APPENDIX F : SCREEN MANAGEMENT

This appendix gives a description of screen management as it may be called by an application for simplifying display handling. The module is written in CREDIT and supplied on the system disk, with module name SCREEN.

F.1 *Introduction*

This module can handle different types of applications which are used for e.g. data entry, inquiry, register updating, transactions etc. It uses a display, print device and keyboard for communication.

Entering the module is done with a Perform instruction followed by the procedure name. When control is passed to the module it will display the required screen layout as defined in a format list, which is prepared outside the package. Also input fields and corresponding data items are fully described in this format list.

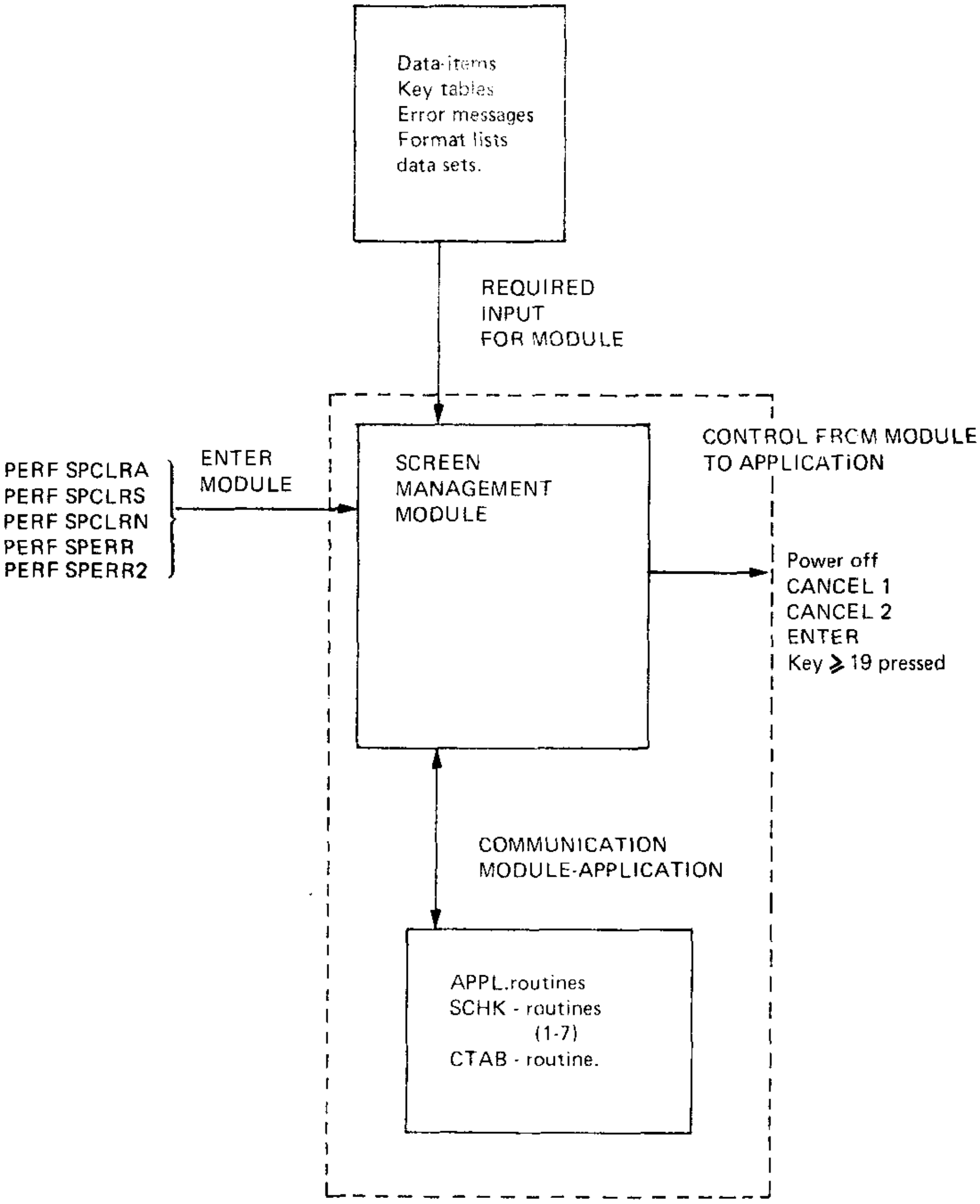
Tabulation functions, error messages, checks on input fields (APPL, MAXL etc.), function keys to terminate input, editing mode allowing cursor positioning, deletion or insertion of a character in a current input field and hardcopy facilities are all included in the module. When the transaction is terminated or e.g. a CANCEL key is pressed control will be given to the application.

Data items, keytables, datasets, error messages, format lists (for screen layout) and routines which are executed on a APPL,CTAB or SCHK option in the current input field, are defined outside the module, using standard names.

The VERIF option is not yet included in the module but the user has the possibility to change the module because all is written in the CREDIT language.

Only input fields of the type FKI are handled. FINP-fields have to be programmed outside the package. When the ENTER-key (end of transaction) is pressed, screen management will check if compulsary fields are filled. If not, the cursor will be positioned in such a field and data must be entered before the transaction can be completed.

It is advised to read also chapter 1.4.3.1 in this manual, which also describes the buffer handling.



Data-items
Key tables
Error messages
Format lists
data sets.
REQUIRED
INPUT
FOR MODULE
PERF SPCLRA
PERF SPCLRS
PERF SPCLRN
PERF SPERR
PERF SPERR2
ENTER
MODULE
SCREEN
MANAGEMENT
MODULE
CONTROL FRCM MODULE
TO APPLICATiON
Power off
CANCEL 1
CANCEL 2
ENTER
Key ≥ 19 pressed
COMMUNICATION
MODULE-APPLICATION
APPL.routines
SCHK - routines
(1-7)
CTAB - routine.

F.2 *Using Screen* [illegible]

Before the module [illegible] list, describing the screen layout and giving the options of input fields, must be [illegible] the Attach Format instruction (ATTFMT) has to be executed. When [illegible] the prompts must be displayed a boolean data item SPPROMPT must be set [illegible] it has to be zero and prompts will not be dis– played.

Entering the module is done by execution of one of the following perform instructions:

PERF SPCLRA
PERF SPCLRS
PERF SPCLRN
PERF SPERR

SPCLRA: When the boolean data item SPPROMPT=1 then following actions are performed on calling the module with this entry name.

- [illegible] the input fields.
- Display [illegible]
- Cursor [illegible] position of current FKI-field. If there is no current field the first input field is made current.
- Read in [illegible]
  The first character is checked with keytable SPKTAB1, following characters are checked with [illegible]2
- Check [illegible] otherwise display error message
- Check is performed on the input field if APPL, CTAB or SCHK option was specified.
- Input field is moved to corresponding data item
- Set boolean SPCHA[illegible] to indicate that a data item contents is changed/ updated
  (This data item has to be reset outside the module).
- Make next input field current, when End of [illegible] key is pressed.
- When ENTER key is pressed, control will be passed to the application. Binary [illegible] is set to value 3.

When SPPR[illegible] = 0

- Clear all [illegible] the screen and corresponding data items.
- Cursor to the [illegible] character position of current FKI-field. If there is no current [illegible] field is made current.

Following [illegible] as with SPPROMPT = 1

SPCLRS: When the [illegible] data item SPPROMPT = 1, then following actions are performed on [illegible] entry [illegible]

- Clear data [illegible] fields, except those which have option NCLR set [illegible]
- [illegible] which are not not cleared. (NCLR set).
- [illegible] position of current FKI-field. If there is no current [illegible] current.

Fo[illegible] RA with SPPROMPT = 1. This routine allows disp[illegible] data item, when in the corresponding FKI-input [illegible]

Whe[illegible]

- [illegible] data items, except when [illegible] N[illegible]
- [illegible] FKI-field. If there is no [illegible]

Following items the same as when SPPROMPT = 1

SPCLRN: When the boolean data item SPPROMPT = 1, then following actions are performed on calling the module with this name;

- No clearing of input fields and corresponding data items is performed.
- Display entire format and input fields.
- Cursor is located at first character position of current FKI-field. If there is no current field, the first input field is made current.

Following items are same as SPCLRA with SPPROMPT = 1. This routine makes it possible to recall a complete screen layout with its corresponding input field information.

When SPPROMPT = 0

- No clearing of input fields and corresponding data items is performed.
- Cursor is located at first character position of current FKI-field. If there is no current field, the first input field is made current.

Further the same as with SPPROMPT = 1

This routine makes it possible to enter the module and continue on a current field without any changing in the format on the screen.

SPERR: When an error is detected outside the module in an application routine other than SPCHKx, SPAPPL or SPTCHK, the user may call the module with this error routine name and pass the index of the error message in a binary data item SPBINW4. Screen management will handle this error as if it was detected in the package itself. A bell signal is sent and an error message will be displayed on the last line of the screen. (See also Error handling 5.1).

SPERR2: When an error is detected outside the module in an application routine other than SPCHKx, SPAPPL or SPTCHK, the user may call the module with this error routine name and pass the index of the error message in a binary data item SPBINW4. Screen management will examine and modify the current input field, and control is returned to the calling routine.
A bell signal is sent and an error message displayed on the bottom line of the screen (see also Error handling 5.1).

F.3 *Communication between Screen Management module and application*

Screen management checks the input field on options APPL (application), SCHK (check) and CTAB (conditional tabulation). Verification is not yet implemented in the package but has to be included by the user if required. When one of these options is specified, a subroutine outside the package is called with respectively the names SPAPPL, SPCHK1 through SPCHK7 or SPTCHK.

These routines are mentioned as externals (EXT) in the screen management module and require a corresponding entry (ENTRY) in the application module(s).

F.3.1 *SPAPPL. (application)*

After data is entered and the input is terminated screen management will check if the APPL option was specified for this input field. The interface between screen management and the routine SPAPPL is as follows:

Input to routine SPAPPL;

- SPINPUT, a string data containing the data input via the keyboard.
- SPBINW1, a binary data item containing the number of transferred characters on input.
- SPBINW3, a binary data item containing the APPL value as mentioned in the FKI input field definition. This value will be processed in the subroutine.
- SPBINW2, a binary data item containing the converted end-if-item key ind :x in the key table.

Output from SPAPPL;

The application has to return in the binary data item SPBINW3 a value 0, 1, 2 or 3 which will result in the following actions in screen management.

(SPBINW3) = 0 The contents of data item SPINPUT will be moved to the data item of the current input field and displayed on the screen when the "REWRT" option is defined for this input field. Screen management continues according to the converted end-of-item key index as present in SPBINW2.

(SPBINW3) = 1 The contents of data item SPINPUT will be moved to the data item of the current input field and is always displayed on the screen. Screen management continues according to the converted end-of-item key index as present in SPBINW2.

(SPBINW3) = 2 The contents of data item SPINPUT is not moved to the data item of the current input-field. Cursor is set at the begin of the current input field and input on this field can be performed.

(SPBINW3) = 3 Error detected. Binary data item SPBINW4 contains ar index for the error print out format. When the contents is zero, no error message is printed.
(See also error handling 5.1).

F.3.2 *SPTCHK (conditional tabulation)*

When a tabulation function key is pressed as forward, backward, upward etc., screen management searches for the required field.

When this input field is found and it has the CTAB option specified, screen management will call the routine SPTCHK, before setting the cursor in this field. The application routine SPTCHK has to check if this new current input field is wanted, or has to be continued on tabulating forward or backward to following/preceding fields. Once the tabulation direction, forward or backward, is set it will continue in this direction until the cursor is positioned. The cursor remains in its original position as long as CTAB option is specified for these input fields. The routine SPTCHK has to indicate in the binary data item SPBINW3 which action has to be taken by screen management.

Output from SPTCHK:

(SPBINW3)= 0 Tabulation is at correct input field. The cursor will be positioned at the first character position in this current field.

(SPBINW3) = 2 Tabulation must continue forward or backward.
The tabulation will continue backward, by screen management if the initial (present in SPBINW2) tabulation key pressed was:

- tabulation backward
- tabulation right most
- tabulation upwards

All other keys pressed will result in continuing forward tabulation.

F.3.3 *SPCHK1 – SPCHK7*

After data is entered and input is terminated, screen management will check if the SCHK option was specified for this input field. Depending on the value (1 to 7) defined after SCHK, the routine SPCHK1, SPCHK3, SPCHK4, SPCHK5, SPCHK6 or SPCHK7 will be called. The user can perform a check on this input field.

Input to routine SPCHKx:

– SPINPUT, a string data item containing the data input via the keyboard.

– SPBINW1, a binary data item containing the number of transferred characters on input.

–SPBINW2, a binary data item containing the converted end-of-item key index in the key table.

– SPBINW4, a binary data item containing the value defined in SCHK option.

Output from SPCHKx:

This interface is the same as for the SPAPPL routine, also in SPBINW3 a value 0, 1, 2 or 3 has to be returned, with same significance.

Note: When both options SCHK and APPL are specified for the current input field, first the SPSCHKx routine will be performed and then the SPAPPL routine.

F.4. *Key tables used by Screen Management*

F.4.1 *General*

Screen Management uses three keytables to terminate input via a keyboard. These key tables have a standard layout for the function keys. The corresponding key codes have to be defined by the user in a module SPLITT.

When a certain function key is not required by the user, it has to be set to the keycode value x'FF' (e.g. CANCEL key 2, CANC2 EQU x'FF').

The keys in keytable SPKTAB1, are function keys which are allowed to be entered as input field. The keys in key table SPKTAB2, are function keys which are allowed to be entered at any other position of the current input field. The keys in key table SPKTAB3, are function keys which are only effective when the package is switched to edit mode.

Reading of input characters is performed as keyboard input with echoing on the display. The function keys are not echoed on the display.

The first character read is checked if it is defined in SPKTAB1. When not present in the keytable and it is a numeric/alphanumeric character, the remaining part of the field on the display will be cleared, the input character is echoed on the display and the remaining character positions are displayed as a number of dots, corresponding to the maximum length of the current input field.

Remaining characters are input and checked with key table SPKTAB2.

If an illegal key code is entered, the acoustic alarm will sound and reading is continued.

After completion of the input, the remaining dots in the field are cleared and a check is performed if options as APPL and SCHK are defined for this input field, before continuing according to the pressed function key.

F.4.2 *Function keys in SPKTAB1*

This key table is used when the cursor is positioned at the first character position of the current input-field. It is also used after an error message is displayed, then only key codes at position 2, 3, 5 and 6 are relevant.

Position number in SPKTAB1 Significance

1 BACKSPACE. When this key is pressed, the cursor is moved one step to the left and the corresponding position on the display is replaced by a dot. If backspace is performed to the very first position of the current input field, the old contents of the data item belonging to this input field will be displayed.

2 CLEAR. The input field and corresponding data item are cleared, when the MAXL value is greater than zero. The cursor remains in the first position of the current input field.

3 RESET. The input field is displayed with the old contents of the to this field belonging data item. The cursor remains at the first position of the current input field.

Position number in SPKTAB1	Significance
4	EOI. Common end of item key. Tabulation forward to the next input field is performed.
5	CANCEL1. A return to the application is performed with in the binary data item SPBINW2, the value 1. No checks are performed on the input field.
6	CANCEL2. Same as CANCEL1 but in SPBINW2, the value 2 is returned.
7	TFWD. Tabulation forward to the next input field. No action is taken if this input field is not present. If an empty compulsory field is found in an earlier defined input field, the cursor is positioned at the begin of this compulsory field. The compulsory field becomes current.
8	TBWD. Tabulation backward to the previous input field. No action is taken if this input field is not present. If an empty compulsory field is found in an earlier defined input field, the cursor is positioned at the begin of this compulsory field. The compulsory field becomes current.
9	THOME. Tabulation to the first input field of this format list.
10	TLDOWN. Tabulation to the first input field on the next line. No action is taken if this input field is not present. If an empty compulsory field is found in an earlier defined input field, the cursor is positioned at the begin of this compulsory field. The compulsory field becomes current.
11	TLEFT. Tabulation to the most left input field on the current line. If an empty compulsory field is found in an earlier defined input field, the cursor is positioned at the begin of this compulsory field. The compulsory field becomes current.
12	TRIGHT. Tabulation to the most right input field on the current line. If an empty compulsory field is found in an earlier defined input field, the cursor is positioned at the begin of the compulsory field. The compulsory field becomes current.
13	TDOWN. Tabulation to the input field on the next line, with a starting column nearest to the starting column of the current input field. When two nearest columns are found, tabulation will be to the left input field. When no input field is found, following lines will be searched. No action is taken if the input field is not present. If an empty compulsory field is found in an earlier defined input field, the cursor is positioned at the begin of the compulsory field. The compulsory field becomes current.

Position number in SPKTAB1	Significance
14	TUP. Tabulation to the input field on the preceding line, with a starting column nearest to the starting column of the current input field. When two nearest columns are found, tabulation will be to the left input field. When no input field is found, preceding lines will be searched. No action is taken if the input field is not present. If an empty compulsory field is found in an earlier defined input field, the cursor is positioned at the begin of the compulsory field. The compulsory field becomes current.
15	COPY. A hardcopy of the entire screen is made on the print device.
16	DUPL. Move of the contents of the duplication data item, as defined by the DUPL option in the current input field, to the field SPINPUT and display. When no DUPL option defined, an error message will be displayed.
17	EDIT. Set to edit mode. (See SPKTAB3 description 4.4).
18	ENTER. Ends the handling of this complete format if all compulsory fields are filled and returns to the application. When not all compulsory fields are filled, an error message is displayed and the cursor is positioned on the first not filled compulsory field, and this field becomes current.
19	Application functions keys. When a function key with a position number 19 or greater in the key table is pressed, a return will be performed to the application. In data item SPBINW2 is returned the value 4 for a key in position 19. For a key in position 20 the value 5 is returned etc.

F.4.3 *Function keys in SPKTAB2*

The key table has the same layout as key table SPKTAB1 and is used when the cursor is positioned anywhere in this input field except on the first position. When a function key is required only in SPKTAB1 and not in SPKTAB2, then the corresponding key position in SPKTAB2, has to get a key code X'FF'.

Keys from position 7 and up will first be handled as an end-of-item key and after the input data is found correct a check on SCHK and APPL options is done. Then the function required is executed.

For position numbers in key table and significance. See SPKTAB1.

F.4.4 *Functions keys in SPKTAB3, edit mode*

The edit mode is entered by pressing the function key for edit mode, which key code is defined in key table SPKTAB1 and SPKTAB2. Furthermore the MAXL option must be greater than zero.

A change to edit mode may be done during data input for the current field e.g. three character positions after a wrong character was entered and this has to be changed before continuing.

In this mode more functions are available to update on character level in the current input field as e.g. non destructive space, delete a character, insert a character etc.

After input is terminated and checked the data item belonging to this current input field will be updated.

If an illegal character is entered the acoustic alarm will sound and editing continues. The character at the cursor position will be overwritten by numeric or alphanumeric keys.

Position number in SPKTAB3	Significance
1	→NON-destructive space. Moves the cursor one step to the right. Accoustic alarm sounds if trying to exceed the effective item limits.
2	← NON-destructive backspace. Moves the cursor one step to the left. No action if most left position is reached.
3	INS. Insert a character at current cursor position. The character under the cursor and the characters right of the cursor in the current field are shifted one step to the right with truncation. One space character will be inserted at the cursor position and the cursor is not moved.
4	DEL. Delete character at current cursor position. The other characters to the right, in the current field, are shifted one step to the left.
5	CLEAR1. The input field and corresponding data item are cleared. Cursor is positioned at the first character position of the field. Out of edit mode.
6	CLEAR2. The input field is displayed with the old contents of the data item belonging to this field. Out of edit mode.
7	CLEAR3. Clear field from current cursor position up to the most right position. Out of edit mode.
8	The same keys as for SPKTAB1 and SPKTAB2 as mentioned from position 4 and up. Out of edit mode.

F.5. *Error handling*

F.5.1 *Errors detected in the Screen Management module*

The module checks:

- the number of characters entered is less than specified in MINL.
- an I/Ø error during input.
- Illegal end-of-item key pressed, e.g. duplication but in the input field the DUPL option is not specified.
- Compulsory field(s) not filled when the ENTER key is pressed.

Before the error message is displayed on the last line on the screen, the acoustic alarm sounds. The error messages are to be defined by the user in a table SPFTBERR and an index for these messages is present in binary data item SPBINW4.
When this index is zero no error message is displayed and reading is resumed in the current input field. After the error message is displayed the cursor will be placed at the current input field. Input is done via the keyboard with using key table SPKTAB1. Only the functions CLEAR1, CLEAR2, CANCEL1, CANCEL2, and EDIT are relevant for this input, and clearing of the current input field is done if the input length as present in SPBINW1 is not zero.

If SPBINW1 is containing zero no clearing of the current input field is done. The error message is cleared and there will be continued according to keytable SPKTAB1.

Contents of SPBINW4 (error message index)	Significance
0	No error message displayed.
1	Number of characters input is less than stated in the MINL option.
2	Not used.
3	I/O-error (e.g. time out, throughput error etc.).
4	Illegal end-of-item key pressed. (e.g. duplication wanted but option not specified in the input field).
5	Compulsory field not filled after pressing the ENTER key.
6	This index value and up may be used by the routines SPAPPL and checkroutines SPCHK1 - SPCHK7.

F.5.2. *Errors detected outside the Screen Management module*

When an error is detected outside the package, it is possible to enter the module in the error handling section by calling the routine "SPERR" (PERF SPERR) or "SPERR2" (PERF SPERR2).

The binary data item SPBINW4 should contain the error message index and SPBINW1 must be unequal to zero if clearing is wanted, or zero if clearing is not wanted.

When the package is entered via SPERR, error handling is the same as described in 5.1. By pressing the CLEAR1, CLEAR2 key or the CANCEL1, CANCEL2 key, the user may keep control of the screen management module or leave the module.

When the package is entered via SPERR2, only the current input field can be modified. All keys, except the EDIT key, are considered as ENTER keys, though a hardcopy is obtained when the COPY key is pressed.

The routines SPCHKx and SPAPPL are assumed to be part of the screen management module. (See 3.1 and 3.3).

F.6. *Control from package to application*

Control is passed to the application when:

- power off is detected
- CANCEL1 key is pressed.
- CANCEL2 key is pressed.
- ENTER key is pressed.
- An application defined, function key is pressed. (≥19)

This information is passed to the application in binary data item SPBINW2. Also the boolean data-item SPCHANGE is set to indicate that at least one of the data items belonging to the input field is changed. It should be reset outside the module.

In the application has to be decided what to do with the result in SPBINW2.

Contents of SPBINW2	Significance
0	Power off detected. Return to application. Screen may be cleared.
1	CANCEL1 key pressed. Return to application. No check on input performed. Current input field (FKI-type) and corresponding data item is cleared.
2	CANCEL2 key pressed. Return to application. No check on input performed. Current input field (FKI-type) and corresponding data item is cleared.
3	ENTER key pressed. Return to application. No empty compulsory field(s) are found.
4 and up	Application defined function key is pressed. For SPKTAB1, a key at position 19 and up. For SPKTAB2, a key at position 19 and up. For SPKTAB3, a key at position 23 and up. Key position 19 (SPKTAB1, SPKTAB2) or 23 (SPKTAB3), will be converted to index value 4 in SPBINW2. Position number 20, respectively 24 will be converted to index value 5 etc.

F.7. *Required definitions outside Screen Management module*

Screen Management uses data items which are defined in the data division of the application. In the module is referred to the data division by means of a DDUM SPDDIV directive. When the user wants to use for his data division an other name, then the name SPDDIV in the module has to be updated. The equates for the key tables and the format lists describing the error messages, are expected to be defined in a module SPLITT. The package uses the instruction INCLUDE SPLITT, LIST. To find the names used in key tables SPKTAB1, SPKTAB2 and SPKTAB3, the module SCREEN must be translated. Error messages are expected to be defined in the format table SPFTBERR. The format list of each error message must start with the FSL directive.

Format list describing the screen layout are defined in the application and do not have to be included in module SPLITT.

Data sets to be defined in the data division and used by screen management.

Name — Significance

SPDSPRT — Data set with fixed buffer for hard copy device.
The buffer size must have at least the same size as the one used for the display. (Buffers may be shared).

SPDSSCRN — Data set with fixed buffer for the display.
Buffer size must be large enough to hold the maximum number of characters (e.g. one line) inclusive the control characters. (Buffers may be shared).

SPDSDYKB — Data set for the keyboard.

A format control I/O declaration (FMTCTL) has to be defined for input dataset and output dataset, which are used in the format control I/O instructions. (e.g. DYKI, DISPLAY).

Data items to be defined in the data division and used by screen management.

Name — Significance

SPBINW1 — A binary data item used to contain the number of characters transferred during input or used as work item.

SPBINW2 — A binary data item used to contain the (converted) end-of-item key.

SPBINW3 — A binary data item used to contain the APPL-value.

SPBINW4 — A binary data item used to contain the SCHK value (1 up to 7) or error code index.

SPCHANGE — A boolean data item which is set by screen management to indicate that a data item belonging to a current input field is changed. Resetting has to be done outside the module.

SPPROMPT — A boolean data item used to indicate that the prompt texts should be displayed. If set (TRUE), the entire format is displayed including the prompts. If reset (FALSE), only the data items belonging to the input fields are displayed and not the prompts. SPPROMPT has to be set or reset outside the module.

SPINPUT — A string data item used as keyboard input buffer. The size must be large enough to contain the maximum input length, inclusive the end-of-item key.

SPERCALL — A boolean data item used by screen management to indicate whether the SPERR or SPERR2 entry is used. This data item is set to one when SPERR2 is called.

SPSTRGW1 — A string data item (≥ 2 characters) used by the error routine.

F.8. *Example of a coded format*

The following picture is defined:

```
   A C C N T N O   :  input field 1
         N A M E   :  input field 2
   A D D R E S S   :  input field 3
 C I T Y C O D E   :  input field 4   C I T Y   :   input field 5
```

field 1 : – numeric
- compulsary field
- exactly 10 digits long
- no end-of-item required
- formatting after input wanted
- CDV-10 check should be performed

field 2 : – alphanumeric
- 2 - 35 characters long

field 3 : – alphanumeric
- 2 - 30 characters long
- duplication from item ITEM10 in an other picture is allowed

field 4 : – numeric
- exactly 5 digits long
- no end-of-item key required
- formatting after input wanted
- after entering the field, the input is used as key in a file for corresponding CITY.
  When found, the CITY is displayed in the next field (field 5). If not found the cursor will be placed at the next field.

field 5 : – alphanumeric
- 2 - 20 characters long

The corresponding format can look as follows:

```
PICT1     FRMT
          FSL
          FCOPY             =   ' A C C N T N O : '
          FKI               11, ME, MINL=10, MAXL=10, NEOI, REWRT,
                            SCHK=1
          FMEL              'XXXXXXE-XXXX', ITEM1
          FNL
          FCOPY             = '    N A M   E : '
          FKI               11, ALPHA, MINL=2, MAXL=35
          FCOPY             ITEM2
          FNL
          FCOPY             = ' A D D R E S S : '
          FKI               11, ALPHA, MINL=2, MAXL=30, DUPL=ITEM10
          FCOPY             ITEM3
          FNL
          FCOPY             = ' C I T Y C O D E : '
```

```
FKI         11, MINL=5, MAXL=5, NEOI, REWRT, APPL=1
FMEL        'XXXBXX', ITEM4
FCOPY       =' CITY: '
FKI         23, ALPHA, MINL=2, MAXL=20
FCOPY       ITEM5
FMEND
```