

9. SCREEN MANAGEMENT

9.1 Introduction

Screen management consists of several CREDIT subroutines utilizing format I/O control, format I/O control was described in section 6.6 of this manual. This package enables a complete screen of information to be displayed, typically with a number of areas where the operator has to enter or change items. These "input fields" can initially contain an existing value or they could be set to blank.

Each prompt and input field is in a format list in the application program. The format list is attached with the ATTFMT instruction prior to screen management being called, the DISPLAY being done within the package.

With display units that permit both high and low intensity display, the items entered from the keyboard will be displayed in high intensity mode; those items of text emanating from the format list will be displayed in low intensity.

This package is held as CREDIT source code in a file called SCREEN, in the user area SCREEN on the system pack. Before the package can be used it has to be copied to the application user area. As the package is written in CREDIT, the routines can be easily modified; for example, the package as written expects the DDIV of the application to be held on a file called SPDDIV; if this is not the case, then the DDUM statement in the screen management package must be altered accordingly.

9.2 Requirements of screen management

Screen management requires that a number of definitions are made within the application. These are three keytables, some extra data items, data sets for keyboard, display and associated printer, a format table of error messages and a number of checking routines. The data item, data sets and format control statements must be included in the DDIV for the application. It is usual to group all other items in a file called 'SPLITT', this file being incorporated into the screen management module at translation time; the screen management package use the INCLUDE directive to bring across the information from SPLITT.

If the module SPLITT is not used, then the INCLUDE directive in the module SCREEN must be removed, or if a module name other than SPLITT is used, the INCLUDE directive must be changed to the new name.

Note:

It is not possible to have an IDENT statement in files which are to be included via the INCLUDE directive. A sample layout of SPLITT is included at the end of this chapter.

CREDIT PROGRAMMERS GUIDE

9.2.1 Data items

The data items given in the table below are required for use by the screen management package and must be defined in the DDIV of the application program. These data items must be accessible to the task which is initiating the perform of screen management.

Data items required by screen management			
Name	Type	Size	Use
SPBINW1	BINARY	One word	Work variable
SPBINW2	BINARY	One word	"
SPBINW3	BINARY	One word	"
SPBINW4	BINARY	One word	"
SPCHANGE	BOOLEAN	One bit	Fields changed
SPPROMPT	BOOLEAN	One bit	Display form
SPINPUT	STRING	Length greater then Largest input field	Input item
SPERCALL	BOOLEAN	One bit	Work variable
SPSTRGW1	STRING	Greater than one	"

9.2.2 Data sets

Screen management uses three data sets, a printer, display unit and a keyboard. These data sets must be defined in the application program with the names shown in the table below. The buffer for the display unit must be large enough to hold the longest line plus the associated control characters; the buffer for the print unit must be at least the same size as the display. The two buffers may be shared. The data division must include the FMTCTL directive to link the two devices and enable some of the format control I/O instructions to be used by screen management.

Data sets used by screen management

Name	Description
SPDSPRT	Printer data set
SPDSSCRN	Display data set
SPSDYKB	keyboard data set

9.2.3 Entry points

There are a number of entry points for the package, as described below. The required entry points must be defined as external (EXT) in the application program. The entry points are subroutine names, hence they will be accessed by a perform (PERF) or indexed perform (PERFI) instruction in the application program.

- SPCLRN The prompts and titles described in the attached format list will be displayed on the screen, only if the boolean data item SPPROMPT has the value 'TRUE'; if SPPROMPT has the value 'FALSE' then they will not be displayed. Irrespective of the value of SPPROMPT, the old contents of the data items in the format list will be displayed, the cursor will be placed at the first data field and will wait for the user to enter information, or move the cursor to another data field.
- SPCLRS The prompts and titles described in the attached format list will be displayed on the screen, only if the boolean data item SPPROMPT has the value 'TRUE'; if SPPROMPT has the value 'FALSE' then they will not be displayed. Irrespective of the value of SPPROMPT, the old contents of the data items in the format list will be displayed only if the no clear (NCLR) option is specified in the FKI command, the other fields will be left blank; the cursor will be placed at the first data field and will wait for the user to enter information, or move the cursor to another data field.
- SPCLRA The prompts and titles described in the attached format list will be displayed on the screen, only if the boolean data item SPPROMPT has the value 'TRUE'; if SPPROMPT has the value 'FALSE' then they will not be displayed. Irrespective of the value of SPPROMPT the old contents of the data items in the format list will not be displayed, the cursor will be placed at the first data field and will wait for the user to enter information, or move the cursor to another data field. The use of this call with an SPPROMPT value of FALSE should be avoided as it results in a blank screen.
- SPERR When screen management detects an error the acoustic alarm is sounded and an error message is displayed on the last line of the screen. This entry point allows errors detected in the application program to be displayed in a similar manner. After the error message has been displayed control will return to the application. The binary data item SPBINW4 is used as an index to the format table SPFTBERR which holds the format lists for the error messages.
- SPERR2 Like SPERR this entry point allows an error detected in the application to be displayed; in addition it allows the errored field to be corrected before control is transferred back to the application. The binary data item SPBINW4 is used as an index to the format table SPFTBERR which contains the format lists for the error messages.

9.2.4 Keytables

Screen management uses three keytables, one to control editing of the current field, one to check the first character entered in a field and one to check all subsequent characters in the field. The hexadecimal values generated by the actual keys on the keyboard used in the application must be present in the keytables, and the entries are position dependant. As supplied the package expects the keytables to be called SPKTAB1, SPKTAB2 and SPKTAB3.

SPKTAB1 is used to check the first character entered in a field. If the character entered matched to an entry in the keytable then the appropriate action will be taken, for example if the 'TAB LEFT' key had been entered then the first input field on the current line is made current.

If the character entered did not match with any of the table entry and was not a valid alphanumeric character then the acoustic alarm in the display unit will be sounded, and the cursor remains at the start of the field.

If the character entered was a valid alphanumeric character, and did not match with the keytable then it will be transferred to the string data item SPINPUT and the remainder of the field on the display unit will be filled with periods '.'.

The field size is either the limit specified in the MAXL option on the DYKI instruction, or the length of the string, if the MAXL is not specified.

SPKTAB2 is used to check the second and subsequent characters entered. If a match is found with the keytable then the specified action will be taken.

If the character entered was not found in the keytable, and it was not an alphanumeric character then the acoustic alarm will be sounded and the cursor remains in its current position.

If a valid alphanumeric character was entered and a match was not found with the keytable then the character is placed in the string data item SPINPUT and the cursor is positioned at the next character; if the field has been filled then all characters entered until a valid terminator is entered, except when the NEOI flag is set in the DYKI instruction. In this case, when a field is filled the next field will become current, the cursor is then positioned at the first input position of that field.

CREDIT PROGRAMMERS GUIDE

SPKTAB3 is used for holding the character codes to be used for editing the current field. When the editing has been completed and the field contains the desired contents then one of the function keys will be used to make the next field current. If the DYKI instruction has the NEOI flag set then when the current field is filled, the next field will become current.

Note:

If a keytable entry is not required then the value of X'FF' should be entered in the appropriate position. This is often called the NOKEY value.

If a 6236 keyboard is being used then it is necessary to include a special character conversion table CTAB01, which specifies the code to be generated by each key. When a table entry is not required it should be set to the bell character X'07'.

An example of the three keytables is given in SPLITT at the end of this section. The position and description of the function key codes within the three keytables is described in the next section.

CREDIT PROGRAMMERS GUIDE

9.2.5 Keytable entries

9.2.5.1 Editor functions

Non-destructive space - Position 1 in SPKTAB3

When this key is pressed the cursor is moved one position to the right; the acoustic alarm is sounded if an attempt is made to go beyond the current last character of the input field. Also it is not possible to position the cursor beyond the current contents of the field.

If for example, the field had been specified with a length of ten characters, and currently the field held only five characters then the acoustic alarm would be sounded if an attempt was made to move the cursor to position six.

If the NEOI flag is set in the DYKI instruction, and this key is pressed when the cursor is in the last position of the field a return will be from edit and the next field will become current.

If the NEOI flag was not set and the cursor was currently in the last position of the field when this key was pressed, the acoustic alarm will be sounded and the cursor position will not be changed.

Non-destructive backspace - Position 2 in SPKTAB3

The cursor is moved one position to the left. The acoustic alarm will be sounded if an attempt is made to go beyond the left hand field limit.

Insert - Position 3 in SPKTAB3

A space character is inserted into the string at the current cursor position, the characters to the right of the cursor being moved one place to the right. The character string will be truncated if it exceeds the field limit.

Delete - Position 4 in SPKTAB3

The character at the current cursor position is deleted, the characters to the right of the cursor will then be shifted left one place, the right most position being replaced by a space character.

Backspace - Position 1 in SPKTAB1 and SPKTAB2

When this key is pressed, the cursor is moved one place to the left and the corresponding position on the display is replaced by a period '.'. If backspace is performed at the first position of the current input field, then the previous contents of the data item, if any, will be displayed.

CREDIT PROGRAMMERS GUIDE

9.2.5.2 Clear functions

Clear 1 - Position 2 in SPKTAB1 and SPKTAB2, position 5 in SPKTAB3

The input field on the display unit and the corresponding data item will be cleared, the cursor is placed at the first position of the current input field. If this key is pressed when in EDIT mode the edit will be terminated and the next input field made current.

Clear 2 - Position 3 in SPKTAB1 and SPKTAB2, position 6 in SPKTAB3

The cursor is placed at the first position of the current input field, the previous contents of the data item, if any, will be displayed. If this key is pressed when in EDIT mode, the edit will be terminated and the next field made current.

Clear 3 - Position 7 in SPKTAB3

Characters from the current cursor position to the end of the field are deleted and the edit operation is terminated, the next field is made current.

End of item - Position 8 in SPKTAB3 and position 4 in SPKTAB1 and SPKTAB2

The current operation is completed, the next field is made current.

9.2.5.3 Cancel functions

Cancel 1 - Position 5 in SPKTAB1 and SPKTAB2, position 9 in SPKTAB3

A return is made to the application program, with the index item SPBINW2 containing the value one. Note no check is made on the contents of the field via SPAPPL etc..

Cancel 2 - Position 6 in SPKTAB1 and SPKTAB2, position 10 in SPKTAB3

This is the same as Cancel 1 but the index SPBINW2 will contain the value two.

CREDIT PROGRAMMERS GUIDE

9.2.5.4 Tabulation functions

Tab forwards - Position 7 in SPKTAB1 and SPKTAB2, position 11 in SPKTAB3

Make the next input item current. If there are no more input items on the screen then the cursor is re-positioned at the start of the current field. The next field can be on the same line as the current field, or it may be on a subsequent line. If there is more than one input field on a line, then the Tab forwards will make the left most field on the next line current.

Tab backwards - Position 8 in SPKTAB1 and SPKTAB2, position 12 in SPKTAB3

Make the preceeding input item current. If the current input field is the first on the screen then the cursor will be positioned at the start of the current field.

Tab home - Position 9 in SPKTAB1 and SPKTAB2, position 13 in SPKTAB3

Move the cursor to the first position of the first input field of the current screen format.

Tab left and down - Position 10 in SPKTAB1 and SPKTAB2, position 14 in SPKTAB3

Tabulate to the first input field on the next line. If no input field exists on the screen, below the position of the current screen then no action will be taken.

Tab left - position 11 in SPKTAB1 and SPKTAB2, position 15 in SPKTAB3

Place the cursor at the first character position of the left most field of the current line.

Tab right - Position 12 in SPKTAB1 and SPKTAB2, position 16 in SPKTAB3

Place the cursor at the first character position of the right most field of the current line.

Tab down - Position 13 in SPKTAB1 and SPKTAB2, position 17 in SPKTAB3

Move the cursor to the data item on the next line, which is in the position nearest the current input field. If the line has two fields equidistant from the current field then the left most of the two fields will be selected. If there are no fields below the current field on the screen then no action will be taken.

Tab upwards - Position 14 in SPKTAB1 and SPKTAB2, position 18 in SPKTAB3

Move the cursor to the data item on the preceding line, which is in the position nearest the current input field. If the preceding line has two fields equidistant from the current field then the left most of the two fields will be selected. If there are no fields above the current field on the display then no action will be taken.

CREDIT PROGRAMMERS GUIDE

9.2.5.5 Miscellaneous functions

Copy - Position 15 in SPKTAB1 and SPKTAB2, position 19 in SPKTAB3

A copy of the screen will be produced on the printer. This will have the effect of terminating input to the current field,

Duplicate - Position 16 in SPKTAB1 and SPKTAB2, position 20 in SPKTAB3

The contents of the field specified in the duplicate option will be moved to the current input field. If the option was not specified in the FKI instruction, error message number four in the format table SPFTBERR, will be displayed on the last line of the screen and the acoustic alarm sounded.

Edit - Position 17 in SPKTAB1 and SPKTAB2, position 21 in SPKTAB3

The keyboard will be set to edit mode. If the package is already in edit mode this instruction is ignored and the acoustic alarm sounded.

Enter - Position 18 in SPKTAB1 and SPKTAB2, position 22 in SPKTAB3

This causes a return to the application. If compulsory fields have not been filled, then an error message is produced, and the empty compulsory field nearest the top left corner is made current.

Application function keys - Positions 18 and above in SPKTAB1 and SPKTAB2, positions 22 and above in SPKTAB3

These are treated as an ENTER. On return to the application the index item SPBINW2 will hold the number of the application key in the following form:- if application key 1 was pressed then SPBINW2 will hold the value 4, if application key 2 was pressed then SPBINW2 will hold the value five, and so on.

CREDIT PROGRAMMERS GUIDE

9.2.6 Format table

The error messages produced by screen management are held as a series of one line format lists, the names of these format lists being given in the format table SPFTBERR. The first five positions of this table have a predefined meaning within screen management, subsequent positions being available for holding application related error messages. The meaning applied to the first five elements is given in the table below:-

Format table entries for error messages in screen management

Pos.	Use
1	Number of characters entered is less than that specified in MINL
2	Not used (available to appl)
3	I/O error (e.g. time out)
4	Illegal end of item key (dupl. key pressed but not specified in FKI)
5	Compulsory field still blank
6	And upwards available to appl.

When an error is detected either in the application, or screen management, the acoustic alarm will be sounded and the appropriate message displayed on the last line of the screen.

When an error message has been displayed on the screen - not via SPERR, a correct value may be entered in the errored field only after one of the following function keys has been pressed:- CLEAR 1, CLEAR 2, EDIT, CANCEL 1, CANCEL 2.

CREDIT PROGRAMMERS GUIDE

9.2.7 Tabulation validation routine

It is necessary to write a subroutine called SPTCHK. This routine is called if the field that has been tabulated to has the GTAB option set, it will be called before data can be entered to the field. Hence a checking program can be written to see if the field can be made current in the existing environment, providing a means of controlling the cursor position. The result of this routine is held in SPBINW3; in the case of a correct tabulation this data item will contain zero, and if the tabulation is to continue it will contain a non zero value.

Contents of SPBINW3	Action
Zero	Correct tabulation
Any other value	Tabulation will continue

CREDIT PROGRAMMERS GUIDE

9.2.8 Value check routines

Data items entered may be checked via a user written subroutine before the value is transferred from the DYKI input area (SPINPUT) to the data item identifier specified in the FCOPY or FMEL instruction. These subroutines are written by the user and are called:-

- . SPAPPL
- . SPCHK1
- . SPCHK2
- . SPCHK3
- . SPCHK4
- . SPCHK5
- . SPCHK6
- . SPCHK7

These routines must be present as ENTRY points in the application, as they are defined as EXTERNAL in the SCREEN module.

The SPAPPL subroutine will be called after the DYKI instruction, if the associated field descriptor (FKI) contained the APPL option. The APPL option is followed by a number between -32768 and 32767. When the routine is called, the number will be held in the data item SPBINW3, to be used in an indexed or ordinary branch.

The subroutines SPCHK1 through SPCHK7 will be called after the DYKI instruction but before the SPAPPL routine, if the associated field descriptor (FKI) contained the SCHK option. The SCHK option is followed by a number between 1 and 7. This number is used to call the appropriate routine.

It is possible to specify both APPL and SCHK routines in the same FKI, in which case the SPCHK routine will be called first, the SPAPPL routine will be called when a RETURN is encountered in the SPCHKx routine.

The data items passed from screen management to the application checking routine are:-

- SPINPUT The string data item containing the data that has been input from the keyboard.
- SPBINW1 A binary data item containing the number of characters transferred
- SPBINW2 A binary data item containing the converted end of item key index in the key table
- SPBINW3 A binary data item containing the value defined in the APPL option of the FKI command, hence a number between -32768 and 32767.

CREDIT PROGRAMMERS GUIDE

The output from the SPCHKx series of routines and SPAPPL is shown below. The data item SPINPUT may be altered by one of these routines, but if its length is changed, then SPBINW1 must also be altered.

The data items passed from the application checking routine to screen management are:-

- SPINPUT The string data item containing the data that has been input from the the keyboard, and possibly changed by the routine.
- SPBINW1 A binary data item containing the number of characters transferred to the screen management package in SPINPUT.
- SPBINW2 A binary data item containing the converted end of item key index in the key table.
- SPBINW3 A binary data item containing one of the values from the table shown below.
- SPBINW4 A binary data item containing an index to the error message to be displayed. If no message is to be displayed, then this will contain the value zero.

Contents of SPBINW3	Action
Zero	The contents of the data item SPINPUT will moved to the data item of the current input field and displayed on the screen when the 'REWRT' option is specified in the FKI command. Screen management will continue according to the end of item key.
One	The contents of the data item will be displayed and moved to the data item of the current input field. Screen management will continue according to the end of item key held in SPBINW2.
Two	The data item is <u>not</u> moved to the current input field, the cursor is set to the begining of the current field, and input can be performed on this field.
Three	Error condition, the binary data item SPBINW4 will contain the index to the error message in SPFTBERR, when SPBINW4 contains zero no message will be displayed.

CREDIT PROGRAMMERS GUIDE

Example of the use of value check routines

```
FNL
FTEXT      'PLEASE ENTER YOUR USER CODE '
FKI        29,MAXL=4,SCHK=1
FCOPY      USER
FNL
FTEXT      'PLEASE ENTER THE CURRENT DATE '
FKI        33,MINL=6,SCHK=5,APPL=8
FCOPY      DATE
FMEND
```

When the user code is entered it is validated by the subroutine SPCHK1. The number of characters entered, excluding the EOI character, will be held in SPBINW1, the actual characters entered in SPINPUT. The transfer of characters from SPINPUT to USER takes place when the FCOPY instruction is encountered.

When the date is entered the subroutine SPCHK5 will be called first and then SPAPPL; when SPCHK5 is called, SPBINW4 will hold the value 5; when SPAPPL is called, SPBINW3 will hold the value 8. If an error was detected by SPCHK5 then the message would be displayed etc. without the SPAPPL routine having been called.

Note:

- If the number of characters in SPINPUT is changed, then SPBINW1 also must be changed to the new number.
- The end of item key index may be changed, for example the EOI code X'03' may be changed to the ENT code X'17'.

REL 4.1 790523 * PROCEDURE DIVISION * IDENT SCREEN * DATE 791015 * PAGE 0002

LINE	LABEL	OPCODE	OPERANDS	COMMENT
0009				
0010	*	PDIV		
0011		ENTRY	SPCLRA	CLEAR ALL VARIABLE FIELDS
0012		ENTRY	SPCLRS	CLEAR SOME VARIABLE FIELDS
0013		ENTRY	SPCLRN	CLEAR NO VARIABLE FIELDS
0014		ENTRY	SPERR	DISPLAY ERROR MESSAGE, UPDATE
0015				.. CURRENT FIELD & CONTINUE IN
0016				.. FORMAT.
0017		ENTRY	SPERR2	.. DISPLAY ERROR MESSAGE, UPDATE
0018				.. CURRENT FIELD & RETURN.
0019	*			
0020		EXT	SPCHK1	.. STANDARD CHECK ROUTINE NO. 1
0021		EXT	SPCHK2	.. STANDARD CHECK ROUTINE NO. 2
0022		EXT	SPCHK3	.. STANDARD CHECK ROUTINE NO. 3
0023		EXT	SPCHK4	.. STANDARD CHECK ROUTINE NO. 4
0024		EXT	SPCHK5	.. STANDARD CHECK ROUTINE NO. 5
0025		EXT	SPCHK6	.. STANDARD CHECK ROUTINE NO. 6
0026		EXT	SPCHK7	.. STANDARD CHECK ROUTINE NO. 7
0027		EXT	SPAPPL	.. USER ROUTINE TO HANDLE
0028				.. APPL VALUES
0029		EXT	SPTCHK	.. USER ROUTINE TO EVALUATE
0030				.. CONDITIONAL-TABULATION
0031				.. APPL VALUES
0032				.. CONDITIONAL-TABULATION
0033	*			
0034		EXT	EMPTYT	.. ASSEMBLY SUBROUTINE EMPTYT
0035				.. TEST IF DATA ITEM IS EMPTY
0036	*			

REL 4.1 790523 * PROCEDURE DIVISION * IDENT SCREEN * DATE 791015 * PAGE 0003 *

LINE	LABEL	OPCODE	OPERANDS	COMMENT
0037		EJECT		
0038		INCLUDE	SPLITT,LIST	
0000-	*			
0001-	*S P L I T T			
0002-	*THIS MODULE IS USED BY SCREEN-PACKAGE.			
0003-	*IT CONTAINS KEYTABLES AND ERROR PRINTOUTS			
0004-	*USED IN CONNECTION WITH SCREEN PACKAGE			
0005-	*			
0006-	BSP	EQV	X'B9'	
0007-	CLEAR	EQV	X'A3'	
0008-	CANCL	EQV	X'A2'	
0009-	CANCL	EQV	X'99'	
0010-	REWRT	EQV	X'95'	
0011-	EOL	EQV	X'90'	
0012-	FWD	EQV	X'B6'	
0013-	BWD	EQV	X'A5'	
0014-	HOME	EQV	X'B8'	
0015-	LDOWN	EQV	X'88'	
0016-	LEFT	EQV	X'A1'	
0017-	RIGHT	EQV	X'8A'	
0018-	DOWN	EQV	X'87'	
0019-	UP	EQV	X'82'	
0020-	HCOPY	EQV	X'91'	
0021-	DUP	EQV	X'96'	
0022-	CFWD	EQV	X'AD'	
0023-	CBWD	EQV	X'A1'	
0024-	ENT	EQV	X'92'	
0025-	MINUS	EQV	X'97'	
0026-	NOKEY	EQV	X'FF'	
0027-	INS	EQV	X'84'	
0028-	DEL	EQV	X'83'	
0029-	CLR3	EQV	X'98'	

REL 4.1 790523 * PROCEDURE DIVISION * IDENT SCREEN * DATE 791015 * PAGE 0004 *

LINE LABEL OPCODE OPERANDS COMMENT C

0030- EQU X'80'
0031- EQU X'93'

* CREDIT TRANSLATOR REL 4.1 790523 * PROCEDURE DIVISION * IDENT SCREEN- * DATE 791015 *

LOC	OC	OPERANDS	LINE	LABEL	OPCODE	OPERANDS	COMMENT	C
0032-						EJECT		
0033-				*KEYTABLE USED WHEN CURSOR IS PLACED AT THE VERY				
0034-				*FIRST POSITION IN THE CURRENT FIELD				
0035-				*				
0036-		SPKTAB1		KTAB		BSP,CLEAR,REWRT,E01,CANCL,CANCL,CANCL,CANCL, FWD,BWD,HOME,LDOWN, LEFT,RIGHT,DOWN,UP,HCOPY, DUP,CFWD,ENT,NOKEY,TYPE		C
0037-								C
0038-								C
0039-								
0040-				*				
0041-				*KEYTABLE USED WHEN CURSOR IS PLACED AT ANY				
0042-				*POSITION BUT THE VERY FIRST IN THE CURRENT FIELD				
0043-				*				
0044-		SPKTAB2		KTAB		BSP,CLEAR,REWRT,E01,CANCL,CANCL,CANCL, FWD,BWD,HOME,LDOWN, LEFT,RIGHT,DOWN,UP,HCOPY, NOKEY,CBWD,ENT,MINUS,TYPE,TEST		C
0045-								C
0046-								C
0047-								
0048-				*				
0049-				*KEYTABLE USED IN EDIT MODE				
0050-				*				
0051-		SPKTAB3		KTAB		CFWD,CBWD,INS,DEL,CLEAR, REWRT,CLR3,E01,CANCL,CANCL,CANCL, FWD,BWD,HOME,LDOWN, LEFT,RIGHT,DOWN,UP,HCOPY, NOKEY,NOKEY,ENT,MINUS,TYPE,TEST		C
0052-								C
0053-								C
0054-								
0055-								C
0056-				*				
0057-				*FORMAT TABLE CONTAINING ERROR PRINTOUTS				
0058-				*				
0059-		SPFTBERR		FTABLE		ERFM01,ERFM02,ERFM03,ERFM04,ERFM05,ERFM06		
0060-				*				
0061-		ERFM01		FRMT				
0062-				FSL				
0063-				FTEXT		'TOO FEW INPUT CHARACTERS'		

0000 E0
0001 C3 18 ..

* CREDIT TRANSLATOR REL 4.1 790523 * PROCEDURE DIVISION * IDENT SCREEN- * DATE 791015

LOC	OC OPERANDS	LINE	LABEL	OPCODE	OPERANDS	COMMENT
		0064-		FMEND		
		0065-	*			
		0066-	ERFM02	FRMT		
		0067-		FSL		
0000	EO	0068-		FTEXT	'UNDEFINED ERROR'	
0001	C3 OF ..	0069-		FMEND		
		0070-	*			
		0071-	ERFM03	FRMT		
		0072-		FSL		
0000	EO	0073-		FTEXT	'I/O-ERROR'	
0001	C3 O9 ..	0074-		FMEND		
		0075-	*			
		0076-	ERFM04	FRMT		
		0077-		FSL		
0000	EO	0078-		FTEXT	'ILLEGAL END-OF-ITEM KEY'	
0001	C3 I7 ..	0079-		FMEND		
		0080-	*			
		0081-	ERFM05	FRMT		
		0082-		FSL		
0000	EO	0083-		FTEXT	'COMPULSORY FIELD NOT FILLED'	
0001	C3 I8 ..	0084-		FMEND		
		0085-	ERFM06	FRMT		
		0086-		FSL		
0000	EO	0087-		FTEXT	'ILLEGAL VALUE'	
0001	C3 O0 ..	0088-		FMEND		