a screen load module, it must be recreated whenever:

•Screen format is changed.

•Field edits are changed.

•Relational edits are changed.

•Field names used on the screen are changed.

•Attributes for fields on the screen are changed.

•Mapping is changed.
  · A new table map is added.
  · Processing for a table map is changed.
  · A previously defined table map is changed.

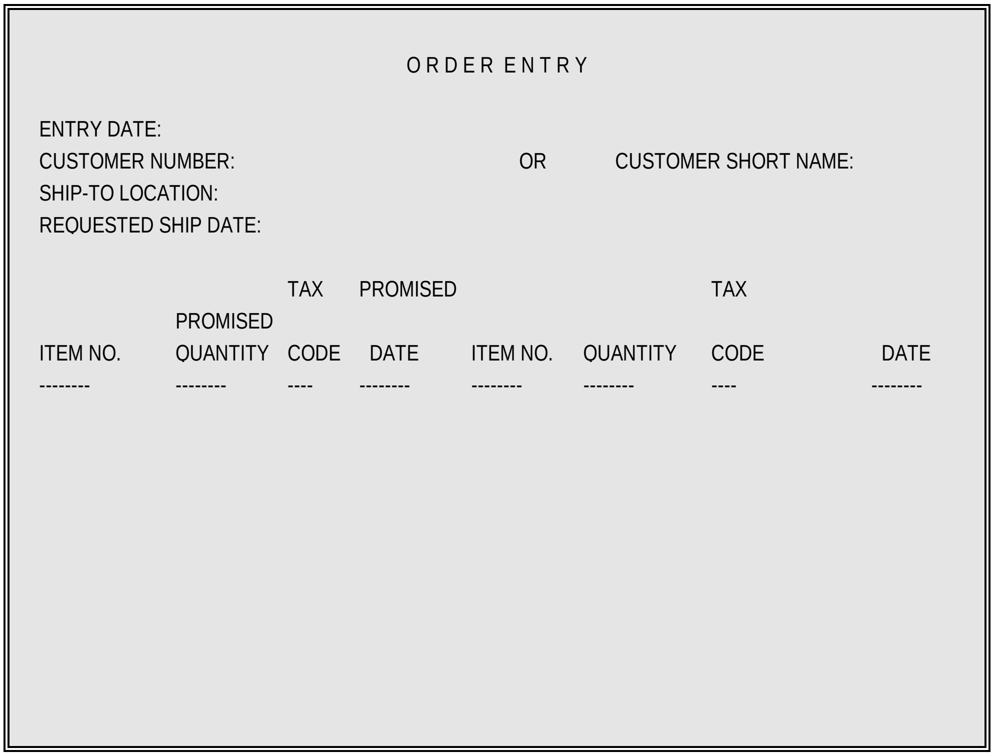
•Screen description variables are changed.

Primary Option Menu
Option ===> 6 - Test Facility

This facility allows input of data to the screen as the end user would. All defined field editing and relational editing is tested. No processing against physical tables is performed.

This process simply tests the editing defined for the screen input fields.

Figure 3-1 is an example of a screen in the test mode.



```
                              ORDER ENTRY

ENTRY DATE:
CUSTOMER NUMBER:                        OR      CUSTOMER SHORT NAME:
SHIP-TO LOCATION:
REQUESTED SHIP DATE:

                      TAX   PROMISED                        TAX
             PROMISED
ITEM NO.     QUANTITY CODE   DATE     ITEM NO.  QUANTITY    CODE         DATE
--------     -------- ----  --------  --------  --------    ----       --------
```

Figure 3-1

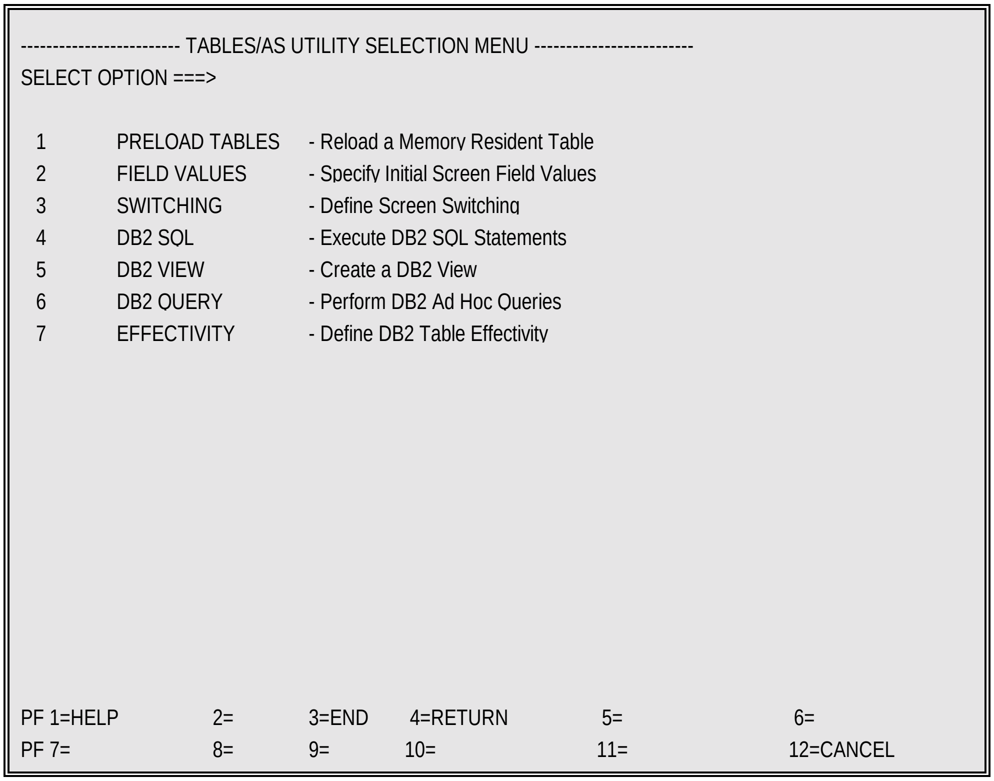
The cursor is positioned at the first field on the screen. Proceed to enter the data. After pressing ENTER all editing takes place. If there are any errors, those fields are highlighted and the error message displayed in the common message field or special message field. Correct the field(s) in error and press ENTER. When all errors are corrected, the screen shows the last valid values entered, the cursor is positioned at the first field and there are no error messages shown.

PF KEYS

PF3	Return to the **TABLES/AS** Primary Option menu
PF4	Return to the **TABLES/AS** Primary Option menu
PF12	Return to the **TABLES/AS** Primary Option menu
ENTER	Activate editing

Primary Option Menu
Option ===> 7 - Utilities

Figure 3-2 show the **TABLES/AS** Utility Selection menu.



```
------------------------ TABLES/AS UTILITY SELECTION MENU ------------------------
SELECT OPTION ===>

   1         PRELOAD TABLES     - Reload a Memory Resident Table
   2         FIELD VALUES       - Specify Initial Screen Field Values
   3         SWITCHING          - Define Screen Switching
   4         DB2 SQL            - Execute DB2 SQL Statements
   5         DB2 VIEW           - Create a DB2 View
   6         DB2 QUERY          - Perform DB2 Ad Hoc Queries
   7         EFFECTIVITY        - Define DB2 Table Effectivity













PF 1=HELP           2=       3=END     4=RETURN          5=              6=
PF 7=               8=       9=        10=               11=             12=CANCEL
```

Figure 3-2

The options are:

1 - Reload a memory resident table
2 - Specify initial screen field values
3 - Define screen switching
4 - Execute DB2 SQL statements
5 - Create a DB2 view
6 - Perform DB2 ad hoc queries
7 - Define DB2 table effectivity

PF KEYS

PF3	Return to the **TABLES/AS** Primary Option menu
PF4	Return to the **TABLES/AS** Primary Option menu
PF12	Return to the **TABLES/AS** Primary Option menu

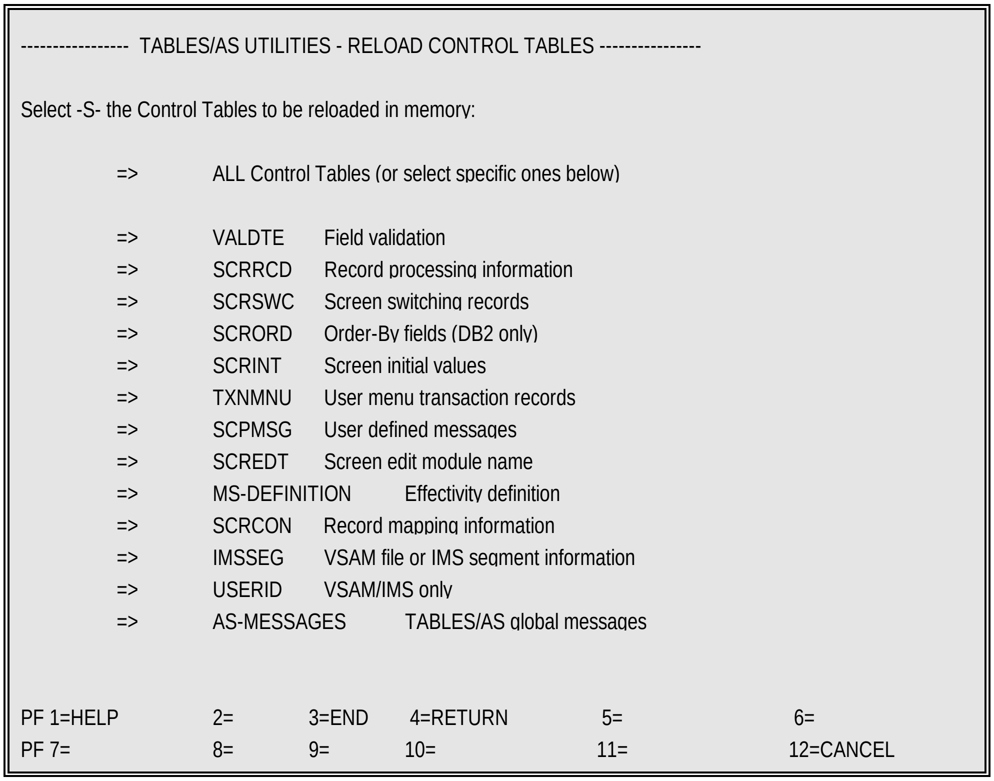
OPTION ===> 1 - PRELOAD TABLES

The **TABLES/AS** Screen Processor preloads a number of tables for more efficient processing. This preload takes place only when a table is not already loaded. In order to reflect changes in preloaded tables, the changed table(s) must be refreshed in the CICS or IMS region.

When the Screen Processor is invoked from Option 9 of the Primary menu, it accesses **all** SSI control tables from disk. When invoked in any other way, the SSI control tables shown in Figure 3-3 are accessed from memory. The other ways the Screen Processor may be invoked are:

- its own transaction code
- a transaction code assigned to a screen with an entry in the TXNMNU table
- a switch from a user-written program.

Figure 3-3 shows the screen used to select which table(s) to purge and reload.

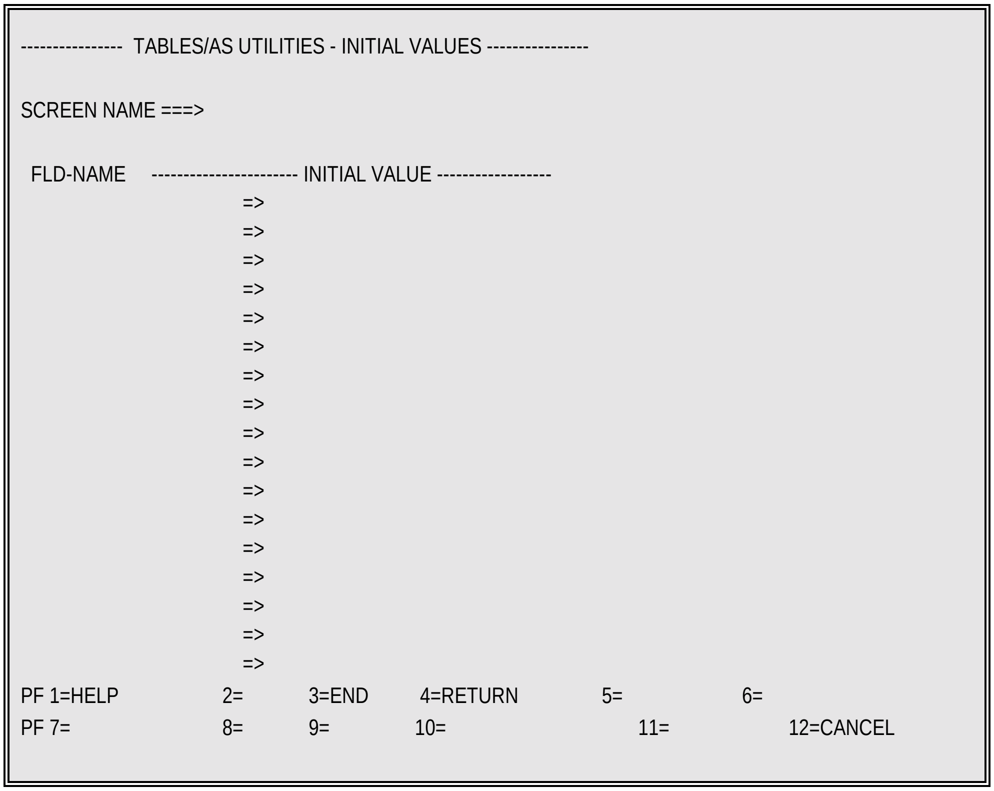


```
----------------  TABLES/AS UTILITIES - RELOAD CONTROL TABLES ----------------

Select -S- the Control Tables to be reloaded in memory:

        =>      ALL Control Tables (or select specific ones below)

        =>      VALDTE      Field validation
        =>      SCRRCD      Record processing information
        =>      SCRSWC      Screen switching records
        =>      SCRORD      Order-By fields (DB2 only)
        =>      SCRINT      Screen initial values
        =>      TXNMNU      User menu transaction records
        =>      SCPMSG      User defined messages
        =>      SCREDT      Screen edit module name
        =>      MS-DEFINITION     Effectivity definition
        =>      SCRCON      Record mapping information
        =>      IMSSEG      VSAM file or IMS segment information
        =>      USERID      VSAM/IMS only
        =>      AS-MESSAGES       TABLES/AS global messages


PF 1=HELP       2=      3=END    4=RETURN       5=            6=
PF 7=           8=      9=       10=            11=           12=CANCEL
```

Figure 3-3

(S) SELECT TABLES BEING RELOADED
Enter an S next to the table(s) being reloaded.
Press ENTER. A message is displayed indicating that the table(s) is reloaded.
Press PF12 when completed.

OPTION ===> 2 - FIELD VALUES
Figure 3-4 shows the screen used to set up initial values on a screen.



```
---------------  TABLES/AS UTILITIES - INITIAL VALUES ---------------

SCREEN NAME ===>

  FLD-NAME    ---------------------- INITIAL VALUE -----------------
                        =>
                        =>
                        =>
                        =>
                        =>
                        =>
                        =>
                        =>
                        =>
                        =>
                        =>
                        =>
                        =>
                        =>
                        =>
                        =>
                        =>
PF 1=HELP         2=      3=END     4=RETURN        5=          6=
PF 7=             8=      9=        10=                 11=         12=CANCEL
```

Figure 3-4

Enter a screen name and the existing initial values are displayed.

This screen allows initial values to be added or changed. There are two key words that are used to establish date and time values on a screen. Field values 'DATE=Format1' and 'TIME=Format2' are used to indicate that the Initial Values are date and time.

Format1 is used to specify date format containing MM (month), DD (day), YY (year), and DDD (day of the year) in any combination. Any special characters can be inserted between.

Format2 is used to specify time format containing HR (hour), MN (minute), SC (second), and HS (hundreds of seconds) in any combination. Any special character can be inserted in between.

To have the screen processor display the Trancode, Terminal ID or User ID enter the keywords:

TRANID=
TERMID=
USERID=

The screen fields for these should be 8 characters alphanumeric, i.e., -A(8).

The initial values are displayed by the Screen Processor at the following selected points:

•Switching to a new screen (Function T) using the PF key or option code

•Sending a screen for the first time from the Screen Processor menu.

•Switching to a different screen whose fields are blank through the editor mode.

•Switching to a different screen whose fields are blank through the table processing mode.

The control table, SCRINT, is used to maintain the initial values by the screen name. This table is viewed using **TABLES/AP**.

NOTE

When field names are changed, remember to update the initial values.

OPTION ===> 3 - SWITCHING

This facility allows for user controlled switching from the screen currently being processed. This can be specified to occur as a result of a PF key being pressed or because of a specific entry on the screen.

Screen switching defines how your application fits together and interacts with your users. **TABLES/AS** provides the ability to switch within and beyond **TABLES/AS** applications.

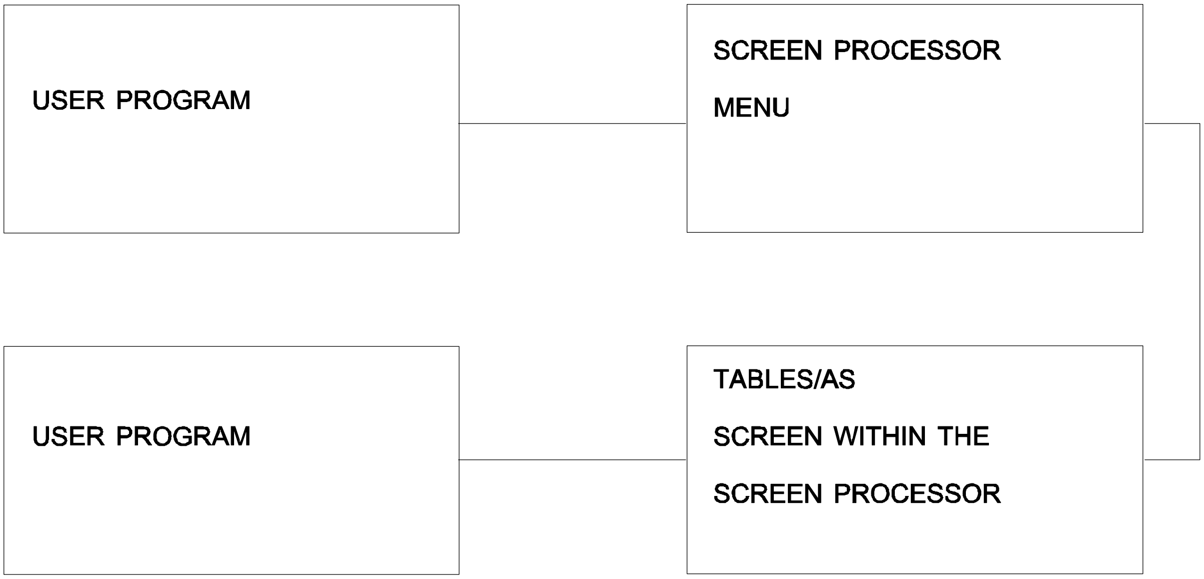
Screen switching involves switching between screens in the Screen Processor, switching to multiple screens from one source screen based on certain conditions, and switching between the Screen Processor in **TABLES/AS** and screens in other applications throughout your system.

When switching is performed, there is also the ability to:

- Save the current screen data to be redisplayed later
- Restore data previously saved for a screen
- Both save the current screen data and restore data previously saved for a screen.

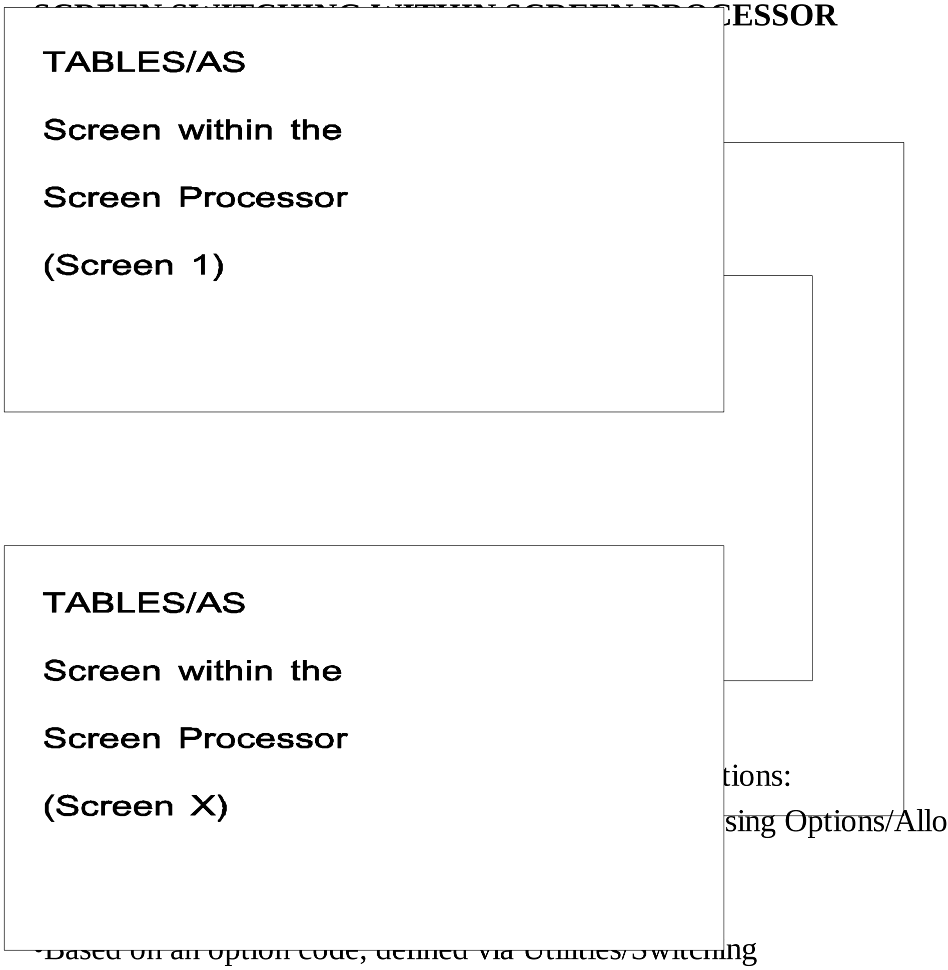
Save is used with the current screen (source screen). If the source screen is the **TABLES/AS** Screen Processor menu, you cannot use the Save or Both options. Restore works with the screen you are switching to (target screen). If screen data has not been previously saved, the target screen is displayed without any data fields filled in.

The following schematics show the switching capabilities.



When switching to or from the Screen Processor menu, data is not passed when the switch occurs. When switching to and from a screen within Screen Processor, data can be passed.

SCREEN SWITCHING WITHIN SCREEN PROCESSOR



tions:

sing Options/Allow

•Based on an option code; defined via Utilities/Switching

•Based on the presence of data; defined via Utilities/Switching

TRANSFERRING DATA

Data to be transferred from one screen to the next is defined using the record mapping process. A COMMON map name, mapped to the same table and containing only the columns being transferred, must exist in both screens.

If the data being passed is part of a repeating group, the repeating group is passed to the target screen, up to the number of entries that the target screen can hold.

To pass data when switching from the source screen to the target screen enter the MAP NAME on the Switching Definition or the Allow Functions screen. If a MAP NAME is not specified, then no data is passed when the switch occurs.

SWITCHING USING A PF KEY OR AN OPTION CODE

In some cases it may be necessary to switch to another screen at a time other than the completion of the table processing; it may be necessary to switch to different screens based on an operator's decision. This is accomplished by defining PF keys or option code as switch conditions.

An option within the **TABLES/AS** utility called Screen Switching is used to define the switch conditions. This process results in updating a control table, SCRSWC, which **TABLES/AS** Screen Processor uses to control switching. This table may be viewed under **TABLES/AP**.

SWITCHING BASED ON THE PRESENCE OF DATA

When switching is selected based on an option code field, and it is required to switch on any non-blank value, then enter a value of '┐ BLANK' under the option value field.

SWITCHING FROM MULTIPLE SCREENS TO A SINGLE SCREEN AND BACK

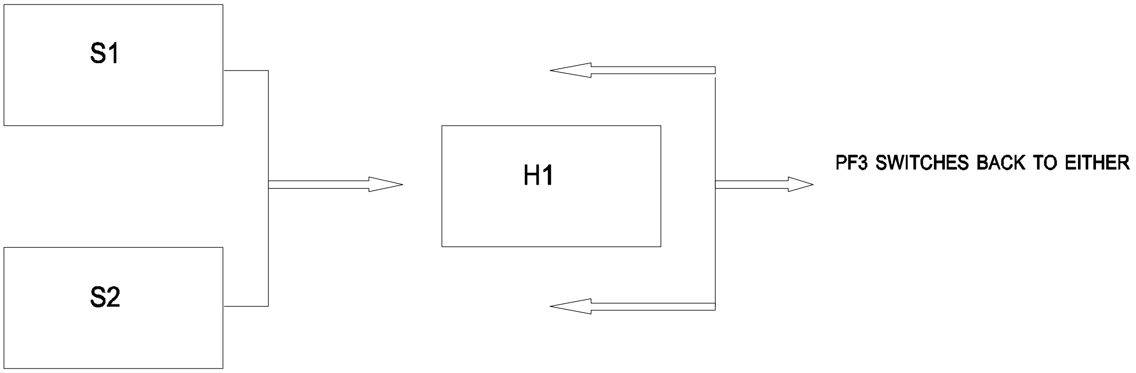
To switch from many screens to one screen (e.g., a help screen) and then switch back to the calling screen, without coding Edit modules or exits, **TABLE/AS** saves the original screen under a special name. To switch back to that original screen, the restore is done to that special name, ($$n).

To accomplish this, screen switching must be defined exactly as they are now with the following two changes:

1. When defining screen switching, you can save a screen under a special name. Setting the Save/Restore Indicator to a number from 1 to 9 will save the screen with a name of $$1 to $$9. Values 2-9 are normally used only when there is more than one level of multi-screen to single screen switching required.
2. To restore a special name screen, you specify the target name of $$1 to $$9 and a Save/Restore Indicator of R.

To illustrate this, assume we have the following situation:

Source screens S1 and S2 want to switch to help screen H1 and PF3 is used to end the help and switch back to the original screen (S1 or S2).



To set up the switching, the following switches would be defined:

	Switch 1	Switch 2	Switch 3	Notes
Source	S1	S2	H1	
Target	H1	H1	$$n	Where n = 1 to 9
PF Key	01	01	03	
Save/Restore Indicator	n	n	R	Where n = 1 to 9

NOTE

The value of n must be the same for each switch record for this to work correctly.

Switch 1 and 2 set up the switch from screens S1 and S2 to H1 using PF1, defined the same as before, except the Save/Restore indicator is new. If set to a value of 1 to 9 (instead of S or B), the screen is saved under a special name of $$1 to $$9.

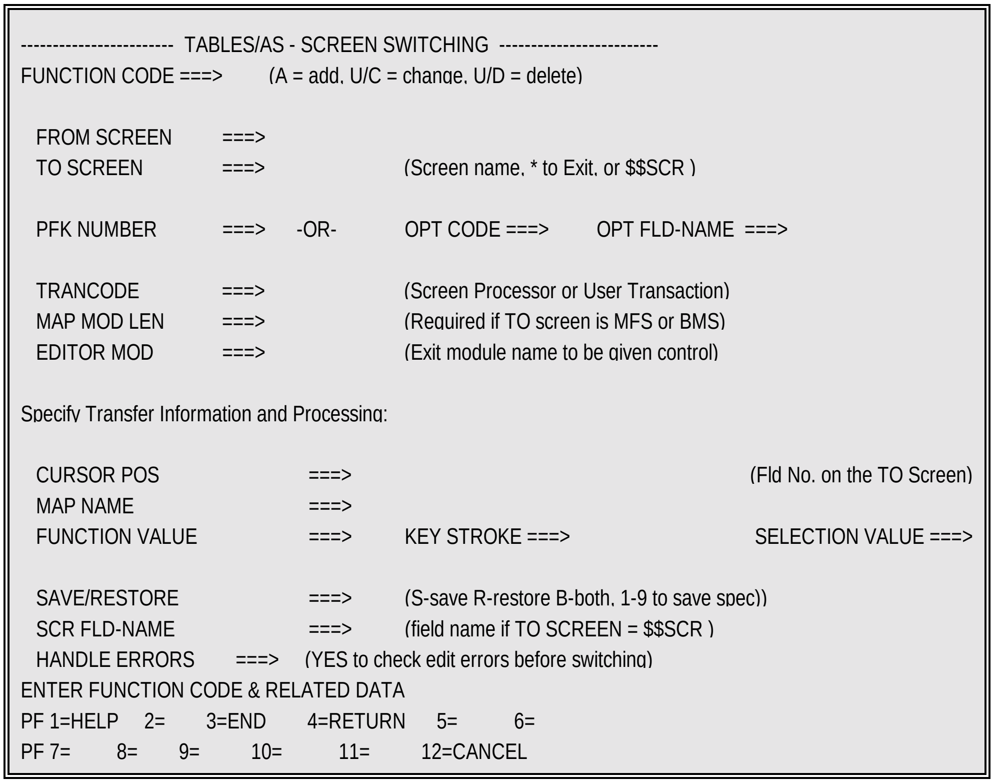
Switch 3 sets up the return from screen H1 using PF3. This is exactly the same as before except the target screen is now $$n (i.e., $$1 to $$9). This allows switching back to any screen that switches to it using the new Save/Restore values 1 to 9.

In the example, if n=1, then if screen S1 or S2 switch to H1, H1 will automatically switch back to the original one. Note that it does not matter how many screens switch to H1, as long as they switch to it and save using $$1, H1 will always switch back to the correct one.

NOTE

Since all saved screens are saved using the terminal name as the first part of the key, one user can never affect another when using the $$n special names. However, for one user, if one screen switches and saves using $$1, and the screen switched to also switches and saves using $$1, then the first screen saved is lost.

When you select Option 3 on the **TABLES/AS** Utility Selection menu the Screen Switching definition panel is displayed.



```
-----------------------  TABLES/AS - SCREEN SWITCHING  ------------------------
FUNCTION CODE ===>        (A = add, U/C = change, U/D = delete)

  FROM SCREEN        ===>
  TO SCREEN          ===>                    (Screen name, * to Exit, or $$SCR )

  PFK NUMBER         ===>      -OR-          OPT CODE ===>        OPT FLD-NAME  ===>

  TRANCODE           ===>                    (Screen Processor or User Transaction)
  MAP MOD LEN        ===>                    (Required if TO screen is MFS or BMS)
  EDITOR MOD         ===>                    (Exit module name to be given control)

Specify Transfer Information and Processing:

  CURSOR POS                  ===>                                       (Fld No. on the TO Screen)
  MAP NAME                    ===>
  FUNCTION VALUE              ===>        KEY STROKE ===>                SELECTION VALUE ===>

  SAVE/RESTORE                ===>        (S-save R-restore B-both, 1-9 to save spec))
  SCR FLD-NAME                ===>        (field name if TO SCREEN = $$SCR )
  HANDLE ERRORS        ===>     (YES to check edit errors before switching)
ENTER FUNCTION CODE & RELATED DATA
PF 1=HELP    2=       3=END       4=RETURN     5=         6=
PF 7=        8=       9=         10=          11=        12=CANCEL
```

Figure 3-5

RETRIEVE EXISTING RECORDS

Retrieval can be performed using any field or combination of fields. Records will be returned where all entered values exist in the returned record. They will be displayed one at a time.

If a retrieval for update has been done, any of the displayed records may be changed or deleted and new records may be added. Press ENTER to display the next record. Up to 50 records are retrieved at one time.

ADD NEW RECORDS

FUNCTION CODE

Enter an A.

FROM SCREEN

Enter a valid **TABLES/AS** screen name.

TO SCREEN

Enter a valid **TABLES/AS** screen name.

Enter * to define exit condition to CICS or IMS.

Enter $$SCR to indicate that the TO SCREEN name will be found under field name SCR FLD-NAME. This will allow the TO SCREEN name to be changed dynamically during processing.

Enter $$n to indicate that the TO SCREEN has been previously saved by TABLES with a save/restore value of n.

PF KEY NUMBER

Enter a value between 01-24 or enter OPT code and FLD name. You may also use PF Key 99 to identify the clear key to cause an exit to the TP monitor.

OPT CODE

Enter the code or value that causes the switch to occur. The maximum length of this value is eight characters.

If it is required to switch on any non-blank value, then enter a value of '┐BLANK' under the option code.

OPT FLD-NAME

Enter the name of the field in which the option code is entered.

TRANSACTION CODE

Enter the transaction code of the program being switched to.

For switching within the screen processor, enter screen processor transaction code.

For switching to CICS or IMS, leave the transaction code blank.

During screen switching, the transaction to process the screen can now be set = '*', as follows:

```
TRANCODE ===> *
```

This means, that whatever transaction code was used to start the program will be used during the screen switch. Therefore, if TS09 was entered initially, then any switch from a screen with '*' as the trancode will continue to use TS09.

If the screen value was set to $$SCR, you may enter a TRANCODE value to be used for the screen named in the SCR FLD-NAME.

MFS MOD LEN/BMS MAP LEN

Enter the MAP length of the non-AS screen being switched to, such as the length of a BMS or MFS screen.

OPTIONAL FIELDS

EDITOR PROGRAM NAME

To call a user-written program, before exiting the source screen and before the target screen is displayed, enter the editor program name.

FIELD NUMBER FOR CURSOR POSITIONING

To alter the standard cursor positioning on the target screen, enter the field number where you want the cursor positioned. The field number corresponds to the sequence of the field in the copy code, not necessarily the sequence on the screen. For example, the fifth data field in the copy code is number five.

> **NOTE**
>
> If a change is made to the screen which affects the copy code (adding or deleting fields) the field number for cursor positioning may have to be updated.

TRANSFER DATA

MAP NAME TO TRANSFER

Enter the map name of the table being transferred.

The table being transferred must be defined as a map on both the source screen and the target screen. The map for this table must have the same name in both screens. Only the data fields common to both table maps are transferred. If data is not being passed from one screen to the other, leave the map name blank.

NOTE: CICS ONLY

2 special capabilities exist to pass the communications area back to a user program. In both cases the TO screen is coded as '*'.

- To return only the **TABLES/AS** req'd 100 byte commarea, enter 'COMMAREA' in the MAP NAME field.
- To return the **TABLES/AS** req'd 100 byte commarea plus up to 8000 bytes of user commarea, previously passed to **TABLES/AS**, enter 'COMMSAVE' in the MAP NAME field.

FUNCTION VALUE

A function to be invoked internally during transfer to the TO SCREEN such that retrieval or insertion of data may be made before the TO SCREEN is displayed. Valid function values are: A (add), I (retrieval for inquiry), U (retrieval for update), R (retrieval for inquiry).

KEY STROKE

Valid key stroke range is 00-24 if entered.

The key stroke entered identifies the particular table to process before displaying the TO SCREEN. The key stroke entered here corresponds to the "Keystroke to Process" defined in the ALLOW FUNCTIONS in the TO SCREEN.

SELECTION VALUE

Selection value is a function code character such as S or X used to select a particular row or rows to be transferred from the FROM SCREEN to the TO SCREEN. The selection value is always entered in the function code field. Therefore, the valid function codes (R, I, B, U, A, C, D) are not valid selection values.

SAVE AND RESTORE DATA

SCREEN SAVE/RESTORE OPTION

- Enter an S to save data fields on the source screen before switching.
- Enter an R to restore data fields previously saved for the target screen when switching occurs.
- Enter a B to both save and restore.
- Leave the field blank if saving or restoring is not desired.
- Enter a value of 1-9 if switching to a common screen such as a Help screen

SCR FLD-NAME

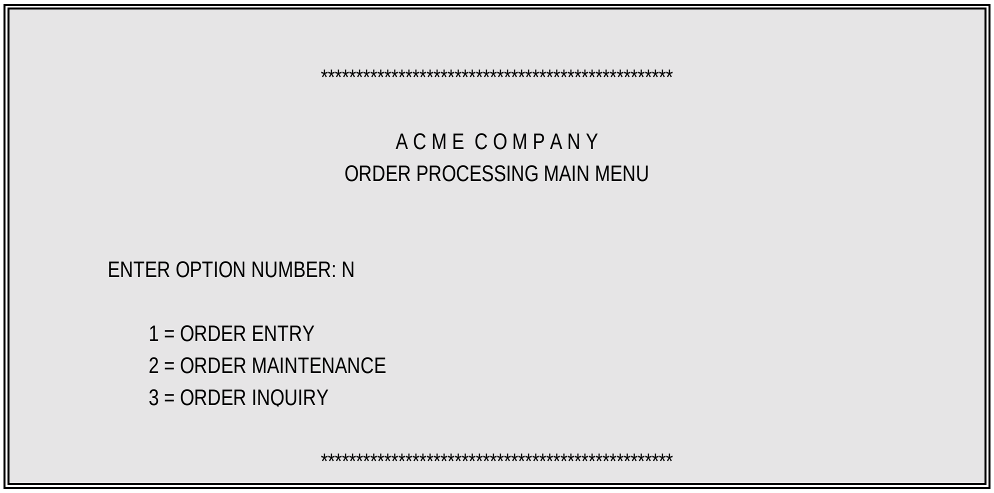
This is the name of a field on the FROM SCREEN which will provide a value for the TO SCREEN name to switch to. This field is used when the TO SCREEN contains a value of $$SCR.

HANDLE ERRORS

Enter the value **Y** if you do not want a switch to occur on an error. Otherwise, leave blank.

After the SCRSWC records are set up for your screen, the switching logic may be tested using the Screen Processor. It is not necessary to recreate the module for screen testing when a change is made to the SCRSWC table.

For example, assume that the menu screen shown in Figure 3-7 called Order Processing Main menu has been designed.



```
                 **************************************************

                                  ACME COMPANY
                          ORDER PROCESSING MAIN MENU


      ENTER OPTION NUMBER: N

         1 = ORDER ENTRY
         2 = ORDER MAINTENANCE
         3 = ORDER INQUIRY

                 **************************************************
```

Figure 3-7

The field name for the option number is OPTNO. Three SCRSWC records are set up. The following is entered:

	Record 1	Record 2	Record 3
FROM SCREEN	MENU	MENU	MENU
TO SCREEN	ORDENT	ORDMAINT	ORDING
OPT CODE	1	2	3
OPT FLD-NAME	OPTNO	OPTNO	OPTNO

If a SCRSWC record for an option code of 4 is created and the user enters a 4 on the screen, the switch occurs. If the user enters an option code that has not been defined to the SCRSWC record, then no switch occurs.

SAMPLES OF SWITCHING

Switching to a APPL program without displaying a screen first.

FROM screen = current screen
TO screen = *
TX Code =transaction code to switch to a non-**TABLES/AS** transaction program without a screen

INQ ONLY

Design a function code field in each row and specify INQ ONLY in Processing options. Specify function field name in Processing options.

•Specify the Selection Code Value desired to identify the particular row. Specify the keystroke (PF Key only)

•Specify the function code name in the option code field name. Specify the selection value as Opt. Value.

To IMS/CICS program without displaying a screen: Use if the program will make the initial screen display itself.

FROM screen= current screen
TO screen = *
TX Code = IMS/CICS transaction code desired
The PF Key or Opt code to effect the switch.

To clear a screen if the "clear" key is not available:

FROM screen = current screen
TO screen = current screen
PF Key = PF Key to cause clear
TX Code = *

To use the "clear" key to exit to IMS/CICS

FROM screen = current screen
TO screen = *
PF Key = 99
TX Code = spaces

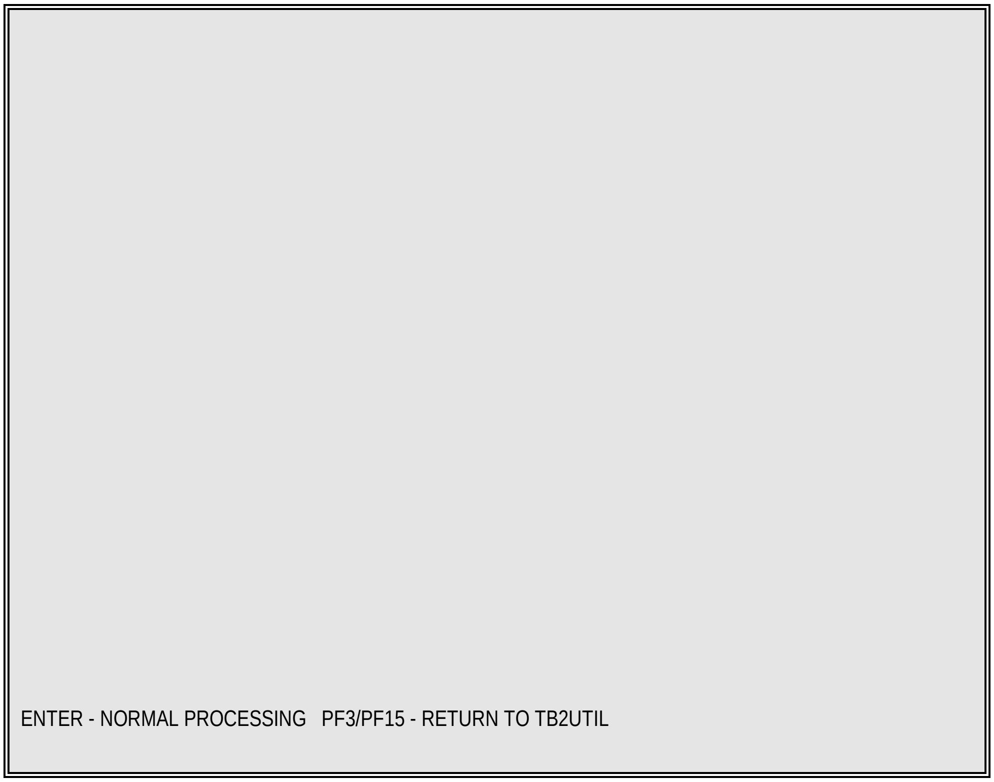
To a Non-Conversational MFS screen in IMS from the Screen Processor menu:

FROM = TB2MAINT
TO = MFS Mod name
PF Key = 01-24
TX = spaces
MFSMod = MFS Mod length

OPTION ===> 4 - DB2 SQL

This facility allows processing of SQL statements on-line. Any SQL statement other than a SELECT statement may be processed.

The screen shown in Figure 3-8 is used for input.



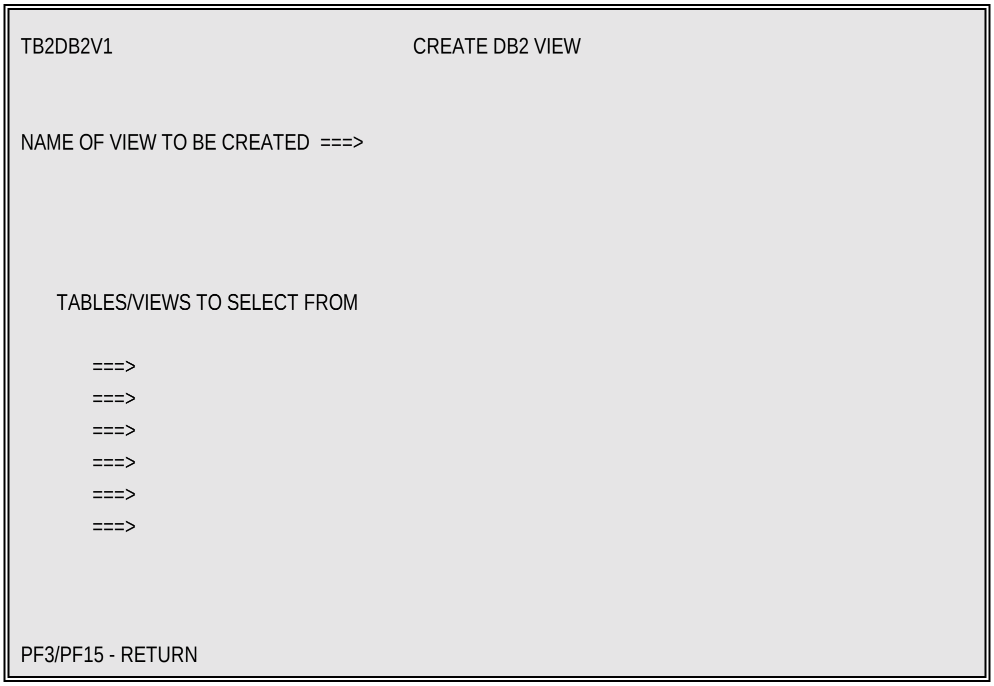
```
ENTER - NORMAL PROCESSING   PF3/PF15 - RETURN TO TB2UTIL
```

Figure 3-8

OPTION ===> 5 - DB2 VIEW

This facility may be used to create DB2 views. Views are created by selecting fields from other tables and views. A maximum of six tables and views may be specified in the selection process.

The screen shown in Figure 3-9 is used for input.



```
TB2DB2V1                              CREATE DB2 VIEW


NAME OF VIEW TO BE CREATED  ===>




   TABLES/VIEWS TO SELECT FROM

       ===>
       ===>
       ===>
       ===>
       ===>
       ===>



PF3/PF15 - RETURN
```

Figure 3-9

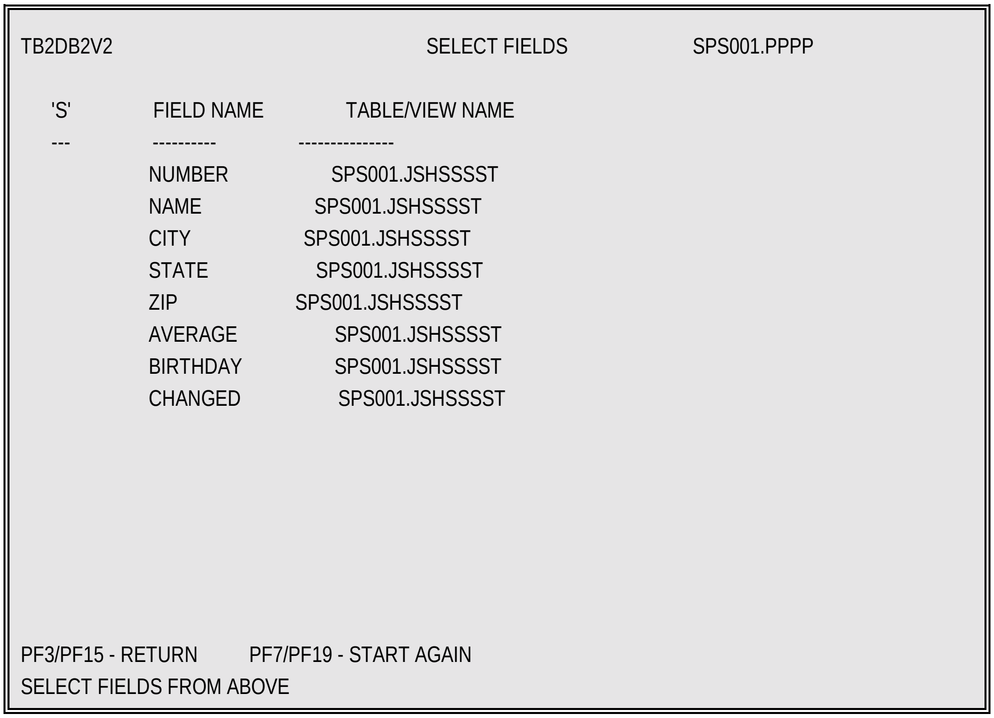
NAME OF VIEW TO BE CREATED

Enter the view name excluding the user authorization ID.

TABLES/VIEWS TO SELECT FROM

Fully qualified Names of up to six tables and views may be entered.

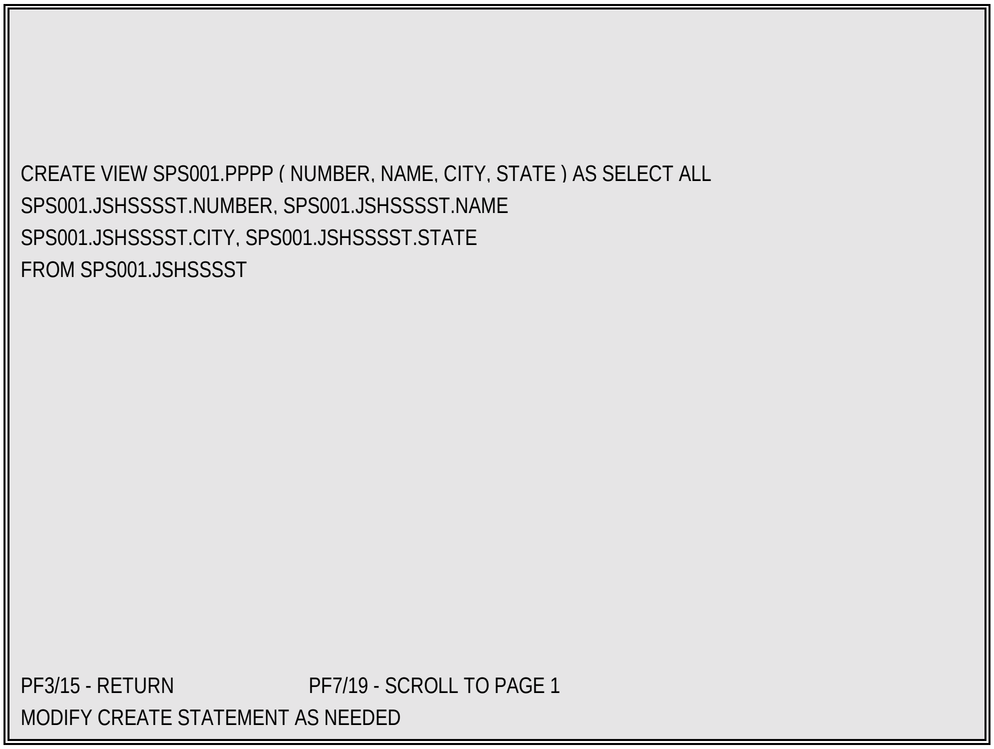
Press ENTER after the input and the TB2DB2V2 screen appears.
The screen shown in Figure 3-10 is an example.



```
TB2DB2V2                                  SELECT FIELDS             SPS001.PPPP

   'S'        FIELD NAME          TABLE/VIEW NAME
   ---        ----------      ---------------
              NUMBER              SPS001.JSHSSSST
              NAME               SPS001.JSHSSSST
              CITY              SPS001.JSHSSSST
              STATE              SPS001.JSHSSSST
              ZIP               SPS001.JSHSSSST
              AVERAGE             SPS001.JSHSSSST
              BIRTHDAY            SPS001.JSHSSSST
              CHANGED             SPS001.JSHSSSST







PF3/PF15 - RETURN      PF7/PF19 - START AGAIN
SELECT FIELDS FROM ABOVE
```

Figure 3-10

Enter selection code 'S' next to each field that is to be viewed.
The next screen shown is TB2DB2V3. See Figure 3-11.



```
CREATE VIEW SPS001.PPPP ( NUMBER, NAME, CITY, STATE ) AS SELECT ALL
SPS001.JSHSSSST.NUMBER, SPS001.JSHSSSST.NAME
SPS001.JSHSSSST.CITY, SPS001.JSHSSSST.STATE
FROM SPS001.JSHSSSST

PF3/15 - RETURN                PF7/19 - SCROLL TO PAGE 1
MODIFY CREATE STATEMENT AS NEEDED
```

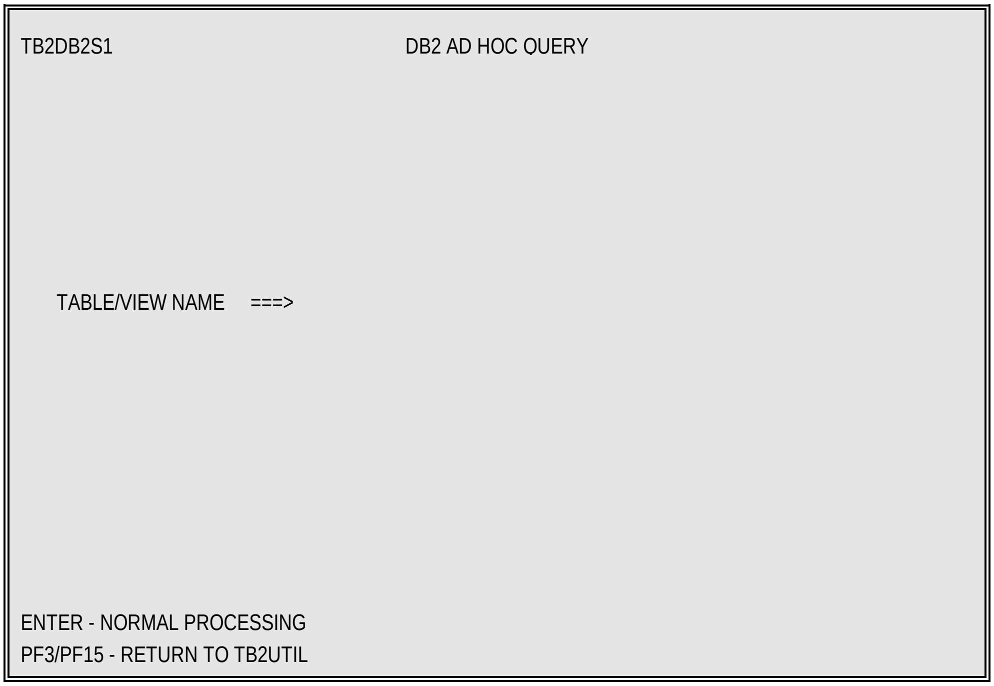
Figure 3-11

After modifying the create statement above, press ENTER.

If the syntax of the create statement is correct, a view is created for the specified name.

OPTION ===> 6 - DB2 QUERY

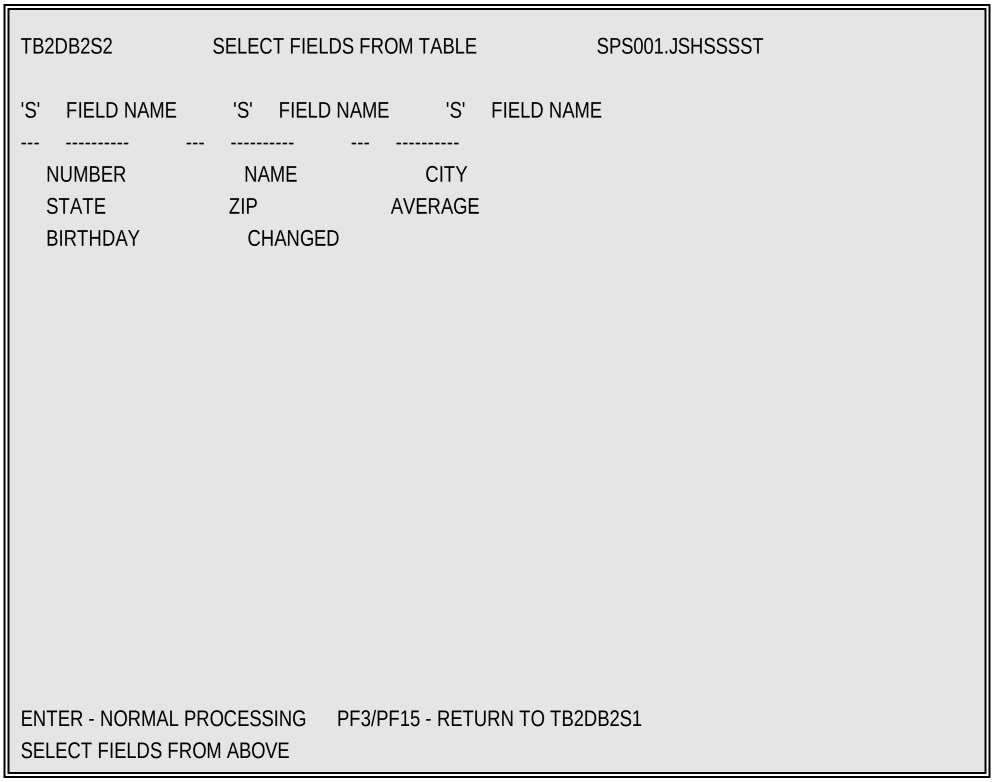
The screen shown in Figure 3-12 is used to perform ad hoc queries.



```
TB2DB2S1                                DB2 AD HOC QUERY




    TABLE/VIEW NAME   ===>




ENTER - NORMAL PROCESSING
PF3/PF15 - RETURN TO TB2UTIL
```

Figure 3-12

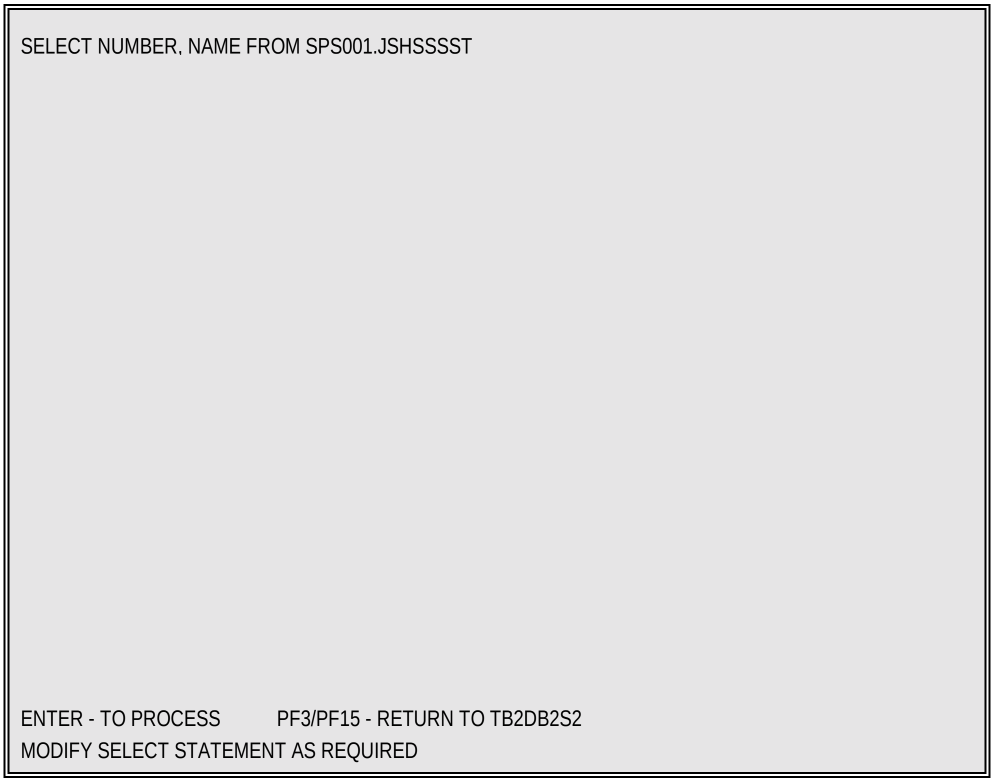
If a table name SPS001.JSHSSSST is entered on the previous screen, the following screen, Figure 3-13, with data appears.



```
TB2DB2S2            SELECT FIELDS FROM TABLE              SPS001.JSHSSSST

'S'   FIELD NAME       'S'   FIELD NAME       'S'   FIELD NAME
---   ----------       ---   ----------       ---   ----------
    NUMBER                 NAME                   CITY
    STATE                ZIP                    AVERAGE
    BIRTHDAY               CHANGED












ENTER - NORMAL PROCESSING    PF3/PF15 - RETURN TO TB2DB2S1
SELECT FIELDS FROM ABOVE
```

Figure 3-13

Enter selection code 'S' next to each field that is to be included for inquiry. The next screen shown, Figure 3-14, contains the SQL select statement.



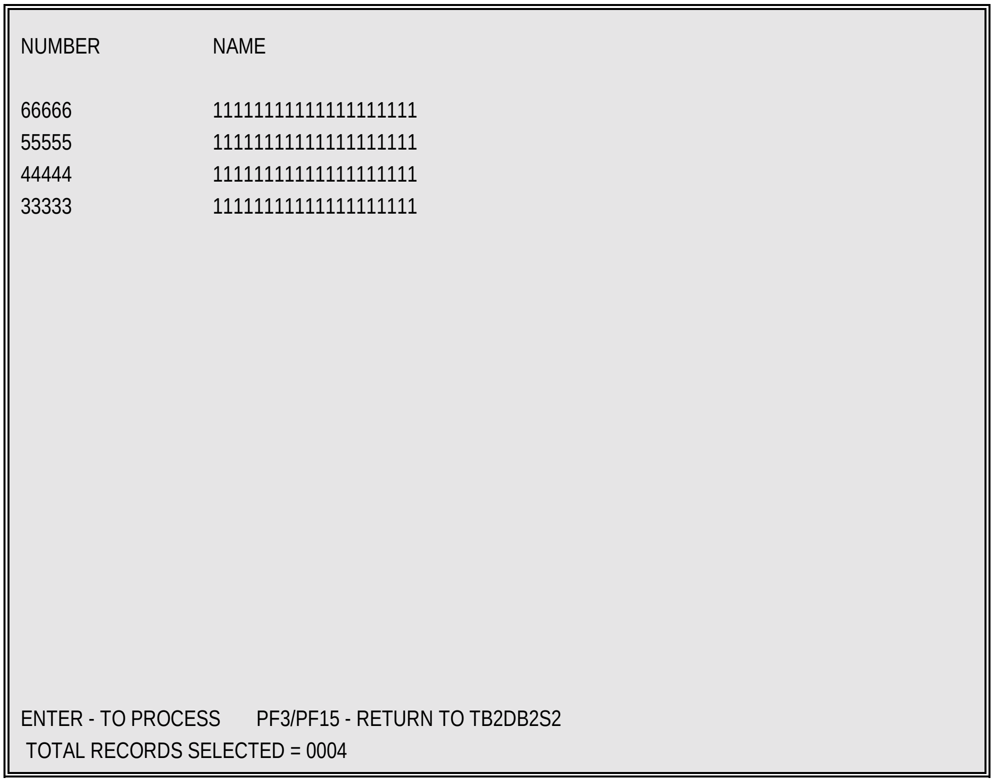
```
SELECT NUMBER, NAME FROM SPS001.JSHSSSST










ENTER - TO PROCESS          PF3/PF15 - RETURN TO TB2DB2S2
MODIFY SELECT STATEMENT AS REQUIRED
```

Figure 3-14

The select statement shown in Figure 3-14 may be modified to include a where clause for search values.

After modifying the select statement, press ENTER.

If the syntax of the select statement is correct, the statement is executed. The next screen shown, Figure 3-15, contains the results of the inquiry.



```
NUMBER              NAME

66666               11111111111111111111
55555               11111111111111111111
44444               11111111111111111111
33333               11111111111111111111















ENTER - TO PROCESS    PF3/PF15 - RETURN TO TB2DB2S2
 TOTAL RECORDS SELECTED = 0004
```

Figure 3-15

Inquiry results are displayed in a report format. Records selected are distributed into pages. One page is displayed each time the enter key is pressed. When there are no records left to be displayed, the select statement screen, Figure 3-13, is again shown for a new selection.

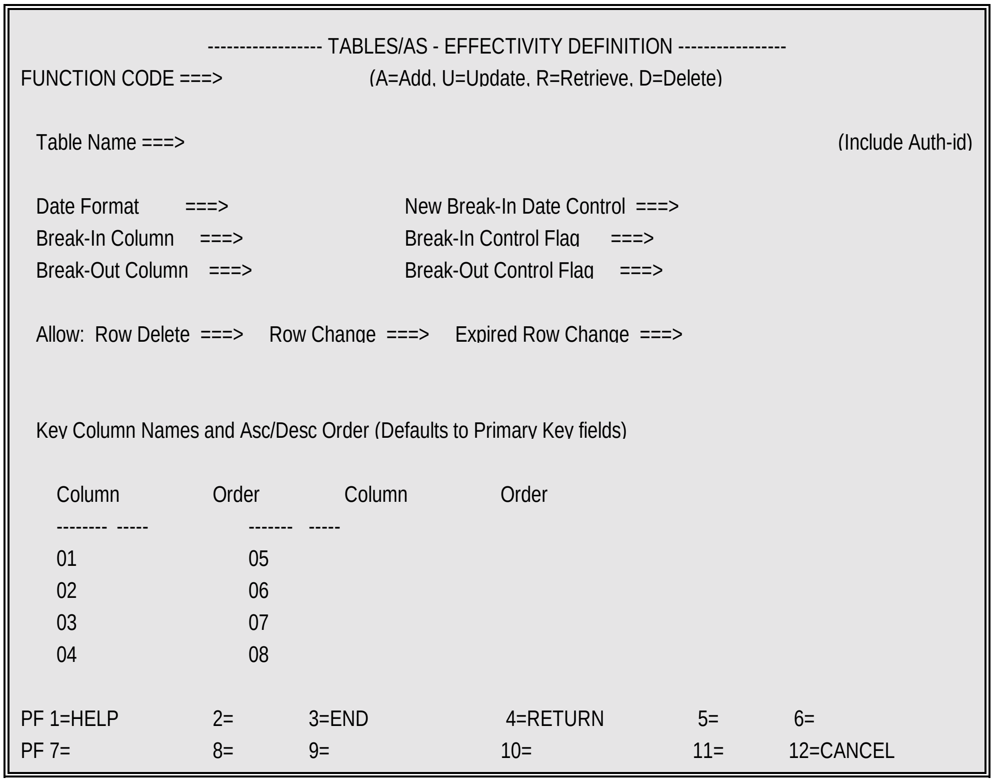
OPTION ===> 7 - DB2 TABLE EFFECTIVITY

Table effectivity provides a capability to maintain and access multiple versions of rows within a table in which there is always a start and there may be an expiration date attached to each row. The effectivity control facility provides the user with automated ways to validate and control storage of multiple versions of a row. It will ensure that overlaps, 2 rows effective at the same time, do not occur. You can ensure that gaps, periods where a row is not effective, do not occur. If desired, you may specify that gaps can occur in rows.

Any table under effectivity control must have a unique or primary index where start date (which for the purposes of this documentation will be called the Break-in date or BI date for short) is included as part of the unique index. A unique key may be made up of two or more columns including the BI date. The last column of the unique index must be the BI date.

The expiration date or the stop date, for purposes of this documentation will be called the Break-out date or BO date for short.

The screen shown in Figure 3-16 is used to define effectivity. EFFECTIVITY DEFINITION is defined only once for a given DB2 table. That definition is used in all TABLES/AS processes and screens to insure conformance to the rules specified.



```
                 ----------------- TABLES/AS - EFFECTIVITY DEFINITION ----------------
FUNCTION CODE ===>                    (A=Add, U=Update, R=Retrieve, D=Delete)

  Table Name ===>                                                              (Include Auth-id)

  Date Format        ===>                      New Break-In Date Control  ===>
  Break-In Column    ===>                      Break-In Control Flag      ===>
  Break-Out Column   ===>                      Break-Out Control Flag     ===>

  Allow:  Row Delete  ===>    Row Change  ===>    Expired Row Change  ===>


  Key Column Names and Asc/Desc Order (Defaults to Primary Key fields)

    Column             Order             Column             Order
    --------  -----                      -------   -----
    01                       05
    02                       06
    03                       07
    04                       08

PF 1=HELP             2=          3=END                4=RETURN             5=          6=
PF 7=                 8=          9=                   10=                  11=         12=CANCEL
```

Figure 3-16

When the screen shown in Figure 3-16 appears, the following information may be entered to define effectivity control of a DB2 table.

TABLE NAME

Enter the fully qualified name of the DB2 table (e.g.,auth-id.tblname).

DATE FORMAT

The date format code which specifies the DB2 stored format. It may be a date (see Appendix B) or a time stamp (date format 'S'). Both BI and BO dates must be in the same format.

The default date format is derived from the installation option module 'TBLOPTS'.

NEW BREAK-IN DATE CONTROL

This option is applied during the insertion of a new row into the table

C = Break-in (BI) date must not be < than current date

L =Break-in (BI) date must not be < than the latest BI for the row with the same key value

Blank = No specification

BREAK-IN COLUMN

The DB2 table start date column name for the default value is the name of the last column defined under the unique index.

BREAK-IN CONTROL FLAG

A =Always default to current date or time
D **=**Always default if not user supplied
U =Always user supplied
Default value is 'A'.

BREAK-OUT COLUMN

The DB2 table end date or the expiration date column name. The default value is a blank which indicates that there is no Break-out column existing in the table.

BREAK-OUT CONTROL FLAG

D =Automatically compute BO date and adjust others during Add. Allow change of BO date during Change function.

The default value is **A**.

See examples which follow.

ALLOW ROW DELETE

Y =Allow deletion

N = Do not allow

The default value is **Y**.

ALLOW ROW CHANGE

Y =Allow change of all non-key fields including BO date (expired BO effectivity change may not be allowed)

R =Allow change of all non-key fields excluding BO date

N =Do not allow any change

The default value is **Y**.

ALLOW EXPIRED ROW CHANGE

Y =Allow change of expired rows excluding keys
N =Do not allow any change
The default value is **Y**.

KEY COLUMN NAMES AND ORDER

Input is not required in these fields.

From the given table, the names of key columns and their order are automatically retrieved for effectivity processing. If there is a primary key, those column names are displayed. If there is a single unique index, those names are displayed if there is no primary key.

To reduce the DB2 catalog access time, enter the column names and order pertaining to the unique or primary key which contains the BI date.

EFFECTIVITY PROCESSING - EXAMPLES

Codes **D** and **U** allow a table to contain GAPS. Code **A** does not. Code **A** enforces the NO GAP criteria as rows are added and deleted. Code **D** operates similarly to Code **A** when adding rows. Delete and change operate differently. Codes **U** and **D** are the same for changes and deletes. The following examples show how Code **D** and Code **A** operate.

BO CONTROL CODE = D

Control Code **D** allows for the automatic calculation of BO date while adding a row and provides the user with the ability to change BO dates thereby allowing GAPS to be created within the table.

The **D** BO Control Code works in the following manner:

A.ADDING ROWS:

A BO date will be computed automatically. It is not acceptable as input for an Add.

Assume a table with the following BI and BO dates:

Bil---------------- Bo1 Assume a GAP exists between
Bi2 ---------------- Bo2 Bo1 & Bi2.
Bi3 ---------------- Bo3 Assume NO GAPS exist from
Bi4 ---------------- Bo4 Bo2 through Bo4.

a)Add a row with a date between Bo1 and Bi2 with a start date of BiX. The result in the rows affected will be:

Bil---------------- Bo1
BiX ---------------- Bi2-1
Bi2 ---------------- Bo2

b)Add a row with a date between Bil and Bo1 with a start date of BiY. The result in the rows affected will be:

Bi1 ---------------- BiY-1
BiY ---------------- Bo1 ** See BO date for Code **A** **
Bi2 ---------------- Bo2 The GAP between Bo1 and Bi2 remains

c)Add a row with a date between Bi3 and Bi4 with a start date of BiZ. The result in the rows affected will be:

Bi3 ---------------- BiZ-1
BiZ ---------------- Bi4-1
Bi4 ---------------- Bo4

d)Add a row with a date between Bi4 and Bo4 with a start date of BiA, where Bo4 is 9999-12-31. The result in the rows affected will be:

Bi4 ---------------- BiA-1
BiA ---------------- Bo4

B.DELETING ROWS:

Any row(s) being deleted will be removed from the table. There will be NO automatic adjustments to BI or BO dates in any other rows.

C.CHANGING BO DATES:

An entered BO date can NOT be less than the existing BI date for the row. An entered BO date can NOT overlap a BI date in another row.

BO CONTROL CODE = A

Use of this BO Control Code insures there will be NO GAPS in the table. By changing between BO Control Code **A** and **D** or **U**, it is possible to create GAPS in a table. The following examples show what would occur in a table that is currently under Code = **A** whether GAPS existed or not.

This Break Out Control Code works in the following manner:

A. ADDING ROWS:

A BO date will be computed automatically. It is not acceptable as input for an Add.

Assume a table with the following BI and BO dates:

Bil---------------- Bo1 Assume a GAP exists between
Bi2 ---------------- Bo2 Bo1 and Bi2.
Bi3 ---------------- Bo3 Assume NO GAPS exist from
Bi4 ---------------- Bo4 Bo2 through Bo4

a)Add a row with a date between Bo1 and Bi2 with a start date of BiX. The result in the rows affected will be:

Bil---------------- Bo1
BiX ---------------- Bi2-1
Bi2 ---------------- Bo2

b)Add a row with a date between Bil and Bo1 with a start date of BiY. The result in the rows affected will be:

Bi1 ---------------- BiY-1
BiY ---------------- Bi2-1 ** See BO date for Code **D** **
Bi2 ---------------- Bo2 The GAP between Bo1 and Bi2 is gone

c)Add a row with a date between Bi3 and Bi4 with a start date of BiZ. The result in the rows affected will be:

Bi3 ---------------- BiZ-1
BiZ ---------------- Bi4-1
Bi4 ---------------- Bo4

d)Add a row with a date between Bi4 and Bo4 with a start date of BiA. The result in the rows affected will be:

Bi4 ---------------- BiA-1
BiA ---------------- 9999-12-31

B. DELETING ROWS:

Any row(s) being deleted will be removed from the table. Deleting one or more contiguous rows will cause the BO date of the row just prior to the first contiguous row deleted to be changed to the BO date that was in the last contiguous row deleted, i.e.,

Delete rows with BI dates of Bi2 and Bi3.
The resultant table will be:

Bi1	----------------	Bo3
Bi4	----------------	Bo4

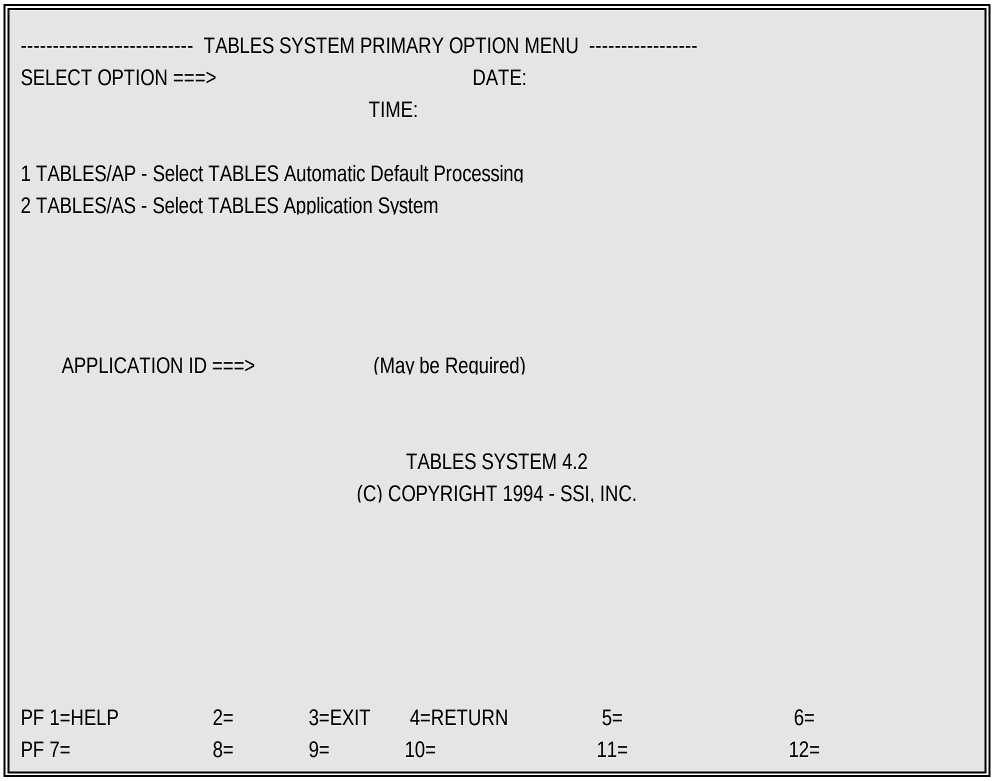
C.CHANGING BO DATES

You can NOT change a BO date when the BO Control Code = **A**. The BO is always calculated internally.

TABLES/AP

The automatic processing feature allows processing of any DB2 table known to DB2; also in IMS any IMS-DB segment, in CICS any VSAM file previously defined to TABLES.

To process automatic screens, you may use transaction code TSMM which displays the following screen:



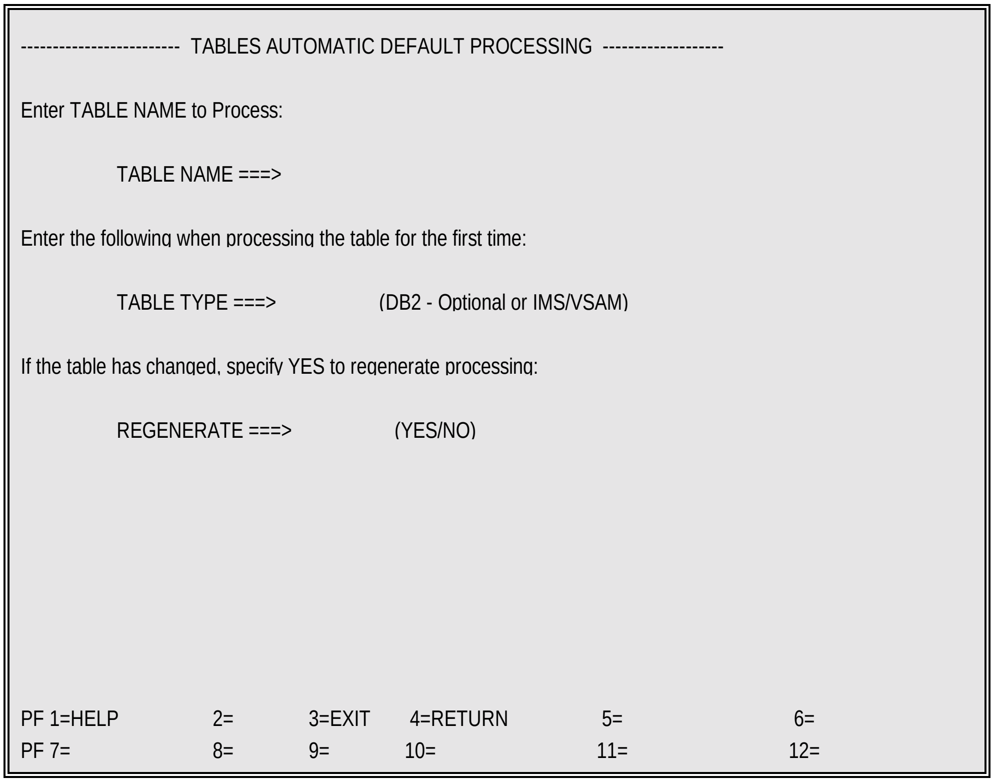
```
-------------------------  TABLES SYSTEM PRIMARY OPTION MENU  ----------------
SELECT OPTION ===>                                  DATE:
                                               TIME:

1 TABLES/AP - Select TABLES Automatic Default Processing
2 TABLES/AS - Select TABLES Application System




     APPLICATION ID ===>                (May be Required)



                                                  TABLES SYSTEM 4.2
                                            (C) COPYRIGHT 1994 - SSI, INC.






PF 1=HELP           2=          3=EXIT     4=RETURN          5=                    6=
PF 7=               8=          9=         10=               11=                   12=
```

Figure 3-17

Select Option 1 for the automatic screens. You may also use transaction code TSAP to select the automatic screen processing.

If you are on the **TABLES/AS** Primary Option menu, Option 8 can be used for automatic screens.

Upon selection of the automatic processing option, the following screen is displayed:



```
-----------------------  TABLES AUTOMATIC DEFAULT PROCESSING  ------------------

Enter TABLE NAME to Process:

          TABLE NAME ===>

Enter the following when processing the table for the first time:

          TABLE TYPE ===>          (DB2 - Optional or IMS/VSAM)

If the table has changed, specify YES to regenerate processing:

          REGENERATE ===>          (YES/NO)






PF 1=HELP        2=       3=EXIT    4=RETURN        5=            6=
PF 7=            8=       9=        10=             11=           12=
```

Figure 3-18

TABLE NAME

Table name must be fully qualified for a DB2 table, i.e., AUTHID.TABLE NAME.

Table name = VSAM table name (may be same as DD name)

Table name = DBD name + segname for IMS/DB (the + is required between the two names)

TABLE TYPE

Defaults to DB2. Enter VSAM or IMS if required.

REGENERATE

YES indicates that the default screen previously generated must be regenerated. This may be required if the table was changed since the screen was generated. The screen name which is automatically assigned to the given table and the date it was created are stored in a control table SCRDFT.

After pressing ENTER, a screen will be displayed with the ability to Add, Change, Delete and Inquire into the table name.

> **NOTE**
>
> For a complete description on using the automatic screens, see Screen Processing.

PF KEYS

PF1	Help
PF3	Cancel default screen generation
PF4	Cancel default screen generation
PF12	Cancel default screen generation

Section 4

TABLES/AS

Screen Processing

Screen Processing Overview

The Screen Processor executes all **TABLES/AS** and **TABLES/AP** screens (default and custom design) regardless of processing intent, i.e., add, change, delete, and inquire, regardless of supported access.

Screens can be created which maintain tables or databases, create test data, or provide inquiry capabilities. Validation criteria can be defined before any updates take place. For example, a customer number can be validated to be present on the customer file before updating the order file.

A screen may be defined as having add, change, delete, or inquiry capabilities in any combination. Security may be defined for a table that is being updated. A user may be given full maintenance and inquiry capabilities, or only a portion. See Section 1 - Security in the Technical Guide.

Transferring between screens with or without data is possible. Switching to multiple screens is permitted depending on field values or PF keys. Switching may occur to any screens designed with **TABLES/AS** or to or from other applications using BMS/MFS designed screens.

There are two methods supplied for entering the Screen Processor start-up screen; from the **TABLES/AS** Primary Option menu, Option 9, or from the installation's standard transaction code.

Other methods available for entering the Screen Processor are:

- Enter a unique transaction code to bypass the Screen Processor start-up screen sending you directly to a user screen executable through the Screen Processor. (See Technical Guide - Customizing Transaction Code.)
- By a switch from a programmed transaction to the Screen Processor by any of the methods defined in Screen Switching in Appendix G or H of the Technical Guide.
- From the installation's standard transaction code followed by a screen name entered in capital letters. This will display the named screen.

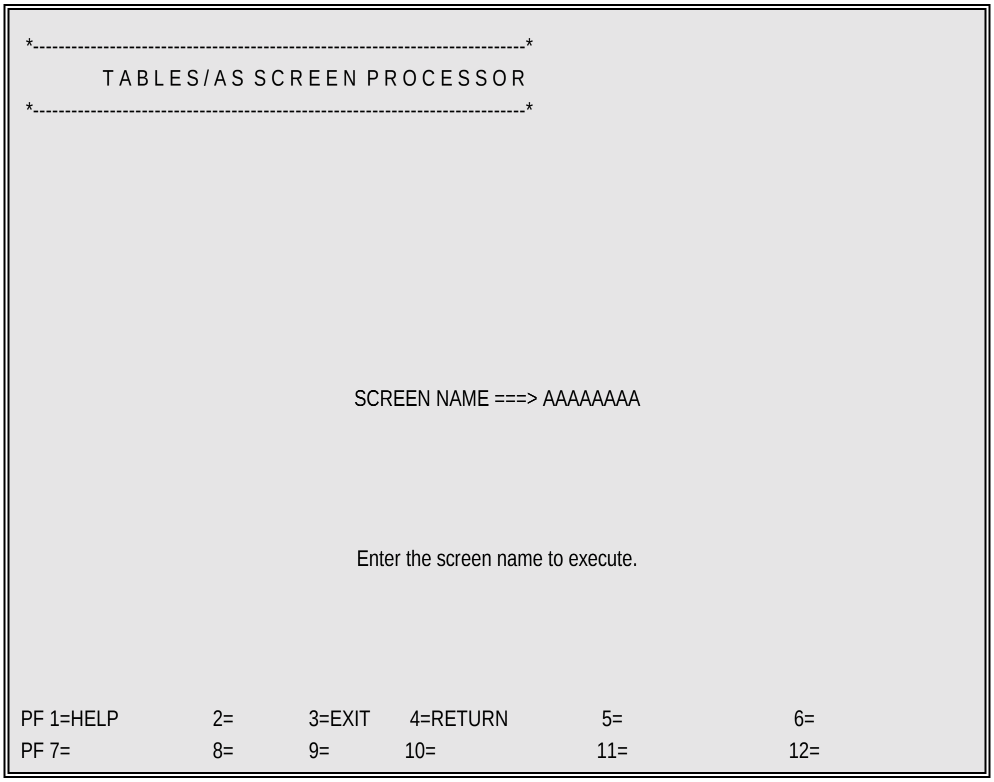
The **TABLES/AS** Screen Processor operates dynamically when entered from the **TABLES/AS** Primary Option menu (Option 9). This is considered to be the testing or development mode.

When the Screen Processor is invoked in any other way, it is considered to be in a production mode. As such, it will access screen load modules and static SQL modules, if they exist. Also, in the production mode, all TABLES control tables are accessed from memory loaded tables rather than dynamically from DB2.

Before a screen can be accessed and processed by the Screen Processor, it must have been "prepared." A screen is prepared by either using Option 5 from the **TABLES/AS** Primary Option menu or by using Option 9 with a Screen Name from the same menu. Using Option 9 prepares the named screen and switches to the **TABLES/AS** Screen Processor.

Primary Option Menu
Option ===> 9 - Screen Processing

Selecting Option 9 from the **TABLES/AS** Primary Option menu brings up the screen shown in Figure 4-1. Selecting Option 9 with a screen name, causes a prepare for the screen and then displays the screen



```
*------------------------------------------------------------------------------*
            TABLES/AS SCREEN PROCESSOR
*------------------------------------------------------------------------------*




                                   SCREEN NAME ===> AAAAAAAA



                                   Enter the screen name to execute.



PF 1=HELP        2=       3=EXIT    4=RETURN        5=             6=
PF 7=            8=       9=        10=             11=            12=
```

Figure 4-1

SCREEN NAME

If no screen name appears or a different screen name is to be accessed, enter the desired screen name.

The next screen displayed is the named screen. Position the cursor at the function code field and enter a function. The screen can now be processed depending on the security defined for both the screen and the user ID.

RETRIEVING RECORDS FOR INQUIRY

If the screen and user ID have inquiry capability, records may be retrieved from a table for inquiry purposes. By entering an **R** in the function code field and pressing ENTER, the system sequentially retrieves up to the maximum number of records allowed. The TABLES Screen Processor changes the function code from 'R' to '*'.

To retrieve records with specific values, enter the R and the specific values interested in. ON A MULTIPLE ROW SCREEN, THE FUNCTION CODE AND THE SEARCH VALUES ARE ALWAYS ENTERED IN THE FIRST ROW. All records with values equal to those specified are retrieved. If search operators have been defined, records can be retrieved based on the values defined.

The records retrieved are displayed on the screen in as many rows as are defined on the screen. If the record is defined as one row, the system displays one record at a time. If there are ten rows, the system can display up to ten rows at a time.

If more records are retrieved than can fit on one screen, pressing ENTER or the key defined to process, pages through the remaining records.

After R, the function code is changed to '*' after the retrieval is done. This function code can be changed to R, I, B, U, or A to perform subsequent function.

To clear the screen for next function, the clear key may be used.

ADDING RECORDS

If both the screen and user ID have Add capability, you may add records to DB2 Tables. To do this, enter an A in the function code field and fill in the necessary data on the screen. Press ENTER or the key that was defined for processing.

If there are record level validations defined for the add function, validation takes place before the record is added. If the validation fails, the record is not added and an error message appears.

After rows have been retrieved for update, you can replace the '_' with an 'A' to add a row. If there are empty rows on the screen, you can enter 'A' in the function field of the next empty row and fill in the data fields to add one or more new rows.

When a screen is first displayed, with all entry fields empty, entering 'A' in the 1st function code field puts the screen in Add mode. For multi-row screens, in Add mode, only adds can be done on that specific screen.

Clear your screen, at least the 1st row, and enter an R or U function to change the mode to Inquiry (R) or Update (U).

CHANGING OR DELETING RECORDS

A retrieve for update must first be done in order to change or delete a record. Enter a U in the function code field to specify multiple updates. To retrieve a record, enter the necessary field values in the record. To retrieve records sequentially, leave all fields blank.

The user ID and screen must have either Change or Delete capability. The TABLES Screen Processor changes from a U to '_' (underscore) when the records are retrieved for update. A function code of '_' can be changed to either D (delete), C (change), A (add), U (retrieve for update), or R, I, or B (retrieve for inquiry).

When changing a function code from U to A or U to R, on a multiple row screen, the new function code must be entered on the first row.

> **NOTE**
>
> During any processing, until after the last retrieved record is displayed or the screen is cleared with the clear key, entering an A in the first function code field does not change the processing mode to add.

To delete a record, change the '_' to D.

To change a record, change the '_' to C and make the necessary changes.

If a record is changed without a 'C' in the function code field, the system assumes a value of 'C' and will replace the record in the table.

To add a record, change the '_' to A and enter or change all required data.

If table level validations are defined for the function being performed, validation takes place before the record is processed. If the validation fails, the record is not updated and an error message appears.

REPEATING GROUPS

If a column, group of columns, or a row repeats on the screen, the first occurrence of that repeating group is examined when performing a retrieval (R, I, B, or U). If search operators exist, they are applied to the field(s) in the first occurrence. If there are no search operators, the equal condition (=) is applied to the field(s) in the first occurrence.

If there is one function code field defined for a screen containing repeating groups, that function applies to all repeating groups. For example, if three records are displayed on a screen after a retrieval for update (Function U) and a D is entered in the function code field, all three records are deleted.

If the function code is part of the repeating group, multiple updates may be performed and controlled at the row level. For example, if three records are to be deleted, a D is entered in the function code field for each line. If only Line 2 is to be deleted, only a D is entered in the function code field of Line 2, leaving the function code of Lines 1 and 3 an '_'.

NOTE

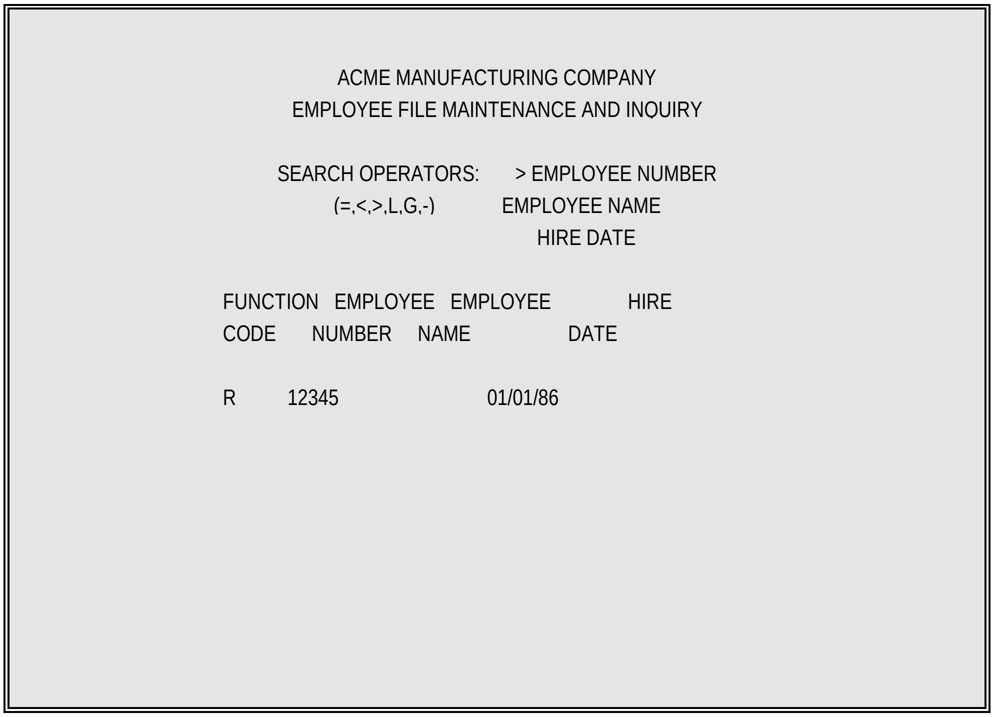
Repeating function code fields are used to provide a multiple update capability. They may also be used to identify a row for transferring data to another screen. **They do not serve any other useful purpose.**

SEARCH OPERATORS

VSAMNon-key fields: All search operators except J amd K are valid.

VSAMKey fields:Only operators
=, >, G are valid.
= requires entire key.
> can use either entire key or partial key.
G requires a partial key.

In the example created in Figure 4-2, three search operators were defined each relating to a different field on the screen. The search operator fields are connected to the corresponding screen fields in the mapping definition process. Let us suppose that we have entered the data as shown on the screen.



```
                ACME MANUFACTURING COMPANY
           EMPLOYEE FILE MAINTENANCE AND INQUIRY

         SEARCH OPERATORS:      > EMPLOYEE NUMBER
             (=,<,>,L,G,-)        EMPLOYEE NAME
                                     HIRE DATE

FUNCTION   EMPLOYEE   EMPLOYEE          HIRE
CODE      NUMBER     NAME          DATE

R        12345                  01/01/86
```

Figure 4-2

All records are retrieved with an employee number greater than 12345 and a hire date equal to 01/01/86. Even though a search operator was not entered for hire date, the system defaults to = (equal to the value of the search field).

If a field has no search operator and data is entered when performing a retrieval, it is included in the match criteria as an equal to condition. For example, if there was a field on the above screen called **Pay Rate** and 10.00 was entered, then the employee records would also have to have a pay rate equal to 10.00 in order to be retrieved.

MULTIPLE TABLES ON A SCREEN

The **TABLES/AS** Screen Processor processes multiple tables on the same screen. This should not be confused with REPEATING GROUPS. Each table can be processed individually. The resulting effect is having one screen sectioned into many smaller screens. Up to 14 tables may be defined on any one screen.

For example, a screen may be designed to add records to a DB2 table and to display a list of values from another table on the same screen.

EDITOR MODE PROCESSING and EDITOR PROGRAMS

Editor Mode processing should not be confused with the more simple use of an editor program. In Editor Mode, the **TABLES/AS** Screen Processor only reads and writes screens. **All** processing logic is controlled by a user-written program. After the screen is read, control is passed to the user-written program. When that program has finished its processing, it returns control back to the Screen Processor to display the screen.

A user-written ~~editor program~~, on the other hand, is called at various processing points by the **TABLES/AS** Screen Processor. In this case, the Screen Processor is in control of the flow from the screen input through editing, DB2 access and updating and finally screen output. The editor program would test of the processing point(s) desired, do some additional editing or calculations and then return control back to the Screen Processor.

In either case, the user-written program can perform all functions valid within a CICS or IMS program.

To specify an editor program for the Screen Processor to invoke, you enter the editor program name in the ALLOW FUNCTIONS or in the SWITCHING DEFINITION panels.

To specify EDITOR MODE, the control table SCREDT must be updated with a record to define the screen name and the user-program name. The SCREDT table may be updated using **TABLES/AP**.

The Screen Processor invokes the Editor Mode after the screen is read and it is determined that none of the conditions defined for screen switching exist.

Refer to **TABLES/AS** Technical Guide in the section of Online Maintenance for details on Editor Programming.

SCREEN PROCESSOR MESSAGES

The Screen Processor uses standard messages and text in the absence of user defined messages. If messages are to be overridden, they must be entered in the control table SCPMSG. The following is required for creating a message:

SCREEN NAMEName of the screen for which the message is applicable.

MESSAGE NUMBERNumber of the message to be overridden. Screen Processor message numbers begin with 'T'.
Example - 'T058, T065'

MESSAGE TEXT1 Up to 60 characters of user message.

MESSAGE TEXT2 Additional 19 characters of message as a continuation of Message Text1.

OVERRIDING DB2 ERROR MESSAGES

Normal DB2 errors are translated using DSNTIAR program (a DB2 standard program); however if these messages are to be overridden, this may be done at the screen level or system wide.

For screen level override, enter a four (4) digit numeric code in which the leading zero equates to the negative (-) sign and a corresponding message in the SCPMSG control table. For example, -911 should be equated to 0911. The screen name is the name of the screen to be processed.

For system wide override, enter a four (4) digit numeric code in which the leading zero equates to the negative (-) sign and a corresponding message in the SCPMSG control table. The screen name is a constant name called SQLCODE.

CUSTOMIZED GLOBAL ERROR MESSAGES

An ability is provided to change standard **TABLES/AS** messages for global use. Messages displayed provide additional information such as what table, what column, what PF Key, etc.

These global messages are sent with the product and reside in the Authid.AS-MESSAGES control table. All **TABLES/AS** messages, currently processed by the Screen Processor, including DB2 messages, may be customized. Use **TABLES/AP** to add or change a message in the AS_MESSAGES control table. The first five digits of the error message, as shown on the screen, should be used to log the global message.

A message in AS_MESSAGES can be up to 800 characters long. To store a message longer than 254 characters, a static DB2 module will have to be created to update the AS_MESSAGES table. A sample program resides in the SOURCE library to further customize AS_MESSAGES that you may modify for your use.

The sequence of searching for a customized message is as follows:

1. The SCPMSG control table is searched first by screen name and message number. If a customized message exists for a given screen, it is used and no further search is made.
2. For negative DB2 error messages, the SCPMSG control table is searched using 'SQLCODE' in the screen field with the negative DB2 message number. If a customized message exists, it is used and no further search is made. Otherwise, the standard DB2 messages, as returned by the DSNTIAR module is used.
3. The AS_MESSAGES control table is searched for the error message. If the error message number is found in AS_MESSAGES, it is used, but its impact is global—not just this screen, but all screens. If the AS_MESSAGES control table does not exist at the user's site, **TABLES/AS** will assume it to be a valid condition and proceed as if no further customization is required.

DEFAULT MESSAGES

T001	FUNCTION CODE REQUIRED
T002	FUNCTION CODE INVALID
T003	SCREEN NAME REQUIRED
T004	SCREEN NAME NOT FOUND
T005	CORRECT USERID IS REQUIRED
T006	RECORD NOT FOUND
T007	FUNCTION COMPLETED
T008	DUPLICATE KEYS ARE NOT ALLOWED
T010	UNABLE TO ACCESS EDITOR MODULE
T011	UNABLE TO ACCESS SCREEN
T012	TABLE NOT DEFINED FOR THE SCREEN
T013	GENERIC RETRIEVAL NOT ALLOWED
T014	ENTER SCREEN NAME & USERID
T015	UNABLE TO ACCESS TABLE DEFN PROGRAM
T016	UNABLE TO ACCESS EDITOR PROGRAM
T017	UNABLE TO FIND THE SCREEN RECORD
T018	UNABLE TO FIND THE SCREEN NAME RECORD
T019	UNABLE TO FIND THE FUNCTION FIELD
T020	UNABLE TO FIND THE 2ND FUNCTION FIELD
T021	FUNCTION CODE ENTERED IS INVALID
T022	UNSUCCESSFUL VALIDATION
T023	FILE IS NOT OPEN
T024	VSAM FILE INFORMATION NOT FOUND
T025	DATA DOES NOT SATISFY TABLE DEFINITION
T027	CHANGE OF KEY FIELD WAS ATTEMPTED
T028	SEARCH VALUES ARE REQUIRED FOR VSAM FILE
T029	RECORDS BEING SEARCHED ARE NOT FOUND
T030	SELECTED FOR INQUIRY
T031	SELECTED FOR UPDATE
T032	FUNCTION COMPLETED, ENTER NEXT SEARCH VALUES
T033	ADDITIONAL RECORDS ARE DISPLAYED FOR INQUIRY
T034	FUNCTION COMPLETED, ADDITIONAL RECORDS ARE DISPLAYED
T035	SEARCH OPERATOR FIELD IS NOT FOUND ON SCREEN
T036	MAXIMUM FILE CALLS PERMITTED EXCEEDING

T037	ALLOWED MAX SEARCHED
T038	RECORD LENGTH ERROR; CHECK DEFINITION
T039	PROCESSING FUNCTION INVALID
T040	ACCESS DENIED, VERIFY PROCESSING OPTIONS
T041	DATA FORMATS TO BE MAPPED ARE INCOMPATIBLE
T042	RECORD NAME TO BE MAPPED UNKNOWN
T043	DBD NAME IS REQUIRED TO ACCESS THIS SEGMENT
T044	RECORD TO BE CHANGED WAS NOT FOUND ON WORK DATA BASE
T045	MUST RETRIEVE ROW TO BE DELETED BY FUNCTION U
T046	UNABLE TO FIND THE MESSAGE FIELD
T047	RECORD TO BE MAPPED OUT NOT FOUND
T048	NO MAP RECORD, PF KEY, OPTION CODE, OR EXIT FROM TARGET
T049	MAP IS NOT DEFINED FOR THE SCREEN
T050	INITIATION SOURCE UNKNOWN, RETURN NOT POSSIBLE
T051	UNABLE TO FIND THE NEXT HIGHER LEVEL KEY
T052	ABNORMAL CONDITION ENCOUNTERED IN EDITOR
T053	NO ACTION HAS BEEN INDICATED
T054	UNABLE TO FIND THE OPTION CODE FIELD NAME
T055	UNRECOGNIZABLE FIELD TYPE (NOT=C,P,B,OR D)
T056	A FIELD OF ZERO LENGTH WAS DETECTED
T057	A QTY FIELD CONTAINS ERRONEOUS DEFINITION
T058	ERROR IN THE DATE INPUT/OUTPUT DEFINITION
T059	INVALID FILE NAME
T060	SECURITY VIOLATION
T062	VSAM FILE KEY FIELD SEARCH ON <,^, AND L NOT VALID
T064	ENTER SCREEN NAME TO BE PROCESSED
T065	NO DATA ENTERED TO PERFORM ADD FUNCTION
T066	ERROR-IN BUILDING SQL STATEMENT
T067	CURRENT PAGE NO
T069	RECORD TO CHANGE IS NO LONGER AVAILABLE
T070	UNABLE TO FIND SWITCHING SCREEN
T071	ENTER NEXT SELECTION
T072	THIS IS THE LAST PAGE
T073	PRESS ENTER KEY TO CONFIRM DELTE
T074	PRESS PFKEY NN TO CONFIRM DELTE

EFFECTIVITY CONTROL MESSAGES

T200 BI COLUMN NAME DEFINED IS INVALID

T201 KEY COLUMN NAME DEFINED IS INVALID

T202 BO COLUMN NAME MUST BE INCLUDED IN THE MAP

T203 BO DATE IS NOT ALLOWED AS USER INPUT

T204 DUPLICATE KEYS ARE NOT ALLOWED

T205 BI DATE IS REQUIRED TO PROCESS

T206 CURRENTLY EFFECTIVE ROW HAS BLANK BO DATE

T207 BI DATE IS NOT ALLOWED AS USER INPUT

T208 PREVIOUS ROW MUST BE BROKEN OUT

T209BI DATE ADDED NOT < THAN ANOTHER EFFECTIVE ROW

T210BO DATE ADDED NOT < THAN ANOTHER EFFECTIVE ROW

T211 BO DATE INVALID, EFFECTIVITY GAP EXISTS

T212 BI DATE MUST BE > THAN BI DATE PREV. ROW

T213 BI DATE CANNOT FALL WITHIN EXISTING BI-BO DATES

T214 DELETION OF ROW IS NOT ALLOWED

T215 CHANGE OF ROW IS NOT ALLOWED

T216 CHANGE OF KEY COLUMN VALUE NOT ALLOWED

T217 CHANGE OF BO DATE VALUE NOT ALLOWED

T218 BO DATE IS EXPIRED, ROW CANNOT CHANGE

T219 BO DATE CANNOT BE < THAN BI IN SAME ROW

T220 CHANGE OF BO DATE TO BLANKS NOT ALLOWED

T221 ROW TO BE CHANGED NOT FOUND IN THE TABLE

T222BI DATE CHANGED NOT < BI DATE NEXT ROW

T223 A BO DATE IS REQUIRED

T224 BO DATE MUST BE < THAN BI DATE NEXT ROW

T225 EFFECTIVITY GAP MUST BE GREATER THAN ZERO

T226 EFFECTIVITY BI DATE IS INVALID

T227 EFFECTIVITY BO DATE IS INVALID

T228 ROW TO BE DELETED NOT FOUND IN THE TABLE

T229 A NEW BO DATE IS INVALID

T230 BI DATE ADDED MUST BE > THAN LATEST BI DATE

T231 BI DATE ADDED MUST BE > THAN CURRENT DATE

T301 VALIDATION BREAK-IN DATE IS INVALID

T302 VALIDATION BREAK-OUT DATE IS INVALID

T303 ALL KEY COLUMNS ARE REQUIRED FOR VALIDATION

T304 OPERATOR OTHER THAN = NOT VALID ON KEY COLUMN

TABLES/AS

Appendices

Appendix A

Attribute Characters

STANDARD ATTRIBUTE CHARACTERS

A Input/output field, low intensity
B Input/output field, high intensity
C Input/output field, non-displayable
D Output field, low intensity
EOutput field, high intensity
FOutput field, non-displayable
GInput/output protected field, low intensity, modified data tag on
HInput/output protected field, high intensity, modified data tag on
IInput/output protected field, non-displayable, modified data tag on (used to hide fields on the screen)

EXTENDED ATTRIBUTE CHARACTERS

B	Blink
R	Reverse video
U	Underline
Blank	Device Default

COLOR ATTRIBUTE CHARACTERS

B	Blue
R	Red
P	Pink
G	Green
T	Turquoise
Y	Yellow
N	Neutral
Blank	Device Default

CURSOR ATTRIBUTE CHARACTERS

E	Position cursor on this field
Blank	Do not position cursor on this field

Appendix B

Date Format Codes

1	MMDDYY	
2	MMDDYYYY	
3	DDMMYY	
4	DDMMYYYY	
5	YYMMDD	
6	YYYYMMDD	
7	MM/DD/YY	
8	MM/DD/YYYY	
9	DD/MM/YY	
A	DD/MM/YYYY	
B	YY/MM/DD	
C	YYYY/MM/DD	
D	YYDDD	
E	YYYYDDD	
F	YY/DDD	
G	YYYY/DDD	
H	DD-MMM-YY	('01-JAN-90',L=9)
I	DD-MMM-YYYY	('01-JAN-1990',L=11)
J	MON DD,YYYY	('JUNE 1, 1990',L=18)
K	MMM DD,YYYY	('OCT 1, 1990',L=12)
L	YYYY-MM-DD	(DB2 FORMAT, ISO, JAPANESE)
M	DD.MM.YYYY	(EUROPEAN STD)

Appendix C

Screen Processing Controls

SEARCH/COMPARE OPERATORS
< Less than
> Greater than
= Equal to
L Less than or equal to
G Greater than or equal to
┐ Not equal to
J Not like (DB2 interface only)
K Like (DB2 interface only)

FUNCTION CODES
R, B, IRetrieve for inquire only
U Retrieve for update
AAdd
C Change
D Delete

Appendix D

Error Messages

The lower portion of the screen will display error messages pertinent to problems encountered while processing the screen. Additionally, the results of the various options selected will also be displayed.

Screen Design Functions

DISPLAY NOT POSSIBLE, RECORD DELETED
A screen description was defined for that screen name, but PF6/12 was used to back out of the paint screen. The system has just deleted the residual screen description.

ENTER PF KEY FOR HELP
Enter the PF Key corresponding to the option with which help is needed.

OPTION CODE INVALID
Valid codes are listed on the screen.

OPTION CODE REQUIRED
Valid codes are listed on the screen.

SCREEN ALREADY EXISTS
Trying to add a screen which already exists.

SCREEN NAME NOT FOUND
A defined screen name is required.

SCREEN NAME REQUIRED
A valid screen name is required.

EDITS EXIST, SCREEN CANNOT BE DELETED
First delete all existing edits before deleting a screen. Use Option **5** on the Edit Definitions menu.

FIELD START INDICATORS WERE INSERTED
A literal was defined with no field start indicator, therefore the system inserted them where necessary.

NAME RECORD NOT FOUND, SCREEN RECORD COPIED
Field names must be defined on the From screen prior to using this option. The system copied only the screen format.

SCREEN EXISTS BUT TABLE NOT FOUND, DELETE THE SCREEN.
A table used on this screen no longer exists.

TABLE DEFINITION NOT FOUND
A table name was specified as a base table for screen design and the table was not found in catalogue.

UNSUPPORTED COLTYPE FOUND
A column type selected for screen design is not supported, one example is a floating point column in a DB2 table.

NO OF COLUMNS EXCEEDS MAX. 120
The total number of columns in the selected table exceeds 120 which is the maximum allowed for a base table.

VALUE MUST BE YES, NO, SEL OR BLANK
To select a default starting screen, value entered must be YES, NO, SEL (to select fields) or blank.

Line Commands

ERROR - NO. OF COLUMNS TO BE SHIFTED IS INVALID
This message occurs when SL or SR commands were entered and the number of columns to shift was greater than eighty (80) or the number of columns to shift was not numeric.

ERROR - DATA TO BE MOVED OR COPIED OUTSIDE OF SCREEN
This message occurs when the number of lines to be moved or copied would go off the top or bottom of the screen. For example, A move M of a single line after A the last line of the screen (line 24). Copying five (5) lines before B line three (3) of the screen.

ERROR - CONFLICTING COMMANDS, ONLY ONE KIND CAN BE ENTERED AT A TIME"
This message occurs when more than one move, copy or repeat command is entered. Also, when any combination of move, copy, repeat or shift commands are entered together. Enter number of shift commands at one time, but not in combination with move, copy or repeat.

ERROR - INVALID COMMAND FORMAT
This message occurs when a command was entered incorrectly.

ERROR - INCOMPLETE COMMAND
This message occurs when one or more of the entries required for a command were not entered. For example, a C was entered on one line but there was no A or B entered to indicate where to move the line.

Change Options

EXTENDED ATTRIBUTES NOT SPECIFIED
The screen description has been defined indicating that extended attributes are not to be used. Change the screen description if extended attributes are now valid for this screen.

OPTION CODE INVALID
Self-explanatory

SCREEN NAME REQUIRED
Self-explanatory

SCREEN NAME NOT FOUND
Self-explanatory

CANCELED
PF12 was invoked to return without changes.

XXXXXXXX CHANGED
Changes made have been applied.
Where, XXXXXXXX = screen name.

FUNCTION COMPLETED
Option 7 (FIELD NAMES) or Option 8 (RESEQUENCE) have been successfully completed.

EDITS EXIST, FIELD CANNOT BE DELETED
While in Option 2 (SCREEN FORMAT), an attempt was made to remove one or more input/output fields which have edits defined. None of the changes have been made. Delete edits for the fields being deleted and then delete the fields from the screen.

XXXXXXXX ERROR IN LINE 1 COL 1
An input/output field has been defined which ends in the last (BOTTOM RIGHT) position of the screen. When this condition occurs, a valid field-start character must be in the first (TOP LEFT) position of the screen.

FLDID CANNOT CHANGE
An attempt was made to change a FIELD ID (FLDID) value. A FIELD ID value is the one or two alphabetic character(s) following the input/output field-start character. None of the changes have been made.

CONFLICTING FIELD DEFINITION=FIELDNAME
An input/output field format was changed which would make it incompatible with a field of the same name on one or more screens.
Either change the field name or correct the definition on this screen so it conforms to the format defined on the other screens.

FIELDID TABLE FULL - PLEASE REDO IN ADD MODE
A screen which has more than thirty-five (35) one character fields can not be changed.

INVALID COLOR ATTR. ONLY P, B, G, Y, T, R, N ALLOWED

INVALID EXTENDED ATTRIBUTE. ONLY B, R, U ALLOWED

INVALID STANDARD ATTRIBUTE. ONLY A THRU I ALLOWED

Resequence Fields in the Copy Code

FLDID NOT FOUND
This message occurs when a **FLDID** is entered which does not exist for this screen.

Copy a Screen

FROM SCREEN IS REQUIRED
Field: From screen
Enter the name of the screen to be copied from.

FROM SCREEN DOES NOT EXIST
Field: From screen
The screen name entered must be an existing screen - Reenter.

MUST INDICATE THE AMOUNT OF DATA TO BE COPIED
Field: Screen format only
Specify one option, for example, screen format only, screen format + field names, etc. Place an X beside the desired option.

Primary Option Menu

APPLICATION ID MUST EXIST IN THE TXNTBL CONTROL TABLE
Using the **TABLES/AP** (Option 8) maintenance automatic default processing function, verify that the application name exists in the control table TXNTBL. This can be accomplished by entering the table name authid.TXNTBL on the **TABLES/AS menu.**

APPLICATION NAME REQUIRED
Enter a valid application name. Check spelling or find out the correct application name to use.

APPLICATION NAME MUST BE A VALID NAME
An application has been entered which has not been previously established. Check spelling, find out the correct name to use, or establish the name as a valid application name thru **TABLES/AP** (Option 8) automatic default processing function.

OPTION CODE REQUIRED
Enter Option Code 1 through 9.

OPTION CODE INVALID
An option code not defined in list of options for this menu has been entered.
****
SCREEN NAME REQUIRED
Enter a screen name. For Options 3 or 4, the name entered must be a name previously used for a screen which was designed with the screen design functions. For Option 1, this can be a new screen name, if designing a new screen, or it must be a name previously used. Check spelling or find out the correct screen name to use.

SCREEN RECORD NOT FOUND
A screen name has been entered which has not been previously assigned. Check spelling or find out the correct name to use.

Edit Definition Functions Menu

NO CONTROL FIELDS ARE DEFINED - OPTION 2
At least one field on the screen has to be defined as a controlling field in relational edits to use this option. Use Field Edit Definitions to identify controlling field.

NO EDIT DEFINITIONS EXIST FOR THIS SCREEN - OPTION 5
Edits must first be defined for the screen before performing these options.

OPTION CODE INVALID
Valid codes are listed on the screen.

OPTION CODE REQUIRED
Valid codes are listed on the screen.

SCREEN MODULE IS NOT FOUND
Either the screen has not been prepared for test or the program library may have gone into extents. Stop and start the message region to resolve the problem for the second case.

SCREEN NAME INVALID
A valid screen name is required to perform any of the options.

SCREEN NAME REQUIRED
A valid screen name is required to perform any of the options.

Field Edits

ABOVE INDICATES THE CURRENT DEFINITIONS
This is not an error message. If defining edits for the first time for this field, these are either the default edits or this field has edits defined for another screen. Pressing the ENTER key will activate these edits for this field on this screen.

ALPHA CHECK & FORMAT CANNOT BOTH BE Y
These two edits are mutually exclusive.

DATE REL. MUST BE BLANK OR = < L > OR G
See the explanation of each of these characters listed earlier in this section under date field edits.

FIELD MUST BE Y OR N
A response of other than a Y or N value has been given. Correct and press ENTER.

FIELD NAME NOT DEFINED FOR THE SCREEN
The field name entered as a special message field must be a predefined field on this screen.

FIELD TOO SHORT FOR STANDARD MESSAGES - CUSTOMIZE. MAX STD MESSAGE IS 73
A special message field must be defined as having a length of 73 or greater in order to utilize the system's generic error messages. If it's defined as less than 73, the error messages must be customized.

FORMAT LENGTH MUST BE EQUAL TO THE FIELD LENGTH

The format specified must be the exact length of the field, including the special characters.

FIELDS LONGER THAN 20 CHARACTERS CANNOT BE FORMATTED.
The maximum length alphanumeric field that can be formatted is 20 characters.

FORMAT IS INVALID
If Y was specified to format check, the format desired must be entered.
If N was specified to format check, the format must not be entered.

VALIDATION ERROR_SCRVAL RECORD NOT FOUND
Edit definitions being changed are not found. Delete the field edits and reestablish using the field edit definition panel.

SPECIAL MESSAGE FIELD CANNOT BE COMMON MSG FIELD
The field name being used as a special message field has already been defined as a common message field. Change one or the other.

USE CUSTOMIZE OPTION & CLEAR ALL MESSAGES
When changing the customize message option from Y to N, first delete the customized messages using the CUSTOMIZE MSGS SCREEN.

Discrete Value Entry Screen

FORMAT MUST BE NUMERIC OR VALUE NOT LONG ENOUGH
The discrete value entered must be numeric and it must be the same length as the field. If more than one discrete value is entered they must be separated by a comma.

INVALID DATE FORMAT OR LENGTH NOT EQUAL TO FORMAT
The date value entered must be the same length as the field and it must be of the same format, i.e., if the date is defined as MMDDYY then **860101** would be invalid.

DECIMAL POINT LOCATION IS INVALID
Discrete values for quantity fields must contain the same number of decimal positions as defined on the Field Edit screen.

FORMAT LENGTH DOES NOT MATCH FIELD LENGTH
The discrete value entered must be the same length as the field. If more than one discrete value is entered they must be separated by a comma.

Relational Edit Rules

COMPARE FIELDS MUST BE SINGLE VALUED OR BELONG TO THE SAME GROUP
The field name of a field in a repeating group other than the repeating group of the control field has been used as the name for compare FLD1 or compare FLD2. Check the spelling of the field name used or remove the name from the condition.

CONTROL FIELD NUMBER ENTERED MUST MATCH THE LIST
Some value for "ENTER CONTROL NUMBER DESIRED" has been entered, but it does not match any of the numbers in the list shown on this screen.

RELATIONAL COMPARE FIELDS MUST MATCH SINGLE VALUE CONTROL FIELD
A field name of a field in a repeating group as compare FLD1 or compare FLD2 has been used and the control field is a non-repeating group field. Either use the repeating group field as the control field or, check the spelling of the field name entered.

RELATIONAL CONTROL FIELD NUMBER IS REQUIRED
No value has been entered for "ENTER CONTROL NUMBER DESIRED".

RELATIONAL FIELDS MUST MATCH SINGLE VALUED CONTROL FIELD
This is a control field which is a non-repeating group field and fields have been selected within a repeating group for use in this relational condition. Either use the repeating group field as the control field or go back and select the correct non-repeating group fields.

THE SPECIFIED CONDITION IS NOT DEFINED, CANNOT PROCESS
A condition number for change has been specified which does not exist. Use view relational edits to see the conditions which are defined.

THE SPECIFIED CONDITION IS NOT DEFINED, CANNOT DELETE
A condition number for delete has been specified which does not exist. Use view relational edits to see the conditions which are defined.

Define Field Values for Conditions

A DATE FIELD CAN ONLY BE COMPARED TO ANOTHER DATE FIELD
An alphanumeric or numeric or quantity field name for FLD1 or FLD2 has been used and the screen field selected to participate in the relational condition is a date field. Check logic, field names and spelling.

ADDED FIELD NAMES MUST HAVE BEEN PREVIOUSLY SELECTED
A field name has been entered in the left most column which was not in the list of field names selected for this condition. To include another field, PF12 must be entered to select that field using a P, A, or C. Check spelling also.

ALPHA & NUMERIC FIELDS CAN ONLY BE COMPARED TO OTHER ALPHA & NUMERIC FIELDS
A date field name or a quantity field name for FLD1 or FLD2 and the screen field selected to participate in the relational condition is either alphanumeric or numeric. Check logic, field names and spelling.

A QUANTITY FIELD CAN ONLY BE COMPARED TO ANOTHER QUANTITY FIELD
An alphanumeric or numeric or date field name for FLD1 or FLD2 and the screen field selected to participate in the relational condition
is a quantity field. Check logic, field names and spelling.

DECIMAL POINT NOT ALLOWED
A decimal point has been entered in one of the values and the field was defined with no decimal points. Remove the decimal point.

FIELD MUST BE NUMERIC OR NOT LONG ENOUGH
Specific values entered must be the same length as the length defined on the
screen. A value in FLD1 or FLD2 which is shorter in length than defined on the screen has been entered or an alpha character has been entered in a numeric value.

FIELD MUST BE Y OR N
A value other than Y or N to the question at the top of the screen has been entered.
Enter either a Y or N.

INVALID A/N FORMAT
A value in FLD1 or FLD2 which has a different format than the format you specified in defining field edits for this field has been entered. Check
entry or check the format defined.

INVALID DATE FORMAT OR LENGTH NOT EQ TO FORMAT
The date entered for FLD1 or FLD2 does not conform to the date format specified when defining field edits for this field.

MULTIPLE DECIMAL POINTS NOT ALLOWED

A quantity field with more than one decimal point has been entered.

RELATIONAL OPERATOR CAN ONLY BE =, ¬, <, >, L, G, P, R or A
A valid relational operator has not been entered. Enter one of the values
shown.

RELATIONAL OPERATOR R, REQUIRES A (-) AT THE END OF THE FIRST FIELD VALUE
Operator R signifies entry of a range of values or field names. There must be a (-) separating the two values/names entered. Only the first value was entered. Either use a different operator or enter two values/names
separated by the hyphen (-).

2ND VALUE OF RANGE MUST BE GREATER THAN 1ST VALUE
A specified a range of values has been specified where the 2nd (to) value is not greater than the 1st (from) value. Check the values entered.

UNACCEPTABLE INPUT CHARACTER AT THE END OF THE FIRST COMPARE VALUE
The value entered in FLD1 does not have a single quote following the last
character. The number of characters between the single quotes must be the same number of characters as the field length defined on the screen.

VALUE MUST BE A FIELD NAME OR ENCLOSED IN SINGLE QUOTES
A specific value was entered for FLD1 or FLD2 and the value was not enclosed in quotes. A field name in FLD1 or FLD2 has been specified which is not a field named on the screen. Check spelling.

Record Mapping Option Menu

OPTION CODE REQUIRED
Specify any option code.

OPTION CODE INVALID
The option code entered is not one of the available options for this screen.

MAP-NO. IS NOT REQUIRED FOR ADD FUNCTION
Option 1 has been selected and a map number has been entered.
Blank the map number field to proceed. Map numbers are generated automatically by the product in the define function.

Record Mapping Definition

DATE CODE MUST BE ABSENT FOR FIELD TYPE NUMERIC
A numeric screen field has been related to a record field and have supplied a date code for the record field. If the screen is not defined as a date field, a date code cannot be used for the record field.
Remove the date code or fix the screen or use the proper FIELD SEQ NO. value.

FIELD VALUE CANNOT BE NON-NUMERIC
A value has been entered with other than digits in the MAP SEQ NO. field.
Correct the value entered.

MAP SEQUENCE NO. DOES NOT MATCH SCREEN FIELD SEQUENCE
A value has been entered in MAP SEQ NO. which does not exist in the list under FIELD SEQ NO. Correct the number entered.

OPERATOR FIELD NAME MUST BE ON THE SELECTED SCREEN
An operator field name has been entered which does not exist on the screen.
Check the spelling of the name.

Define Mapping Source

INPUT IS REQUIRED FOR THE OPTION SELECTED
A table name and/or a copy code name and/or a **VSAM** filename has been specified but an X was not entered to indicate the source for mapping.

OPTION CODE INVALID
A value other than X was entered next to **TABLES/AS** definitions or next to copy code source input.

ONLY ONE TYPE OF MAP CAN BE DEFINED AT A TIME
More than one X was entered to indicate the source for mapping.

** XXXXXXX TABLE DEFINITION DOES NOT EXIST IN TABLES/DB2 CATALOG*
An invalid table name was entered.

** XXXXXXXX COPY CODE MEMBER MUST BE EXTRACTED USING UTILITY PROGRAM*
Either the copy code member name was misspelled or JCL401MC was not run.

*** Note: XXXXXXXX = TABLE NAME OR COPY CODE MEMBER NAME.**

CONVERSION TO NUMERIC FORMAT IS NOT POSSIBLE

A date code containing slashes (/) for the MAPPED DATE FORMAT and the record format is numeric. There is inconsistency. Check the date code used or the wrong FIELD SEQ NO may have been entered.

VALIDATION ERROR_SCRVAL RECORD NOT FOUND
One or more fields being mapped have lost their edit definition even though fields appear to be marked as being edited.

This problem can be cleared by switching to the edit definition panel and deleting the field edits for all fields that are being mapped and then redefine the edits.

Allow Functions

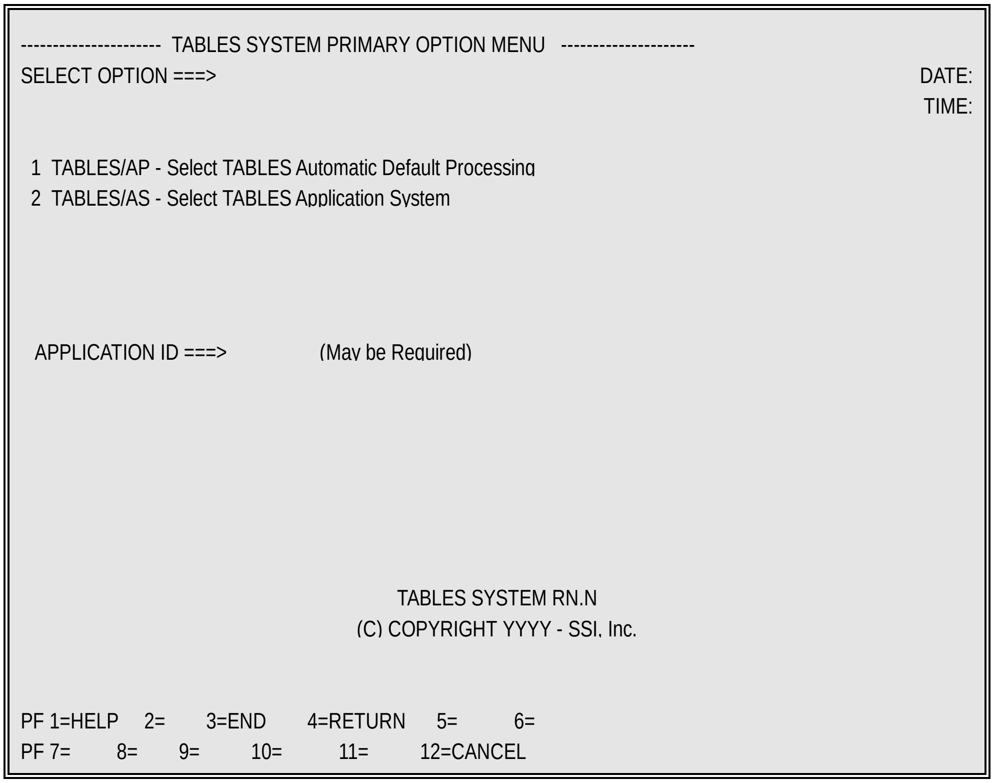
INVALID NAME WAS ENTERED, VALID NAME SHOWN, VERIFY AND REENTER
The MSG field name entered does not exist on the screen. The name of a field defined as a message field in FIELD EDITS is shown. If OK, press ENTER. If not OK, review names and enter an existing MSG field name.

INVALID NAME WAS ENTERED, VERIFY AND REENTER
The Function Code field name does not exist for this screen.

INVALID KEY, CANNOT BE SAME AS THE KEYSTROKE TO PROCESS
Confirm Delete key cannot be the same as the Keystroke to Process.

Appendix E

TABLES/AS System Menu



```
---------------------  TABLES SYSTEM PRIMARY OPTION MENU   ---------------------
SELECT OPTION ===>                                                          DATE:
                                                                            TIME:

  1  TABLES/AP - Select TABLES Automatic Default Processing
  2  TABLES/AS - Select TABLES Application System




   APPLICATION ID ===>            (May be Required)








                                TABLES SYSTEM RN.N
                          (C) COPYRIGHT YYYY - SSI, Inc.


PF 1=HELP    2=     3=END     4=RETURN    5=      6=
PF 7=      8=     9=      10=      11=      12=CANCEL
```

Figure E-1

Transaction TSMM displays the above menu.

Transaction TS01 displays the **TABLES** Application System menu (same menu as having selected Option 2 from the above menu).

Transaction TSAP displays the **TABLES** Automatic Default Processing panel (same panel as having selected Option 1 from the above menu).

Appendix F

VSAM Data Set Access Via TABLES/AS

JCL401MC
Extract copy code for the VSAM table. See **TABLE/S/AS** Reference Manual, Copy Code Requirements for copy code limitations.

•**MEMBER NAME** in copy code DD may be equal to the MEMBER NAME specified in the Control DD

•**TABLE NAME** specified in Filekey Record **MUST** be equal to the MEMBER NAME specified in the Control DD

•**DD NAME** specified in the Filekey Record **MUST** be the DD NAME specified in the CICS FCT (File Control Table)

•**CARDS 2 - 7** are continuation cards beginning in column 1 and continuing through column 80 in each card.

•**DD STATEMENT** in the JCL must be the same as the DD Statement in the CICS start-up JCL. The DD NAME in the Filekey Record **MUST** be equal to the DD NAME in this statement.

TABLES/AP - ACCESSING A VSAM FILE
•A USERID CONTROL TABLE record must exist

- USER ID = Your sign-on Userid or "ALLOW"
- TABLE NAME = DD NAME as specified in the CICS FCT
- AUTHORIZATION = "Y" or "N" for each function desired.

Usually a "Y" is entered for all. If using "ALLOW", normally a "Y" is entered only for the Inquire and Browse functions to allow all users inquiry ability.

- Use **TABLES/AP** to update the AUTHID.USERID TABLE

• On the **TABLES/AP** menu

- TABLE NAME =TABLE NAME as specified in the JCL401MC Filekey Record.
- TABLE TYPE = VSAM

TABLES/AS DEFAULT SCREEN GENERATION
• On the SCREEN DESCRIPTION panel

- TABLE NAME =TABLE NAME as specified in the JCL401MC Filekey Record.
- SOURCE =VSAM
- DEFAULT SCREEN= "YES" or "SEL"

TABLES/AS CUSTOM SCREEN MAPPING

- Enter "X" for Copy Code Source
- Enter DD NAME as defined in the CICS FCT
- Enter COPY CODE MEMBER NAME = TABLE NAME as specified in JCL401MC Filekey Record.

TABLES/AS CUSTOM SCREEN PROCESSING OPTIONS - DATABASE ACCESS

The first time the file/table is used, the values should be checked

• RECORD LENGTH	=Maximum length for this table
• KEY FIELD START	=Location in the record where the key field begins (i.e., the 1st position = 0001)
• KEY FIELD LENGTH	=Length of the key field

TABLES/AS SCREENS ACCESSING A VSAM DATA SET

From Option 9 **- TABLES/AS** Primary Option menu

- Enter Option 9, Screen name and press ENTER
- **TABLES/AS** Screen Processor menu will be displayed with your screen name in it. Press ENTER to process your screen.

From Transaction TS09

- A USERID Control Table Record must exist for your sign-on ID or "ALLOW". See **TABLES/AP** above for description.
- You **MUST** have run Option 5 **-** Prepare Screen for this screen **OR** Option 9 Screen Processor with this screen name.
- You may have created a SCREEN LOAD MODULE first.

> **NOTE**
>
> If a screen load module exists, TS09 will use it. Therefore, once a screen is in load module form, if any changes were made to the screen, always recreate the load module before using TS09 **OR** delete the load module from the library.

VSAM DATASET ACCESS VIA TABLES

Please note that when accessing the **TABLES/AS** Screen Processor menu from Option 9, there is no verification of userid. Option 9 assumes development mode and therefore does not require a userid.

the USERID control table in **TABLES** must be updated in each separate region where it will be used. It is not migrated!

USERID - The USERID is obtained from the CICS sign-on ID.

The CICS sign-on USERID, if entered, **MUST** be a valid entry in the USERID control table. A valid entry consists of either:

a) that userid connected to that dataset

or

b)a userid of "ALLOW" for that dataset.

If you use CICS sign-on, you will not be allowed to use a userid different than your sign-on ID.

If using your own CICS TRAN code, with an entry in the TXNMNU table, the above USERID requirements exist. Depending on whether you use CICS sign-on or not the entries in the USERID control table, the following will occur:

·If the user is signed on to CICS and the USERID is valid, the screen is displayed.

·If the USERID table has the value "Allow", the screen is displayed.

·If the user is signed on to CICS and the sign-on userid is not valid, the **TABLES/AS** Screen Processor menu will be displayed. The sign-on userid cannot be over-ridden on the menu. It must have a valid entry in the USERID table.

Appendix G

Connected Tables

Within the **TABLES/AS** processing capabilities, there is limited support and specific requirements for using connected tables.

The following screen formats should work:

```
A.   1 row    --->         1 row connection
B.   1 row    --->         M row connection
C.   1 row    --->         1 row    ---> M row connection

                            |-----> M row connection
                            |
                            |    The search values for the 1 multi-row tables
                            |     must come from the single row connection
D.   1 row    --->          |    that started the connected tables.
                            |    The 1 ---> M connections are basically
                            |     independent of each other.
                            |
                            |-----> M row connection
```

FOR EXAMPLE D.:
Although the 1 ---> M sets are like independent tables, they may be accessed as connected as long as the search values are all part of the first table OR existed on the screen (by being passed from another screen or populated with initial values).

FOR EXAMPLE A., B. and C.:
The first table that has 'MANY', rows on the screen must be the last of the connected tables on that screen. To proceed further you need to switch to another screen.

FOR EXAMPLE D.:
Although each 1 ---> M sets are like independent tables, they may be accessed as connected as long as the search values are all a part of the first table.

```
1 row  ---> M row                  Screen1
            |
            |
            V
            1 row  ---> M row                  Screen 2
```

When specifying the number of rows to retrieve in the Process Options, Allow Functions screen, you must specify the same number as can be held on the screen., i.e., one in all cases except for the last table if the screen has multiple rows. For that value, it must be the same as the number of rows on the screen for that table.

For Inquiry Only screens, using a DB2 View, you could build a view that allows multiple multi-row tables. Using the DB2 View, you could possibly handle the data on a screen that could not have been accomplished without the DB2 View.

Any deviations from these constraints and requirements can produce unexpected results.

Appendix H

Test TABLES/AS Default and Custom Screens

TEST ALL DEFINED FUNCTIONS

NOTE

This process performs all edits and physically accesses and maintains the table. ALL defined features of **TABLES/AS** are available in this process.

At the **TABLES/AS** Primary Option menu:
Select Option 9 - Screen Processor
Enter the 1st screen to test
Press ENTER

The 1st screen to test will be displayed
Add some rows (function = A)
Retrieve some rows (function = R)
Retrieve some rows for Update (function = U or nothing)
Change some retrieved rows (function = C or nothing)
If data was changed in the row, it will be updated
Delete some retrieved rows (function = D)

Things to Remember
•You MAY NOT change or delete rows unless previously retrieved for update
•The top left function code field changes the screen status
•Any values entered in the 1st screen row will be used for searching
•Any blank columns in the 1st screen row will not be used for searching
•PF12 when done testing

TEST FIELD & RELATIONAL EDITS ONLY

NOTE

This process only tests the screen edits defined. It does not access the physical table.

At the **TABLES/AS** Primary Option menu
Select Option 6 - Test Facility
Enter data on the screen and press ENTER
Any edit errors will be reported in the message field
Correct errors as reported.
When the entire screen is correct, it will redisplay without data.

Appendix I

Default Screen Generation

STEPS TO FOLLOW

1. Select Screen Design

2. Enter A for Add and enter a screen name

3. On the Screen Description panel
   •Enter a description for the screen (shows up in lists)
   •Enter a table name
   •Enter a source (VSAM or IMS). DB2 is the default.
   •Enter YES or SEL for default screen

4. Once the default screen has been generated, you may:
   Select Option 2 **-** Format to change the default screen format.
   •PF2 to use Line Commands to modify the screen
   •PF5 to see table columns and the **TABLES/AS** control information
   •PF3 to save the changes

5. Proceed directly to TEST the screen
   See 'Testing **TABLES/AS** Default & Custom Screens' in Appendix H

6. If screen changes are required, go to Step 1 and select the Change function.

7. Add additional functionality as desired.

Custom Screen Generation

STEPS TO FOLLOW

1. Design a screen
   - Select Screen Design
     · Enter A for add and enter a screen name

     On the Screen Design - Description panel
     •Enter a description for the screen (shows up in lists)
     •Enter NO or spaces for 'Default Screen ==>'

     Paint on the blank screen all desired fields
     •Function code field(s)
     •Message field(s)
     •Table I/O field(s)
     •Literals
     Enter PF2 to edit the screen in Line Commands

PF3 to return to paint
PF3 to save and return to the Change Screen menu
Select Option 7 to Name fields on the screen
•Name the screen fields
•Use the repeat names function for repeating groups of field names
Return to the Primary menu

NOTE

•Steps 2 through 4 are required for any screen that will process a table.
•For menu or help screens that do not process tables, perform Step 3 to mark the common message field and specify edits if desired.
•Then go to Step 5.

2. Map table columns to screen fields
Select Option 3 - Record Mapping
Add a map for each table to be processed on the screen
Return to the Primary menu

3. Define field edits and identify the common message field
Select Option 2 - Edit Definition
•Select Option 1 - Field Editing
·Mark the common message field(s)
·Define new edits or change default edits, if needed
·Specify ranges or discrete values, if needed
·Change the custom messages, if needed
Return to the Primary menu

4. Define the Process options for each table
Select Option 4 - Process Options
•Select Option 1 - Allow functions
·Define Allow functions for each table to be maintained
·Define Cross Table Validation requirements, if needed
Return to the Primary menu

5. Define Switching, if required
Select Option 7 - Utilities
•Select Option 3 - Switching
·Add switching conditions
Return to the Primary menu

6. Prepare the screen
Select Option 5 - Prepare Screen and enter screen name

7. Test the screen
Select Option 9 - Screen Processing with the screen name of the first screen to execute
•Test your screen - see write-up Appendix H

8. Add additional functionality as desired.

Appendix J

Painting Custom Screens

There are two methods that may be used to paint a custom **TABLES/AS** screen:

When a table name has been specified
and
When a table name has NOT been specified

By entering NO or blanks to the question 'Default Screen ==>' on the Screen Description panel, a blank screen will be presented for painting.

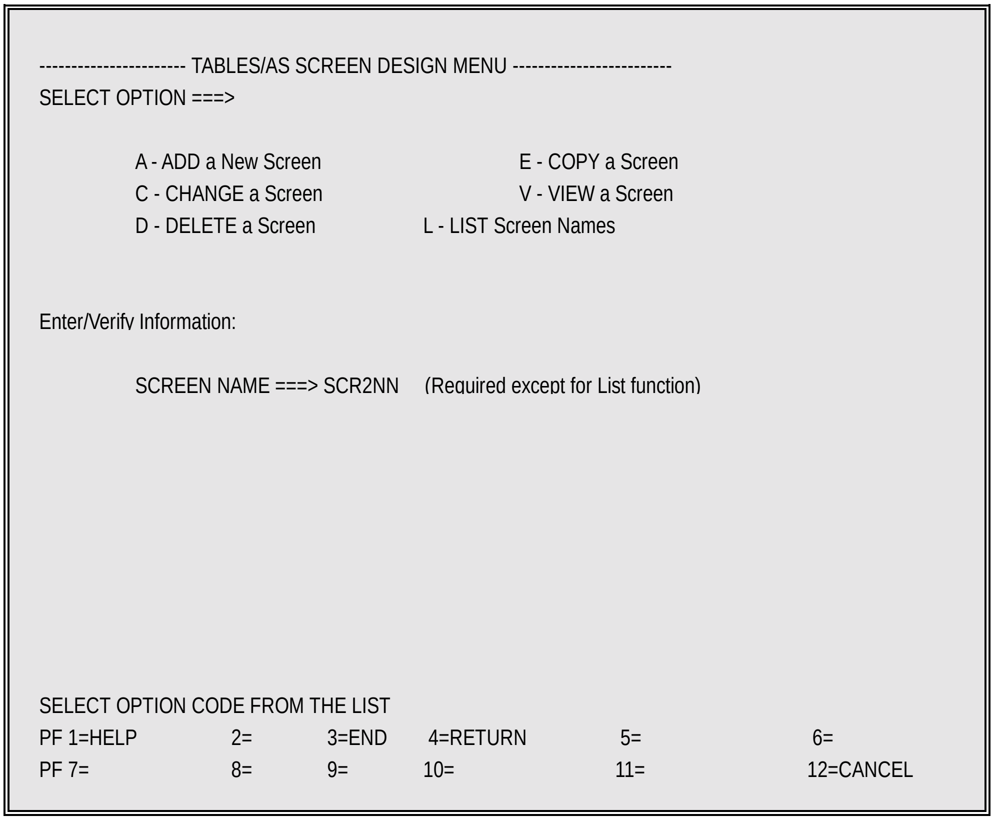
We will develop an example for each method. Each example will walk through a series of screens to show how the screen format is designed. Once the screen has been designed, all of the other facilities of **TABLES/AS** may be defined for the screen. These examples are only showing how to create the actual screen format, i.e., Add a Screen. There is additional information regarding screen design in the section Option ===> C **-** Change a Screen.

ADD A SCREEN - TABLE NAME SPECIFIED

With this method, **TABLES/AS** has information available regarding the table you named on the screen description panel. To see the information known to **TABLES/AS**, press PF5. Using this information, you can lay out the screen format you desire. The following screens are presented to show the flow and method.

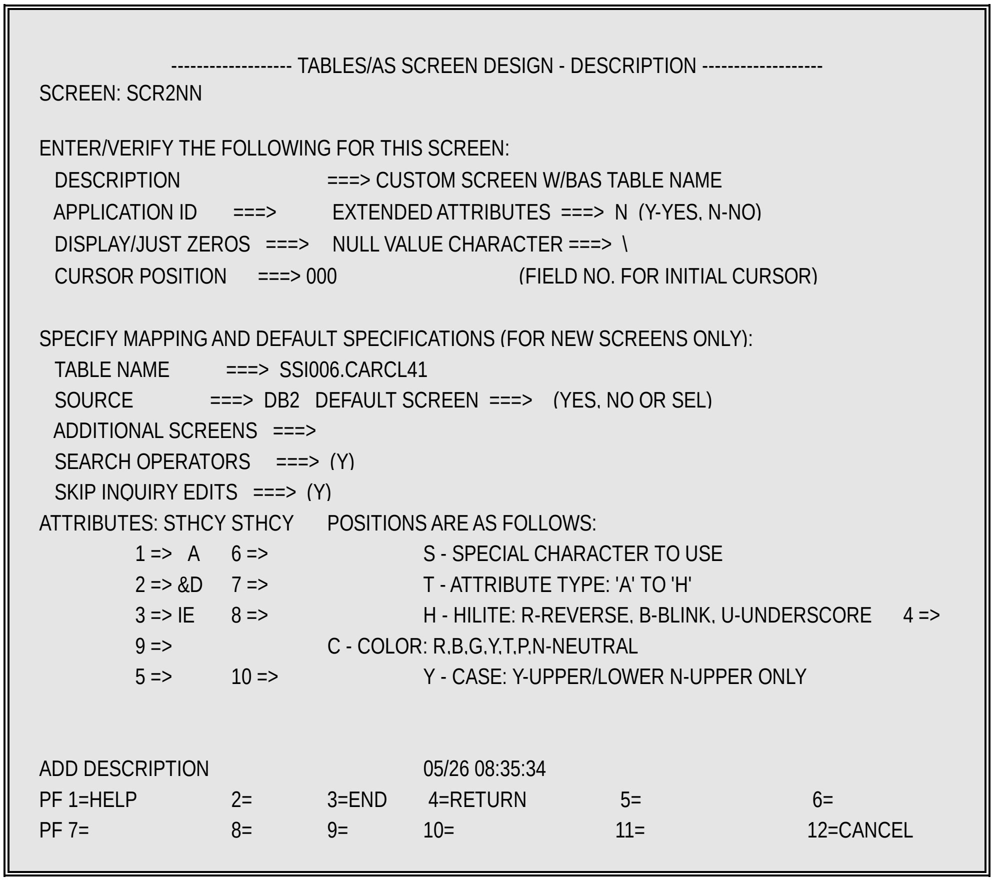
After selecting Screen Design, enter 'A' for Add and the name of the new screen and then press ENTER.

We are going to create a screen named SCR2NN



```
---------------------- TABLES/AS SCREEN DESIGN MENU ------------------------
SELECT OPTION ===>

          A - ADD a New Screen                    E - COPY a Screen
          C - CHANGE a Screen                     V - VIEW a Screen
          D - DELETE a Screen          L - LIST Screen Names


Enter/Verify Information:

          SCREEN NAME ===> SCR2NN    (Required except for List function)









SELECT OPTION CODE FROM THE LIST
PF 1=HELP           2=        3=END     4=RETURN          5=               6=
PF 7=               8=        9=        10=               11=              12=CANCEL
```

The Screen Description panel is displayed next. Note that we have specified a table name and have not entered anything for the question 'Default
Screen ==>'. These two entries are key to identifying that you have a table to use and that you want to create a custom screen. We could have answered NO to the default screen question to achieve the same results.



```
                  ------------------ TABLES/AS SCREEN DESIGN - DESCRIPTION ------------------
SCREEN: SCR2NN

ENTER/VERIFY THE FOLLOWING FOR THIS SCREEN:
  DESCRIPTION                 ===> CUSTOM SCREEN W/BAS TABLE NAME
  APPLICATION ID     ===>        EXTENDED ATTRIBUTES  ===>  N  (Y-YES, N-NO)
  DISPLAY/JUST ZEROS   ===>    NULL VALUE CHARACTER ===>  \
  CURSOR POSITION      ===> 000                          (FIELD NO. FOR INITIAL CURSOR)

SPECIFY MAPPING AND DEFAULT SPECIFICATIONS (FOR NEW SCREENS ONLY):
  TABLE NAME          ===>  SSI006.CARCL41
  SOURCE             ===>  DB2   DEFAULT SCREEN  ===>    (YES, NO OR SEL)
  ADDITIONAL SCREENS   ===>
  SEARCH OPERATORS     ===>  (Y)
  SKIP INQUIRY EDITS   ===>  (Y)
ATTRIBUTES: STHCY STHCY      POSITIONS ARE AS FOLLOWS:
             1 =>   A     6 =>              S - SPECIAL CHARACTER TO USE
             2 => &D      7 =>              T - ATTRIBUTE TYPE: 'A' TO 'H'
             3 => IE      8 =>              H - HILITE: R-REVERSE, B-BLINK, U-UNDERSCORE      4 =>
             9 =>                   C - COLOR: R,B,G,Y,T,P,N-NEUTRAL
             5 =>         10 =>             Y - CASE: Y-UPPER/LOWER N-UPPER ONLY


ADD DESCRIPTION                          05/26 08:35:34
PF 1=HELP          2=        3=END      4=RETURN           5=                 6=
PF 7=              8=        9=         10=                11=                12=CANCEL
```

When we press ENTER here, a completely blank screen will be displayed for painting.

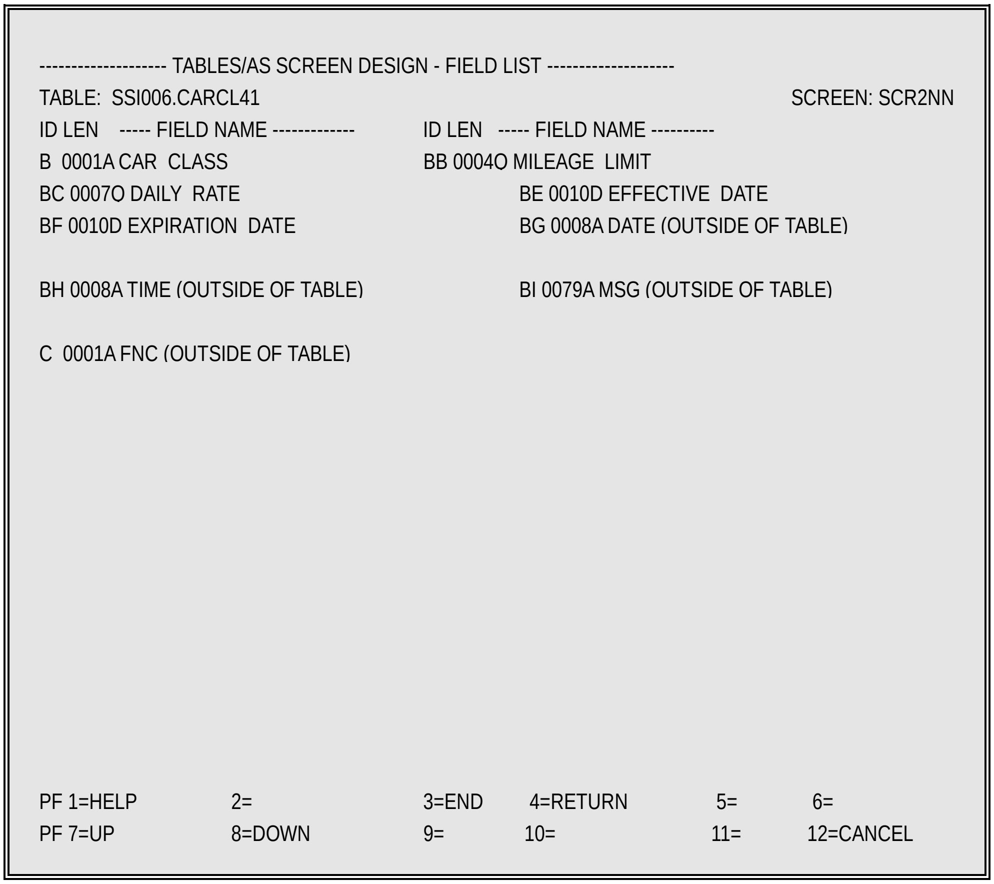
The next 3 screens show the development of a custom screen once the blank screen has been displayed for painting.

There are some instructions and/or comments in the screens talking about the process of development.

Position 1,1 of the screen is reserved for a Field Start Indicator. If one is not here TABLES/AS will insert one.

We will design this screen using the information available because we used a base table name in the screen description panel. When the ENTER key was pressed, a blank screen was presented. All of the information thus far has simply been typed in. It will be treated as a literal. From here on we will use literals for field names and specify the particular I/O field. We press PF5 to see the list of fields known to TABLES/AS.

When we press PF5 here, a Field List panel will be displayed.



```
-------------------- TABLES/AS SCREEN DESIGN - FIELD LIST --------------------
TABLE:  SSI006.CARCL41                                           SCREEN: SCR2NN
ID LEN    ----- FIELD NAME -------------         ID LEN   ----- FIELD NAME ----------
B  0001A CAR  CLASS                              BB 0004O MILEAGE  LIMIT
BC 0007O DAILY  RATE                                     BE 0010D EFFECTIVE  DATE
BF 0010D EXPIRATION  DATE                                BG 0008A DATE (OUTSIDE OF TABLE)

BH 0008A TIME (OUTSIDE OF TABLE)                         BI 0079A MSG (OUTSIDE OF TABLE)

C  0001A FNC (OUTSIDE OF TABLE)















PF 1=HELP        2=               3=END      4=RETURN        5=        6=
PF 7=UP          8=DOWN           9=         10=             11=       12=CANCEL
```

When **TABLES/AS** is asked to automatically generate a screen, in addition to the columns in the table, four additional fields are placed in the screen. These fields are: DATE, TIME, FNC, MSG.

The ID shown is a **TABLES/AS** generated control value. For single character columns, a single character control field is generated. For columns with 2 or more characters, a 2 character control field is used.

The LEN shown is the total length of the column for the screen display.

The character after the LEN is the **TABLES/AS** screen format character. Valid **TABLES/AS** screen format characters are A, D, Q and N. See the Change a Screen section for a complete explanation of these codes.

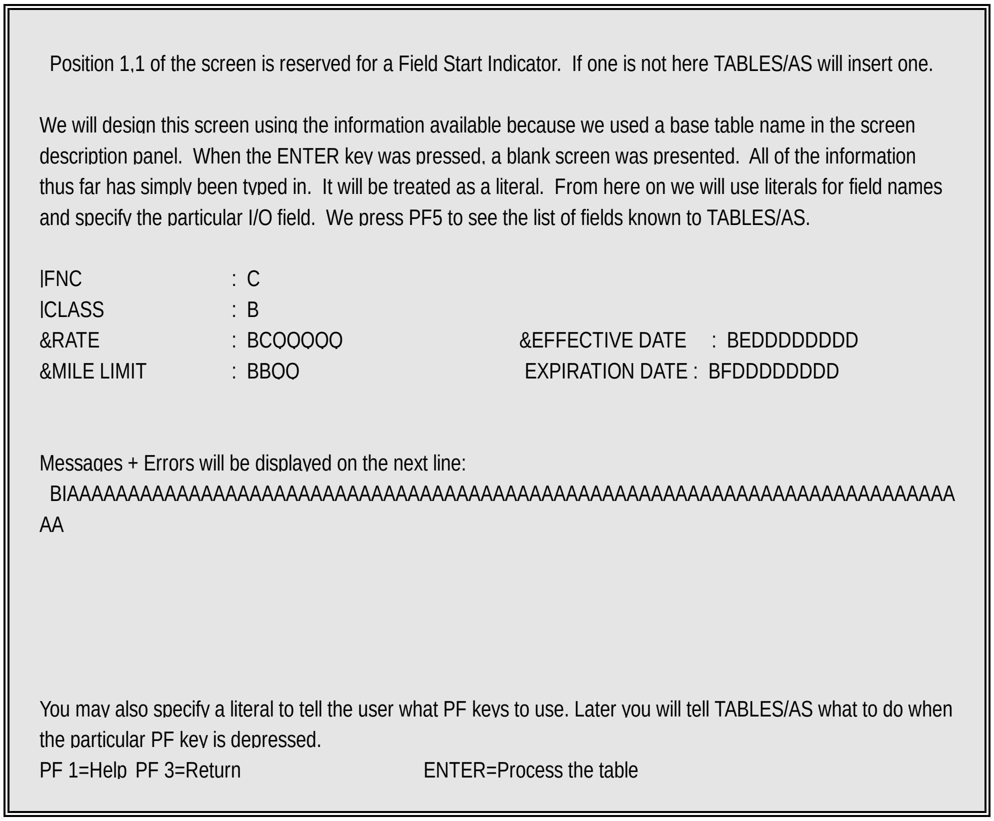
The FIELD NAME is the name of the column from the table or one of the four **TABLES/AS** generated fields mentioned above.

To position one of these fields onto the screen being painted, we use the information shown on this screen about the field. We toggle back and forth between this screen and the screen being painted as we use the information to complete the screen design.

To paint one of these fields on our screen:
- •Enter a literal as a name for the screen field.
- •Enter the Field Start Indicator for an I/O field. **TABLES/AS** is shipped with '_' (underscore) as this indicator.
- •Enter the 1 or 2 character ID assigned for to the column.
- •For 3 character fields or longer, enter (LEN minus 2) screen format characters.

You toggle back and forth between this screen and the screen being designed to obtain the correct formats and to paint them on the screen.

The next screen shows a number of fields painted on the SCR2NN screen.

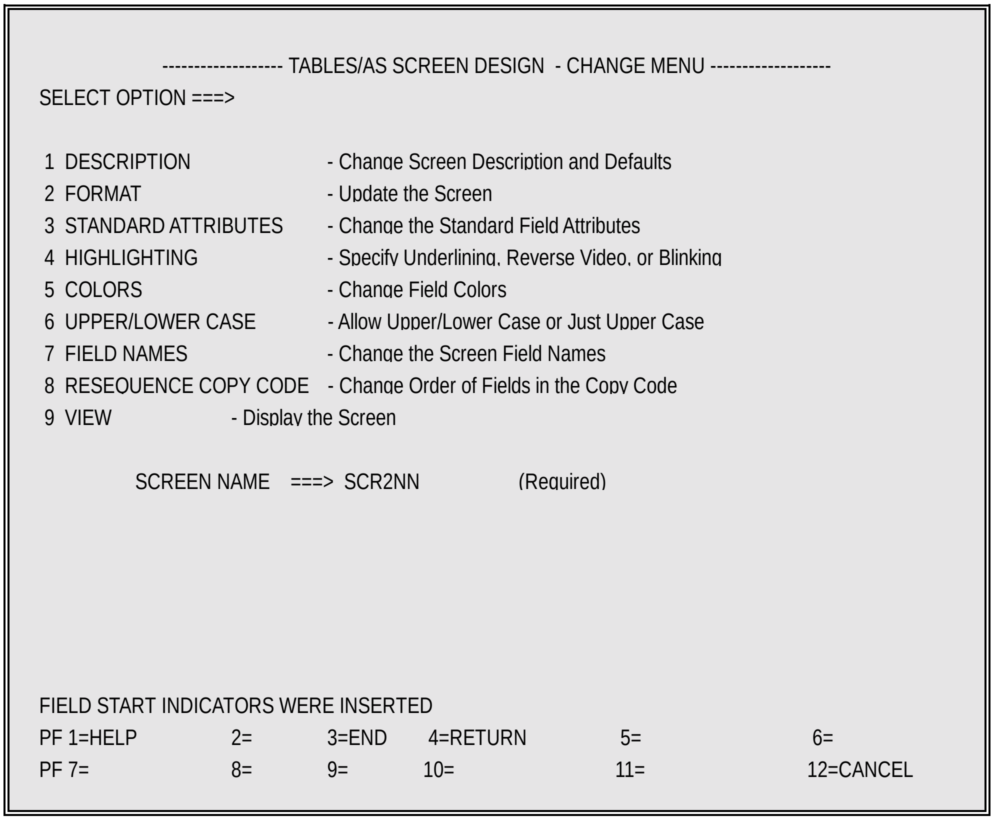


```
  Position 1,1 of the screen is reserved for a Field Start Indicator.  If one is not here TABLES/AS will insert one.

We will design this screen using the information available because we used a base table name in the screen
description panel.  When the ENTER key was pressed, a blank screen was presented.  All of the information
thus far has simply been typed in.  It will be treated as a literal.  From here on we will use literals for field names
and specify the particular I/O field.  We press PF5 to see the list of fields known to TABLES/AS.

|FNC                 :  C
|CLASS               :  B
&RATE                :  BCQQQQQ                  &EFFECTIVE DATE    :  BEDDDDDDDDD
&MILE LIMIT          :  BBQQ                      EXPIRATION DATE :  BFDDDDDDDDD


Messages + Errors will be displayed on the next line:
  BIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AA




You may also specify a literal to tell the user what PF keys to use. Later you will tell TABLES/AS what to do when
the particular PF key is depressed.
PF 1=Help  PF 3=Return                      ENTER=Process the table
```

Once the screen has been fully painted, press PF3 to save it. If it has been designed correctly, you will get one of the following messages at the bottom of the Change Menu screen:

- Field Start Indicators Were Inserted or
- Screen Added

If any errors have been detected, an appropriate message will be displayed. In some cases it may be at the top of the screen you are painting. In others, it will be at the bottom of the Change Menu screen.



```
             ------------------- TABLES/AS SCREEN DESIGN  - CHANGE MENU -------------------
SELECT OPTION ===>

 1  DESCRIPTION                 - Change Screen Description and Defaults
 2  FORMAT                      - Update the Screen
 3  STANDARD ATTRIBUTES         - Change the Standard Field Attributes
 4  HIGHLIGHTING                - Specify Underlining, Reverse Video, or Blinking
 5  COLORS                      - Change Field Colors
 6  UPPER/LOWER CASE            - Allow Upper/Lower Case or Just Upper Case
 7  FIELD NAMES                 - Change the Screen Field Names
 8  RESEQUENCE COPY CODE        - Change Order of Fields in the Copy Code
 9  VIEW                - Display the Screen

          SCREEN NAME    ===>  SCR2NN            (Required)






FIELD START INDICATORS WERE INSERTED
PF 1=HELP          2=        3=END     4=RETURN          5=                 6=
PF 7=              8=        9=        10=               11=                12=CANCEL
```

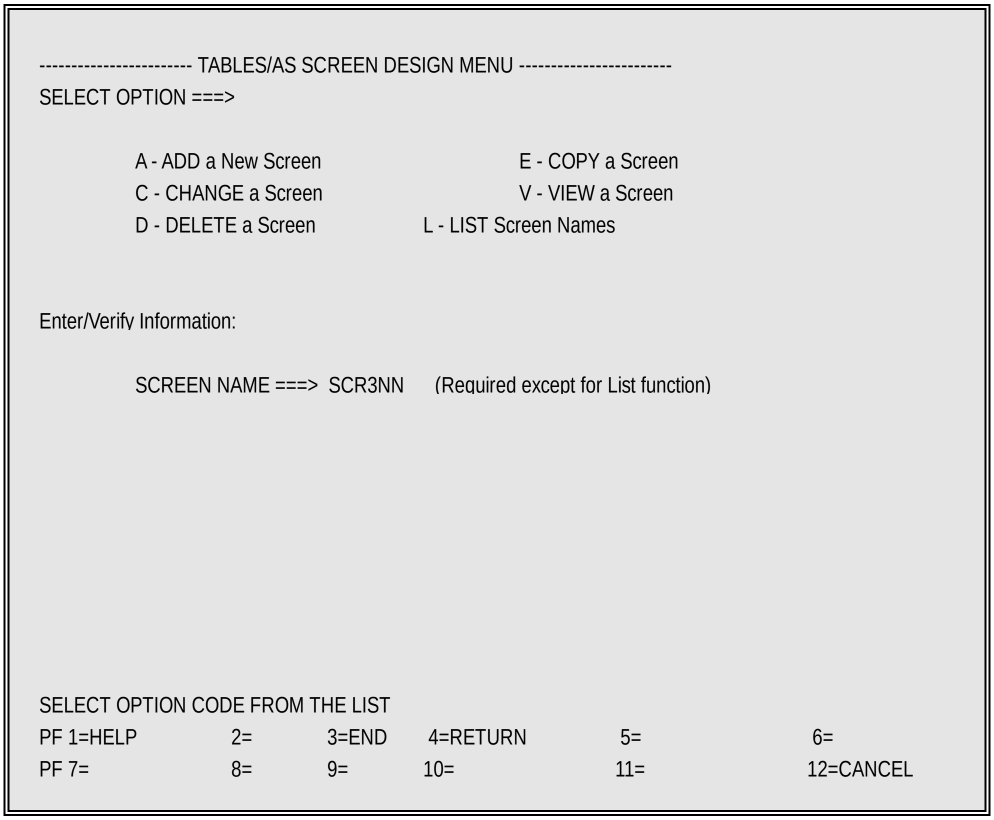
This message appears when a screen has been successfully saved.

ADD A SCREEN - TABLE NAME 'NOT' SPECIFIED

With this method, **TABLES/AS** has NO information available regarding the table(s) you will be using with this screen. To use this method, you need to know the format of the columns in the table(s) you will be using on this screen. The following screens are presented to show the flow and method.

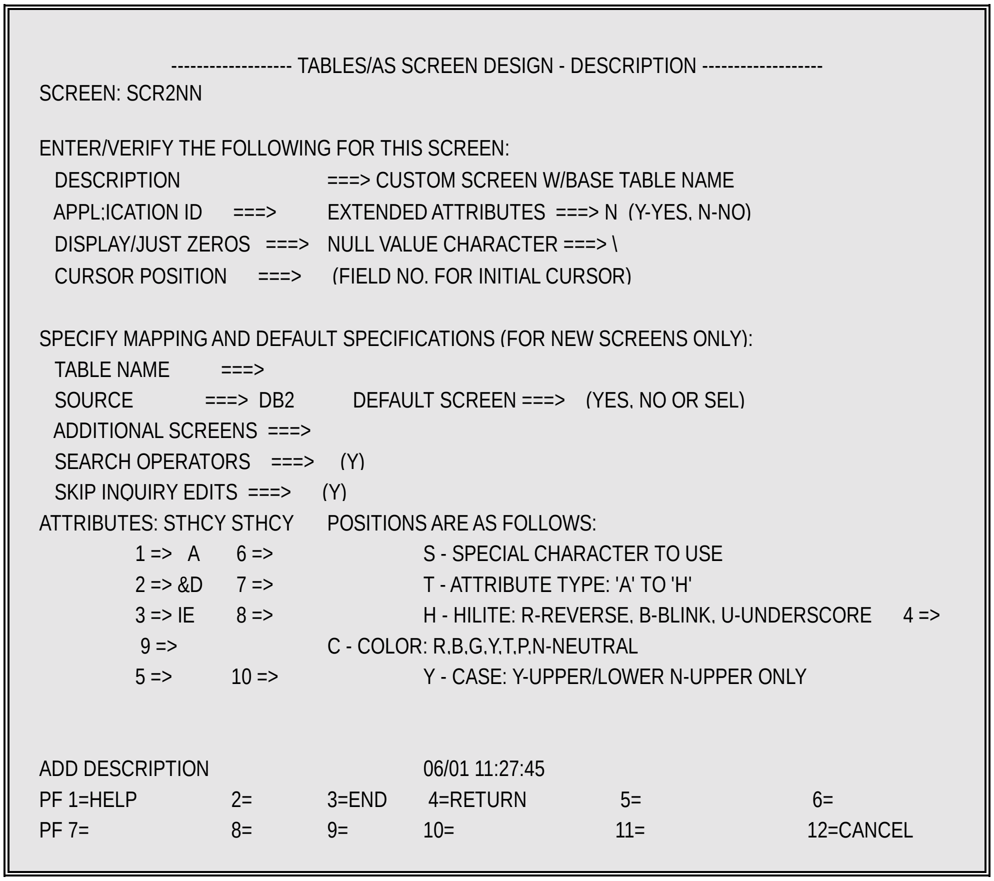
We are going to create a screen named SCR3NN. This screen will look similar to the one used when a table name was entered (see TABLE NAME SPECIFIED section in this appendix). In this way you will see the difference in effort needed. The steps needed to define edits, mapping and process options were not required when the table name was specified.

After selecting Screen Design, enter 'A' for Add and the name of the new screen and press ENTER.



```
----------------------- TABLES/AS SCREEN DESIGN MENU -----------------------
SELECT OPTION ===>

          A - ADD a New Screen                    E - COPY a Screen
          C - CHANGE a Screen                     V - VIEW a Screen
          D - DELETE a Screen           L - LIST Screen Names


Enter/Verify Information:

          SCREEN NAME ===>  SCR3NN     (Required except for List function)








SELECT OPTION CODE FROM THE LIST
PF 1=HELP           2=         3=END      4=RETURN           5=                  6=
PF 7=               8=         9=        10=                11=                 12=CANCEL
```

The Screen Description panel is displayed next. Note that we have NOT specified a table name and have NOT entered anything for the question 'Default Screen ==>'. These two entries are key to identifying that you want to create a custom screen, manually specifying all options. We could have answered NO to the default screen question to achieve the same results.



```
                  ------------------ TABLES/AS SCREEN DESIGN - DESCRIPTION ------------------
SCREEN: SCR2NN

ENTER/VERIFY THE FOLLOWING FOR THIS SCREEN:
  DESCRIPTION                   ===> CUSTOM SCREEN W/BASE TABLE NAME
  APPL;ICATION ID     ===>      EXTENDED ATTRIBUTES  ===> N  (Y-YES, N-NO)
  DISPLAY/JUST ZEROS  ===>      NULL VALUE CHARACTER ===> \
  CURSOR POSITION     ===>       (FIELD NO. FOR INITIAL CURSOR)

SPECIFY MAPPING AND DEFAULT SPECIFICATIONS (FOR NEW SCREENS ONLY):
  TABLE NAME       ===>
  SOURCE           ===>  DB2        DEFAULT SCREEN ===>    (YES, NO OR SEL)
  ADDITIONAL SCREENS  ===>
  SEARCH OPERATORS    ===>     (Y)
  SKIP INQUIRY EDITS  ===>     (Y)
ATTRIBUTES: STHCY STHCY         POSITIONS ARE AS FOLLOWS:
             1 =>   A     6 =>                S - SPECIAL CHARACTER TO USE
             2 => &D      7 =>                T - ATTRIBUTE TYPE: 'A' TO 'H'
             3 => IE      8 =>                H - HILITE: R-REVERSE, B-BLINK, U-UNDERSCORE     4 =>
              9 =>                  C - COLOR: R,B,G,Y,T,P,N-NEUTRAL
             5 =>        10 =>                Y - CASE: Y-UPPER/LOWER N-UPPER ONLY


ADD DESCRIPTION                               06/01 11:27:45
PF 1=HELP        2=        3=END      4=RETURN          5=              6=
PF 7=            8=        9=         10=               11=             12=CANCEL
```

When we press ENTER here, a completely blank screen will be displayed for painting.

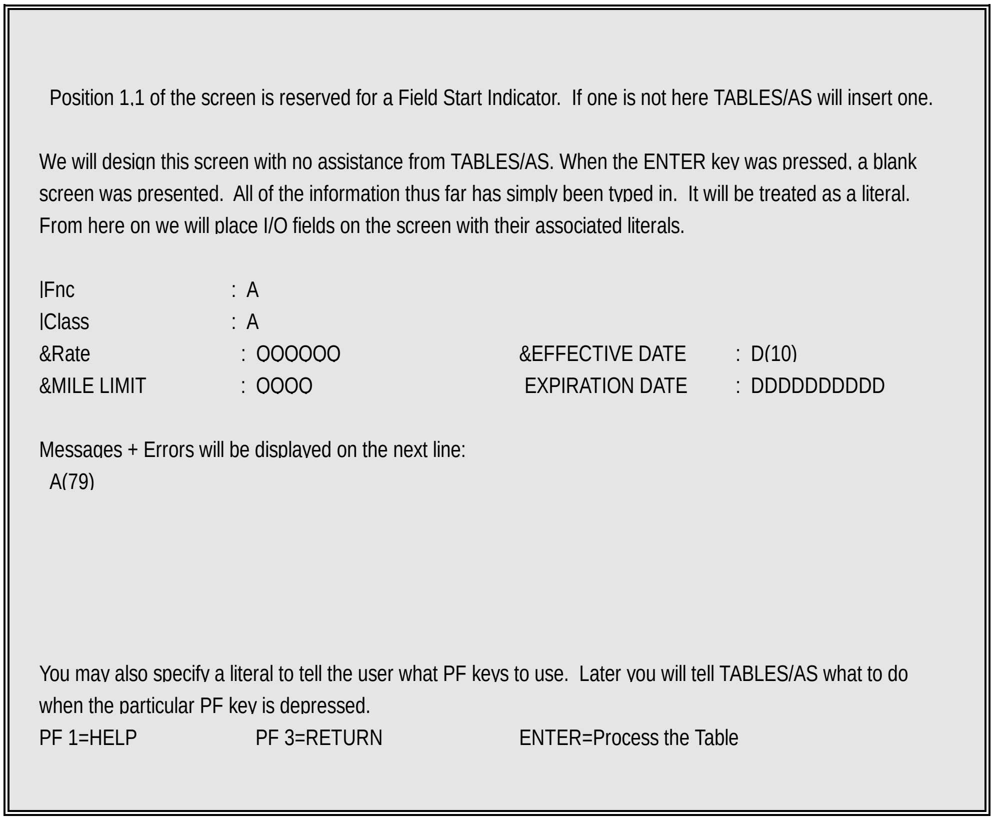
The next two screens show the development of a custom screen once the blank screen has been displayed for painting.

There are some instructions and/or comments in the screens talking about the process of development.

Position 1,1 of the screen is reserved for a Field Start Indicator. If one is not here TABLES/AS will insert one.

We will design this screen with no assistance from TABLES/AS. When the ENTER key was pressed, a blank screen was presented. All of the information thus far has simply been typed in. It will be treated as a literal. From here on we will place I/O fields on the screen with their associated literals.

When a table was specified, we were able to press PF 5 to obtain information about the columns in the table. Since we have NOT entered a table name, we need to already know about the columns in the table and begin to design them into our screen.



```
 Position 1,1 of the screen is reserved for a Field Start Indicator.  If one is not here TABLES/AS will insert one.

We will design this screen with no assistance from TABLES/AS. When the ENTER key was pressed, a blank
screen was presented.  All of the information thus far has simply been typed in.  It will be treated as a literal.
From here on we will place I/O fields on the screen with their associated literals.

|Fnc                  :  A
|Class                :  A
&Rate                  :  OOOOOO                    &EFFECTIVE DATE      :  D(10)
&MILE LIMIT            :  OOOO                       EXPIRATION DATE     :  DDDDDDDDDD

Messages + Errors will be displayed on the next line:
  A(79)




You may also specify a literal to tell the user what PF keys to use.  Later you will tell TABLES/AS what to do
when the particular PF key is depressed.
PF 1=HELP              PF 3=RETURN              ENTER=Process the Table
```

To enter literals on a custom screen is exactly the same whether a table name was entered or not. The entry of the I/O fields on the screen is a little different.

To paint a field on our screen:

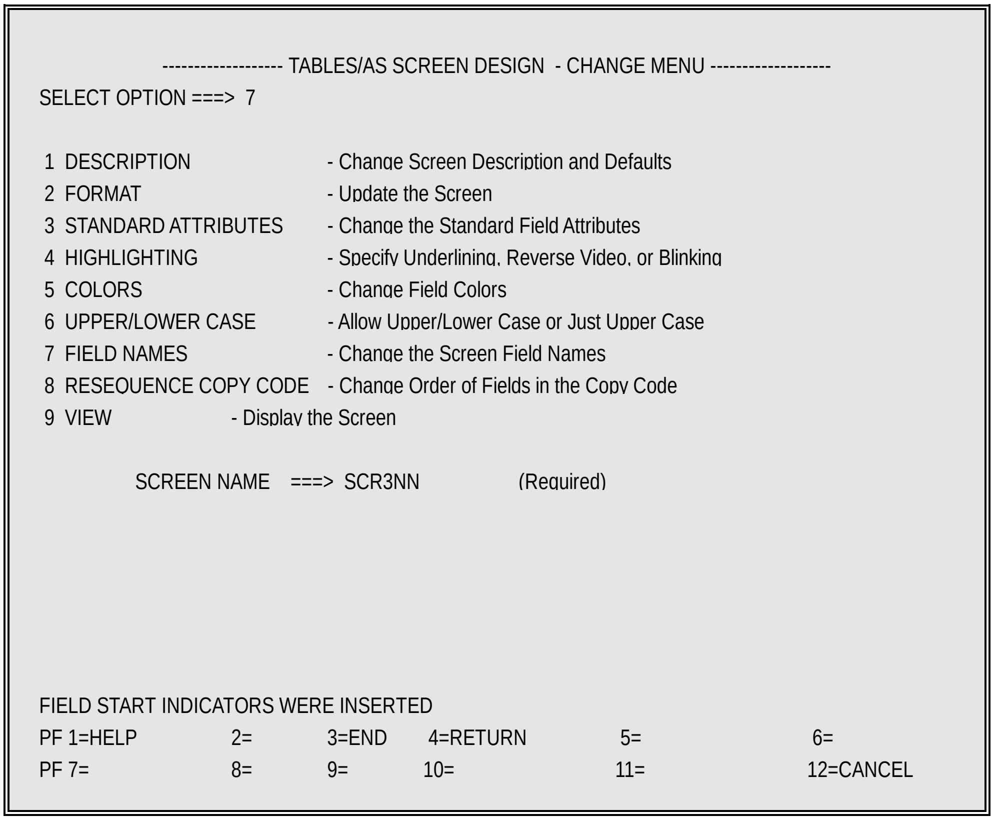
•Enter a literal as a name for the screen field.

•Enter the Field Start Indicator for an I/O field. TABLES/AS is shipped with '_' (underscore) as this indicator.

•Enter a 1 to X TABLES/AS field type characters. Enough characters of the correct type should be entered to hold the table column. Valid field type characters are A (alphanumeric), D (date), Q (quantity), and N (numeric). Looking at the two date fields in the screen above, we see there are two ways to specify a 10 character date field. This is true for each of the four field types.

Once the screen has been fully painted, press PF 3 to save it. If it has been designed correctly, you will get one of the following messages at the bottom of the Change Menu screen: *Field Start Indicators Were Inserted* or *Screen Added.*

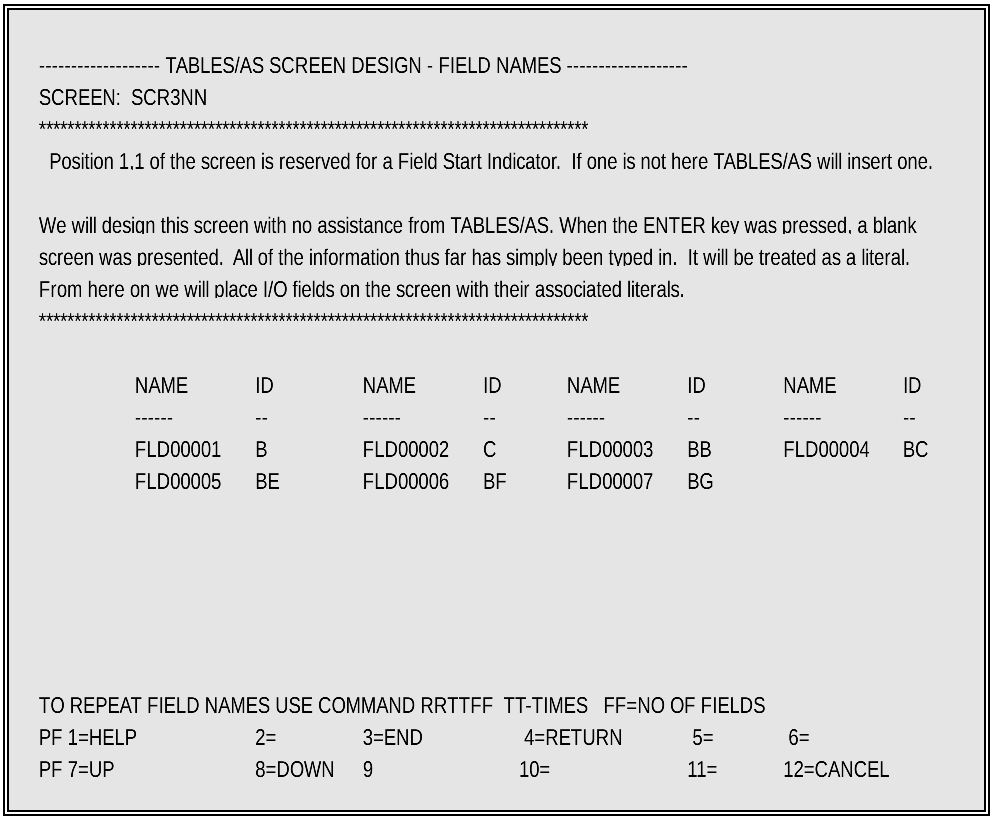
If any errors have been detected, an appropriate message will be displayed. In some cases it may be at the top of the screen you are painting. In others, it will be at the bottom of the Change Menu screen.



```
                  ------------------- TABLES/AS SCREEN DESIGN  - CHANGE MENU -------------------
SELECT OPTION ===>  7

 1  DESCRIPTION                  - Change Screen Description and Defaults
 2  FORMAT                       - Update the Screen
 3  STANDARD ATTRIBUTES          - Change the Standard Field Attributes
 4  HIGHLIGHTING                 - Specify Underlining, Reverse Video, or Blinking
 5  COLORS                       - Change Field Colors
 6  UPPER/LOWER CASE             - Allow Upper/Lower Case or Just Upper Case
 7  FIELD NAMES                  - Change the Screen Field Names
 8  RESEQUENCE COPY CODE         - Change Order of Fields in the Copy Code
 9  VIEW                 - Display the Screen

              SCREEN NAME    ===>  SCR3NN               (Required)






FIELD START INDICATORS WERE INSERTED
PF 1=HELP              2=          3=END      4=RETURN             5=                    6=
PF 7=                  8=          9=        10=                  11=                   12=CANCEL
```

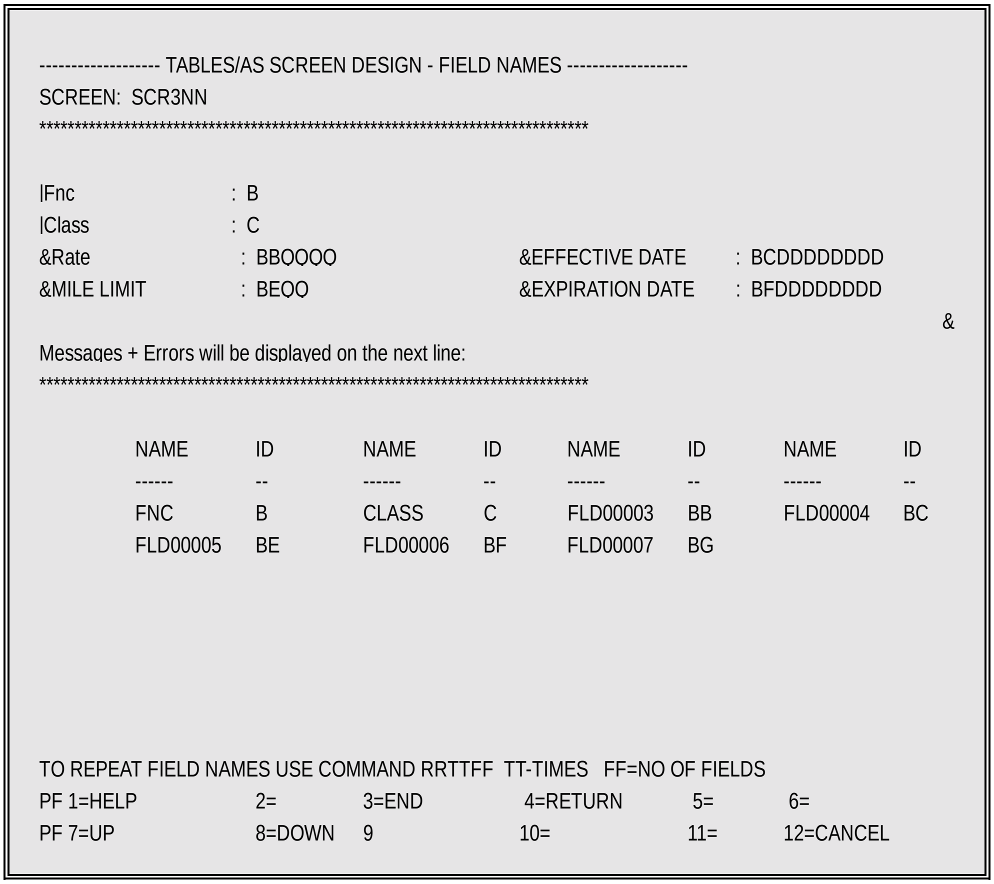
This message appears when a screen has been successfully saved.

We will enter Option 7 to assign Field Names. The next three screens will show the process of naming the screen fields.



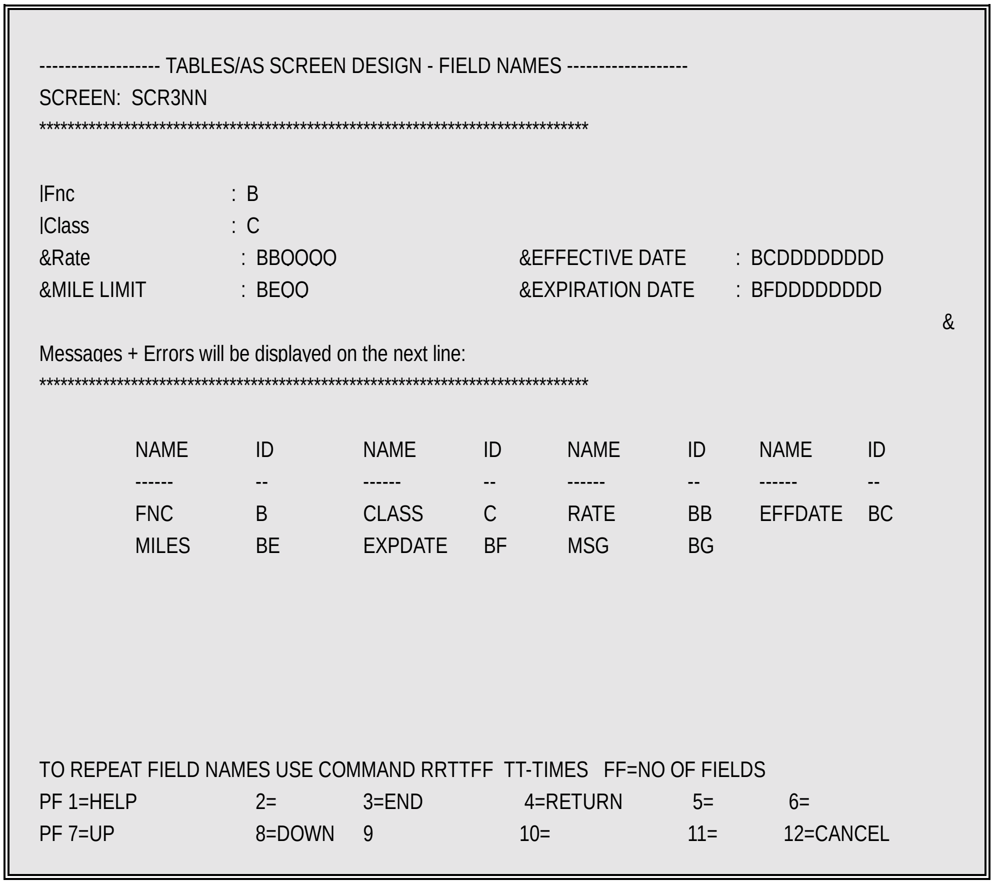
```
------------------- TABLES/AS SCREEN DESIGN - FIELD NAMES -------------------
SCREEN:  SCR3NN
*******************************************************************************
  Position 1,1 of the screen is reserved for a Field Start Indicator.  If one is not here TABLES/AS will insert one.

We will design this screen with no assistance from TABLES/AS. When the ENTER key was pressed, a blank
screen was presented.  All of the information thus far has simply been typed in.  It will be treated as a literal.
From here on we will place I/O fields on the screen with their associated literals.
*******************************************************************************

         NAME       ID       NAME       ID       NAME       ID       NAME       ID
         ------     --       ------     --       ------     --       ------     --
         FLD00001   B        FLD00002   C        FLD00003   BB       FLD00004   BC
         FLD00005   BE       FLD00006   BF       FLD00007   BG






TO REPEAT FIELD NAMES USE COMMAND RRTTFF  TT-TIMES   FF=NO OF FIELDS
PF 1=HELP          2=       3=END        4=RETURN     5=       6=
PF 7=UP            8=DOWN   9            10=          11=      12=CANCEL
```

There is an upper half (our designed screen) + a lower half (an area to enter field names. See Option 7 - Update Field Names for a detailed explanation of this process.



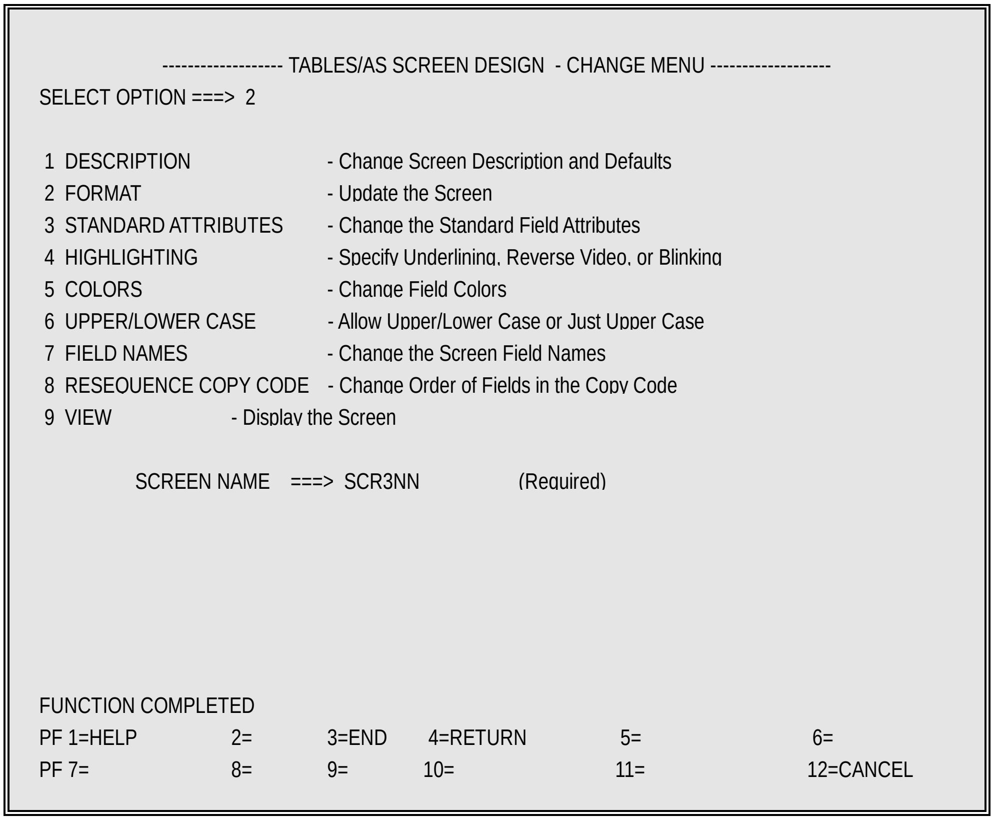
```
------------------ TABLES/AS SCREEN DESIGN - FIELD NAMES ------------------
SCREEN:  SCR3NN
*******************************************************************************

|Fnc              :  B
|Class            :  C
&Rate              :  BBQQQQ                &EFFECTIVE DATE     :  BCDDDDDDDDD
&MILE LIMIT        :  BEQQ                  &EXPIRATION DATE    :  BFDDDDDDDDD
                                                                              &
Messages + Errors will be displayed on the next line:
*******************************************************************************

          NAME       ID         NAME       ID       NAME       ID        NAME       ID
          ------     --         ------     --       ------     --        ------     --
          FNC        B          CLASS      C        FLD00003   BB        FLD00004   BC
          FLD00005   BE         FLD00006   BF       FLD00007   BG






TO REPEAT FIELD NAMES USE COMMAND RRTTFF  TT-TIMES   FF=NO OF FIELDS
PF 1=HELP            2=         3=END           4=RETURN         5=         6=
PF 7=UP              8=DOWN     9              10=              11=        12=CANCEL
```

We have named two of the fields and paged down to see our I/O fields



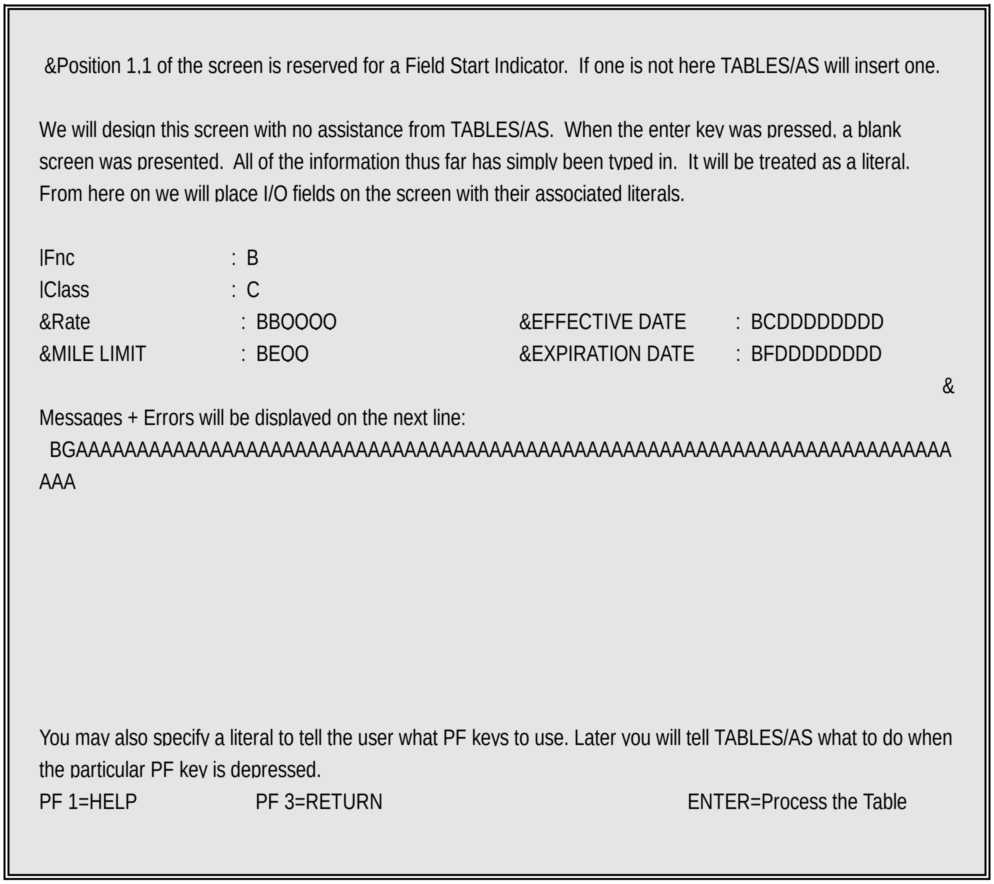
```
------------------ TABLES/AS SCREEN DESIGN - FIELD NAMES ------------------
SCREEN:  SCR3NN
*******************************************************************************

|Fnc              :  B
|Class            :  C
&Rate              :  BBOOOO                &EFFECTIVE DATE     :  BCDDDDDDDD
&MILE LIMIT        :  BEOO                  &EXPIRATION DATE    :  BFDDDDDDDD
                                                                            &
Messages + Errors will be displayed on the next line:
*******************************************************************************

         NAME       ID        NAME       ID       NAME       ID     NAME      ID
         ------     --        ------     --       ------     --     ------    --
         FNC        B         CLASS      C        RATE       BB     EFFDATE   BC
         MILES      BE        EXPDATE    BF       MSG        BG





TO REPEAT FIELD NAMES USE COMMAND RRTTFF  TT-TIMES   FF=NO OF FIELDS
PF 1=HELP            2=        3=END          4=RETURN        5=        6=
PF 7=UP              8=DOWN    9              10=             11=       12=CANCEL
```

All of the fields have been named. Press ENTER to save the names.



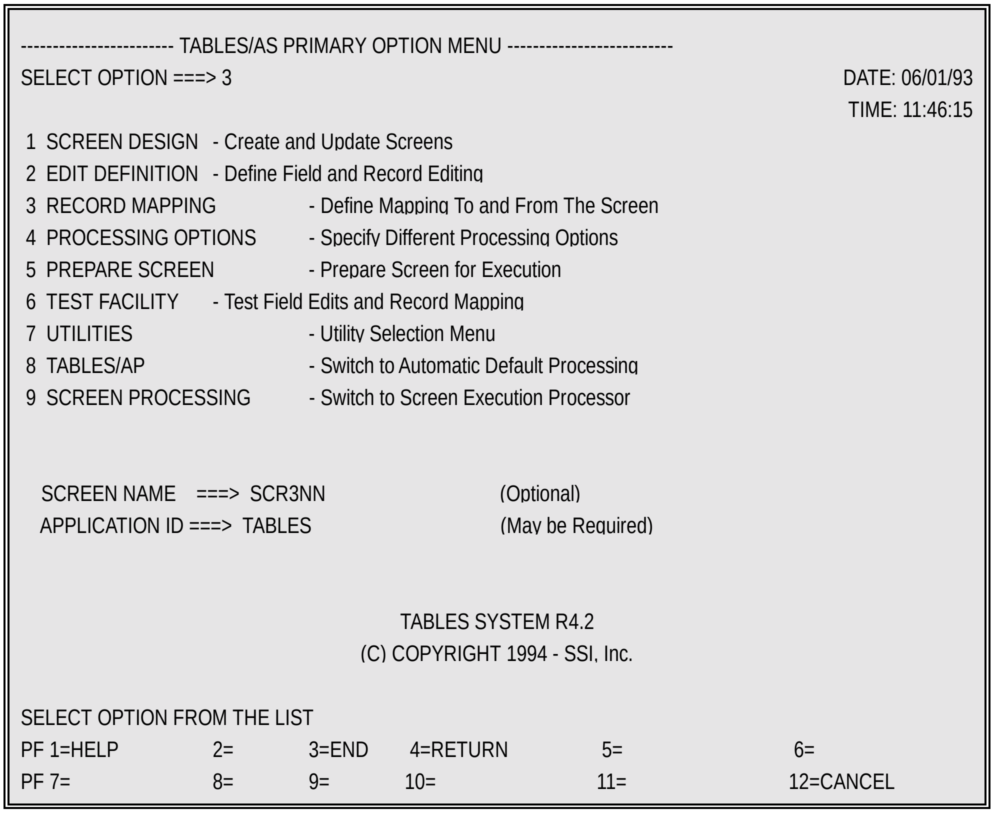
```
                ------------------ TABLES/AS SCREEN DESIGN  - CHANGE MENU ------------------
SELECT OPTION ===>  2

 1  DESCRIPTION                - Change Screen Description and Defaults
 2  FORMAT                     - Update the Screen
 3  STANDARD ATTRIBUTES        - Change the Standard Field Attributes
 4  HIGHLIGHTING               - Specify Underlining, Reverse Video, or Blinking
 5  COLORS                     - Change Field Colors
 6  UPPER/LOWER CASE           - Allow Upper/Lower Case or Just Upper Case
 7  FIELD NAMES                - Change the Screen Field Names
 8  RESEQUENCE COPY CODE       - Change Order of Fields in the Copy Code
 9  VIEW                 - Display the Screen

          SCREEN NAME    ===>  SCR3NN             (Required)






FUNCTION COMPLETED
PF 1=HELP          2=        3=END      4=RETURN         5=              6=
PF 7=              8=        9=        10=              11=             12=CANCEL
```

The FUNCTION COMPLETED message tells us we have successfully named the fields. From here we will take a look at the designed screen by entering Option 2 - Format.



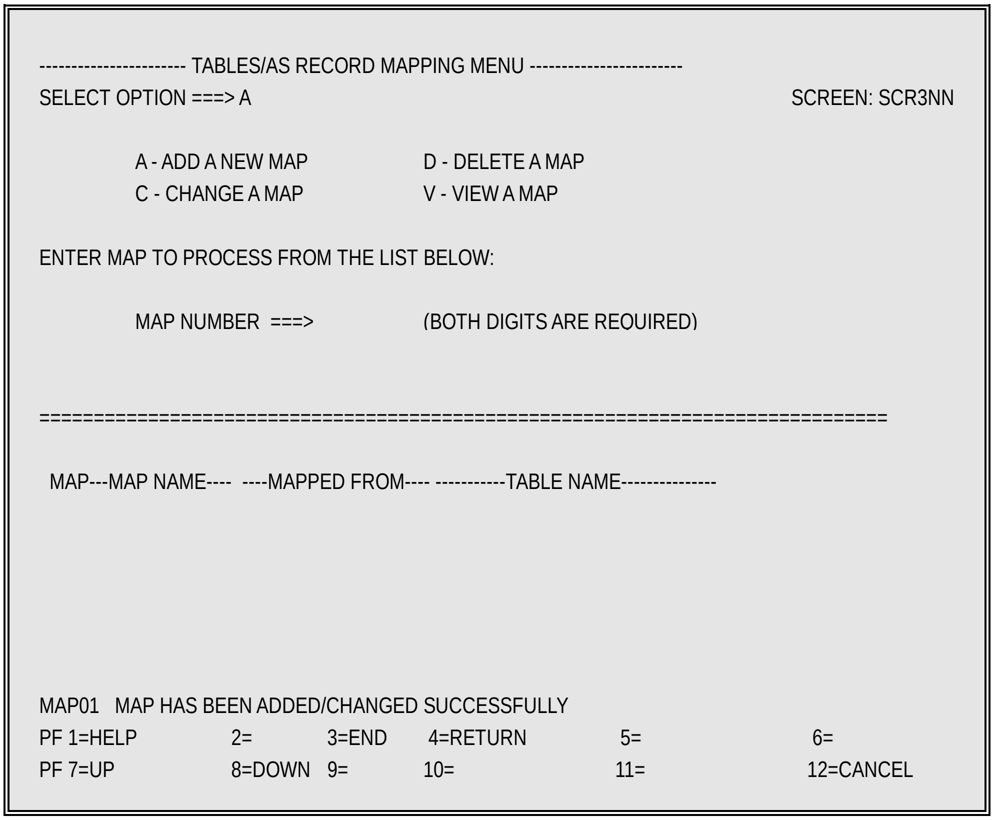
```
 &Position 1,1 of the screen is reserved for a Field Start Indicator.  If one is not here TABLES/AS will insert one.

We will design this screen with no assistance from TABLES/AS.  When the enter key was pressed, a blank
screen was presented.  All of the information thus far has simply been typed in.  It will be treated as a literal.
From here on we will place I/O fields on the screen with their associated literals.

|Fnc                  :  B
|Class                :  C
&Rate                  :  BBOOOO                  &EFFECTIVE DATE     :  BCDDDDDDDD
&MILE LIMIT            :  BEOO                    &EXPIRATION DATE    :  BFDDDDDDDD
                                                                                        &
Messages + Errors will be displayed on the next line:
  BGAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAA







You may also specify a literal to tell the user what PF keys to use. Later you will tell TABLES/AS what to do when
the particular PF key is depressed.
PF 1=HELP              PF 3=RETURN                              ENTER=Process the Table
```

Looking at the format, you can see that once the screen was saved from the painting step, the **TABLES/AS** product assigned 1 or 2 character field identifiers to each I/O field. See Change a Screen for how to move these fields around on the screen if desired.



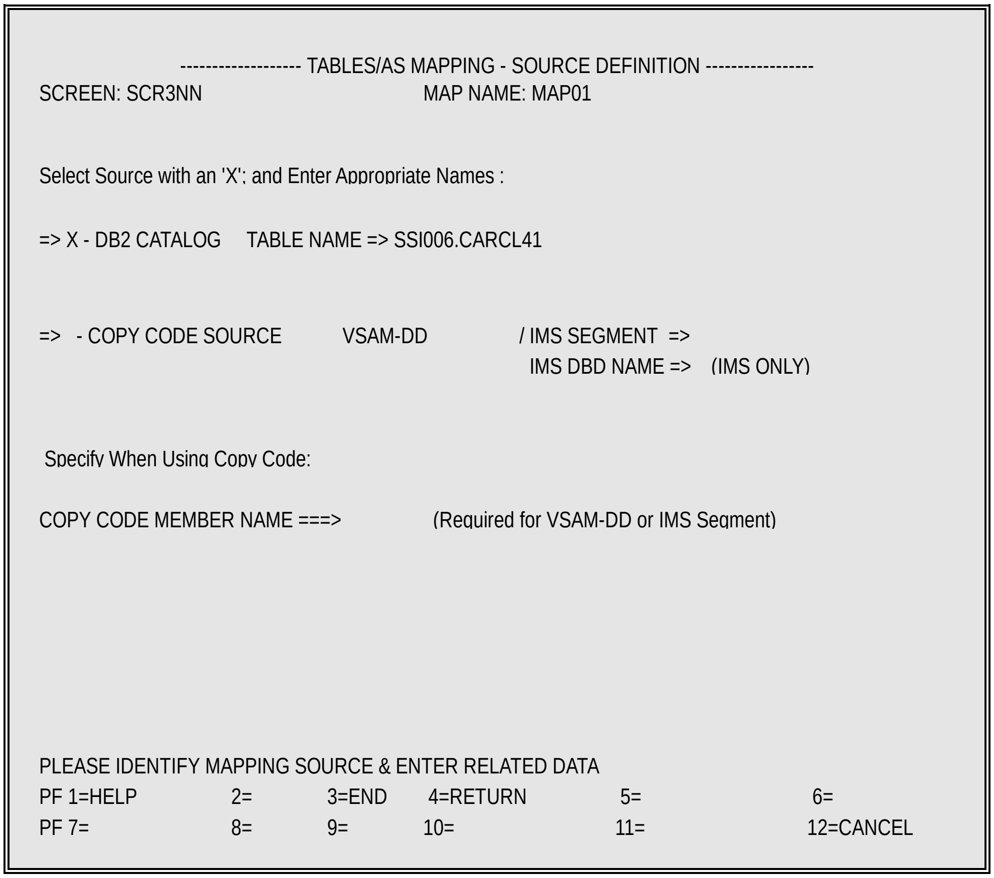
```
----------------------- TABLES/AS PRIMARY OPTION MENU -------------------------
SELECT OPTION ===> 3                                                DATE: 06/01/93
                                                                    TIME: 11:46:15
 1  SCREEN DESIGN   - Create and Update Screens
 2  EDIT DEFINITION   - Define Field and Record Editing
 3  RECORD MAPPING          - Define Mapping To and From The Screen
 4  PROCESSING OPTIONS      - Specify Different Processing Options
 5  PREPARE SCREEN          - Prepare Screen for Execution
 6  TEST FACILITY     - Test Field Edits and Record Mapping
 7  UTILITIES               - Utility Selection Menu
 8  TABLES/AP               - Switch to Automatic Default Processing
 9  SCREEN PROCESSING       - Switch to Screen Execution Processor


   SCREEN NAME    ===>  SCR3NN                   (Optional)
   APPLICATION ID ===>  TABLES                   (May be Required)



                          TABLES SYSTEM R4.2
                      (C) COPYRIGHT 1994 - SSI, Inc.

SELECT OPTION FROM THE LIST
PF 1=HELP          2=      3=END     4=RETURN        5=            6=
PF 7=              8=      9=        10=             11=           12=CANCEL
```

We will select Option 3 - Record Mapping. In the case of a screen where NO table name was specified, this is where we will connect screen fields with table fields. Up to 14 tables (maps) may be defined for use with a screen.



```
---------------------- TABLES/AS RECORD MAPPING MENU -----------------------
SELECT OPTION ===> A                                              SCREEN: SCR3NN

         A - ADD A NEW MAP              D - DELETE A MAP
         C - CHANGE A MAP               V - VIEW A MAP

ENTER MAP TO PROCESS FROM THE LIST BELOW:

         MAP NUMBER  ===>               (BOTH DIGITS ARE REQUIRED)


==============================================================================

 MAP---MAP NAME----  ----MAPPED FROM---- -----------TABLE NAME--------------






MAP01   MAP HAS BEEN ADDED/CHANGED SUCCESSFULLY
PF 1=HELP         2=        3=END      4=RETURN         5=              6=
PF 7=UP           8=DOWN    9=        10=              11=             12=CANCEL
```

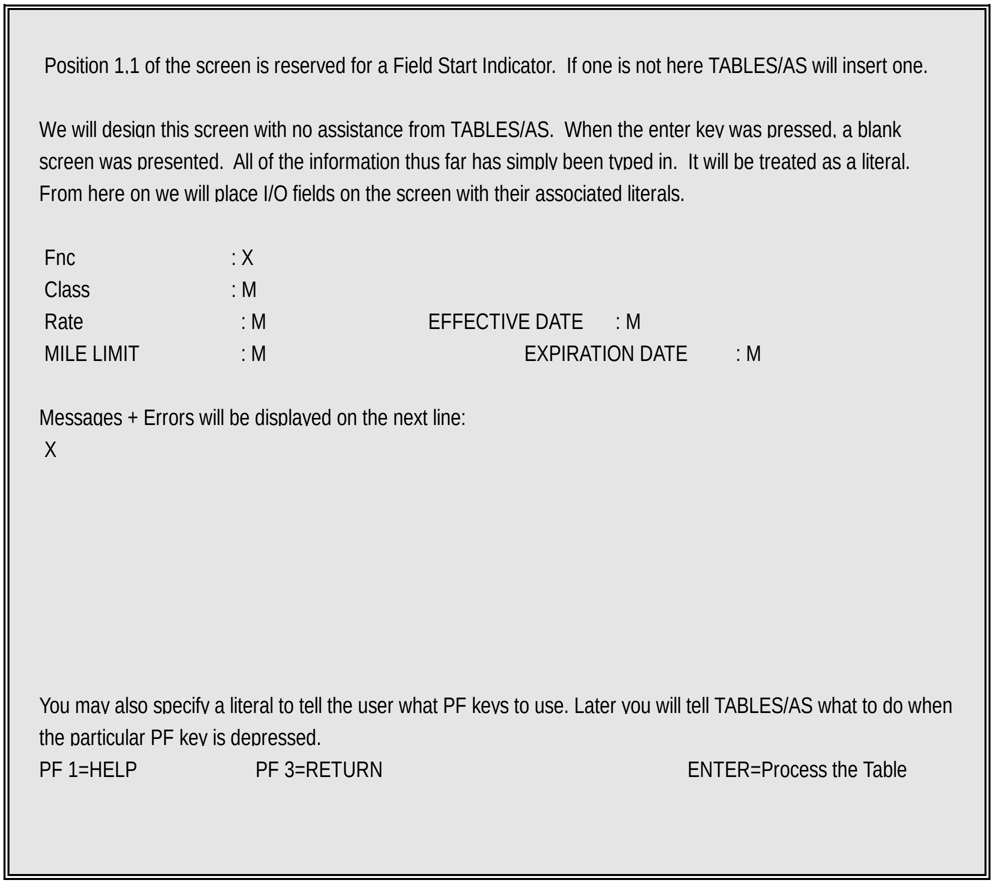
The bottom half of this screen shows the name and source for all maps already defined for this screen. We will add a map by entering Option A.



```
                   ------------------ TABLES/AS MAPPING - SOURCE DEFINITION ----------------
SCREEN: SCR3NN                              MAP NAME: MAP01


Select Source with an 'X'; and Enter Appropriate Names :

=> X - DB2 CATALOG     TABLE NAME => SSI006.CARCL41



=>   - COPY CODE SOURCE          VSAM-DD              / IMS SEGMENT  =>
                                                        IMS DBD NAME =>    (IMS ONLY)



 Specify When Using Copy Code:

COPY CODE MEMBER NAME ===>                (Required for VSAM-DD or IMS Segment)







PLEASE IDENTIFY MAPPING SOURCE & ENTER RELATED DATA
PF 1=HELP            2=          3=END       4=RETURN            5=                   6=
PF 7=                8=          9=          10=                 11=                  12=CANCEL
```

Since this screen will be processing against a DB2 table, we enter an 'X' next to DB2 CATALOG and enter the fully qualified DB2 table name. When we press ENTER, the SCR3NN screen will be displayed so we can identify which of the screen fields are part of the table we just specified.

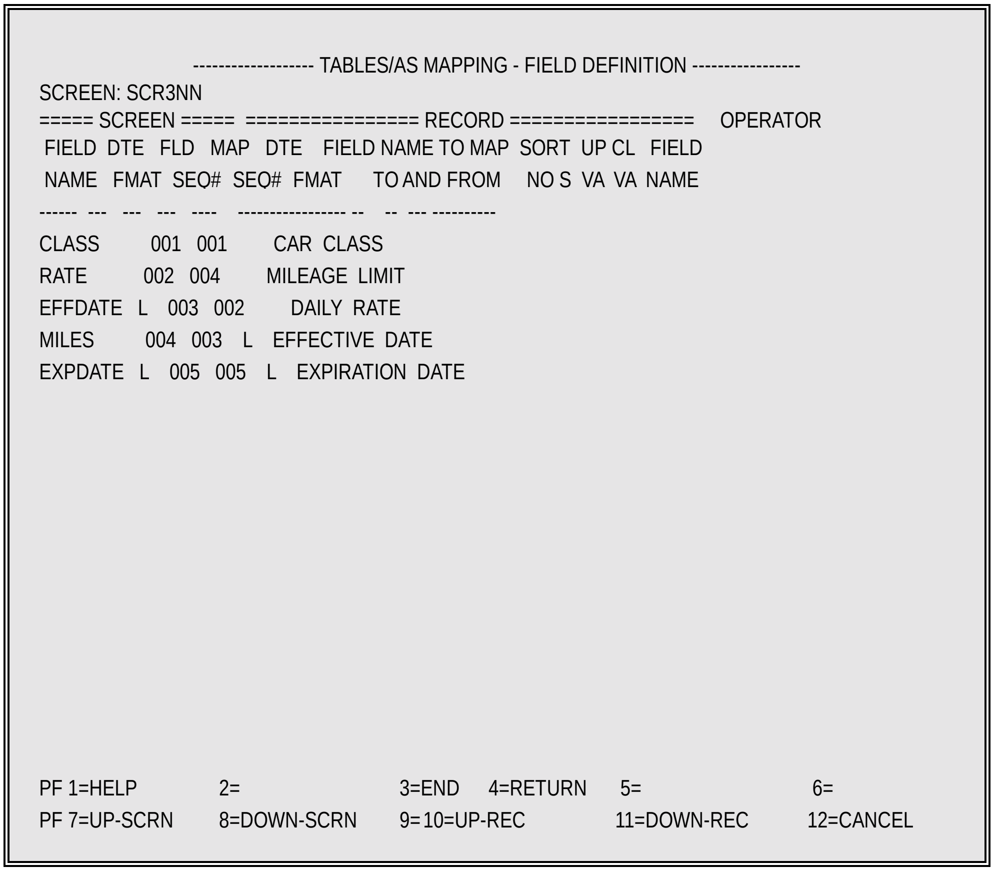
If you want a particular map name, the only chance you have to change the one assigned, is on this panel during the Add a Map process. Position the cursor to the assigned map name (MAP01) and type over up to an 8 character name.



```
 Position 1,1 of the screen is reserved for a Field Start Indicator.  If one is not here TABLES/AS will insert one.

We will design this screen with no assistance from TABLES/AS.  When the enter key was pressed, a blank
screen was presented.  All of the information thus far has simply been typed in.  It will be treated as a literal.
From here on we will place I/O fields on the screen with their associated literals.

 Fnc                  : X
 Class                : M
 Rate                  : M                EFFECTIVE DATE    : M
 MILE LIMIT            : M                           EXPIRATION DATE     : M

Messages + Errors will be displayed on the next line:
 X







You may also specify a literal to tell the user what PF keys to use. Later you will tell TABLES/AS what to do when
the particular PF key is depressed.
PF 1=HELP              PF 3=RETURN                                  ENTER=Process the Table
```

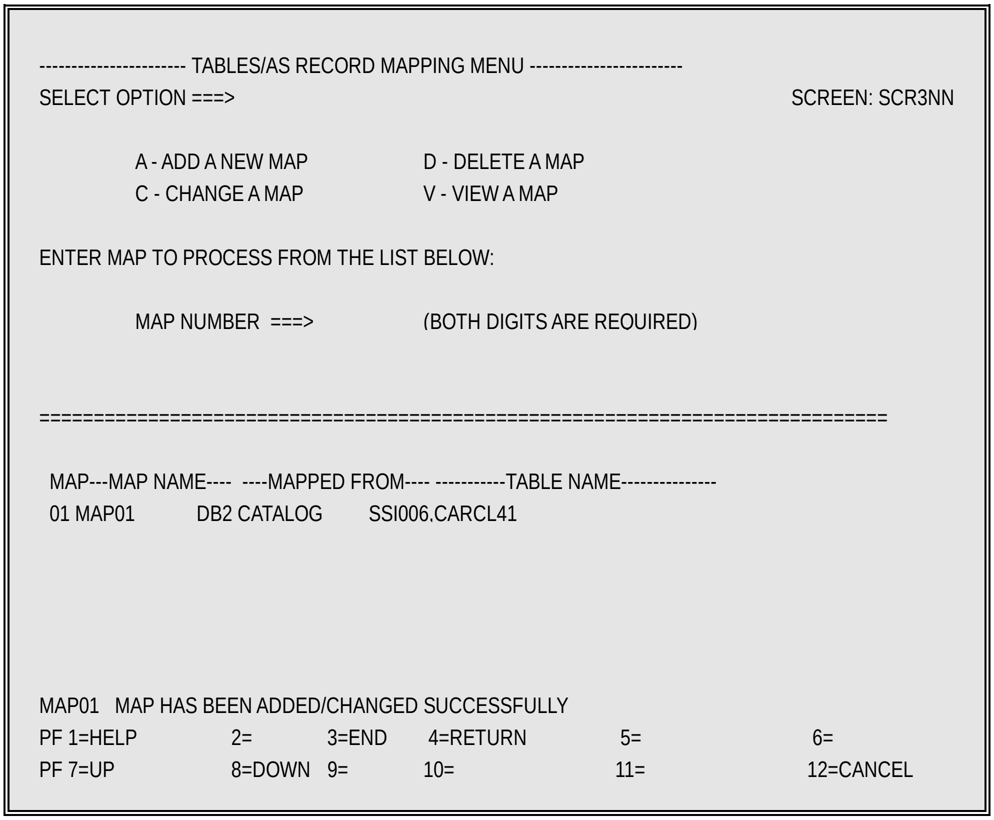
For each of the I/O fields in SCR3NN that are part of the specified table, we enter an 'M'. In this case we have entered five M's.

When we press ENTER, we will be given a panel to use to relate the screen fields with the table columns.



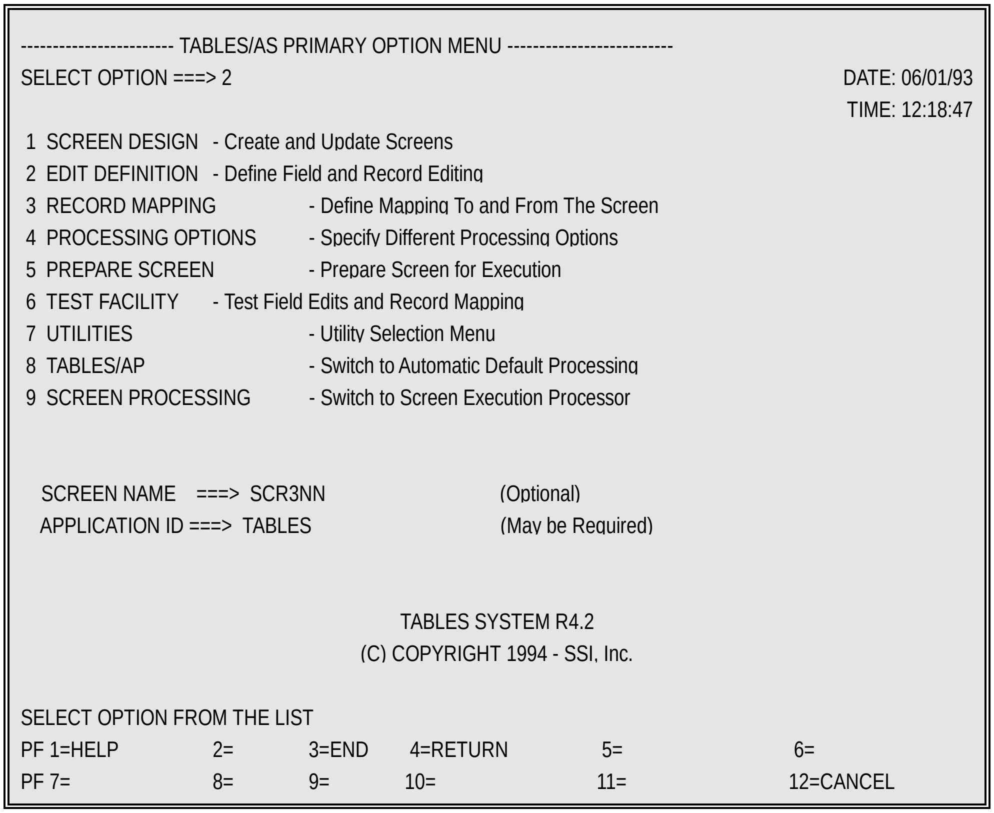
```
                    ------------------ TABLES/AS MAPPING - FIELD DEFINITION ----------------
SCREEN: SCR3NN
===== SCREEN =====  ================ RECORD =================    OPERATOR
 FIELD  DTE   FLD   MAP   DTE    FIELD NAME TO MAP  SORT  UP CL   FIELD
 NAME   FMAT  SEQ#  SEQ#  FMAT      TO AND FROM     NO S  VA  VA  NAME
------  ---   ---   ---   ----    ----------------- --    --  --- ----------
CLASS         001   001          CAR  CLASS
RATE          002   004         MILEAGE  LIMIT
EFFDATE   L   003   002            DAILY  RATE
MILES         004   003    L    EFFECTIVE  DATE
EXPDATE   L   005   005    L    EXPIRATION  DATE










PF 1=HELP           2=                  3=END     4=RETURN     5=                    6=
PF 7=UP-SCRN        8=DOWN-SCRN         9= 10=UP-REC          11=DOWN-REC          12=CANCEL
```

On this screen we enter under MAP SEQ #, the corresponding FIELD SEQ #. This is determined based on the field names and the knowledge of the table. Once we enter the corresponding MAP SEQ #'s, we press PF 3 to save. See Record Mapping for a complete description of this process.



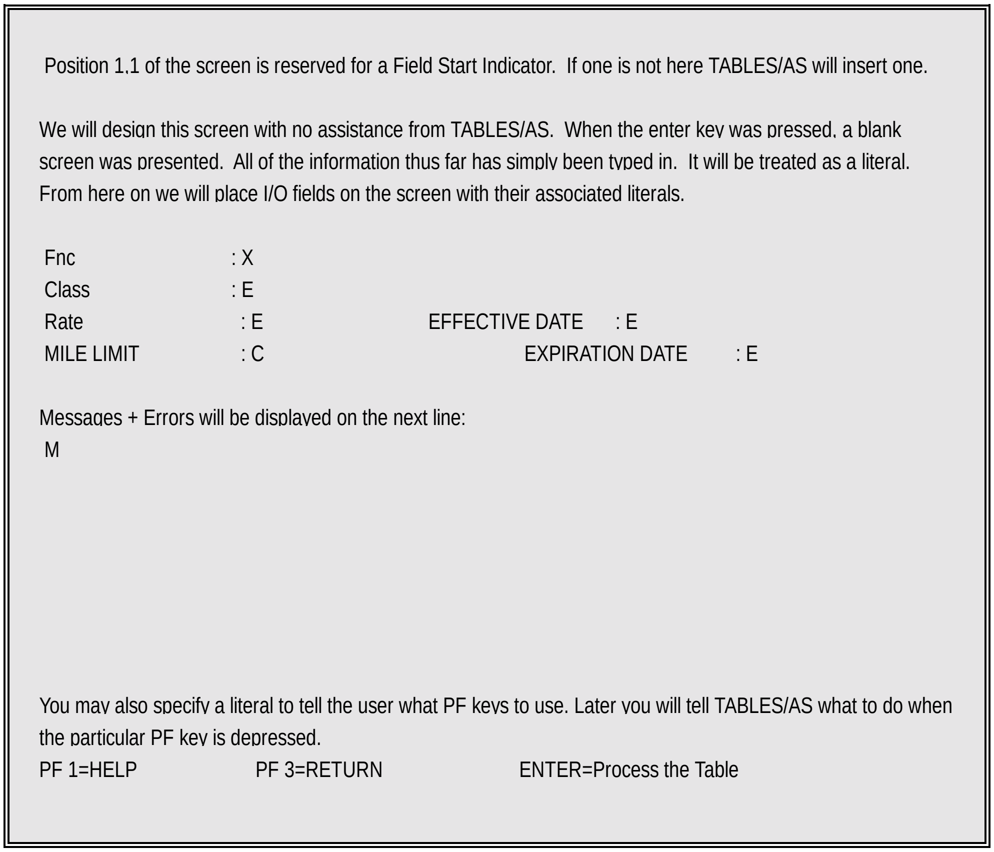
```
---------------------- TABLES/AS RECORD MAPPING MENU -----------------------
SELECT OPTION ===>                                                SCREEN: SCR3NN

         A - ADD A NEW MAP            D - DELETE A MAP
         C - CHANGE A MAP             V - VIEW A MAP

ENTER MAP TO PROCESS FROM THE LIST BELOW:

         MAP NUMBER  ===>             (BOTH DIGITS ARE REQUIRED)


==============================================================================

 MAP---MAP NAME----  ----MAPPED FROM---- -----------TABLE NAME--------------
 01 MAP01      DB2 CATALOG     SSI006.CARCL41




MAP01   MAP HAS BEEN ADDED/CHANGED SUCCESSFULLY
PF 1=HELP        2=       3=END     4=RETURN         5=              6=
PF 7=UP          8=DOWN   9=        10=              11=             12=CANCEL
```

On the bottom half of the screen, we now can see that a map exists, what the table name is and from where it was mapped. The message at the bottom of the screen tells us the map was completed successfully.



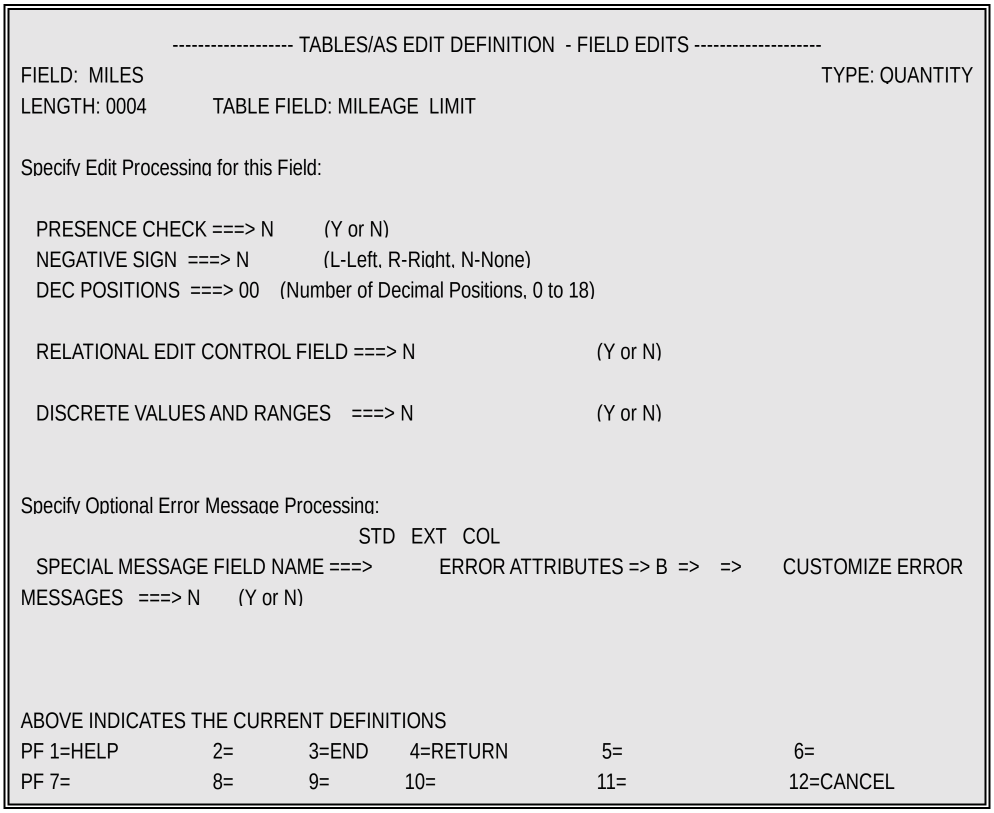
```
----------------------- TABLES/AS PRIMARY OPTION MENU -------------------------
SELECT OPTION ===> 2                                              DATE: 06/01/93
                                                                  TIME: 12:18:47
 1  SCREEN DESIGN   - Create and Update Screens
 2  EDIT DEFINITION   - Define Field and Record Editing
 3  RECORD MAPPING            - Define Mapping To and From The Screen
 4  PROCESSING OPTIONS        - Specify Different Processing Options
 5  PREPARE SCREEN            - Prepare Screen for Execution
 6  TEST FACILITY     - Test Field Edits and Record Mapping
 7  UTILITIES                 - Utility Selection Menu
 8  TABLES/AP                 - Switch to Automatic Default Processing
 9  SCREEN PROCESSING         - Switch to Screen Execution Processor


   SCREEN NAME    ===>  SCR3NN                    (Optional)
   APPLICATION ID ===>  TABLES                    (May be Required)



                              TABLES SYSTEM R4.2
                         (C) COPYRIGHT 1994 - SSI, Inc.

SELECT OPTION FROM THE LIST
PF 1=HELP        2=       3=END     4=RETURN       5=            6=
PF 7=            8=       9=        10=            11=           12=CANCEL
```

We will now select Option 2 - Edit Definition. We will use this for identifying the common message field as well as changing one of the edits for a field.



```
 Position 1,1 of the screen is reserved for a Field Start Indicator.  If one is not here TABLES/AS will insert one.

We will design this screen with no assistance from TABLES/AS.  When the enter key was pressed, a blank
screen was presented.  All of the information thus far has simply been typed in.  It will be treated as a literal.
From here on we will place I/O fields on the screen with their associated literals.

 Fnc                    : X
 Class                  : E
 Rate                    : E                  EFFECTIVE DATE     : E
 MILE LIMIT              : C                           EXPIRATION DATE      : E

Messages + Errors will be displayed on the next line:
 M






You may also specify a literal to tell the user what PF keys to use. Later you will tell TABLES/AS what to do when
the particular PF key is depressed.
PF 1=HELP               PF 3=RETURN                ENTER=Process the Table
```

For the Mile Limit field, we will replace the 'E' with a 'C' to denote we want to change the field edits. We also entered an 'M' in the common message field to identify it as such. Pressing ENTER we will get a panel to change the edits on the Mile Limit field.

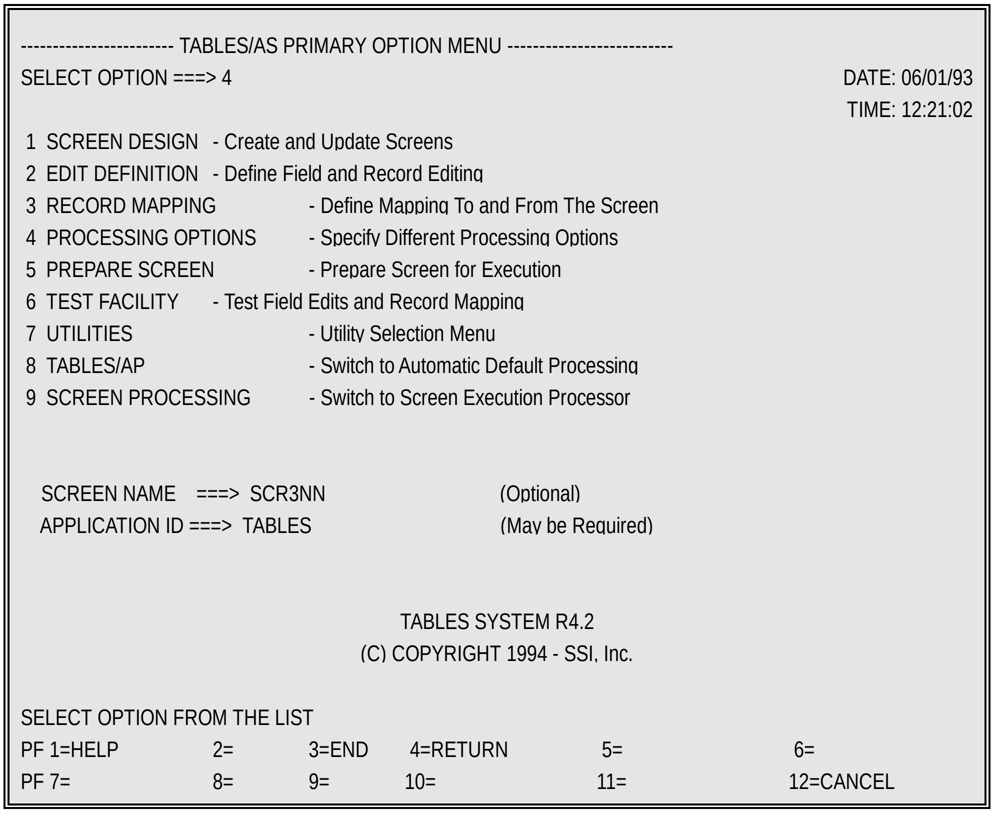


```
                  ------------------ TABLES/AS EDIT DEFINITION  - FIELD EDITS -------------------
FIELD:  MILES                                                                         TYPE: QUANTITY
LENGTH: 0004          TABLE FIELD: MILEAGE  LIMIT

Specify Edit Processing for this Field:

   PRESENCE CHECK ===> N         (Y or N)
   NEGATIVE SIGN  ===> N         (L-Left, R-Right, N-None)
   DEC POSITIONS  ===> 00    (Number of Decimal Positions, 0 to 18)

   RELATIONAL EDIT CONTROL FIELD ===> N                    (Y or N)

   DISCRETE VALUES AND RANGES    ===> N                    (Y or N)


Specify Optional Error Message Processing:
                                       STD   EXT   COL
   SPECIAL MESSAGE FIELD NAME ===>          ERROR ATTRIBUTES => B  =>    =>       CUSTOMIZE ERROR
MESSAGES   ===> N       (Y or N)



ABOVE INDICATES THE CURRENT DEFINITIONS
PF 1=HELP          2=        3=END      4=RETURN          5=                  6=
PF 7=              8=        9=         10=               11=                 12=CANCEL
```

We have changed the edit for Negative Sign from 'L' to 'N'. In this case we did it because we know that Miles will never be negative. When we first pained SCR3NN, we did not allow room for a sign because we knew there was no need for one.

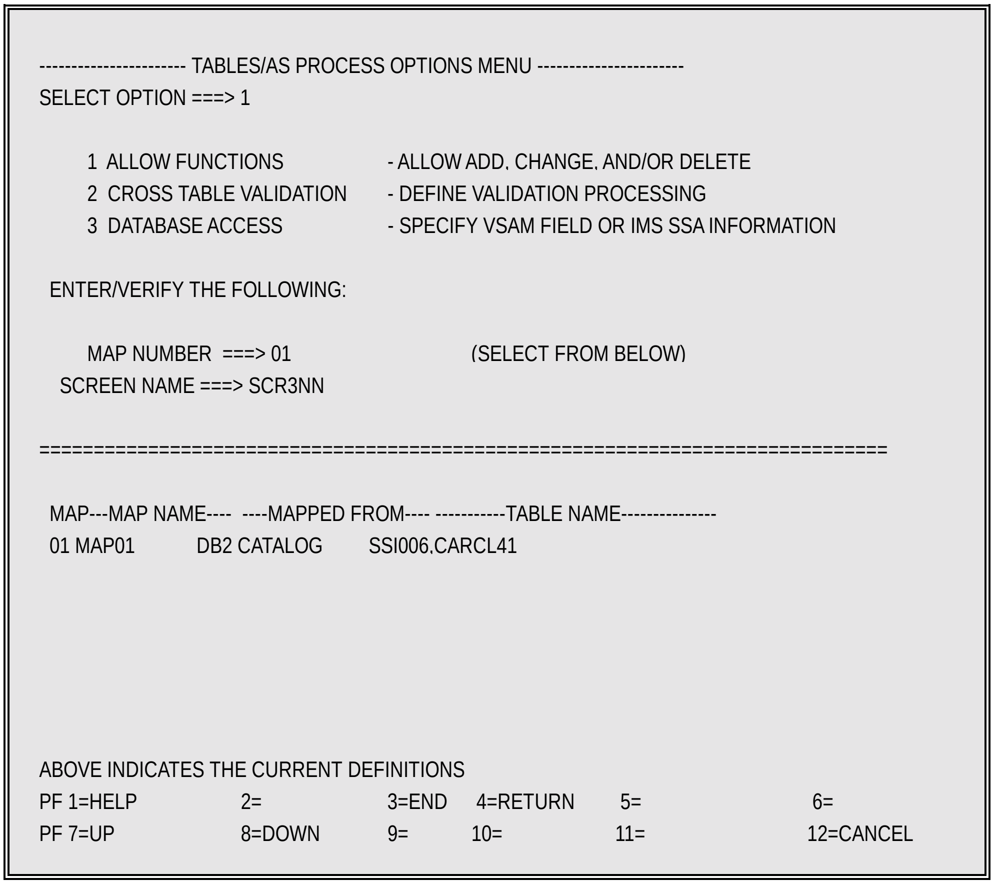
NOTE

One of the benefits of not specifying a table name is the ease of designing a screen that contains only the characters necessary for the particular application. When a table name is entered, all decimal fields will be designed with room for a sign and a decimal point. Designing it yourself, you can exclude either or both. Press ENTER to save the change.



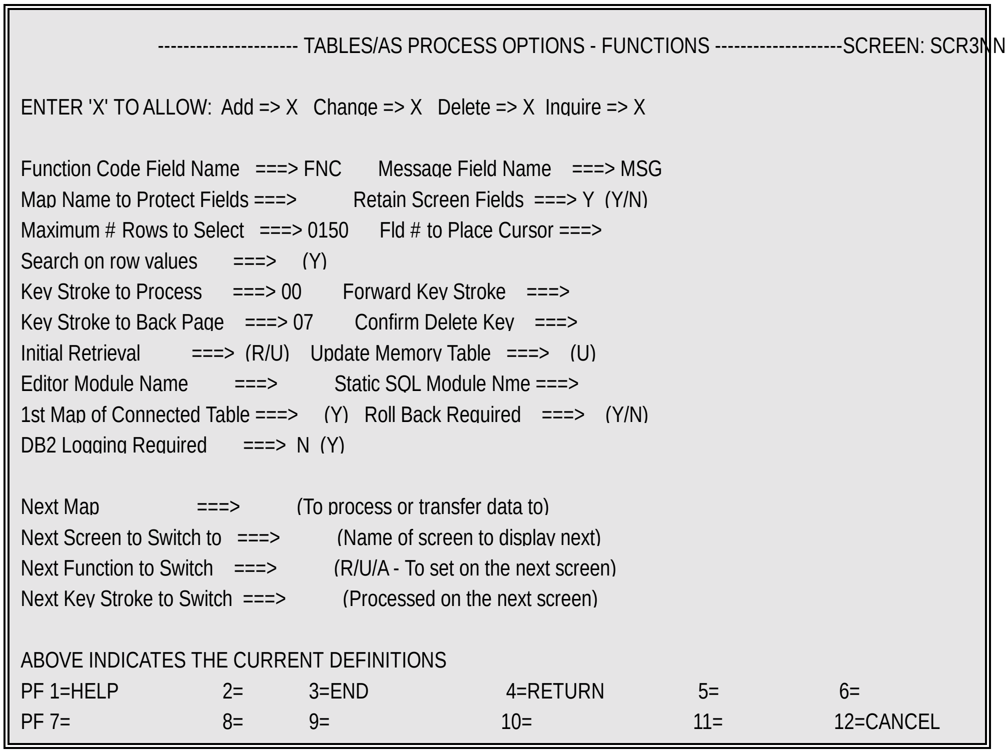
```
----------------------- TABLES/AS PRIMARY OPTION MENU -------------------------
SELECT OPTION ===> 4                                                DATE: 06/01/93
                                                                    TIME: 12:21:02
 1  SCREEN DESIGN   - Create and Update Screens
 2  EDIT DEFINITION   - Define Field and Record Editing
 3  RECORD MAPPING            - Define Mapping To and From The Screen
 4  PROCESSING OPTIONS        - Specify Different Processing Options
 5  PREPARE SCREEN            - Prepare Screen for Execution
 6  TEST FACILITY      - Test Field Edits and Record Mapping
 7  UTILITIES                 - Utility Selection Menu
 8  TABLES/AP                 - Switch to Automatic Default Processing
 9  SCREEN PROCESSING         - Switch to Screen Execution Processor


   SCREEN NAME    ===>  SCR3NN                      (Optional)
   APPLICATION ID ===>  TABLES                      (May be Required)


                              TABLES SYSTEM R4.2
                        (C) COPYRIGHT 1994 - SSI, Inc.

SELECT OPTION FROM THE LIST
PF 1=HELP          2=         3=END      4=RETURN        5=             6=
PF 7=              8=         9=         10=             11=            12=CANCEL
```

We will enter Option 4 **-** Processing Options so we can specify the processing to take place against a table (map) when using this screen.



```
---------------------- TABLES/AS PROCESS OPTIONS MENU ----------------------
SELECT OPTION ===> 1

       1  ALLOW FUNCTIONS              - ALLOW ADD, CHANGE, AND/OR DELETE
       2  CROSS TABLE VALIDATION       - DEFINE VALIDATION PROCESSING
       3  DATABASE ACCESS              - SPECIFY VSAM FIELD OR IMS SSA INFORMATION

  ENTER/VERIFY THE FOLLOWING:

       MAP NUMBER  ===> 01                       (SELECT FROM BELOW)
   SCREEN NAME ===> SCR3NN

================================================================================

  MAP---MAP NAME----  ----MAPPED FROM---- -----------TABLE NAME--------------
  01 MAP01          DB2 CATALOG        SSI006.CARCL41





ABOVE INDICATES THE CURRENT DEFINITIONS
PF 1=HELP          2=            3=END     4=RETURN     5=               6=
PF 7=UP            8=DOWN        9=        10=          11=              12=CANCEL
```

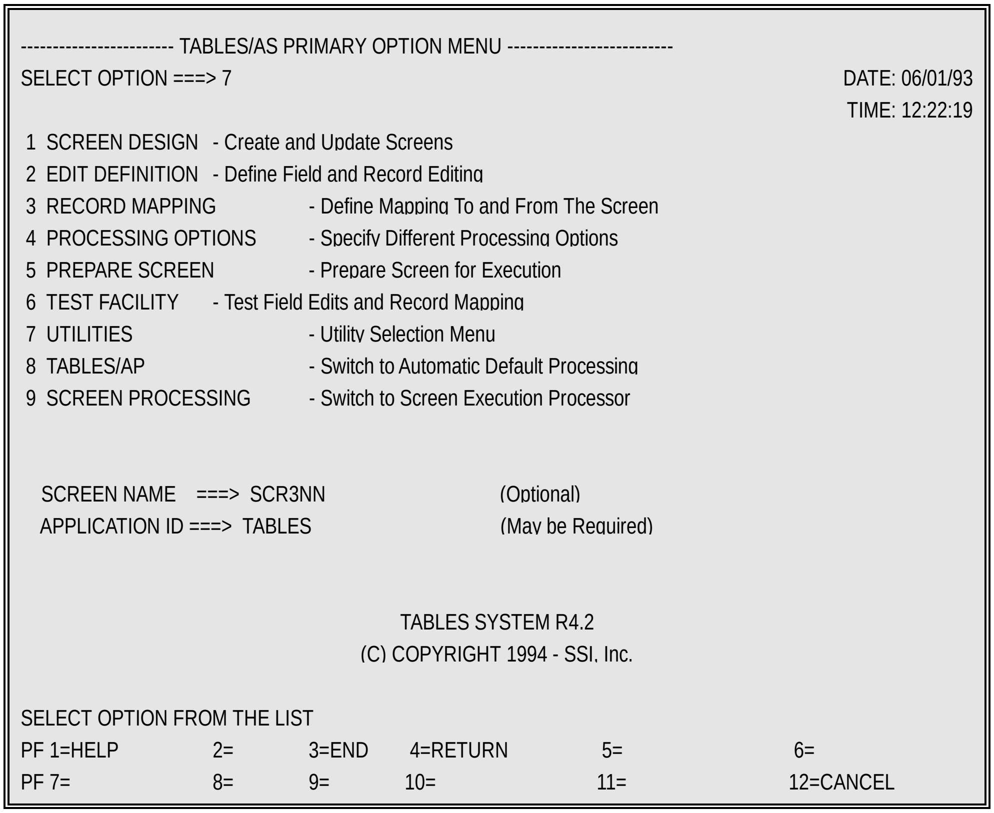
We have entered Option 1 - Allow Functions and Map Number 01. Pressing ENTER now gives us a panel to specify the valid processing for this table on this screen.



```
          --------------------- TABLES/AS PROCESS OPTIONS - FUNCTIONS --------------------SCREEN: SCR3NN

ENTER 'X' TO ALLOW:  Add => X   Change => X   Delete => X  Inquire => X

Function Code Field Name   ===> FNC       Message Field Name    ===> MSG
Map Name to Protect Fields ===>           Retain Screen Fields  ===> Y  (Y/N)
Maximum # Rows to Select   ===> 0150      Fld # to Place Cursor ===>
Search on row values       ===>     (Y)
Key Stroke to Process      ===> 00        Forward Key Stroke    ===>
Key Stroke to Back Page    ===> 07        Confirm Delete Key    ===>
Initial Retrieval          ===>  (R/U)    Update Memory Table   ===>    (U)
Editor Module Name         ===>           Static SQL Module Nme ===>
1st Map of Connected Table ===>     (Y)   Roll Back Required    ===>    (Y/N)
DB2 Logging Required       ===>  N  (Y)

Next Map                   ===>           (To process or transfer data to)
Next Screen to Switch to   ===>           (Name of screen to display next)
Next Function to Switch    ===>           (R/U/A - To set on the next screen)
Next Key Stroke to Switch  ===>           (Processed on the next screen)

ABOVE INDICATES THE CURRENT DEFINITIONS
PF 1=HELP        2=     3=END          4=RETURN        5=         6=
PF 7=            8=     9=             10=             11=        12=CANCEL
```

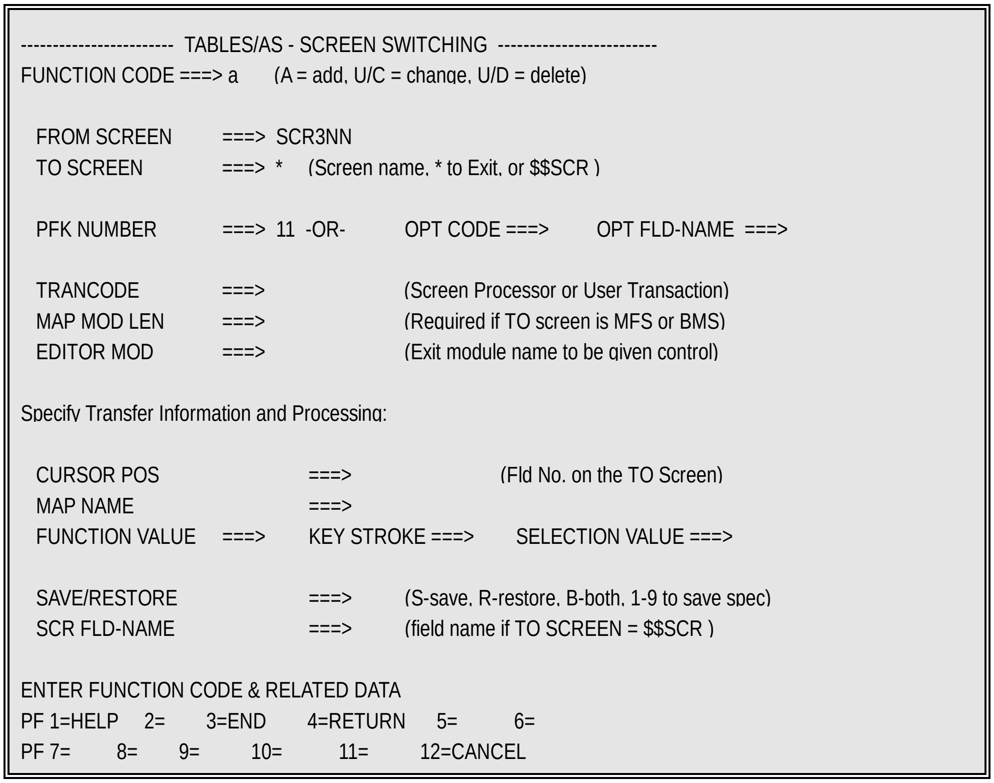
We have made the following entries on the screen:

- X's for Add, Change, Delete & Inquire
- FNC & MSG to tell T**ABLES/AS** which fields are the function code and message field on the screen for this table.
- 0020 to specify the maximum number of rows to retrieve
- 00 to specify the keystroke an operator will press to cause **TABLES/AS** to process against the table related to MAP01 for screen SCR3NN.
- 07 to specify the keystroke an operator will press to cause **TABLES/AS** to backpage to previous pages of rows retrieved for inquiry only. To page forward, press the processing key.



```
----------------------- TABLES/AS PRIMARY OPTION MENU -------------------------
SELECT OPTION ===> 7                                              DATE: 06/01/93
                                                                  TIME: 12:22:19
 1  SCREEN DESIGN   - Create and Update Screens
 2  EDIT DEFINITION   - Define Field and Record Editing
 3  RECORD MAPPING          - Define Mapping To and From The Screen
 4  PROCESSING OPTIONS      - Specify Different Processing Options
 5  PREPARE SCREEN          - Prepare Screen for Execution
 6  TEST FACILITY     - Test Field Edits and Record Mapping
 7  UTILITIES               - Utility Selection Menu
 8  TABLES/AP               - Switch to Automatic Default Processing
 9  SCREEN PROCESSING       - Switch to Screen Execution Processor


   SCREEN NAME    ===>  SCR3NN                    (Optional)
   APPLICATION ID ===>  TABLES                    (May be Required)


                                 TABLES SYSTEM R4.2
                             (C) COPYRIGHT 1994 - SSI, Inc.

SELECT OPTION FROM THE LIST
PF 1=HELP          2=        3=END      4=RETURN        5=              6=
PF 7=              8=        9=         10=             11=             12=CANCEL
```

We have entered Option 7 - Utilities and we will then enter Option 3 - Switching to define a switch from SCR3NN to the TP Monitor being used. The next panel is the Screen Switching panel used to specify operator-controlled switching conditions.



```
-----------------------  TABLES/AS - SCREEN SWITCHING  ------------------------
FUNCTION CODE ===> a      (A = add, U/C = change, U/D = delete)

  FROM SCREEN        ===>  SCR3NN
  TO SCREEN          ===>  *    (Screen name, * to Exit, or $$SCR )

  PFK NUMBER         ===>  11  -OR-        OPT CODE ===>        OPT FLD-NAME  ===>

  TRANCODE           ===>                  (Screen Processor or User Transaction)
  MAP MOD LEN        ===>                  (Required if TO screen is MFS or BMS)
  EDITOR MOD         ===>                  (Exit module name to be given control)

Specify Transfer Information and Processing:

  CURSOR POS                 ===>                      (Fld No. on the TO Screen)
  MAP NAME                   ===>
  FUNCTION VALUE     ===>    KEY STROKE ===>      SELECTION VALUE ===>

  SAVE/RESTORE               ===>        (S-save, R-restore, B-both, 1-9 to save spec)
  SCR FLD-NAME               ===>        (field name if TO SCREEN = $$SCR )

ENTER FUNCTION CODE & RELATED DATA
PF 1=HELP    2=      3=END      4=RETURN     5=        6=
PF 7=       8=      9=        10=        11=        12=CANCEL
```

We have made the following entries on the screen:

- •SCR3NN for the From Screen
- •'*' for the To Screen. The * means to switch to the TP Monitor.
- •'11' for the PF key. This switch occurs when PF key 11 is pressed.

Finally, we will enter Option 5 - Prepare Screen and then Option 9 - Screen processor to test the screen and its functioning. See Appendix H - Test **TABLES/AS** Default & Custom Screens.

Appendix K

Migrate: Test to Production

Suggested set of steps required to migrate **TABLES/AS** screens from a test environment to a production environment:

1.Install **TABLES/AS** system in production: This will be performed once.

a)(Ideally) install all SSI supplied **TABLES/AS** transactions, but (minimally) install TS09 and TSAP transactions or their equivalents.

•Keep in mind that you will be able to install as many TS09 equivalent transactions as you need.

•If you install all **TABLES/AS** transactions on the production system, you should RACF protect all transactions except TS09 and TSAP.

b)Install all PPT entries as supplied by SSI in your production CICS. We do not recommend installing a subset of PPT entries.

c)Install **TABLES/AS** load library as supplied by SSI in your production CICS. We do not recommend installing a subset of **TABLES/AS** programs.

2.Migrate new/modified **TABLES/AS** screens into production: This will be performed each time you are migrating new screens.

a)Each new TS09 equivalent transaction must be put into PCT using the same TSCM0900 program.

b)Each new **TABLES/AS** transaction that is invoked by a user must be entered in the **TABLES/AS** control table **TXNMNU** using the TSAP transaction. An example of a user-invoked transaction would be a transaction code entered by a user to display a menu screen.

3.Establish new DB2 plans.

a)Each DB2 plan must be bound and authorized.

•Keep in mind that static SQL modules produce DBRMs. All SQL modules that are executed from a given plan must be bound together.

•Each new transaction and its corresponding plan must be included in the DSN RCT table.

4.Create static SQL modules: A static SQL module may be created for each table processed by a screen.

a)Move the static SQL source into a production library.

b)Change the fully qualified test DB2 table name to the fully qualified production DB2 table name to the fully qualified production DB2 table name.

c)Compile static COBOL module into a production load library.

•You must include the resulting DBRM member in your DB2 plan.

d)Add an entry in the production PPT to include static COBOL program.

5.Create static screen modules: A static screen module may be created for any screen.

a)If you are using a different naming standard for your production screens, consider the following:

•You must copy your test screen under the production name on your test system itself.

•You may want to lock your production screen using the **TABLES/AS** control table **SCRLOK**. This can be done using the TSAP transaction on your test system. Once a screen is locked, it cannot be changed without unlocking it.

b)Using JCL401M7, create a static screen module.

c)Move the static screen load module into your production library.

d)Add an entry in the production PPT to include static ASSEMBLER screen program.

e)Migrate the screen format and related information:

•Extract the screen using STA of JCLMIG02. If table name change is required, specify the corresponding production table name under the DD name CHGDB2TB.

•Keep in mind that a PDS file may be specified for CHGDB2TB which may contain all test and production table names for all screens.

•The user running the extract job must be authorized to perform the SELECT access to all **TABLES/AS** control tables. This is required because the extract program uses dynamic SQL calls.

•Load the screen extracted in STA above into the production system. The load should be performed using STB of JCLMIG02.

•The load step uses a DB2 plan which should be granted for EXECUTE to the user running the load job. The load program uses static SQL calls only and therefore GRANT SELECT is not required.

6.Migrate effectivity definitions: This is performed once for the tables under **TABLES/AS** effectivity definition.

a)Extract effectivity definition from the test system using STC of JCLMIG02. If a table name change is required, specify the corresponding production table name under the DD name CHGDB2TB.

Keep in mind that a PDS file may be specified for CHGDB2TB which may contain all test and production tables names for all screens. This PDS file may be the same file which was used during the STA of screen migration.

The extract step uses a DB2 plan which should be granted for EXECUTE to the user running the extract job. The extract program uses static SQL calls only and therefore GRANT SELECT is not required.

b)Load effectivity definition using STD of JCLMIG02.

The load step uses a DB2 plan which should be granted for EXECUTE to the user running the load job. The load program uses static SQL calls only and therefore GRANT SELECT is not required.

7.Move user editor programs: You must migrate the user-written editor programs into your production libraries.

a)Move source and load modules as required by your installation. No recompilation is required.

b)Add entries in the production PPT to include COBOL editor programs.

8.Establish security: You should use your standard security procedures to secure transactions and tables.

a)Transactions may be secured using RACF, ACF2, etc.

b)DB2 tables should be secured by DB2 plans.

9.Naming requirements: The names used for the static SQL modules, static screen modules and the user-written editor programs must be unique.

10.**TABLES/AS** control table refresh: If migration is performed while the production CICS system is running, then refresh **TABLES/AS** control tables using one of the methods below:

a)Stop/Start CICS.

b)Run TSLU transaction to free **TABLES/AS** control tables.

c)Run TS01 transaction and execute the Utility Option 7 to reload **TABLES/AS** control tables.

Index