1 Preface

This manual describes all the macros and exits available for working with the Spool & Print Services for BS2000/OSD.

1.1 Brief product description of the Spool & Print services

The Spool & Print Services for BS2000/OSD consist of various subsystems and utilities.

Spool & Print subsystems

SPOOL

The local SPOOL (Simultaneous Peripheral Operation On-Line) is a BS2000 subsystem. It controls asynchronous output to printers and magnetic tapes. SPOOL is the fundamental component for the other products described below, which cannot run without it. For further details, see the "SPOOL (BS2000/OSD)" manual.

RSO

Remote Spool Output controls output to decentralized printers (RSO printers) which are connected to remote systems. RSO allows only one point-to-point transfer of the data. For further details, see the "RSO (BS2000/OSD)" manual.

Dprint

Distributed **Print** Services also controls output to printers which are connected to remote systems, but it uses the local SPOOL of the target system and retains control over the print output. For further details, see the "Distributed Print Services (BS2000/OSD)" manual.

SPS

SPOOLAPA Printing System can be attached to the SPOOL system as a subsystem and performs the task of a printer driver for APA printers. For further details, see the "PRISMAproduction/BS2000 - SPS" manual.

IDOM

Integrated **D**ocument and **O**utput **M**anagement is a subsystem in Spool & Print Services on BS2000/OSD. It extends the Spool & Print Services by allowing you to use document management facilities. For further details, see the "IDOM (BS2000/OSD)" manual.

Additional utility routines for administration

PRM

Print Resource Management creates and manages SPOOL print resources. PRM is mandatory for SPOOL Version 3.0 and higher, as all actions concerning the print resources are carried out via PRM. PRM is described in detail in the manual of the same name.

SNS

The **S**POOL **N**otification **S**ervice provides a tool to send and to manage notifications in the frame of the BS2000.

SPSERVE

SPSERVE allows systems support - and to a limited extent also RSO device administrators and nonprivileged users - to enter, modify, delete or display information in a SPOOL parameter file. SPSERVE is described in detail in the manual of the same name.

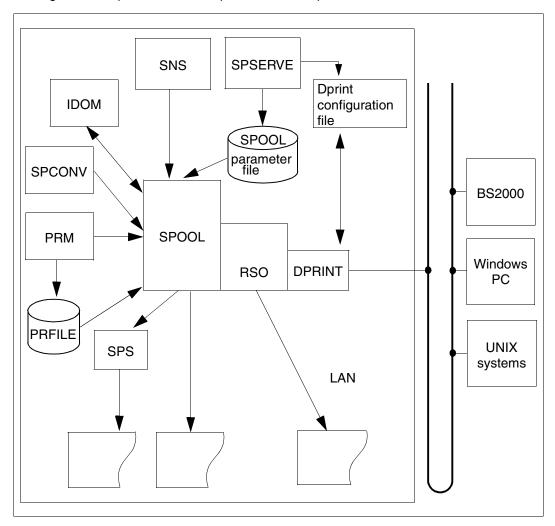
SPCONV

The SPOOL converter implements the filter technology for printing in distributed, heterogeneous environments with BS2000, UNIX systems and PC systems, i.e. it uses filters to change documents and print resources into formats which can be printed by the various printers. For further details on SPCONV, see the manual with the same title.

Spool & Print Explorer

The Spool&Print Explorer is the graphical user interface (GUI) for Spool & Print administration on BS2000 systems. The Spool&Print Explorer is a central, graphically-oriented administration program on a PC for Spool & Print administrators and/or end users The Spool & Print Explorer is described in detail in the "SPOOL (BS2000/OSD)" manual.

The figure below presents the components of the Spool & Print Services.



1.2 Target group of the manual

This manual is intended for

- BS2000 systems support staff (Exits)
- users who want to print files or write files to external media by using the programming interfaces.
- SPOOL administrators (as of BS2000/OSD V2.0)
- RSO device administrators
- nonprivileged users of the Spool & Print services who want to print their data to BS2000 high speed printers.

1.3 Summary of contents

The descriptions in this manual are sorted alphabetically within functional groups. A short description of the contents is included at the start of each functional group.

The chapter "Macros for outputting files"

describes the macros used to output data and assign print jobs.

The chapter "Macros for outputting information"

describes the macros used to query the contents of the SPOOL parameter file.

The chapter "Macros for managing print jobs"

describes the macros used for managing print jobs.

The chapter "Macros for managing printers"

describes the macros used for managing printers.

The chapter "Macros for virtual printers"

describes application models for virtual printers and the macros with which virtual printers can be used.

The chapter "SPOOL exits"

describes the exits of the Spool & Print Services.

The appendix contains the following sections:

- "Macro syntax" with a description of the macro syntax used in this manual.
- "Functional principle of the exit mechanism" describes the function of the exits.
- "Trailer page for the local SPOOL" with a printout of the trailer page of a local SPOOL print job.
- "Spoolout accounting record (SPLO)" with a description of the SPOOL accounting record.

"Application rules for C" with a description of the application rules for C.

A list of related publications and an index can be found at the back of this manual.

1.4 Changes to the macros and exits since the last edition

General changes and additions available from Distributed Print Services V1.1B, RSO V3.4., SNS V1.0B, SPOOL V4.6, SPS V3.8, and SPSERVE V2.8 have been included in the manual. The following changes/additions have been made in the program interface:

- In the macros SASDSPO, STRSPPR, and SUPSPPR a *CLIST value has been added for some *_TYP parameters.
- The correct syntax description of the SPVDRCV and SPVDRET macro is given.
- A new return code has been added to the SHOPJST macro.
- The new macro SNPPRNT describes the parameter list for the support of the Notification Service.
- The PRNTDOC macro has been extended. The operand PRNAME has been modified in order to be able to specify an URL, and the operand NOTIFPAR has been added in order to allow the notification access by program.
- The macros PRNTRSO, MODJAT and MODJRSO have been extended in order to be able to specify a URL.
- The new macro SSVBARC for the barcode support has been added.
- The following operands of the PRNTDOC macro and MODPJAT macro are now also valid for APA printers:
 INTRAY, LEFTMAR, LEFTOFF, OUTTRAY, ROT, TOPOFF, TWOSIDE
- Two new operands have been added to the PRNTSPS and MPJASPS macro: FRONTOVL and BACKOVL
- The new macro OLTPRQX macro offers UTM applications several options for defining print parameters when printing messages.
- Accounting:
 To assure a synchronisation between SPOOL V4.4 and SPS 03.08.xy, the APA page printer extension of the Spoolout accounting record has been modified. All other fields
- As of BS2000/OSD V4.0, diskette devices are no longer supported, therefore the macro PNCH is no longer described in this manual.
- SCSIPCL
 The product SCSIPCL is no longer supported as of SPOOL V4.3.

of the SPOOL accounting are not modified.

1.5 Notational conventions



This symbol indicates that the following indented paragraph contains essential information.

Note

The word "Note" indicates that the following indented paragraph contains important information.

"Reference"

Cross-references to chapters, sections or other manuals are enclosed in double quotes.

[]

Square brackets within syntax examples: the characters within the brackets may be omitted.

Boldface font

Where syntax examples are explained, the lines actually explained are shown in boldface font.

The rules described in the relevant chapters of the reference section apply for the remaining syntax examples.

SYNTAX/example

Syntax examples and example inputs/outputs are highlighted with other fonts. Syntax examples are also enclosed in a frame.

1.6 README file

Information on any functional modifications and additions to the current product version described in this manual can be found in the product-specific README file. You will find the README file on your BS2000 system under the file name

SYSRME.product.version.E

The user ID under which the README file is cataloged can be obtained from your systems support. You can view the README file using the SHOW-FILE command or an editor, and print it out on a standard printer using the following command:

/PRINT-DOCUMENT FROM-FILE=filename, - /DOCUMENT-FORMAT=*TEXT(LINE-SPACING=*BY-EBCDIC-CONTROL)

2 Macros for outputting files

The following macros are described in this section:

"OLTPRQX - define UTM print parameters" on page 8

With the OLTPRQX macro offers UTM applications several options for defining print parameters when printing messages.

"PRNTDOC (for SPOOL, SPS, RSO and Dprint) - output file" on page 27

The PRNTDOC macro outputs files to printers and other peripheral devices. The devices concerned do not have to be connected to the local computer as PRNTDOC also cooperates with RSO and Dprint if the parameters are set accordingly.

"PRNTRSO - define RSO parameters for PRNTDOC macro" on page 166

If PRNTDOC is to output files on remote systems via RSO, you have to set the relevant parameters appropriately with PRNTRSO.

"PRNTDPC - define Dprint parameters for PRNTDOC macro" on page 172

If PRNTDOC is to output files on remote systems via Dprint, you have to set the relevant parameters appropriately with PRNTDPC.

"PRNTSPS - define SPS-APA parameters for PRNTDOC macro" on page 177

If PRNTDOC is to output files on HP printers, you have to set the relevant parameters appropriately with PRNTSPS.

"SNPPRNT - define notification parameters for PRNTDOC" on page 182

The macro SNPPRNT describes the parameter list for the support of the Notification Service.

"SSVBARC - translate data for barcodes" on page 187

The macro SSVBARC provides a standard way to call the barcode program (LLM) that translates a character into its corresponding barcode representation.

OLTPRQX - define UTM print parameters

User group: UTM applications

programming languages: Assembler, CPP, C, Cobol

Macro type: M, S

The OLTPRQX macro offers UTM applications the following options, when printing messages:

- selecting the orientation: portrait, landscape, inverse portrait and inverse landscape
- selecting the paper source by paper type (A4, A3 etc.) or by the input tray number.
- selecting the output tray
- selecting the print presentation: offset of the logical page on the medium, simplex, duplex or tumble-duplex printing, number of copies of each page.

Format

Operation	Operands
OLTPRQX	,INTRAY=*IGNORE / <integer 199=""> / *BY-FORMAT</integer>
	,INTRAYF=*C5 / *DL / *COMMERCIAL_10 / *MONARCH /
	,LOFSIGN= <u>*LOFSPOS</u> / *LOFSNEG
	,LOFVAL=*IGNORE / <integer 1255=""></integer>
	,OUTTRAY=*IGNORE / <integer 199=""> / *SORTER</integer>
	,PAGECOP= <u>*IGNORE</u> / <c_string 33=""></c_string>
	,ROT= <u>*NO</u> /*ROT_0 / *ROT_90 / *ROT_180 / *ROT_270
	,SRTMODE= <u>*NO</u> / *NO_SORTER / *GROUP / *COLLATE / *STACKER / *AUTOMATIC
	,TOFSIGN= <u>*TOPSPOS</u> / *TOPSNEG
	,TOFVAL=*IGNORE / <integer 1255=""></integer>
	,TWOSIDE= <u>*NO</u> / *SIMPLEX / *DUPLEX / *TUMBL

Description of the operands

INTRAY=<u>*IGNORE</u> / <integer 1..99> / *BY-FORMAT

Paper input tray for the RSO printers 2030-PCL, 4011, 4812, 4813, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL, 4830-PCL, 4850-PCL, 9000-PCL, 9000-EPLQ, 9000-EPSQ, 9004, 9011, 9012, 9013, 9014, 9015, 9021, 9022, 9022-200, 9025, 9026-PCL, 9026-RENO, 9097 and DJET.

INTRAY=*IGNORE

The printer default is used.

INTRAY=<integer 1..99>

The number of the input tray from which the paper for both the file itself and the header and trailer pages is taken. The value entered is not verified by RSO. Values between 1 and 99 are possible. Further information on the possible values that can be specified for individual printers can be found in the description of the PRINT-DOCUMENT command in the "Spool & Print - Commands (BS2000/OSD)" manual.

INTRAY=*BY-FORMAT

The paper input tray is defined by means of a keyword for the format in the INTRAYF operand.

INTRAYF=*C5 / *DL / *COMMERCIAL_10 / *MONARCH / *DOUBLE-LETTER / *LETTER / *LEGAL / *EXEC / *INVOICE / *FOLIO / *B5 / *B4 / *A5 / *A4 / *A3 / *MANUAL

Selects the paper input tray by means of a keyword for the format. Further information on the possible values that can be specified for individual printers can be found in the description of the PRINT-DOCUMENT command in the "Spool & Print - Commands (BS2000/OSD)" manual.

INTRAYF= *MANUAL

When the *MANUAL operand is specified, the paper is fed manually and you must feed a new sheet of paper whenever a new page is printed or when the printer demands it. The following printers support manual paper feed: 9014, 9021, 9022, 9022-200, 9026-RENO, 9026-PCL, 4812, DJET, 4814-PCL, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL and 9000-PCL.

LOFSIGN=*LOFSPOS / *LOFSNEG

Specifies the offset sign: positive (*LOFSPOS) or negative (*LOFSNEG). If the value is positive, the first line is printed to the right; if the value is negative, the first line is printed to the left.

LOFVAL=*IGNORE / <integer 1..255>

Defines in millimeters the offset between the left margin of the print page and the left margin of the physical page. First, the print page is positioned on the paper, and then the text is rotated within the print page and positioned. This means that the text orientation within the print page is not considered, if the print page is displaced on the paper. This operand may only be specified for the following printers: 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL, 4830-PCL, 4850-PCL, 9000-PCL, 9021, 9022-200, 9026-PCL and 2030-PCL.

LOFVAL=*IGNORE

The print page is not displaced horizontally on the paper.

LOFVAL=<integer 1..255>

The print page is displaced horizontally on the paper according to the value specified.

OUTTRAY=*IGNORE / <integer 1..99> / *SORTER

Specifies the paper output tray for the RSO printers 2030-PCL, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL, 4830-PCL, 4850-PCL, 9000-PCL, 9014, 9015, 9026-PCL and 9026-RENO.

OUTTRAY=*IGNORE

The printer default is used.

OUTTRAY=<integer 1..99>

Specifies the paper output tray that is to be used for the current print job. Further information on the possible values that can be specified for individual printers can be found in the description of the PRINT-DOCUMENT command in the "Spool & Print - Commands (BS2000/OSD)" manual.

OUTTRAY=*SORTER

The sort mechanism specified in the SRTMODE operand is to be used for the current print job. The sort mechanism is valid for the following printers: 9026-RENO, 4818-PCL, 4822-PCL and 4825-PCL, and can be used for up to 20 output trays. It cannot be controlled by the defaults defined in the device entry.

PAGECOP=*IGNORE / <c_string 3..3>

Specifies how often each individual page is to be repeated. Only for the following RSO printers: 2030-PCL, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL, 4830-PCL, 4850-PCL, 9000-PCL, 9021, 9022, 9022-200, 9025, 9026-PCL, 9026-RENO and DJET.

PAGECOP=*IGNORE

The printer default is used.

PAGECOP=<c_string 3..3>

Number of page copies. Possible values '000' .. '255'.

PAGECOP=255 has the same effect as PAGECOP=254: one original and 254 copies are printed.

ROT=*NO /*ROT 0 / *ROT 90 / *ROT 180 / *ROT 270

Specifies whether the pages of a print request are to be rotated, and if so, by how many degrees.

ROT=*NO

No page rotation is selected. Further information can be found in the description of the PRINT-DOCUMENT command in the "Spool & Print - Commands (BS2000/OSD)" manual.

ROT=*ROT 0 / *ROT 90 / *ROT 180 / *ROT 270

Each print page is rotated 0° / 90° / 180° / 270° (clockwise) and printed. Rotation 0/90 is supported by printer types 2030-PCL, 4812, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL, 4830-PCL, 4850-PCL, 9000-PCL, 9021, 9022-200, 9025, 9026-PCL, 9026-RENO and DJET.

Rotation 180/270 is supported by printer types 2030-PCL, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL, 4830-PCL, 4850-PCL, 9000-PCL, 9026-PCL and 9026-RENO.

SRTMODE=*NO / *NO SORTER / *GROUP / *COLLATE / *STACKER / *AUTOMATIC

The sort mechanism is to be used for the current print job. The sort mechanism is valid for the following printers: 9026-RENO, 4818-PCL, 4822-PCL and 4825-PCL and can be used for up to 20 output trays. It cannot be controlled by the defaults defined in the device entry. Relevant examples can be found in the description of the PRINT-DOCUMENT command in the "Spool & Print - Commands (BS2000/OSD)" manual.

SRTMODE=*NO

The printer default is used.

SRTMODE=*NO SORTER

All document pages are deposited in the sort trays from bottom to top. Printer 9026-RENO is an exception: The print pages are sorted into a tray according to

easiest accessibility.

SRTMODE=*GROUP

If SRTMODE=*GROUP together with PAGECOP is specified, each copy of a particular document page is sorted to a separate tray. Each subsequent document page is sorted on top of the previous page. At the end of the print job, each tray contains a complete copy of the printed document.

SRTMODE=*COLLATE

If SRTMODE=*COLLATE together with PAGECOP is specified, all copies of a particular document page are sorted to one tray. The copies of the next document page are sorted to the next tray. The sort trays are filled from bottom to top.

SRTMODE=*STACKER

Not applicable to printer 9026.

All printed pages are sorted to a mass output tray in the sort mechanism (to a maximum of 500 pages). This mode is available for printing a single copy of a very long document.

SRTMODE=*AUTOMATIC

Only applicable to printer 9026.

The method of sorting is automatically selected, according to the number of copies per page specified in PAGE-COPIES and the number of sort trays available. The printed pages can either be sorted or not. The printed pages are deposited unsorted as for *NO if the number of copies specified in PAGE-COPIES is exactly the same as the number of sort trays. The printed pages are sorted according to document as for *GROUP if the number of copies specified in PAGE-COPIES is smaller than the number of sort trays. The printed pages are sorted according to pages as for *COLLATE if the number of copies specified in PAGE-COPIES is larger than the number of sort trays.

TOFSIGN=*TOPSPOS / *TOPSNEG

Specifies the offset sign: positive (*TOPSPOS) or negative (*TOPSNEG). If the value is positive, the first line is printed lower; if the value is negative, the first line is printed higher.

TOFVAL=*IGNORE / <integer 1..255>

Defines in millimeters the offset between the top of the print page and the top of the physical sheet. First, the print page is positioned on the paper, and then the text is rotated within the print page and positioned. This means that the text orientation within the print page is not considered if the print page is displaced on the paper.

This operand is only valid for the following printers: 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL, 4830-PCL, 4850-PCL, 9000-PCL, 9021, 9022-200, 9026-PCL and 2030-PCL.

TOFVAL=*IGNORE

The print page is not displaced vertically on the paper.

TOFVAL=<integer 1..255>

The print page is displaced vertically on the paper according to the value specified.

TWOSIDE=*NO / *SIMPLEX / *DUPLEX / *TUMBLE

Defines for RSO printers whether printing is to be single-sided or double-sided.

TWOSIDE=*NO

The printer default is used.

TWOSIDE=*SIMPLEX

The print job is printed in simplex mode, i.e. single-sided.

TWOSIDE=*DUPLEX

The print job is printed in duplex mode, i.e. double-sided. The pages are bound along a vertical edge. Duplex mode is allowed for the following printers: 2030-PCL, 9026-PCL, 9026-RENO, 4822-PCL, 4825-PCL, 4824-PCL, 4830-PCL, 4850-PCL and 9000-PCL.

TWOSIDE=*TUMBLE

The document is printed in duplex mode, i.e. double-sided, and the pages are not turned from left to right, but from top to bottom. The tumble-duplex mode is allowed for the following printers: 2030-PCL, 9026-PCL, 9026-RENO, 4822-PCL, 4825-PCL, 4824-PCL, 4830-PCL, 4850-PCL and 9000-PCL.

Structure layouts

ASSEMBLER specifications

```
parameter area
OLTXHDR
        FHDR MF=(C,OLTX),EQUATES=NO
                                          header layout
OLTXEXT
          DS
                0XL28
                                          extension part layout
OLTXROT
         DS
               FI1
                                          rotation code
   description of the rotation codes
         ATF
              ('&EQUATES' NE 'YES').Q01001
OLTXNROT FOU
                                          no rotation given
OLTXROTO EOU
              241
                                          no rotation
OLTXR90
          EQU
               242
                                          rotation 90
OLTXR180 FOU
              243
                                          rotation 180
OLTXR270 EOU
               244
                                          rotation 270
.Q01001 ANOP
OLTXIT#
         DS
               0XL1
                                          intray
OLTXINT
         DS
                Χ
                                          input tray number
         ORG
              OLTXIT#
OLTXITFS DS
               FL1
                                          input tray format
   description of the input tray selection mode
              ('&EQUATES' NE 'YES').Q01002
        AIF
                                          selection via a format name
OLTXITBF EOU
               255
OLIXITIG FOU
                                          no input tray given
.Q01002 ANOP
         ORG
               OLTXIT#
OLTXITRA DS
               FL1
                                          input tray code
   description of given input tray
               ('&EQUATES' NE 'YES').Q01003
         ATF
OLTXNITR EQU
               0
                                          no input tray given
OLTXITM# EQU
               99
                                          max num tray #
OLTXITC5 EOU
               239
                                          x'EF' C5
OLTXITDL EQU
              240
                                          x'F0' DL
OLTXIC10 EQU
              241
                                          x'F1' COM10
OLTXIMRC EOU
               242
                                          x'F2' monarch
OLTXIDBL
          FOU
               243
                                          x'F3' double letter
OLTXILET EQU
               244
                                          x'F4' letter
```

```
OLTXILEG EOU
                245
                                           x'F5' legal
OLTXIXEC
         EQU
                246
                                           x'F6' exec
OLTXIVCE
          EQU
                247
                                           x'F7' invoice
OLTXIFOL EOU
                248
                                           x'F8' folio
OLTXITB5 EOU
                249
                                           x'F9' B5
OLTXITB4
         EQU
                250
                                           x'FA' B4
OLTXITA5
         EQU
                251
                                           x'FB' A5
OLTXITA4
          EQU
                252
                                           x'FC' A4
OLTXITA3
         EQU
                253
                                           x'FD' A3
                                           x'FE' manual
OLTXIMAN EQU
                254
.001003 ANOP
         ORG
               OLTXIT#+1*
OLTXOT#
          DS
                0XL1
                                           outtray
          DS
OLTXOUTT
                Χ
                                           output tray number
         ORG
               OLTXOT#
OLTXOTRA DS
               FI1
                                           output tray code
    description of the output tray code
               ('&EQUATES' NE 'YES').Q01004
         AIF
OLTXNOTR
         FOU
                                           no output tray
OLTXOTM#
          EQU
                99
                                           output tray max #
OLTXOTSR EQU
                129
                                           x'81' otray sorter
.001004 ANOP
         ORG
               OLTXOT#+1
OLTXSORT DS
                FI1
                                           sorter mode
    description of the sorter codes
               ('&EQUATES' NE 'YES').Q01005
         AIF
OLTXNSRG EOU
                                           no sorter given
OLTXNSRT
          EQU
                241
                                           x'F1' sorter=no
OLTXSGRP FOU
                242
                                           x'F2' sorter=group
OLTXSCOL
         EQU
              243
                                           x'F3' sorter=collate
                244
OLTXSSTK
         EQU
                                           x'F4' sorter=stacker
OLTXSAUT
          FOU
                245
                                           x'F5' sorter=automat
.001005 ANOP
OLTXLOFS DS
                FL1
                                           left offset sign
    description of the sign codes
         AIF
               ('&EQUATES' NE 'YES').Q01006
OLTXPOST
          FOU
                78
                                           positive
OLTXNEGA
         FOU
                96
                                           negative
.Q01006 ANOP
OLTXLOFT
         DS
                Χ
                                           left offset binary; x'00'=no
                                           left offset
OLTXTOFS
          DS
                FL1
                                           top offset sign
OLTXTOFT
          DS
                χ
                                           top offset binary; x'00'= no
                                           top offset
```

```
OLTXDUPX DS
              FL1
                                         duplex
   description of the twoside codes
        AIF ('&EQUATES' NE 'YES').Q01007
OLTXNDUP EQU
                                         no simplex/duplex
OLTXSPLX EOU
               241
                                         x'F1' simplex
               242
                                         x'F2' duplex
OLTXDPLX
        EQU
OLTXTDPX EQU
               243
                                         x'F3' tumble duplex
.001007 ANOP
OLTXHCOP DS
               CL3
                                         hdwr page copies; c'000' or
                                         first position=x'00' means
                                         no copies
OLTXRSV2 DS
               CL16
                                         free for use
        AIF ('&EQUATES' NE 'YES').Q01008
OLTX#
         EQU
              *-OLTXHDR
.001008 ANOP
```

C++ specifications

```
*/
/* description of the rotation codes
/* ENUM rotation
                                                                    */
#define OLTPRQXNO ROT GIVEN 0
                                     /* no rotation given
                                                                    */
                                     /* no rotation
                                                                    */
#define OLTPRQXROT_0 241
#define OLTPRQXROT 90 242
                                     /* rotation 90
                                                                    */
#define OLTPRQXROT 180 243
                                     /* rotation 180
                                                                    */
#define OLTPRQXROT_270 244
                                                                    */
                                      /* rotation 270
                                                                    */
/* description of the input tray selection mode
                                                                    */
/* ENUM _intray_sel
                                      /* selection via a format
#define OLTPRQXBY FORMAT 255
                                                                    */
                                                                    */
                                      /* name
#define OLTPROXIGNORE O
                                      /* no input tray given
                                                                    */
/* description of given input tray
                                                                    */
/* ENUM input tray
                                                                    */
#define OLTPRQXNO ITRAY O
                                      /* no input tray given
                                                                    */
                                                                    */
#define OLTPRQXI TRAY NR 99
                                      /* max num tray #
#define OLTPRQXI TRAY C5 239
                                      /* x'EF' C5
                                                                    */
                                                                    */
#define OLTPRQXI_TRAY_DL 240
                                      /* x'F0' DL
#define OLTPRQXI TRAY COM10 241
                                                                    */
                                      /* x'F1' COM10
#define OLTPRQXI TRAY MONARCH 242
                                                                    */
                                      /* x'F2' monarch
#define OLTPRQXI_TRAY_DOUBLET 243
                                      /* x'F3' double letter
                                                                    */
#define OLTPRQXI TRAY LETTER 244
                                                                    */
                                      /* x'F4' letter
#define OLTPRQXI TRAY LEGAL 245
                                                                    */
                                      /* x'F5' legal
#define OLTPRQXI_TRAY_EXEC 246
                                      /* x'F6' exec
                                                                    */
#define OLTPRQXI_TRAY_INVOICE 247
                                                                    */
                                      /* x'F7' invoice
```

```
#define OLTPRQXI TRAY FOLIO 248
                                      /* x'F8' folio
                                                                    */
                                                                    */
#define OLTPROXI TRAY B5 249
                                      /* x'F9' B5
#define OLTPRQXI_TRAY_B4 250
                                                                    */
                                      /* x'FA' B4
                                                                    */
#define OLTPRQXI TRAY A5 251
                                      /* x'FB' A5
                                                                    */
#define OLTPROXI TRAY A4 252
                                      /* x'FC' A4
#define OLTPRQXI TRAY A3 253
                                      /* x'FD' A3
                                                                    */
#define OLTPRQXI TRAY MANUAL 254
                                      /* x'FE' manual
                                                                    */
                                                                    */
/* description of the output tray code
/* ENUM _output_tray
                                                                    */
                                                                    */
#define OLTPRQXNO OUTPUT TRAY O
                                   /* no output tray
#define OLTPRQXO TRAY NR 99
                                     /* output tray max #
                                                                    */
#define OLTPRQXO TRAY SORTER 129 /* x'81' otray sorter
                                                                    */
/* description of the twoside codes
                                                                    */
/* ENUM _simplex_duplex
                                                                    */
#define OLTPRQXNO DUPLEX 0
                                     /* no simplex/duplex
                                                                    */
#define OLTPRQXSIMPLEX 241
                                     /* x'F1' simplex
                                                                    */
#define OLTPRQXDUPLEX 242
                                     /* x'F2' duplex
                                                                    */
#define OLTPRQXTUMBLE DUPLEX 243
                                     /* x'F3' tumble duplex
                                                                    */
/* description of the sorter codes
                                                                    */
/* ENUM sorter
                                                                    */
#define OLTPRQXNO SORTER GIVEN O
                                     /* no sorter given
                                                                    */
                                                                    */
#define OLTPRQXNO SORTER 241
                                      /* x'F1' sorter=no
                                                                    */
#define OLTPRQXSORTER GROUP 242
                                      /* x'F2' sorter=group
                                                                    */
#define OLTPRQXSORTER_COLLATE 243
                                     /* x'F3' sorter=collate
#define OLTPRQXSORTER_STACKER 244
                                      /* x'F4' sorter=stacker
                                                                    */
#define OLTPRQXSORTER AUTO 245
                                      /* x'F5' sorter=automat
                                                                    */
                                                                    */
/* description of the sign codes
                                                                    */
/* ENUM sign
#define OLTPROXPOSITIVE 78
                                     /* positive
                                                                    */
#define OLTPROXNEGATIVE 96
                                      /* negative
                                                                    */
                                                                     */
/* description of extension part
struct OLTPRQXext_md1 {
#pragma aligned 1
       unsigned char rotation;
                                     /* rotation code
                                                                     */
       /* intray
                                                                     */
        union /* intray union */ {
               unsigned char intray;
                                                                     */
                                       /* input tray number
               unsigned char intray_format;
                                                                     */
                                       /* input trav format
               unsigned char input_tray;
                                       /* input tray code
                                                                     */
        } intray union;
```

```
/* outtrav
                                                                        */
        union /* outtray_union */ {
                unsigned char outtray;
                                        /* output tray number
                                                                       */
                unsigned char output tray;
                                         /* output tray code
                                                                       */
        } outtray_union;
        unsigned char sorter:
                                        /* sorter mode
                                                                       */
        unsigned char left_offset_sign; /* left offset sign
                                                                       */
        unsigned char left_offset;
                                       /* left offset binary;
                                                                       */
                                        /* x'00'=no left offset
                                                                       */
        unsigned char top_offset_sign; /* top offset sign
                                                                       */
        unsigned char top offset:
                                        /* top offset binary; x'00'=
                                                                       */
                                        /* no top offset
                                                                       */
        unsigned char duplex:
                                        /* duplex
                                                                       */
        char hdwr copies[3];
                                        /* hdwr page copies; c'000'
                                                                       */
                                        /* or first position=x'00'
                                                                       */
                                        /* means no copies
                                                                       */
                                        /* free for use
                                                                       */
        char reserved 2[16];
}:
/* parameter area
                                                                       */
struct OLTPRQX md1 {
#pragma aligned 4
        /* header layout
                                                                        */
        struct ESMFHDR hdr:
        struct OLTPRQXext mdl extension;
                                         /* extension part layout
                                                                       */
}:
```

C specifications

The C interface implements the same functionality as the OLTPRQX Assembler macro. The header file is contained in the include file OLTPRQX.H. This file is located in the standard RSO library that is that is supplied with RSO.

The include file is divided into the following sections:

- definitions of constants for specific function groups
- complex data structures (records)
- a parameter list with references to complex data structures

The OLTPRQX.H file is included in the application program using an include call in a C application program, thus making the parameter list is available.

The standard header must be initialized with values that correspond to the interface.

Macro name	Unit	Function	Version
OLTPRQX	275	3	1

```
/* description of the rotation codes
                                                                    */
/* ENUM _rotation
                                                                    */
#define OLTPRQXNO ROT GIVEN 0
                                     /* no rotation given
                                                                    */
#define OLTPRQXROT 0 241
                                     /* no rotation
                                                                    */
#define OLTPRQXROT_90 242
                                     /* rotation 90
                                                                    */
#define OLTPRQXROT 180 243
                                     /* rotation 180
                                                                    */
#define OLTPRQXROT 270 244
                                      /* rotation 270
                                                                    */
                                                                    */
/* description of the input tray selection mode
                                                                    */
/* ENUM intray sel
#define OLTPRQXBY_FORMAT 255
                                     /* selection via a format
                                                                    */
                                      /* name
                                                                    */
#define OLTPROXIGNORE O
                                      /* no input tray given
                                                                    */
/* description of given input tray
                                                                    */
                                                                    */
/* ENUM input tray
                                                                    */
#define OLTPRQXNO ITRAY 0
                                     /* no input tray given
#define OLTPRQXI TRAY NR 99
                                     /* max num tray #
                                                                    */
#define OLTPRQXI_TRAY_C5 239
                                     /* x'EF' C5
                                                                    */
#define OLTPRQXI_TRAY_DL 240
                                      /* x'F0' DL
                                                                    */
#define OLTPRQXI TRAY COM10 241
                                     /* x'F1' COM10
                                                                    */
#define OLTPRQXI_TRAY_MONARCH 242
                                     /* x'F2' monarch
                                                                    */
#define OLTPRQXI_TRAY_DOUBLET 243
                                      /* x'F3' double letter
                                                                    */
#define OLTPRQXI TRAY LETTER 244
                                      /* x'F4' letter
                                                                    */
#define OLTPRQXI_TRAY_LEGAL 245
                                     /* x'F5' legal
                                                                    */
#define OLTPRQXI_TRAY_EXEC 246
                                                                    */
                                      /* x'F6' exec
#define OLTPRQXI TRAY INVOICE 247
                                                                    */
                                      /* x'F7' invoice
#define OLTPRQXI TRAY FOLIO 248
                                      /* x'F8' folio
                                                                    */
#define OLTPRQXI_TRAY_B5 249
                                      /* x'F9' B5
                                                                    */
#define OLTPRQXI TRAY B4 250
                                                                    */
                                      /* x'FA' B4
#define OLTPRQXI TRAY A5 251
                                                                    */
                                      /* x'FB' A5
#define OLTPRQXI_TRAY_A4 252
                                      /* x'FC' A4
                                                                    */
#define OLTPRQXI TRAY A3 253
                                      /* x'FD' A3
                                                                    */
#define OLTPRQXI_TRAY_MANUAL 254 /* x'FE' manual
                                                                    */
                                                                    */
/* description of the output tray code
/* ENUM output tray
                                                                    */
#define OLTPRQXNO OUTPUT TRAY O
                                                                    */
                                     /* no output tray
#define OLTPRQXO_TRAY_NR 99
                                     /* output tray max #
                                                                    */
#define OLTPRQXO TRAY SORTER 129 /* x'81' otray sorter
                                                                    */
                                                                    */
/* description of the twoside codes
```

```
/* ENUM simplex duplex
                                                                      */
#define OLTPRQXNO DUPLEX 0
                                                                      */
                                      /* no simplex/duplex
#define OLTPRQXSIMPLEX 241
                                       /* x'F1' simplex
                                                                      */
#define OLTPROXDUPLEX 242
                                       /* x'F2' duplex
                                                                      */
#define OLTPRQXTUMBLE_DUPLEX 243
                                       /* x'F3' tumble duplex
                                                                      */
/* description of the sorter codes
                                                                      */
/* ENUM sorter
                                                                      */
#define OLTPRQXNO SORTER GIVEN O
                                       /* no sorter given
                                                                      */
                                       /* x'F1' sorter=no
#define OLTPRQXNO_SORTER 241
                                                                      */
#define OLTPRQXSORTER GROUP 242
                                       /* x'F2' sorter=group
                                                                      */
#define OLTPRQXSORTER COLLATE 243
                                       /* x'F3' sorter=collate
                                                                      */
#define OLTPRQXSORTER STACKER 244
                                       /* x'F4' sorter=stacker
                                                                      */
#define OLTPRQXSORTER AUTO 245
                                        /* x'F5' sorter=automat
                                                                      */
                                                                      */
/* description of the sign codes
/* ENUM sign
                                                                      */
#define OLTPRQXPOSITIVE 78
                                       /* positive
                                                                      */
#define OLTPRQXNEGATIVE 96
                                       /* negative
                                                                      */
/* description of extension part
                                                                      */
struct OLTPRQXext mdl {
       unsigned char rotation;
                                      /* rotation code
        /* intray
                                                                      */
        union /* intray union */ {
                unsigned char intray;
                                      /* input tray number
                                                                      */
                unsigned char intray_format;
                                        /* input tray format
                unsigned char input_tray;
                                        /* input tray code
                                                                      */
        } intray union;
                                                                      */
        /* outtray
        union /* outtray union */ {
               unsigned char outtray; /* output tray number
                                                                      */
                unsigned char output_tray;
                                        /* output tray code
                                                                      */
        } outtray union;
        unsigned char sorter;
                                        /* sorter mode
                                                                      */
        unsigned char left_offset_sign; /* left offset sign
                                                                      */
        unsigned char left offset;
                                       /* left offset binary;
                                                                      */
                                        /* x'00'=no left offset
                                                                      */
        unsigned char top_offset_sign; /* top offset sign
                                                                      */
        unsigned char top offset;
                                       /* top offset binary; x'00'= */
                                        /* no top offset
                                                                      */
                                       /* duplex
                                                                      */
        unsigned char duplex;
        char hdwr copies[3];
                                       /* hdwr page copies; c'000'
                                                                      */
```

```
/* or first position=x'00'
                                                                          */
                                          /* means no copies
                                                                          */
        char reserved_2[16];
                                          /* free for use
                                                                          */
}:
/* parameter area
struct OLTPRQX md1 {
        /* header layout
                                                                          */
        struct ESMFHDR hdr:
        struct OLTPRQXext mdl extension;
                                          /* extension part layout
                                                                          */
} :
```

COBOL specifications

The COBOL interface implements the same functionality as the OLTPRQX Assembler macro.

The interface is called in COBOL with a call statement and a parameter area. The fields of this interface, described below, can be copied with the copy statement to the COBOL source text in the working storage structure. OLTPRQX-I-PL allows the initialization of the parameter list with default values.

OLTPRQXY is located in the standard RSO library that is supplied with RSO.

The macros esmhexay and fhdry are located in the standard SPOOL library that is supplied with SPOOL versions as of V3.0A.

```
000100***********************
000200* BEGIN-INTERFACE OLTPROX
000300*
                       OLTPRO extension definition
000400* TITLE
000500* NAME
                       OLTPROXY
000600* DOMAIN
                        RS0
000700* LANGUAGE
                        COBOL
000800* COPYRIGHT
                        (C) Fujitsu Siemens Computers GmbH 2001
000900*
                           ALL RIGHTS RESERVED
001000* COMPILATION-SCOPE USER
001100* INTERFACE-TYPE
                       CALL
001200* RUN-CONTEXT
                       TU
001300*
001400* PURPOSE
                        Interface parameter for OLTPRQ extension
001500*
001600* SYNTAX
                        Syntax Variant 1:
001700*
                        COPY OLTPROXY
```

```
001800*
001900* REMARKS
                       Welcome to RSO
002000*
002200*
002300* VERSION
                       001
002400* CRDATE
                       2000-12-14
002500* AUTHOR
                       F. Langelez
002600* UPDATE
                       Original issue
002700*
002800* END-INTERFACE
                       OLTPROX.
002900*************************
003000*********************
003100*
           OLTPROXY
                     - Declarations
003200**********************
003300* parameter area
003400 01 OLTPROX-PL.
003500*
         Standard header.
             COPY fhdry.
003600
003700
         02 OLTPRQX-EXTENSION.
003800*
             extension part layout
           03 OLTPRQX-ROTATION
                                         PIC X.
003900
004000*
               rotation code
004100
             88 OLTPRQX-NO-ROT-GIVEN
                                         VALUE SYS-X00.
004200*
                 no rotation given
004300
             88 OLTPROX-ROT-0
                                          VALUE "1".
004400*
                no rotation
                                         VALUE "2".
004500
             88 OLTPRQX-ROT-90
004600*
                rotation 90
004700
             88 OLTPRQX-ROT-180
                                         VALUE "3".
004800*
                 rotation 180
             88 OLTPROX-ROT-270
                                         VALUE "4".
004900
005000*
                 rotation 270
                                          PIC X.
005100
           03 OLTPROX-INTRAY
005200*
               input tray number
           03 OLTPRQX-INPUT-TRAY REDEFINES OLTPRQX-INTRAY PIC X.
005300
005400*
               input tray code
             88 OLTPROX-NO-ITRAY
                                          VALUE SYS-X00.
005500
005600*
                 no input tray given (0)
005700
             88
                OLTPRQX-I-TRAY-NR
                                          VALUE SYS-X63.
005800*
                \max num tray # (99)
                OLTPRQX-I-TRAY-C5
005900
             88
                                         VALUE SYS-XEF.
                x'EF' C5
006000*
006100
             88 OLTPRQX-I-TRAY-DL
                                         VALUE "0".
                 x'F0' DL
006200*
                OLTPRQX-I-TRAY-COM10
                                         VALUE "1".
006300
             88
006400*
                x'F1' COM10
                                         VALUE "2".
006500
             88 OLTPROX-I-TRAY-MONARCH
```

000000-1-		,50,	
006600*	00	x'F2' monarch	VALUE 602
006700	88	OLTPRQX-I-TRAY-DOUBLET	VALUE "3".
006800*	0.0	x'F3' double letter	VALUE 66.4.22
006900	88	OLTPRQX-I-TRAY-LETTER	VALUE "4".
007000*	0.0	x'F4' letter	VALUE 45.
007100	88	OLTPRQX-I-TRAY-LEGAL	VALUE "5".
007200*	0.0	x'F5' legal	
007300	88	OLTPRQX-I-TRAY-EXEC	VALUE "6".
007400*		x'F6' exec	
007500	88	OLTPRQX-I-TRAY-INVOICE	VALUE "7".
007600*		x'F7' invoice	
007700	88	OLTPRQX-I-TRAY-FOLIO	VALUE "8".
007800*		x'F8' folio	
007900	88	OLTPRQX-I-TRAY-B5	VALUE "9".
*008000		x'F9' B5	
008100	88	OLTPRQX-I-TRAY-B4	VALUE SYS-XFA.
008200*		x'FA' B4	
008300	88	OLTPRQX-I-TRAY-A5	VALUE SYS-XFB.
008400*		x'FB' A5	
008500	88	OLTPRQX-I-TRAY-A4	VALUE SYS-XFC.
008600*		x'FC' A4	
008700	88	OLTPRQX-I-TRAY-A3	VALUE SYS-XFD.
008800*		x'FD' A3	
008900	88	OLTPRQX-I-TRAY-MANUAL	VALUE SYS-XFE.
009000*		x'FE' manual	
009100	03 01	LTPRQX-OUTTRAY	PIC X.
009200*	Ol	utput tray number	
009300	03 01	LTPRQX-OUTPUT-TRAY REDEFINES	OLTPRQX-OUTTRAY PIC X.
009400*	Ol	utput tray code	
009500	88 OLTPRQX-NO-OUTPUT-TRAY		VALUE SYS-X00.
009600*		no output tray given	
009700	88	OLTPRQX-O-TRAY-NR	VALUE SYS-X63.
009800*		output tray max # (99)	
009900	88	OLTPRQX-O-TRAY-SORTER	VALUE SYS-X81.
010000*		x'81' otray sorter	
010100	03 01	LTPRQX-SORTER	PIC X.
010200*	S	orter mode	
010300	88	OLTPRQX-NO-SORTER-GIVEN	VALUE SYS-X00.
010400*		no sorter given	
010500	88	OLTPRQX-NO-SORTER	VALUE "1".
010600*		x'F1' sorter=no	
010700	88	OLTPRQX-SORTER-GROUP	VALUE "2".
010800*		x'F2' sorter=group	
010900	88	OLTPRQX-SORTER-COLLATE	VALUE "3".
011000*		x'F3' sorter=collate	
011100	88	OLTPRQX-SORTER-STACKER	VALUE "4".
011200*		x'F4' sorter=stacker	
011300	88	OLTPRQX-SORTER-AUTO	VALUE "5".

011500 03 0LTPROX-LEFT-OFFSET-SIGN PIC X. 011600* left offset sign 011700 88 0LTPROX-LOFPOSITIVE VALUE SYS-X4E. 011800* positive (78) 011900 88 0LTPROX-LOFNEGATIVE VALUE SYS-X6O. 012000* negative (96) 012100 03 0LTPROX-LEFT-OFFSET PIC X. 012200* left offset binary; 012300* x'00"no left offset 012400 03 0LTPROX-TOP-OFFSET-SIGN PIC X. 012500* top offset sign 012600 88 0LTPROX-TOPPOSITIVE VALUE SYS-X6O. 012700* positive (78) 012800 88 0LTPROX-TOPPOSITIVE VALUE SYS-X6O. 012900* negative (96) 013000 03 0LTPROX-TOP-OFFSET PIC X. 013100* top offset binary; 013200* x'00"-no top offset 013300 03 0LTPROX-TOP-OFFSET PIC X. 013400* duplex PIC X. 013400* duplex VALUE SYS-X6O. 013600* no simplex/duplex VALUE SYS-X6O. 013600* no simplex/duplex VALUE SYS-X0O. 0144000* x'F2' duplex VALUE SYS-X0O. 014500* no simplex/duplex VALUE SYS-X0O. 015700* unit number PIC X VALUE SYS-X03. 015500 04 OLTPROX-I-PUNTI PIC S9(4) COMP VALUE Z75. 015500 04 OLTPROX-I-FUNCTION PIC X VALUE SYS-X03. 016600* function 016100 04 OLTPROX-I-VERSION PIC X	0114001		1551	
Olifon* left offset sign Olifon* S8	011400*	00 0	x'F5' sorter=automat	DIC V
011700				PIC X.
011800*				VALUE CVC VAE
Oligon		00		VALUE SISTA4E.
012000*		0.0	·	VALUE CVC V60
012100		88		VALUE 313-X60.
Ol2200*		02 0		DIC V
012300*				PIC X.
012400			•	
1012500*				DIC V
012600				PIC X.
012700* positive (78)				VALUE CVC VAE
012800 88 OLTPRQX-TOFNEGATIVE VALUE SYS-X60. 012900* negative (96) PIC X. 013000 03 OLTPRQX-TOP-OFFSET PIC X. 013100* top offset binary; PIC X. 013200* x'00'=no top offset PIC X. 013400* duplex PIC X. 013400* duplex VALUE SYS-X00. 013600* no simplex/duplex VALUE "1". 013700 88 OLTPRQX-SIMPLEX VALUE "1". 013800* x'F1' simplex VALUE "2". 014000* x'F2' duplex VALUE "2". 014000* x'F2' duplex VALUE "3". 014200* x'F3' tumble duplex VALUE "3". 014300 03 OLTPRQX-TUMBLE-DUPLEX VALUE "3". 014400* hdwr page copies; c'000' or Out Town or 014500* first position=x'00' means no copies 014600 03 OLTPRQX-I-HDR. 015000 free for use 014800/ header layout 015300 03		88		VALUE SYS-X4E.
012900* negative (96)		0.0	·	VALUE CVC VCO
013000		88		VALUE 313-X60.
013100*		02 0		DIC V
013200* x'00'=no top offset 013300 03 OLTPRQX-DUPLEX-MODE PIC X. 013400* duplex 013500 88 OLTPRQX-NO-DUPLEX VALUE SYS-X00. 013600* no simplex/duplex 013700 88 OLTPRQX-SIMPLEX VALUE "1". 013800* x'F1' simplex 013900 88 OLTPRQX-DUPLEX VALUE "2". 014000* x'F2' duplex 014100 88 OLTPRQX-TUMBLE-DUPLEX VALUE "3". 014200* x'F3' tumble duplex 014300 03 OLTPRQX-HDWR-COPIES PIC X(3). 014400* hdwr page copies; c'000' or 014500* first position=x'00' means no copies 014600 03 OLTPRQX-RESERVED-2 PIC X(16). 014700* free for use 014800/ 014900* parameter area (initialization) 015000 01 OLTPRQX-I-PL. 015100 02 OLTPRQX-I-HDR. 015200* header layout 015300 03 OLTPRQX-I-IFID. 015400* header layout 015500 04 OLTPRQX-I-UNIT PIC S9(4) COMP 015600 015700* unit number 015800 04 OLTPRQX-I-FUNCTION PIC X 015900 016000* function				PIC X.
013300 03 OLTPRQX-DUPLEX duplex PIC X. 013400* duplex VALUE SYS-X00. 013500 88 OLTPRQX-NO-DUPLEX volume VALUE SYS-X00. 013600* no simplex/duplex VALUE "1". 013700 88 OLTPRQX-SIMPLEX volume VALUE "1". 013800* x'F1' simplex VALUE "2". 014900 88 OLTPRQX-DUPLEX volume VALUE "3". 014000* x'F2' duplex VALUE "3". 014100 88 OLTPRQX-TUMBLE-DUPLEX volume VALUE "3". 014200* x'F3' tumble duplex volume VALUE "3". 014300 03 OLTPRQX-HDWR-COPIES volume PIC X(3). 014400* hdwr page copies; c'000' or OI Means no copies 014600 03 OLTPRQX-RESERVED-2 pic X(16). PIC X(16). 014700* free for use PIC X(16). 014800/ 01 OLTPRQX-I-HDR. OLTPRQX-I-HDR. 015200* header layout OLTPRQX-I-INIT pic X 015300 04 OLTPRQX-I-FUNCTION pic X 015700				
O13400* duplex O13500 88 OLTPRQX-NO-DUPLEX VALUE SYS-X00. O13600* no simplex/duplex O13700 88 OLTPRQX-SIMPLEX VALUE "1". O13800* x'F1' simplex VALUE "2". O13900 88 OLTPRQX-DUPLEX VALUE "2". O14000* x'F2' duplex VALUE "3". O14200* x'F3' tumble duplex O14300 O3 OLTPRQX-TUMBLE-DUPLEX O14300 O3 OLTPRQX-HDWR-COPIES O14500* first position=x'00' means no copies O14600 O3 OLTPRQX-RESERVED-2 O14700* free for use O14800/ O14900* parameter area (initialization) O15000 O1 OLTPRQX-I-PL. O15100 O2 OLTPRQX-I-HDR. O15200* header layout O15300 O3 OLTPRQX-I-IFID. O15400* header layout O15500 O4 OLTPRQX-I-UNIT O15500 O4 OLTPRQX-I-UNIT O15600 O4 OLTPRQX-I-FUNCTION O15700* unit number O15800 O4 OLTPRQX-I-FUNCTION O15700* Unit number O15800 O4 OLTPRQX-I-FUNCTION O15700* Unit number O15800 O4 OLTPRQX-I-FUNCTION O15900 O15			·	DIC V
013500 88 OLTPRQX-NO-DUPLEX VALUE SYS-X00. 013600* no simplex/duplex 013700 88 OLTPRQX-SIMPLEX VALUE "1". 013800* x'F1' simplex 013900 88 OLTPRQX-DUPLEX VALUE "2". 014000* x'F2' duplex 014100 88 OLTPRQX-TUMBLE-DUPLEX VALUE "3". 014200* x'F3' tumble duplex 014300 03 OLTPRQX-HDWR-COPIES PIC X(3). 014400* hdwr page copies; c'000' or 014500* first position=x'00' means no copies 014600 03 OLTPRQX-RESERVED-2 PIC X(16). 014700* free for use 014800/ 014900* parameter area (initialization) 015000 01 OLTPRQX-I-PL. 015100 02 OLTPRQX-I-HDR. 015200* header layout 015300 03 OLTPRQX-I-IFID. 015400* header layout 015500 04 OLTPRQX-I-UNIT PIC S9(4) COMP 015600 VALUE 275. 015700* unit number 015800 04 OLTPRQX-I-FUNCTION PIC X 015900 016000* function				PIC X.
013600* no simplex/duplex 013700 88 OLTPRQX-SIMPLEX VALUE "1". 013800* x'F1' simplex 013900 88 OLTPRQX-DUPLEX VALUE "2". 014000* x'F2' duplex 014100 88 OLTPRQX-TUMBLE-DUPLEX VALUE "3". 014200* x'F3' tumble duplex 014300 03 OLTPRQX-HDWR-COPIES PIC X(3). 014400* hdwr page copies; c'000' or 014500* first position=x'00' means no copies 014600 03 OLTPRQX-RESERVED-2 PIC X(16). 014700* free for use 014800/ 014900* parameter area (initialization) 015000 01 OLTPRQX-I-PL. 015100 02 OLTPRQX-I-HDR. 015200* header layout 015300 03 OLTPRQX-I-IFID. 015400* header layout 015500 04 OLTPRQX-I-UNIT PIC S9(4) COMP 015600 VALUE 275. 015700* unit number 015800 04 OLTPRQX-I-FUNCTION PIC X 015900 016000* function				VALUE CVC_VOO
013700 88 OLTPRQX-SIMPLEX VALUE "1". 013800* x'F1' simplex 013900 88 OLTPRQX-DUPLEX VALUE "2". 014000* x'F2' duplex 014100 88 OLTPRQX-TUMBLE-DUPLEX VALUE "3". 014200* x'F3' tumble duplex 014300 03 OLTPRQX-HDWR-COPIES PIC X(3). 014400* hdwr page copies; c'000' or 014500* first position=x'00' means no copies 014600 03 OLTPRQX-RESERVED-2 PIC X(16). 014700* free for use 014800/ 014900* parameter area (initialization) 015000 01 OLTPRQX-I-PL. 015100 02 OLTPRQX-I-HDR. 015200* header layout 015300 03 OLTPRQX-I-IFID. 015400* header layout 015500 04 OLTPRQX-I-UNIT PIC S9(4) COMP 015600 015700* unit number 015800 04 OLTPRQX-I-FUNCTION PIC X 015900 016000* function		00		VALUE 313-XUU.
013800* x'F1' simplex 013900 88 OLTPRQX-DUPLEX VALUE "2". 014000* x'F2' duplex 014100 88 OLTPRQX-TUMBLE-DUPLEX VALUE "3". 014200* x'F3' tumble duplex 014300 03 OLTPRQX-HDWR-COPIES PIC X(3). 014400* hdwr page copies; c'000' or 014500* first position=x'00' means no copies 014600 03 OLTPRQX-RESERVED-2 PIC X(16). 014700* free for use 014800/ 014900* parameter area (initialization) 015000 01 OLTPRQX-I-PL. 015100 02 OLTPRQX-I-HDR. 015200* header layout 015300 03 OLTPRQX-I-IFID. 015400* header layout 015500 04 OLTPRQX-I-UNIT PIC S9(4) COMP 015600		ΩΩ		VALUE "1"
013900 88 OLTPRQX-DUPLEX		00		VALUE 1.
014000* x'F2' duplex 014100 88 OLTPRQX-TUMBLE-DUPLEX VALUE "3". 014200* x'F3' tumble duplex 014300 03 OLTPRQX-HDWR-COPIES PIC X(3). 014400* hdwr page copies; c'000' or 014500* first position=x'00' means no copies 014600 03 OLTPRQX-RESERVED-2 PIC X(16). 014700* free for use 014800/ 014900* parameter area (initialization) 015000 01 OLTPRQX-I-PL. 015100 02 OLTPRQX-I-HDR. 015200* header layout 015300 03 OLTPRQX-I-IFID. 015400* header layout 015500 04 OLTPRQX-I-UNIT PIC S9(4) COMP 015600 unit number 015700* unit number 015800 04 OLTPRQX-I-FUNCTION PIC X 015900 VALUE SYS-X03. 016000* function		ΩΩ		VALUE "2"
014100 88 OLTPRQX-TUMBLE-DUPLEX VALUE "3". 014200* x'F3' tumble duplex 014300 03 OLTPRQX-HDWR-COPIES PIC X(3). 014400* hdwr page copies; c'000' or 014500* first position=x'00' means no copies 014600 03 OLTPRQX-RESERVED-2 PIC X(16). 014700* free for use 014800/ 014900* parameter area (initialization) 015000 01 OLTPRQX-I-PL. 015100 02 OLTPRQX-I-HDR. 015200* header layout 015300 03 OLTPRQX-I-IFID. 015400* header layout 015500 04 OLTPRQX-I-UNIT PIC S9(4) COMP 015600 vALUE 275. 015700* unit number 015800 04 OLTPRQX-I-FUNCTION PIC X 015900 VALUE SYS-X03. 016000* function		00		VALUL Z.
014200* x'F3' tumble duplex 014300 03 OLTPRQX-HDWR-COPIES PIC X(3). 014400* hdwr page copies; c'000' or 014500* first position=x'00' means no copies 014600 03 OLTPRQX-RESERVED-2 PIC X(16). 014700* free for use 014800/ 014900* parameter area (initialization) 015000 01 OLTPRQX-I-PL. 015100 02 OLTPRQX-I-HDR. 015200* header layout 015300 03 OLTPRQX-I-IFID. 015400* header layout 015500 04 OLTPRQX-I-UNIT PIC S9(4) COMP 015600 VALUE 275. 015700* unit number 015800 04 OLTPRQX-I-FUNCTION PIC X 015900 VALUE SYS-X03. 016000* function		ΩΩ		VALUE "2"
014300		00		VALUE 3.
O14400*		U3 U	The state of the s	DIC X(3)
014500* first position=x'00' means no copies 014600 03 OLTPRQX-RESERVED-2 PIC X(16). 014700* free for use 014800/ 014900* parameter area (initialization) 015000 01 OLTPRQX-I-PL. 015100 02 OLTPRQX-I-HDR. 015200* header layout 015300 03 OLTPRQX-I-IFID. 015400* header layout 015500 04 OLTPRQX-I-UNIT PIC S9(4) COMP 015600 VALUE 275. 015700* unit number 015800 04 OLTPRQX-I-FUNCTION PIC X 015900 016000* function				110 /(3).
014600				conies
014700* free for use 014800/ 014900* parameter area (initialization) 015000 01 OLTPRQX-I-PL. 015100 02 OLTPRQX-I-HDR. 015200* header layout 015300 03 OLTPRQX-I-IFID. 015400* header layout 015500 04 OLTPRQX-I-UNIT PIC S9(4) COMP 015600 VALUE 275. 015700* unit number 015800 04 OLTPRQX-I-FUNCTION PIC X 015900 VALUE SYS-X03. 016000* function				·
014800/ 014900* parameter area (initialization) 015000 01 OLTPRQX-I-PL. 015100 02 OLTPRQX-I-HDR. 015200* header layout 015300 03 OLTPRQX-I-IFID. 015400* header layout 015500 04 OLTPRQX-I-UNIT PIC S9(4) COMP 015600 VALUE 275. 015700* unit number 015800 04 OLTPRQX-I-FUNCTION PIC X 015900 VALUE SYS-X03. 016000* function			-	110 /(10).
014900* parameter area (initialization) 015000 01 OLTPRQX-I-PL. 015100 02 OLTPRQX-I-HDR. 015200* header layout 015300 03 OLTPRQX-I-IFID. 015400* header layout 015500 04 OLTPRQX-I-UNIT PIC S9(4) COMP 015600 VALUE 275. 015700* unit number 015800 04 OLTPRQX-I-FUNCTION PIC X 015900 VALUE SYS-X03. 016000* function		'	ree for ase	
015000 01 OLTPRQX-I-PL. 015100 02 OLTPRQX-I-HDR. 015200* header layout 015300 03 OLTPRQX-I-IFID. 015400* header layout 015500 04 OLTPRQX-I-UNIT PIC S9(4) COMP 015600 VALUE 275. 015700* unit number 015800 04 OLTPRQX-I-FUNCTION PIC X 015900 VALUE SYS-X03. 016000* function		rameter	area (initialization)	
015100	·			
015200* header layout 015300 03 OLTPRQX-I-IFID. 015400* header layout 015500 04 OLTPRQX-I-UNIT PIC S9(4) COMP 015600 VALUE 275. 015700* unit number 015800 04 OLTPRQX-I-FUNCTION PIC X 015900 VALUE SYS-X03. 016000* function				
015300				
015400* header layout 015500 04 OLTPRQX-I-UNIT PIC S9(4) COMP 015600 VALUE 275. 015700* unit number 015800 04 OLTPRQX-I-FUNCTION PIC X 015900 VALUE SYS-X03. 016000* function				
015500				
015600 VALUE 275. 015700* unit number 015800 04 OLTPRQX-I-FUNCTION PIC X 015900 VALUE SYS-X03. 016000* function				PIC S9(4) COMP
015700* unit number 015800 04 OLTPRQX-I-FUNCTION PIC X 015900 VALUE SYS-X03. 016000* function		0 1		
015800			unit number	
015900 VALUE SYS-X03. 016000* function		04		PIC X
016000* function				
			function	
		04	OLTPRQX-I-VERSION	PIC X

016200			VALUE SYS-X01.
016300*		version	
016400	03	OLTPRQX-I-RC-NBR	PIC S9(9) USAGE BINARY
016500			VALUE −1.
016600*	00 0	return code	
016700		LTPRQX-I-EXTENSION.	
016800*		xtension part layout	DIC V
016900	03	OLTPRQX-I-ROTATION	PIC X
017000			VALUE SYS-X00.
017100*	00	rotation code	DIC V
017200	03	OLTPRQX-I-INTRAY	PIC X
017300			VALUE SYS-X00.
017400*	0.0	input tray number	DIC V
017500	03	OLTPRQX-I-OUTTRAY	PIC X
017600			VALUE SYS-X00.
017700*	00	output tray number	DIC V
017800	03	OLTPRQX-I-SORTER	PIC X
017900			VALUE SYS-X00.
018000*	03	sorter mode	PIC X
018100 018200	03	OLTPRQX-I-LEFT-OFFSET-SIGN	VALUE SYS-X4E.
018300*		loft offset sign	VALUE 313-14E.
018300^	03	<pre>left offset sign OLTPRQX-I-LEFT-OFFSET</pre>	PIC X
018500	03	ULIPRUX-I-LEFI-UFFSEI	VALUE SYS-X00.
018600*		left offset binary;x'00'=no l	
018700	03	OLTPRQX-I-TOP-OFFSET-SIGN	PIC X
018700	03	OLIFNAN I TOF OLISLI SIGN	VALUE SYS-X4E.
018900*		top offset sign	VALUE 313 A4L.
010900	03	OLTPRQX-I-TOP-OFFSET	PIC X
019000	03	OLIFNON I FOR OTTSET	VALUE SYS-X00.
019100*		top offset binary; x'00'=no t	
019200"	03	OLTPRQX-I-DUPLEX	PIC X
019300	03	OLIFNON I DOFLEN	VALUE SYS-X00.
019400*		duplex	VALUE 313-X00.
019600	03	OLTPRQX-I-HDWR-COPIES	PIC 9(3)
019000	03	OLITINGN I HOWN COLLES	VALUE 000.
019700*		hdwr page copies; c'000' or	VALUE UUU.
019800*		first position=x'00' means no	conies
020000	03	OLTPRQX-I-RESERVED-2	PIC X(16)
020000	03	OLITINGA I NESERVED Z	VALUE LOW-VALUES.
020100			VALUE LOW VALUES.

Example

```
IDENTIFICATION DIVISION.
 PROGRAM-ID. tutmexcb.
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SPECIAL-NAMES.
    TERMINAL IS v-terminal,
    SYMBOLIC CHARACTERS
    COPY esmhexav.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 COPY oltprqxy.
 PROCEDURE DIVISION.
MAIN SECTION.
 p-main.
    PERFORM s-test001.
    PERFORM s-test002.
 p-exit.
    STOP RUN.
* TEST 001 : use default values
 s-test001 SECTION.
*----*
p-test001-strt.
*----*
    DISPLAY "TEST1 (DFT VALUES) OF TUTMEXTCB" UPON v-terminal.
    MOVE oltprqx-i-pl TO oltprqx-pl.
    DISPLAY oltprqx-pl UPON v-terminal.
 p-test001-exit.
    EXIT.
* TEST 002 : initialise some fields
 s-test002 SECTION.
  ____*
 p-test002-strt.
*----*
    MOVE oltprqx-i-pl TO oltprqx-pl.
```

```
* select rotation=0, input tray= 3, output tray=sorter stacker,

* left offset of +4, top offset of -4,no duplex selection and one

* hardware copy.

*

SET oltprqx-rot-0 TO TRUE.

MOVE SYS-X03 TO oltprqx-input-tray.

SET oltprqx-o-tray-sorter TO TRUE.

SET oltprqx-sorter-stacker TO TRUE.

SET oltprqx-lofpositive TO TRUE.

MOVE SYS-X04 TO oltprqx-left-offset.

SET oltprqx-tofnegative TO TRUE.

MOVE SYS-X04 TO oltprqx-top-offset.

SET oltprqx-no-duplex TO TRUE.

MOVE "001" TO oltprqx-hdwr-copies.

*

p-test002-exit.

*------*

*

EXIT.
```

PRNTDOC (for SPOOL, SPS, RSO and Dprint) - output file

User group: Nonprivileged users

Programming languages: Assembler, C, COBOL

Macro type: S

The PRNTDOC macro enables you to output files on a printer. By specifying the relevant operands you can control the print job. If a print job contains nothing other than the name of the file to be printed, SPOOL uses default values.

The specifications in the PRNTDOC macro are entered in a SPOOL control block (SCB) along with the default values for device selection. The SCB is placed in one of the spoolout queues (local spoolout, RSO) and managed as a separate job with a separate TSN.

You can monitor the processing of your spoolout jobs by means of the SHOW-PRINT-JOB-STATUS command, modify specific parameters of a print job by means of the MODIFY-PRINT-JOB-ATTRIBUTES command, change the priority of your jobs by means of the CHANGE-TASK-PRIORITY command or cancel them by means of the CANCEL-PRINT-JOB command.



The PRNTDOC macro is an enhanced version of the PRNT macro. The functional scope of PRINT-FILE meets the needs of SPOOL Version 2.7B, but only PRNTDOC allows you to use the additional and enhanced features available with the new version of SPOOL, Version 3.0A.

The new SPOOL version, V3.0, offers programmers the functionality of this macro not only in assembly language (as in the case of PRNT up to now) but also in C and COBOL. A "conventional" call format is available for assembly language, C programmers can use the PRNTDOC functions via an SVC call, and COBOL programmers can access a special module from the runtime library that implements PRNTDOC functions. You will find the parameter list descriptions required to use the macro and an example in each of the programming languages after the operand descriptions as of page 109.

Controlling SPOOLOUT

Selecting a printer

If a print job does not specify which printer to use, SPOOL assigns it to any available high speed printer. The number of characters this printer prints per line depends on the form.

If the job is to be printed on a particular printer, you specify this printer in the PRTYPE operand structure (see the operand description on page 89).

The operands listed below have minimum requirements with regard to the printer(s) used. Thus, if you specify one of these operands, you influence printer selection.

Operand	Printer required (minimum requirement)
PRNAME=device	RSO printer
PRTYPE=*LP65-PRINTER	LP65 printer
PRTYPE=*APA-PRINTER	APA printer
PRTYPE=*HP-PRINTER	HP printer or HP90 printer
USERRES [≡*DUMMY]	HP, HP90 or APA printer
PAGECOP [=*STD and =0]	HP, HP90 or LP65 printer
PAGE-DEF = <name></name>	APA printer
POOLNAM	HP printer or HP90 printer
FOB	HP printer or HP90 printer
ROT	HP printer or HP90 printer
OVERLAY (<face>,<reverse>)</reverse></face>	LP65 printer

If none of the operands in this table is specified, the default value in the SPOOL parameter file applies for printer selection.

Specifying the form

If a print job does not specify which form to use, SPOOL uses the default form (STD) defined in the SPOOL parameter file.

The default STD form normally refers to:

- the default loop C1 with a line density of 6 lpi and the vertical tab "channel 1" (normally the beginning of the page) on line 3
- the default rotation loop R06 for HP printers and HP90 printers with a line density of 6 lpi and a form length of 13.5 inches

To use another form, enter the desired form using FORM=....

Controlling paper feed

Each form is accompanied by the appropriate paper feed (loop) information. Unless you specify otherwise, the loop assigned to the form is used automatically. However, LOOP=... also lets you specify a different loop name. For all printers, this loop must be contained in the system PRFILE \$SYSSPOOL.PRFILE or an appropriate user PRFILE.



When explicitly specifying a loop, you should ensure that the length of the loop corresponds to the length of the form.

For printers with a loadable vertical format buffer, specifying a paper form (FORM operand) loads the loop associated with this form into the buffer. If the LOOP operand is specified in the same spoolout job, the feed is governed by the loop specified in LOOP. In all cases, the loop called must be available in a PRFILE.

If the paper form and font are not specified, SPOOL assumes that the vertical tab "channel 1" is set to the start of a new page in the loop (line 3).

The default loop "C1", the default font "101" and other fonts intended for laser printers are normally contained in the system PRFILE.

Defining loops yourself

You can use the PRM utility routine to define and administer your own loops. For more details, see the "PRM (BS2000/OSD)" manual.

Feed control characters

EBCDIC feed control characters:

Feed control characters	Effect
X'40' through X'4F'	n line feeds before printing, 1 line feed after printing
X'00' through X'0F'	n line feeds after printing
X'C1' through X'CB'	Channel feed before printing
X'81' through X'8B'	Channel feed after printing

Siemens (EBCDIC) and IBM printer control byte formats:

Value	Bit position	Bit position and meaning (EBCDIC)							
	2 ⁷	2 ⁶	2^5	2 ⁴	2^3	2 ²	2 ¹	2^{0}	
0	Line feed	After printing	Always 0	Always 0	Number of lines or		Number of lines or	Number of lines or	
1	Channel feed	Before printing			V tab number Value: 8	V tab number Value: 4	V tab number Value: 2	V tab number Value 1	

Value	Bit position	Bit position and meaning (IBM)							
	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2^3	2 ²	2 ¹	2^{0}	
0	Line feed	V tab number	V tab number	Number of lines or	Number of lines or	Always 0	After printing	Always 1	
1	Channel feed	Value 8	Value 4	V tab number Value 2	V tab number Value 1		Before printing		

^{*)} V tab is the vertical tab "channel n"

ASA feed control characters and corresponding EBCDIC feed control characters:

ASA feed	EBCDIC feed	Effect
C'+'	X'00'	No line feed after printing
C'-'	X'42'	2-line feed before printing
C'0'	X'40'	1-line feed before printing
C'1'	X'C1'	Jump to channel 1 before printing
C'A'	X'CA'	Jump to channel 10 before printing
C'B'	X'CB'	Jump to channel 11 before printing
others	X'40'	No line feed before printing

Any invalid control characters are corrected in laser printers.

i

The vertical tab "channel 12" (X'CC') or (X'8C') is reserved for SPOOL.

The differences in printer hardware mean that ASA and Siemens feed control characters are not completely identical: Siemens printers advance not before but **after** printing a line - except in the case of X'00' (see above).

Specifying fonts

Every form entered in the SPOOL parameter file is assigned a font. This assignment is entered in the SPOOL parameter file.

If no font is specified in PRNTDOC, the font "101" is used as the default. This font is assigned to the "STD" form in the SPOOL parameter file for all printers except types 3337, 3338 and 3339. The font "101" has the OCR-B font and is defined for a line density of 6 lines per inch (lpi) and a character density of 10 characters per inch (cpi).

The CHARSET operand enables you to specify a font from the system PRFILE for HP/HP90 printers. If the desired font is in a user PRFILE, you must specify this in the USERRES operand.

For the CHARSET operand, specify

- either the names of up to four fonts (for HP/HP90 printers)
- or the name of a font pool containing up to 64 fonts (HP/HP90 printers)

to be used to print a file. The fonts are contained in the \$SYSSPOOL.PRFILE resource library. However, you can also create a private user PRFILE containing one or more of your own fonts that you have created using the PRM utility routine. You can also define a font pool yourself in a user PRFILE. To print a file using your own font, you must specify the file name using the USERRES operand.

You can also specify a font directly from the standard SPSLIB when printing in APA mode. If the desired font is contained in a user SPSLIB, this must be specified simultaneously via the USERRES operand, in the same way as for HP/HP90 printers. The fonts are contained in the printer control file \$SYSSPOOL.SYSPRT.SPS.021. You can use the software product SPM to create your own fonts in a user SPSLIB. For more detailed information see the "PRISMAproduction/BS2000 - SPS" manual. To print a file with a user-defined font, you must specify the file name in the USERRES operand.

Before starting to print the file, the SPOOL system loads the specified fonts from the specified PRFILE into the character memory of the laser printer. If a file is printed with DOCFORM=*TEXT in the PRNTDOC macro - i.e. control characters in the text are not interpreted (except in printer type 3365, see the "SPOOL (BS2000/OSD)" manual) - the first font specified in the CHARSET operand of the PRNTDOC macro is used to print the whole file. It is thus preferable to specify just one font. Conversely, if the print file text contains control characters to be interpreted by SPOOL - i.e. CONTMOD=*PAGE_MODE must be specified in PRNTDOC - the font to be used in each instance is selected from the fonts loaded as follows:

 The control information in the first record on each print page determines, among other things, which font should be used to start printing the page. Control characters within the print data enable the font to be changed anywhere on the page (for every character, if need be).

- The fonts specified for CHARSET are loaded into the printer's character memory when the command is processed.
- The procedure for switching from one font to another by means of control characters is as follows:

All fonts you specify (either explicitly or by specifying a pool name) are loaded on the requested device during scheduling. Only the first font specified can be used if DOCFORM=*TEXT applies. The number of fonts (specified either explicitly in PRNTDOC or implicitly by means of a pool) is compared with the upper limit defined as a global parameter at installation time. The scheduler checks whether the specified device has enough memory for all explicitly or implicitly specified fonts to be loaded.

Whenever possible, you should refer to the pools set up by systems support:

- either by specifying the pool name alone
- or by specifying a pool name and an index

The scheduler forms groups of jobs using the same font or the same pool, thus avoiding unnecessary font changing and reloading operations.

If PRNTDOC contains references to fonts not yet loaded into the device's character buffer, and if there is no more space in the character buffer, the controller checks the loaded fonts for those which are called least frequently; these are then replaced with new ones.

If more than four fonts are requested in the PRINT-DOCUMENT command, it may happen that none of the available HP laser printers is able to load the requested number of fonts. In this case, the relevant job cannot be processed.

In the information you request by means of the SHOW-USER-STATUS command, all jobs requiring more than four fonts (or overlays or the page rotation module) are marked by an asterisk in the OPT field. The SHOW-PRINT-JOB-STATUS command displays the number of fonts requested.

If you issue the PRNTDOC macro with the operands LOOP, CHARSET and so on for laser printers, the \$SYSSPOOL.PRFILE resource library is checked to see whether the relevant entries are available. If not, the PRNTDOC macro is rejected.

If you issue the PRNTDOC macro with the FORM operand and/or LOOP operand for printers with a loadable VFB, the resource library is checked to see whether the relevant entry is available. If not, the PRNTDOC macro is rejected.

Example

A file containing four different fonts is to be printed on a laser printer.

```
PRNTDOC ,..., CHARSET=(100,360,370, #XY)),...
```

To switch to font 370 (the third font), for example, one of the following control characters should be selected, depending on the desired character density:

X'06'	Character density 10 characters/inch and font in MXM section 3.
X'0A'	Character density 12 characters/inch and font in MXM section 3.
X'0E'	Character density 15 characters/inch and font in MXM section 3.

The character density for which font 3 (370) was originally defined should of course be taken into account.

Interpreting layout control characters

If you want to print a file containing layout control characters, when you create the job you can specify whether or not these control characters are to be interpreted by assigning the appropriate value to the DOCFORM operand. For more details of control characters, see the "SPOOL (BS2000/OSD)" manual.

SPOOL loads the font buffer and feed buffer as appropriate when processing the spoolout.



The operator can use the START-PRINTER-OUTPUT command to ensure that the laser printer jobs specified with DOCFORM=*TEXT may also be printed on an impact printer (line printer). In this case, the operands LOOP, PAGECOP, ADDCOP, USERRES, CHARSET and OVERLAY are not interpreted.

Examples of control characters in a file

Printable characters

The data record only contains printable characters (without linefeed characters)



The data record starts with a linefeed control character (V), followed by text (including an ISAM KEY (IK) with ISAM files).



The data record starts with the SAM KEY (KEY-POS=5), followed by the linefeed control character (V) and then the text.

Printable and control chars.

The data record contains text mixed with control characters (e.g. change font).

Defining your own forms, loops and fonts

For HP/HP90 printers:

Paper forms, loops and fonts are generally defined by systems support. You can, however, also create loops and fonts yourself, store them in a user-specific resource library (PRFILE) and use them for SPOOL jobs. The loops and fonts must relate to a paper form already available in the SPOOL parameter file. You can use the USERRES operand to direct SPOOL to use your resource library for this job. For further details, see the "PRM (BS2000/OSD)" manual.

For APA printers:

You can also create a private printer control file (SPSLIB) containing your own fonts etc. for printers that run in APA mode. In this case too, you use the USERRES operand to indicate to SPOOL that it should use this file. See also the "PRISMAproduction/BS2000 - SPS" manual.

Monitoring spoolout with job variables

SPOOL jobs can be monitored with job variables. Entering the name of a monitor variable (MONJV) in the PRNTDOC macro causes SPOOL to store a continuous stream of information on the processing of the job in the monitoring job variable.

As of BS2000 V10.0, these SPOOL job variables are not under the control of the software product "JV", but are controlled by SPOOL itself. This means that SPOOL also protects them from being accessed by other jobs. For instance, if a job variable is assigned to a spoolout job, it cannot be requested or used by another job, regardless of whether the spoolout job is queued, has been canceled or is currently being processed.

SPOOL job variables can also be password-protected. Before a password-protected SPOOL job variable is interrogated, the password must first be entered in the password table by means of the ADD-PASSWORD command or in the PRNTDOC macro by means of the JV-PASSWORD operand.

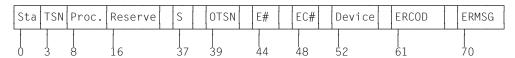
Error processing involving SPOOL job variables

If there are access problems involving SPOOL job variables, SPOOL can respond as follows:

- If the error occurs during syntax checking of the PRNTDOC macro, the macro is rejected on account of the operand in which the error has occurred. In the case of shared processing of several jobs (FAMILY operand), only the relevant file is rejected. Processing of the command then continues. If only one job is being processed, this is canceled.
 - If the operand STPROC=*AT_FILE_CLOSING is specified in the PRNTDOC macro, SPOOL continues processing despite the problem with the variable, since otherwise part of the data would be printed. Instead, SPOOL gives a warning to the user and continues processing the job without monitoring job variables.
- If the error occurs during spoolout, SPOOL outputs message SPS0450 on the trailer page but otherwise continues spoolout processing as normal.

Contents of the SPOOL job variable

The job variable is structured as follows:



The first three fields of the job variable are reserved for ENTER jobs. The subsequent fields are assigned to SPOOL and reflect the various processes involved in SPOOL job processing.

The fields "Status" and "Processor" are left-justified and are initially filled with blanks. The other fields are right-justified and are also initially filled with blanks. If the job variable does not exist, it is created if necessary when the command is checked for validity. You can use the SHOW-JV command to display the contents of a job variable.

Overview

Field	Contents	Meaning
Sta	\$S \$R \$P \$T \$A \$K	(spooled-in) job accepted and queued (running) job active (pre-processing) job of which a part is currently being preprocessed (terminated) job successfully terminated (aborted) job canceled or abnormally terminated (kept) job kept
TSN	tsn	Contains the TSN of the job preceded by the filler character "0"; in the case of replay jobs, tsn is the current TSN. The original TSN is in another field
Proc.	proc	As with ENTER-JOB jobs, contains the catalog ID for the system (home pubset) on which the job was issued or the tape read in
Res.		Reserved
S	0 1 2 3 otsn	Specifies whether the job involves tape processing Job does not contain preprocessed sections General job processing Job is on tape or has been stored there Job comes from a tape Contains the original TSN in the case of a replay job
E#	nnn	Number of copies to be printed
EC#	nnn	Number of copies already processed
Dev.	xxxxxx	For active jobs (\$R, \$T or \$A), specifies the device name, device short name or "TP" for user tapes
ERCOD	xxxxxxx	Error code for RSO jobs
ERMSG	xxxxxx	Identification number of the error message in the case of RSO jobs



The normal progression of job status is \$S, \$R and \$T (or \$A/\$K). A replay job receives the status \$T after the file has been transferred to tape. This tape can be played back on the same system. In this case, the job variable is checked as soon as the file is in the system. If the original job variable still exists, it is reinitialized and receives the status \$S.

The job variable subsystem is optional, but it cannot be unloaded while the system is running. If it is not available during the PRNTDOC macro validity check, and if a job variable is requested, SPOOL rejects the macro.

The SPOOL job variables cannot be specified in the commands SHOW-PRINT-JOB-STATUS, CANCEL-PRINT-JOB and CHANGE-TASK-PRIORITY.

SPOOL job variables are not supported for floppy disk operation.

Brief description of the PRNTDOC functions

Operand	Function	
ADDCOP	Requests additional printouts of the file	
CALLER	Specifies the caller of the macro	
CHARSET	Selects fonts for the output	
CHECKP	Specifies the checkpoint processing mode	
CONTMOD	Specifies how control characters are interpreted	
COVPAGE	Specifies settings controlling system exit routines for printing header and trailer pages	
DELF	Specifies whether the file is to be deleted on completion of output	
DNAME	Specifies the name of the cataloged file to be printed and the length of the file name	
DOCFORM	Specifies the type of the document contents	
DPPAR	Specifies the address of a parameter list in which Dprint-specific values fo the macro call are stored	
DSEMPAR	Specifies the address of a parameter list in which DSEM-specific values for the macro call are stored	
DTYPE	Specifies the type of the files to be output	
EAMN	Specifies the numbers of the EAM object module files to be output	
EFO	Specifies whether a film overlay - for HP90 and 3365 printers an EFO data overlay - is to be used	
FAMILY	Specifies whether a common TSN is assigned when several files or library elements are specified in a PRNTDOC macro	
FIRSREC	Specifies whether only a certain number of the input file's records are to be processed	
FOB	Specifies whether an FOB data overlay is to be used for processing the spoolout job	
FORM	Specifies the paper format to be used for the output	
FRMNAME	Type of document contents	
HEADLIN	Specifies whether a header line is to be printed on each page (except for the header and trailer pages)	
INTRAY	Specifies the paper input tray for the LP65 printer type	
INTRAYF	Specifies the paper input tray via a keyword	
JVPASSW	Specifies the password by which the job variable is protected	
LASTREC	Specifies whether only a certain number of the input file's records are to be processed	

(part 1 of 3)

Operand	Function	
LEFTMAR	Specifies whether the output text is to be indented by the specified number of columns	
LEFTOFF	Specifies the offset in millimeters from the left edge of the physical page, to the first character to be printed	
LIBELEM	Specifies the name, version and type of a PLAM library element to be printed	
LINEPP	Specifies how many lines are to be printed on a page	
LINESP	Specifies the number of line feeds or the type of control character interpretation	
LOCKF	Specifies whether the file is to be protected while the spoolout job is in the wait state	
LOOP	Specifies one or more loops	
LSTNN	Specifies the name and number of a SYSLST or SYSOUT system file	
MACID	Specifies the second to fourth characters (inclusively) of the field names and equates	
MF	Macro call type	
MONJV	Specifies the job variable in which information on job processing is to be stored	
OUTFORM	Specifies whether the output is to be only in character format or also in hexadecimal format	
OUTPART	Specifies whether the output is to be limited to a subset of all the logical print pages	
OUTTRAY	Specifies the paper output tray for the LP65 printer type	
OVERLAY	Specifies EFO data overlays for printers of the type LP65	
PAGECOP	Specifies the number of page copies	
PAGEPCL	Specifies which page definition is to be used for output on LP65 or APA printers	
PARAM	Specifies the address of the operand list	
POOLIND	Specifies the number of the font from a font pool	
POOLNAM	Specifies the name of the font pool	
PREFIX	Specifies the first character of the field names and equates	
PRJCLAS	Specifies the job class of the spoolout job	
PRJNAME	Specifies the job name of the spoolout job	
PRJPRIO	Specifies the priority of the spoolout job	
PRNAME	Specifies the print job's requested target printer or printer pool	

(part 2 of 3)

Operand	Function	
PROCADM	Specifies whether the spoolout job is to be executed under your own or a different user ID and account number	
PROGRAM Specifies the device name of a virtual printer		
PRTYPE	Specifies which printer type is to process the print job	
RECPART	Specifies that only a specific part of each selected record is to be processed	
ROT	Specifies whether the pages to be printed are to be rotated	
ROTLOOP	Specifies the loop with which the output is to be controlled in landscape format	
RSOPAR	Specifies the address of a parameter list in which RSO-specific values for the macro call are stored	
SECTREC	Specifies whether the file is to be structured by means of file marks	
SCHEDTIME	Specifies the time from which on a job can be printed. From that point on, t job will be subject to the standard rules governing the processing of print jobs.	
SPSPAR	Specifies the address of a parameter list in which SPS-specific values for the macro call are stored	
SRTMODE	Specifies the sort mechanism for the paper output tray	
STPROC	Specifies the time at which the system file is printed before the end of the job	
TRANTAB	B Specifies whether a code translation table is to be used to process the spoolout job	
TRUNC	Specifies what happens when lines are truncated	
TWOSIDE	Specifies for the LP65 printer whether paper is to be printed on one side or both sides	
TOPOFF	Specifies the offset in millimeters from the top edge of the physical page, to the first line to be printed	
USERRES	Specifies a user file containing all the required resources for printing on different printer types	
VARIANT	Variant of the generated parameter list	
VIRTUAL	Specifies whether a job is to be passed on via a virtual printer	

(part 3 of 3)

Format (assembly language)

Operation	Operands
PRNTDOC	MF=C/D/E/L/M
	,PREFIX= <u>S</u> / <name 11=""></name>
	,MACID= <u>CPA</u> / <name 13=""></name>
	,PARAM= <name 127=""></name>
	,VARIANT= <u>001</u> / <c-string 33=""></c-string>
	,CALLER= <u>*USER</u> / *SYSTEM
	,RSOPAR=*NONE / <var: pointer=""> / (<reg: pointer="">)</reg:></var:>
	,SPSPAR=*NONE / <var: pointer=""> / (<reg: pointer="">)</reg:></var:>
	,DPPAR=*NONE / <var: pointer=""> / (<reg: pointer="">)</reg:></var:>
	,DSEMPAR=*NONE / <var: pointer=""> / (<reg: pointer="">)</reg:></var:>
	,DTYPE= <u>*FILE</u> / *LIBRARY_ELEMENT / *POSIX / *EAM / *SYSTEM_FILE / <var: dtype_set="" enum-of=""></var:>
	,DNAME=(pointer, length) pointer: *NONE / <var: pointer=""> / (<reg: pointer="">) length: *STD / <integer 11024=""> / <var: 2="" int:=""> / (<reg: int:2="">)</reg:></var:></integer></reg:></var:>
	,EAMN=array(16): *STD / <integer 065535=""> / <var: 4="" int:=""> / (<reg: int:4="">)</reg:></var:></integer>
	,LSTNN=array(16): *DUMMY / <integer 0101=""> / *SYSLST / *SYSOUT / <var: 2="" int:=""> / (<reg: int:2="">)</reg:></var:></integer>

(part 1 of 7)

Operation	Operands
PRNTDOC	,LIBELEM=(element, version, type) element: *DUMMY / <var: 80="" char:=""> / (<reg: char:80="">) /</reg:></var:>
	,SECTREC=(ident, length, position) ident: *NONE / <c-string 160=""> / <var: 60="" char:=""> / (<reg: char:60="">) length: 0 / <integer 160=""> / <var: 1="" int:=""> / (<reg: int:1="">) position: *STD / <integer 12047=""> / <var: 2="" int:=""> / (<reg: int:2="">)</reg:></var:></integer></reg:></var:></integer></reg:></var:></c-string>
	,FIRSREC=(begin, ident, length, position, occur) begin: *BEGIN_OF_FILE / *BY_STRING_ID /
	,LASTREC=(begin, ident, length, position, occur) begin: *END OF FILE / *BY_STRING_ID /
	,RECPART=(first, last) first: 1 / <integer 132767=""> / <var: 2="" int:=""> / (<reg: int:2="">) last: *STD / <integer 132767=""> / <var: 2="" int:=""> / (<reg: int:2="">)</reg:></var:></integer></reg:></var:></integer>
	,LINESP=(spacing, position) spacing: *STD / *SPACE_1 / *SPACE_2 / *SPACE_3 / *BY_ASA_CONTROL / *BY_EBCDIC_CONTROL / *BY_IBM_CONTROL / *NO / <var: enum-of="" space_set:1=""> position: *STD / <integer 12040=""> / <var: 2="" int:=""> / (<reg: int:2="">)</reg:></var:></integer></var:>

(part 2 of 7)

Operation	Operands
PRNTDOC	,DOCFORM= <u>*TEXT</u> / *PAGE_FORMAT / *SPECIAL_FORMAT / <var: docform_set:1="" enum-of=""></var:>
	,CONTMOD=(mode, pcc, type) mode: *STD / *PHYSICAL / *LINE_MODE / *LOGICAL / *APA / *PAGE_MODE / <var: control_mode_set:1="" enum-of=""> pcc: *DUMMY / *YES / *NO / <var: enum-of="" pcc_set:1=""> type: *DUMMY / *COMPATIBLE / *HP / <var: control_type_set:1="" enum-of=""></var:></var:></var:>
	,ADDCOP= <u>0</u> / <integer 0255=""> / <var: 1="" int:=""> / (<reg: int:1="">)</reg:></var:></integer>
	,FAMILY= <u>*STD</u> / *YES / *NO / <var: enum-of="" family_set:1=""></var:>
	,CHECKP=*ON_PAGES / *ON_SECTION_RECORDS / <var: checkp_set:1="" enum-of=""></var:>
	,STPROC= <u>*IMMEDIATE</u> / *NO / *AT_FILE_CLOSING / <var: enum-of="" stdproc_set:1=""></var:>
	,MONJV=*NONE / *STD / <var: 54="" char:=""> / (<reg: char:54="">) / <c-string: 154="" filename="" with-wild=""></c-string:></reg:></var:>
	,JVPASSW=*NONE / <var: 4="" char:=""> / (<reg: char:4="">) / <c-string: 14="" c-string="" with-low=""></c-string:></reg:></var:>
	,PRJNAME=*JOB_NAME / <var: 8="" char:=""> / (<reg: char:8="">) / <c-string: 18="" c-string="" with-low=""></c-string:></reg:></var:>
	,PRJPRIO=*JOB_PRIORITY / <integer 30255=""> / <var: 1="" int:=""> / (<reg: int:1="">)</reg:></var:></integer>

(part 3 of 7)

Operation	Operands
PRNTDOC	,PROCADM=(user, account, passwd) user: *SAME / <var: 8="" char:=""> / (<reg: char:8="">) /</reg:></var:>
	,TRUNC= <u>*STD</u> / *DELETE_FILE / *KEEP_FILE / <var: enum-of="" truncation_set:1=""></var:>
	,LOCKF= <u>*STD</u> / *YES / *NO / <var: enum-of="" lockf_set:1=""></var:>
	,DELF= <u>*NO</u> / *YES / *DESTROY / <var: delf_set:1="" enum-of=""></var:>
	,OUTPART=(begin, end, dimension) begin: *BEGIN OF FILE / <var: 4="" int:=""> / (<reg: int:4="">) /</reg:></var:>
	,PAGECOP= <u>*STD</u> / <integer 0255=""> / <var: 2="" int:=""> / (<reg: int:2="">)</reg:></var:></integer>
	,LEFTMAR= <u>*STD</u> / <integer 031=""> / <var: 1="" int:=""> / (<reg: int:1="">)</reg:></var:></integer>
	,LINEPP= <u>*STD</u> / <integer 132767=""> / <var: 2="" int:="">/ (<reg: int:2="">)</reg:></var:></integer>
	,HEADLIN=*NO / *STD / *DATE / *FRECORD /
	,OUTFORM= <u>*CHARACTER</u> / *HEXADECIMAL / <var: enum-of="" outform_set:1=""></var:>

(part 4 of 7)

Operation	Operands
PRNTDOC	,TWOSIDE= <u>*STD</u> / *NO / *YES / *TUMBLE / <var: enum-of="" twoside_set:1=""></var:>
	,ROT=*NO / *ROT_0 / *ROT_90 / *ROT_180 / *ROT_270 /
	,INTRAY= <u>*STD</u> / <integer 199=""> / *IGNORE / <var: 1="" int:=""> / *BY-FORMAT</var:></integer>
	,INTRAYF= <var: _intray_format_set:1="" enum-of="">/*A3 / *A4 / *A5 /</var:>
	,OUTTRAY= <u>*STD</u> / <integer 199=""> / <var: 1="" int:=""> / *IGNORE / *SORTER</var:></integer>
	,COVPAGE=(text, header-exit, trailer-exit) text: *NONE / <var: 32="" char:=""> / (<reg: char:32="">) / <c-string: 132="" c-string=""> header-exit: *NO / <var: 4="" int:=""> / (<reg: int:4="">) / <integer 02147483639=""> trailer-exit: *NO / <var: 4="" int:=""> / (<reg: int:4="">) / <integer 02147483639=""> /</integer></reg:></var:></integer></reg:></var:></c-string:></reg:></var:>
	,FORM=*STD / <var: 6="" char:=""> / (<reg: char:6="">) / <c-string: 16="" alphanum-name=""></c-string:></reg:></var:>
	,LOOP= <u>*STD</u> / <var: 3="" char:=""> / (<reg: char:3="">) / <c-string: 13="" alphanum-name=""></c-string:></reg:></var:>
	,ROTLOOP= <u>*STD</u> / <var: 3="" char:=""> / (<reg: char:3="">) / <c-string: 13="" alphanum-name=""></c-string:></reg:></var:>

(part 5 of 7)

Operation	Operands
PRNTDOC	,CHARSET=array(16): *STD / <var: 3="" char:=""> / (<reg: char:3="">) / <c-string: 13="" alphanum-name=""></c-string:></reg:></var:>
	,POOLNAM=*NONE / <var: 4="" char:=""> / (<reg: char:4="">) / <c-string: 14="" name=""></c-string:></reg:></var:>
	,POOLIND= <u>0</u> / <integer 064=""> / <var: 1="" int:=""> / (<reg: int:1="">)</reg:></var:></integer>
	,EFO=*NONE / <var: 2="" char:=""> / (<reg: char:2="">) / <c-string: 22="" c-string=""></c-string:></reg:></var:>
	,OVERLAY=(face, reverse) face: *STD / *NONE / <integer 1127=""> / <var: 1="" int:=""> / (<reg: int:1="">) reverse: *STD / *NONE / <integer 1127=""> / <var: 1="" int:=""> / (<reg: int:1="">)</reg:></var:></integer></reg:></var:></integer>
	,FOB=*NONE / <var: 4="" char:=""> / (<reg: char:4="">) / <c-string: 14="" 14,="" alphanum-name="" c-string=""></c-string:></reg:></var:>
	,PAGEPCL=*STD / <integer 150000=""> / <var: 4="" int:=""> / (<reg: int:4="">)</reg:></var:></integer>
	,USERRES=*DUMMY / <var: 44="" char:=""> / (<reg: char:44="">) <c-string: 144="" c-string="" filename="" without-userid,=""></c-string:></reg:></var:>
	,TRANTAB=(name, file) name: *NONE / <var: 8="" char:=""> / (<reg: char:8="">) /</reg:></var:>
	,PRNAME=*STD / <c-string: 18="" c-string=""> / <var: 18="" char=""> / *IPP</var:></c-string:>
	,PRTYPE= <u>*ANY</u> / *LP65_PRINTER / *HP_PRINTER / *APA_PRINTER / <var: enum-of="" prtype_set:1=""></var:>

(part 6 of 7)

Operation	Operands
PRNTDOC	,SRTMODE= <u>*NO</u> / <var: enum-of_srtmode_set:1=""> / *GROUP / *COLLATE / *STACKER / *AUTOMATIC</var:>
	,VIRTUAL= <u>*STD</u> / <var: enum-of="" virtual_set:1=""> / *ALLOWED / *NOT-ALLOWED / *MUST</var:>
	,PROGRAM=(name, string) name: *ANY / <var: 8="" char:=""> /</var:>
	<pre><c-string: 18="" 18,="" alphanum-name="" c-string=""> string: *NONE / <var: 32="" char:=""> /</var:></c-string:></pre>
	,PRJCLAS= <u>*JOB-CLASS</u> / <var: 1="" int:=""> / <integer 1255=""></integer></var:>
	,FRMNAME=*STD / <c-string: 163="" c-string=""> / <var: 63="" char:=""> / (<reg: char:63="">)</reg:></var:></c-string:>
	,TOPOFF=*IGNORE / <integer -255255=""></integer>
	,LEFTOFF=*IGNORE / <integer -255255=""></integer>
	,SCHEDTIME = (schdate, schtime) schdate: *TODAY / <var: 110="" char=""> / <c-string 1010=""> schtime: *NOW / <var: 15="" char=""> / <c-string 55=""></c-string></var:></c-string></var:>
	,NOTIFPAR = <var: pointer=""> / *NONE</var:>

(part 7 of 7)

Description of the operands

ADDCOP=0 / <integer 0..255> / <var: int: 1> / (<reg: int:1>)

Specifies how many additional times the file is to be printed. Each additional printout has its own header page. An integer from 0 to 255 or the name of a field defined with FL with a length of 1 byte can be specified.

Default: 0; no additional printout.

CALLER=

Caller of the macro

CALLER=*USER

The caller is the user (TU).

CALLER=*SYSTEM

The caller is the system (TPR).

CHARSET=array(16): *STD / <var: char: 3> / (<reg: char:3>) /

<c-string: alphanum-name 1..3>

Names of the fonts or font pools to be used for output.

Up to 4 fonts may be specified in the list for local SPOOL, up to 16 for RSO. For local SPOOL, the fonts must be contained in the \$SYSSPOOL.PRFILE resource library or in a user PRFILE (specified by means of the USERRES operand).

If more than four fonts are to be used, a font pool must be specified (POOLNAM operand). If more than one font is to be used in a spoolout job, CONTMOD=

*PAGE_MODE must be specified. For DOCFORM=*TEXT, only the first specified font is used for printing the (entire) file. The name of the first font and the number of specified fonts are shown in the output for the command SHOW-PRINT-JOB-STATUS JOB-IDENTIFI-CATION=TSN(TSN=...).

Only FUJITSU-SIEMENS standard fonts may have names consisting of three numerical characters.

If the HEADLIN operand is specified, the first font specified is used for the header line. The default value for CHARSET for the form used can be displayed by means of SHOW-SPOOL-FORMS; the information is given in the C-S output field.

Note for HP printers:

systems support can specify whether the header page is printed with the default font or with the font specified in CHARSET. This default setting can be displayed by means of the SHOW-SPOOL-PARAMETERS command or SPSERVE statement; the information is given in the HEADER-PAGE: CHARACTER-SET=... output field.

CHARSET=*STD

The default font is selected for this printer type from the desired form. It can be displayed by means of SHOW-SPOOL-FORMS.

CHARSET=<c-string: alphanum-name 1..3>

Names of the fonts with which the spoolout job is to be executed. The string must be enclosed in quotes.

If DOCFORM = *TEXT (default) is specified, only the first font specified is used for printing. Up to 4 fonts can be specified for laser printers in local SPOOL operation. The printer control characters for switching fonts are interpreted only when CONTMOD = *PAGE_MODE is specified at the same time.

Up to 16 fonts can be specified for RSO printers. Font or character set identifiers (CSIs) in the text are interpreted only when CONTMOD = *LOGICAL is specified at the same time. The macro is rejected if a list of fonts is specified in conjunction with DOCFORM = *TEXT.

CHARSET=<var: char: 3> / (<reg: char:3>)

Name of a field defined with CL or a register containing the value. A string (3 bytes in length) is stored in this field or register and interpreted as the name of the font.

CHECKP=

Specifies whether checkpoint processing is to be performed by the controller on the basis of pages or SECTIONs.

CHECKP=*ON PAGES

Default restart mechanism.

When an interrupted job is restarted, processing is resumed from a point a given number of pages back.

CHECKP=*ON_SECTION_RECORDS

The operand value can be specified for all printer types but offers advantages particularly in the case of output to HP90 printers using the TWO-UP procedure and LP65 printers. With these types of printer, a physical page can comprise a number of logical pages without SPOOL detecting it (the information is in the PCL file); in other words, the default restart mechanism, which is geared to logical pages, is highly prone to errors with this type of printer.

SECTION records are used here as restart markers. You divide your files into sections with the aid of SECTION records. These SECTION records must contain the printer commands needed to ensure correct data processing. If a physical page contains a number of logical pages, the start of a physical page must also be clearly indicated in the SECTION record. If an error occurs, processing is resumed by means of HOLD-PRINT-JOB and RESUME-PRINT-JOB a given number of sections further on in the file; i.e. RESTART-POSITION = PAGE(...) or BACK(...) refers to sections rather than pages. Similarly, the values shown in error messages are not pages but sections.

To arrive at a correct result, a section must correspond to at least one physical page (ideally to precisely one page). If the operand value CONTMOD=*PAGE_MODE is specified together with CHECKP=*ON_SECTION_RECORDS, in the case of output to a laser printer, you must ensure that the SECTION records are located immediately ahead of records which contain the control character line at the start of the print page. As soon as repositioning takes place (PRNTDOC or restart of an interrupted job), the first record that SPOOL prints must contain the control character line.

CHECKP=<var: enum-of checkp set:1>

The restart is not controlled directly by means of the appropriate operand value; instead, it is controlled indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*ON_PAGES
1	*ON_SECTION_RECORDS
See the parameter list description as of page 109.	

CONTMOD=(*mode*, *pcc*, *type*)

Specifies how the control characters are to be interpreted (*mode*), whether the control character list has to be at the beginning of each page (*pcc*), and whether the control characters can be processed on printers of the type HP or HP90 (*type*).

mode: *STD / *PHYSICAL / *LINE_MODE / *LOGICAL / *APA / *PAGE-MODE /
<var: enum-of control mode set:1>

Specifies how the control characters are to be interpreted.

mode: *PHYSICAL

Only for RSO.

LINESP=*NO is set automatically, which means that you must implement page and line feeds with LINE-MODE control characters (i.e. VTSU codes, printer control characters and RENO commands) in the file itself.

If you specify LINESP=*BY_EBCDIC_CONTROL for records of type D-2, the feed control character in the first byte is interpreted as a line or page feed control character. Non-printable characters are also transferred to the printer (in contrast to CONTMOD= *LINE_MODE). You are responsible for the correctness of the control characters in the file (including feed control).

mode: *LINE_MODE

Only for RSO.

Records of type C (see appendix) are printed out. The records may contain data mixed with LINE-MODE control characters. LINE-MODE control characters are (in any combination):

- printer control characters (i.e. physical control character beginning with X'27' or X'3C')
- RENO commands
- VTSU codes

Control over record and file formats (including page and line feed with LINE-MODE control characters) rests solely with the user. With the 9025/9026 RENO page printer, users must also ensure that the printer is set to the correct start position on the paper.

A loop is not interpreted. Non-printable characters, i.e. characters with a hexadecimal value < X'40', are output as blanks.

The spoolout job is executed with the default font of the form used until you change the font in the file with the aid of LINE-MODE control characters.

mode: *LOGICAL

Only for RSO printers.

Specifies that records of type B-1 or B-2 are to be printed out, i.e. records which, in addition to a feed control character in the first byte, may contain data mixed with font identifiers, printer control characters, RENO commands and VTSU codes (see the "RSO (BS2000/OSD)" manual, section "Record type B-1/B-2"). With the exception of the VTSU codes VPA, NP, VT, NL and CR, which are output as blanks, the abovementioned control characters are interpreted.

A character set identifier, a VTSU code or a RENO command remains valid until a new control character is specified.

If the font identifier is omitted, the default font for the form is used.

Since page feed is implemented via a loop or constant line feed when CONTMODE= *LOGICAL is specified, the file should not contain the RENO commands \LF, \FF and \CR. Setting the form height is likewise not permitted.

mode: *APA

Specifies that the file to be printed contains APA printer control characters that are to be interpreted.

mode: *PAGE MODE

The control characters are to be interpreted as specific control characters for page printers.

mode: <var: enum-of control_mode_set:1>

How the control characters are to be interpreted is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

1	*STD	
2	*PHYSICAL	
4	*PAGE_MODE	
8	*LOGICAL	
16	*LINE_MODE	
64	*APA	
See the parameter list description as of page 109.		

pcc: *DUMMY / *YES / *NO / <var: enum-of pcc_set:1>

Specifies whether the control character list must be at the beginning of each page (i.e. in the loop always after the jump to the vertical tab "channnel 1"). The operand is only evaluated in the case of output to HP and HP90 printers (setting CONTMODE=*PAGE MODE).

pcc: *DUMMY

Default: the operand is not to be evaluated.

pcc: *YES

The control character list must be at the beginning of the page.

pcc: *NO

No control character list at the beginning of the page. In the case of output to HP printers, this means that the following functions cannot be controlled:

- Film overlay on individual pages of the file. If specified in the PRNTDOC macro, a film overlay is used on every page of the print file of the spoolout job.
- Page copies for individual pages of the file. As many copies as are specified in the PRNTDOC macro are output of all the print file's pages.
- Column-by-column indentation on individual pages. The value specified in the PRNTDOC macro applies to all the print file's pages.
- FOB data overlay on individual pages of the file. If specified in the PRNTDOC macro, a FOB data overlay is used on every page of the print file of the spoolout job.
- It is not possible to specify a copy reference number.
- Page rotation control for individual pages of the print file. All the pages are output either in portrait or landscape format, as specified in the PRNTDOC macro.

pcc: <var: enum-of pcc_set:1>

The control character list is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*YES	
1	*NO	
2	*DUMMY	
See the parameter list description as of page 109.		

type: *DUMMY / *COMPATIBLE / *HP / <var: enum-of control_type_set:1> Specifies whether the control characters are suitable for processing on HP or HP90 printers or if they must be converted to this format.

type: *DUMMY

The operand is not to be evaluated.

type: *COMPATIBLE

No HP or HP90 printer-specific control characters are stored in the file. SPOOL has to convert the control characters to this format.

type: *HP

HP or HP90 printer-specific control characters are stored in the file. Only printers of these types can process them.

type: <var: enum-of control_type_set:1>

Control character conversion is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*DUMMY	
1	*COMPATIBLE	
2	*HP	
See the parameter list description as of page 109.		

COVPAGE=(text, header-exit, trailer-exit)

Specifies parameters for system exit routines, which govern the printing of header and trailer pages.

text: *NONE / <c-string 1..32> / <var: char: 32> / (<reg: char:32>) Specifies whether a string is to appear on header and trailer pages.

text: *NONE

No string is to appear on header and trailer pages.

text: <c-string 1..32>

The specified information (up to 32 characters) is stored in the SCB for the processing of exits. The first 8 characters are printed on the header page under the mailing box in large print.

Only letters, digits and selected special characters appear in large print. All other characters are replaced automatically in the header page by a printable character, namely the question mark (?).

Structure of the header page:

- 1. User ID in oversize print (10 lines + 2 blank lines)
- 2. Account number in oversize print (10 lines + 2 blank lines)
- 3. Job name in oversize print (10 lines + 2 blank lines)
- 4. Mailing box (address and identification field; 12 lines + 2 blank lines)

5. 'text' in oversize print (10 lines + 2 blank lines)

Order of priority

- 1. Mailing box (address and identification field; 12 lines + 2 blank lines)
- 2. 'text' in oversize print (10 lines + 2 blank lines)
- 3. Job name in oversize print (10 lines + 2 blank lines)
- 4. User ID in oversize print (10 lines + 2 blank lines)
- 5. Account number in oversize print (10 lines + 2 blank lines)

text: <var: char: 32> / (<reg: char:32>)

Name of a field defined with CL or a register containing the value. A string (32 bytes in length) is stored in this field or register and interpreted as information to be saved.

header-exit: *NO / <integer 0..2147483647> / <var: int: 4> / (<reg: int:4>)

Number of the desired header pages. The exact meaning of the operand is computer center-specific.

header-exit: <var: int: 4> / (<reg: int:4>)

Name of a field defined with FL or a register containing the value. An integer (4 bytes in length) is stored in this field or register and interpreted as the header page number.

trailer-exit: *NO / <integer 0..2147483647> / <var: int: 4> / (<reg: int:4>)

Number of the desired trailer pages. The exact meaning of the operand is computer center-specific.

trailer-exit: <var: int: 4> / (<reg: int:4>)

Name of a field defined with FL or a register containing the value. An integer (4 bytes in length) is stored in this field or register and interpreted as the trailer page number.

DELF=

Specifies whether the file is to be deleted at the end of output and, if so, whether its catalog entry and the data are to be overwritten with X'00..0'.

By default, the file is not deleted after printing and not overwritten with binary zeros. You must have write access to the file.

If the file to be output belongs to a file generation group, the DELF operand is ignored. If a spoolout job is used to print several elements of a PLAM library with SECTREC=..., the DELF operand is set to *NO (i.e. suppressed). The operand must not be specified together with *SYSLST, SYSLSTnn or *SYSOUT.

DELF=*NO

The file is not to be deleted after printing (exceptions: EAM and system files).

DELF=*YES

The file is to be deleted as soon as output has been completed.

DELF=*DESTROY

Not for EAM and cataloged system files.

Specifies that the catalog entry and data of the file be overwritten with binary zeros after the file is printed.

DELF=<var: enum-of delf set:1>

Deletion of files is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

1	*YES		
2	*DESTROY		
3	*NO		
See the parameter list description as of page 109.			

DNAME=(pointer, length)

Specifies the name of the cataloged file to be printed and the length of the file name.

pointer: *NONE / <var: pointer> / (<reg: pointer>)

Specifies the name of the cataloged file to be printed.

pointer: *NONE

A library element or system file is to be printed, not a cataloged file.

pointer: <var: pointer> / (<reg: pointer>)

A pointer is specified; in other words, the variable or field contains not the required value but the address of the memory location at which the value is stored (A(field) or a register).

length: *STD / <integer 1..1024> / <var: int: 2> / (<reg: int:2>)

Specifies the length of the file name.

length: *STD

The length of the file name is 54 characters.

length: <integer 1..1024>

Specifies an integer for the length of the file name.

length: <var: int: 2> / (<reg: int:2>)

Name of a field defined with FL or a register containing the value. An integer (2 bytes in length) is stored in this field or register and interpreted as the length of the file name.

Example

PRNTDOC DNAME=(A(VAR1),41)

The name of the file to be printed is stored in the field VAR1. 41 characters from this field are to be evaluated as the file name length.

PRNTDOC DNAME=(A(VAR1))

The name of the file to be printed is stored in the field VAR1. 54 characters from this field (default) are to be evaluated as the file name length.

PRNTDOC DNAME=(A(VAR1), VAR2)

The name of the file to be printed is stored in the field VAR1. The length of the file name is stored in the field VAR2.

PRNTDOC DNAME=(*NONE,*STD)

Default: no cataloged file is to be printed. The entry for the default length is ignored.

DOCFORM=

Specifies the type of the document contents, i.e. the format of the file to be printed as regards the interpretation of feed control characters, printer control characters, font identifiers, RENO commands and/or VTSU codes.

DOCFORM=*TEXT

Except for feed control characters, the file has no printer-specific control characters. The position of the feed control character in the record can be specified in the LINESP operand. The data is sent to the printer unchanged. Since only the data to be printed is sent to the printer, records can be truncated.

The following applies to HP and HP90 printers:

When the operand PRTYPE=*ANY is specified, the character X'FF' is replaced by the character X'1F' in order to be compatible with the PRM statement CONVERT-PRINT-RESOURCES.

The following applies to LP65 printers:

Spoolout jobs for which DOCFORM=*TEXT is specified can contain any LP65 control characters and printer control characters. Only records longer than 8192 characters are truncated.

The following applies to RSO:

Records of type A-1 or A-2 are to be printed (no control characters in the data stream); in other words, except for feed control characters in the first column of the records, no control characters are interpreted. This is also the reason why a font change is not possible. The form's default font or the first font specified for CHARSET, if specified, is used. Non-printable characters, i.e. characters with a hexadecimal value less than X'40', are output as blanks. If the record length exceeds the maximum line length, the record is truncated in the printout. The maximum line length depends on the character spacing, which is defined by means of the font used (see the CHARSET operand).

DOCFORM=*PAGE FORMAT

The file has laser printer-specific control characters. The CONTMOD operand allows you to specify how the control characters are to be interpreted.

DOCFORM=*SPECIAL_FORMAT

A printer-specific language is used in the document. The SPOOL and Distributed Print Services (Dprint) subsystems provide transparent control for the document; i.e. the file is transferred to an RSO or Xprint printer without interpretation. The value of the FRMNAME operand defines exactly which format the document contains. See the FRMNAME operand on page 63.

DOCFORM=<var: enum-of docform_set:1>

The type of the document contents is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*TEXT	
1	*PAGE_FORMAT	
2	*SPECIAL_FORMAT	
See the parameter list description as of page 109.		

DPPAR=

The address of a parameter list can be specified in which Dprint-specific values are stored for the macro call.

DPPAR=*NONE

The values are not to be supplied via a parameter list.

DPPAR=<var: pointer> / (<reg: pointer>)

A pointer is specified; in other words, the variable or field does not contain the Dprint parameter list itself but the address of a memory location at which the Dprint parameter list is stored (A(field) or a register).

DSEMPAR=

The address of a parameter list can be specified in which DSEM-specific values are stored for the macro call.

DSEMPAR=*NONE

The values are not to be supplied via a parameter list.

DSEMPAR=<var: pointer> / (<reg: pointer>)

A pointer is specified; in other words, the variable or field does not contain the DSEM parameter list itself but the address of a memory location at which the DSEM parameter list is stored (A(field) or a register).

DTYPE=

Type of the files to be output.

DTYPE=*FILE

A cataloged file is to be printed.

DTYPE=*LIBRARY ELEMENT

A PLAM library element is to be printed.

DTYPE=*POSIX

A file from a POSIX file system is to be printed.

DTYPE=*EAM

A temporary object module file is to be printed.

DTYPE=*SYSTEM FILE

A system file (SYSLST or SYSOUT) is to be printed.

DTYPE=<var: enum-of dtype_set:1>

The type of the files to be output is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

1	*FILE	
2	*LIBRARY_ELEMENT	
3	*POSIX	
4	*EAM	
5	*SYSTEM_FILE	
See the parameter list description as of page 109.		

EAMN=array(16): *STD / <integer 0..65535> / <var: int: 4> / (<reg: int:4>)

Specifies the numbers of the EAM object module files to be output. Up to 16 numbers can be output.

You will find a detailed description of EAM files in the manual "Introductory Guide to DMS".

EAMN=*STD

The temporary object module file of the current job is to be output.

EAMN=<integer 1..65535>

The number of the EAM file to be output is specified directly by means of an integer.

EAMN=<var: int: 4> / (<reg: int:4>)

Name of a field defined with FL or a register containing the value. An integer (4 bytes in length) is stored in this field or register and interpreted as the number of the EAM file.

EFO=

Specifies whether a film overlay - in the case of HP90 and 3365 printers an EFO data overlay - is to be used when a spoolout job is processed.

EFO=*NONE

No film overlay (HP90 and 3365: no EFO data overlay) is used for output.

EFO=<c-string 2..2>

Name of the film overlay (HP90 and 3365: EFO data overlay) to be used for processing the spoolout job (the name must be confirmed by systems support).

EFO=<var: char: 2> / (<reg: char:2>)

Name of a field defined with CL or a register containing the value. A string (2 bytes in length) is stored in this field or register and interpreted as the name of the film overlay.

FAMILY=

Specifies whether a common TSN is allocated if a number of file or library elements are specified in a PRNTDOC macro (in order to ensure that these files are output sequentially on the same printer).

FAMILY=*STD

For spoolout jobs to local printers and RSO devices, the default value from the SPOOL parameter file applies. This can be defined separately (and hence differently) for local and RSO printers in the SPSERVE statement MODIFY-SPOOL-PARAMETERS. You can display the value by means of the SHOW-SPOOL-PARAMETERS command or SPSERVE statement (FAMILY-PROCESS field).

FAMILY=*YES

A common TSN (FAMILY-PRINT) is to be allocated if a number of files or library elements are specified in a spoolout job. The individual files or library elements cannot be processed in parallel.

FAMILY=*NO

The files or library elements specified at the same time in a PRNTDOC macro are to be processed under separate TSNs. This means that parallel processing is possible.

FAMILY=<var: enum-of family_set:1>

The assignment of a common TSN is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

1	*STD	
2	*YES	
3	*NO	
See the parameter list description as of page 109.		

FIRSREC=(begin, ident, length, position, occur)

Specifies whether only a certain number of the input file's records are to be processed. The first and last record can be specified by means of a record number, section number or string. The settings for the last record can be made via the LASTREC operand.

"begin" specifies the record as of which the file is to be processed.

"ident" specifies as the first record a record in which the specified string occurs.

"length" specifies the length of the record.

"position" specifies the position in the record at which the specified string begins.

"occur" specifies the occurrence of the record containing the string as of which output is to begin.

begin: *BEGIN_OF_FILE / *BY_STRING_ID / <integer 1..2147483647> /

<var: int: 4> / (<reg: int:4>)

Specifies the record as of which the file is to be processed. You can specify the first record in the file, the number of a record or file mark, or a string in a record.

begin: *BEGIN OF FILE

Output begins with the file's first record, even if SECTION records are specified.

begin: *BY_STRING_ID

The record in which a specified string occurs is the first record to be output.

begin: <integer 1..2147483647>

The first record is specified as an integer from 1 to 2147483647.

begin: <var: int: 4> / (<reg: int:4>)

Name of a field defined with FL or a register containing the value. An integer (4 bytes in length) is stored in this field or register and interpreted as the number of the first record.

ident: *DUMMY / <c-string 1..60> / <var: char: 60> / (<reg: char:60>)

Specifies whether output is to begin with a record in which a specific string of printable or hexadecimal characters is found.

ident: *DUMMY

No string of printable or hexadecimal characters is specified.

ident: <c-string 1..60>

Output is to begin with the record in the file in which the specified string of printable or hexadecimal characters is found.

ident: <var: char: 60> / (<reg: char:60>)

Name of a field defined with CL or a register containing the value. A string (60 bytes in length) is stored in this field or register and interpreted as a string.

length: 0 / <integer 1..60> / <var: int: 1> / (<reg: int:1>)

Specifies the length of the record.

length: 0

Default: the record has a length of 0 bytes.

length: <integer 1..60>

The length of the record is specified as an integer from 1 to 60.

length: <var: int: 1> / (<reg: int:1>)

Name of a field defined with FL or a register containing the value. An integer (1 byte in length) is stored in this field or register and interpreted as the length of the record.

position: *STD / <integer 1...2047> / <var: int: 2> / (<reg: int:2>)

Specifies the position as of which the specified string begins in the SECTION record (as of which byte after the record length field).

position: *STD

The string searched for begins by default at the beginning of the record:

- in the case of a SAM file: with the first byte after the record length field
- in the case of an ISAM file with KEY-POS=5: with the first byte after the key
- in the case of an ISAM file with KEY-POS > 5: with the first byte after the record length field

position: <integer 1..2047>

Specifies an integer for the position of the string in the SECTION record.

position: <var: int: 2> / (<reg: int:2>)

Name of a field defined with FL or a register containing the value. An integer (2 bytes in length) is stored in this field or register and interpreted as the position of the record.

occur: 1 / <integer 1..32767> / <var: int: 2> / (<reg: int:2>)

Specifies the occurrence of the record containing the string as of which output is to begin.

occur: 1

Output is to begin as of the first occurrence of the record.

occur: <integer 1..32767>

Integer specifying the number of the occurrence of the record.

occur: <var: int: 2> / (<reg: int:2>)

Name of a field defined with FL or a register containing the value. An integer (2 bytes in length) is stored in this field or register and interpreted as the number of the occurrence.

FOB=

Specifies whether an FOB data overlay is to be used for processing the spoolout job.

FOB=*NONE

No FOB data overlay is used for output.

FOB='<alphanum-name 1..4>' / <c-string 1..4>

Name of the overlay to be used for processing the spoolout job. If an overlay is specified, the file is printed out on an HP or HP90 printer. Use of an overlay for the spoolout job is indicated in the output of the command SHOW-PRINT-JOB-STATUS JOB-IDENTIFICATION=TSN(TSN=...).

FOB=<var: char: 4> / (<reg: char:4>)

Name of a field defined with CL or a register containing the value. A string (4 bytes in length) is stored in this field or register and interpreted as the name of the overlay.

FORM=

Specifies the paper (form) to be used for output (e.g. STD, STDSF1, STDWA4). Default forms must be defined in the SPOOL parameter file for all printer types. By means of SHOW-SPOOL-FORMS you can output the entries to SYSOUT. The SPOOL parameter file also specifies whether header and trailer pages are to be printed.

FORM=*STD

Default form.

FORM='<alphanum-name 1..6>'

Name of the form with which the spoolout job is to be processed. A loop (or a page and format definition for APA printers) is implicitly named when the form is specified. The associated loop (or the PAGEDEF and FORMDEF) must be in a specific printer control file. The following table shows which printer control file must contain this loop (or the PAGEDEF and FORMDEF) for the relevant printer type:

Output device	Printer control file with the loop for the specified form
Printer 3337, 3338, 3339, 3348, 3349, 3365, LP-EMULATED	\$SYSSPOOL.PRFILE
Printer: 3351, 3353, 2090, 2140, 2240	\$SYSSPOOL.PRFILE or the user PRFILE specified in the USERRES operand
APA printer	\$SYSSPOOL.SYSPRT.SPS.021 or the user SPSLIB specified in the USERRES operand

The loop named implicitly via the FORM operand is ignored if the LOOP operand is specified at the same time. If the FORM and LOOP operands are omitted, the file is printed out using the default form entered for the printer type.

A loop explicitly specified in the LOOP operand must have the same length as the loop record assigned to the form used.

No loops can be specified for APA printers. If page and format specifications are made in the FORM operand, these are used for printing header, trailer and message pages.

FORM=<var: char: 6> / (<reg: char:6>)

Name of a field defined with CL or a register containing the value. A string (6 bytes in length) is stored in this field or register and interpreted as the name of the form.

FRMNAME=<u>*STD</u> / <c-string: c-string 1..63> / <var: char: 63> / (<reg: char:63>)

Name of the format to be processed. The document is processed in transparent mode. Name of the format in which the data to be output is passed to the printer. If no value is specified for FRMNAME, a default value is automatically derived from the value of the CONTMODE operand. This results in the following:

CONTMODE=*PAGE-MODE(...) FRMNAME=*HP

CONTMODE=*APA(...) FRMNAME=*SPDS

CONTMODE=*LOGICAL / *PHYSICAL / *LINEMODE FRMNAME=*STD

The value FRMNAME=*STD means that the contents of the file to be printed are irrelevant.

FRMNAME=*STD

Operand default value.

FRMNAME=<c-string: c-string 1..63>

The name of the format is specified directly.

FRMNAME=<var: char: 63>

The name of the format is specified indirectly via a field.

FRMNAME=(<reg: char:63>)

A register points to a field containing the name of the format.

HEADLIN=

Specifies whether a header line is to be printed on every page (except the header and trailer pages).

HEADLIN=*NO

No header line is printed.

HEADLIN=*STD

The header has the following format:

Header	DATE yyyy-mm-dd	userid	file	PAGE nnnn
Column	1	41(11,11)	60(21,67)	124 (77)

The first value in parentheses applies in the case of a form definition with a line length < 132 and ≥ 80 characters (with differing values). The second value in parentheses applies in the case of a form definition with a line length < 80 and ≥ 70 characters (with differing values)..

jj-yy-mm-dd year-month-day

userid user identification

file file name

nnnn page number

Unless otherwise specified, the header line is followed by a blank line. The header is shifted in accordance with the LEFMAR operand, but the header line will be truncated at column 132. DATE and PAGE are only present in the header line if the line size ≥ 032.

HEADLIN=*DATE / *FRECORD / *PAGE / *DATE_FRECORD / *DATE_PAGE / *FRECORD PAGE / *DATE FRECORD PAGE

The header has the following format:

HEADLIN =	*DATE	*FRECORD	*PAGE
Header	DATEyyyy-mm-dd (yyyy-mm-dd)		PAGE nnnn (nnnn, nnnn)
Column	1	21	124 (77,67)

*DATE:

The value in parentheses applies in the case of a form definition with a line length < 132 and ≥ 70 characters (with differing values).

*PAGE:

The first value in parentheses applies in the case of a form definition with a line length < 132 and ≥ 80 characters (with differing values). The second value in parentheses applies in the case of a form definition with a line length < 80 and ≥ 70 characters (with differing values).

yyyy-mm-dd year-month-day

First record first logical record of the file

nnnn page number

If *DATE, *FRECORD or *PAGE is omitted, the appropriate section is filled with blanks. Unless otherwise specified, the header line is followed by a blank line.

The header is shifted in accordance with the value in the LEFTMAR operand, but the header line will be truncated at column 132. *DATE and *PAGE are only present in the header line if the line size ≥ 032. If *FRECORD is specified, the first record is regarded as not belonging to the date. The entries can be linked by an underscore (_).

HEADLIN=<var: enum-of headline set:1>

The header is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

1	*NO		
2	*STD		
4	*FRECORD		
8	*DATE		
12	*DATE_FRECORD		
16	*PAGE		
20	*FRECORD_PAGE		
24	*DATE_PAGE		
28	*DATE_FRECORD_PAGE		
See the parameter list description as of page 109.			

INTRAY=

Specifies the paper input tray for LP65 printers, PCL printers, APA printers and for the RSO printers 2030-PCL, 4011, 4812, 4813, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL, 4830-PCL, 4850-PCL, 9000-EPLQ, 9000-EPSQ, 9000-PCL, 9004, 9011, 9012, 9013, 9014, 9015, 9021, 9022, 9022-200, 9025, 9026-PCL, 9026-RENO, 9097and DJET. Tray numbers 1 - 3 and can be specified for printer type LP65 and tray numbers 1 - 99 for RSO printers.

INTRAY=*STD

If the print job is addressed to the local SPOOL or to a special printer pool, this operand is ignored. But if the print job is addressed to a specific RSO printer, for instance, the paper is taken from the paper input tray defined in the device record of the LP65 printer in the SPOOL parameter file. The input tray selection is valid for the entire spoolout process, i.e. including processing of the header and trailer pages.

INTRAY=*IGNORE

Only for RSO.

The paper input tray is selected by means of a PCL file or an entry at the printer control console. The order of priority for selecting an input tray is as follows:

- 1. the number of the input tray specified in PRNTDOC
- 2. any selection via the PCL file specified in the PRNTDOC macro
- 3. any selection via the PCL file specified in the form definition in the SPOOL parameter file
- 4. the default values defined in the device record

INTRAY=<integer 1..99>

Number of the input tray from which the paper is taken for printing the file itself as well as the header and trailer pages.

The specified value is not checked by RSO. Possible values: 1...99.

The following table shows the printer type and the maximum number of input trays for this printer type (column "max."):

printer	max.	printer	max.	printer	max.	printer	max.
2030-PCL	2	8121	0	9002	0	9022	2
4011	2	9000	0	9003	0	9022-200	2
4812	2	9000-PCL	2	9004	3	9025	2
4813	2	9000-PRO	0	9011	2	9026-PCL	4
4818-PCL	2	9000-PS	0	9012	2	9026-RENO	4
4821-PCL	2	9000-EPFX	0	9013	3	9045-ANSI	0
4822-PCL	3	9000-EPLQ	2	9014	3	9046	0
4824-PCL	2	9000-EPSQ	2	9015	2	9645	0
4825-PCL	3	9001	0	9021	2	DJET	1
4830-PCL	3	9001-31	0		+	н	1
4850-PCL	2	PCL printer	3				

INTRAY=<var: int: 1>

Name of a field defined with FL or a register containing the value. An integer (1 byte in length) is stored in this field or register and interpreted as the number of the paper input tray.

INTRAY=*BY-FORMAT

The paper input tray is specified via a keyword for the format in the INTRAYF operand.

INTRAYF=<var: enum-of_intray_format_set: 1> / *MANUAL / *A3 / *A4 / *A5 / *B4 / *B5 / *DOUBLE-LETTER / *EXEC / *FOLIO / *INVOICE / *LEGAL / *LETTER / *MONARCH / *COMMERCIAL_10 / *DL / *C5 / *A3_UNCUT / *A4_UNCUT / *LEDGER Specifies the paper input tray via a keyword for the format.

To make sure that the paper size and line length values specified in the INTRAYF operand do not exceed the maximum values permitted for the selected paper format, the values of the INTRAYF operand are compared with the maximum values permitted for the paper format concerned. The maximum permitted values for the different paper formats are listed in the table below.

Paper format	Maximum value for paper size	Maximum value for line length
A3	165	116
A4	116	82
A5	82	58
B4	143	101
B5	101	71
FOLIO	129	85
INVOICE	85	55
EXEC	105	72
LEGAL	140	85
LETTER	110	85
DOUBLE-LETTER	150	117
MONARCH	75	38
COMMERCIAL-10	95	41
DL	86	43
C5	90	63
A3-UNCUT	120	120
A4-UNCUT	169	84
LEDGER	170	110

INTRAYF=*MANUAL

The paper is fed in manually and you should therefore insert a page whenever a new page is to be printed or when the printer requests you to.

Manual paper input feed is supported by the following printers: 9014, 9021, 9022, 9022-200, 9026-Reno, 9026-PCL, 4812, DJET, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL and 9000-PCL.

INTRAYF=*A3

The pages are taken from the first input tray which contains A3 pages. The value *A3 can be used for printers 4830-PCL, 9026-PCL and 9026-RENO.

INTRAYF=*A4

The pages are taken from the first input tray which contains A4 pages. The value *A4 can be used for printers 9021, 9000-PCL, 9026-PCL, 9026-RENO, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL and 4830-PCL.

INTRAYF=*A5

The pages are taken from the first input tray which contains A5 pages. The value *A5 can be used for printers 4830-PCL, 9026-PCL and 9026-RENO.

INTRAYF=*B4

The pages are taken from the first input tray which contains B4 pages. The value *B4 can be used for the printers 4830-PCL and 99026-RENO.

INTRAYF=*B5

The pages are taken from the first input tray which contains B5 pages. The value *B5 can be used for printers 4830-PCL, 9026-PCL and 9026-RENO.

INTRAYF=*FOLIO

The pages are taken from the first input tray which contains FOLIO pages. The value *FOLIO can be used for the printer 9026-RENO.

INTRAYF=*INVOICE

The pages are taken from the first input tray which contains INVOICE pages. The value *INVOICE can be used for printers 9026-PCL and 9026-RENO.

INTRAYF=*EXEC

The pages are taken from the first input tray which contains EXEC pages. The value *EXEC can be used for printers 9021, 9000-PCL, 9026-PCL, 9026-RENO, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL and 4830-PCL.

INTRAYF=*LEGAL

The pages are taken from the first input tray which contains LEGAL pages. The value *LEGAL can be used for printers 9021, 9000-PCL, 9026-PCL, 9026-RENO, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL and 4830-PCL.

INTRAYF=*LETTER

The pages are taken from the first input tray which contains LETTER pages. The value *LETTER can be used for printers 9021, 9000-PCL, 9026-PCL, 9026-RENO, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL and 4830-PCL.

INTRAYF=*DOUBLE-LETTER

The pages are taken from the first input tray which contains DOUBLE-LETTER pages. The value *DOUBLE-LETTER can be used for the printer 9026-RENO.

INTRAYF=*MONARCH

The pages are taken from the first input tray which contains MONARCH pages. The value *MONARCH can be used for printers 9021, 9000-PCL, 9026-PCL, 9026-RENO, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL and 4830-PCL.

INTRAYF=*COMMERCIAL-10

The pages are taken from the first input tray which contains COMMERCIAL-10 pages. The value *COMMERCIAL-10 can be used for printers 9021, 9000-PCL, 9026-PCL, 9026-RENO, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL and 4825-PCL.

INTRAYF=*DL

The pages are taken from the first input tray which contains International-DL size envelopes.

The value *DL can be used for printers 9021, 9000-PCL, 9026-PCL, 9026-RENO, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL and 4830-PCL.

INTRAYF=*C5

The pages are taken from the first input tray which contains International-C5 size envelopes.

The value *C5 can be used for printers 9021, 9000-PCL, 9026-PCL, 9026-RENO, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL and 4830-PCL.

INTRAYF=*A3_UNCUT

The pages are taken from the first input tray which contains uncut A3 pages. The value *A3_UNCUT can only be used for PCL printers.

INTRAYF=*A4 UNCUT

The pages are taken from the first input tray which contains uncut A4 pages. The value *A4_UNCUT can only be used for PCL printers.

INTRAYF=*LEDGER

The pages are taken from the first input tray which contains uncut LEDGER pages. The value *LEDGER can only be used for PCL printers.

INTRAYF=<var: enum-of _intray_format_set:1>

The paper input tray is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

128	MANUAL
129	A3
130	A4
131	A5
132	B4
133	B5
134	DOUBLE-LETTER
135	EXEC
136	FOLIO
137	INVOICE
138	LEGAL
139	LETTER
140	MONARCH

141	COMMERCIAL-10
142	DL
143	C5
144	A3_UNCUT
145	A4_UNCUT
146	LEDGER

These values are defined in the ASSEMBLER, C and COBOL interface. For example, the following is in the C interface:

```
/* intray_format_set
                                                                            */
/* ENUM _intray_format_set
                                                                            */
<< The input tray is not defined explicitly, but rather via the>>
<< following constant definition>>
<< (the integer value stored in the constant can therefore be used>>
<< (in the parameter list of the function call):>>
#define PRNTDOCintray_format_manual 128
                                               /* intray_format = *manual
                                                                            */
#define PRNTDOCintray_format_a3 129
                                               /* intray format = *a3
                                                                            */
#define PRNTDOCintray_format_a4 130
                                               /* intray_format = *a4
                                                                            */
#define PRNTDOCintray_format_a5 131
                                               /* intray_format = *a5
                                                                            */
#define PRNTDOCintray_format_b4 132
                                               /* intray format = *b4
                                                                            */
#define PRNTDOCintray_format_b5 133
                                                /* intray format = *b5
                                                                            */
#define PRNTDOCintray_format_double-letter 134
/* intray_format = *double-letter*/
                                                                            */
#define PRNTDOCintray_format_exec 135
                                                /* intray_format = *exec
#define PRNTDOCintray_format_folio 136
                                                /* intray_format = *folio
                                                                            */
#define PRNTDOCintray format invoice 137
                                                /* intray format = *invoicel*/
                                                /* intray_format = *legal
#define PRNTDOCintray_format_legal 138
#define PRNTDOCintray format letter 139
                                                /* intray format = *letter
                                                                            */
#define PRNTDOCintray format monarch 140
                                                /* intray format = *monarch */
#define PRNTDOCintray_format_commercial-10 141
/* intray format = *commercial-10 */
                                                                            */
#define PRNTDOCintray format dl 142
                                               /* intray format = *dl
#define PRNTDOCintray_format_c5 143
                                               /* intray_format = *c5
                                                                            */
#define PRNTDOCintray_format_a3_uncut 144
                                               /* intray format = *a3 uncut */
#define PRNTDOCintray format a4 uncut 145
                                               /* intray format = *a4 uncut */
#define PRNTDOCintray_format_ledger 146
                                              /* intray format = *ledger
```

JVPASSW=

Specifies the password by which the monitoring job variable is protected (see the MONJV operand).

JVPASSW=*NONE

The job variable is not protected by a password.

JVPASSW=<c-string 1..4 with-low>

Password by which the job variable is protected.

JVPASSW=<var: char: 4> / (<reg: char:4>)

Name of a field defined with CL or a register containing the value. A string (4 bytes in length) is stored in this field or register and interpreted as a password.

LASTREC=(begin, ident, length, position, occur)

Specifies the last of the file's records to be processed. The last record can be specified by means of a record number, section number or string.

"begin" specifies the record at which processing of the file terminates.

"ident" specifies as the last record a record in which the specified string occurs.

"length" specifies the length of the record.

"position" specifies the position in the record at which the specified string begins.

"occur" specifies the occurrence of the record containing the string at which output is to terminate.

begin: *END_OF_FILE / *BY_STRING_ID / <integer 1..2147483647> /

<var: int: 4> / (<reg: int:4>)

Specifies the last record in the part of the file to be output. You can specify the last record in the file, the number of a record or file mark, or any string in a record.

begin: *END OF FILE

Output terminates with the last record in the file, even if SECTION records are specified.

begin: *BY_STRING_ID

Specifies that the last record is to contain a specified string.

begin: <integer 1..2147483647>

The last record is specified as an integer from 1 to 2147483647.

begin: <var: int: 4> / (<reg: int:4>)

Name of a field defined with FL or a register containing the value. An integer (4 bytes in length) is stored in this field or register and interpreted as the number of the last record.

ident: *DUMMY / <c-string 1..60> / <var: char: 60> / (<reg: char:60>)

Specifies whether output is to end with a record containing a specified string of printable or hexadecimal characters at a specified position in the file.

ident: *DUMMY

No string of printable or hexadecimal characters is defined.

ident: <c-string 1..60>

Output is to end with the record containing the specified string of printable or hexadecimal characters at a specified position in the file.

ident: <var: char: 60> / (<reg: char:60>)

Name of a field defined with CL or a register containing the value. A string (60 bytes in length) is stored in this field or register and interpreted as a string.

length: <u>0</u> / <integer 1..60> / <var: int: 1> / (<reg: int:1>)

Specifies the length of the record.

length: 0

Default: the record has a length of 0 bytes.

length: <integer 1..60>

The length of the record is specified as an integer from 1 to 60.

length: <var: int: 1> / (<req: int:1>)

Name of a field defined with FL or a register containing the value. An integer (1 byte in length) is stored in this field or register and interpreted as the length of the record.

position: *STD / <integer 1..2047> / <var: int: 2> / (<reg: int:2>)

Specifies as of which position (as of which byte after the record length field) the specified string begins in the SECTION record.

position: *STD

The string searched for begins by default at the beginning of the record:

- in the case of a SAM file: with the first byte after the record length field
- in the case of an ISAM file with KEY-POS=5: with the first byte after the key
- in the case of an ISAM file with KEY-POS > 5: with the first byte after the record length field

position: <integer 1..2047>

Specifies an integer for the position of the string in the SECTION record.

position: <var: int:2> / (<reg: int:2>)

Name of a field defined with FL or a register containing the value. An integer (2 bytes in length) is stored in this field or register and interpreted as the position of the record.

occur: 1 / <integer 1..32767> / <var: int: 2> / (<reg: int:2>)

Specifies the occurrence of the record containing the LASTREC string as of which output is to terminate.

occur: 1

Output is to terminate at the first occurrence of the record.

occur: <integer 1..32767>

Integer specifying the occurrence of the record.

occur: <var: int:2> / (<reg: int:2>)

Name of a field defined with FL or a register containing the value. An integer (2 bytes in length) is stored in this field or register and interpreted as the number of the occurrence.

LEFTMAR=

Specifies whether the output text is to be indented. The LEFTMAR operand is ignored if CONTMOD=*PHYSICAL is specified at the same time.

LEFTMAR=*STD

The default is stored in the SPOOL parameter file and can be queried by means of the SHOW-SPOOL-PARAMETERS command or SPSERVE statement; the information is taken from the output field PRINT-CMD-DEFAULTS:...LEFT-MARGIN =

LEFTMAR=<integer 0..31>

For all printers except RSO printers:

The output text is to be indented by the specified number of columns.

For all RSO printers:

The output text is to be indented by the specified number * 1/10 inches.

LEFTMAR=<var: int: 1> / (<reg: int:1>)

Name of a field defined with FL or a register containing the value. An integer (1 byte in length) is stored in this field or register and interpreted as the number of columns.

LEFTOFF=*IGNORE / <integer -255..255>

Specifies the offset in millimeters of the left edge of the print page from the left edge of the physical page. The print page is first positioned on the paper and then the text within the print page is rotated and positioned. This means that the text orientation within the print page is not taken into account if the print page is shifted with respect to the paper page.

LEFTOFF=*IGNORE

Operand default value. The print page is not offset horizontally on the paper.

LEFTOFF=<integer -255..255>

The print page is offset horizontally by the specified value on the paper.

LIBELEM=(element, version, type)

Specifies the name, version and type of a PLAM library element to be printed. An element is fully defined by its name, type and version number. The records of an element are assigned to specific record types. There are 255 record types. A distinction is drawn between user record types (1 to 159) and special record types (160 to 255). Only an element's user record types can be printed with SPOOL.

element: *DUMMY / '<filename 1..80 with-wild>' / <var: char: 80> /

(<reg: char:80>)

Specifies the name of the element.

element: *DUMMY

No library element is to be printed.

element: '<filename 1..80 with-wild>'

The name of the element is specified as a string in single quotes.

element: <var: char: 80> / (<reg: char:80>)

Name of a field defined with CL or a register containing the value. A string (80 bytes in length) is stored in this field or register and interpreted as the name of the element.

version: *HIGHEST / '<filename 1..40 with-wild>' / <var: char: 40> /

(<reg: char:40>)

Version number of the element to be output.

version: *HIGHEST

The element with the alphanumerically highest version number is to be printed.

version: '<filename 1..40 with-wild>'

The version number of the element is specified as a string in single quotes.

version: <var: char: 40> / (<reg: char: 40>)

Name of a field defined with CL or a register containing the value. A string (40 bytes in length) is stored in this field or register and interpreted as the version number of the element.

type: '<alphanum-name 1..12 with-wild>' / <var: char: 12> / (<reg: char:12>)

Type of the library element to be output. The records of an LMS element of type C or R belong to special record types (160 to 255), so no special records can be printed from these elements (type C or R).

type: '<alphanum-name 1..12 with-wild>'

The type of the library element to be output is specified as a string in single quotes.

type: <var: char: 12> / (<reg: char:12>)

Name of a field defined with CL or a register containing the value. A string (12 bytes in length) is stored in this field or register and interpreted as the type of the library element to be output.

Example

PRNTDOC LIBELEM=('LMSLIB-ELEM1',*HIGHEST,'A')

The element called "LMSLIB-ELEM1" is to be printed. The last element in alphanumeric order is to be searched for. The element is of the type A.

LINEPP=

Specifies how many lines (including the header and blank lines) are to be printed on a page.

LINEPP=*STD

If the operand is omitted, the number of lines per print page is calculated using the following formula, regardless of what has been specified for the HEADLIN operand:

Number of lines = P * L - N - 6

Where:

P = paper size in inches

L = line density

N = number of lines before the first vertical tab "channel 1".

Printers with a loadable vertical format buffer

- The vertical tab "channel 1" controls the line on which printing is to start. Unless otherwise specified, 2 blank lines are set before printing starts; i.e. channel 1 (CHANNEL 01) is in the third line of the loop.
- If the value specified for the LINEPP operand is greater than the specified number of lines in the loop, the value in the loop is used.
- If the LINEPP operand is specified together with the HEADLIN and LINESP operands, a value specified here must be at least three times that of the line feed specified for LINESP = *SPACE_1 / *SPACE_2 / *SPACE_3.

LINEPP=<integer 0..32767>

Number of lines on a page.

LINEPP=<var: int: 2> / (<reg: int:2>)

Name of a field defined with FL or a register containing the value. An integer (2 bytes in length) is stored in this field or register and interpreted as the number of lines on a page.

LINESP=(spacing, position)

Specifies the number of line feeds and how control characters are interpreted.

spacing: *STD / *SPACE_1 / *SPACE_2 / *SPACE_3 /

*BY-ASA-CONTROL / *BY-EBCDIC-CONTROL / *BY-IBM-CONTROL / *NO

The following can be specified:

The records are to be printed with 1-, 2- or 3-line spacing (*SPACE_n).

The contents of the first byte of each record are to be interpreted as ASA feed control characters (*BY_ASA_CONTROL).

The contents of the first byte of each record are to be interpreted as EBCDIC feed control characters (*BY_EBCDIC_CONTROL).

The contents of the first byte of each record are to be interpreted as IBM feed control characters (*BY_IBM_CONTROL)

The contents of the first byte of each record are not to be interpreted as feed control characters (*NO).

The default (*STD) depends on the character control interpretation mode (CONTMOD operand).

spacing: <var: enum-of space set:1>

The number of line feeds is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*STD
1	*SPACE_1
2	*SPACE_2
4	*SPACE_3
8	*BY_EBCDIC_CONTROL
16	*BY_ASA_CONTROL
32	*BY_IBM_CONTROL
144	*NO
See the parameter list description as of page 109.	

position: *STD / <integer 1..2040> / <var: int: 2> / (<reg: int:2>)

Number of the data byte in which SPOOL finds the feed control character. In the case of records of variable length, the fields containing the length are not counted as part of the data.

position: *STD

The default depends on the control character interpretation mode (CONTMOD operand).

position: <integer 1..2040>

Integer specifying the number of the data byte.

position: <var: int:2> / (<reg: int:2>)

Name of a field defined with FL or a register containing the value. An integer (2 bytes in length) is stored in this field or register and interpreted as the number of the data byte.

LOCKF=

Specifies whether the file is to be protected as long as the spoolout job is in the wait state (TYPE 4, see output of the SHOW-PRINT-JOB-STATUS command). During this time the file can only be read. As a rule, tape files are never locked. During processing of the spoolout job (TYPE 5, see output of the SHOW-PRINT-JOB-STATUS command), the file is protected irrespective of the setting of this operand.

A spoolout job is also generated if the file to be output is reserved by a SECURE-

RESOURCE-ALLOCATION command. This reservation must, however, be canceled by the time the spoolout job is processed; otherwise the job is not executed.

The file to be output is locked until the end of the session if the operand LOCKF=*YES is specified in the PRNTDOC macro and the job cannot be executed owing to reservation.

LOCKF=*STD

The value defined at system generation time is valid (NO or YES); this value is entered in the SPOOL parameter file.

LOCKF=*YES

The file is protected while the spoolout job is in the wait state. LOCKF=*YES is ignored if *EAM or *SYSTEM_FILE is specified in the DTYPE operand at the same time.

A PRNTDOC macro for a library element is rejected if LOCKF=*YES is specified at the same time. File protection offered by LOCKF=*YES remains in force even if the spoolout job is not processed until the next system run.

LOCKF=*NO

The file is not protected while the spoolout job is in the wait state. The file can be deleted or modified before processing of the spoolout job commences.

LOCKF=*NO is ignored for temporary files.

LOCKF=<var: enum-of lockf_set:1>

File protection is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*STD
1	*NO
2	*YES
See the parameter list description as of page 109.	

LOOP=

Name of the loop to be loaded into the line feed information buffer (VFB).

The loop name must not contain the '\$', '&' or '@' characters.

LOOP=*STD

Feed control for the spoolout job is to be implemented with the default loop of the form used.

LOOP='<alphanum-name 1..3>'

Name of the loop which is to control line feed. The length of the specified loop must match the length of the default loop of the form used. A loop for feed control is needed for the HP and HP90 printers (3351, 3353, 2090, 2140) and printers of types 3337, 3338, 3339, 3348, 3349 and 3365. Loops are stored in the PRFILE printer control file. If no loop is specified, the implicit entries in the FORM operand are used. If the FORM or LOOP operand is omitted, default values apply.

LOOP=<var: char: 3> / (<reg: char:3>)

Name of a field defined with CL or a register containing the value. A string (3 bytes in length) is stored in this field or register and interpreted as the name of the loop.

MACID=CPA / '<name 1..3>'

Specifies the second to fourth characters (inclusive) of the field names and equates.

MF=C/D/E/L/M

Type of the macro call. You will find more information in the "Executive Macros" manual.

Note

You can only specify pointer variables ("var:pointer" operand value) with MF=M. In addition, a list must be generated with MF=D or MF=C.

MONJV=

Specifies the job variable in which information on job processing is to be stored.

MONJV=*NONE

No job variable is to be linked to the job.

MONJV=*STD

The job variable should be given the name of the file to be printed (without catalog ID and without user ID). It is created under the user ID and catalog ID of the caller. The call PRNTDOC MONJV=*STD is rejected if:

- a file generation is specified
- a temporary file is specified
- an OMF or EAM file is specified
- a PLAM element is specified.

The call PRNTDOC (XX,XX),MONJV=*STD is rejected. The reason for this is that if the job variable XX has been created for the first job, it is no longer available for a second job.

MONJV='<filename 1..54 with-wild>'

The job variable is to contain the name specified as a string. If the job contains several files, the following suffix is added to the name of the job variable: a consecutive number <1..9999> if FAMILY=*YES is also specified.

MONJV=<var: char: 54> / (<reg: char:54>)

Name of a field defined with CL or a register containing the value. A string (54 bytes in length) is stored in this field or register and interpreted as the name of the job variable.

NOTIFPAR=<var: pointer> / *NONE

This operand allows to anchor a specific notification area described by the macro SNPPRNT.

OUTFORM=

Specifies whether output is to be in character format only or in hexadecimal format as well.

OUTFORM=*CHARACTER

Specifies that the records be output in character format only. Records that exceed the print line length are truncated.

OUTFORM=*HEXADECIMAL

Specifies that the records be output in character format and hexadecimal format.

Output format

At the beginning of each output line there is a prefix of 8 bytes followed by 50 bytes of data. Each output line is printed first in the specified font and then in hexadecimal format.

Structure of the output line

Column	Contents
1-4	Column number as of which the data of the output record begins
5-8	Blanks
As of 9	Characters of the input record in the specified font. The characters are separated by a blank in each case. They are repeated in hexadecimal format in the next line.

OUTFORM=<var: enum-of outform set:1>

Character output is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

1	*CHARACTER
2	*HEXADECIMAL
See the parameter list description as of page 109.	

OUTPART=(begin, end, dimension)

Specifies that the whole input file is to be edited for printing but that only a subset of all the logical pages is to be output.

begin: *BEGIN OF FILE / <integer -2147483647..2147483647> /

<var: int: 4> / (<reg: int:4>)

Allows you to specify a page or line number in the print file at which output is to begin. The *dimension* operand specifies whether it is a page or line number.

begin: *BEGIN OF FILE

Output begins by default at the beginning of the file. The pages of a print file are specified as described for the LINEPP operand (unless control characters cause a page feed to be executed prematurely).

begin: <integer -2147483647..2147483647>

Integer specifying a page or line number in the print file. If a value < 0 is specified, the number of pages or lines to be output is counted from the end of the file. The DIMENSION operand specifies whether the integer is to be interpreted as a page or line number.

begin: <var: int:4> / (<reg: int:4>)

Name of a field defined with FL or a register containing the value. An integer (4 bytes in length) is stored in this field or register and interpreted as a page or line number.

end: *END_OF_FILE / <integer 1..2147483647> / <var: int: 4> / (<reg: int:4>)
Allows you to specify a page or line number in the print file at which output is to terminate. The dimension operand specifies whether it is a page or line number. The value specified here must be greater than the value specified for the begin operand.

end: *END OF FILE

Output terminates at the end of the file. The pages of a print file are specified as described for the LINEPP operand (unless control characters cause a page feed to be executed prematurely).

end: <integer 1..2147483647>

Integer specifying a page or line number in the print file.

end: <var: int:4> / (<reg: int:4>)

Name of a field defined with FL or a register containing the value. An integer (4 bytes in length) is stored in this field or register and interpreted as a page or line number.

dimension: *PAGES / *LINES / <var: enum-of layout part dim set:1>

Specifies whether the values specified for the begin and end operands are to be interpreted as page or line numbers.

dimension: *PAGES

The values specified for the begin and end operands are to be interpreted as page numbers.

dimension: *LINES

The values specified for the begin and end operands are to be interpreted as line numbers.

dimension: <var: enum-of layout_part_dim_set:1>

The page/line numbers are not specified directly by means of an operand value; instead, they are specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

1	*PAGES
2	*LINES
See the parameter list description as of page 109.	

OUTTRAY=*STD / <integer 1..99> / <var: int: 1> / *IGNORE / *SORTER

Defines the paper output tray for LP65 printers, APA printers, and for the RSO printers 2030-PCL, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL, 4830-PCL, 4850-PCL, 9000-PCL, 9014, 9015, 9026-PCL, 9026-RENO, Pagestream 55 and Pagestream 75.

The following applies to PCL printers:

The output tray number specified is not checked against the number of output trays on the destination printer. Specifying an output tray number for which no tray exists might cause a printer error.

The job can only be executed on single page feed printers.

OUTTRAY=*STD

Output goes into the paper output tray defined as the standard output tray in the device entry (SPOOL parameter file).

The paper output tray selection applies for the entire spoolout process, i.e. including the header and trailer pages.

The paper output tray selection by BS2000 can be prevented via a PCL file or an input at the printer console. The OUTTRAY entries have no effect in these cases. A paper output tray can also be defined in the PCL file.

The order of priority for selecting an output tray is as follows:

- 1. the number of the output tray specified in PRNTDOC, provided this is not prevented by a PCL file or an entry at the printer console
- 2. any selection via the PCL file specified in the PRNTDOC macro
- any selection via the PCL file specified in the form definition in the SPOOL parameter file
- 4. the default values defined in the device record

OUTTRAY=<integer 1..99>

Defines the paper output tray to be used for the current print job.

OUTTRAY=<var: int: 1>

Name of a field defined with FL or a register containing the value. An integer (1 byte in length) is stored in this field or register and interpreted as the number of the paper output tray.

OUTTRAY=*IGNORE

If you specify this value, the print control does not send an output tray selection code to the printer. This allows you to specify the output tray in the prolog file.

OUTTRAY=*SORTER

Specifies that the sort mechanism is to be used for the current job, using the SRTMODE operand. The sort mechanism is permitted for printers 9026-RENO, 4818-PCL, 4822-PCL and 4825-PCL and can be employed for up to 20 paper output trays. It cannot be controlled via the default values defined in the device entry.

Note

Print resources such as PROLOG, EPILOG, DIA, MEMBER and font character files are sent before the paper output tray is selected.

OVERLAY=(face, reverse)

Specifies whether an EFO data overlay is to be used for LP65 printers on the recto and/or verso (front/back of the paper). They must be available in the printer memory. The OVERLAY, TWOSIDE and PAGECOP operands are part of one and the same LP65 printer command and are therefore linked to each other. If only the OVERLAY operand is specified, SPOOL generates default values for the other two operands. These default values are overwritten by any value specified in a PCL file or at the printer console. An overview of the possible combinations of these three linked functions can be found in the description of printer type LP65 in the "SPOOL (BS2000/OSD)" manual.

face: *STD / *NONE / <integer 1..127> / <var: int: 1> / (<reg: int:1>) Identification number of the overlay to be used on the recto.

face: *STD

The default value for the identification number of the overlay (or *NONE) is stored in the SPOOL parameter file.

face: *NONE

No overlay is to be used on the recto.

face: <integer 1..127>

Integer specifying the identification number of the overlay to be used on the recto.

face: <var: int: 1> / (<reg: int:1>)

Name of a field defined with FL or a register containing the value. An integer (1 byte in length) is stored in this field or register and interpreted as an identification number.

reverse: *STD / *NONE / <integer 1..127> / <var: int: 1> / (<reg: int:1>)

Identification number of the overlay to be used on the verso.

reverse: *STD

The default value for the identification number of the overlay (or *NONE) is stored in the SPOOL parameter file.

reverse: *NONE

No overlay is to be used on the verso.

reverse: <integer 1..127>

Integer specifying the identification number of the overlay to be used on the verso.

reverse: <var: int: 1> / (<reg: int:1>)

Name of a field defined with FL or a register containing the value. An integer (1 byte in length) is stored in this field or register and interpreted as an identification number.

PAGECOP=

Number of copies. In local SPOOL mode, only for the printer types HP, HP90, LP65, 4830-PCL and 4850-PCL. Number of page copies.

PAGECOP=<u>*STD</u> / <integer 0..255> / <var: int: 2> / / (<reg: int: 2>)

Specifies how many times each individual page is to be repeated. The ADDCOP operand allows additional printouts of the whole file to be requested.

A PRNTDOC macro with the PAGECOP operand is rejected if the line number specified in the LINEPP operand is greater than the number of lines in the loop record minus the number of lines before the line on which the vertical tab "channel 1" is defined. On an HP or HP90 printer a maximum of 255 copies of a page can be printed in succession.

PAGECOP=255 has the same effect as PAGECOP=254: one original and 254 copies are printed.

PAGECOP=*STD

For LP65 printers: The number of page copies is as specified in the PCL file. For all other printer types: PAGECOP=0.

PAGECOP=<integer 0..255>

Integer specifying the number of page copies.

PAGECOP=<var: int: 2> / (<reg: int:2>)

Name of a field defined with FL or a register containing the value. An integer (2 bytes in length) is stored in this field or register and interpreted as the number of page copies.

PAGEPCL=

Specifies which page definition is to be used when printing to LP65 or APA printers.

PAGEPCL=*STD

For APA printers:

The standard definition specified in the SPSLIB is to be used; see the table of SPSLIB standard definitions in the appendix.

For LP65 printers:

Number of the PCL file with which the print file is to be output. Only the print file itself is printed with this PCL file. The header and trailer pages are controlled by the PCL file defined in the SPOOL parameter file's form.

Note

A PCL file that makes it impossible to select a different PCL file by means of a channel command must not be used.

When the specified PCL file is started, the operating mode of the printer is checked. If EXCCW mode is active and an error occurs during printing, the restart point is the last SECTION record; if nothing is specified for the SECTREC operand in the PRNTDOC macro, the entire file is printed again.

PAGEPCL=<integer 1..50000>

Only for LP65 printers:

Integer specifying the number of the PCL file with which the print file is to be output.

PAGEPCL=<var: int: 4> / (<reg: int:4>)

Name of a field defined with FL or a register containing the value. An integer (4 bytes in length) is stored in this field or register and interpreted as the number of the PCL file.

PARAM='<name 1..27>'

Specifies the address of the operand list (permitted only in the case of MF formats 2 and 3). You will find more information in the "Executive Macros" manual.

POOLIND=

Number of the font from the font pool with which the spoolout job is to be processed. The number of the font is determined by its position in the definition of the font pool. The specified font is used when POOLIND is specified with CONTMOD = *PAGE_MODE.

POOLIND=0

No number of a font from the font pool is specified.

POOLIND = <integer 0..64>

Integer specifying the number of the font from the font pool.

POOLIND = <var: int:1> / (<reg: int:1>)

Name of a field defined with FL or a register containing the value. An integer (1 byte in length) is stored in this field or register and interpreted as the number of the font.

POOLNAM=

Name of the font pool (for HP and HP90 printers only) to be used for output.

POOLNAM = *NONE

No font pool is required for output.

POOLNAM = '<name 1..4>'

Specifies a font pool (with a maximum of 64 fonts in the case of an HP printer and 46 fonts in the case of an HP90 printer) from which one or more fonts are to be used for output. All the fonts in this pool are loaded into the font buffer when the spoolout job is processed.

POOLNAM = <var: char:4> / (<reg: char:4>)

Name of a field defined with CL or a register containing the value. A string (4 bytes in length) is stored in this field or register and interpreted as the name of the font pool.

PREFIX=S / '<name 1..1>'

Specifies the first character of field names and equates.

PRJCLAS=*JOB-CLASS / <var: int: 1> / <integer 1..255>

Only for the SPOOL administrator.

Specifies the spoolout job class.

PRJCLAS=*JOB-CLASS

The job class default value is taken from the JOINFILE.

PRJCLAS=<var: int: 1>

Name of a field defined with FL or a register containing the value. A 1-byte integer is stored in this field or register and is interpreted as the job class number.

PRJCLAS=<integer 1..255>

The specified job class is assigned to the spoolout job. The value 0 (zero) is not permitted for compatibility reasons.

PRJNAME=

Job name for the spoolout job.

The job name can be formed from a maximum of 8 characters from the set (A,...Z,0,...9,@,#,\$,..,-) but must not start with a hyphen or end with a period. It may only start with a period if this is followed by an alpha character; in this case, the period is not printed as part of the job name on the header page. The special character string period and hyphen '.-' may only be specified in single quotes.

The job name is printed on the header page in the third uppercase line and also appears in the output of the SHOW-PRINT-JOB-STATUS command. If this operand is omitted, the job name from the SET-LOGON-PARAMETERS command is used instead.

PRJNAME=*JOB NAME

The spoolout job does not have its own name. If a job name has been assigned (in the SET-LOGON-PARAMETERS command) to the job issuing the command, this name is printed in the third uppercase line on the header page.

PRJNAME = <c-string 1..8 with-low>

Job name which is to be assigned to the spoolout job (and which overwrites a job name assigned to the job issuing the command).

PRJNAME = <var: char:8> / (<reg: char:8>)

Name of a field defined with CL or a register containing the value. A string (8 bytes in length) is stored in this field or register and interpreted as a job name.

PRJPRIO=

Defines the urgency with which this spoolout job is started relative to other spoolout jobs.

PRJPRIO = *JOB PRIORITY

Default: the spoolout job has the same priority as the job issuing the command.

PRJPRIO = <integer 30..255>

Integer specifying the priority to be assigned to the spoolout job. The highest priority you can assign here is defined in the user catalog and can be displayed by means of the SHOW-USER-ATTRIBUTES command. If an invalid priority is entered (or no priority at all), the spoolout job is given the same priority as the job issuing the command.

PRJPRIO = <var: int:1> / (<reg: int:1>)

Name of a field defined with FL or a register containing the value. An integer (1 byte in length) is stored in this field or register and interpreted as a priority.

PRNAME=

Specifies the requested target printer of the print job. You can specify a distributed local printer pool, a non-distributed local printer pool, an RSO printer pool or an RSO printer. With Distributed Print Services you can specify other target printers. For more information, refer to the "Distributed Print Services (BS2000/OSD)" manual.

If you specify a printer pool:

The job is output on any printer in the specified device pool. The pool must be defined in the SPOOL parameter file and can contain up to 16 RSO devices or 16 local SPOOL devices, but not both at the same time. Device pools are managed by means of the SPSERVE statements ADD-, MODIFY-, REMOVE-, and SHOW-PRINTER-POOL (see the "SPSERVE (BS2000/OSD)" manual).

The spoolout job is rejected if:

- no printer type from the device pool is assigned in the PRNTDOC macro (see the SHOW-SPOOL-FORMS command or SPSERVE statement)
- the OVERLAY operand is specified.

Please bear in mind that the pool can contain various types of devices: if the file to be printed contains control characters that are only interpreted by a certain printer type, a printer of this type should be specified (implicitly) in the PRNTDOC macro. One possibility is to specify in the PRNTDOC macro a form that is defined only for the desired printer type (ADD-SPOOL-FORM).

PRNAME=*STD

The spoolout job is to be processed on the default device type specified in the SPOOL parameters (PRINT-CMD-DEFAULTS).

PRNAME=<c-string: c-string 1..8>

Only for RSO.

Symbolic name of the RSO device on which the spoolout job is to be processed.

PRNAME=<var: char:8>

Name of a field defined with CL or a register containing the value. A string (8 bytes in length) is stored in this field or register and interpreted as the name of the printer.

PRNAME=*IPP

Only for RSO.

Indicates that the spoolout job is to be processed on an IPP printer addressed by its URL.

PROCADM=(user, account, passwd)

Only for spoolout jobs under the user ID TSOS (OSD V1.0) or under the user ID with the privilege PRINT-SERVICE-ADM (OSD V2.0). It allows you to specify whether the spoolout job is to be executed under your own user ID and account number or a different, specified, user ID and account number.

user: *SAME / '<alphanum-name 1..8>' / <c-string 1..8> /

<var: char: 8> / (<reg: char:8>)

User ID under which the spoolout job is to be executed.

user: *SAME

The spoolout job is to be executed under your own user ID.

user: '<alphanum-name 1..8>' / <c-string 1..8>

Specifies as a string in single quotes a user ID under which the spoolout job is to be executed.

user: <var: char:8> / (<reg: char:8>)

Name of a field defined with CL or a register containing the value. A string (8 bytes in length) is stored in this field or register and interpreted as a user ID.

account: *NONE / '<alphanum-name 1..8>' / <c-string 1..8> /

<var: char: 8> / (<reg: char:8>)

Account number under which the spoolout job is to be executed.

account: *NONE

No account number is specified.

account: '<alphanum-name 1..8>' / <c-string 1..8>

Specifies as a string in single quotes an account number under which the spoolout job is to be executed.

account: <var: char:8> / (<reg: char:8>)

Name of a field defined with CL or a register containing the value. A string (8 bytes in length) is stored in this field or register and interpreted as an account number.

passwd: *NONE / <c-string 1..8> / <var: char: 8> / (<reg: char:8>)

Password of the user ID under which the spoolout job is to be executed.

passwd: *NONE

No password is specified.

passwd: <c-string 1..8>

Specifies as a string in quotes the password of a user ID under which the spoolout job is to be executed.

passwd: <var: char:8> / (<reg: char:8>)

Name of a field defined with CL or a register containing the value. A string (8 bytes in length) is stored in this field or register and interpreted as a password.

PROGRAM=(name, string)

Defines the device name of a virtual printer which is to process the spoolout job.

name = specifies the device name entered in the SPOOL parameter file.

string = passes a specific character string to the application program.

name: *ANY / <var: char: 8> / <c-string: alphanum-name 1..8, c-string 1..8> / Defines the device name of a virtual printer which is to process the spoolout job.

name: *ANY

Any virtual printer may process the spoolout job.

name: <var: char: 8>

Name of a field defined with CL. A string (8 bytes in length) is stored in this field and interpreted as the name of the virtual printer.

name: <c-string: alphanum-name 1..8, c-string 1..8>

The name of the virtual printer which is to process the spoolout job is specified by a string enclosed in quotes.

string: *NONE / <var: char: 8> / <c-string: alphanum-name 1..8, c-string 1..8> / Defines supplementary information for the virtual printer which is to process the spoolout job.

string: *NONE

No supplementary information for the virtual printer.

string: <var: char: 32>

Name of a field defined with CL. A string (32 bytes in length) is stored in this field and interpreted as supplementary information for the virtual printer.

string: <c-string: alphanum-name 1..32, c-string 1..32>

The supplementary information for the virtual printer which is to process the spoolout job is specified by a string enclosed in quotes.

PRTYPE=

Specifies which printer type is to process the print job. Only local printer types can be specified.

PRTYPE=*ANY

A specific printer type is not requested. In this case, the SPOOL subsystem automatically determines the permitted printer types that can process the user request. *ANY must be specified for output on RSO printers and printers in UNIX systems.

PRTYPE=*LP65 PRINTER

The spoolout job is to be processed on an LP65 printer.

PRTYPE=*HP PRINTER

The spoolout job is to be processed on an HP or HP90 printer. *HP_PRINTER includes the following printer types: 2090/2140/2240(HP90). The control characters for HP and HP90 laser printers are identical; a spoolout job for an HP printer can be processed on an HP90 printer and vice versa.

Selection of the printer type is affected by whether or not the CONTMOD operand is specified at the same time:

Printer selection	CONTMOD=*PAGE_MODE (default)	CONTMOD=(*PAGE_MODE,,*HP)
PRTYPE=* ANY	Output is possible on all printer types	Output is possible on HP and HP90 printers. Other control characters are converted to HP/HP90 control characters.
	The following applies to HP/HP90 printing The OVERPRINT function is converted.	

Printer selection	CONTMOD=*PAGE_MODE (default)	CONTMOD=(*PAGE_MODE,,*HP)	
PRTYPE = *HP_PRINTER	Output is possible only on HP and HP90 printers. The OVERPRINT function is converted to the LINE-MERGE function.		
	PRFILE restrictions: The file must not contain the X'FF' character.	HP-specific control characters are supported. The following restriction applies: The file can contain the X'FF' character only as an escape character if control character interpretation is active	

Note for HP/HP90 printers

If the entries in PRNTDOC specify output to an HP or HP90 laser printer, a PRFILE must be available, otherwise the command is rejected. If there is no HP/HP90 available in an installation, or only devices with insufficient configurations (not enough fonts, no graphics buffer for FOBs or no page rotation module, for example), the jobs can only be output to replay tape. Systems support can issue the SHOW-SPOOL-JOB-STATUS command to obtain information on these jobs.

PRTYPE=*APA PRINTER

The spoolout job is to be processed on a 2050-APA-PRINTER, 2090-APA-PRINTER or 2090-TWIN-PRINTER.

PRTYPE=<var: enum-of prtype_set:1>

The printer type is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*ANY
1	*LP65_PRINTER
2	*APA_PRINTER
3	*HP_PRINTER
See the parameter list description as of page 109.	

RECPART=(first, last)

Specifies that only a certain part of each selected record is to be processed.

first: 1 / <integer 1..32767> / <var: int: 2> / (<reg: int:2>)

Allows you to specify a byte number (record column) as of which the records of a file are to be output. (The bytes of a record are numbered from left to right, beginning at 1; ISAM keys and control characters are components of a record).

What happens depends on whether or not one of the values *BY_EBCDIC_CONTROL, *BY_IBM_CONTROL or *BY_ASA_CONTROL is specified for the LINESP operand.

If one of these values is specified:

Output starts with the data byte following the specified byte number. The feed control character is interpreted irrespective of what is specified for *first*, provided the value for *first* is less than the length of the record. If the specified value is greater than the length of the record, the record is ignored (i.e. not printed or fed).

If none of these values is specified:

Output starts with the data byte corresponding to the specified byte number.

first: 1

Output starts with the first byte of each record.

first: <integer 1..32767>

Integer specifying the byte number (record column) as of which the records of a file are to be output.

first: <var: int:2> / (<reg: int:2>)

Name of a field defined with FL or a register containing the value. An integer (2 bytes in length) is stored in this field or register and interpreted as a record column.

last: *STD / <integer 1..32767> / <var: int: 2> / (<reg: int:2>)

This specifies the last byte to be printed from each record. What happens depends on whether or not one of the values *BY_EBCDIC_CONTROL, *BY_IBM_CONTROL or *BY_ASA_CONTROL is specified for the LINESP operand.

If one of these values is specified:

Output ends with the data byte following the specified byte number (except when FIRST-CHARACTER is specified for an ISAM file with KEY-POSITION=5, in which case output terminates with the data byte corresponding to the specified byte number).

If none of these values is specified:

Output ends with the data byte corresponding to the specified byte number. If the records are longer than permitted by the form definition, they are continued on the next line.

last: *STD

Default:

- the end of the print line (136)
- 2048 for an EAM file, SYSLST or SYSOUT

last: <integer 1..32767>

Integer specifying the last byte to be printed from each record.

last: <var: int:2> / (<reg: int:2>)

Name of a field defined with FL or a register containing the value. An integer (2 bytes in length) is stored in this field or register and interpreted as the last byte.

ROT=*NO/ROT_0/*ROT_90/*ROT_180/*ROT_270/*ROT_0-180/*ROT_180-0/*ROT_90-270/*ROT_270-90/*BY-CONTROL-CODES*/

<var: enum-of rotation set: 1>

Specifies whether the pages to be printed from the spoolout job are to be rotated, and if so by how many degrees. For the printer types 4830-PCL and 4850-PCL, only the values 0, 90, 180 and 270 can be specified.

The print page set up in the printer is rotated (clockwise) by a certain number of degrees and printed on the form; for example, paper inserted in the printer in portrait format can be printed in landscape format. A separate loop is needed for pages rotated through 90°/270° (see the ROTLOOP operand). Unless ROT=NO is specified, output is directed (automatically) to HP and HP90 printers, PCL printers and 4830-PCL and 4850-PCL printers. SHOW-SPOOL-PARAMETERS indicates whether or not an HP or HP90 printer with a page rotation module is available in the current SPOOL configuration: output field DEVICE-TYPE:..., ROT=YES/NO.

Spoolout jobs with page rotation are displayed in the outputs for the SHOW-USER-STATUS and SHOW-PRINT-JOB-STATUS JOB-IDENTIFICATION=TSN(TSN=...) commands. The feed for rotated pages is generally controlled via a separate loop (ROTLOOP). If you have specified neither a ROTLOOP for PRNTDOC nor a form with a defined ROTLOOP (see ADD-SPOOL-FORM), the default rotation loop R06 in the default form handles feed control for the rotated pages. Header and trailer pages are not printed out in rotated format.

ROT=*NO

Page rotation is not performed. Any control characters for page rotation in the file are not interpreted.

ROT=*BY-CONTROL-CODES

Control characters for page rotation in the file are interpreted when CONTMOD= *PHYSICAL is specified.

ROT=*ROT_90 / *ROT_180 / *ROT_270

Each print page is rotated by 90^0 / 180^0 / 270^0 (clockwise) and printed out. Control characters for page rotation contained in the file are not interpreted. A separate loop is needed for pages rotated through 90^0 / 270^0 . You must check that output with the specified loop does not lead to errors. If an error occurs, the job is rejected.

ROT=*ROT 0 180 / *ROT 180 0 / *ROT 90 270 / *ROT 270 90

The odd pages (number before the hyphen) and even pages (number after the second hyphen) are to be printed at different angles.

ROT=<var: enum-of rotation set:1>

Page rotation is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

213	*NO
232	*BY_CONTROL_CODES
241	*ROT_90
242	*ROT_180
243	*ROT_270
244	*ROT_0_180
245	*ROT_90_270
246	*ROT_180_0
247	*ROT_270_90
See the parameter list description as of page 109.	

ROTLOOP=

Only for HP and HP90 printers with a page rotation module.

Specifies the loop to control output in landscape format.

The loop name must not include the character '\$', '&' or '@'.

ROTLOOP=*STD

Output of rotated pages is to be controlled by the default rotation loop of the specified form or the default rotation loop R06 of the default form (if no form was specified in the PRNTDOC macro).

ROTLOOP='<alphanum-name 1..3>'

String in single quotes for the name of the loop to control the feed for the rotated pages.

ROTLOOP=<var: char: 3> / (<reg: char:3>)

Name of a field defined with CL or a register containing the value. A string (3 bytes in length) is stored in this field or register and interpreted as the name of the loop.

RSOPAR=

The address of a parameter list containing RSO-specific values for the macro call can be specified.

RSOPAR=*NONE

Specifies that the values are not to be supplied via a parameter list.

RSOPAR=<var: pointer> / (<reg: pointer>)

Specifies a pointer. In other words, the variable or field does not contain the RSO parameter list itself but the address of a memory location at which the RSO parameter list is stored (A(field) or a register).

SECTREC=(*ident*, *length*, *position*)

Specifies whether the file is structured by means of file marks. "ident" allows you to specify the strings in the records that are to function as file marks, "length" allows you to specify the length of the string, and "position" allows you to specify the position of the specified string in the record.

ident: *NONE / <c-string 1..60> / <var: char: 60> / (<reg: char:60>)

Specifies whether the file is to be structured by means of file marks. The FIRSREC operand specifies which parts of the print file subdivided into sections are to be output. Any strings in the records can be used as file marks. These strings can either be in SECTION records, which are not printed, or in the print file's normal records, which are printed. A string can be specified in the form of printable characters or hexadecimal characters.

The search for the start of the section to be printed is executed in a separate "pseudo controller" task. Neither the user task nor the printer is locked while this is being done. A pseudo controller writes the address of the first record in the section to the SPOOL control block; the spoolout job can then be processed (PREPROCESSING). If the desired section is not found in the file, an error message appears on the trailer page (for the layout of the trailer page see page 591).

ident: *NONE

Specifies that the file is not to be structured by means of file marks.

ident: <c-string 1..60>

Specifies in single quotes the string in the records.

ident: <var: char:60> / (<reg: char:60>)

Name of a field defined with CL or a register containing the value. A string (60 bytes in length) is stored in this field or register and interpreted as a string.

length: 0 / <integer 1..60> / <var: int:1> / (<reg: int:1>)

Specifies the length of the string.

length: 0

Default: the string has a length of 0 bytes.

length: <integer 1..60>

The length of the string is specified as an integer from 1 to 60.

length: <var: int:1> / (<reg: int:1>)

Name of a field defined with FL or a register containing the value. An integer (1 byte in length) is stored in this field or register and interpreted as the length of the string.

position: *STD / <integer 1..2047> / <var: int:2> / (<reg: int:2>)

Specifies as of which position (as of which byte after the record length field) the specified string begins in the SECTION record.

position: *STD

The string searched for begins by default at the beginning of the record:

- in the case of a SAM file: with the first byte after the record length field
- in the case of an ISAM file with KEY-POS=5: with the first byte after the key
- in the case of an ISAM file with KEY-POS > 5: with the first byte after the record length field

position: <integer 1..2047>

Integer specifying the position as of which the specified string begins in the SECTION record.

position: <var: int:2> / (<reg: int:2>)

Name of a field defined with FL or a register containing the value. An integer (2 bytes in length) is stored in this field or register and interpreted as a position.

SPSPAR=

The address of a parameter list containing SPS-specific values for the macro call can be specified.

SPSPAR = *NONE

Specifies that the values are not to be supplied via a parameter list.

SPSPAR = <var: pointer> / (<reg: pointer>)

Specifies a pointer. In other words, the variable or field does not contain the SPS parameter list itself but the address of a memory location at which the SPS parameter list is stored (A(field) or a register).

SRTMODE=*NO / <var: enum-of _srtmode_se:1> / *GROUP / *COLLATE / *STACKER / *AUTOMATIC

Specifies that the sort mechanism is to be used for the current job. The sort mechanism is permitted for printers 9026-RENO, 4818-PCL, 4822-PCL and 4825-PCL and can be used for up to 20 paper output trays. It cannot be controlled via the default values defined in the device entry.

SRTMODE=*NO

All document pages are placed in the sort trays from bottom to top. The 9026-RENO is an exception. In this case, the pages are placed in one paper tray according to optimum accessibility.

SRTMODE=*GROUP

If specified with PAGECOP, each example of a specific single page is placed into a separate sort tray. Each subsequent page in the sequence is then also output to the same tray. After the print job is finished, each output tray contains a copy of the printed document.

Example

The file comprises three pages and PAGECOP=2 was specified:

Page 3	Page 3	
Page 2	Page 2	
Page 1	Page 1	
Tray 1	Tray 2	Tray 3

SRTMODE=*COLLATE

If specified with PAGECOP, all examples of a specific page are collected into one paper tray. each subsequent page in the sequence is then output to a separate tray. The output trays are used from bottom to top.

Example

The file comprises three pages and PAGECOP=2 was specified:

Page 1	Page 2	Page 3
Page 1	Page 2	Page 3
Tray 1	Tray 2	Tray 3

SRTMODE=*STACKER

Cannot be used with 9026 Printers.

All printed pages, up to a maximum of 500, are output to the sort mechanism main output tray. This mode is useful for printing a single copy of a very long document.

SRTMODE=*AUTOMATIC

Can only be used with 9026 Printers.

Paper output is controlled automatically, depending on the number of page copies specified with PAGECOP and the number of available sort trays. The printed pages are output unsorted as with *NO, if the number of copies specified with PAGECOP equals the number of available sort trays. The printed pages are sorted into separate documents as with *GROUP if the number of copies specified with PAGECOP is less than the number of available sort trays. Or the printed pages are output into sort trays according to page numbers as with *COLLATE, if the number of copies specified with PAGECOP is greater than the number of available sort trays.

SRTMODE=<var: enum-of 'srtmode set:1>

The sort mechanism definition is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

1	NO
2	GROUP
3	COLLATE
4	STACKER
5	AUTOMATIC
See the parameter list description as of page 109.	

These values are defined in the ASSEMBLER, C and COBOL interface. For example, the following is in the C interface:

```
/* srtmode_set
                                                                 */
/* ENUM srtmode set
                                                                 */
<< The sort mechanism is not specified explicitly, but rather via>>
<< the following constant definition>>
<< (the integer value stored in the constant can therefore be used>>
<< in the parameter list of the function call): >>
*/
                                                                 */
#define PRNTDOCsrtmode collate 3 /* srtmode =*collate
#define PRNTDOCsrtmode stacker 4 /* srtmode =*stacker
                                                                 */
#define PRNTDOCsrtmode automatic 5 /* srtmode =*automatic
                                                                 */
```

STPROC=

Specifies when the system file is to be printed before the end of the job.

STPROC=*IMMEDIATE

The spoolout job is to be created and processed immediately the command is entered.

STPROC=*NO

Specifies that the operand is not to be evaluated.

STPROC=*AT FILE CLOSING

The spoolout job is to be processed immediately after the system file is closed. A system file is closed:

a) In the case of primary assignment: by means of the LOGOFF or CANCEL-PRINT-JOB command (from another ID), i.e. after the end of the job

- b) In the case of assignment to a cataloged file:
 - by means of the LOGOFF or CANCEL-PRINT-JOB command (from another ID), i.e. after the end of the job
 - by means of another ASSIGN-SYSFILE command (change of assignment) for the same system file
- c) In the case of a procedure run, once procedure level 0 is reached.

STPROC=<var: enum-of stproc set:1>

The time is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

1	*NO
2	*AT_FILE_CLOSING
4	*IMMEDIATE
See the parameter list description as of page 109.	

SYSNN=array(16): *DUMMY / <integer 0..101> / *SYSLST / *SYSOUT /

<var: int: 2> / (<reg: int:2>)

A SYSLST or SYSOUT system file is to be output. Up to 16 elements can be specified in a list.

SYSNN=*DUMMY

Default: no system file is output.

SYSNN = <integer 0..101>

The integer is interpreted as follows:

1..99: the number nn, which is to be used in the file name SYSLSTnn.

100: the SYSLST system file is to be output.

101: the SYSOUT system file is to be output.

SYSNN = *SYSLST

The SYSLST system file is to be output.

SYSNN = *SYSOUT

The SYSOUT system file is to be output.

SYSNN = <var: int:2> / (<reg: int:2>)

Name of a field defined with FL or a register containing the value. An integer (2 bytes in length) is stored in this field or register and interpreted as the number of the system file to be output.

TOPOFF=*IGNORE / <integer -255..255>

Defines the offset of the top edge of the print page from the top edge of the physical page in millimeters. First the print page is positioned on the paper and then the writing is rotated and positioned within the print page. This means that when the print page is moved around on the paper, the orientation of the text within it is ignored.

TOPOFF=*IGNORE

Operand default, the print page is not shifted vertically on the paper.

TOPOFF=<integer -255..255>

The print page is shifted vertically by the specified value on the paper.

TRANTAB=(name, file)

Specifies whether a code translation table is to be used to process the spoolout job and, if so, from which file it is to be taken. The code translation table is required when the default escape character, 'FF', is to be replaced by another character.

name: *NONE / '<alphanum-name 1..8>' / <c-string 1..8> /

<var: char: 8> / (<reg: char:8>)

Specifies whether a code translation table is to be used to process the spoolout job.

name: *NONE

No code translation table is used.

name: '<alphanum-name 1..8>' / <c-string 1..8>

The name of the code translation table to be used to process the spoolout job is specified as a string in quotes.

name: <var: char: 8> / (<reg: char:8>)

Name of a field which is defined with CL or a register containing the value. A character string (8 bytes long) which is interpreted as the name of the code conversion table is stored in this field/register.

file: *SYSTEM / *STD / '<filename 1..44 without-userid>' / <c-string 1..44> /

<var: char: 44> / (<reg: char:44>)

Specifies the file from which the code translation table is to be taken.

file: *SYSTEM

The specified code translation table is taken by default from the \$SYSSPOOL.PRFILE file; in the case of RSO printers, a user RSOFILE can also be specified.

file: *STD

For RSO print jobs, the specified code translation table used is that from the specified file (with FILE=<filename..>) or from the standard resource file \$TSOS.RSOFILE (with FILE=*STD/*SYSTEM).

Only FILE=*STD is relevant for SPOOL print jobs. Other values are ignored and changed to *STD. Consequently, the code translation table is always taken from the user resource file specified in USER-RESOURCE-FILE or from the standard resource file \$SYSSPOOL.PRFILE if no user resource file is specified.

file: '<filename 1..44 without-userid>' / <c-string 1..44>

The name of the file from which the specified code translation table comes is specified as a string in single quotes.

file: <var: char:44> / (<reg: char:44>)

Name of a field defined with CL or a register containing the value. A string (44 bytes in length) is stored in this field or register and interpreted as the name of the file.

TRUNC=

Specifies what happens when lines are truncated.

TRUNC=*STD

The default from the SPOOL parameter file applies. You can use the SHOW-SPOOL-PARAMETERS command or SPSERVE statement to display this value (field: ERROR-PR=(TRUNC=)).

TRUNC=*DELETE FILE

Processing of the spoolout job continues (i.e. DELF = *YES is executed). An appropriate warning is printed on the trailer page.

TRUNC=*KEEP FILE

Processing of the spoolout job continues, but the file is not deleted afterwards.

TRUNC=<var: enum-of truncation set:1>

Continuation of the spoolout job is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*STD
1	*DELETE_FILE
2	*KEEP_FILE
See the parameter list description as of page 109.	

TWOSIDE=

Specifies for LP65, PCL, APA and RSO printers whether the paper is to be printed on one side or both.

TWOSIDE=*STD

The paper is printed on one side (simplex mode) or two (duplex mode), as defined in the PCL file.

TWOSIDE = *NO

The job is printed in simplex mode, i.e. on only one side of the paper.

TWOSIDE = *YES

The job is printed in duplex mode, i.e. on both sides of the paper.

Besides on LP65 printers with single sheet processing, the job can also be printed on 2030-PCL, 9026-PCL, 9026-RENO, 4822-PCL, 4825-PCL, 4824-PCL, 4830-PCL and 9000-PCL.

TWOSIDE = *TUMBLE

The job is printed in duplex mode, i.e. on both sides of the paper, and the pages are turned over from top to bottom rather than from left to right.

Besides on LP65 printers with single sheet processing, the job can also be printed on 2030-PCL, 9026-PCL, 9026-RENO, 4822-PCL, 4825-PCL, 4824-PCL, 4830-PCL and 9000-PCL.

TWOSIDE=<var: enum-of twoside set:1>

One- or two-sided printing is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*STD
1	*YES
2	*TUMBLE
4	*NO
See the parameter list description as of page 109.	

Note

The functions TWOSIDE, OVERLAY and PAGECOP are linked and part of one and the same printer command. If only one of these operands is specified, default values are generated for the others.

USERRES=

Specifies a user file containing all the resources required for output on different printer types: user-defined loops, fonts, overlays, font pools, code translation tables and SPS data stream definitions. The following can be specified:

- a user PRFILE containing loops, fonts, overlay entries (FOB) and font pool entries (CHARSET operand)
- a user SPSLIB containing the PAGEDEFs (PAGEPCL operand), FORMDEFs, fonts, page segments, overlays and raster image data
- a user RSOFILE (only for RSO) containing loops

If no user PRFILE, SPSLIB or RSOFILE is specified, the information is taken from the following file: \$SYSSPOOL.PRFILE, \$SYSSPOOL.SYSPRT.SPS.021 or \$SYSSPOOL.RSOFILE, respectively.

USERRES = *DUMMY

The required resources are taken from \$SYSSPOOL.PRFILE, \$SYSSPOOL.SYSPRT.SPOOLSPS.021 or \$TSOS.RSOFILE.

USERRES='<filename 1..44>' / <c-string 1..44>

Name of a user PRFILE, SPSLIB or RSOFILE, which may contain a catalog ID and a user ID. SPOOL uses this file name with the suffix .PRFILE, .SPSLIB or .RSOFILE. The string can contain up to 28 characters without the catalog ID and user ID, to ensure that this user file can be called from any ID. If the file name is specified without a user ID, the file is searched for under the user ID of the caller first, then under SYSSPOOL. If it is not found, the command is rejected. If a user ID is specified, the file is searched for under this ID only.

Example 1

PRNTDOC DNAME=(DATEI), USERRES='\$XX.XX'

A search is carried out for the \$XX.XX.PRFILE file. If the file is not found, the command is rejected.

Example 2

If the catalog ID is specified, the search is limited to the specified pubset:

PRNTDOC DNAME=(DATEI), USERRES=':A:XXXXX'

The file :A:\$userid.XXXXX.PRFILE is searched for. If it is not found, a search is carried out for the file :A:\$SYSSPOOL.XXXXX.PRFILE. If this file is not found either, the command is rejected.

Example 3

PRNTDOC DNAME=(DATEI), USERRES=':A:\$XX.XXXXX'

The file: A:\$XX.XXXX.PRFILE is searched for. If it is not found, the command is rejected.

If the file is on an exported PVS (EXPORT-PUBSET command), all spoolout jobs that require this PVS are placed in the KEEP queue. When the PVS becomes available again (IMPORT-PUBSET command), the spoolout jobs are restarted.

USERRES=<var: char: 44> / (<reg: char:44>)

Name of a field defined with CL or a register containing the value. A string (44 bytes in length) is stored in this field or register and interpreted as the name of the resource file.

VARIANT=001 / <c-string 3..3>

Specifies the variant of the parameter list.

VIRTUAL=*STD / *ALLOWED / <var: enum-of virtual_set:1> / *NOT-ALLOWED

This operand allows the print job to be forwarded via a virtual printer to an application for processing.

VIRTUAL=*ALLOWED

The print job is preferentially passed to a virtual device. The print job is assigned to a real device if no virtual device is active when the print job is entered into the print job list.

VIRTUAL=*NOT-ALLOWED

This print job is not to be passed to an application program via a virtual printer.

VIRTUAL=<var: enum-of virtual set:1>

Whether and how a print job is passed to an application via a virtual printer is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	ALLOWED
1	NOT-ALLOWED
See the parameter list description as of page 109.	

These values are defined in the ASSEMBLER, C and COBOL interface. For example, the following is in the C interface:

VIRTUAL=*MUST

The print job is passed to a particular virtual printer that is specified with the parameters in parentheses. If the supervisor task of this virtual printer is inactive, the print job remains in a wait state.

SCHEDTIME = (schdate, schtime)

Specifies the scheduling time, i.e. the time at which the print job can be started at the earliest. The print job will be started from the specified point in time on, depending on printer availability and the processing regulations. If there are jobs with the status WFT, data transfer will be started from the specified time on. The operand may not be used in conjunction with UNIX systems.

schdate: *TODAY / <var: char 1..10> / <c-string 10..10> Specifies the day on which the print job will be started.

schdate: *TODAY

The print job will be started on the current day.

schdate: <var: char 1..10> / <c-string 10..10>

schtime: *NOW / <var: char 1..5> / <c-string 5..5>

Specifies the point in time at which the print job will be started.

schtime: *NOW

The print job is started immediately.

schtime: <var: char 1..5> / <c-string 5..5>

Specifies the time in the form <hour>:<minutes>. The value consists of two unsigned integers separated by a colon. The values allowed are 0 to 23 for the hour and 0 to 59 for the minutes. A zero can precede one-position values.

Notes

- 1. The date and time must be specified in the local time zone format in this command.
- 2. In order to secure the coherency of client and server time, both date and time are internally converted to the UTC format (Universal Time Coordinated). They are reconverted each time they are output.
- The date and time issued when the command SHOW-PRINT-JOB-ATTRIBUTES is used, is presented in the local time presentation format of the computer from which the command was issued. In other words, the date and time displayed may differ from the specifications originally made in the PRINT DOCUMENT command.

Differences may occur in the following cases:

- In the case of distributed print jobs, the time is displayed from different time zones on the client and the server.
- Although the time zone on a client and a server is the same, their clocks are not synchronized.
- The print jobs were defined during daylight saving time or winter time, but the display will be in a different time.
- 4. In a distributed environment, we recommend you synchronize all clocks in order to avoid inconsistencies in date and time displays.
- 5. If the SCHEDULING TIME operand is not supported by the spool version in a distributed environment, the operand is ignored and print jobs are processed in the usual way.
- 6. A print job has to have reached its scheduling time limit, for a print job to be assigned a replay tape.

Example

SCHEDTIME = (SCHDATE=*TODAY,SCHTIME=*NOW)

No scheduling time. The print job is processed in accordance with printer availability and processing regulations.

Incompatible operands in the PRNTDOC macro

The PRNTDOC macro is rejected if one of the operands in the column on the left is specified at the same time as one of the corresponding operands in the column on the right.

Operand	Incompatible operands
CHARSET =chars-name	PRTYPE=*LP65_PRINTER / *APA_PRINTER, CONTMOD=*APA
POOLNAM=nam	CONTMOD=*LOGICAL / *APA / *LINE-MODE / *PHYSICAL, PRTYPE=*LP65_PRINTER / *APA_PRINTER
CONTMOD=*APA	RECPART, DOCUMENT_PART (if no sections are specified), LEFTMAR, CHECKPOINT=*ON_SECTION_RECORDS
Only for RSO: CONTMOD =*LINE_MODE / *PHYSICAL	OUTFORM=*HEXADECIMAL, ROT=*BY_CONTROL_CODES / *ROT_n CONTMOD=(,*NO,) OVERLAY, ROTLOOP, POOLNAM, FOB, PRTYPE=*LP65_PRINTER / *APA_PRINTER / *HP_PRINTER,
Only for RSO: CONTMOD =*LOGICAL	OUTFORM=*HEXADECIMAL, ROT=*BY_CONTROL_CODES / *ROT_n ROTLOOP, POOLNAM, FOB, PRTYPE=*LP65 _PRINTER/ *APA_PRINTER / *HP_PRINTER,
CONTMOD =*PAGE-MODE(,,*HP)	PRTYPE=*LP65_PRINTER / *APA_PRINTER, INTRAY OUTTRAY
DELF=*YES	DTYPE=*EAM
FAMILY=*YES	SECTREC

(part 1 of 2)

Operand	Incompatible operands
DTYPE=*EAM	DELF=*DESTROY
DTYPE =*LIBRARY_ELEMENT	LOCKF=*YES
EFO	PRTYPE=*APA_PRINTER, CONTMOD=*LINE_MODE / *APA / *PHYSICAL, DOCFORM=*SPECIAL_FORMAT
FOB	CONTMOD=*LOGICAL / *APA / *LINE_MODE / *PHYSICAL, DOCFORM=*SPECIAL_FORMAT, PRTYPE=*LP65_PRINTER / *APA_PRINTER
LEFTMAR	CONTMOD=*APA
LOCKF=*YES	DTYPE=*LIBRARY_ELEMENT
LOOP	PRTYPE=*APA_PRINTER, CONTMOD=*APA
OUTFORM =*HEXADECIMAL	CONTMOD=*PHYSICAL
PAGECOP	CONTMOD=*APA, PRTYPE=*APA_PRINTER
PRTYPE=*APA	CONTMOD=*LOGICAL / *PHYSICAL / *LINE_MODE / *PAGE_MODE, DOCFORM=*SPECIAL_FORMAT, PAGECOP, ROTLOOP, LOOP, ROT, CHARSET, EFO, FOB
RECPART	CONTMOD=*APA
ROT =*BY_CONTROL_CODES / =*ROT_n	CONTMOD=*LOGICAL / *APA / *LINE_MODE / *PHYSICAL DOCFORM=*SPECIAL_FORMAT / *TEXT, PRTYPE=*LP65_PRINTER / *APA_PRINTER
USERRES	PRTYPE=*LP65_PRINTER

(part 2 of 2)

Return codes

The **PRNTDOC** macro is rejected with an error message if the file to which the call refers is already open in output mode. The following return codes are defined in the SCPRC macro, which is created as a submacro together with the PRNTDOC macro.



A return code indicating the result of execution of the PRNTDOC macro is transferred in the right-justified byte of register 15.

SPOOL is not loaded, R15 contains X'01000000'.

	·
X 'aa'	Meaning
X '00'	Successfully processed
X '01'	Invalid file name
X '02	Invalid PL access
X '04'	Error in the queue routine
X '08'	Contradictory operands
X '0C'	Error during catalog search
X '10'	Error during EAM processing
X '18'	Password invalid or missing
X '1C'	\$REQM error
X '20'	\$RELM error
X '24'	Invalid operand specification
X '2C'	File cannot be accessed by another user
X '30'	RECFORM=F and RECSIZE=0
X '34'	Creation date/expiration date not available. LASTPG = 0000 for the file
X '40'	EAM file not available
X '44'	System file contains no data
X '48'	*SYSLST, *SYSLSTnn and *SYSOUT cannot be specified together with the DELF operand
X '54'	A DELETE-FILE command is active for the output file
X '6C'	FLUSH/CLOSE error for the system file
X '70'	File currently in use
X '78'	The call PRNTDOC *SYSOUT is not permitted for an interactive job
X '7C'	LINESP=(*BY_EBCDIC_CONTROL,) is not permitted for a file with RECFORM=U

(part 1 of 2)

X 'aa'	Meaning
X '80'	File generation is not permitted
X '88'	Specified target printer does not exist
X '90'	Operands for HP/HP90 printer invalid or missing
X '94'	SPOOL subsystem no longer active
X '98'	System file to be printed is assigned to a dummy (*DUMMY) file (ASSIGN-SYSFILE command)
X '9C'	System file to be printed is assigned to a tape file
X 'A0'	System file to be printed is assigned to a temporary (user) file (ASSIGN-SYSFILE command)
X 'A4'	The SYSLSTnn file is not assigned to a SAM file (ASSIGN-SYSFILE command)
X 'A8'	The specified file is not available for other access
X 'B4'	Nonexistent form
X 'B8'	SAM error during conversion of the EAM file to a SAM file
X 'BC'	The ROT operand was specified (≠ *NO) although the function is not permitted (systems support can change this by means of M-S-P)
X 'C0'	The number of character sets or the size of the desired overlay exceeds the predefined value (systems support can change the values by means of M-S-P)
X 'C4'	The index specified for the character set pool is greater than the number of character sets in the pool
X 'C8'	The page rotation module is available, but there is no ROTLOOP definition in the specified form
X 'CC'	A new TSN cannot be assigned
X 'D4'	No information on user job; system error
X 'DC'	The specified library is not a PLAM library
X 'E0'	The specified library is not available
X 'E4'	The specified element has not been found
X 'E8'	MONJV error
X 'EC'	Error in authorization for processing
X 'F0'	Distributed jobs are not permitted
X 'FC'	Internal error
X 'FF'	Interface version is not supported

(part 2 of 2)

Parameter list description and assembly language example

The PRNTDOC macro can be called in assembly language in accordance with the format description by specifying all the operands directly or by specifying a parameter list containing the operand values. The example below, which follows the field description of the macro, shows the possible calls of the macro.

Parameter list description

* parameter list description					
SCPAPL DS	OF				
SCPAHDR FHDR	MF=(C,SCPA),EQUATES=NO	Standard header			
<pre>* main return</pre>	n codes				
SCPAOK EQU	0	successful processing			
SCPAW048 EQU	72	warning			
SCPAE001 EQU	1	invalid filename			
SCPAE002 EQU	2	invalid pl access			
SCPAE008 EQU	8	inconsistency between param			
SCPAE024 EQU	36	invalid parameter spec			
SCPAE030 EQU	48	recform=F not allowed with			
*		recsize=0			
SCPAE078 EQU	120	print *sysout not allowed			
SCPAE07C EQU	124	l-spac=ebcdic and recform U			
SCPAE080 EQU	128	fgg not allowed			
SCPAE090 EQU	144	invalid or missing parameters			
*		for HP-HP90			
SCPAE098 EQU	152	system file assigned to			
*		*dummy			
SCPAE09C EQU	156	system file assigned to tape			
SCPAEOAO EQU	160	system file assigned to			
*		temporary one			
SCPAE0A4 EQU	164	system file not assigned to			
*		SAM file			
SCPAEODC EQU	220	not a plam library			
SCPAWVER EQU	65535	interface version not			
*		supported			
SCPAE01C EQU	28	memory request error			
SCPAE020 EQU	32	memory release error			
SCPAE004 EQU	4	queue routine error			
SCPAEOD4 EQU	212	no information on user job			
SCPAEOFC EQU	252	internal error			
SCPAEOOC EQU	12	DMS error			
SCPAE010 EQU	16	error in EAM processing			
SCPAE018 EQU	24	invalid or missing password			

SCPAE02C	EQU	44	invalid file access
SCPAE034	EQU	52	creation-expiration date not
*			available
SCPAE040	EQU	64	EAM file not available
SCPAE044	EQU	68	syst file contains no data
SCPAE054	EQU	84	DELETE-FILE command active
SCPAE06C	EQU	108	syst file close error
SCPAE070	EQU	112	file currently in use
SCPAE088	EQU	136	destination does not exist
SCPAE0A8	EQU	168	file access not available
SCPAE0B4	EQU	180	form does not exist
SCPAE0B8	EQU	184	SAM error in EAM processing
SCPAEOBC	EQU	188	no rotation supported
SCPAE0C0	EQU	192	exceeding char-set-num or FOB
*			size
SCPAE0C4	EQU	196	exceeding char-set-num in
*			pool
SCPAE0C8	EQU	200	no rotation-loop defined
SCPAEOCC	EQU	204	new TSN can not be assigned
SCPAE0E0	EQU	224	PLAM library not available
SCPAE0E4	EQU	228	library element not found
SCPAE0E8	EQU	232	monjv error
SCPAE0EC	EQU	236	authorization error
SCPAE0F0	EQU	240	distribution not allowed
SCPAE094	EQU	148	SPOOL subsystem pended
SCPAFFFF	EQU	65535	S&P subsystems not available
*			
SCPARSOP	DS	A	address of rso_pl
SCPASPSP	DS	A	address of sps_pl
SCPADPP	DS	A	address of dprint_pl
SCPASEMP	DS	A	address of dsem_pl
SCPARES2	DS	A	free space
SCPARES3	DS	A	free space
SCPARES4	DS	A	free space
SCPADTY		Н	type of the document to print
* _dtyp			
SCPADTYF		1	print a file
SCPADTYL	EQU	2	print a library-element
SCPADTYP	EQU	3	print a POSIX-file
SCPADTYE	EQU	4	print a EAM-file
SCPADTYS	EQU	5	print a system file
*	DC	11	
SCPADNAL	DS	Н	Length of the name of the
*			document(s) to be printed

SCPADNAP *	DS	A	Address of the name of the document(s) to be printed
SCPAEAMN *	DS	16F	EAM-file-number(s) to be printed
SCPASYSN *	DS	16H	*SYSOUT/SYSLSTnn to be printed
*			
SCPALIBE *	DS	0XL132	Library element(s) to be printed
SCPAELNA	DS	CL80	element name
SCPAELVE	DS	CL40	element version
SCPAELTY	DS	CL12	element type
*		0.11	erement type
SCPASECR	DS	0XL64	Identifier of the optional
*			section records
SCPASEID	DS	CL60	section id
SCPASEP0	DS	Н	section id position
SCPASELE	DS	Χ	section id length
SCPASEIN		AL1	sect ind
SCPASEGI		X,80,	This bit must be set if
*			sect-id is given, i.e. if it
*			is not *NONE.
SCPASEIC	EOU	X'40'	section-id is given in
*			character (= true) or
*			hexadecimal (= false). This
*			field serves to indicate in
*			which format the section-id
*			should be printed on the
*			trailer-page.
SCPASERF	EQU	X'3F'	not used
*			
*			
SCPAFREC	DS	0XL72	Beginning part of the
*			document(s) to be printed
SCPASPOS	DS	F	record number
SCPASSID	DS	CL60	string-id
SCPASSP0	DS	Н	string-id position
SCPASTOC	DS	Н	string-id occurrence
SCPASSLE	DS	Χ	string-id length
SCPAFIL6	DS	CL1	not used
SCPASSII	DS	Y	strid_ind
SCPASSIC	EQU	X,8000,	string-id is given in
*	_ ~ 0		character (= true) or
			2 4000. (0.40) 01

```
*
                                         hexadecimal (= false). This
                                          field serves to indicate in
                                         which format the section-id
                                          should be printed on the
                                         trailer-page.
SCPASSIF EOU
                X'7FFF'
                                         not used
SCPALREC DS
                0XL72
                                          End part of the document(s)
                                         to be printed
                                          last record number
SCPAEPOS DS
SCPAESID DS
                CI 60
                                         strina-id
SCPAESPO DS
                                          string-id position
SCPAETOC
         DS
                                          string-id occurrence
                Н
SCPAESLE DS
                                          string-id length
                Χ
SCPAFIL7 DS
                CL1
                                         not used
SCPAESII DS
                γ
                                         strid_ind
SCPAEEIC EQU
              X,8000,
                                          string-id is given in
                                          character (= true) or
                                          hexadecimal (= false). This
                                         field serves to indicate in
                                         which format the section-id
                                          should be printed on the
                                         trailer-page.
SCPAESIF EQU
                X'7FFF'
                                         not used
SCPARECP DS
                                         Part of the input record to
                0XL4
                                         be processed
SCPARPEC DS
                                         first character
SCPARPLC DS
                                          last character
                Н
SCPALISP DS
                0XI4
                                         Number of line feeds after a
                                          line has been printed
SCPACCPO DS
                                         control char position
SCPASPAC DS
                FL1
                                          line spacing
    _space_set
SCPASPA1 EQU
                1
                                          spacing = *space_1
SCPASPA2 EQU
                2
                                          spacing = *space_2
                                          spacing = *space_3
SCPASPA3 EQU
SCPASPAF FOU
                8
                                          spacing = *by_ebcdic_control
SCPASPAA EQU
              16
                                          spacing = *by_asa_control
SCPASPAI
         EQU
                32
                                          spacing = *by ibm control
```

```
SCPASPAN EQU 144
                                       spacing = *no
SCPASPAS EQU 0
                                       spacing = *std
SCPAFIL4 DS CL1
                                       not used
SCPADOCF DS
              FL1
                                       document format
* _docform_set
SCPADCFT EQU
             0
                                       docform = *text
SCPADCFP EOU
               1
                                       docform = *page-format
SCPADCFS EQU
                                       docform = *special-format
SCPACONM DS OXL3
                                       Control mode applied by the
                                       Spool & Print subsystem on
                                       the document(s) copies
SCPACTLM DS
            FL1
                                       control mode
* _control_mode_set
SCPACTMS EQU
             1
                                       mode = *std
SCPACTPH EQU
                                       mode = *physical
SCPACTMP EQU
                                       mode = *page-mode
             4
SCPACTML EQU
                                       mode = *logical
               8
SCPACTLI EQU
             16
                                       mode = *line-mode
SCPACTMA EQU
                                       mode = *apa
               64
SCPAPCC DS FL1
                                       page control character
* _pcc_set
SCPATOPY EQU
               0
                                       pcc = *yes
SCPATOPN EQU
             1
                                       pcc = *no
SCPATOPD FOU
                                       pcc = *dummy
SCPACTLT DS FL1
                                       control type
* _control_type_set
SCPACTDU EQU 0
                                       ctltype = *dummy
SCPACTHC EQU
               1
                                       ctltype = *compatible
                                       ctltype = *hp
SCPACTHP EQU
SCPAFAM DS FL1
                                       a common TSN is allocated if
                                       a number of files or library
                                       elements are specified
   _family_set
SCPAFAMS EQU 1
                                       family = *std
SCPAFAMY EQU
                                       family = *yes
SCPAFAMN EQU 3
                                       family = *no
```

```
*
                                         Checkpoint processing is to
SCPACHKP DS
               FL1
                                         be performed by the
*
                                         controller on the basis of
                                         pages or section
    _checkp_set
SCPACKPP EOU
                0
                                         checkp = *on-page
SCPACKPR EOU
                1
                                         checkp = *on-section-record
SCPADOCN DS
                Н
                                         Number of files
                F
                                         ONLY TPR: Time of printout
SCPASPV
          DS
                                         for the system file prior to
                                         termination of the job
SCPASPS DS
                FL1
                                         ONLY TPR: specifies whether
                                         time of printout for the
                                         system file is prior to
*
                                         termination of the job
    _stproc_set
SCPASTPV FOU
                                         stproc = value TPR only
                0
SCPASTPN EQU
                1
                                         stproc = no TPR only
                                         stproc = close TPR only
SCPASTPC EQU
                2
SCPASTPI EQU
                                         stproc = immediate TPR only
SCPALOCK DS
                FI1
                                         Lock while job is waiting
   _lockf_set
SCPALOCS EQU
                                         lockf = *std
                0
SCPALOCN EQU
                1
                                         lockf = *no
SCPALOCY EQU
                2
                                         lockf = *yes
SCPADELE DS
                FI1
                                         Delete (destroy) document
* _delf_set
SCPADELY EQU
                1
                                         delf = *yes
SCPADELD EQU
                2
                                         delf = *destroy
                                         delf = *no
SCPADELN EQU
SCPACOPY DS
                χ
                                         how many additional times the
                                         print-job must be repeated
                                         (number of additional
                                         document(s) copies)
SCPAJVP
          DS
                0XI4
                                         jvpass
SCPAJVPI DS
                F
                                         Password with which the job
                                         variable is protected
                                         (numeric)
         ORG
               SCPAJVP
```

SCPAJVPW	DS	CL4	Password with which the job
*			variable is protected
	ORG	SCPAJVP+4	
SCPAMJV	DS	CL54	Job variable into which
*			information on print-job
*			processing is to be stored
SCPAPRJN	DS	CL8	Job-name of the print-job
SCPAPRIO	DS	Χ	Priority of the print-job
*			
SCPAPRAD	DS	0XL24	Allows to execute the
*			print-job under another
*	D.C.	01.0	user-id
SCPAPRAU *	DS	CL8	user-id under which the
	DC	CLO	print-job is launched
SCPAPRAA *	D2	CL8	account number under which
SCPAPRAP	Dς	CL8	the print-job is launched
*	DS	CLO	password for the user-id given with the <user></user>
*			parameter
*			pur une cer
SCPATRNC	DS	FL1	Line truncation
* _trur	ncatio	n_set	
SCPATRUS	EQU	0	trunc = *std
SCPATRUD	EQU	1	<pre>trunc = *delete_file</pre>
SCPATRUK	EQU	2	trunc = *keep_file
*			
*			
SCPAPART	DS	0XL12	layout control (number of
*			pages or lines)
SCPARCFR	DS	F	from record
SCPARCTO	DS	F	till record
SCPALAYD		FL1	dimension
		rt_dim_set	
SCPALAYN	EQU	0	dim = *none
SCPALAYP	EQU	1	dim = *pages
SCPALAYL *	EQU	2	dim = *line
SCPAFIL5	DS	CL3	not used
*			
SCPACLAS	DS	Χ	job class
SCPALFTM	DS	Χ	Indentation for the output
*			text
SCPALIPP	DS	Н	Lines per page (including
*			header and blanks)

```
SCPAHDLI DS
               FL1
                                        Type of header line
   headline set
SCPAHLNO EOU
               1
                                        headlin = *no
                                        headlin = *std
SCPAHLST EOU
                                        headlin = *frecord
SCPAHLR EOU
SCPAHLD
         EQU
                                        headlin = *date
SCPAHLDR EOU
              12
                                        headlin = *date frecord
SCPAHLP
         EQU
              16
                                        headlin = *page
SCPAHLRP EOU
               20
                                        headlin = *frecord page
SCPAHLDP EOU
               24
                                        headlin = *date page
                                        headlin = *date frecord page
SCPAHDRP EOU
               28
SCPAOUTF DS
               FL1
                                        The output format is
                                        character format only or
                                        character and hexadecimal
                                        format
   _outform_set
SCPAOUTC EQU
                                        outform = *character
               1
SCPAOUTH FOU
                                         outform = *hexadecimal
SCPATWSD DS
               FL1
                                        The document is to be printed
                                        on one or both paper sides
                                         (supported only on LP65
                                        printers)
   _twoside_set
SCPATWOS EQU
                                        twoside = *std
SCPATWOY EQU
               1
                                        twoside = *yes
                                        twoside = *tumble
SCPATWOT EQU
               2
SCPATWON FOU
               4
                                         twoside = *no
                                        twoside = *ignore
SCPATWOI FOU
SCPAROT
         DS
               FL1
                                        The pages to be printed from
                                         the print-job are rotated
   _rotation_set
SCPAROTN EQU
              213
                                         rot = *no
SCPAROTO EQU
               240
                                         rot = *rot_0
SCPAROT1 EQU
              241
                                         rot = *rot_90
SCPAROT2 EQU
              242
                                         rot = *rot 180
SCPAROT3 EQU
             243
                                         rot = *rot_270
SCPAROT4 EQU
              244
                                         rot = *rot_0_180
SCPAROT5 EQU
              245
                                         rot = *rot_90_270
SCPAROT6 FOU
              246
                                         rot = *rot_180_0
SCPAROT7 EQU
              247
                                         rot = *rot_270_90
SCPAROTY
        EQU
               232
                                         rot = *by control codes
```

```
SCPAIT# DS
               0XL1
                                        intray
SCPAINT DS
               Χ
                                        Paper input tray to be used
                                        (only on LP65 and RS0
                                        printers)
        ORG
              SCPAIT#
SCPAITS DS
              FL1
* intray set
SCPAITBF EOU
                                        intray = *by-format
               0
SCPAITIG FOU
               255
                                        intray = *ignore
              SCPAIT#
        ORG
SCPAITES DS
               FL1
   _intray_format_set
SCPAITMA EQU
              128
                                        intray_format = *manual
SCPAITA3 EQU
              129
                                        intray_format = *a3
SCPAITA4 EQU
             130
                                        intray_format = *a4
             131
                                        intray_format = *a5
SCPAITA5 EQU
SCPATTB4 FOU
              132
                                        intray_format = *b4
SCPAITB5 EQU
              133
                                        intray_format = *b5
SCPAITDB EQU
               134
                                        intray format =
                                        *double-letter
SCPAITEX EQU
              135
                                        intray_format = *exec
SCPAITFO EQU
               136
                                        intray_format = *folio
              137
                                        intray_format = *invoice
SCPAITIV EQU
SCPAITLG EQU
              138
                                        intray_format = *legal
SCPAITLT EQU
              139
                                        intray_format = *letter
SCPAITMO EQU
              140
                                        intray_format = *monarch
SCPAIT10 EQU
              141
                                        intray_format =
                                        *commercial-10
SCPAITDL EQU
              142
                                        intray format = *dl
SCPAITC5 EQU
              143
                                        intray_format = *c5
SCPAIT3U EQU
             144
                                        intray_format = *A3-uncut
SCPAIT4U EQU
              145
                                        intray format = *A4-uncut
                                        intray_format = *ledger
SCPAITLD EQU
              146
        ORG
              SCPAIT#+1
SCPAOT#
         DS
               0XL1
                                        outtray
SCPAOUTT DS
               Χ
                                        Paper output tray to be used
*
                                        (only on LP65 and RS0
                                        printers)
        ORG
              SCPAOT#
         DS
               FL1
SCPAOTS
* _outtray_set
```

SCPAOUTI	EQU	128	outtray = *ignore
SCPAOUTS *	EQU	129	outtray = *sorter
	ORG	SCPAOT#+1	
SCPAPGCP *	DS	Н	Copies of each printed page
SCPACVPG	DS	0XL40	Cover-page
SCPAHTXT	DS	CL32	header page text
SCPAHDEX	DS	F	header exit number
SCPATREX *	DS	F	trailer exit number
SCPAFORM	DS	CL6	Form-name to be used
SCPAL00P	DS	CL3	Loop-name to be used
SCPARTLP	DS	CL3	Rotation-loop name to be used
SCPACHS	DS	16CL3	One or more character-sets
*	FOLL	1.0	used for printing
SCPACHS#	EQU	16 CL4	Pool-name
SCPAPLNM SCPAPLIN	DS DS	X	Pool-index
SCPAFEIN SCPAEFO	DS	^ CL2	Film or EFO overlay for
*	DS	CLZ	printing
*			p
SCPAOVLY *	DS	0XL2	Double sided overlay (only for LP65 printer type)
SCPAOVFA	DS	Χ	face side
SCPAOVRE *	DS	Χ	reverse side
SCPAFOB	DS	CL4	FOB (Forms Overlay Buffer)
*			data overlay to be used for
*			printing the document (only
*			supported on HP printer
*			equipped with a graphic
*			buffer
	DS	FL1	allow virtual controller
	ual_se		
SCPAMUST	EQU	4	virtual = *must
SCPAALLO SCPANOTA	EQU	2 1	<pre>virtual = *allowed virtual = *not_allowed</pre>
SCPANUTA	EQU	0	virtual = *std
*	LQU	·	
SCPAFIL2	DS	CL2	slack bytes
0001000		_	
SCPAPGPC *	DS	F	PCL file number (only for LP65 printer type)

*			used for print resources
*			(user PRFILE generated by
*			PRM)
*			
SCPATRTB	DS	0XL52	Translation table to be used
SCPATRNA	DS	CL8	translation table name
SCPATRFI	DS	CL44	translation table file
*			
SCPAPRNM	DS	CL8	Printer or printer pool name
*			on which the document is to
*			be printed
SCPAPRTY	DS	FL1	Printer type requested
* _prty	pe_set		
SCPAPTAN	EQU	0	prtype = *any
SCPAPT65		1	prtype = *1p65_printer
SCPAPTAP		2	<pre>prtype = *apa_printer</pre>
SCPAPTHP	EQU	3	<pre>prtype = *hp_printer</pre>
*			
SCPASRTM		FL1	Sorter mode
* _srtm			
SCPASMNO	EQU	1	<pre>srtmode = *no</pre>
SCPASMGR		2	<pre>srtmode = *group</pre>
SCPASMCL		3	<pre>srtmode = *collate</pre>
SCPASMST		4	<pre>srtmode = *stacker</pre>
SCPASMAU *	EQU	5	<pre>srtmode = *automatic</pre>
	DC	CL 2	
SCPAFIL3 SCPATSN	DS DS	CL2 CL4	slack bytes
*	DS	CL4	output: TSN obtained from the system
SCPACLUS	DS	CL8	for compatibility with
*	DS	CLO	MODIFY-PRINT-JOB
SCPASP1	ns	AL1	specified1
SCPASP11		X,80,	document type
SCPASP12		X '40'	document name
SCPASP13		X'20'	document number
SCPASP14		X'10'	library element
SCPASP15	EQU	X,08,	section record
SCPASP16	EQU	X'04'	first record
SCPASP17	EQU	X'02'	last record
SCPASP18	EQU	X'01'	record part
SCPASP2	DS	AL1	specified2
SCPASP21	EQU	X,80,	line spacing
SCPASP22	EQU	X'40'	document format
SCPASP23	EQU	X'20'	control mode

SCPASP24	EQU	X'10'	additional copies
SCPASP25	EQU	X'08'	family
SCPASP26	EQU	X'04'	checkpoint
SCPASP27	EQU	X'02'	monjv
SCPASP28	EQU	X'01'	jv-password
SCPASP3	DS	AL1	specified3
SCPASP31	EQU	X,80,	print-job-name
SCPASP32	EQU	X'40'	print-job-priority
SCPASP33	EQU	X,50,	processing admission
SCPASP34	EQU	X'10'	line-truncation
SCPASP35	EQU	X'08'	lock-file
SCPASP36	EQU	X'04'	delete-after-print
SCPASP37	EQU	X'02'	output part
SCPASP38	EQU	X'01'	page-copies
SCPASP4	DS	AL1	specified4
SCPASP41	EQU	X.80.	left-margin
SCPASP42	EQU	X'40'	line-per-page
SCPASP43	EQU	X'20'	header-line
SCPASP44	EQU	X'10'	output-format
SCPASP45	EQU	X,08,	two-sided
SCPASP46	EQU	X'04'	rotation
SCPASP47	EQU	X'02'	input-tray
SCPASP48	EQU	X'01'	output-tray
SCPASP5	DS	AL1	specified5
SCPASP51	EQU	X,80,	cover-page
SCPASP52	EQU	X'40'	form-name
SCPASP53	EQU	X'20'	loop-name
SCPASP54	EQU	X'10'	rotation-loop-name
SCPASP55	EQU	X'08'	character-set
SCPASP56	EQU	X'04'	pool-name
SCPASP57	EQU	X'02'	pool-index
SCPASP58	EQU	X'01'	electronic-form-overlay
SCPASP6	DS	AL1	specified6
SCPASP61	EQU	X,80,	overlay
SCPASP62	EQU	X'40'	fob
SCPASP63	EQU	X'20'	1p65-pagedef
SCPASP64	EQU	X'10'	user resources file
SCPASP65	EQU	X,08,	translation table
SCPASP66	EQU	X'04'	printer-name
SCPASP67	EQU	X'02'	printer-type
SCPASP68	EQU	X'01'	tsn
SCPASP7	DS	AL1	specified7
SCPASP71	EQU	X,80,	rso parameters
SCPASP72	EQU	X'40'	sps parameters

SCPASP73 SCPASP74 SCPASP7F SCPASP8 SCPASP81 SCPASP82 SCPASP83 SCPASP84 SCPASP85	EQU EQU DS EQU EQU EQU EQU EQU	X'20' X'10' X'07' AL1 X'80' X'40' X'20' X'10' X'08'	dprint parameters dsem parameters specified8 start-processing sorter mode virtual control job-class prog. name string field
SCPASP87 SCPASP8F *	EQU EQU	X,01,	format name
SCPAVALL SCPAPVNA SCPAPVST *	DS DS DS	0XL40 CL8 CL32	virtual device synchrone virtual program name virtual program string
SCPAFRMT SCPARES9 SCPASP9 SCPASP91 SCPASP92 SCPASP93 SCPASP9F	DS DS DS EQU EQU EQU	CL63 XL2 AL1 X'80' X'40' X'3F'	format name reserved for modify specified9 top-offset left-offset scheduling time
SCPASP9F SCPATOPO * * SCPADATE	DS CL	10	Specifies the length of the white band that must be inserted
* SCPATIME *	CL	5	Specifies the date from which the job can be started Specifies the time from which the job can be started
SCPALEFO * *	DS	Н	Specifies the width of the white band that must be inserted
SCPAUN9 SCPA#	DS EQU	XL11 *-SCPAHDR	unused

```
Example
TC5103
       START
TC5103
       AMODE ANY
TC5103
       RMODE ANY
       GPARMOD 31
       PRINT NOGEN
TC5103
       @ENTR TYP=M,LOCAL=WORKAREA,ENV=SPLSPEC
       @DATA CLASS=B, BASE=R9, DSECT=CONSTANT
            R9.=A(CONSTANT)
       1
            R12,=V(ITSPRV)
       EJECT
*******************
* TEST 1 : FORM RESOURCE
*******************
       MVC
            TESTNR,=CL30'TEST1 (FORM) OF SPGF5103'
       MVC
           TCDESC, TESTNR
       VTCSET LOG
       WROUT PRESENT. TERM. VTSUCBA=VTSUPAR
       WRTRD PARFRM,, VALFRM,,, TERM, PARMOD=31, VTSUCBA=VTSUPAR
       LA
            R4,PLC
       LA
            R5,SCPA#
       ΙA
            R6,PLL
       LR
            R7, R5
       MVCL R4,R6
       PRINT GFN
       IΑ
            R4, FILENAMF
       ST
            R4, POINTER
       PRNTDOC MF=M, DNAME=(POINTER,), FORM=FORMNAME
       PRNTDOC MF=E, PARAM=PLC
       PRINT NOGEN
       UNPK APIRET(9), SCPARET(5)
       MVI APIRET+8, X'40'
            APIRET, HEXNC
       NC
       TR
            APIRET(8), HEXTAB
       WROUT MSGRET, TERM
       WRTRD CONT., KEY.,, TERM, PARMOD=31
       EJECT
*******************
* TEST 2 : LOOP RESOURCE
***********************
       MVC
            TESTNR,=CL30'TEST2 (LOOP) OF SPGF5103'
            TCDESC.TESTNR
       MVC
       WROUT PRESENT, TERM, VTSUCBA=VTSUPAR
       WRTRD PARLOOP,,,VALLOOP,,,,TERM,PARMOD=31,VTSUCBA=VTSUPAR
```

```
LA
            R4.PLC
            R5.SCPA#
       IΑ
             R6.PLL
       LA
       ΙR
             R7.R5
       MVCL R4.R6
       PRINT GFN
       ΙA
            R4, FILENAML
       ST
            R4, POINTER
       PRNTDOC MF=M, DNAME=(POINTER,), LOOP=LOOPNAME
       PRNTDOC MF=E, PARAM=PLC
       PRINT NOGEN
       UNPK APIRET(9), SCPARET(5)
       MVI APIRET+8, X'40'
       NC
            APIRET, HEXNC
       TR
            APIRET(8), HEXTAB
       WROUT MSGRET, TERM
       WRTRD CONT,, KEY,,, TERM, PARMOD=31
       EJECT
*******************
* TEST 3 : ROTATION LOOP RESOURCE
*******************
       MVC
            TESTNR,=CL30'TEST3 (ROTLOOP) OF SPGF5103'
            TCDESC.TESTNR
       MVC
       WROUT PRESENT, TERM, VTSUCBA=VTSUPAR
       WRTRD PARROTL,, VALROTL,,, TERM, PARMOD=31, VTSUCBA=VTSUPAR
            R4, PLC
       ΙA
       LA
            R5.SCPA#
       ΙA
            R6,PLL
       LR
            R7, R5
       MVCL R4,R6
       PRINT GEN
       IΑ
             R4, FILENAMR
             R4, POINTER
       PRNTDOC MF=M, DNAME=(POINTER,), ROTLOOP=ROTLNAME
       PRNTDOC MF=E, PARAM=PLC
       PRINT NOGEN
       UNPK APIRET(9), SCPARET(5)
       MV T
            APIRET+8,X'40'
       NC
            APIRET, HEXNC
            APIRET(8), HEXTAB
       WROUT MSGRET, TERM
       WRTRD CONT., KEY.,, TERM, PARMOD=31
       EJECT
********************
* TEST 4 : CHARACTER SET RESOURCE
***********************
```

```
MVC
               TESTNR,=CL30'TEST4 (CHARSET) OF SPGF5103'
         MVC
               TCDESC.TESTNR
         WROUT PRESENT.TERM.VTSUCBA=VTSUPAR
         WRTRD PARCHAR,, VALCHAR,,, TERM, PARMOD=31, VTSUCBA=VTSUPAR
               R4, PLC
         LA
         ΙA
               R5, SCPA#
               R6,PLL
         ΙA
               R7,R5
         LR
         MVCL R4,R6
         PRINT GEN
               R4,FILENAMC
         ΙA
         ST
               R4, POINTER
         PRNTDOC MF=M, DNAME=(POINTER,), CHARSET=CHARNAME
         PRNTDOC MF=E, PARAM=PLC
         PRINT NOGEN
         UNPK APIRET(9), SCPARET(5)
         MVI APIRET+8, X'40'
         NC
              APIRET, HEXNC
         TR
              APIRET(8), HEXTAB
         WROUT MSGRET, TERM
         WRTRD CONT,, KEY,,, TERM, PARMOD=31
*
TERM
         TERM
         @END
         EJECT
WORKAREA @PAR D=YES
         PRINT GEN
PLC
         PRNTDOC MF=C
         PRINT NOGEN
WORKAREA @PAR LEND=YES
         FJFCT
POINTER DS
            Α
VTSUPAR VTSUCB MODE=EXTEND
PARFRM
         DS
               0Н
         DC
               Y(EPARFRM-PARFRM)
         DS
               3 X
         DC
               AL1(LOGNL)
         DC
               C'FILE NAME : ,
         DC
               AL1(LOGEPA)
         DC
               CL54'
         DC
               AL1(LOGNL)
               C'FORM NAME : ,
         DC
         DC
               AL1(LOGEPA)
         DC
               CL6'.
         DC
               AL1(LOGNL)
```

```
EPARFRM
         EQU
         DS
PARLOOP
                0Н
         DC
                Y(EPARLOOP-PARLOOP)
         DS
                3 X
         DC
                AL1(LOGNL)
         DC
                C'FILE NAME : ,
         DC
                AL1(LOGEPA)
         DC
                CL54',
         DC
                AL1(LOGNL)
                C'LOOP NAME : ,
         DC
         DC
                AL1(LOGEPA)
         DC
                CL3',
         DC
                AL1(LOGNL)
FPARLOOP FOU
PARROTL
         DS
                0H
         DC
                Y(EPARROTL-PARROTL)
         DS
                3 X
         DC
                AL1(LOGNL)
                C'FILE NAME
         DC
         DC
                AL1(LOGEPA)
         DC
                CL54'.
         DC
                AL1(LOGNL)
         DC
                C'ROTATION LOOP NAME : .
         DC
                AL1(LOGEPA)
         DC
                CL3'.
         DC
                AL1(LOGNL)
EPARROTL EQU
PARCHAR DS
                0Н
         DC
                Y(EPARCHAR-PARCHAR)
         DS
                3 X
         DC
                AL1(LOGNL)
         DC
                C'FILE NAME
         DC
                AL1(LOGEPA)
         DC
                CL54'.
         DC
                AL1(LOGNL)
         DC
                C'CHARACTER SET NAME : ,
         DC
                AL1(LOGEPA)
         DC
                CL3'.
         DC
                AL1(LOGNL)
EPARCHAR EQU
PRESENT
         DS
                0Н
         DC
                Y(EPRESENT-PRESENT)
         DS
                3 X
         DC
                AL1(LOGNP)
                700'*'
         DC
         DC
                AL1(LOGNL)
                C.*.
         DC
TCDESC
         DS
                CL30
         DC
                AL1(LOGNL)
         DC
                70C'*'
```

```
DC
                AL1(LOGNL)
EPRESENT EQU
VALFRM
         DS
                0CL64
         DS
                CL4
                CL54
FILENAMF DS
FORMNAME DS
                CL6
                0CL61
VALLOOP
         DS
         DS
                CL4
FILENAML DS
                CL54
LOOPNAME DS
                CL3
VALROTL
         DS
                0CL61
         DS
                CL4
FILENAMR DS
                CL54
ROTLNAME DS
                CL3
VALCHAR DS
                0CL61
         DS
                CL4
FILENAMC DS
                CL54
CHARNAME DS
                CL3
CONT
         DS
                0Н
         DC
                Y(ECONT-CONT)
         DS
                3 X
         DC
                AL1(LOGNL)
         DC
                C'PLEASE ACKNOWLEDGE'
ECONT
         EQU
KEY
         DS
                0CL5
         DS
                CL4
         DS
                CL1
         EJECT
CONSTANT DS
                0Н
         DC
HEXTAB
                C'0123456789ABCDEF'
HEXNC
         DC
                X'0F0F0F0F0F0F0F0F'
MSGRET
         DS
                0F
         DC
                Y(EMSGRET-MSGRET)
         DS
                CL3
         DC
                C'PRNTDOC ERROR RETURN CODE : ,
APIRET
         DS
                CL8
         DC
                C' FOR ,
TESTNR
         DS
                CL30
EMSGRET
         EQU
         EJECT
         PRINT GEN
PLL
         PRNTDOC MF=L
         PRINT NOGEN
         END
```

2.1 The C interface for the functions of the PRNTDOC macro

The C interface implements the same functionality as the PRNTDOC assembly language macro. The individual functions can be specified analogously using C language elements. The comments in the header files listed below are therefore often merely references to the operand description of the PRNTDOC assembly language macro.

The header file and the function call (SVC call) are in the include file "PRNTDOC.H". This file is in the standard SPOOL library shipped with SPOOL V3.0.

The include file is subdivided into the following components:

- return codes (in decimal form)
- constant definitions for specific function groups
- complex data structures (records)
- a parameter list with references to the complex data structures
- a function call

As shown in the sample program after the interface description, the "PRNTDOC.H" file is linked to the application program by means of an include call in a C application program, and the parameter list is made available.

Interface definition for prntdoc.h

```
#ifndef _PRNTDOC_H
#define PRNTDOC H
#if 0
/************************
BEGIN-INTERFACE
                PRNTDOC
TITLE
                 (/ prntdoc /)
NAMF
                 PRNTDOC.H
DOMAIN
                 SP00L
LANGUAGE
COPYRIGHT
                 (C) Siemens AG 1998
                    ALL RIGHTS RESERVED
COMPILATION-SCOPE USER
INTERFACE-TYPE
                CALL
RUN-CONTEXT
                 TU.
                   TPR
PURPOSE
                 (/ Print-job management /)
```

```
END-INTERFACE
                   PRNTDOC.
*****************************
#endif
                                                                       */
/* main return codes
/* mret code
                                                                       */
#define PRNTDOCok 0
                                        /* successful processing
                                                                       */
#define PRNTDOCdelete_ignored 72
                                        /* warning
                                                                       */
#define PRNTDOCinvalid_filename 1
                                        /* invalid filename
                                                                       */
#define PRNTDOCinvalid_access 2
                                        /* invalid pl access
                                                                       */
#define PRNTDOCinconsistency 8
                                        /* inconsistency between
                                                                       */
                                        /* param
                                                                       */
                                        /* invalid parameter spec
                                                                       */
#define PRNTDOCinvalid_param 36
#define PRNTDOCwrong_recform 48
                                        /* recform=F not allowed with */
                                        /* recsize=0
                                                                       */
                                        /* print *sysout not allowed
                                                                      */
#define PRNTDOCinteract_job 120
                                        /* 1-spac=ebcdic and recform
#define PRNTDOCwrong_spacing 124
                                                                      */
                                                                       */
                                                                       */
#define PRNTDOCfgg_not_allowed 128
                                        /* fgg not allowed
#define PRNTDOCwrong HP param 144
                                        /* invalid or missing
                                                                       */
                                        /* parameters for HP-HP90
                                                                       */
                                        /* system file assigned to
                                                                       */
#define PRNTDOCsf_dummy 152
                                        /* *dummy
                                                                       */
#define PRNTDOCsf_tape 156
                                        /* system file assigned to
                                                                       */
                                                                       */
#define PRNTDOCsf temporary 160
                                        /* system file assigned to
                                                                       */
                                        /* temporary one
                                                                       */
                                        /* system file not assigned
                                                                       */
#define PRNTDOCsf_not_sam 164
                                        /* to SAM file
                                                                       */
#define PRNTDOCnot_plam_lib 220
                                        /* not a plam library
                                                                       */
                                                                       */
#define PRNTDOCwrong_version 65535
                                        /* interface version not
                                        /* supported
                                                                       */
#define PRNTDOCregm error 28
                                        /* memory request error
                                                                       */
                                                                       */
#define PRNTDOCrelm_error 32
                                        /* memory release error
#define PRNTDOCqueue error 4
                                        /* queue routine error
                                                                       */
#define PRNTDOCno info 212
                                        /* no information on user job */
#define PRNTDOCinternal_error 252
                                        /* internal error
                                                                       */
#define PRNTDOCdms_error 12
                                        /* DMS error
                                                                       */
#define PRNTDOCeam proc error 16
                                        /* error in EAM processing
                                                                       */
#define PRNTDOCwrong_password 24
                                        /* invalid or missing
                                                                       */
                                        /* password
                                                                       */
#define PRNTDOCinvalid file access 44
                                        /* invalid file access
                                                                       */
#define PRNTDOCdate_not_available 52
                                        /* creation-expiration date
                                                                       */
                                        /* not available
                                                                       */
                                        /* EAM file not available
                                                                      */
#define PRNTDOCeam not available 64
#define PRNTDOCno_data 68
                                        /* syst file contains no data */
#define PRNTDOCdelete_active 84
                                        /* DELETE-FILE command active */
#define PRNTDOCsf close error 108
                                        /* syst file close error
                                                                       */
```

```
#define PRNTDOCfile in use 112
                                        /* file currently in use
                                                                       */
                                        /* destination does not exist */
#define PRNTDOCdest not found 136
#define PRNTDOCf_access_not_avail 168
                                        /* file access not available */
#define PRNTDOCform not found 180
                                        /* form does not exist
                                                                       */
#define PRNTDOCsam error 184
                                        /* SAM error in EAM
                                                                       */
                                        /* processing
                                                                       */
#define PRNTDOCno rotation 188
                                        /* no rotation supported
                                                                       */
#define PRNTDOCwrong res value 192
                                        /* exceeding char-set-num or
                                                                       */
                                        /* FOB size
                                                                       */
#define PRNTDOCwrong_cs_index 196
                                        /* exceeding char-set-num in
                                                                       */
                                        /* pool
                                                                       */
#define PRNTDOCno_rotation_loop 200
                                        /* no rotation-loop defined
                                                                       */
#define PRNTDOCno tsn assigned 204
                                        /* new TSN can not be
                                                                       */
                                        /* assigned
                                                                       */
#define PRNTDOCplam_lib_not_avail 224
                                        /* PLAM library not available */
#define PRNTDOCplam elem not found 228
                                        /* library element not found
                                                                       */
#define PRNTDOCmonjv_error 232
                                        /* monjv error
                                                                       */
#define PRNTDOCauthorization error 236 /* authorization error
                                                                       */
#define PRNTDOCno dist allowed 240
                                        /* distribution not allowed
                                                                       */
#define PRNTDOCspool pended 148
                                        /* SPOOL subsystem pended
                                                                       */
#define PRNTDOCsubsyst_not_avail 65535 /* S&P subsystems not
                                                                       */
                                        /* available
                                                                       */
/* _dtype_set
                                                                       */
/* ENUM _dtype_set
                                                                       */
#define PRNTDOCdtype file 1
                                       /* print a file
                                                                       */
#define PRNTDOCdtype_libel 2
                                       /* print a library-element
                                                                       */
#define PRNTDOCdtype posix 3
                                       /* print a POSIX-file
                                                                       */
#define PRNTDOCdtype eam 4
                                       /* print a EAM-file
                                                                       */
#define PRNTDOCdtype_sysf 5
                                        /* print a system file
                                                                       */
                                                                       */
/* family set
/* ENUM family set
                                                                       */
#define PRNTDOCfamily_std 1
                                                                       */
                                       /* family = *std
#define PRNTDOCfamily yes 2
                                       /* family = *yes
                                                                       */
#define PRNTDOCfamily no 3
                                        /* family = *no
                                                                       */
/* checkp set
                                                                       */
/* ENUM _checkp_set
                                                                       */
#define PRNTDOCon_pages 0
                                        /* checkp = *on-page
                                                                       */
#define PRNTDOCon_section_records 1
                                        /* checkp =
                                                                       */
                                        /* *on-section-record
                                                                       */
/* _lockf_set
                                                                       */
/* ENUM _lockf_set
                                                                       */
#define PRNTDOClockf_std 0
                                                                       */
                                       /* lockf = *std
#define PRNTDOClockf_no 1
                                       /* lockf = *no
                                                                       */
#define PRNTDOClockf ves 2
                                        /* lockf = *ves
                                                                       */
```

```
*/
/* delf set
/* ENUM _delf_set
                                                                     */
#define PRNTDOCdelf yes 1
                                      /* delf = *yes
                                                                     */
#define PRNTDOCdelf_destr 2
                                      /* delf = *destroy
                                                                     */
#define PRNTDOCdelf no 3
                                       /* delf = *no
                                                                     */
/* _outform_set
                                                                     */
/* ENUM _outform_set
                                                                     */
#define PRNTDOCout_character 1
                                      /* outform = *character
                                                                     */
#define PRNTDOCout hexadecimal 2
                                      /* outform = *hexadecimal
                                                                     */
/* _twoside_set
                                                                     */
/* ENUM _twoside_set
                                                                     */
#define PRNTDOCtwoside_std 0
                                      /* twoside = *std
                                                                     */
#define PRNTDOCtwoside yes 1
                                      /* twoside = *yes
                                                                     */
#define PRNTDOCtwoside_tumble 2
                                      /* twoside = *tumble
                                                                     */
                                      /* twoside = *no
#define PRNTDOCtwoside_no 4
                                                                     */
#define PRNTDOCtwoside ignore 8
                                       /* twoside = *ignore
                                                                     */
                                                                     */
/* _rotation_set
/* ENUM _rotation_set
                                                                     */
#define PRNTDOCrotation no 213
                                      /* rot = *no
                                                                     */
#define PRNTDOCrotation_0 240
                                      /* rot = *rot 0
                                                                     */
#define PRNTDOCrotation_90 241
                                      /* rot = *rot 90
                                                                     */
#define PRNTDOCrotation 180 242
                                      /* rot = *rot 180
                                                                     */
#define PRNTDOCrotation_270 243
                                       /* rot = *rot_270
                                                                     */
#define PRNTDOCrotation_0_180 244
                                       /* rot = *rot_0_180
                                                                     */
#define PRNTDOCrotation_90_270 245
                                      /* rot = *rot 90 270
                                                                     */
#define PRNTDOCrotation_180_0 246
                                       /* rot = *rot_180_0
                                                                     */
#define PRNTDOCrotation_270_90 247
                                       /* rot = *rot_270_90
                                                                     */
#define PRNTDOCby control codes 232
                                       /* rot = *by control codes
                                                                     */
                                                                     */
/* _intray_set
/* ENUM _intray_set
                                                                     */
#define PRNTDOCintray by format 0
                                      /* intray = *by-format
                                                                     */
#define PRNTDOCintray_ignore 255
                                       /* intray = *ignore
                                                                     */
/* intray format set
                                                                     */
/* ENUM _intray_format_set
                                                                     */
#define PRNTDOCitf_manual 128
                                      /* intray_format = *manual
                                                                     */
#define PRNTDOCitf a3 129
                                      /* intray format = *a3
                                                                     */
#define PRNTDOCitf_a4 130
                                      /* intray_format = *a4
                                                                     */
#define PRNTDOCitf_a5 131
                                      /* intray_format = *a5
                                                                     */
#define PRNTDOCitf b4 132
                                       /* intray format = *b4
                                                                     */
#define PRNTDOCitf_b5 133
                                       /* intray_format = *b5
                                                                     */
#define PRNTDOCitf_double_letter 134
                                       /* intray_format =
                                                                     */
                                       /* *double-letter
                                                                     */
```

```
#define PRNTDOCitf exec 135
                                        /* intray format = *exec
                                                                      */
#define PRNTDOCitf folio 136
                                        /* intray format = *folio
                                                                      */
#define PRNTDOCitf_invoice 137
                                        /* intray_format = *invoice
                                                                      */
#define PRNTDOCitf_legal 138
                                        /* intray format = *legal
                                                                      */
#define PRNTDOCitf letter 139
                                        /* intray format = *letter
                                                                      */
#define PRNTDOCitf_monarch 140
                                        /* intray_format = *monarch
                                                                      */
#define PRNTDOCitf_commercial_10 141
                                        /* intray format =
                                                                      */
                                        /* *commercial-10
                                                                      */
#define PRNTDOCitf dl 142
                                        /* intray_format = *dl
                                                                      */
#define PRNTDOCitf_c5 143
                                        /* intray_format = *c5
                                                                      */
#define PRNTDOCitf A3U 144
                                        /* intray_format = *A3-uncut
                                                                      */
#define PRNTDOCitf_A4U 145
                                       /* intray_format = *A4-uncut
                                                                      */
#define PRNTDOCitf_ledger 146
                                        /* intray format = *ledger
                                                                      */
                                                                      */
/* _outtray_set
/* ENUM _outtray_set
                                                                      */
#define PRNTDOCouttray_ignore 128
                                       /* outtray = *ignore
                                                                      */
#define PRNTDOCouttray_sorter 129
                                        /* outtray = *sorter
                                                                      */
                                                                      */
/* srtmode set
/* ENUM _srtmode_set
                                                                      */
#define PRNTDOCsrtmode no 1
                                        /* srtmode = *no
                                                                      */
#define PRNTDOCsrtmode group 2
                                       /* srtmode = *group
                                                                      */
#define PRNTDOCsrtmode_collate 3
                                        /* srtmode = *collate
                                                                      */
#define PRNTDOCsrtmode stacker 4
                                        /* srtmode = *stacker
                                                                      */
#define PRNTDOCsrtmode automatic 5
                                        /* srtmode = *automatic
                                                                      */
                                                                      */
/* _docform_set
/* ENUM docform set
                                                                      */
                                        /* docform = *text
#define PRNTDOCdocform_text 0
                                                                      */
#define PRNTDOCdocform_page_format 1
                                        /* docform = *page-format
                                                                      */
#define PRNTDOCdocform spec format 2
                                        /* docform = *special-format
                                                                      */
                                                                      */
/* _control_mode_set
/* ENUM _control_mode_set
                                                                      */
#define PRNTDOCctl mode std 1
                                        /* mode = *std
                                                                      */
#define PRNTDOCctl_mode_phys 2
                                        /* mode = *physical
                                                                      */
#define PRNTDOCctl_mode_page_m 4
                                        /* mode = *page-mode
                                                                      */
#define PRNTDOCctl mode logic 8
                                        /* mode = *logical
                                                                      */
#define PRNTDOCctl_mode_line_m 16
                                        /* mode = *line-mode
                                                                      */
#define PRNTDOCctl_mode_apa 64
                                        /* mode = *apa
                                                                      */
                                                                      */
/* _pcc_set
/* ENUM _pcc_set
                                                                      */
                                                                      */
#define PRNTDOCtop_mode_yes 0
                                       /* pcc = *ves
                                        /* pcc = *no
#define PRNTDOCtop_mode_no 1
                                                                      */
#define PRNTDOCtop_mode_dummy 2
                                        /* pcc = *dummy
                                                                      */
```

```
*/
/* stproc set
/* ENUM stproc set
                                                                      */
#define PRNTDOCstproc_value 0
                                      /* stproc = value TPR only
                                                                      */
#define PRNTDOCstproc no 1
                                       /* stproc = no TPR only
                                                                      */
                                      /* stproc = close TPR only
                                                                     */
#define PRNTDOCstproc close 2
#define PRNTDOCstproc imm 4
                                       /* stproc = immediate TPR
                                                                      */
                                       /* only
                                                                      */
/* _space_set
                                                                      */
/* ENUM _space_set
                                                                      */
#define PRNTDOCspace 1 1
                                      /* spacing = *space_1
                                                                      */
#define PRNTDOCspace_2 2
                                      /* spacing = *space_2
                                                                      */
#define PRNTDOCspace 3 4
                                       /* spacing = *space 3
                                                                      */
                                       /* spacing =
#define PRNTDOCspace_e 8
                                                                      */
                                       /* *by_ebcdic_control
                                                                      */
                                       /* spacing = *by_asa_control
                                                                     */
#define PRNTDOCspace a 16
#define PRNTDOCspace_i 32
                                      /* spacing = *by_ibm_control
                                                                     */
#define PRNTDOCspace_no 144
                                       /* spacing = *no
                                                                     */
#define PRNTDOCspace std 0
                                       /* spacing = *std
                                                                      */
                                                                      */
/* _control_type_set
/* ENUM _control_type_set
                                                                      */
#define PRNTDOCctl type dummy 0
                                      /* ctltype = *dummy
                                                                     */
#define PRNTDOCctl_type_hp_com 1
                                      /* ctltype = *compatible
                                                                      */
#define PRNTDOCctl_type_hp 2
                                       /* ctltype = *hp
                                                                      */
                                                                      */
/* _headline_set
/* ENUM _headline_set
                                                                      */
#define PRNTDOCheadline no 1
                                      /* headlin = *no
                                                                      */
#define PRNTDOCheadline_std 2
                                       /* headlin = *std
                                                                     */
#define PRNTDOCheadline r 4
                                       /* headlin = *frecord
                                                                     */
#define PRNTDOCheadline d 8
                                       /* headlin = *date
                                                                     */
#define PRNTDOCheadline dr 12
                                      /* headlin = *date frecord
                                                                     */
                                       /* headlin = *page
#define PRNTDOCheadline_p 16
                                                                     */
#define PRNTDOCheadline rp 20
                                      /* headlin = *frecord_page
                                                                     */
#define PRNTDOCheadline dp 24
                                       /* headlin = *date page
                                                                     */
#define PRNTDOCheadline_drp 28
                                       /* headlin =
                                                                      */
                                       /* *date_frecord_page
                                                                      */
                                                                      */
/* _truncation_set
/* ENUM _truncation_set
                                                                      */
                                                                      */
#define PRNTDOCtrunc std 0
                                       /* trunc = *std
#define PRNTDOCtrunc_delete_file 1
                                     /* trunc = *delete_file
                                                                      */
#define PRNTDOCtrunc_keep_file 2
                                       /* trunc = *keep_file
                                                                      */
```

```
*/
/* layout part dim set
/* ENUM layout part dim set
                                                                      */
#define PRNTDOClayout_none 0
                                       /* dim = *none
                                                                      */
#define PRNTDOClayout pages 1
                                       /* dim = *pages
                                                                      */
#define PRNTDOClayout lines 2
                                       /* dim = *line
                                                                      */
/* _prtype_set
                                                                      */
/* ENUM _prtype_set
                                                                      */
#define PRNTDOCprtype any 0
                                       /* prtype = *any
                                                                      */
#define PRNTDOCprtype_lp65 1
                                       /* prtype = *lp65_printer
                                                                      */
#define PRNTDOCprtype_apa 2
                                       /* prtype = *apa printer
                                                                      */
#define PRNTDOCprtype_hp 3
                                        /* prtype = *hp printer
                                                                      */
                                                                      */
/* virtual set
/* ENUM virtual set
                                                                      */
#define PRNTDOCvirtual_must 4
                                                                      */
                                       /* virtual = *must
#define PRNTDOCvirtual allo 2
                                       /* virtual = *allowed
                                                                      */
#define PRNTDOCvirtual nota 1
                                       /* virtual = *not allowed
                                                                      */
#define PRNTDOCvirtual stda 0
                                        /* virtual = *std
                                                                      */
                                                                      */
/* _libelem_mdl
struct PRNTDOC_libelem_mdl {
        char element[80];
                                       /* element name
                                                                      */
        char version[40]:
                                       /* element version
                                                                      */
        char type[12];
                                       /* element type
                                                                      */
} :
                                                                      */
/* _sectrec_mdl
struct PRNTDOC sectrec mdl {
        char sect_id[60];
                                       /* section id
                                                                      */
        unsigned short sect_pos;
                                       /* section id position
                                                                      */
        unsigned char sect len;
                                       /* section id length
                                                                      */
                                                                      */
        /* sect_ind
        struct /* sect ind */ {
                                        /* This bit must be set if
                                                                      */
                char sect given: 1;
                                        /* sect-id is given, i.e. if
                                                                      */
                                        /* it is not *NONE.
                                                                      */
                char input_char: 1;
                                        /* section-id is given in
                                                                      */
                                        /* character (= true) or
                                                                      */
                                        /* hexadecimal (= false).
                                                                      */
                                        /* This field serves to
                                                                      */
                                        /* indicate in which format
                                                                      */
                                        /* the section-id should be
                                                                      */
                                                                      */
                                        /* printed on the
                                        /* trailer-page.
                                                                      */
                char sectrec_free: 6;
                                        /* not used
                                                                      */
        } sect ind;
```

```
} :
/* _firsrec_mdl
                                                                        */
struct PRNTDOC firsrec mdl {
                                                                        */
        unsigned long stpos;
                                        /* record number
        char ststrid[60];
                                        /* string-id
                                                                        */
        unsigned short ststrpos;
                                        /* string-id position
                                                                        */
        unsigned short ststrocc;
                                        /* string-id occurrence
                                                                        */
        unsigned char ststrlen:
                                        /* string-id length
                                                                        */
        char filler6[1]:
                                         /* not used
                                                                        */
        /* strid_ind
                                                                        */
        struct /* strid ind */ {
                unsigned short st_char: 1;
                                         /* string-id is given in
                                                                        */
                                         /* character (= true) or
                                                                        */
                                         /* hexadecimal (= false).
                                                                        */
                                         /* This field serves to
                                                                        */
                                         /* indicate in which format
                                                                        */
                                         /* the section-id should be
                                                                        */
                                                                        */
                                         /* printed on the
                                                                        */
                                         /* trailer-page.
                unsigned short ststrid free: 15;
                                         /* not used
                                                                        */
        } strid_ind;
} :
                                                                        */
/* _lastrec_mdl
struct PRNTDOC lastrec mdl {
        unsigned long enpos;
                                         /* last record number
                                                                        */
                                         /* string-id
        char enstrid[60]:
                                                                        */
        unsigned short enstrpos;
                                        /* string-id position
                                                                        */
        unsigned short enstrocc;
                                        /* string-id occurrence
                                                                        */
        unsigned char enstrlen;
                                        /* string-id length
                                                                        */
                                         /* not used
        char filler7[1]:
                                                                        */
        /* strid_ind
                                                                        */
        struct /* strid ind */ {
                unsigned short en char: 1;
                                         /* string-id is given in
                                                                        */
                                         /* character (= true) or
                                                                        */
                                         /* hexadecimal (= false).
                                                                        */
                                         /* This field serves to
                                                                        */
                                         /* indicate in which format
                                                                        */
                                         /* the section-id should be
                                                                        */
                                         /* printed on the
                                                                        */
                                         /* trailer-page.
                                                                        */
                unsigned short enstrid free: 15;
```

```
/* not used
                                                                        */
        } strid ind;
} :
/* recpart mdl
                                                                        */
struct PRNTDOC_recpart_mdl {
        unsigned short first ch;
                                       /* first character
                                                                        */
        unsigned short last ch;
                                        /* last character
                                                                        */
} :
/* _linesp_mdl
                                                                        */
struct PRNTDOC_linesp_mdl {
                                                                        */
        unsigned short cc_pos;
                                        /* control char position
        unsigned char spacing;
                                        /* line spacing
                                                                        */
        char filler4[1];
                                         /* not used
                                                                        */
};
/* _contmod_mdl
                                                                        */
struct PRNTDOC_contmod_md1 {
                                                                        */
        unsigned char ctlmode;
                                        /* control mode
        unsigned char pcc;
                                        /* page control character
                                                                        */
        unsigned char ctltype;
                                        /* control type
                                                                        */
} :
/* _procadm_ mdl
                                                                        */
struct PRNTDOC procadm mdl {
        char user[8]:
                                         /* user-id under which the
                                                                        */
                                         /* print-job is launched
                                                                        */
                                         /* account number under which */
        char account[8]:
                                         /* the print-job is launched */
                                         /* password for the user-id
                                                                        */
        char password[8];
                                         /* given with the <user>
                                                                        */
                                         /* parameter
                                                                        */
}:
/* outpart mdl
                                                                        */
struct PRNTDOC_outpart_mdl {
        long rec from;
                                        /* from record
                                                                        */
        unsigned long rec to;
                                         /* till record
                                                                        */
        unsigned char dim;
                                                                        */
                                         /* dimension
        char filler5[3]:
                                         /* not used
                                                                        */
};
/* _covpage_mdl
                                                                        */
struct PRNTDOC covpage mdl {
                                                                        */
        char headtext[32];
                                         /* header page text
        long headexit;
                                         /* header exit number
                                                                        */
                                         /* trailer exit number
        long traiexit;
                                                                        */
```

```
} :
/* _overlay_mdl
                                                                        */
struct PRNTDOC overlay mdl {
                                                                        */
        unsigned char face;
                                       /* face side
        unsigned char reverse;
                                        /* reverse side
                                                                        */
} :
/* _transtab_mdl
                                                                        */
struct PRNTDOC_transtab_mdl {
        char name[8];
                                        /* translation table name
                                                                        */
        char file[44]:
                                        /* translation table file
                                                                        */
} :
                                                                        */
/* _transtab_mdl
struct PRNTDOC_virtual_mdl {
        char name[8];
                                        /* virtual program name
                                                                        */
        char string[32];
                                        /* virtual program string
                                                                        */
} :
                                                                        */
/* parameter list description
struct PRNTDOC_pl_mdl {
        /* Standard header
                                                                        */
        struct ESMFHDR hdr:
                                        /* address of rso pl
                                                                       */
        void* rsopar;
                                        /* address of sps_pl
                                                                       */
        void* spspar;
                                        /* address of dprint_pl
        void* dppar;
                                                                       */
                                       /* address of dsem pl
        void* dsempar;
                                                                       */
                                       /* free space
        void* reserved 2:
                                                                        */
        void* reserved 3:
                                       /* free space
                                                                        */
        void* reserved 4;
                                       /* free space
                                                                        */
                                        /* type of the document to
                                                                       */
        unsigned short dtype;
                                                                        */
                                        /* print
        unsigned short dname len;
                                        /* Length of the name of the
                                                                       */
                                        /* document(s) to be printed */
        void* dname_ptr;
                                        /* Address of the name of the */
                                        /* document(s) to be printed */
        unsigned long eamn[16];
                                         /* EAM-file-number(s) to be
                                                                       */
                                         /* printed
                                                                        */
        unsigned short sysnn[16];
                                         /* *SYSOUT/SYSLSTnn to be
                                                                       */
                                         /* printed
                                                                       */
        struct PRNTDOC_libelem_mdl libelem;
                                         /* Library element(s) to be
                                                                        */
                                                                       */
                                         /* printed
        struct PRNTDOC_sectrec_mdl sectrec;
                                         /* Idenfier of the optional
                                                                        */
                                         /* section records
                                                                        */
```

```
struct PRNTDOC firsrec mdl firsrec;
                                 /* Beginning part of the
                                                                */
                                 /* document(s) to be printed
                                                                */
struct PRNTDOC lastrec mdl lastrec;
                                 /* End part of the
                                                                */
                                 /* document(s) to be printed
                                                                */
struct PRNTDOC recpart mdl recpart;
                                 /* Part of the input record
                                                                */
                                 /* to be processed
                                                                */
struct PRNTDOC_linesp_mdl linesp;
                                 /* Number of line feeds after */
                                 /* a line has been printed
                                                                */
unsigned char docform:
                                 /* document format
                                                                */
struct PRNTDOC contmod mdl contmod;
                                 /* Control mode applied by
                                                                */
                                 /* the Spool & Print
                                                                */
                                 /* subsystem on the
                                                                */
                                 /* document(s) copies
                                                                */
                                 /* a common TSN is allocated
                                                                */
unsigned char family:
                                 /* if a number of files or
                                                                */
                                 /* library elements are
                                                                */
                                 /* specified
                                                                */
unsigned char checkp;
                                 /* Checkpoint processing is
                                                                */
                                 /* to be performed by the
                                                                */
                                 /* controller on the basis of */
                                 /* pages or section
                                                                */
                                 /* Number of files
                                                                */
unsigned short doc_number;
                                 /* ONLY TPR: Time of printout */
long stproc_val;
                                 /* for the system file prior
                                 /* to termination of the job
                                                               */
                                 /* ONLY TPR: specifies
                                                                */
unsigned char stproc;
                                 /* whether time of printout
                                                                */
                                 /* for the system file is
                                                                */
                                 /* prior to termination of
                                                                */
                                 /* the job
                                                                */
                                 /* Lock while job is waiting
                                                                */
unsigned char lockf;
unsigned char delf:
                                 /* Delete (destroy) document
                                                                */
unsigned char addcop;
                                 /* how many additional times
                                                                */
                                 /* the print-job must be
                                                                */
                                 /* repeated (number of
                                                                */
                                 /* additional document(s)
                                                                */
                                 /* copies)
                                                                */
/* jvpass
                                                                */
union /* jvpass */ {
        long jvpassi;
                                 /* Password with which the
                                                                */
                                 /* job variable is protected
                                                                */
                                 /* (numeric)
                                                                */
```

```
char jvpassw[4];
                                 /* Password with which the
                                                                */
                                 /* iob variable is protected
                                                                */
} jvpass;
char monjv[54];
                                 /* Job variable into which
                                                                */
                                 /* information on print-iob
                                                                */
                                 /* processing is to be stored */
char prjname[8];
                                 /* Job-name of the print-job
                                                                */
unsigned char prjprio;
                                 /* Priority of the print-job
                                                                */
struct PRNTDOC_procadm_mdl procadm;
                                 /* Allows to execute the
                                                                */
                                 /* print-job under another
                                                                */
                                 /* user-id
                                                                */
                                 /* Line truncation
                                                                */
unsigned char truncation;
struct PRNTDOC outpart mdl outpart;
                                                                */
                                 /* layout control (number of
                                 /* pages or lines)
                                                                */
unsigned char prjclas;
                                 /* job class
                                                                */
unsigned char leftmarg;
                                 /* Indentation for the output */
                                                                */
                                 /* text
unsigned short linepp;
                                 /* Lines per page (including
                                                                */
                                 /* header and blanks)
                                                                */
                                 /* Type of header line
                                                                */
unsigned char headline;
unsigned char outform;
                                 /* The output format is
                                                                */
                                 /* character format only or
                                                                */
                                 /* character and hexadecimal
                                                                */
                                 /* format
                                                                */
                                 /* The document is to be
unsigned char twoside;
                                                                */
                                 /* printed on one or both
                                                                */
                                 /* paper sides (supported
                                                                */
                                 /* only on LP65 printers)
                                                                */
                                                                */
unsigned char rotation;
                                 /* The pages to be printed
                                 /* from the print-job are
                                                                */
                                 /* rotated
                                                                */
/* intray
                                                                */
union /* intray union */ {
                                 /* Paper input tray to be
                                                                */
        unsigned char intray;
                                 /* used (only on LP65 and RSO */
                                 /* printers)
                                                                */
        unsigned char in_tray;
                                                                */
        unsigned char intray_format;
                                 /*
                                                                */
} intray_union;
                                                                */
/* outtray
union /* outtray_union */ {
                                 /* Paper output tray to be
                                                                */
        unsigned char outtray;
                                 /* used (only on LP65 and RSO */
```

```
/* printers)
                                                                */
        unsigned char out_tray; /*
                                                                */
} outtray_union;
                                                                */
short pagecop;
                                 /* Copies of each printed
                                 /* page
                                                                */
struct PRNTDOC covpage mdl covpage:
                                 /* Cover-page
                                                                */
char form[6];
                                 /* Form-name to be used
                                                                */
char loop[3]:
                                 /* Loop-name to be used
                                                                */
char rotloop[3];
                                 /* Rotation-loop name to be
                                                                */
                                 /* used
                                                                */
char charset[16][3];
                                 /* One or more character-sets */
                                /* used for printing
                                                                */
                                /* Pool-name
char poolnam[4];
                                                                */
unsigned char poolind;
                                 /* Pool-index
                                                                */
char efo[2]:
                                 /* Film or EFO overlay for
                                                                */
                                 /* printing
                                                                */
struct PRNTDOC_overlay_mdl overlay;
                                 /* Double sided overlay (only */
                                 /* for LP65 printer type)
                                                                */
char fob[4]:
                                 /* FOB (Forms Overlay Buffer) */
                                 /* data overlay to be used
                                                                */
                                 /* for printing the document
                                                                */
                                 /* (only supported on HP
                                                                */
                                 /* printer equipped with a
                                                                */
                                 /* graphic buffer
                                                                */
                                 /* allow virtual controller
unsigned char virtual_s;
                                                                */
char filler2[2]:
                                 /* slack bytes
                                                                */
unsigned long pagepcl;
                                /* PCL file number (only for
                                                                */
                                 /* LP65 printer type)
                                                                */
                                 /* User resources file to be
                                                                */
char userres[44]:
                                 /* used for print resources
                                                                */
                                 /* (user PRFILE generated by
                                                                */
                                                                */
                                 /* PRM)
struct PRNTDOC transtab mdl transtab;
                                                                */
                                 /* Translation table to be
                                 /* used
                                                                */
char prname[8];
                                 /* Printer or printer pool
                                                                */
                                 /* name on which the document */
                                 /* is to be printed
                                                                */
unsigned char prtype;
                                /* Printer type requested
                                                                */
unsigned char srtmode;
                                /* Sorter mode
                                                                */
char filler3[2]:
                                 /* slack bytes
                                                                */
char tsn[4]:
                                /* output: TSN obtained from
                                                                */
                                                                */
                                /* the system
char cluster[8]:
                                                                */
                                /* for compatibility with
                                 /* MODIFY-PRINT-JOB
                                                                */
```

```
/* specified1
                                                               */
struct /* specified1 */ {
        char spec1_dtype: 1;
                                /* document type
                                                               */
        char spec1 dname: 1;
                                /* document name
                                                               */
                                                               */
        char spec1 dnum: 1;
                               /* document number
        char spec1 libelem: 1; /* library element
                                                               */
        char spec1 sectrec: 1; /* section record
                                                               */
        char spec1 firsrec: 1; /* first record
                                                               */
        char spec1_lastrec: 1; /* last record
                                                               */
        char spec1_recpart: 1;
                                /* record part
                                                               */
} specified1:
/* specified2
                                                               */
struct /* specified2 */ {
        char spec2_linesp: 1;
                                /* line spacing
                                                               */
        char spec2 docform: 1;
                                                               */
                                /* document format
        char spec2_contmod: 1; /* control mode
                                                               */
        char spec2_addcop: 1;
                                                               */
                                /* additional copies
        char spec2_family: 1;
                                                               */
                                /* familly
        char spec2_checkp: 1; /* checkpoint
                                                               */
        char spec2_monjv: 1;
                                                               */
                                /* monjv
        char spec2_jvpassw: 1;
                                /* .jv-password
                                                               */
} specified2;
/* specified3
                                                               */
struct /* specified3 */ {
        char spec3_prjname: 1;
                                                               */
                                /* print-job-name
                                                               */
        char spec3_prjprio: 1;
                                /* print-job-priority
                                /* processing admission
        char spec3 procadm: 1;
                                                               */
        char spec3_trunc: 1;
                                /* line-truncation
                                                               */
        char spec3_lockf: 1;
                                /* lock-file
                                                               */
        char spec3 delf: 1;
                                /* delete-after-print
                                                               */
        char spec3 outpart: 1; /* output part
                                                               */
                                                               */
        char spec3_pagecop: 1;
                                /* page-copies
} specified3;
/* specified4
                                                               */
struct /* specified4 */ {
        char spec4 leftmar: 1;
                                /* left-margin
                                                               */
        char spec4_linepp: 1;
                                /* line-per-page
                                                               */
        char spec4_headlin: 1; /* header-line
                                                               */
        char spec4_outform: 1;
                                                               */
                                /* output-format
        char spec4_twoside: 1; /* two-sided
                                                               */
        char spec4_rot: 1;
                                /* rotation
                                                               */
                                                               */
        char spec4 intray: 1;
                               /* input-trav
        char spec4_outtray: 1;
                                /* output-tray
                                                               */
} specified4;
```

```
/* specified5
                                                               */
struct /* specified5 */ {
        char spec5_covpage: 1; /* cover-page
                                                               */
        char spec5 form: 1;
                                /* form-name
                                                               */
                                                               */
        char spec5 loop: 1;
                               /* loop-name
        char spec5 rotloop: 1: /* rotation-loop-name
                                                               */
        char spec5 charset: 1; /* character-set
                                                               */
        char spec5 poolnam: 1; /* pool-name
                                                               */
        char spec5_poolind: 1; /* pool-index
                                                               */
        char spec5_efo: 1;
                                /* electronic-form-overlay
                                                               */
} specified5;
/* specified6
                                                               */
struct /* specified6 */ {
        char spec6_overlay: 1; /* overlay
                                                               */
        char spec6 fob: 1;
                                                               */
                                /* fob
        char spec6_pagepcl: 1; /* lp65-pagedef
                                                               */
                                                               */
        char spec6_userres: 1; /* user resources file
                                                               */
        char spec6 transtab: 1; /* translation table
        char spec6 prname: 1; /* printer-name
                                                               */
        char spec6_prtype: 1;
                                /* printer-type
                                                               */
        char spec6 tsn: 1;
                                /* tsn
                                                               */
} specified6;
                                                               */
/* specified7
struct /* specified7 */ {
                                                               */
        char spec7_rsopar: 1;
                                /* rso parameters
                                                               */
        char spec7_spspar: 1;
                                /* sps parameters
                                                               */
        char spec7 dppar: 1;
                                /* dprint parameters
        char spec7_dsempar: 1;
                                /* dsem parameters
                                                               */
        char spec7_free: 3;
                                /*
                                                               */
} specified7;
/* specified8
                                                               */
struct /* specified8 */ {
        char spec8 stproc: 1;
                                /* start-processing
                                                               */
        char spec8_srtmode: 1; /* sorter mode
                                                               */
        char spec8_virtual: 1; /* virtual control
                                                               */
        char spec8 jclass: 1;
                                /* job-class
                                                               */
        char spec8_prgname: 1;
                                /* prog. name
                                                               */
        char spec8_strprog: 1; /* string field
                                                               */
        char spec8_frmname: 1:
                                /* format name
                                                               */
        char spec8_free: 1;
                                /*
                                                               */
} specified8;
```

```
struct PRNTDOC virtual mdl program;
                                       /* virtual device synchrone
                                                                      */
        char format_name[63];
                                       /* format name
                                                                      */
        char mod res[2];
                                       /* reserved for modify
                                                                      */
        /* specified9
                                                                      */
                char spec9_topoff: 1; /* top-offset
                                                                      */
                char spec9_leftoff: 1; /* left-offset
                                                                      */
                char spec9_schedtime: 1;/*scheduling time
                                                                      */
                char spec9_free: 5;
                                                                      */
        } specified9;
        short top_offset;
                                        /* Specifies the length of
                                                                      */
                                        /* the white band that must
                                                                      */
                                        /* be inserted
                                                                      */
                                        /* Specifies the width of the */
        short left_offset;
                                        /* white band that must be
                                                                      */
                                        /* inserted
                                                                      */
                                       /* inserted
                                                                      */
                                       /* scheduling date
                                                                      */
        char schdate[10];
        char schtime[5];
                                      /* scheduling time
                                                                      */
                                       /* unused
                                                                      */
        char unused9[11]:
} :
/* call PRNTDOC via svc 108 */
#ifdef __SNI_HOST_BS2000
void SVC(int, void*);
#define PRNTDOC(p) _SVC(108, &p)
#endif
#endif
               /* PRNTDOC H */
```

2.2 The COBOL interface for the functions of the PRNTDOC macro

The COBOL interface implements the same functionality as the PRNTDOC assembly language macro. The individual functions can be specified analogously using COBOL language elements. The comments in the data structures listed below are therefore often merely references to the operand description of the PRNTDOC assembly language macro. The data structures and the function call are in the runtime module SCPADDR. This module is part of the SPOOL delivery scope as of version 3.0.

The interface is called in COBOL by means of a call statement and a parameter area. The fields of this interface described below can be copied by means of the copy statement to the COBOL source text within the working storage structure (see also the example on page 162ff).

The COBOL application program must be linked with the SCPADDR runtime module. The following functions are available to the user in this module:

- SCPADDR: modifies the "param-1" data area with the address passed in "param-2".
- SCPSETB: sets the bits of the "param-2" data area in the "param-1" data area with the length specified in "param-3".
- SCPSETB1: sets the bits of the "param-2" data area in the "param-1" data area with a length of 1 byte.
- SCPRSTB: resets the bits of the "param-2" data area in the "param-1" data area with the length specified in "param-3".
- SCPRSTB1: resets the bits of the "param-2" data area in the "param-1" data area with a length of 1 byte.
- PRNTDOC: initializes a print job with the parameters specified in the data areas.

For each field in which an address rather than a value is to be stored, the SCPADDR function must be called in order to enter the required address. This call is mandatory for the file name field and the field with the address of the parameter list for the subsystem RSO, DPRINT, SPS or DSEM.

A flag must be set for each required macro operand by calling the SCPSETB or SCPSETB1 function. Once each operand has been supplied the appropriate value and the flag has been set, the PRNTDOC function can be called (via a call statement) to initialize a print job.

Parameter list description

```
*********************
       PRNTDOCY
                 - Declarations
*********************
* yes-no-set
01 PRNTDOC-yes-no-set.
    02 PRNTDOC-yes-no-yes
                                       PIC X(01) VALUE SYS-X01.
                                       PIC X(01) VALUE SYS-X02.
    02 PRNTDOC-yes-no-no
* doc-part-type-set
01 PRNTDOC-doc-part-type-set.
<< Data structure comprising all the information required for a first
                                                                      >>
<< and last record when only a certain number of the input file's
                                                                      >>
<< records are to be processed. See also the description of the FIRSREC
                                                                      >>
<< operand on page 60 and the LASTREC operand on page 71.</pre>
                                                                      >>
    02 PRNTDOC-doc-part-begin-of-file PIC S9(09) COMP
                                       VALUE 1.
    02 PRNTDOC-doc-part-eof-char.
      03 FILLER
                                       PIC X(1)
                                       VALUE SYS-X7F.
      03 FILLER
                                       PIC X(3)
                                       VALUE HIGH-VALUES.
    02 PRNTDOC-doc-part-end-of-file
                                       REDEFINES
                                       PIC S9(09) COMP.
        PRNTDOC-doc-part-eof-char
    02 PRNTDOC-doc-part-by-string-id PIC S9(09) COMP
                                       VALUE ZERO.
* sect-ind
01 PRNTDOC-sect-ind-bits.
<< Data structure comprising information indicating whether the input
<< file is divided up by means of file marks and whether the marks are
                                                                      >>
<< in character format or hexadecimal format. See also the description
                                                                      >>
<< of the SECTREC operand on page 94.
                                                                      >>
                                       PIC X(01).
    02 PRNTDOC-sect-id-given
                                                 VALUE SYS-X80.
      88 PRNTDOC-sect-id-given-yes
      88 PRNTDOC-sect-id-given-no
                                                 VALUE SYS-X00.
    02 PRNTDOC-sect-id-input
                                      PIC X(01).
      88 PRNTDOC-sect-id-input-char
                                                 VALUE SYS-X40.
                                                VALUE SYS-X00.
      88 PRNTDOC-sect-id-input-hex
* prtype-set
                            PIC X(01).
01 PRNTDOC-prtype-set
<< Data structure comprising the possible printer types for the printout >>
<< via condition name definitions. See also the description of the
                                                                      >>
<< PRTYPE operand on page 89.
                                                                      >>
    88 PRNTDOC-1p65-printer
                                                 VALUE SYS-X04.
    88 PRNTDOC-hp-printer
                                                 VALUE SYS-X02.
    88 PRNTDOC-apa-printer
                                                 VALUE SYS-X01.
01 PRNTDOC-specified-set.
<< Data structure that lists the subset of possible options (operands
                                                                      >>
<< of the PRNTDOC assembly language macro) for each block (specified1
                                                                      >>
<< to specified7). If an operand from one of the blocks is required
                                                                      >>
```

```
<< for the spoolout job, access to this field must be enabled by setting >>
<< the bit indicator (assigned field name in the parameter list:
                                                                           >>
<< PRNTDOC-specified; see page 156). The field names essentially</pre>
                                                                          >>
<< correspond to the operand names of the PRNTDOC assembly language
                                                                           >>
<< macro.
                                                                           >>
     02 PRNTDOC-specified1-set
                                        PIC X(01).
       88 PRNTDOC-specified1-dtype
                                                      VALUE SYS-X80.
       88 PRNTDOC-specified1-dname
                                                     VALUE SYS-X40.
       88 PRNTDOC-specified1-dnum
                                                     VALUE SYS-X20.
       88 PRNTDOC-specified1-libelem
                                                     VALUE SYS-X10.
       88 PRNTDOC-specified1-sectrec
                                                     VALUE SYS-X08.
       88 PRNTDOC-specified1-firsrec
                                                     VALUE SYS-X04.
       88 PRNTDOC-specified1-lastrec
                                                     VALUE SYS-X02.
       88 PRNTDOC-specified1-recpart
                                                     VALUE SYS-X01.
     02 PRNTDOC-specified2-set
                                        PIC X(01).
       88 PRNTDOC-specified2-linesp
                                                     VALUE SYS-X80.
       88 PRNTDOC-specified2-docform
                                                     VALUE SYS-X40.
       88 PRNTDOC-specified2-contmod
                                                     VALUE SYS-X20.
       88 PRNTDOC-specified2-addcop
                                                     VALUE SYS-X10.
       88 PRNTDOC-specified2-family
                                                     VALUE SYS-X08.
       88 PRNTDOC-specified2-checkp
                                                     VALUE SYS-X04.
       88 PRNTDOC-specified2-monjv
                                                     VALUE SYS-X02.
       88 PRNTDOC-specified2-jvpassw
                                                     VALUE SYS-X01.
     02 PRNTDOC-specified3-set
                                        PIC X(01).
       88 PRNTDOC-specified3-prjname
                                                     VALUE SYS-X80.
       88 PRNTDOC-specified3-prjprio
                                                     VALUE SYS-X40.
       88 PRNTDOC-specified3-procadm
                                                     VALUE SYS-X20.
       88 PRNTDOC-specified3-trunc
                                                     VALUE SYS-X10.
       88 PRNTDOC-specified3-lockf
                                                     VALUE SYS-X08.
       88 PRNTDOC-specified3-delf
                                                     VALUE SYS-X04.
       88 PRNTDOC-specified3-outpart
                                                     VALUE SYS-X02.
       88 PRNTDOC-specified3-pagecop
                                                     VALUE SYS-X01.
     02 PRNTDOC-specified4-set
                                        PIC X(01).
       88 PRNTDOC-specified4-leftmar
                                                     VALUE SYS-X80.
       88 PRNTDOC-specified4-linepp
                                                     VALUE SYS-X40.
       88 PRNTDOC-specified4-headlin
                                                     VALUE SYS-X20.
       88 PRNTDOC-specified4-outform
                                                     VALUE SYS-X10.
       88 PRNTDOC-specified4-twoside
                                                     VALUE SYS-X08.
       88 PRNTDOC-specified4-rot
                                                     VALUE SYS-X04.
       88 PRNTDOC-specified4-intray
                                                     VALUE SYS-X02.
       88 PRNTDOC-specified4-outtray
                                                     VALUE SYS-X01.
     02 PRNTDOC-specified5-set
                                         PIC X(01).
       88 PRNTDOC-specified5-covpage
                                                     VALUE SYS-X80.
       88 PRNTDOC-specified5-form
                                                     VALUE SYS-X40.
       88 PRNTDOC-specified5-loop
                                                     VALUE SYS-X20.
       88 PRNTDOC-specified5-rotloop
                                                     VALUE SYS-X10.
       88 PRNTDOC-specified5-charset
                                                     VALUE SYS-X08.
       88 PRNTDOC-specified5-poolnam
                                                    VALUE SYS-X04.
       88 PRNTDOC-specified5-poolind
                                                     VALUE SYS-X02.
       88 PRNTDOC-specified5-efo
                                                     VALUE SYS-X01.
```

```
02 PRNTDOC-specified6-set
                                     PIC X(01).
       88 PRNTDOC-specified6-overlay
                                                   VALUE SYS-X80.
       88 PRNTDOC-specified6-fob
                                                  VALUE SYS-X40.
       88 PRNTDOC-specified6-pagepcl
                                                 VALUE SYS-X20.
                                                 VALUE SYS-X10.
       88 PRNTDOC-specified6-userres
       88 PRNTDOC-specified6-trantab
                                                 VALUE SYS-X08.
       88 PRNTDOC-specified6-prname
                                                 VALUE SYS-X04.
    88 PRNTDOC-specified6-prtype

02 PRNTDOC-specified7-set

PIC X(01).
       88 PRNTDOC-specified6-prtype
                                                  VALUE SYS-X02.
       88 PRNTDOC-specified7-rsopar
                                                  VALUE SYS-X80.
      88 PRNTDOC-specified7-spspar
88 PRNTDOC-specified7-dppar
88 PRNTDOC-specified7-dsempar
                                                  VALUE SYS-X40.
                                                 VALUE SYS-X20.
VALUE SYS-X10.
/---+----*
     Parameter list description
 01 PRNTDOC-pl.
<< Creation of the parameter list for the function call
                                                                        >>
    Standard header
    COPY fhdry.
          main return codes
<< Definition of the decimal return codes via condition names</pre>
                                                                        >>
          PRNTDOC-mret-code-SFT.
          88 PRNTDOC-mret-warning
          88 PRNTDOC-mret-rc-ok
                                                  VALUE 0.
                                                  VALUE 1.
          88 PRNTDOC-mret-recoverable-error VALUE
          88 PRNTDOC-mret-fatal-error
                                                  VALUE 3.
          88 PRNTDOC-mret-internal-error
                                                  VALUE 4.
    02 PRNTDOC-rso-pl PIC S9(09) COMP.
<< Data field for storing the address of a parameter list in which
                                                                        >>
<< RSO-specific values are stored
                                                                        >>
       address of rso-pl
    88 PRNTDOC-rso-pl-none VALUE

02 PRNTDOC-sps-pl PIC S9(09) COMP.
                                                   VALUE -1.
<< Data field for storing the address of a parameter list in which
<< SPS-specific values are stored
                                                                        >>
        address of sps-pl
                                                  VALUE -1.
      88 PRNTDOC-sps-pl-none
02 PRNTDOC-dprint-pl PIC S9(09) COMP.
<< Data field for storing the address of a parameter list in which
<< Dprint-specific values are stored</pre>
                                                                        >>
        address of dprint-pl
      88 PRNTDOC-dprint-pl-none VALUE
PRNTDOC-dsem-pl PIC S9(09) COMP.
                                                   VALUE -1.
    02 PRNTDOC-dsem-pl
<< Data field for storing the address of a parameter list in which
                                                                        >>
<< DSEM-specific values are stored</pre>
```

```
address of dsem-pl
      88 PRNTDOC-dsem-pl-none
                                                   VALUE -1.
    02 PRNTDOC-reserved-1
                                       PIC S9(09) COMP.
    02 PRNTDOC-reserved-2
                                       PIC S9(09) COMP.
    02 PRNTDOC-reserved-3
                                       PIC S9(09) COMP.
    02 PRNTDOC-reserved-4
                                        PIC S9(09) COMP.
    02 PRNTDOC-dtype
                                        PIC 9(04) COMP.
The possible types of files to be output are defined in this
<< structured data field. An integer is assigned to each condition name. >>
<< For information on the relationships between the values and the file
<< type, see also the description of the DTYPE operand on page 57.
                                                                        >>
    dtype-set
      88 PRNTDOC-dtype-file
                                                   VALUE 1.
      print a file
      88 PRNTDOC-dtype-lib-elem
                                                   VALUE 2.
      print a library-element
      88 PRNTDOC-dtype-posix-path
                                                  VALUE 3.
      print a POSIX-file
      88 PRNTDOC-dtype-eamn
                                                  VALUE 4.
      print a EAM-file
      88 PRNTDOC-dtype-system-file
                                                  VALUE 5.
      print a system-file
    02 PRNTDOC-dname.
<< The length of the file name can be stored in this data field.
                                                                        >>
      03 PRNTDOC-dname-len
                                       PIC 9(04) COMP.
          length of the document-name
        88 PRNTDOC-dname-len-std
                                                   VALUE 54.
                                 PIC S9(09) COMP.
      03 PRNTDOC-dname-ptr
        name of the file
        88 PRNTDOC-dname-ptr-none
                                                   VALUE -1.
    02 PRNTDOC-eamn OCCURS 16 TIMES PIC 9(05) BINARY.
<< Definition of a one-dimensional table with 16 elements for file
                                                                        >>
<< numbers when EAM files are printed
                                                                        >>
         eam-file-number (range 0..65535)
      88 PRNTDOC-eamn-std
                                                   VALUE ZERO.
    02 PRNTDOC-sysnn OCCURS 16 TIMES PIC 9(04) COMP.
<< Definition of a one-dimensional table with 16 elements for file
                                                                        >>
<< numbers when SYSLST files are printed
                                                                        >>
      88 PRNTDOC-sysnn-std
                                                   VALUE ZERO.
    02 PRNTDOC-libelem.
<< Data structure comprising the name, version and type of an element
                                                                        >>
<< to be printed from a PLAM library. See also the description of the
                                                                        >>
<< LIBELEM operand on page 73.</pre>
                                                                        >>
      03 PRNTDOC-element
                                       PIC X(80).
          element name
        88 PRNTDOC-element-dummy
                                                   VALUE .. ...
      03 PRNTDOC-version
                                        PIC X(40).
          element version
```

```
VALUE "".
         88 PRNTDOC-element-highest
       03 PRNTDOC-type
                                         PIC X(12).
           element type
     02 PRNTDOC-sectrec.
<< Data structure comprising both the string in the records that is to
                                                                          >>
<< function as a file mark and the position of the string. See also the
                                                                          >>
<< description of the SECTREC operand on page 94.</pre>
                                                                          >>
         section records
                                         PIC X(60).
       03 PRNTDOC-sect-id
       03 PRNTDOC-sect-pos
                                         PIC S9(04) COMP.
       03 PRNTDOC-sect-len
                                         PIC
                                             X(01).
       03 PRNTDOC-sect-ind
                                         PIC
                                              X(01).
     02 PRNTDOC-firsrec.
<< Data structure comprising all the information on the first record
                                                                          >>
<< when only a certain number of the input file's records are to be
                                                                          >>
<< processed. The settings for the last record can be made in the
                                                                          >>
<< "-lastrec" data structure. The data structure described here
                                                                          >>
<< comprises the following information: the number of the record as of
                                                                          >>
<< which the file is to be processed, the string whose occurrence is
                                                                          >>
<< to represent the first record, the position as of which the string
                                                                          >>
<< begins in the record, the number of the occurrence of the string, the >>
<< length of the string, and an indication as to whether the string is
<< specified in character format or hexadecimal format. See also the
                                                                          >>
<< description of the FIRSREC operand on page 60.</pre>
                                                                          >>
         beginning part
       03 PRNTDOC-stpos
                                         PIC S9(09) COMP.
                                         PIC X(60).
       03 PRNTDOC-ststrid
                                         PIC S9(04) COMP.
       03 PRNTDOC-ststrpos
       03 PRNTDOC-ststrocc
                                         PIC S9(04) COMP.
       03 PRNTDOC-ststrlen
                                         PIC X(01).
       03 PRNTDOC-filler6
                                         PIC
                                             X(01).
       03 PRNTDOC-strid-ind.
         04 PRNTDOC-st-char
                                         PIC X(01).
           88 PRNTDOC-st-char-char
                                                     VALUE SYS-X80.
           88 PRNTDOC-st-char-hexa
                                                     VALUE SYS-X00.
         04 PRNTDOC-ststrid-free
                                         PIC
                                             X(01).
     02 PRNTDOC-lastrec.
<< Data structure comprising all the information on the last record
                                                                          >>
<< when only a certain number of the input file's records are to be
                                                                          >>
<< processed. The settings for the first record can be made in the
                                                                          >>
<< "-firsrec" data structure. The data structure described here
                                                                          >>
<< comprises the following information: the number of the record up to
                                                                          >>
<< which the file is to be processed, the string whose occurrence is
                                                                          >>
<< to represent the last record, the position as of which the string
                                                                          >>
<< begins in the record, the number of the occurrence of the string, the >>
<< length of the string, and an indication as to whether the string is
                                                                          >>
<< specified in character format or hexadecimal format. See also the
                                                                          >>
```

```
<< description of the LASTREC operand on page 71.</pre>
                                                                          >>
         end part
       03 PRNTDOC-enpos
                                         PIC S9(09) COMP.
       03 PRNTDOC-enstrid
                                         PIC X(60).
                                         PIC S9(04) COMP.
       03 PRNTDOC-enstroos
       03 PRNTDOC-enstrocc
                                         PIC S9(04) COMP.
       03 PRNTDOC-enstrlen
                                         PIC X(01).
       03 PRNTDOC-filler7
                                         PIC X(01).
       03 PRNTDOC-strid-ind.
         04 PRNTDOC-en-char
                                         PIC
                                              X(01).
           88 PRNTDOC-en-char-char
                                                    VALUE SYS-X80.
           88 PRNTDOC-en-char-hexa
                                                    VALUE SYS-X00.
         04 PRNTDOC-enstrid-free
                                         PIC
                                              X(01).
     02 PRNTDOC-recpart.
<< Data structure comprising the first and last column of all selected
                                                                          >>
<< records when only a certain part of each record is to be processed.
                                                                          >>
<< See also the RECPART operand on page 90.</pre>
                                                                          >>
         record-part
       03 PRNTDOC-first-ch
                                         PIC S9(04) COMP.
       03 PRNTDOC-last-ch
                                         PIC S9(04) COMP.
         88 PRNTDOC-last-ch-std
                                                    VALUE ZERO.
     02 PRNTDOC-linesp.
<< Data structure comprising the position of the feed control character
<< and the number of line feeds or the control character interpretation
<< mode. Control character interpretation is controlled by integers that >>
<< each represent a control character interpretation mode. An integer is >>
<< assigned to each condition name in this structured data field. For
                                                                          >>
<< information on the relationships between the values and the file
                                                                          >>
<< type, see also the description of the LINESP operand on page 75.</pre>
                                                                          >>
         line feeds after
       03 PRNTDOC-cc-pos
                                         PIC S9(04) COMP.
       03 PRNTDOC-spacing
                                         PIC X(01).
         88 PRNTDOC-space-1
                                                    VALUE SYS-X01.
         88 PRNTDOC-space-2
                                                    VALUE SYS-X02.
         88 PRNTDOC-space-3
                                                    VALUE SYS-X04.
         88 PRNTDOC-space-by-ebc-control
                                                    VALUE SYS-X08.
         88 PRNTDOC-space-by-asa-control
                                                    VALUE SYS-X10.
         88 PRNTDOC-space-by-ibm-control
                                                    VALUE SYS-X20.
         88 PRNTDOC-space-no
                                                    VALUE SYS-X90.
         88 PRNTDOC-space-std
                                                    VALUE SYS-X00.
       03 PRNTDOC-filler4
                                         PIC X(01).
     02 PRNTDOC-docform
                                         PIC X(01).
<< Data structure comprising the type of the document contents. The type >>
<< of the document contents is controlled by integers that each
                                                                          >>
<< represent a specific type. An integer is assigned to each condition
                                                                          >>
<< name in this structured data field. For information on the
<< relationships between the values and the type of document contents,
                                                                          >>
<< see also the description of the DOCFORM operand on page 56.</pre>
                                                                          >>
```

```
docform-set
         88 PRNTDOC-docform-text
                                                   VALUE SYS-X00.
         88 PRNTDOC-docform-page-format
                                                   VALUE SYS-X01.
         88 PRNTDOC-docform-spec-format
                                                   VALUE SYS-X02.
     02 PRNTDOC-contmod.
<< Data structure comprising the control character interpretation mode.
<< Control character interpretation is controlled by integers that each
                                                                         >>
<< represent a control character interpretation mode. An integer is
                                                                         >>
<< assigned to each condition name in this structured data field. For
                                                                         >>
<< information on the relationships between the values and control
                                                                         >>
<< character interpretation, see also the description of the
                                                                         >>
<< CONTMOD=mode operand on page 50.</pre>
                                                                         >>
         control mode
       03 PRNTDOC-ctlmode
                                         PIC X(01).
       control-mode-set
         88 PRNTDOC-ctl-mode-std
                                                    VALUE SYS-X01.
         88 PRNTDOC-ctl-mode-physical
                                                    VALUE SYS-X02.
         88 PRNTDOC-ctl-mode-page-mode
                                                    VALUE SYS-X04.
         88 PRNTDOC-ctl-mode-logical
                                                    VALUE SYS-X08.
         88 PRNTDOC-ctl-mode-line-mode
                                                    VALUE SYS-X10.
         88 PRNTDOC-ctl-mode-apa
                                                    VALUE SYS-X40.
       03 PRNTDOC-pcc
                                         PIC X(01).
       pcc-set
         88 PRNTDOC-pcc-yes
                                                    VALUE SYS-X00.
         88 PRNTDOC-pcc-no
                                                    VALUE SYS-X01.
         88 PRNTDOC-pcc-dummy
                                                    VALUE SYS-X02.
                                       PIC X(01).
       03 PRNTDOC-ctltype
       control-type-set
         88 PRNTDOC-ctl-type-dummy
                                                    VALUE SYS-X00.
         88 PRNTDOC-ctl-type-hp
                                                    VALUE SYS-X01.
         88 PRNTDOC-ctl-type-compatible
                                                   VALUE SYS-X02.
02 PRNTDOC-family
                                  PIC X(01).
<< Data structure comprising the assignment of a common TSN. The
                                                                         >>
<< assignment of a common TSN is controlled by integers that each
                                                                         >>
<< represent a specific option. An integer is assigned to each condition >>
<< name in this structured data field. For information on the
                                                                         >>
<< relationships between the values and the family function, see also
                                                                         >>
<< the description of the FAMILY operand on page 50.
                                                                         >>
     family-set
       88 PRNTDOC-family-std
                                                    VALUE SYS-X01.
                                                    VALUE SYS-X02.
       88 PRNTDOC-family-yes
                                                    VALUE SYS-X03.
       88 PRNTDOC-family-no
     02 PRNTDOC-checkp
                                         PIC
                                             X(01).
     02 PRNTDOC-checkp
                                         PIC X(01).
<< Data structure comprising the checkpoint processing. Checkpoint
                                                                         >>
<< processing is controlled by integers that each represent a specific
                                                                         >>
```

```
<< option. An integer is assigned to each condition name in this
                                                                          >>
<< structured data field. For information on the relationships between</pre>
                                                                          >>
<< the values and restarting, see also the description of the CHECKP
                                                                          >>
<< operand on page 49.
                                                                          >>
     checkp-set
       88 PRNTDOC-checkp-on-pages
                                                     VALUE SYS-X00.
       88 PRNTDOC-checkp-on-sect-recs
                                                    VALUE SYS-X01.
     02 PRNTDOC-doc-number
                                         PIC 9(04) COMP.
     02 PRNTDOC-stproc-val
                                         PIC S9(09) COMP.
     02 PRNTDOC-stproc
                                         PIC X(01).
         stproc-val and stproc are only for internal usage
     02 PRNTDOC-lockf
                                        PIC X(01).
<< Data structure comprising the file protection during processing.</pre>
                                                                          >>
<< File protection is controlled by integers that each represent a
                                                                          >>
<< specific option. An integer is assigned to each condition name in
                                                                          >>
<< this structured data field. For information on the relationship
                                                                          >>
<< between the value and file protection, see also the description of
                                                                          >>
<< the LOCKF operand on page 77.
                                                                          >>
         lock while job is waiting
     lockf-set.
       88 PRNTDOC-lockf-std
                                                     VALUE SYS-X00.
       88 PRNTDOC-lockf-no
                                                     VALUE SYS-X01.
       88 PRNTDOC-lockf-yes
                                                    VALUE SYS-X02.
     02 PRNTDOC-delf
                                        PIC X(01).
<< Data structure comprising the deletion of the file. The deletion
                                                                          >>
<< of the file is controlled by integers that each represent a specific
                                                                          >>
<< option. An integer is assigned to each condition name in this
                                                                          >>
<< structured data field. For information on the relationships between
                                                                          >>
<< the values and file protection, see also the description of the LOCKF >>
<< operand on page 54.
                                                                          >>
         delete (destroy) document
     delf-set.
       88 PRNTDOC-delf-yes
                                                    VALUE SYS-X01.
                                                     VALUE SYS-X02.
       88 PRNTDOC-delf-destrov
       88 PRNTDOC-delf-no
                                                    VALUE SYS-X03.
     02 PRNTDOC-addcop
                                          PIC X(01).
<< Data field for a value for additional printouts of the file. See
                                                                          >>
<< also the description of the ADDCOP operand on page 48.
                                                                          >>
         additional copies
     02 PRNTDOC-jvpassi
                                        PIC S9(09) COMP.
<< Data field for a numeric password by which the job-monitoring job
                                                                          >>
<< variable is protected. See also the description of the JVPASSW
                                                                          >>
<< operand on page 70.
                                                                          >>
         password on monitor job-variable (numeric)
     02 PRNTDOC-jvpassw
                               REDEFINES PRNTDOC-ivpassi
                                         PIC X(04).
<< Data field for a numeric password by which the job-monitoring job
                                                                          >>
<< variable is protected. See also the description of the JVPASSW
                                                                          >>
```

```
<< operand on page 70.
                                                                          >>
         password on monitor job-variable
     02 PRNTDOC-moniv
                                          PIC X(54).
<< Data field for a monitoring job variable. See also the</pre>
                                                                          >>
<< description of the MONJV operand on page 78.</pre>
                                                                          >>
         monitor job-variable
     02 PRNTDOC-priname
                                         PIC X(08).
<< Data field for the job name. See also the description of the PRJNAME
<< operand on page 85.
                                                                          >>
         job-name of printjob
     02 PRNTDOC-priprio
                                        PIC X(01).
<< Data field for the job priority. See also the description of the
                                                                          >>
<< PRJPRIO operand on page 86.
                                                                          >>
        priority
     02 PRNTDOC-procadm.
<< Data structure in which the attributes of a user ID are stored when
                                                                          >>
<< the spoolout job is to be processed under a different user ID to the
                                                                          >>
<< caller's user ID. See also the description of the PROCADM operand on
                                                                          >>
<< page 87.
                                                                          >>
         execute print-job under another user-id
       03 PRNTDOC-user
                                         PIC X(08).
       03 PRNTDOC-account
                                         PIC X(08).
       03 PRNTDOC-password
                                         PIC X(08).
     02 PRNTDOC-truncation
                                        PIC X(01).
<< Data structure in which what happens when lines are truncated is
                                                                          >>
<< specified. An integer is assigned to each condition name in this
                                                                          >>
<< structured data field. For information on the relationships between
                                                                          >>
<< the values and what happens when lines are truncated, see also the
                                                                          >>
<< TRUNC operand on page 100.
                                                                          >>
     truncation-type-set
       88 PRNTDOC-truncation-std
                                                    VALUE SYS-X00.
       88 PRNTDOC-truncation-delete-file
                                                    VALUE SYS-X01.
       88 PRNTDOC-truncation-keep-file
                                                    VALUE SYS-X02.
     02 PRNTDOC-outpart.
<< Data structure in which the subset of all logical pages to which
                                                                          >>
<< output is to be restricted can be specified. The data structure
                                                                          >>
<< comprises a page or line number at which output is to begin and a
                                                                          >>
<< page or line number at which it is to end. A constant listed in
                                                                          >>
<< "layout-part-dim-set" can be used to specify the page or line number. >>
See also the description of the OUTPART operand on page 80.
         layout control (number of pages or lines)
                                         PIC S9(09) COMP.
       03 PRNTDOC-rec-from
       03 PRNTDOC-rec-to
                                         PIC S9(09) COMP.
       03 PRNTDOC-dim
                                         PIC X(01).
       lavout-part-dim-set
         88 PRNTDOC-layout-part-pages
                                                    VALUE SYS-X01.
         88 PRNTDOC-layout-part-lines
                                                    VALUE SYS-X02.
       03 PRNTDOC-filler5
                                         PIC X(03).
```

```
02 PRNTDOC-filler1
                                       PIC X(01).
     02 PRNTDOC-leftmarg
                                      PIC X(01).
<< Data field in which the indentation of the output text is specified.
See also the description of the LEFTMAR operand on page 73.
         indentation for the output text
       88 PRNTDOC-leftmarg-std
                                                    VALUE SYS-XFF.
                                 PIC S9(04) COMP.
     02 PRNTDOC-linepp
<< Data field in which the number of lines per page is specified. See
                                                                        >>
<< also the description of the LINEPP operand on page 75.</pre>
                                                                        >>
         lines per page (including header and blanks)
     02 PRNTDOC-headline
                                        PIC X(01).
<< Data structure in which the header line is specified. An integer is
                                                                        >>
<< assigned to each condition name in this structured data field. For
                                                                        >>
<< information on the relationships between the values and the header
                                                                        >>
<< line, see also the HEADLIN operand on page 63.
                                                                        >>
           date/first-record/page-number
       88 PRNTDOC-headline-no
                                                   VALUE SYS-X01.
       88 PRNTDOC-headline-std
                                                   VALUE SYS-X02.
       88 PRNTDOC-headline-record
                                                   VALUE SYS-X04.
       88 PRNTDOC-headline-date
                                                  VALUE SYS-X08.
       88 PRNTDOC-headline-date-record
                                                   VALUE SYS-XOC.
       88 PRNTDOC-headline-page
                                                  VALUE SYS-X10.
       88 PRNTDOC-headline-record-page
                                                  VALUE SYS-X14.
       88 PRNTDOC-headline-date-page
                                                  VALUE SYS-X18.
       88 PRNTDOC-headline-date-rec-page
                                                   VALUE SYS-X1C.
     02 PRNTDOC-outform
                                        PIC X(01).
<< Data structure in which the output format is specified. An integer
                                                                        >>
<< is assigned to each condition name in this structured data field.
                                                                        >>
<< For information on the relationship between the values and character
                                                                        >>
<< output, see also the description of the OUTFORM operand on page 79.
                                                                        >>
         output-format
     out.form-set
       88 PRNTDOC-outform-character
                                                   VALUE SYS-X01.
       88 PRNTDOC-outform-hexadecimal
                                                   VALUE SYS-X02.
     02 PRNTDOC-twoside
                                        PIC X(01).
<< Data structure in which two-sided printing is specified. An</pre>
                                                                        >>
<< integer is assigned to each condition name in this structured data
                                                                        >>
<< field. For information on the relationship between the values and
                                                                        >>
<< two-sided printing, see also the description of the TWOSIDE operand
                                                                        >>
<< on page 100.
                                                                        >>
        print on one or both sides of the paper
     twoside-set
       88 PRNTDOC-twoside-std
                                                    VALUE SYS-X00.
       88 PRNTDOC-twoside-ves
                                                    VALUE SYS-X01.
       88 PRNTDOC-twoside-tumble
                                                    VALUE SYS-X02.
       88 PRNTDOC-twoside-no
                                                   VALUE SYS-X04.
     02 PRNTDOC-rotation
                                      PIC X(01).
<< Data structure in which the page rotation of the printout is
                                                                        >>
```

```
<< specified. An integer is assigned to each condition name in this
                                                                          >>
<< structured data field. For information on the relationship between</pre>
                                                                          >>
<< the values and page rotation, see also the description of the ROT
                                                                          >>
<< operand on page 93.
                                                                          >>
         rotate the text when printing it
     rotation-set
       88 PRNTDOC-rotation-no
                                                    VALUE ..N".
       88 PRNTDOC-rotation-0
                                                    VALUE "O".
       88 PRNTDOC-rotation-90
                                                    VALUE ..1".
       88 PRNTDOC-rotation-180
                                                   VALUE ..2".
                                                   VALUE "3".
       88 PRNTDOC-rotation-270
       88 PRNTDOC-rotation-0-180
                                                    VALUE ..4".
                                                    VALUE "5".
       88 PRNTDOC-rotation-90-270
       88 PRNTDOC-rotation-180-0
                                                    VALUE "6".
       88 PRNTDOC-rotation-270-90
                                                    VALUE ..7".
       88 PRNTDOC-rot-by-control-codes
                                                    VALUE ..Y".
     02 PRNTDOC-intray
                                        PIC X(01).
<< Data field in which the paper input tray for LP65 printers is
                                                                          >>
<< specified. See also the description of the INTRAY operand on
                                                                          >>
<< page 65.
                                                                          >>
         input-tray (only LP65 and RS0-printers)
     02 PRNTDOC-outtray
                                        PIC X(01).
<< Data field in which the paper output tray for LP65 printers is
                                                                          >>
<< specified. See also the description of the OUTTRAY operand on
                                                                          >>
                                                                          >>
         output-tray (only LP65 and RS0-printers)
     02 PRNTDOC-pagecop
                                          PIC S9(04) COMP.
<< Data field in which the number of copies of each page is specified.
                                                                          >>
<< See also the description of the PAGECOP operand on page 83.</pre>
                                                                          >>
         copies of each printed page
       88 PRNTDOC-pagecop-std
                                                    VALUE -1.
     02 PRNTDOC-covpage.
<< Data structure comprising the information required for system exit
                                                                          >>
<< routines, which control the printing of header and trailer pages.
                                                                          >>
<< See also the description of the COVPAGE operand on page 53.</pre>
                                                                          >>
         cover-page
       03 PRNTDOC-headtext
                                         PIC X(32).
       03 PRNTDOC-headexit
                                         PIC S9(09) COMP.
         88 PRNTDOC-headexit-no
                                         VALUE -1.
       03 PRNTDOC-traiexit
                                         PIC S9(09) COMP.
         88 PRNTDOC-traiexit-no
                                         VALUE -1.
     02 PRNTDOC-form
                                          PIC X(06).
<< Data field in which the name of the form to be used is specified.
                                                                          >>
<< See also the description of the FORM operand on page 62.
                                                                          >>
     02 PRNTDOC-loop
                                         PIC X(03).
<< Data field in which the name of the loop to be used is specified.
                                                                          >>
See also the description of the LOOP operand on page 78.
                                                                          >>
     02 PRNTDOC-rotloop
                                        PIC X(03).
```

```
<< Data field in which the name of the rotation loop to be used is
                                                                         >>
<< specified. See also the description of the ROTLOOP operand on</pre>
                                                                         >>
<< page 93.
                                                                         >>
     02 PRNTDOC-charset OCCURS 16 TIMES PIC X(03).
<< Definition of a one-dimensional table with 16 elements for fonts used >>
<< for printing. See also the description of the CHARSET operand on
                                                                         >>
<< page 48.
                                                                         >>
         one or more character-sets used for printing
    02 PRNTDOC-poolnam
                                         PIC X(04).
<< Data field in which the name of the font pool to be used is
                                                                         >>
<< specified. See also the description of the POOLNAM operand on
                                                                         >>
<< page 85.
                                                                         >>
       88 PRNTDOC-poolnam-none
                                         VALUE SPACES.
     02 PRNTDOC-poolind
                                        PIC X(01).
<< Data field in which the number of a font from the font pool is
                                                                         >>
<< specified. See also the description of the POOLIND operand on
                                                                         >>
<< page 84.
                                                                         >>
     02 PRNTDOC-efo
                                         PIC X(02).
<< Data field in which the name of a film overlay is specified. See also >>
<< the description of the EFO operand on page 58.
                                                                         >>
         film formovl for printing
     02 PRNTDOC-overlay
                                        PIC X(02).
<< Data structure comprising the information required for EFO data
                                                                         >>
<< overlays for LP65 printers. See also the description of the OVERLAY
                                                                         >>
<< operand on page 82.
                                                                         >>
     02 PRNTDOC-overlay-par REDEFINES PRNTDOC-overlay.
       03 PRNTDOC-face
                                         PIC X(01).
         88 PRNTDOC-face-std
                                                    VALUE SYS-XFF.
         88 PRNTDOC-face-none
                                                    VALUE SYS-X00.
       03 PRNTDOC-reverse
                                         PIC X(01).
                                                    VALUE SYS-XFF.
         88 PRNTDOC-reverse-std
         88 PRNTDOC-reverse-none
                                                    VALUE SYS-X00.
     02 PRNTDOC-fob
                                         PIC X(04).
<< Data field in which the name of a FOB data overlay is specified.
                                                                         >>
<< See also the description of the FOB operand on page 61.
                                                                         >>
         Forms Overlay Buffer
     02 PRNTDOC-filler2
                                         PIC X(03).
         slack bytes
     02 PRNTDOC-pagepcl
                                         PIC S9(09) COMP.
<< Data field in which the number of the PCL file is specified. See
                                                                         >>
<< also the description of the PAGEPCL operand on page 84.
                                                                         >>
     02 PRNTDOC-userres
                                         PIC X(44).
<< Data field in which the name of the resource file is specified. See
                                                                         >>
<< also the description of the USERRES operand on page 101.
                                                                         >>
         user resources file
     02 PRNTDOC-transtab.
<< Data structure in which the information required for processing a
                                                                         >>
<< code translation table (name of the table, file) is specified. See
                                                                         >>
```

```
<< also the description of the TRANTAB operand on page 99.
                                                                 >>
        translation table
      03 PRNTDOC-name
                                     PIC X(08).
      03 PRNTDOC-file
                                     PIC X(44).
    02 PRNTDOC-prname
                                    PIC X(08).
<< Data field in which the name of the target printer is specified. See >>
<< also the description of the PRNAME operand on page 86.</pre>
                                                                  >>
       printer or printers pool
    02 PRNTDOC-prtype
                                    PIC X(01).
<< Data field in which the printer type is specified. The data structure >>
<< on page 144 comprises the printer types that can be specified. See >>
<< also the description of the PRNAME operand on page 86.
                                                                  >>
<< See also the description of the PRTYPE operand on page 89.</p>
                                                                  >>
        combination of printers
      88 PRNTDOC-prtype-any
                                               VALUE SYS-X00.
    02 PRNTDOC-filler3
                                    PIC X(03).
    02 PRNTDOC-tsn
                                     PIC X(04).
        Output-field: TSN obtained from the system
                                    PIC X(08).
    02 PRNTDOC-cluster
        Only for compatibility with MODJPRT
    02 PRNTDOC-specified.
<< Data structure in which access to a set of possible options (operands >>
<< of the assembly language macro) is permitted by setting the bit
>> indicator. The operands are contained in blocks (specified1 to
                                                                  >>
<< specified7) and listed in the "specified-set" data structure on
                                                                  >>
<< page 144.
                                                                  >>
        bits indicating which operands are given
      03 PRNTDOC-specified1
                                    PIC X(01).
      03 PRNTDOC-specified2
                                    PIC X(01).
      03 PRNTDOC-specified3
                                    PIC X(01).
                                    PIC X(01).
      03 PRNTDOC-specified4
      03 PRNTDOC-specified5
                                    PIC X(01).
      03 PRNTDOC-specified6
                                    PIC X(01).
      03 PRNTDOC-specified7
                                    PIC X(01).
      03 PRNTDOC-specified8
                                    PIC X(01).
Initialization of PRNTDOC-pl.
01 PRNTDOC-I-pl.
    02 PRNTDOC-I-hdr.
      03 PRNTDOC-I-ifid.
        04 PRNTDOC-I-unit
                                     PIC S9(04) COMP
                                     VALUE 46.
        04 PRNTDOC-I-function
                                     PIC X(01)
                                     VALUE SYS-X05.
          PRNTDOC-I-version
                                     PIC X(01)
                                     VALUE SYS-X01.
                                     PIC S9(09) COMP
      03 PRNTDOC-I-rc-nbr
```

	VALUE −1.			
02 PRNTDOC-I-rso-pl	PIC S9(09) COMP			
oz immode i iso pi	VALUE -1.			
02 PRNTDOC-I-sps-p1	PIC S9(09) COMP			
	VALUE −1.			
02 PRNTDOC-I-dpri-pl	PIC S9(09) COMP			
	VALUE −1.			
02 PRNTDOC-I-dsem-pl	PIC S9(09) COMP			
	VALUE −1.			
02 PRNTDOC-I-reserved-1	PIC S9(09) COMP			
00 DDNTD00 I	VALUE ZERO.			
02 PRNTDOC-I-reserved-2	PIC S9(09) COMP			
02 DDNTDOC I maganyad 2	VALUE ZERO.			
02 PRNTDOC-I-reserved-3	PIC S9(09) COMP VALUE ZERO.			
02 PRNTDOC-I-reserved-4	PIC S9(09) COMP			
02 FRNIDOC I TESETVEU 4	VALUE ZERO.			
02 PRNTDOC-I-dtype	PIC 9(04) COMP			
oz imirboo i dojpe	VALUE 1.			
02 PRNTDOC-I-dname.				
03 PRNTDOC-I-dname-1en	PIC 9(04) COMP			
	VALUE 54.			
03 PRNTDOC-I-dname-ptr	PIC S9(09) COMP			
	VALUE −1.			
02 PRNTDOC-I-eamn OCCURS 16 TIMES	PIC 9(05) BINARY			
	VALUE ZERO.			
02 PRNTDOC-I-sysnn OCCURS 16 TIMES				
	VALUE ZERO.			
02 PRNTDOC-I-libelem.	DIO ((00)			
03 PRNTDOC-I-element	PIC X(80)			
O2 DDNTDOC I	VALUE SPACES.			
03 PRNTDOC-I-version	PIC X(40)			
03 PRNTDOC-I-type	VALUE SPACES. PIC X(12)			
03 PRNTDOC-I-type	VALUE SPACES.			
02 PRNTDOC-I-sectrec.				
03 PRNTDOC-I-sect-id	PIC X(60)			
03 TIMIBOO I SCCC IQ	VALUE SPACES.			
03 PRNTDOC-I-sect-pos	PIC S9(04) COMP			
oo Timirada I dada pad	VALUE ZERO.			
03 PRNTDOC-I-sect-len	PIC X(01)			
	VALUE SYS-X00.			
O3 PRNTDOC-I-sect-ind	PIC X(01)			
VALUE SYS-X00.				
02 PRNTDOC-I-firsrec.				
03 PRNTDOC-I-stpos	PIC S9(09) COMP			
	VALUE 1.			
03 PRNTDOC-I-ststrid	PIC X(60)			

		VALUE SPACES.
03	PRNTDOC-I-ststrpos	PIC S9(04) COMP
		VALUE ZERO.
03	PRNTDOC-I-ststrocc	PIC S9(04) COMP
		VALUE 1.
03	PRNTDOC-I-ststrlen	PIC X(01)
		VALUE SYS-X00.
03	PRNTDOC-I-filler6	PIC X(01)
		VALUE SPACE.
03	PRNTDOC-I-strid-ind.	
C	04 PRNTDOC-I-st-char	PIC X(01)
		VALUE SYS-X00.
C	04 PRNTDOC-I-ststrid-free	PIC X(01)
		VALUE LOW-VALUE.
02 P	PRNTDOC-I-lastrec.	
	PRNTDOC-I-enpos.	
)4 FILLER	PIC X(01)
		VALUE SYS-X7F.
C)4 FILLER	PIC X(03)
		VALUE HIGH-VALUES.
03	PRNTDOC-I-enstrid	PIC X(60)
		VALUE SPACES.
03	PRNTDOC-I-enstrpos	PIC S9(04) COMP
		VALUE ZERO.
03	PRNTDOC-I-enstrocc	PIC S9(04) COMP
		VALUE 1.
03	PRNTDOC-I-enstrlen	PIC X(01)
		VALUE SYS-X00.
03	PRNTDOC-I-filler7	PIC X(01)
		VALUE SPACE.
03	PRNTDOC-I-strid-ind.	
C	04 PRNTDOC-I-en-char	PIC X(01)
		VALUE SYS-X00.
C	04 PRNTDOC-I-enstrid-free	PIC X(01)
		VALUE LOW-VALUE.
02 P	PRNTDOC-I-recpart.	
	PRNTDOC-I-first-ch	PIC S9(04) COMP
		VALUE 1.
03	PRNTDOC-I-last-ch	PIC S9(04) COMP
		VALUE ZERO.
02 P	PRNTDOC-I-linesp.	
03	PRNTDOC-I-cc-pos	PIC S9(04) COMP
	13 poo	VALUE O.
03	PRNTDOC-I-spacing	PIC X(01)
		VALUE SYS-X00.
03	PRNTDOC-I-filler4	PIC X(01)
		VALUE SPACE.
02 P	PRNTDOC-I-docform	PIC X(01)
'		,

	VALUE SYS-X00.
02 PRNTDOC-I-contmod. 03 PRNTDOC-I-ctlmode	PIC X(01)
03 PRNTDUC-1-Ctilliode	VALUE SYS-X00.
03 PRNTDOC-I-pcc	PIC X(01)
03 PRINTDUC-1-PCC	VALUE SYS-X02.
03 PRNTDOC-I-ctltype	PIC X(01)
03 FRNIDOC I CUITYPE	VALUE SYS-X00.
02 PRNTDOC-I-family	PIC X(01)
oz ikinboc i ramiriy	VALUE SYS-X01.
02 PRNTDOC-I-checkp	PIC X(01)
oz miniboo i eneekp	VALUE SYS-X00.
02 PRNTDOC-I-doc-number	PIC 9(04) COMP
of Timesoc I doe Tamber	VALUE ZERO.
02 PRNTDOC-I-stproc-val	PIC S9(09) COMP
·	VALUE ZERO.
02 PRNTDOC-I-stproc	PIC X(01)
·	VALUE SYS-X04.
02 PRNTDOC-I-lockf	PIC X(01)
	VALUE SYS-X00.
02 PRNTDOC-I-delf	PIC X(01)
	VALUE SYS-X03.
02 PRNTDOC-I-addcop	PIC X(01)
	VALUE LOW-VALUE.
02 PRNTDOC-I-jvpassi	PIC S9(09) COMP
	VALUE ZERO.
02 PRNTDOC-I-monjv	PIC X(54)
OO DRUTDOO I	VALUE SPACES.
02 PRNTDOC-I-prjname	PIC X(08)
OO DRNITROC I mainuis	VALUE SPACES.
02 PRNTDOC-I-prjprio	PIC X(01) VALUE LOW-VALUE.
02 DDNTDOC I procedm	VALUE LUW-VALUE.
02 PRNTDOC-I-procadm. 03 PRNTDOC-I-user	PIC X(08)
03 FRAIDOC I USEI	VALUE SPACES.
03 PRNTDOC-I-account	PIC X(08)
03 TRATEGO I decount	VALUE SPACES.
03 PRNTDOC-I-password	PIC X(08)
oo maada i pacemena	VALUE SPACES.
02 PRNTDOC-I-truncation	PIC X(01)
	VALUE SYS-X00.
02 PRNTDOC-I-outpart.	
03 PRNTDOC-I-rec-from	PIC S9(09) COMP
	VALUE ZERO.
03 PRNTDOC-I-rec-to	PIC S9(09) COMP
	VALUE ZERO.
03 PRNTDOC-I-dim	PIC X(01)
	VALUE SYS-X01.

03 PRNTDOC-I-filler5	PIC X(03)
02 PRNTDOC-I-filler1	VALUE SPACES. PIC X(01)
	VALUE SPACE.
02 PRNTDOC-I-leftmarg	PIC X(01) VALUE SYS-XFF.
02 PRNTDOC-I-linepp	PIC S9(04) COMP
02 PRNTDOC-I-headline	VALUE ZERO.
02 PRNIDOC-1-HeadTiffe	PIC X(01) VALUE SYS-X01.
02 PRNTDOC-I-outform	PIC X(01) VALUE SYS-X01.
02 PRNTDOC-I-twoside	PIC X(01)
02 PRNTDOC-I-rotation	VALUE SYS-X00. PIC X(01)
02 PRNTDOC-I-intray	VALUE "N". PIC X(01)
02 PRNTDOC-I-outtray	VALUE SYS-X00. PIC X(01)
02 PRNTDOC-I-pagecop	VALUE SYS-X00. PIC S9(04) COMP
OO DONTDOO I	VALUE −1.
O2 PRNTDOC-I-covpage. O3 PRNTDOC-I-headtext	PIC X(32)
	VALUE SPACES.
03 PRNTDOC-I-headexit	PIC S9(09) COMP VALUE -1.
03 PRNTDOC-I-traiexit	PIC S9(09) COMP VALUE -1.
02 PRNTDOC-I-form	PIC X(06)
02 PRNTDOC-I-loop	VALUE SPACES. PIC X(03)
02 TRATBOC 1 100p	VALUE SPACES.
02 PRNTDOC-I-rotloop	PIC X(03) VALUE SPACES.
02 PRNTDOC-I-charset-all	PIC X(48)
02 PRNTDOC-I-poolnam	VALUE SPACES. PIC X(04)
oz i kwiboc i pootiiaiii	VALUE SPACES.
02 PRNTDOC-I-poolind	PIC X(01) VALUE LOW-VALUE.
02 PRNTDOC-I-efo	PIC X(02) VALUE SPACES.
02 PRNTDOC-I-ovlmode-face	PIC X(01)
02 PRNTDOC-I-ovlmode-reverse	VALUE SYS-XFF. PIC X(01)
02 PRNTDOC-I-fob	VALUE SYS-XFF. PIC X(04)

02 PRNTDOC-I-filler2	VALUE SPACES. PIC X(03)
	VALUE SPACES.
02 PRNTDOC-I-pagepcl	PIC S9(09) COMP
F 2 - F	VALUE ZERO.
02 PRNTDOC-I-userres	PIC X(44)
02 1	VALUE SPACES.
02 PRNTDOC-I-transtab.	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
03 PRNTDOC-I-name	PIC X(08)
oo mane	VALUE SPACES.
03 PRNTDOC-I-file	PIC X(44)
03 THATBOOT TITE	VALUE SPACES.
02 PRNTDOC-I-prname	PIC X(08)
02 TRITIDOC I PITIAIIIE	VALUE SPACES.
02 PRNTDOC-I-ind3.	VALUE SPACES.
03 PRNTDOC-I-prtype	PIC X(01)
03 PKNTDOC-1-pr type	VALUE LOW-VALUE.
02 PRNTDOC-I-filler3	PIC X(03)
UZ PRNIDUC-1-IIITEMS	VALUE SPACES.
OO DONTDOC I +	
02 PRNTDOC-I-tsn	PIC X(04)
OO DRUTROO I I	VALUE SPACES.
02 PRNTDOC-I-cluster	PIC X(08)
	VALUE SPACES.
02 PRNTDOC-I-specified	PIC X(08)
	VALUE LOW-VALUES.
02 PRNTDOC-I-program.	
03 PRNTDOC-I-name	PIC X(08)
	VALUE SPACES.
03 PRNTDOC-I-string	PIC X(32)
	VALUE SPACES.
02 PRNTDOC-I-format-name	PIC X(63)
	VALUE SPACES.
02 PRNTDOC-I-modres	PIC S9(02) COMP
	VALUE ZERO.
02 PRNTDOC-I-specified9	PIC X(01)
	VALUE LOW-VALUE.
02 PRNTDOC-I-top-offset	PIC S9(04) COMP
	VALUE -1 .
02 PRNTDOC-I-left-offset	PIC S9(04 COMP
	VALUE -1 .
02 PRNTDOC-I-scheddate	PIC X(10)
	VALUE "0000-00-00"
02 PRNTDOC-I-schedtime	PIC X(05)
	VALUE "99:99".
02 PRNTDOC-I-unused	PIC X(11)
	VALUE LOW-VALUES.

Example

```
IDENTIFICATION DIVISION.
program-id.
*----*
     cobf5001.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
 SPECIAL-NAMES.
*----*
    TERMINAL IS v-terminal,
    SYMBOLIC CHARACTERS
    COPY esmhexay. .
DATA DIVISION.
WORKING-STORAGE SECTION.
01 file-names.
                       PIC X(54) VALUE "$TSOS.HELGA".
    02 file-name-1
01 hexa-chars
                          PIC X(16) VALUE "0123456789ABCDEF".
01 maincode-edit.
    02 maincode-dec
                            PIC S9(9) COMP.
    02 FILLER
                             REDEFINES maincode-dec.
                           PIC X(01) OCCURS 4 TIMES.
      03 maincode-byte
    02 maincode-hex
                            PIC X(08).
    02 FILLER
                              REDEFINES maincode-hex.
      03 maincode-char2 0CCURS 4 TIMES.
04 maincode-char PIC X(01) 0CCURS 2 TIMES.
   work-fields.
    02 work-counters.
       03 i
                              PIC S9(4) COMP.
       03 work-hw
                             PIC S9(4) COMP.
       03 FILLER
                             REDEFINES work-hw.
                          PIC X(01).
        04 work-hw-1
        04 \text{ work-hw-1} PIC X(01). 04 \text{ work-hw-2} PIC X(01).
    COPY prntdocy SUPPRESS.
```

```
PROCEDURE DIVISION.
s-main SECTION.
*----*
p-main.
*----*
   PERFORM s-test001.
p-exit.
*----*
    STOP RUN.
/---+----*
* TEST 001 : all default values
s-test001 SECTION.
p-test001-strt.
    DISPLAY "TEST1 (DEF) OF COBF5001" UPON v-terminal.
    MOVE prntdoc-i-pl TO prntdoc-pl.
DISPLAY "File-name" UPON v-terminal,
ACCEPT file-name-1 FROM v-terminal.
p-test001-init.
    CALL "SCPADDR" USING prntdoc-dname-ptr, file-name-1.
    COMPUTE prntdoc-dname-len = FUNCTION LENGTH(file-name-1),
    SET prntdoc-specified1-dname TO TRUE,
    CALL "SCPSETB1"
                              USING prntdoc-specified1,
                                      prntdoc-specified1-set.
p-test001-call.
    CALL "PRNTDOC"
                                USING prntdoc-pl.
p-test001-retc.
    IF esmfhdr-rc-nbr IN prntdoc-pl = ZERO
```

```
THEN
      DISPLAY "MAINCODE = X'00000000" UPON v-terminal.
      DISPLAY "TSN = ", prntdoc-tsn UPON v-terminal,
      MOVE esmfhdr-rc-nbr IN prntdoc-pl TO maincode-dec,
      PERFORM s-edit-maincode,
      DISPLAY "MAINCODE = X'", maincode-hex, "'" UPON v-terminal,
    END-IF.
p-test001-exit.
    EXIT.
s-edit-maincode SECTION.
p-edit-maincode-strt.
    PERFORM WITH TEST AFTER VARYING i FROM 1 BY 1
      UNTIL i > FUNCTION LENGTH(maincode-dec)
      MOVE ZERO
                                 TO work-hw,
      MOVE maincode-byte(i) TO work-hw-2,
      MOVE hexa-chars(work-hw / 16 + 1: 1)
                                 TO maincode-char(i, 1),
      MOVE hexa-chars(FUNCTION MOD(work-hw, 16) + 1: 1)
                                 TO maincode-char(i, 2),
    END-PERFORM.
p-edit-maincode-exit.
    FXIT.
*********
*** FNDF *****
```

2.3 Support macros for PRNTDOC

The following macros support use of the PRNTDOC macro and are employed for setting the subsystem-specific parameters for the PRNTDOC macro.

"PRNTRSO - define RSO parameters for PRNTDOC macro" on page 166 Sets parameters for the RSO subsystem.

"PRNTDPC - define Dprint parameters for PRNTDOC macro" on page 172 Sets parameters for the Dprint subsystem.

"PRNTSPS - define SPS-APA parameters for PRNTDOC macro" on page 177 Sets parameters for the SPS subsystem.

PRNTRSO - define RSO parameters for PRNTDOC macro

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

The PRNTRSO macro stores RSO-specific values for calling the PRNTDOC macro in a parameter list.

Format

Operation	Operands
PRNTRSO	,VARIANT= <u>001</u> / <c-string 33=""></c-string>
	,CHRA= <u>*STD</u> / *ALL / *RESTRICTED / <var: _char_att_type_set:1="" enum-of=""></var:>
	,RDIR= <u>*STD</u> / *YES / *NO / <var: _redir_type_set:1="" enum-of=""></var:>
	,TOPOFF= <u>*IGNORE</u> / <integer -255255=""></integer>
	,LEFTOFF= <u>*IGNORE</u> / <integer -255255=""></integer>
	URL=(urlptr,urllen,prtype) urlptr: <var: pointer=""> / *NONE urllen: <var: 065535="" int="">/ <integer 11023=""> / *NONE</integer></var:></var:>
	prtype: <var: _prtype_set="" enum-of:1=""> / *NONE /</var:>

Description of the operands

VARIANT=<u>001</u> / <c-string 3..3>

Specifies the variant of the parameter list.

CHRA=*STD / *ALL / *RESTRICTED / <var: enum-of _char_att_type_set:1>

Specifies whether all or only specific character set properties are to be taken into account for the RSO job. Properties concerned are, e.g. character type, NLQ, color, etc. (see the command or SPSERVE statement SHOW-SPOOL-CHARACTER-SETS). This parameter is ignored for header and trailer pages.

CHRA=*STD / *ALL

All properties of the character set to be used for the printout are to be taken into account for the current RSO job.

CHRA=*RESTRICTED

Only the CHARACTER TYPE, LANGUAGE and NLQ (NEAR-LETTER-QUALITY) properties are taken into account:

CHRA=<var: enum-of _char_att_type_set:1>

The character set properties to be used is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*STD	
1	*ALL	
2	*RESTRICTED	
See the parameter list description as of page 171.		

RDIR=*STD / *YES / *NO / <var: enum-of _redir_type_set:1>

Specifies whether a device administrator may redirect the RSO job to another printer. Job redirection by the user or the systems support is not affected by this.

RDIR=<u>*STD</u> / *YES

A device administrator may redirect the RSO job to another printer.

RDIR=*NO

A device administrator may not redirect the RSO job to another printer.

RDIR=<var: enum-of _redir_type_set:1>

Redirection of the RSO job is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*STD
1	*YES
2	*NO
See the parameter list description as of page 171.	

TOPOFF=*IGNORE / <integer -255..255>

Specifies the offset in millimeters of the first line to be printed from the top edge of the physical page.

This operand is only permitted for printers 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL, 4830-PCL, 4850-PCL, 9000-PCL, 9021, 9022-200, 9025, 9026-PCL and 9026-RENO.

TOPOFF=*IGNORE

Print control leaves no spacing between the first print line and the top edge of the page. The offset which is effective when printing the document is defined by either the printer default or the prolog file setting.

TOPOFF=<integer -255..255>

Positive values shift the first print line down, while negative values shift it up. The permitted value range is -255 to +255.

LEFTOFF=*IGNORE / <integer -255..255>

Specifies the offset in millimeters of the first print line from the left edge of the physical page. This operand is only permitted for printers 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL, 4830-PCL, 4850-PCL, 9000-PCL, 9021, 9022-200, 9025, 9026-PCL and 9026-RENO.

LEFTOFF=*IGNORE

Print control leaves no spacing between the first print line and the left edge of the page. The offset which is effective when printing the document is defined by either the printer default or the prolog file setting.

LEFTOFF=<integer -255..255>

Positive values shift the first print line to the right, while negative values shift it to the left. The permitted value range is -255 to +255.

URL=(urlptr,urllen,prtype)

Specifies the IPP printer addressed by its URL

urlptr: <var: pointer>

Specifies an area containing the printer URL.

urlptr: *NONE

Operand default value. No printer URL has been specified.

urllen: <var: int 0..65535>
Specifies the area length.
urllen: <integer 1..1023>

Specifies the area length.

urllen: *NONE

Operand default value. No URL has been specified.

prtype: <var: enum-of:1 _PRTYPE_SET>

The printer type is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationship exists between the values and the desired functions:

Value	Printer type
72	*4850_PCL
73	*4830_PCL
74	*4818_PCL
75	*9000_EPSQ
76	*9000_EPLQ
77	*9000_EPFX
78	*9000_PRO
79	*9000_PS
7B	*9046
7D	*9045_ANSI
7E	*9015
7F	*4825_PCL
80	*4822_PCL
84	*9001
88	*9025
8A	*9013
8B	*9000

Value	Printer type
8C	*9001_31
8D	*9011
8E	*9012
8F	*9022
A0	*9014
A1	*9021
A2	*9022_200
A4	*9000_PCL
A6	*4824_PCL
A7	*9026_RENO
A8	*9026_PCL
A9	*4812
AA	*9097
AB	*4011
AC	*4813
AD	*4821_PCL
AE	*2030_PCL

prtype: *4850_PCL | *4830_PCL | *4818_PCL | *9000_EPSQ | *9000_EPLQ |
*9000_EPFX | *9000_PRO | *9000_PS | *9046 | *9045_ANSI | *9015 | *4825_PCL |
*4822_PCL | *9001 | *9025 | *9013 | *9000 | *9001_31 | *9011 | *9012 | *9022 | *9014 |
*9021 | *9022_200 | *9000_PCL | *4824_PCL | *9026_RENO | *9026_PCL | *4812 |
*9097 | *4011 | *4813 | *4821_PCL | *2030_PCL
Specifies the RSO device type of the target printer.

prtype: *NONE

Operand default value. No URL has been specified.

Parameter list description

```
parameter list description
SROAMDL
         DS
                0F
SROAHDR FHDR MF=(C,SROA), EQUATES=NO
                                         Standard header
                                         Specifies whether all or only
SROACHRA DS
               FL1
                                         specific character set
                                         attributes are to be used
                                         for the print-job. These
                                         attributes include character
                                         type, near letter quality.
                                         color, etc. This operand
                                         does not apply to header and
                                         trailer pages.
    char att type set
SROACSTD EOU
                0
SROACALL EOU
                1
SROACRST
         EQU
                2
SROARDIR DS
                FL1
                                         Specifies whether a device
                                         administrator may redirect
                                         print-job to another
                                         printer. Redirection of
                                         print-job by users or
                                         system-administration is not
                                         affected by this operand.
    redir type set
SROARSTD EOU
                0
SROARYES EOU
                1
                2
         EQU
SROARNO
SROASPEC DS
                AL1
                                         specified; for compatibility
                                         with MODJRSO
SROACSPC EQU
              X'80'
SROARSPC EOU
              X'40'
SROATSPC EOU
              X'20'
SROALSPC
         EQU
                X'10'
                X'OF'
SROAUNUS EOU
SROAFIL1
         DS
                CL1
                                         slack byte
         DS
SROATOPO
                                         Specifies the length of the
                                         white band that must be
                                         inserted
SROALEFO DS
                Н
                                         Specifies the width of the
                                         white band that must be
                                         inserted
SROAUNS1
         DS
                XL8
                                         unused
SROA#
          EQU
              *-SROAHDR
```

PRNTDPC - define Dprint parameters for PRNTDOC macro

User group: Nonprivileged users

Programming language: Assembler, C

Macro type: S

This macro defines the Dprint parameters for the PRNTDOC macro.

Format

Operation	Operands
PRNTDPC	VARIANT=001 / <c-string 33=""></c-string>
	,FORMAT=*STD / <var: 63="" char:=""> / <c-string 163=""></c-string></var:>
	,CLUSTER= <u>*LOCAL-CLUSTER</u> / <var: 8="" char:=""> / <c-string 18=""></c-string></var:>
	.RESLOC= <u>*STD</u> / *HOME / *SERVER / <var: enum-of_resloc_set:1=""></var:>
	,OUTFORM=*STD / <c-string 163=""></c-string>

Description of the operands

VARIANT=<u>001</u> / <c-string 3..3>

Specifies the variant of the parameter list.

FORMAT=*STD / <var: char: 63> / <c-string 1..63>

type of document contents.

*STD

By default, no special format name is assumed.

<var: char: 63>

Name of the format name field.

<c-string 1..63>

Format name; specified directly.

CLUSTER=*LOCAL-CLUSTER / <var: char: 8> / <c-string 1..8>

specification of the Dprint cluster.

*LOCAL-CLUSTER

By default, the local cluster is assumed.

<var: char: 8>

Name of the cluster name field.

<c-string 1..8>

Cluster name; specified directly.

RESLOC=

Location of the resources to be used for printing.

*STD

By default, the values from the SPOOL parameter file are assumed.

*HOME

The client resources are used.

*SERVER

The server resources are used.

<var: enum-of resloc set:1>

Name of the equate for the operand value. Only possible in conjunction with MF=M.

OUTFORM=

Only relevant for interoperability with SPOOL in UNIX systems. Specifies the print language to be used for the print job.

*NONE

No output format is specified. The document is transferred to UNIX systems without change. It is assumed that the printer in the UNIX system knows the format name which was specified in the DOCFORM=*SPECIAL_FORMAT(...) operand structure.

<c-string: 1..63 with-low>

Format name known and supported by the printer in the UNIX system specified in the PRNAME operand of the PRNTDOC and MODPJAT macros. See description of the operand on page 86 and section "MODPJAT - modify print job attributes" on page 210.

Parameter list description

SDCA#

EQU

*-SDCAHDR

```
parameter list description
SDCAPL
        DS
               0F
SDCAHDR FHDR MF=(C,SDCA), EQUATES=NO
                                        Standard header
SDCACLUS DS
                                        cluster name to which the
               CL8
                                        print-job is submitted
SDCAFMT
         DS
                                        document content type
               CL63
SDCARESL DS
               FL1
                                        location of the resources
                                        used to process the
                                        print-job is submitted
   _resloc_set
SDCARSTD EQU
               0
               1
SDCARHOM EQU
SDCARSRV
        EQU
               2
SDCASP1
         DS
               AL1
                                        specified
SDCASP11 EQU
              X'80'
SDCASP12 EQU
             X'40'
SDCASP13 EQU
              X'20'
SDCASP14 EOU
              X'10'
SDCASP1F
        EQU
              X'OF'
SDCAOUTF DS
              CL63
                                        printer specific language
SDCAFIL1 DS
               CL4
                                        slack bytes
```

};

C header file

```
/* resloc set
                                                                        */
/* ENUM _resloc_set
                                                                        */
#define PRNTDPCresloc_std 0
                                         /*
                                                                        */
#define PRNTDPCresloc home 1
                                         /*
                                                                        */
#define PRNTDPCresloc_serv 2
                                         /*
                                                                        */
/* parameter list description
                                                                        */
struct PRNTDPC_pl_md1 {
        /* Standard header
                                                                        */
        struct ESMFHDR hdr:
        char cluster[8]:
                                         /* cluster name to which the
                                                                        */
                                                                        */
                                         /* print-job is submitted
        char format[63];
                                         /* document content type
                                                                        */
                                         /* location of the resources
        unsigned char resloc:
                                                                        */
                                         /* used to process the
                                                                        */
                                         /* print-job is submitted
                                                                        */
        /* specified
                                                                        */
        struct /* specified */ {
                char spec_format: 1;
                                         /*
                                                                        */
                char spec cluster: 1;
                                         /*
                                                                        */
                char spec_resloc: 1;
                                         /*
                                                                        */
                char spec_outfrm: 1;
                                         /*
                                                                        */
                                         /*
                char spec free: 4;
                                                                        */
        } specified;
        char out_format[63];
                                         /* printer specific language
                                                                        */
        char filler1[4]:
                                         /* slack bytes
                                                                        */
```

Structure layout

Offset	Identification	Data type:value	Meaning/comment
FW	PRNTDPC_pl_mdl	STRUCT:84	Parameter list description
000	hdr	ESMFHDR:8	Standard header
800	cluster	CHAR:8	Cluster name to which the print job is passed
010	format	CHAR:63	Document contents type
04F	resloc	ENUM:1	Location of the resources used for the transferred print job
	std	0	Assembler list: ASS:SDCARESLOC_STD
	home	1	Assembler list: ASS:SDCARESLOC_HOME
	server	2	Assembler list: ASS:SDCARESLOC_SERV
050	specified	FLAG:1	Specified
00	format	BIT:1	Assembler list: ASS:SDCASPEC_FORMAT
01	cluster	BIT:1	Assembler list: ASS:SDCASPEC_CLUSTER
	C-F:spec_cluster		
02	resloc	BIT:1	Assembler list: ASS:SDCASPEC_RESLOC
00	C-F:spec_cluster	DIT	
03	spec_free	BIT:5	
051	filler1	CHAR:3	Slack bytes

PRNTSPS - define SPS-APA parameters for PRNTDOC macro

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

This macro sets the operands specific for SPS-APA.

Format

Operation	Operands
PRNTSPS	VARIANT= <u>001</u> / <c-string33></c-string33>
	FONTS=*STD / array(4): <var: 8:="" alphanum_name_18="" char:=""></var:>
	PAGEDEF=*STD / <var: 8:="" alphanum_name_18="" char:=""></var:>
	FORMDEF= <u>*STD</u> / <var: 8:="" alphanum_name_18="" char:=""></var:>
	TRC=*STD / <var: _trc_set:1="" enum-of=""> / *YES / *NO</var:>
	MSGPAGE= <u>*STD</u> / <var: _msgpage_set:1="" enum-of=""> / *YES / *NO</var:>
	FRONTOVL= <u>*STD</u> / <var: 8:alphanum_name_18="" char:=""></var:>
	BACKOVL=*STD / <var: 8:alphanum_name_18="" char:=""></var:>

Description of the operands

VARIANT=<c-string 3..3>

Specifies the variant of the parameter list.

FONTS=*STD / array(4): <var: char: 8: alphanum_name_1..8>

Only for local SPOOL.

Specifies the fonts to be evaluated by SPS if the TABLE-REFERENCE-CHAR operand is used.

FONTS=*STD

The standard font for this printer type is selected from the desired form. It can be queried with SHOW-SPOOL-FORMS.

FONTS=array(4): <var: char: 8: alphanum_name_1..8>

Name of the fonts to be linked to the TRC values in the print file. Each TRC value represents a specific font.

Regardless of the number of different TRC values in the file, a maximum of four fonts, which are to be specified in the form of a list, can be used for printing the data records. TRC values higher than X'03' (corresponds to the forth list element) automatically refer back to the first font in the list.

PAGEDEF=*STD / <var: char: 8: alphanum_name_1..8>

Only for local SPOOL.

Specifies the page definition to be used for printing on LP65 or APA printers.

PAGEDEF=*STD

For APA printers:

The page definition assigned to the form specified in FORM-NAME is used.

For LP65 printers:

Number of the PCL file with which the print file is to be output. Only the print file itself is printed with this PCL file. The header and trailer pages are controlled via the PCL file defined in the SPOOL parameter file form.

Notes

- A PCL file which prevents selecting a different PCL file via a channel command must not be used.
- The printer operating mode is checked after the specified PCL file is started. If EXCCW mode is active and an error occurs during printing, the resumption point is the last SECTION record. If no entry was made for the SECTION operand, printing restarts from the beginning of the file.

PAGEDEF=<var: char: 8: alphanum_name_1..8>

Only for APA printers:

The page definition with the specified name is to be used. This must be in the SPSLIB.

Note

The first two characters of the specified name must be "P1", otherwise the macro is rejected.

FORMDEF=*STD / <var: char: 8: alphanum_name_1..8>

Only for local spool;

specifies the format definition to be used for printing on APA printers.

FORMDEF=*STD

The format definition assigned to the form specified in FORM-NAME is used.

FORMDEF=<var: char: 8: alphanum_name_1..8>

The format definition with the specified name is to be used. This must be in the SPSLIB.

Note

The first two characters of the specified name must be "F1", otherwise the macro is rejected.

TRC=*STD / <var: enum-of _trc_set:1> / *YES / *NO

Specifies whether the user selects fonts for print page layout via control characters in the text. This selection can be made with the TRC (Table Reference Character) control characters, which refer to a list of fonts within the data records.

The list can either be part of the PAGE-DEFINITION or can be made known using the CHARACTER-SETS operand. A detailed description on using the TRC can be found in the "SPOOL (BS2000/OSD)" manual.

TRC=*STD

TRC=*YES

The print file contains font selection control characters (TRC) which have to be evaluated by SPS. Each TRC value represents a specific font with which the data record concerned is to be printed. The separate values (from X'00' to X'0F') are to be linked either according to their order with the entries in the FONTLIST (font list of the page definition PAGEDEF) or explicitly with the fonts specified in the CHARACTER-SETS operand. Regardless of the number of different TRC values in the file, a maximum of four fonts, which are to be specified in the form of a list, can be used for printing the data records.

TRC values higher than X'03' (corresponds to the forth list element) automatically refer back to the first font in the list.

TRC=*NO

There are either no font selection control character TRCs in the print file or, if there are, they are to be ignored.

MSGPAGE=*STD / <var: enum-of _msgpage_set:1> / *YES / *NO

Specifies whether the APA message page is to be printed or not. The APA message page contains error messages and warnings. The entry is ignored if an APA printer is not in use.

FRONTOVL=*STD / <var: char: 8:alphanum_name_1..8>

Specifies whether an APA overlay is to be used for the front page.

FRONTOVL=*STD

No APA overlays are used for the output, except when an APA overlay is specified in the corresponding form definition.

FRONTOVL=<var: char: 8:alphanum_name_1..8>

Name of the APA overlay that is to be used for the front page.

BACKOVL=*STD / <var: char: 8:alphanum_name_1..8>

Specifies whether an APA overlay is to be used for the back page.

BACKOVL=*STD

No APA overlays are used for the output, except when an APA overlay is specified in the corresponding form definition.

BACKOVL=<var: char: 8:alphanum_name_1..8>

Name of the APA overlay that is to be used for the back page.

Parameter list description

```
parameter list description
        DS
               0F
SPSAPL
SPSAHDR FHDR MF=(C,SPSA), EQUATES=NO
                                       Standard header
                                       When TRCs are used, allows to
SPSAFNT DS
              4CL8
                                       reference a list of fonts
SPSAFNT# EQU
                                       Element name of the page
SPSAPAGD DS
             CL8
                                       definition to be used.
                                       Element name of the format
             CL8
SPSAFRMD DS
                                       definition to be used.
SPSATRC DS
              FL1
                                       Specifies whether the file to
                                       print contains TRCs.
   trc set
SPSATRCY EQU
               1
SPSATRON FOU
SPSATRCS EQU
SPSAMSGP DS
               FI1
   _msgpage_set
SPSAMSGY EQU
             1
SPSAMSGN EQU
               2
SPSAMSGS EOU
SPSASP1
         DS
              AL1
                                       specified
SPSASP11 EOU
             X'80'
SPSASP12 EQU
             X'40'
SPSASP13 EQU
             X'20'
SPSASP14 EOU
             X'10'
SPSASP15 EQU
             X'08'
SPSASP1F EQU
             X'07'
SPSAFIL1 DS
               CL1
                                       slack byte
SPSA#
         EOU *-SPSAHDR
```

SNPPRNT - define notification parameters for PRNTDOC

User group: Nonprivileged users

Programming languages: Assembler, C, CPP

Macro type: S

The macro SNPPRNT describes the parameter list for the support of the Notification Service.

Format

Operation	Operands
SNPPRNT	MF = D / I / M
	,PREFIX = SNPPRNT / <name></name>
	,MACID = <name 13=""></name>
	,PARAM = <name 127=""></name>
	,NOTIFICATION = *STD / *NO / *YES
	,MTH_NAME = *MAIL / <c-string 18=""> / <var: 18="" char=""></var:></c-string>
	,OBJ_ATTR_TYP = <u>*NONE</u> / *ALL / *LIST
	,OBJ_ATTR = array(20): elem: <c-string 164=""> / <var:char 164=""></var:char></c-string>
	,EVT_NAME_TYP = <u>*ALL</u> / *LIST
	,EVT_NAME = array(20): elem: <c-string 124=""> / <var:char 124=""></var:char></c-string>
	,USER_DATA = *NONE / <c-string 163=""> / <var: 163="" char=""></var:></c-string>
	,RECIPIENT_ADDR = <c-string 1224=""> / <var: 1224="" char=""></var:></c-string>

Description of the operands

MF=D/I/M

Type of the macro call. You will find more information in the "Executive Macros" manual. Possible values for the different programming languages:

Assembler:	C, D, L, M
C:	
CPP:	D, I, M

Note

You can only specify pointer variables ("var:pointer" operand value) with MF=M. In addition, a list must be generated with MF=D.

PREFIX=SNPPRNT / <name>

Specifies the first character of field names and equates.

Default values for the different programming languages:

Assembler:	S
C:	
CPP:	SNPPRNT

MACID=<name 1..3>

Specifies the second to fourth characters (inclusive) of the field names and equates. Default values for the different programming languages:

Assembler:	PRT
C:	
CPP:	

PARAM=<name 1..27>

Specifies the address of the operand list (permitted only in the case of MF formats 2 and 3). You will find more information in the "Executive Macros" manual.

NOTIFICATION=*STD / *NO / *YES

Selects the notification processing for the associated print jobs.

NOTIFICATION=*STD

Notification delivery will be processed for the current print job if the owner of this print job has previously recorded permanent subscriptions in the notification resources file.

NOTIFICATION=*NO

Notification processing is turned off for this print job. Notification will not be generated even if there are valid subscriptions belonging to the owner of this print job recorded in the notification resources file. However, the notification generated for subscriptions belonging to other privileged users is not turned off.

NOTIFICATION=*YES

This operand creates one temporary subscription resource. This subscription exists as long as the current print job exists. This operand allows a user to associate a subscription resource to a particular print job.

MTH_NAME=*MAIL / <c-string 1..8> / <var: char 1..8>

Specifies the notification delivery method.

OBJ ATTR TYP=*NONE/*ALL/*LIST

Specifies the object attributes associated with the notifications.

Objects, e.g. print jobs, for which notifications are sent may have attributes associated with them. The user may want to have one or more of these associated attributes returned with a particular notification. Generally these may include any attribute associated with the object emitting the notification.

OBJ ATTR TYP=*NONE

No attribute is selected.

OBJ ATTR TYP=*ALL

All the attributes associated with the print job are selected.

OBJ ATTR TYP=*LIST

Some of the attributes are selected. The operand OBJ_ATTR must be specified.

OBJ_ATTR=array(20): elem: <c-string 1..64> / <var:char 1..64>

Specifies the explicit list of attributes. Only relevant with OBJ_ATTR_TYP=*LIST.

EVT NAME TYP=*ALL/*LIST

Specifies the type of events associated with the print job that are selected.

EVT_NAME_TYP=*ALL

All the events associated with the print job are selected.

EVT NAME TYP=*LIST

Some events associated with the print job are selected. The operand EVT_NAME must be specified.

EVT_NAME=array(20): elem: <c-string 1..24> / <var:char 1..24>

Specifies the explicit list of subscribed events. Only relevant with EVT_NAME_TYP = *LIST.

USER_DATA=*NONE / <c-string 1..63> / <var: char 1..63>

Contains opaque data that some delivery methods include in the notification messages.

USER_DATA=*NONE

No user data is specified.

USER_DATA=<c-string 1..63>

Specifies the user data value.

USER DATA=<var: char 1..63>

Specifies the user data value.

RECIPIENT_ADDR=<c-string 1..224> / <var: char 1..224>

Specifies the delivery address for the notifications according to the selected method.

Structure layouts

SNPPRNT (Assembler)

For Assembler, the macro SNPPRNT has the following structure layout:

```
*********************
  BEGIN-INTERFACE
                    SNPPRNT
  TITLE
                   (/ Notification PRINT-DOC description /)
*
  NAME
                   SNPPRNT
  DOMAIN
                    SP00L
* LANGUAGE
                    ASS
* COPYRIGHT
               (C) Fujitsu Siemens Computers GmbH 2003
                        ALL RIGHTS RESERVED
 COMPILATION-SCOPE RESTRICTED
  INTERFACE-TYPE CALL
  RUN-CONTEXT
                    TU
  PURPOSE
                    (/ Layout of notification part of PRINT-DOC api /)
*
  SYNTAX
                    (/ Syntax Variant 1:
                            SNPPRNT MF = C|D|M|L
*
                          , PREFIX
                                    = [S] | <name>
                          . MACID
                                      = [PRT] | <name>
                          , EQUATES
                                      = [YES] \mid NO
                          , NOTIFICATION= *STD |
                                         *N0 |
                                         *YES |
                                         default *STD
                          , MTH NAME
                                      = *MAII |
                                        <c-string 1..8> |
                                        <var: char 1..8>
                          , OBJ_ATTR_TYP= *NONE |
                                         *ALL |
                                         *LIST |
                                         default *NON
                          . OBJ ATTR
                                      = array(20)
                                        elem: <c-string 1..64> |
                                              <var: char 1..64>
                          , EVT_NAME_TYP= *ALL |
                                         *LIST |
                                         default *ALL
                          , EVT_NAME
                                      = array(20)
                                        elem: < c-string 1...24 > 1
                                              <var: char 1..24>
                          , USER_DATA = *NONE
                                        <c-string 1..63> |
```

SNPPRNT.H (C)

See Assembler description

SNPPRNTC (CPP)

See Assembler description

SSVBARC - translate data for barcodes

User group: privileged user

Programming languages: Assembler

Macro type: M

This macro provides a standard way to call the barcode program (LLM) that translates a character into its corresponding barcode representation.

Format

Operation	Operands
SSVBARC	MF = C/D/L/M
	,PREFIX= <u>S</u> / <name 11=""></name>
	,MACID=BRC / <name 13=""></name>
	,EQUATES= <u>YES</u> / NO
	,INDTAADR= <var: pointer=""></var:>
	,INDTALEN= <var: 02147483647="" int=""></var:>
	,OUTDTAADR= <var: pointer=""></var:>
	,OUTDTALEN= <var: 02147483647="" int=""></var:>

Description of the operands

MF=C/D/L/M

Type of the macro call. You will find more information in the "Executive Macros" manual.

PREFIX=S / <name 1..1>

Specifies the first character of field names and equates.

MACID=BRC / <name 1..3>

Specifies the second to fourth characters (inclusive) of field names and equates.

EQUATES=YES / NO

Specifies whether the equates are generated or not.

INDTAADR=<var: pointer>

Specifies the address of the characters string to be translated.

INDTALEN=<var: int 0..2147483647>

Specifies the number of characters to be translated.

OUTDTAADR=<var: pointer>

Specifies the address of the area that will contain the barcode (translated characters string).

OUTDTALEN=<var: int 0..2147483647>

Specifies the length of the barcode area.

Return codes

(SC2)	SC1	Maincode	Meaning
00	00	0000	Successful processing
00	01	0001	Request memory error
00	02	0001	Buffer too short

Structure layout

```
SSVBARC (ASS)
********************
  BEGIN-INTERFACE SSVBARC
  TITLE
                  (/ TRANSLATE DATA FOR BARCODE PARAM LIST /)
* NAME
                   SSVBARC
* DOMAIN
                   SPOOL
 LANGUAGE
                   ASS
* COPYRIGHT
                   (C) Fujitsu Siemens Computers GmbH 2004
                      ALL RIGHTS RESERVED
* COMPILATION-SCOPE RESTRICTED
 INTERFACE-TYPE
                  CALL
* RUN-CONTEXT
                  TPR
 PURPOSE
                  (/ TRANSLATE DATA FOR BARCODE PARAM LIST /)
  SYNTAX
                  (/ Syntax Variant 1:
                          SSVBARC MF = C|D|L|M
                         , PREFIX = [S] \mid < name >
                                   = [BRC] | <name>
                         , MACID
                         , EQUATES = [YES] \mid NO
                         , INDTAADR = <var: pointer>
                         . INDTALEN = < var: int 0..2147483647 > 
                         , OUTDTAADR = <var: pointer>
                         , OUTDTALEN = < var: int 0..2147483647 > / 
                   SSVBARC.
  END-INTERFACE
```

3 Macros for outputting information

This section describes the SPSINF macro with which you can obtain information about the SPOOL parameter file.

SPSINF - request information

User group: Systems support, nonprivileged users

Programming language: Assembler

Macro type: O

The SPSINF macro provides information on the contents of the SPOOL parameter file. The following information is output:

- number of entries for a type (device, form and character set entries); see also the structure of the SPOOL parameter file (see the "SPSERVE (BS2000/OSD)" manual)
- entry contents for a specified type
- SPOOL global default settings
- list of device names, character set names, forms or device pools
- contents of all entries for all types.

The requested information is identified via a short name and transferred to an output area. The list on page 194 contains the short name to be specified, the minimum output area length required and the name of the macro which generates a description of the output area concerned.

The output area size generally depends on whether just information on an entry in the SPOOL parameter file is desired or a list of several or all entries is to be output. In some cases, the required memory is requested by the macro function (internal REQM) but has to be confirmed by the user. The minimum output area size can be taken from the table on page 194 or the DSECTs for the output areas. If the output area is too small for the amount of information requested, the remaining parts can be retrieved with one or more macro calls (with the same parameter list).

The SPSxxxD macros generate descriptions of the output areas as DSECT or data areas while the SPSPL macro only generates them as data areas. The full name for SPSxxxD and possible operand values for SPSPL PL=.... can be found in the table in "Short names for the INFO operand, minimum lengths and explanations" on page 194.

If an SPSINF call is issued while SPOOL is not loaded, code x'01000000' is returned in register 15 (this applies for all SPOOL calls).

Format

Operation	Operands
SPSINF	,INFO=info [,ITEM= \begin{cases} name \ (addr) \end{cases}] [,DVCTYP=device] [,SEQN=number] [,\begin{cases} [OUTPL=ad-\ dr][,OUTL=length] \end{cases}]
SPSxxxD SPSPL	$ [, \left\{ \begin{array}{l} [OUTPL=ad-\\ dr][,OUTL=length] \end{array} \right\}] $ $ [,MF= \left\{ \begin{array}{l} E\\ (E,ad-\\ dr)\\ L\\ (L,p)\\ C\\ (C,p)\\ D \end{array} \right\}] $ $ MF=(L[,p]) \ / \ (C[,p]) \ / \ (D[,p]) $ $ PL=list,P=p $

Description of the operands

INFO

Describes the information to be output via a short name. The short name can be specified as either a name or a number (in parentheses).

=info

Short name; see the list at the end of the operand description. Default: info=SPSGEN.

Note

If INFO=SPSDVC is specified and no entry is made for the ITEM operand, a device entry is still output, namely the general default device entry (DVCTYP=X'00').

ITEM

Designates the entry to be output by specifying a name or the address of a field containing the name. The name may include wildcards (see "Wildcards" on page 588 for a description of the wildcards). The name length is limited, but very long names can be specified using wildcards and an entry in the field.

Туре	Name length				
	without wildcards	with wildcards			
Device Device pool Character set Form	≤ 8 characters ≤ 8 characters ≤ 3 characters ≤ 6 characters	 ≤ 8 characters if specified directly or ≤ 24 characters if specified in the field 			

Wildcards cannot be used if the short names SPSDVC, SPSFRM, SPSPOL or SPSDVP are specified for INFO=...

=name

Name of a device, device pool, character set or form.

=(addr)

Address of the field containing the name. Field length \geq 24 bytes; entry left-justified and padded with blanks (X'40') to 24 bytes.

DVCTYP

Designates a device type. DVCTYP must be specified if INFO=SPSDVC is specified and no entry is made for ITEM=... The form entries for the specified device type are output. The following default applies in conjunction with INFO=SPSFRM: the first form entry in the SPOOL parameter file is output.

=type

Name of the device type.

SEQN

Must be specified in conjunction with INFO=SPSPOL or INFO=SPSDVP and designates the device entry to be output within the device pool or from the list of device entries (INFO=SPSDVP).

=number

Location of the entry within the device pool or its position in the device entry list (1st entry, 2nd entry, 3rd entry, etc.). $0 \le \text{number} \le 65535$; if number=0, the number of devices for each device type is output (number=0 is only meaningful with INFO=SPSPOL).

OUTPL

Address of the output area. Address specification is mandatory if the short name SPSGEN, SPSDVC, SPSFRM, SPSCHR, SPSPOL or SPSDVP is specified for INFO=...

Length of the output area. The minimum lengths are listed in the

=addr

Symbolic address of the output area.

OUTL

following table. Length specification is mandatory if the short name SPSPOL oder SPSDVP is specified for INFO=... If SPSGEN, SPSDVC, SPSFRM or SPSCHR is specified, the lengths listed in the table are used as default.

With INFO=SPSDVCA/SPSCHRA/SPSFRMA/SPSDAB/ SPSCAB/SPSFAB/SPSPOAB/PSDPAB, OUTL designates the part of the entries to be output. With OUTL=20, only the first 20 bytes of each entry are output.

Address of the output area if the short name SPSDLB, SBSFLB, SPSCLB, SPSPOLB, SPSDPLB, SPSDAB, SPSFAB, SPSFAB,

=length

Length of the output area.

BUFFER

SPSCAB, SPSOAB or SPSDPAB is specified. Address of the output area. The area must be aligned to word

=addr

boundaries.

BUFL

Length of the output area if BUFFER=... is specified.

=length

Length of the output area.

If BUFFER and BUFL are used, several calls are generally required to retrieve all entries. After processing ends, the name of the last

start point for the next call.

Before the first call, you must make sure that this field in the parameter list contains either null or blanks.

The name of this field is &P.KEY. The buffer is generally not filled with the last call. You can always definitely determine whether all entries have been read since making one call too many results in the return code X'08'.

entry output is stored in the parameter list and can be used as the

192

MF	Macro call form; see also the "Executive Macros" manual. Default value: standard form, i.e. the command part and the data area with the operand list are generated.
=E	E form: only the command part is generated.
=(E,addr)	addr = address of the operand list. Default value: the address must be passed in register R1.
=L	L form: a data area with the operand list is generated.
=(L,p)	The list contains the operand values specified in the call as well as symbolic names and explanatory equates. When the SPSxxxD macro is called, the output area is generated. p = prefix for the symbolic names; default: p=SPP.
=C	C form: a data area with the operand list is generated.
=(C,p)	Operand values specified in the call are not entered. The data area contains symbolic names and explanatory equates. When the SPSxxxD macro is called, the output area is generated. p = prefix for the symbolic names; default: p=SPP.
=D	D form: a dummy section (DSECT) for the operand list or output area is generated.
=(D,p)	p = prefix for the symbolic names in the DSECT; default: p=SPP.

Short names for the INFO operand, minimum lengths and explanations

Short name		Length	DSECT macro		Output	
Name	Num ber	std/min	Macro	SPSPL		
SPSGEN	(1)	63 / 16	SPSGEND	SPG	SPOOL global defaults	
SPSDVC	(2)	112 / 16	SPSDVCD	SPD	DVC entry for a device	
SPSCHR	(3)	60 / 16	SPSCHRD	SPC	CHR entry for a character set	
SPSFRM	(4)	64 / 17	SPSFRMD	SPF	FRM entry for a form	
SPSPOL	(5)	24		SPPOL	POL entry for a pool	
SPSDVP	(6)	24		SPDVP	DVP entry for a device-specific pool	
SPSDVCL	(12)	internal	SPSDVCLD	SPDL	List (subset) of device names	
SPSCHRL	(13)	internal	SPSCHRLD	SPCL	List (subset) of character set names	
SPSFRML	(14)	internal	SPSFRMLD	SPFL	List (subset) of form names	
SPSDLB	(15)	28	SPSDVCLD	SPDL	Subset of device names	
SPSCLB	(16)	28	SPSCHRLD	SPCL	Subset of character set names	
SPSFLB	(17)	29	SPSFRMLD	SPFL	Subset of form names	
SPSPOLB	(18)	10		SPPOL	Subset of pool names	
SPSDPLB	(19)	10		SPDVP	Subset of device-specific pool names	
SPSDVCA	(22)	internal	SPSDVCAD	SPDA	List (subset) of DEV entries	
SPSCHRA	(23)	internal	SPSCHRAD	SPCA	List (subset) of CHR entries	
SPSFRMA	(24)	internal	SPSFRMAD	SPFA	List (subset) of FRM entries	
SPSDAB	(25)	1)	SPSDVCAD	SPDA	List (subset) of DEV entries	
SPSCAB	(26)	1)	SPSCHRAD	SPCA	List (subset) of CHR entries	
SPSFAB	(27)	1)	SPSFRMAD	SPFA	List (subset) of FRM entries	
SPSPOAB	(28)	2)		SPPOL	List (subset) of POL entries	
SPSDPAB	(29)	2)		SPDVP	List (subset) of DVP entries	
SPSDVCC	(32)				Number of DEV entries	
SPSCHRC	(33)				number of CHR entries	
SPSFRMC	(34)				Number of FRM entries	
SPSPOLC	(35)				Number of POL entries	
SPSDVPC	(36)				Number of DVP entries	
SPSRELM	releas	releases the internally requested memory.				

Explanation of table columns

Short name:

Short name for the desired information. The name or number (in parentheses) are specified with INFO=....

Length: std/min

Standard and minimum length of the output area (OUTL oder BUFL). If it is not specified, the length must be determined using DSECT which describes the output area. "internal" means that the memory area is requested by the macro function. The address and length of the memory area are returned in the BUFA and BUFL fields of the operand list. The memory area is released via a subsequent SPSINF call with INFO=SPSRELM.

- Minimum value for BUFL: OUTL + 20;
 Minimum value for OUTL: 8;
- 2. Minimum value for BUFL: OUTL: Minimum value for OUTL: 8.

Macro:

These macro calls return descriptions of the output areas as a DSECT or data area for the INFO value on the left.

SPSPL:

The SPSPL macro returns descriptions of the output area as a data area for the INFO value on the left. The entries in the SPSPL column designate the operand values for the PL=... operand.

Output:

Brief description of the information output. A subset is always output if a name with wildcards is specified with ITEM=....

The number of DEV-/CHR-/FRM entries is output in the COUNT field of the operand list.

Return codes

A macro execution return code is always passed back in register R15 (left justified) and in the RETC field of the operand list.

X' aa'	Meaning
X' 00'	Normal execution
X' 04'	Invalid operand or incorrect operand list
X' 08'	Entry or entries in the SPOOL parameter file not found
X, 0C,	No memory space to output a list
X' 10'	System error

Example

The defaults defined in the SPOOL parameter file are to be output. The SPSGEND macro generates a description of the output area.

```
BEISPIEL START
        PRINT NOGEN
        BALR 3,0
        USING *,3
        SPSINF MF=(E, SPSINF) -
                                                                                        - (01)
TERM
        DS
              0F
SPSINF
        SPSINF INFO=SPSGEN,OUTPL=OUTADR1,OUTL=64,MF=L -
OUTADR1 DS
        END
        START-PROGRAM $ASSEMBH -
(IN)
                                                                                        - (03)
( OUT )
        % ASS6010 V 1.1A00 OF BS2000 ASSEMBH READY
(IN)
        COMPILE SOURCE=BEISPIEL, MACRO-LIBRARY=$RZV110.GCLIB.UR.V11.0S10.921012, LISTING=P
( )
        ARAMETERS(OUTPUT=LST.ASSEMBH), TEST-SUPPORT=YES
( OUT )
        % ASS6011 ASSEMBLY TIME: 1302 MSEC
(OUT)
        % ASS6018 O FLAGS, O PRIVILEGED FLAGS, O MNOTES
(OUT)
        % ASS6019 HIGHEST ERROR-WEIGHT: NO ERRORS
(OUT)
        % ASS6006 LISTING GENERATOR TIME: 618 MSEC
        END
(IN)
        % ASS6012 END OF ASSEMBH
(OUT)
(IN)
        LOAD-PROG *MOD(*OMF).TEST=AID -
                                                                                        -(04)
        % BLS0001 ### DBL VERSION 119 RUNNING
( OUT )
(OUT)
        % BLS0517 MODULE 'BEISPIEL' LOADED
        %in term
(IN)
(IN)
        %r
(TUO)
(OUT)
        STOPPED AT LABEL: TERM , SRC REF: 21 , SOURCE: BEISPIEL , PROC BEISPIEL
(IN)
        %d outadr1%x -
                                                                                         -(05)
        ** ITN: #'000B01BF' *** TSN: 1GY6 ********************************
(OUT)
                                CSECT: BEISPIEL ***************************
(NL)
        CURRENT PC: 00000018
        V'0000006C' = OUTADR1 + \#'00000000'
(NL)
        0000006C (00000000) 00000027 C7C5D540 F0F24BF7 C1F9F100
                                                                   ....GEN 02.7..
(NL)
        0000007C (00000010) 01010201 88020001 40404040 40404040
                                                                   ...h...
(NL)
(NI)
        0000008C (00000020) 020A0400 00C8000A 0064000A 01000101
                                                                   .....H.......
         0000009C (00000030) 0032C8C8 14360201 02010000 40010201
(NL)
                                                                   ..HH.......
```

- (01) The SPSINF macro is called in the E form.
- (02) The SPSINF macro is called in the L form. The defaults defined for printers, character sets, forms, spoolout options,, are to be output.
- (03) The (example) program BEISPIEL is compiled.
- (04) BEISPIEL is loaded and started. The output area is displayed using AID (Advanced Interactive Debugger).

(05) The entries in the output have the following meanings (see also the description of the output area below):

```
Version
                     : 39
                                 (Byte 0-3)
Info-Type
                     : GEN
                                 (Byte 4-7)
SPOOL-Version
                     : 02.7
                                 (Byte 8-14)
SPACE-AFTER-HEADER
                    : YES
                                 (Byte 16 = X'01')
FAMILY-MEMBER-HEADER: NO
                                 (Byte 17 = X'02')
                                 (Byte 18 = X'02')
COPY-HEADER
                     : NO
CHARACTER-SET
                     : FILE
                                 (Byte 19 = X'02')
PRINTER SIZE
                     : 136
                                 (Byte 20 = X'88')
LOCK DEFAULT
                     : NO
                                 (Byte 21 = X'02')
u.s.w
```

Output area layout (DSECT for the GEN entry)

```
LABFL1
            SPSGEND
1 LABEL1
            DS
                  0F
1
                                                                               С
            SPSPL PL=SPG.
1
                  P=SPG
2 SPGOUT
                  0 F
            DS
2 SPGVER
            DS
                                     VERSION
2 SPGINFO
           DS
                  CL4'GEN'
                                     INFO TYPE
2 SPGSPOOL DS
                  CL7
                                     SPOOL VERSION
                                                                            204
                                                                            204
            DS
                  0Н
                                     FOR ALIGNEMENT
2 SPGHSPAC DS
                  χ
                                     SPACE AFTER HEADER
2 SPGYES
            FOU
                  1
                                        YES
                  2
            EQU
2 SPGNO
                                        NO.
2 SPGHFAM
           DS
                  Χ
                                     FAMILY MEMBER HEADER
2 * SPGYES EQU
                  1
                                       YES
                  2
            FOU
2 * SPGNO
                                       NO.
2 SPGHCOP
            DS
                  Χ
                                     COPY HEADER
2 * SPGYES EOU
                  1
                                       YES
                  2
            FOU
2 * SPGNO
                                       NO.
2 SPGHCHR DS
                  Χ
                                     CHARACTER SET
                  1
2 SPGDEF
            EQU
                                        DEFAULT
2 SPGFILE
            FOU
                  2
                                        FILE
2 SPGPSIZE DS
                                     /PRINT PRINTER SIZE
                  Χ
2 SPG132
            EQU
                  132
                                        132
2 SPG136
            FOU
                  136
                                        136
2 SPG160
            EQU
                  160
                                          160
                                                                            254
                                     /PRINT LOCK DEFAULT
2 SPGPLOCK DS
                  Χ
2 * SPGYES EQU
                                       YES
                  1
2 * SPGNO
            EQU
                                       NO
                  FL1
                                     /PRINT SHIFT DEFAULT
2 SPGPSHIF DS
            DS
                  χ
                                     /PRINT DEFAULT DESTINATION
2 SPGPDES
2 SPGLOC
            EQU
                  1
                                        LOCAL
                  2
            EQU
2 SPGREM
                                        REMOTE
```

2	SPGPOL	EQU	3		POOL	
2	SPGPOLR	EQU	3		POOL REMOTE	332
2	SPGPOLL	EQU	4		POOL LOCAL	332
2	SPGPDVC	DS	CL8		/PRINT DEVICE NAME (IF PDES=SPGREM)	
2		DS	OH		FOR ALIGNMENT	
2	* SPGYES	EQU	1		YES	
2	* SPGNO	EQU	2		NO	
2	SPGSPUSC	DS	FL1		SPOOLOUT PUNCH SEPARATOR CARDS	
2	SPGRSOBS	DS	Χ		RSO BUFFER SIZE	
	SPGRBPLI		Χ		RBP_STATION_LIMIT	255
	SPGRSOLI		Н		RSO LIMIT	
	SPGSCKLI		Н		CHECKPOINT FOR LINE PRINTERS	
	SPGSCKPA		Н		PAGE PRINTERS	
	SPGSCKRE		Н		REMOTE PRINTERS	
	SPGSTUNI		X		/STATUS SIZE UNIT	
	SPGPP	EQU	1		PAM PAGES	
	SPGLI	EQU	2		LINES	
	SPGSTLPB		X		/STATUS LINES FACTOR	
	SPGFAMLY		X		FAMILY (PRINT -> SPOOL 2.5A)	309
	SPEMYES	EQU	1		YES	315
	SPEMNO	EQU	2		NO	315
			os introduced i	· NI 《		251
	SPGFDLO		X 10 INTRODUCED I	. IV		
					SPOOLIN LOGGING FLOPPY	250
	* SPGYES		1			250
	* SPGNO	EQU	2		CTART CROOL LOW WAL	250
	SPGSTRSP		Н		START-SPOOL-LOW-VAL	251
	SPGPRRSP		X		PRIORITY REPLAY SESSION	253
	SPGPRFPP		X		PRIORITY FROM POSITIONING	253
	SPGRMTLI		FL1		REMOTE-MESSGE-TASK	302
		DS	X		MIN_LINES_PER_PAGE	255
			DS INTRODUCED I	. N .		300
	SPGDND	DS	Χ		DEVICE-TYPE ND	300
	SPGDSD	DS	Χ		SD	300
	SPGDLP	DS	Χ		LP	300
	SPGDHP	DS	Χ		HP	300
	SPGHFOB	DS	Н		FORM-OVERLAY-BUFFER	300
	SPGHCS	DS	Χ		CHARACTER-SET-NUMBER	300
2	SPGHROT	DS	Χ		ROTATION	300
2	SPGEMPTF	DS	Χ	/	ACCEPT EMPTY FILE (-> PRINT CMD)	311
2	SPEFYES	EQU	1		YES	315
2	SPEFN0	EQU	2		NO	315
2	SPGTRUNC	DS	Χ	-	TRUNCATION (-> PRINT CMD)	311
2	SPFIGN	EQU	1		IGNORE	311
2	SPFKEEP	EQU	2		KEEP FILE	311
2	SPGCLSCH	DS	8X		CURRENT LOCAL SCHED PARAM	306
2	SPGCLPRI	DS	8X		CURRENT LOCAL PRIO. PARAM	306
2	SPGCLWEI	DS	0XL16		CURRENT LOCAL DVC WEIGTH	306
2	SPGCLDFO	DS	Χ		CURRENT LOCAL DVC WEIGTH FORM	306

198

2 SPGCLDD	I DS	Χ	CURRENT	LOCAL	DVC	WEIGTH	DIA	306
2 SPGCLDF	3 DS	Χ	CURRENT	LOCAL	DVC	WEIGTH	FOB NAME	306
2 SPGCLDCI	H DS	Χ	CURRENT	LOCAL	DVC	WEIGTH	CHAR-SET	306
2 SPGCLDRO	DS C	Χ	CURRENT	LOCAL	DVC	WEIGTH	ROTATION	306
2 SPGCLDFS	S DS	Χ	CURRENT	LOCAL	DVC	WEIGTH	FORM-SIZE	306
2 SPGCLDC ₇	# DS	Χ	CURRENT	LOCAL	DVC	WEIGTH	FOB CHR NUI	1306
2 SPGCLDPI	R DS	Χ	CURRENT	LOCAL	DVC	WEIGTH	PRIORITY	306
2 SPGCLDUS		Χ	CURRENT	LOCAL	DVC	WEIGTH	USERID	306
2 SPGCLDCI		Χ	CURRENT					306
2 SPGCLDF:	I DS	Χ	CURRENT	LOCAL	DVC	WEIGTH	FILE-SIZE	306
2 SPGCLDPI		Χ	CURRENT					306
2 SPGCLDA		Χ					ACCOUNT	306
2 SPGCLDA		Χ	CURRENT					306
2 SPGCLDS(X					HT SEQUENCE	
2 SPGCLDRS		X					RESERVED	331
2 SPGCRSCH		8X	CURRENT			ED PARAN		306
2 SPGCRPR		8X	CURRENT			D. PARAN		306
2 SPGCRWE		0XL16	CURRENT			WEIGTH	•	306
2 SPGCRDF(X	CURRENT			WEIGTH	FORM	306
2 SPGCRDD:		X	CURRENT			WEIGTH		306
2 SPGCRDF		X	CURRENT				FOB NAME	306
2 SPGCRDCH		X	CURRENT				CHAR-SET	306
2 SPGCRDR		X	CURRENT				ROTATION	306
2 SPGCRDFS		X	CURRENT				FORM-SIZE	306
2 SPGCRDC 1		X	CURRENT				FOB CHR NUI	
2 SPGCRDPI		X	CURRENT				PRIORITY	306
2 SPGCRDUS		X	CURRENT			WEIGTH		306
2 SPGCRDCI		X	CURRENT			WEIGTH		306
2 SPGCRDF:		X	CURRENT				FILE-SIZE	306
2 SPGCRDPI		X	CURRENT			WEIGTH		306
2 SPGCRDA		X	CURRENT				ACCOUNT	306
2 SPGCRDA(2 SPGCRDA(X	CURRENT			WEIGTH		306
				NT RSO				
2 SPGCRDS(X					HT SEQUENCE	
2 SPGCRDRS		X	CURRENT				RESERVED	306
2 SPGNLSCH		8X	NEXT			ED PARAN		306
2 SPGNLPR		8X	NEXT			O. PARAN	1	306
2 SPGNLWE		0XL16	NEXT			WEIGTH	FORM	306
2 SPGNLDF(X	NEXT			WEIGTH		306
2 SPGNLDD:		X	NEXT			WEIGTH		306
2 SPGNLDF		X	NEXT				FOB NAME	306
2 SPGNLDCH		X	NEXT				CHAR-SET	306
2 SPGNLDRO		X	NEXT				ROTATION	306
2 SPGNLDFS		X	NEXT				FORM-SIZE	306
2 SPGNLDC		X	NEXT				FOB CHR NUI	
2 SPGNLDPI		X	NEXT				PRIORITY	306
2 SPGNLDUS		X	NEXT			WEIGTH		306
2 SPGNLDCI		X	NEXT			WEIGTH		306
2 SPGNLDF:	L DS	Χ	NEXT	LUCAL	DAC	WEIGTH	FILE-SIZE	306

	2 SPGNLDPN		Χ	NEXT			WEIGTH		306
	2 SPGNLDAC		Χ	NEXT				ACCOUNT	306
	2 SPGNLDAG		Χ	NEXT			WEIGTH		306
i	2 SPGNLDSQ	DS	Χ	NEXT	LOCA	۹L D۱	VC WEIG	HT SEQUENCE	
i	2 SPGNLDRE	DS	Χ	NEXT	LOCAL	DVC	WEIGTH	RESERVED	306
i	2 SPGNRSCH	DS	8X	NEXT	RS0	SCH	ED PARAI	M	306
i	2 SPGNRPRI	DS	8X	NEXT	RS0	PRI(O. PARAI	M	306
i	2 SPGNRWEI	DS	OXL16	NEXT	RS0	DVC	WEIGTH		306
i	SPGNRDF0	DS	Χ	NEXT	RS0	DVC	WEIGTH	FORM	306
i	2 SPGNRDDI	DS	Χ	NEXT	RS0	DVC	WEIGTH	DIA	306
i	2 SPGNRDFB	DS	Χ	NEXT	RS0	DVC	WEIGTH	FOB NAME	306
i	SPGNRDCH	DS	Χ	NEXT	RS0	DVC	WEIGTH	CHAR-SET	306
i	SPGNRDRO	DS	Χ	NEXT	RS0	DVC	WEIGTH	ROTATION	306
1	SPGNRDFS	DS	Χ	NEXT	RS0	DVC	WEIGTH	FORM-SIZE	306
1	SPGNRDC#	DS	Χ	NEXT	RS0	DVC	WEIGTH	FOB CHR NU	M306
i	SPGNRDPR	DS	Χ	NEXT	RS0	DVC	WEIGTH	PRIORITY	306
i	SPGNRDUS	DS	Χ	NEXT	RS0	DVC	WEIGTH	USERID	306
1	SPGNRDCL	DS	Χ	NEXT	RS0	DVC	WEIGTH	CLASS	306
i	SPGNRDFI	DS	Χ	NEXT	RS0	DVC	WEIGTH	FILE-SIZE	306
i	SPGNRDPN	DS	Χ	NEXT	RS0	DVC	WEIGTH	PNAME	306
i	SPGNRDAC	DS	Χ	NEXT	RS0	DVC	WEIGTH	ACCOUNT	306
	SPGNRDAG	DS	Χ	NEXT	RS0	DVC	WEIGTH	AGE	306
	SPGNRDSQ	DS	Χ	NEXT	RS0	D١	VC WEIG	HT SEQUENCE	101
i	SPGNRDRE	DS	Χ	NEXT	RS0	DVC	WEIGTH	RESERVED	306
	SPGP00L	DS	CL8	POOL NA	ME				306
i	SPGCTR#	DS	Н	MAX N	UMBER (OF CO	ONTROLL	ER	306
i	SPGDVC#	DS	Н	MAX N	UMBER (OF DE	EVICE/C	NTRL	306
i	SPGCHKTI	DS	Χ	CHECK	TIME	VALUE	E		306
i	SPGACCNB	DS	Χ	PRINT AC	COUNT I	NUMB	ER (PRI	NT CMD)	316
1	SPACYES	EQU	1	YES					315
i	SPACNO	EQU	2	NO					315
1	SPGRELMM	DS	Χ	RELEA	SE MEMO	ORY ((YES-NO)	306
i	SPGFNMLG	DS	Χ	FORM	NAME LI	ENGTH	H (SD CI	MD)	316
1	SPGRPTIM	DS	Н	REPEA	T TIME				306
i	SPGMLPID	DS	Χ	MIN L-P-	P INDI	C			318
1	2 SPIDNSP	EQU	0	NOT	SPECI	FIED			318
i	SPIDSP	EQU	1	SPE	CIFIED				318
i	2 SPGSCMAX	DS	Χ	# MAXIM	UM OF S	SECT	ION CTR	L	319
i	SPGND50L	DS	Χ	DEVICE-	TYPE NI	D50-I	LINE-PR	INTER	330
i	SPMND50Y	EQU	1	YES					330
i	SPMND50N	EQU	2	NO					330
i	SPGRSOFM	DS	Χ	RSO FAM	ILY PRO	OCES:	SING		330
i	SPMRFMY	EQU	1	YES					330
	SPMRFMN	EQU	2	NO					330
	SPGLSTFR	DS	CL8	DEFAU	LT SYSI	LST I	FORM		
i	SPGOUTFR	DS	CL8	DEFAU	LT SYS	TUC	FORM		
	SPGNDX	DS	Χ	DEVICE-	TYPE NI	DX-PI	RINTER		330
2	SPMNDXY	EQU	1	YES					330

200

2 SPMNDXN EQU 2 NO 330 2 SPMFOY EQU 1 YES 330 2 SPMFON EQU 2 NO 330 2 SPMFON EQU 1 1 YES 330 2 SPMFON EQU 2 NO 330 2 SPMFON EQU 1 1 1800 330 2 SPMFON EQU 2 3600 330 2 SPMFO EQU 1 1800 330 2 SPMFO EQU 2 3600 330 2 SPMFO EQU 4 2048 2 SPM10 EQU 8 4096 2 SPM12288 EQU 32 12288 406 2 SPM12288 EQU 32 12288 406 2 SPM16384 EQU 64 16384 406 2 SPM187N EQU 1 YES 406 2 SPMNXRN EQU 1 YES 406 2 SPMNXRN EQU 2 NO 406 2 SPMNXRN EQU 1 YES 406 2 SPMNXRN EQU 2 NO 406 2 SPMSYN EQU 1 YES 406 2 SPMSYN EQU 1 YES 406 2 SPMSYN EQU 2 NO 406 2 SPENSON DS X NOX-PRINTER : FORM-OVERLAY-BUFFER 300 2 SPENSON DS X NOX-PRINTER : CHARACTER-SET-NUMBER 330 2 SPENSON DS X NOX-PRINTER : CHARACTER-SET-NUMBER 330 2 SPENSON DS X DEVICE-TYPE APA-PRINTER 400 2 SPMSYN EQU 1 YES 400 2 SPMSYN EQU 1 YES 400 2 SPMSYN EQU 2 NO 400 2 SPENSON DS X DEVICE-TYPE APA-PRINTER 400 2 SPMSYN EQU 1 YES 400 2 SPENSON DS X DEVICE-TYPE APA-PRINTER 400 2 SPMSYN EQU 2 NO 400 2 SPENSON DS X DEVICE-TYPE APA-PRINTER 400 2 SPMSYN EQU 1 YES 400 2 SPMSYN EQU 1 YES 400 2 SPMSYN EQU 1 YES 400 2 SPMSYN EQU 2 NO 400 2 SPENSON DS X DEVICE-TYPE TWIN-PRINTER *620 2 SPMSYN EQU 1 YES 400 3 SPMSYN EQU 2 NO 400 3 SPMSYN EQU 1 YES 400 400 400 400 400 400 400 400 400 400						
2 SPMFOY EQU 1 YES 330 2 SPMFON EQU 2 NO 330 2 SPGPM DS X NDX-PRINTER: RASTER PATTERN MEMORY 330 2 SPM36 EQU 2 3600 330 2 SPM36 EQU 2 3600 330 2 SPM20 EQU 4 2048 2 SPM20 EQU 8 4096 2 SPM36 EQU 32 12288 406 2 SPM363 EQU 40 EQU 8 80 2 SPM363 EQU 50	2	SPMNDXN	EQU	2	NO	330
2 SPMFON EQU 2 NO 330 2 SPGRPM DS X NDX-PRINTER: RASTER PATTERN MEMORY 330 2 SPM8 EQU 1 1800 330 2 SPM20 EQU 4 2048 2 SPM8192 EQU 16 8192 406 2 SPM16384 EQU 64 16384 406 2 SPM16385 EQU 64 16384 406 2 SPM16386 EQU 64 16384 406 2 SPMNXRV EQU 1 YES 406 2 SPMNXRY EQU 1 YES 406 2 SPMXFOB DS H NDX-PRINTER: FORM-OVERLAY-BUFFER 30 2 SPGNXFOB DS H NDX-PRINTER: FORM-OVERLAY-BUFFER 30 2 SPGNXFOB DS N NDX-PRINTER: FORM-OVERLAY-BUFFER 330 2 SPGNXFOB DS X PRIORITY SECTION POSITIONING 330 2 SPGNXFOB DS X PRIORITY SECTION POSITIONING	2	SPGNDXFO	DS	Χ	NDX-PRINTER : FORMS OVERLAY	330
2 SPGRPM DS X NDX-PRINTER: RASTER PATTERN MEMORY 330 2 SPM36 EQU 1 1 1800 330 2 SPM36 EQU 2 3600 330 2 SPM36 EQU 4 2048 2 SPM40 EQU 8 4096 2 SPM8192 EQU 16 8192 406 2 SPM12288 EQU 32 12288 406 2 SPM1288 EQU 32 12288 406 2 SPM16384 EQU 64 16384 406 2 SPM187Y EQU 1 YES 406 2 SPMNXRY EQU 1 YES 406 2 SPMNXRY EQU 1 YES 406 2 SPMNXRY EQU 1 YES 406 2 SPGNXFOB DS H NDX-PRINTER: FORM-OVERLAY-BUFFER 330 2 SPGNXFOB DS X PRIORITY SECTION POSITIONING 339 2 SPGAPA DS X PRIORITY SECTION POSITIONING 339 2 SPGAPA EQU 1 YES 400 2 SPMRYAPY EQU 1 YES 4620 2 SPMRYAPY EQU 1 YES 4620 2 SPMGTWY EQU 1 YES 4620 3 SPMGTWY EQU 1 YES 4620 4620 4620 4630 4640 4660 4660 4660 4660 4660 466	2	SPMFOY	EQU	1	YES	330
2 SPM36 EQU 2 3600 330 2 SPM20 EQU 4 2048 2 SPM20 EQU 8 4096 2 SPM8192 EQU 16 8192 406 2 SPM12288 EQU 32 12288 406 2 SPM16384 EQU 64 16384 406 2 SPM187 EQU 1 YES 406 2 SPMNXRY EQU 1 YES 406 2 SPMXED DS X NDX-PRINTER : FORM-OVERLAY-BUFFER 330 2 SPGRXFOB DS X DEVICE-TYPE APA-PRINTER 400 2 SPMAPAY EQU 1 YES 400 2 SPMGTWY EQU 1 YES 400 2 SPMGTWY EQU 1 YES 400 2 SPMGTWY EQU 1 YES 4620 3 SPMGTWY EQU 1 YES 4620 4620 4620 4620 4620 4620 4630 464 466 465 465 465 465 4665 4665 4665	2	SPMFON	EQU	2	NO	330
2 SPM36 EQU 2 3600 330 2 SPM20 EQU 4 2048 2 SPM40 EQU 8 4096 2 SPM192 EQU 16 8192 406 2 SPM12288 EQU 32 12288 406 2 SPM16384 EQU 64 16384 406 2 SPM16384 EQU 64 16384 406 2 SPM17 EQU 1 YES 406 2 SPM17 EQU 1 YES 406 2 SPM17 EQU 1 YES 406 2 SPM17 EQU 2 NO 406 2 SPM17 EQU 2 NO 406 2 SPM17 EQU 1 YES 406 2 SPM18 EQU 2 NO 406 2 SPM18 EQU 1 YES 400 3 SPM18 EQU 2 NO 400 4 SPM18 EQU 1 YES 400 4 SPM18 EQU 1	2	SPGRPM	DS	Χ	NDX-PRINTER: RASTER PATTERN MEMORY	330
2 SPM20 EQU 4 2048 2 SPM40 EQU 8 4096 2 SPM12288 EQU 16 8192 406 2 SPM12288 EQU 32 12288 406 2 SPM12384 EQU 64 16384 406 2 SPMNXRY EQU 1 YES 406 2 SPMNXRY EQU 1 YES 406 2 SPMXFOB DS H NDX-PRINTER: FORM-OVERLAY-BUFFER 330 2 SPGNXFOB DS H NDX-PRINTER: FORM-OVERLAY-BUFFER 330 2 SPGNXFOB DS H NDX-PRINTER: FORM-OVERLAY-BUFFER 330 2 SPGRAPO DS X NDX-PRINTER: FORM-OVERLAY-BUFFER 330 2 SPGRAPO DS X NDX-PRINTER: FORM-OVERLAY-BUFFER 330 2 SPGRAPO DS X NDX-PRINTER: FORM-OVERLAY-BUFFER 330 32 SPGNXFOB DS X DEVICE-TYPE TWIN-PRINTER 400 2 SPGAPA DS X DEVICE-	2	SPM18	EQU	1	1800	330
2 SPM40 EQU 8 4096 2 SPM8192 EQU 16 8192 406 2 SPM16384 EQU 32 12288 406 2 SPM16384 EQU 64 16384 406 2 SPGNXROT DS X DEVICE—TYPE APA—PRINTER 406 2 SPMNXRY EQU 1 YES 406 2 SPMNXRN EQU 2 NO 406 2 SPMNXRN EQU 2 NO 406 2 SPGNXCSN DS X NDX—PRINTER: FORM—OVERLAY—BUFFER 330 2 SPGNXCSN DS X NDX—PRINTER: CHARACTER—SET—NUMBER 330 2 SPGNXCSN DS X NDX—PRINTER: CHARACTER—SET—NUMBER 330 2 SPGAPA DS X DEVICE—TYPE APA—PRINTER 400 2 SPGAPA DS X DEVICE—TYPE APA—PRINTER 400 2	2	SPM36	EQU	2	3600	330
2 SPM8192 EQU 16 8192 406 2 SPM12288 EQU 32 12288 406 2 SPM16384 EQU 64 16384 406 2 SPMNXRY EQU 1 YES 406 2 SPMNXRY EQU 1 YES 406 2 SPMNXRY EQU 2 NO 406 2 SPMRY DS K NDX-PRINTER: FORM-OVERLAY-BUFFER 330 2 SPGNXCSN DS X NDX-PRINTER: CHARACTER-SET-NUMBER 330 2 SPGNXCSN DS X PRIORITY SECTION POSITIONING 330 2 SPGRAPA DS X DEVICE-TYPE APA-PRINTER 400 2 SPGAPA DS X DEVICE-TYPE APA-PRINTER *620	2	SPM20	EQU	4	2048	
2 SPM12288 EQU 32 12288 406 2 SPM16384 EQU 64 16384 406 2 SPMNXROT DS X DEVICE-TYPE APA-PRINTER 406 2 SPMNXRN EQU 1 YES 406 2 SPMNXRN EQU 2 NO 406 2 SPGNXCSN DS H NDX-PRINTER: FORM-OVERLAY-BUFFER 330 2 SPGNXCSN DS X NDX-PRINTER: CHARACTER-SET-NUMBER 330 2 SPGRSCP DS X PRIORITY SECTION POSITIONING 339 2 SPGAPA DS X DEVICE-TYPE APA-PRINTER 400 2 SPMAPAY EQU 1 YES 400 2 SPMAPAN EQU 2 NO 400 2 SPMGTHY EQU 1 YES 620 2 SPMGTHY EQU 1 YES 620 2 SPMGTHY EQU 1 YES 620 2 SPGRCRP DS CL38 RECOVERY-RULES-PREFIX 6620 2 SPGDDST DS AL1 RESOURCE LOCATION 643 2 SPGDBSTL DS AL1	2	SPM40	EQU	8	4096	
2 SPM16384 EQU 64 16384 406 2 SPGNXROT DS X DEVICE—TYPE APA—PRINTER 406 2 SPMNXRY EQU 1 YES 406 2 SPMNXRN EQU 2 NO 406 2 SPGNXFOR DS H NDX—PRINTER: FORM—OVERLAY—BUFFER 330 2 SPGNXCSN DS X NDX—PRINTER: CHARACTER—SET—NUMBER 330 2 SPGRXCSN DS X PRIORITY SECTION POSITIONING 339 2 SPGAPA DS X DEVICE—TYPE APA—PRINTER 400 2 SPMAPAY EQU 1 YES 400 2 SPMAPAN EQU 2 NO 400 2 SPGWIN DS X DEVICE—TYPE TWIN—PRINTER *620 3 SPMGTWY EQU 1 YES 6620 3 SPMGTWY EQU 1 YES 6620 3 SPMGTWY EQU 1 YES 6620 4 SPGDDST DS AL1 DPRINT DESTINATION 6643 4 SPGDDOL DS CL8 DPRINT DEFAULT POOL 6643 4 SPGDSTL DS AL1 SERVER TASK LIMIT 6630 4 SPGDSTL DS AL1 RESOURCE LOCATION 6643 4 SPGDRTL DS AL1 RESOURCE TASK LIMIT 6630 4 SPGDRTL DS A	2	SPM8192	EQU	16	8192	406
2 SPGNXROT DS X DEVICE—TYPE APA—PRINTER 406 2 SPMNXRY EQU 1 YES 406 2 SPMNXRN EQU 2 NO 406 2 SPGNXFOB DS H NDX—PRINTER : FORM—OVERLAY—BUFFER 330 2 SPGNXCSN DS X NDX—PRINTER : CHARACTER—SET—NUMBER 330 2 SPGAPA DS X PRIORITY SECTION POSITIONING 339 2 SPGAPA DS X DEVICE—TYPE APA—PRINTER 400 2 SPMAPAY EQU 1 YES 400 2 SPMAPAN EQU 2 NO 400 2 SPMAPAN EQU 1 YES 400 2 SPMGTWY EQU 1 YES 4620 2 SPMGTWY EQU 1 YES 4620 2 SPMGTWY EQU 1 YES 4620 2 SPMGTWN EQU 2 NO 4620 2 SPGRCP DS CL38 RECOVERY—RULES—PREFIX 4625 3 SPGDDST DS AL1 DPRINT DESTINATION 4643 3 SPGDPOL DS CL8 DPRINT DEFAULT POOL 4643 2 SPGDRSL DS AL1 RESOURCE LOCATION 4643 2 SPGDRSL DS AL1 RESOURCE LOCATION 4643 2 SPGDRSL DS AL1 RESOURCE TASK LIMIT 4630 2 SPGDRSL DS AL1 RESOURCE TASK LIMIT 4630 2 SPGDRSL DS AL1 RESOURCE TASK LIMIT 4630 3 SPGDRSL DS AL1 RESOURCE TASK LIMIT 4630 4 SPGDRSL DS AL1 RESOURCE TASK LIMIT	2	SPM12288	EQU	32	12288	406
2 SPMNXRY EQU 1 YES 406 2 SPMNXRN EQU 2 NO 406 2 SPGNXFOB DS H NDX-PRINTER: FORM-OVERLAY-BUFFER 330 2 SPGNXCSN DS X NDX-PRINTER: CHARACTER-SET-NUMBER 330 2 SPGPRSCP DS X PRIORITY SECTION POSITIONING 339 2 SPGAPA DS X DEVICE-TYPE APA-PRINTER 400 2 SPMAPAY EQU 1 YES 400 2 SPMGTWIN DS X DEVICE-TYPE APA-PRINTER 400 2 SPMGTWIN EQU 2 NO 400 2 SPMGTWIN DS X DEVICE-TYPE TWIN-PRINTER *620 2 SPMGTWIN EQU 1 YES *620 2 SPMGTWIN EQU 1 YES *620 2 SPMGTWIN EQU 2 NO *620 2 SPGDRTU DS CL38 RECOVERY-RULES-PREFIX *625 2 SPGDRSD DS AL1 DPRINT DEFAULT POOL *643 2 SPGDRSL DS AL1	2	SPM16384	EQU	64	16384	406
2 SPMNXRY EQU 1 YES 406 2 SPMNXRN EQU 2 NO 406 2 SPGNXFOB DS H NDX-PRINTER: FORM-OVERLAY-BUFFER 330 2 SPGNXCSN DS X NDX-PRINTER: CHARACTER-SET-NUMBER 330 2 SPGPRSCP DS X PRIORITY SECTION POSITIONING 339 2 SPGAPA DS X DEVICE-TYPE APA-PRINTER 400 2 SPMAPAY EQU 1 YES 400 2 SPMGTWIN DS X DEVICE-TYPE APA-PRINTER 400 2 SPMGTWIN EQU 2 NO 400 2 SPMGTWIN DS X DEVICE-TYPE TWIN-PRINTER *620 2 SPMGTWIN EQU 1 YES *620 2 SPMGTWIN EQU 1 YES *620 2 SPMGTWIN EQU 2 NO *620 2 SPGDRTU DS CL38 RECOVERY-RULES-PREFIX *625 2 SPGDRSD DS AL1 DPRINT DEFAULT POOL *643 2 SPGDRSL DS AL1				Χ	DEVICE-TYPE APA-PRINTER	
2 SPGNXFOB DS H NDX-PRINTER: FORM-OVERLAY-BUFFER 330 2 SPGNXCSN DS X NDX-PRINTER: CHARACTER-SET-NUMBER 330 2 SPGPRSCP DS X PRIORITY SECTION POSITIONING 339 2 SPGAPA DS X DEVICE-TYPE APA-PRINTER 400 2 SPMAPAY EQU 1 YES 400 2 SPMAPAN EQU 2 NO 400 2 SPGTWIN DS X DEVICE-TYPE TWIN-PRINTER *620 2 SPMGTWY EQU 1 YES *620 2 SPMGTWY EQU 1 YES *620 2 SPGCRP DS CL38 RECOVERY-RULES-PREFIX *625 2 SPGDDST DS AL1 DPRINT DESTINATION *643 2 SPGDPOL DS CL8 DPRINT DEFAULT POOL *643 2 SPGDRSL DS AL1 RESOURCE LOCATION *643 2 SPGDRSL DS AL1 RESOURCE LOCATION *643 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRTD DS X RESERVED *645 2 SPGDOCT DS H MAX LOGICAL MSG LENGTH *645 2 SPGDOCT DS H MAX LOGICAL MSG LENGTH *645 2 SPGGRSD DS 34X RESERVED *645 2 SPGGRSD DS 34X RESERVED *645 2 SPGGRSD DS 34X RESERVED *645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64					YES	
2 SPGNXCSN DS X NDX-PRINTER: CHARACTER-SET-NUMBER 330 2 SPGPRSCP DS X PRIORITY SECTION POSITIONING 339 2 SPGAPA DS X DEVICE-TYPE APA-PRINTER 400 2 SPMAPAY EQU 1 YES 400 2 SPMAPAN EQU 2 NO 400 2 SPGTWIN DS X DEVICE-TYPE TWIN-PRINTER *620 2 SPMGTWY EQU 1 YES *620 2 SPMGTWY EQU 1 YES *620 2 SPGRCRP DS CL38 RECOVERY-RULES-PREFIX *625 2 SPGDDST DS AL1 DPRINT DESTINATION *643 2 SPGDDST DS AL1 RESOURCE LOCATION *643 2 SPGDSTL DS AL1 RESOURCE LOCATION *643 2 SPGDSTL DS AL1 SERVER TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 3 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 4 SPGDRTL DS AL1 RESERVED *645 4 SPGTALN DS H TRANSFER ADMISSION LENGTH *646 4 SPGGRSD DS 34X RESERVED *645 4 SPGGRSD DS 34X RESERVED *645 4 SPGGRSD DS 34X RESERVED *645 4 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64	2	SPMNXRN	EQU	2	NO	406
2 SPGPRSCP DS X DEVICE—TYPE APA—PRINTER 400 2 SPMAPAY EQU 1 YES 400 2 SPMAPAN EQU 2 NO 400 2 SPGTWIN DS X DEVICE—TYPE TWIN—PRINTER *620 2 SPMGTWY EQU 1 YES *620 2 SPMGTWY EQU 1 YES *620 2 SPMGTWN EQU 2 NO *620 2 SPGRCRP DS CL38 RECOVERY—RULES—PREFIX *625 2 SPGDDST DS AL1 DPRINT DESTINATION *643 2 SPGDDST DS AL1 RESOURCE LOCATION *643 2 SPGDRSL DS AL1 RESOURCE LOCATION *643 2 SPGDSTL DS AL1 SERVER TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRTD DS X RESERVED *645 2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDRTD DS H MAX LOGICAL MSG LENGTH *646 2 SPGTATP DS X TRANSFER ADMISSION LENGTH *646 2 SPGGRSD DS 34X RESERVED *645 2 SPGGRSD DS 34X RESERVED *645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64	2	SPGNXFOB	DS	Н	NDX-PRINTER : FORM-OVERLAY-BUFFER	330
2 SPGAPA DS X DEVICE-TYPE APA-PRINTER 400 2 SPMAPAY EQU 1 YES 400 2 SPMAPAN EQU 2 NO 400 2 SPGTWIN DS X DEVICE-TYPE TWIN-PRINTER *620 2 SPMGTWY EQU 1 YES *620 2 SPMGTWN EQU 2 NO *620 2 SPGRCRP DS CL38 RECOVERY-RULES-PREFIX *625 2 SPGDDST DS AL1 DPRINT DESTINATION *643 2 SPGDPOL DS CL8 DPRINT DEFAULT POOL *643 2 SPGDRSL DS AL1 RESOURCE LOCATION *643 2 SPGDSTL DS AL1 SERVER TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRJP DS XL32 TRANSFER ADMISSION *630 2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDTAD DS X L32 TRANSFER ADMISSION LENGTH *646 2 SPGTATP DS X TRANSFER ADMISSION SDF TYPE *646 2 SPGGRSD DS 34X RESERVED 645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64	2	SPGNXCSN	DS	Χ	NDX-PRINTER: CHARACTER-SET-NUMBER	330
2 SPMAPAY EQU 1 YES 400 2 SPMAPAN EQU 2 NO 400 2 SPGTWIN DS X DEVICE—TYPE TWIN—PRINTER *620 2 SPMGTWY EQU 1 YES *620 2 SPMGTWN EQU 2 NO *620 2 SPGRCRP DS CL38 RECOVERY—RULES—PREFIX *625 2 SPGDDST DS AL1 DPRINT DESTINATION *643 2 SPGDRSL DS AL1 RESOURCE LOCATION *643 2 SPGDSTL DS AL1 RESOURCE LOCATION *643 2 SPGDSTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRJP DS AL1 RESERVED *645 2 SPGDTAD DS X RESERVED *645 2 SPGTALN DS H MAX LOGICAL MSG LENGTH *646 2 SPGTATP DS X TRANSFER ADMISSION SDF TYPE *646 2 SPGLEN EQU *-SPGOUT				Χ	PRIORITY SECTION POSITIONING	
2 SPMAPAN EQU 2 NO 400 2 SPGTWIN DS X DEVICE—TYPE TWIN—PRINTER *620 2 SPMGTWY EQU 1 YES *620 2 SPMGTWN EQU 2 NO *620 2 SPGRCRP DS CL38 RECOVERY—RULES—PREFIX *625 2 SPGDDST DS AL1 DPRINT DESTINATION *643 2 SPGDPOL DS CL8 DPRINT DEFAULT POOL *643 2 SPGDRSL DS AL1 RESOURCE LOCATION *643 2 SPGDRSL DS AL1 SERVER TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRJP DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRJP DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRJP DS AL1 REMOTE JOB PRIORITY *630 2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDOCT DS H MAX LOGICAL MSG LENGTH *645 2 SPGTALN DS H TRANSFER ADMISSION LENGTH *646 2 SPGTATP DS X RESERVED *645 2 SPGRSD DS 34X RESERVED *645 2 SPGRSD DS 34X RESERVED *645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64	2	SPGAPA	DS	Χ	DEVICE-TYPE APA-PRINTER	400
2 SPGTWIN DS X DEVICE—TYPE TWIN—PRINTER *620 2 SPMGTWY EQU 1 YES *620 2 SPMGTWN EQU 2 NO *620 2 SPGRCRP DS CL38 RECOVERY—RULES—PREFIX *625 2 SPGDDST DS AL1 DPRINT DESTINATION *643 2 SPGDPOL DS CL8 DPRINT DEFAULT POOL *643 2 SPGDRSL DS AL1 RESOURCE LOCATION *643 2 SPGDRSL DS AL1 SERVER TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRJP DS AL1 REMOTE JOB PRIORITY *630 2 SPGDRJP DS X RESERVED *645 2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDCT DS H MAX LOGICAL MSG LENGTH *645 2 SPGTALN DS H TRANSFER ADMISSION LENGTH *646 2 SPGRSD DS 34X RESERVED *645 2 SPGRSD DS 34X RESERVED 645 2 SPGLEN EQU *—SPGOUT OUTADR1 DS CL64	2	SPMAPAY	EQU	1	YES	400
2 SPGTWIN DS X DEVICE—TYPE TWIN—PRINTER *620 2 SPMGTWY EQU 1 YES *620 2 SPMGTWN EQU 2 NO *620 2 SPGRCRP DS CL38 RECOVERY—RULES—PREFIX *625 2 SPGDDST DS AL1 DPRINT DESTINATION *643 2 SPGDPOL DS CL8 DPRINT DEFAULT POOL *643 2 SPGDRSL DS AL1 RESOURCE LOCATION *643 2 SPGDRSL DS AL1 SERVER TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRJP DS AL1 REMOTE JOB PRIORITY *630 2 SPGDRJP DS X RESERVED *645 2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDCT DS H MAX LOGICAL MSG LENGTH *645 2 SPGTALN DS H TRANSFER ADMISSION LENGTH *646 2 SPGRSD DS 34X RESERVED *645 2 SPGRSD DS 34X RESERVED 645 2 SPGLEN EQU *—SPGOUT OUTADR1 DS CL64					NO	
2 SPMGTWN EQU 2 NO *620 2 SPGRCRP DS CL38 RECOVERY-RULES-PREFIX *625 2 SPGDDST DS AL1 DPRINT DESTINATION *643 2 SPGDPOL DS CL8 DPRINT DEFAULT POOL *643 2 SPGDRSL DS AL1 RESOURCE LOCATION *643 2 SPGDSTL DS AL1 SERVER TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRJP DS AL1 REMOTE JOB PRIORITY *630 2 SPGDRJP DS X RESERVED *645 2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDOCT DS H MAX LOGICAL MSG LENGTH *645 2 SPGTALN DS H TRANSFER ADMISSION LENGTH *646 2 SPGTATP DS X TRANSFER ADMISSION SDF TYPE *646 2 SPGGRSD DS 34X RESERVED 645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64	2	SPGTWIN			DEVICE-TYPE TWIN-PRINTER	*620
2 SPMGTWN EQU 2 NO *620 2 SPGRCRP DS CL38 RECOVERY-RULES-PREFIX *625 2 SPGDDST DS AL1 DPRINT DESTINATION *643 2 SPGDPOL DS CL8 DPRINT DEFAULT POOL *643 2 SPGDRSL DS AL1 RESOURCE LOCATION *643 2 SPGDSTL DS AL1 SERVER TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRJP DS AL1 REMOTE JOB PRIORITY *630 2 SPGDRJP DS X RESERVED *645 2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDOCT DS H MAX LOGICAL MSG LENGTH *645 2 SPGTALN DS H TRANSFER ADMISSION LENGTH *646 2 SPGTATP DS X TRANSFER ADMISSION SDF TYPE *646 2 SPGRSD DS 34X RESERVED 645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64	2	SPMGTWY	EQU	1	YES	*620
2 SPGRCRP DS CL38 RECOVERY-RULES-PREFIX *625 2 SPGDDST DS AL1 DPRINT DESTINATION *643 2 SPGDPOL DS CL8 DPRINT DEFAULT POOL *643 2 SPGDRSL DS AL1 RESOURCE LOCATION *643 2 SPGDSTL DS AL1 SERVER TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRJP DS AL1 REMOTE JOB PRIORITY *630 2 SPGDRJP DS X RESERVED *645 2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDOCT DS H MAX LOGICAL MSG LENGTH *645 2 SPGTALN DS H TRANSFER ADMISSION LENGTH *646 2 SPGTATP DS X TRANSFER ADMISSION SDF TYPE *646 2 SPGRSD DS 34X RESERVED 645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64					NO	*620
2 SPGDDST DS AL1 DPRINT DESTINATION *643 2 SPGDPOL DS CL8 DPRINT DEFAULT POOL *643 2 SPGDRSL DS AL1 RESOURCE LOCATION *643 2 SPGDSTL DS AL1 SERVER TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRJP DS AL1 REMOTE JOB PRIORITY *630 2 DS X RESERVED *645 2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDOCT DS H MAX LOGICAL MSG LENGTH *645 2 SPGTALN DS H TRANSFER ADMISSION LENGTH *646 2 SPGTATP DS X TRANSFER ADMISSION LENGTH *646 2 SPGRSD DS 34X RESERVED *645 2 SPGRSD DS 34X RESERVED 645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64	2	SPGRCRP	DS	CL38		
2 SPGDPOL DS CL8 DPRINT DEFAULT POOL *643 2 SPGDRSL DS AL1 RESOURCE LOCATION *643 2 SPGDSTL DS AL1 SERVER TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRJP DS AL1 REMOTE JOB PRIORITY *630 2 DS X RESERVED *645 2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDOCT DS H MAX LOGICAL MSG LENGTH *645 2 SPGTALN DS H TRANSFER ADMISSION LENGTH *646 2 SPGTATP DS X TRANSFER ADMISSION LENGTH *646 2 SPGRSD DS 34X RESERVED 645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64			DS	AL1	DPRINT DESTINATION	*643
2 SPGDSTL DS AL1 SERVER TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRJP DS AL1 REMOTE JOB PRIORITY *630 2 DS X RESERVED *645 2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDOCT DS H MAX LOGICAL MSG LENGTH *645 2 SPGTALN DS H TRANSFER ADMISSION LENGTH *646 2 SPGTATP DS X TRANSFER ADMISSION LENGTH *646 2 SPGRSD DS 34X RESERVED 645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64	2	SPGDPOL	DS	CL8	DPRINT DEFAULT POOL	*643
2 SPGDSTL DS AL1 SERVER TASK LIMIT *630 2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRJP DS AL1 REMOTE JOB PRIORITY *630 2 DS X RESERVED *645 2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDOCT DS H MAX LOGICAL MSG LENGTH *645 2 SPGTALN DS H TRANSFER ADMISSION LENGTH *646 2 SPGTATP DS X TRANSFER ADMISSION LENGTH *646 2 SPGRSD DS 34X RESERVED 645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64	2	SPGDRSL	DS	AL1	RESOURCE LOCATION	*643
2 SPGDRTL DS AL1 RESOURCE TASK LIMIT *630 2 SPGDRJP DS AL1 REMOTE JOB PRIORITY *630 2 DS X RESERVED *645 2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDOCT DS H MAX LOGICAL MSG LENGTH *645 2 SPGTALN DS H TRANSFER ADMISSION LENGTH *646 2 SPGTATP DS X TRANSFER ADMISSION SDF TYPE *646 2 SPGRSD DS 34X RESERVED 645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64			DS	AL1	SERVER TASK LIMIT	*630
2 DS X RESERVED *645 2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDOCT DS H MAX LOGICAL MSG LENGTH *645 2 SPGTALN DS H TRANSFER ADMISSION LENGTH *646 2 SPGTATP DS X TRANSFER ADMISSION SDF TYPE *646 2 SPGGRSD DS 34X RESERVED 645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64	2	SPGDRTL	DS	AL1	RESOURCE TASK LIMIT	*630
2 DS X RESERVED *645 2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDOCT DS H MAX LOGICAL MSG LENGTH *645 2 SPGTALN DS H TRANSFER ADMISSION LENGTH *646 2 SPGTATP DS X TRANSFER ADMISSION SDF TYPE *646 2 SPGGRSD DS 34X RESERVED 645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64			DS	AL1	REMOTE JOB PRIORITY	*630
2 SPGDTAD DS XL32 TRANSFER ADMISSION *630 2 SPGDOCT DS H MAX LOGICAL MSG LENGTH *645 2 SPGTALN DS H TRANSFER ADMISSION LENGTH *646 2 SPGTATP DS X TRANSFER ADMISSION SDF TYPE *646 2 SPGGRSD DS 34X RESERVED 645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64				Χ		
2 SPGDOCT DS H MAX LOGICAL MSG LENGTH *645 2 SPGTALN DS H TRANSFER ADMISSION LENGTH *646 2 SPGTATP DS X TRANSFER ADMISSION SDF TYPE *646 2 SPGGRSD DS 34X RESERVED 645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64		SPGDTAD		XL32		
2 SPGTALN DS H TRANSFER ADMISSION LENGTH *646 2 SPGTATP DS X TRANSFER ADMISSION SDF TYPE *646 2 SPGGRSD DS 34X RESERVED 645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64				Н		
2 SPGGRSD DS 34X RESERVED 645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64				Н		*646
2 SPGGRSD DS 34X RESERVED 645 2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64	2	SPGTATP	DS	Χ	TRANSFER ADMISSION SDF TYPE	*646
2 SPGLEN EQU *-SPGOUT OUTADR1 DS CL64						
OUTADR1 DS CL64						
	_					
				- - -		

4 Macros for managing print jobs

A programming interface comprising four macros is provided for managing print jobs. These macros provide the same functionality as the relevant commands.

- "CANPJOB cancel print job" on page 205 (CANCEL-PRINT-JOB command)
 This macro cancels a print job.
- "MODPJAT modify print job attributes" on page 210 (MODIFY-PRINT-JOB-ATTRIBUTES command)
 This macro modifies the attributes of a print job.
- "SHOPJAT show print job attributes" on page 283 (SHOW-PRINT-JOB-ATTRIBUTES command)
 - This macro outputs the attributes of a print job.
- "SHOPJST show print job status" on page 291 (SHOW-PRINT-JOB-STATUS command)

This macro outputs the status of a print job.

In section "Support macros for MODPJAT" on page 340 the following macros are described:

- "MODJRSO define RSO parameters for MODPJAT macro" on page 341
- "MPJADPC define Dprint parameters for MODPJAT macro" on page 348
- "MPJASPS define SPS parameters for MODPJAT macro" on page 350

In section "Output structures of the SHOPJAT macro" on page 355 the following output structures are described:

- "SJAODOF output structure for INFO=*DOCUMENT_FORMAT" on page 356
- "SJAODOP output structure for INFO=*DOCUMENT_PART" on page 358
- "SJAOLCT output structure for INFO=*LAYOUT_CONTROL" on page 361
- "SJAOPJC output structure for INFO=*PRINT_JOB_CONTROL" on page 364
- "SJAORSD output structure for INFO=*RESOURCES_DESCRIPTION" on page 365
- "SJAOTOP output structure for INFO=*TO_PRINTER" on page 367
- "SPJARSO RSO output structure for SHOPJAT macro" on page 368
- "SPJADPC Dprint output structure for SHOPJAT macro" on page 369
- "SPJASPS SPS output structure for SHOPJAT macro" on page 370

In section "Support macro for the SHOPJST macro" on page 371 the following macro is described:

"SPJSRSO - define RSO parameters for SHOPJST macro" on page 371

In section "Output structures of the SHOPJST macro" on page 374 the following output structures are described:

- "SJSODES output structure for DESTINATION" on page 375
- "SJSODES output structure for SPOOL-FILTER" on page 378
- "SJSODES output structure for RSO-FILTER" on page 381
- "SJSODIS output structure for DISTRIBUTED" on page 384
- "SJSODSS output structure for DESTINATION and UNIX systems" on page 386
- "SJSOORI output structure for ORIGIN" on page 388
- "SJSOORS output structure for ORIGIN and UNIX systems" on page 390
- "SJSOTRA output structure for TRAITS" on page 391
- "SJSOTRS output structure for TRAITS and UNIX systems" on page 393

CANPJOB - cancel print job

User group: Nonprivileged users

Programming languages: Assembler, C, COBOL

Macro type: C, D, E, L, M

This macro cancels a print job. It corresponds to the CANCEL-PRINT-JOB command.

Format

Operation	Operands
CANPJOB	VARIANT= <u>001</u> / <c-string 33=""></c-string>
	,TSN=(tsn1, cluste1) tsn1: <var: 4="" char:=""> / <c-string 44:="" alphanum-name_44=""> cluste1: *LOCAL CLUSTER / <var: 8="" char:=""> / <c-string 18:="" name_18=""></c-string></var:></c-string></var:>
	SERVTSN=(tsn2, server) tsn2: <var: 4="" char:=""> / <c-string 44:="" alphanum-name_44=""> server: <var: 8="" char:=""> / <c-string 18:="" name_18=""></c-string></var:></c-string></var:>
	MONJV=*NONE / <var: 54="" char:=""> / <c-string 154:="" filename_154_with-catid=""></c-string></var:>
	FOREIGN=(identif, cluste2): identif: *NONE / <var: 4="" int:=""> / <integer 12147483647=""> cluste2: <var: 8="" char:=""> / <c-string 18:="" name_18=""></c-string></var:></integer></var:>
	SYSFILE=array(16): *DUMMY / <var 1="" :int:=""> / *SYSOUT / *SYSLST / <integer099></integer099></var>

Description of the operands

VARIANT=<u>001</u> / <c-string 3..3>

Specifies the variant of the parameter list.

Notes

The TSN, SERVTSN, MONJV, FOREIGN and SYSFILE operands are mutually exclusive and only one of them may be used at any one time.

TSN=(tsn1, cluste1)

Identifies the print job via its references on the client computer.

tsn1: <var: char: 4> / <c-string 4..4: alphanum-name 4..4>

Task sequence number of the job to be canceled.

tsn1: <var: char: 4>

The TSN of the print job is specified indirectly via a field and not directly via the relevant operand value.

tsn1: <c-string 4..4: alphanum-name 4..4>

The TSN of the print job is specified directly.

cluste1: *LOCAL_CLUSTER / <var: char: 8> / <c-string 1..8: name_1..8>

Name of the cluster in which the print job is being processed. If a cluster name is specified, the specified TSN is the TSN on the gateway host in the specified remote cluster. Only a BS2000 cluster can be specified.

cluste1: *LOCAL CLUSTER

Default value for the name of the cluster in which the print job is to be processed.

cluste1: <var: char: 8>

The cluster name is specified indirectly via a field and not directly via the relevant operand value.

cluste1: <c-string 1..8: name_1..8>

The cluster name is specified directly.

SERVTSN=(tsn2, server)

Identifies the print job via its references on the server computer. Only print jobs in the local cluster can be addressed in this way.

tsn2: <var: char: 4> / <c-string 4..4: alphanum-name_4..4>

TSN of the job to be canceled on the server.

tsn2: <var: char: 4>

The TSN of the print job is specified indirectly via a field and not directly via the relevant operand value.

tsn2: <c-string 4..4: alphanum-name_4..4>

The TSN of the print job is specified directly.

server: <var: char: 8> / <c-string 1..8: name_1..8>

Name of the server computer on which the print job is processed.

server: <var: char: 8>

The server computer name is specified indirectly via a field and not directly via the relevant operand value.

server: <c-string 1..8: name_1..8>

The server computer name is specified directly.

MONJV=*NONE / <var: char: 54> / <c-string 1..54: filename_1..54_with-catid>

This operand can be used to address a print job by means of its MONJV, provided the specified MONJV is accessible on the host at which the command is issued.

MONJV=*NONE

No monitoring job variable was specified.

MONJV=<var: char: 54>

The MONJV name is specified indirectly via a field and not directly via the relevant operand value.

MONJV=<c-string 1..54: filename_1..54_with-catid>

The MONJV name is specified directly.

FOREIGN=(*identif*, *cluste2*):

The print job is addressed via a non-BS2000 identification.

identif: *NONE / <var: int: 4> / <integer 1..2147483647>

Identification of the non-BS2000 reference to the print job.

identif: *NONE

Default value. No identification specified.

identif: <var: int: 4>

The identification is specified indirectly via a field and not directly via the relevant operand value.

identif: <integer 1..2147483647>

The identification is specified directly.

cluste2: <var: char: 8> /<c-string 1..8: name_1..8>

Name of the cluster in which the print job is processed.

cluste2: <var: char: 8>

The cluster name is specified indirectly via a field and not directly via the relevant operand value.

cluste2: <c-string 1..8: name_1..8>

The cluster name is specified directly.

SYSFILE=array(16): *DUMMY / <var :int: 1> / *SYSOUT / *SYSLST / <integer 0..99> Specifies the system file to be processed. This operand stops the START-PROCESSING started by a prior PRINT-DOCUMENT command.

SYSFILE=*DUMMY

Operand default value.

SYSFILE=<var :int: 1>

The system file identification is specified indirectly via a field and not directly via the relevant operand value.

SYSFILE=*SYSOUT SYSFILE=*SYSLST SYSFILE=<integer 0..99>

The system file identification is specified directly.

Return codes

(SC2)	SC1	Maincode	Meaning
00	00	0000	No errors
02	00	0000	Print job not found or processing not allowed
01	01	FFFF	Invalid parameter
02	01	FFFF	Invalid address / length
03	01	FFFF	Output area too short
00	20	FFFF	System error
02	40	FFFF	Subcode 1 = 40 : no authorization
04	40	FFFF	Operand value error
05	40	FFFF	P/L version not supported
00	80	FFFF	Subsystem not loaded
01	80	FFFF	SPOOL not loaded
02	80	FFFF	RSO not loaded
03	80	FFFF	PLAM not loaded
04	80	FFFF	JV not loaded
05	80	FFFF	SPS not loaded
06	80	FFFF	DPRINTCL not loaded
07	80	FFFF	DPRINTCM not loaded
08	80	FFFF	DPRINTSV not loaded
09	80	FFFF	DSEM not loaded

Note

The CANPJOB macro passes the return code back. The return codes may also be passed back by HDRCHECK and \$VALID.

If no data area was provided for the header, the value X'0000003C is returned in register 15.

Parameter list description

		t description	
SCPJPL	DS	OF	
		=(C,SCPJ),EQUATES=NO	Standard header
* main re			
SCPJRCNO	EQU	0	<pre>subcode1 = 00 : no error</pre>
SCPJWARN	EQU	2	job not found or processing
*			not permitted
SCPJPARM	EQU	1	subcode1 = 01 : wrong
*			parameter
SCPJVALD	EQU	2	invalid address/length
SCPJSHRT	EQU	3	output area too short
SCPJSYSE	EQU	0	<pre>subcode1 = 20 : system error</pre>
SCPJPRIV	EQU	2	subcode1 = 40 : no
*			authorization
SCPJVERR	EQU	4	operand value error
SCPJVERE	EQU	5	p/l version not supported
SCPJNOSS	EQU	0	<pre>subcode1 = 80 : subsystem not</pre>
*			loaded
SCPJNSP0	EQU	1	spool not loaded
SCPJNRS0	EQU	2	rso not loaded
SCPJNPLM	EQU	3	plam not loaded
SCPJNJV	EQU	4	jv not loaded
SCPJNSPS	EQU	5	sps not loaded
SCPJNDCL	EQU	6	dprintcl not loaded
SCPJNDCM	EQU	7	dprintcm not loaded
SCPJNDSV	EQU	8	dprintsv not loaded
SCPJNDSM	EQU	9	dsem not loaded
*	5.0		
SCPJTSN	DS	CL4	client or server TSN
SCPJCLUS	DS	CL8	cluster name
SCPJSERV	DS	CL8	server name
SCPJMJV	DS	CL54	MONJV name
SCPJSYSFILE		16X	list of system files
SCPJOPT	DS	AL1	_option
SCPJOTSN	EQU	X,80,	*TSN
SCPJOSVT	EQU	X'40'	*SERVER-TSN
SCPJOMJV	EQU	X'20'	*MONJV
SCPJOFRG	EQU	X'10'	*FOREIGN
SCPJOSYS	EQU	X'08'	*SYSFILE
SCPJOFRE	EQU	X'07'	free bits
SCPJUN2	DS	XL1	
SCPJIDEN	DS	F	print job identification on
*	FOLL		non-BS2000 cluster
SCPJ#	EQU	*-SCPJHDR	

MODPJAT - modify print job attributes

User group: Nonprivileged users

Programming languages: Assembler, C, COBOL

Macro type: C, D, E, L, M

This macro modifies the attributes of a print job. The macro corresponds to the MODIFY-PRINT-JOB-ATTRIBUTES command.

Format

Operation	Operands
MODPJAT	VARIANT=001 / <c-string 33=""></c-string>
	CALLER=*USER / *SYSTEM
	RSOPAR=*NONE / <var: pointer=""></var:>
	SPSPAR=*NONE / <var: pointer=""></var:>
	DPPAR=*NONE / <var: pointer=""></var:>
	DSEMPAR=*NONE / <var: pointer=""></var:>
	TSN=(tsn1, cluste1) tsn1: <var: 4="" char:=""> / <c-string 44:="" alphanum-name_44=""> cluste1: *LOCAL CLUSTER / <var: 8="" char:=""> / <c-string 18:="" name_18=""></c-string></var:></c-string></var:>
	SRV_TSN=(tsn2, server) tsn2: <var: 4="" char:=""> / <c-string 44=""> server: <var: 8="" char:=""> / <c-string 18:="" name_18=""></c-string></var:></c-string></var:>
	CLT_TSN=(tsn2, client) tsn2: <var: 4="" char:=""> / <c-string 44:="" alphanum-name_44=""> client: <var: 8="" char:=""> / <c-string 18:="" name_18=""></c-string></var:></c-string></var:>

(part 1 of 6)

Operation	Operands
MODPJAT	MONJV=*NONE / <var: 54="" char:=""> /</var:>
	<pre><c-string 154:="" filename_154_with-catid=""></c-string></pre>
	FOREIGN=(identif, cluste2)
	identif: <u>*NONE</u> / <var: 4="" int:=""> / <integer 12147483647=""> / *WHOLE-FILE</integer></var:>
	cluste2: <var: 8="" char:=""> / <c-string 18:="" name18=""></c-string></var:>
	secttyp: *UNCHANGED / *CHAR / <var: bit:1=""> / *HEXA</var:>
	sectpos:*UNCHANGED / <var: 2="" int:=""> / <integer 12047=""> / *STD</integer></var:>
	SECTREC=(sectid, sectlen, secttyp, sectpos)
	sectid: *UNCHANGED / <var: 60="" char:=""> /</var:>
	<pre><c-string 160:="" c-string_160=""> / *WHOLE-FILE sectlen: 0 / <var: 1="" int:=""> / <integer 160=""></integer></var:></c-string></pre>
	secttyp: *UNCHANGED / *CHAR / <var: bit:1=""> / *HEXA</var:>
	sectpos:*UNCHANGED / <var: 2="" int:=""> / <integer 12047=""> / *STD</integer></var:>
	FIRSREC=(stpos, ststrid, ststrle, secttyp, ststrpo, ststroc) stpos: <var: 4="" int:=""> / <integer 12147483647=""> / *BEGIN_OF_FILE / *BY_STRING_ID</integer></var:>
	ststrid: *UNCHANGED / <var: 60="" char:=""> / <c-string 160:="" c-string_160=""> ststrle: 0 / <var: 1="" int:=""> / <integer 160=""></integer></var:></c-string></var:>
	secttyp: *UNCHANGED / *CHAR / <var: bit:1=""> / *HEXA</var:>
	ststrpo: *UNCHANGED / <var: 2="" int:=""> / <integer 12047=""> / *STD ststroc: *UNCHANGED / <var: 2="" int:=""> / <integer 132767=""></integer></var:></integer></var:>
	LASTREC=(enpos, enstrid, enstrle, secttyp, enstrpo, enstroc) enpos: <var: 4="" int:=""> / <integer 12147483647=""> / *END_OF_FILE / *BY STRING ID</integer></var:>
	enstrid: *UNCHANGED / <var: 60="" char:=""> / <c-string 160:="" c-string_160=""> enstrie: 0 / <var: 1="" int:=""> / <integer 160=""></integer></var:></c-string></var:>
	secttyp: *UNCHANGED / *CHAR / <var: bit:1=""> / *HEXA</var:>
	enstrpo: *UNCHANGED / <var: 2="" int:=""> / <integer 12047=""> / *STD enstroc: *UNCHANGED / <var: 2="" int:=""> / <integer 132767=""></integer></var:></integer></var:>
	RECPART=(firstch, lastch) firstch: *UNCHANGED / <var: 2="" int:=""> / <integer 132767=""> lastch: *UNCHANGED / <var: 2="" int:=""> / <integer 132767=""></integer></var:></integer></var:>

(part 2 of 6)

Operation	Operands
MODPJAT	OUTPART=(from, to, dim) from: *UNCHANGED / <var: 4="" int:=""> /</var:>
	DOCFORM= <u>*UNCHANGED</u> / <var: 1="" _docform_set:="" enum-of=""> /*TEXT / *PAGE_FORMAT / *SPECIAL_FORMAT</var:>
	LINESP=(spacing, ccpos) spacing: *UNCHANGED / <var: _space_set:1="" enum-of=""> / *SPACE_1 /</var:>
	CONTMOD=(mode, pcc, ctltype) mode: *UNCHANGED / <var: 1="" _control_mode_set:="" enum-of=""> / *STD /</var:>
	ADDCOP=*UNCHANGED / <var: 1="" int:=""> / <integer 0255=""></integer></var:>
	CHECKP= <u>*UNCHANGED</u> / <var: _checkp_set:1="" enum-of=""> / *ON_PAGES / *ON_SECTION_RECORDS</var:>
	PRJNAME= <u>*UNCHANGED</u> / <var: 8="" char:=""> /</var:>
	PRJPRIO=*UNCHANGED / <var: 1="" int:=""> / <integer 30255=""></integer></var:>
	TRUNC= <u>*UNCHANGED</u> / <var: _truncation_set:1="" enum-of=""> / *STD / *DELETE_FILE / *KEEP_FILE</var:>

(part 3 of 6)

Operation	Operands
MODPJAT	LOCKF=*UNCHANGED / <var: _lockf_set:1="" enum-of=""> / *YES / *NO</var:>
	DELF= <u>*UNCHANGED</u> / <var: 1="" _delf_set:="" enum-of=""> / *NO / *YES / *DESTROY</var:>
	PAGECOP=*UNCHANGED / <var: 2="" int:=""> / <integer 0255=""> / *STD</integer></var:>
	LEFTMAR=*UNCHANGED / <var: 1="" int:=""> / <integer 031=""> / *STD</integer></var:>
	LINEPP=*UNCHANGED / <var: 2="" int:=""> / <integer 132767=""> / *STD</integer></var:>
	HEADLIN= <u>*UNCHANGED</u> / <var: _headline_set:1="" enum-of=""> / *NO / *STD / *DATE / *FRECORD / *PAGE / *DATE_FRECORD / *DATE_PAGE / *FRECORD_PAGE / *DATE_FRECORD_PAGE</var:>
	OUTFORM= <u>*UNCHANGED</u> / <var: _outform_set:1="" enum-of=""> / *CHARACTER / *HEXADECIMAL</var:>
	TWOSIDE= <u>*UNCHANGED</u> / <var: _twoside_set:1="" enum-of=""> / *STD / *NO / *YES / *TUMBLE / *IGNORE</var:>
	ROT= <u>*UNCHANGED</u> / <var: 1="" _rotation_set:="" enum-of=""> / *NO / *ROT_0 / *ROT_90 / *ROT_180 / *ROT_270 / *ROT_0_180 / *ROT_90_270 / *ROT_180_0 / *ROT_270_90 / *BY_CONTROL_CODES</var:>
	INTRAY=*UNCHANGED / <var: 1="" int:=""> / <integer 199=""> / *STD / *IGNORE / *BY_FORMAT</integer></var:>
	INTRAYF= <var: _intray_format_set:1="" enum-of=""> / *MANUAL / *A3 / *A4 /</var:>
	OUTTRAY= <u>*UNCHANGED</u> / <var: 1="" int:=""> / <integer 199=""> / *STD / *IGNORE / *SORTER</integer></var:>

(part 4 of 6)

Operation	Operands
MODPJAT	COVPAGE=(headtxt, headexi, traiexi) headtxt: <var: 32="" char:=""> / <c-string 132:="" c-string_132=""> / *NONE headexi: *UNCHANGED / <var: 4="" int:=""> / <integer 02147483639=""> / *NO traiexi: *UNCHANGED / <var: 4="" int:=""> / <integer 02147483639=""> / *NO</integer></var:></integer></var:></c-string></var:>
	FORM=*UNCHANGED / <var: 6="" char:=""> / <c-string 16:="" alphanum_name_16=""> / *STD</c-string></var:>
	LOOP= <u>*UNCHANGED</u> / <var: 3="" char:=""> / <c-string 13:="" alphanum_name_13=""> / *STD</c-string></var:>
	ROTLOOP=*UNCHANGED / <var: 3="" char:=""> / <c-string 13:="" alphanum_name_13=""> / *STD</c-string></var:>
	CHARSET=*UNCHANGED / array(16): <var: 3="" char:=""> / <c-string 13:="" alphanum_name_13=""> / *STD</c-string></var:>
	POOLNAM=*UNCHANGED / <var: 4="" char:=""> / <c-string 14:="" name_14=""> / *NONE</c-string></var:>
	POOLIND=*UNCHANGED / <var: 1="" int:=""> / <integer 064=""></integer></var:>
	EFO=*UNCHANGED/ <var: 2="" char:="">/<c-string 22:="" c-string_22="">/*NONE</c-string></var:>
	OVERLAY=(face, reverse) face: <var: 1="" int:=""> / <integer 1127=""> / *STD / *NONE reverse : <var: 1="" int:=""> / <integer 1127=""> / *STD / *NONE</integer></var:></integer></var:>
	FOB=*UNCHANGED / <var: 4="" char:=""> /</var:>
	PAGEPCL=*UNCHANGED / <var: 4="" int:=""> / <integer 150000=""> / *STD</integer></var:>
	USERRES= <u>*UNCHANGED</u> / <var: 44="" char:=""> /</var:>

(part 5 of 6)

Operation	Operands
MODPJAT	TRANTAB=(name, file) name: <var: 8="" char:=""> /</var:>
	PRNAME= *UNCHANGED / <var: 18="" char=""> / <c-string: 18="" c-string=""> / *STD / *IPP</c-string:></var:>
	PRTYPE=*UNCHANGED / <var: _prtype_set:1="" enum-of=""> / *ANY / *LP65_PRINTER / *HP_PRINTER / *APA_PRINTER</var:>
	SRTMODE= <u>*UNCHANGED</u> / <var: _srtmode_set:1="" enum-of=""> / *NO / *GROUP / *COLLATE / *STACKER / *AUTOMATIC</var:>
	VIRTUAL= <u>*UNCHANGED</u> / <var: _virtual_set:1="" enum-of=""> / *ALLOWED / *NOT_ALLOWED / *STD / *MUST</var:>
	PRJCLAS=*UNCHANGED / <var: 1="" int:=""> / <integer 1255=""></integer></var:>
	PROGRAM=(name, string) name: *UNCHANGED / <var: 8="" char:=""> /</var:>
	FRMNAME=*UNCHANGED / *STD / <c-string: 163="" c-string=""> / <var: 63="" char:=""> / (<reg: char:63="">)</reg:></var:></c-string:>
	TOPOFF=*UNCHANGED / *IGNORE / <integer -255255=""></integer>
	,LEFTOFF=*UNCHANGED / *IGNORE / <integer -255255=""></integer>
	,SCHEDTIME = (schdate, schtime) schdate: *UNCHANGED / *TODAY/ <var: 110="" char=""> / <c-string 1010=""> schtime: *UNCHANGED / *NOW <var: 15="" char=""> / <c-string 55=""></c-string></var:></c-string></var:>

(part 6 of 6)

Description of the operands

VARIANT=001 / <c-string 3..3>

Specifies the variant of the parameter list.

CALLER=*USER / *SYSTEM

Caller of the macro.

CALLER=*USER

Operand default value. A user calls the macro.

CALLER=*SYSTEM

The macro is called by the system.

RSOPAR=*NONE / <var: pointer>

Address of the RSO parameter list.

RSOPAR=*NONE

Operand default value. No RSO parameter list exists.

RSOPAR=<var: pointer>

Pointer to the RSO parameter list.

SPSPAR=*NONE / <var: pointer>

Address of the SPS parameter list.

SPSPAR=*NONE

Operand default value. No SPS parameter list exists.

SPSPAR=<var: pointer>

Pointer to the SPS parameter list.

DPPAR=*NONE / <var: pointer>

Address of the Dprint parameter list.

DPPAR=*NONE

Operand default value. No Dprint parameter list exists.

DPPAR=<var: pointer>

Pointer to the Dprint parameter list.

DSEMPAR=*NONE / <var: pointer>

Address of the DSEM parameter list.

DSEMPAR=*NONE

Operand default value. No DSEM parameter list exists.

DSEMPAR=<var: pointer>

Pointer to the DSEM parameter list.

TSN=(tsn1, cluste1)

Identifies the print job via its references on the client computer.

tsn1: <var: char: 4> / <c-string 4..4: alphanum-name_4..4>

TSN of the job to be modified.

tsn1: <var: char: 4>

The TSN of the print job is specified indirectly via a field and not directly via the relevant operand value.

tsn1: <c-string 4..4: alphanum-name 4..4>

The TSN of the print job is specified directly.

cluste1: *LOCAL_CLUSTER / <var: char: 8> / <c-string 1..8: name_1..8>

Name of the cluster in which the print job is processed. If you specify a cluster name, the specified TSN is the TSN on the gateway host in the specified remote cluster. Only BS2000 clusters can be specified.

cluste1: *LOCAL CLUSTER

Default value for the name of the cluster in which the print job is to be processed.

cluste1: <var: char: 8>

The cluster name is specified indirectly via a field and not directly via the relevant operand value.

cluste1: <c-string 1..8: name_1..8>

The cluster name is specified directly.

SRV_TSN=(tsn2, server)

Identifies the print job via its references on the server computer.

tsn2: <var: char: 4> / <c-string 4..4: alphanum-name_4..4>

Task sequence number of the job to be modified on the server.

tsn2: <var: char: 4>

The TSN of the print job is specified indirectly via a field and not directly via the relevant operand value.

tsn2: <c-string 4..4: alphanum-name_4..4>

The TSN of the print job is specified directly.

server: <var: char: 8> / <c-string 1..8: name_1..8>

Name of the server on which the print job is processed.

server: <var: char: 8>

The server computer name is specified indirectly via a field and not directly via the relevant operand value.

server: <c-string 1..8: name_1..8>

The server computer name is specified indirectly.

CLT_TSN=(tsn2, server)

Identifies the print job via its references on the client computer. Only print jobs in the local cluster can be addressed in this way.

This option is reserved for the cluster administrator.

tsn2: <var: char: 4> / <c-string 4..4: alphanum-name_4..4>

Task sequence number of the job to be modified on the client.

tsn2: <var: char: 4>

The TSN of the print job is specified indirectly via a field and not directly via the relevant operand value.

tsn2:<c-string 4..4: alphanum-name_4..4>

The TSN of the print job is specified directly.

client: <var: char: 8> / <c-string 1..8: name_1..8>
Name of the client on which the print job is processed.

client: <var: char: 8>

The client computer name is specified indirectly via a field and not directly via the relevant operand value.

client: <c-string 1..8: name_1..8>

The client computer name is specified directly.

MONJV=*NONE / <var: char: 54> / <c-string1..54: filename_1..54_with-catid>

This operand allows a print job to be addressed by means of its MONJV, provided the specified MONJV is accessible on the host at which the command is issued.

MONJV=*NONE

Default value for the MONJV operand. No monitoring job variable was specified.

MONJV=<var: char: 54>

The MONJV name is specified indirectly via a field and not directly via the relevant operand value.

MONJV=<c-string1..54: filename_1..54_with-catid>

The MONJV name is specified directly.

FOREIGN=(*identif*, *cluste2*):

The print job is addressed via a non-BS2000 identification.

identif: *NONE / <var: int: 4> / <integer 1..2147483647>
Identification of the non-BS2000 reference to the print job.

identif: *NONE

Default value, no identification specified.

identif: <var: int: 4>

The identification is specified indirectly via a field and not directly via the relevant operand value.

identif: <integer 1..2147483647>

The identification is specified directly.

cluste2: <var: char: 8> /<c-string 1..8: name_1..8>

Name of the cluster in which the print job is to be processed.

cluste2: <var: char: 8>

The cluster name is specified indirectly via a field and not directly via the relevant operand value.

cluste2: <c-string 1..8: name_1..8>

The cluster name is specified directly.

SECTREC=(sectid, sectlen, secttyp, sectpos)

Specifies whether the file is to be structured using section marks. The strings in the data records which are to be used as section marks can be specified with "sectid". The length of the specified string can be specified with "sectlen" and its position with "sectpos".

sectid: *UNCHANGED / <var: char: 60> / <c-string 1..60: c-string_1..60> / *WHOLE-FILE

It is possible to specify whether the file is to be structured using section marks.

The FIRSBEC operand can be used to define which part of the sectioned print f

The FIRSREC operand can be used to define which part of the sectioned print file is to be output.

Any strings in the records can be used as file marks. These strings can be either in SECTION records, which are not printed, or in the print file's normal records, which are printed. A string can be specified in the form of printable characters or hexadecimal characters.

The search for the start of the section to be printed is executed in a separate "pseudo controller" task. Neither the user task nor the printer is locked while this is being done. A pseudo controller writes the address of the first record in the section to the SPOOL

control block; the spoolout job can then be processed (PREPROCESSINPUT-PARTING). If the desired section is not found in the file, an error message appears on the trailer page (for the layout of the trailer page, see page 591).

sectid: *UNCHANGED

Operand default value. The current setting is not changed.

sectid: <var: char: 60>

Name of a field defined with CL. This field contains a string with a length of 60 bytes that is interpreted as a string.

sectid: <c-string 1..60: c-string_1..60>

Specifies the strings in the data records, enclosed in quotes.

sectid: *WHOLE-FILE

The file is not to be structured using section marks.

sectlen: <u>0</u> / <var: int: 1> / <integer 1..60>

Specifies the string length.

sectlen: 0

Default: the string is 0 bytes long.

sectlen: <var: int: 1>

Name of a field defined with FL. This field contains an integer with a length of 1 byte that is interpreted as the string length.

sectlen: <integer 1..60>

The string length is specified as an integer of from 1 to 60.

secttyp: *UNCHANGED / *CHAR / <var: bit:1> / *HEXA

Type of section identification.

secttyp: *UNCHANGED

Default: the section identification is not changed.

secttyp: *CHAR

The section identification consists of characters.

secttyp: <var: bit:1>

The section identification type is specified in a field.

secttyp: *HEXA

The section identification is specified in hexadecimal notation.

sectpos: *UNCHANGED / <var: int: 2> / <integer 1..2047> / *STD

Specifies the position (from which byte after the record length field) as of which the specified string begins in the SECTION record.

sectpos: *UNCHANGED

Operand default value. The current setting is not changed.

sectpos: <var: int: 2>

Name of a field defined with FL. This field contains an integer with a length of two bytes which is interpreted as the position.

sectpos: <integer 1..2047>

Integer value for the position as of which the specified string begins in the SECTION record.

sectpos: *STD

The relevant string begins by default at the start of the record, i.e.:

- in a SAM file: at the first byte after the record length field
- in an ISAM file with KEY-POS=5: at the first byte after the key
- in an ISAM file with KEY-POS > 5: at the first byte after the record length field.

FIRSREC=(stpos, ststrid, ststrle, secttyp, ststrpo, ststroc)

Specifies whether only a particular number of data records are to be processed from the input file. The start and end data record can be specified via a record number, a section number or any string. The entries for the end data record can be set with the LASTREC operand.

"stpos" can be used to specify a start data record as of which the file is to be processed.
"ststrtid" can be used to select a data record in which the specified string occurs as the start data record.

The data record length can be specified with "ststrle".

The string type can be specified with "secttyp".

"ststrpo" is used to specify the position within the data record as of which the specified string begins.

"ststroc" can be used to specify the number of occurrences of the record containing the string as of which the output is to start.

stpos: <var: int: 4> / <integer 1..2147483647> / *BEGIN_OF_FILE / *BY_STRING_ID Specifies the record as of which the file is to be processed. You can specify the first record in the file, the number of a record or file mark, or a string in a record.

stpos: <var: int: 4>

The start data record is specified in a field as an integer of from 1 to 2147483647.

stpos: <integer 1..2147483647>

The start data record is specified as an integer of from 1 to 2147483647.

stpos: *BEGIN_OF_FILE

Output begins with the file's first record, even if SECTION records are specified.

stpos: *BY_STRING_ID

A data record containing a string, which must be specified subsequently, is selected as the start data record.

ststrid: *UNCHANGED / <var: char: 60> / <c-string 1..60: c-string_1..60>

Specifies whether output is to start with a data record in which a particular string of printable or hexadecimal characters is found at a specific position in the file.

ststrid: *UNCHANGED

Operand default value. The current setting is not changed.

ststrid: <var: char: 60>

Name of a field defined with CL. This field contains a string with a length of 60 bytes which is interpreted as a string.

ststrid: <c-string 1..60: c-string_1..60>

Output is to start with the data record containing the specified string of printable or hexadecimal characters which is found at a specific position in the file.

ststrle: <u>0</u> / <var: int: 1> / <integer 1..60>

Specifies the data record length.

ststrle: 0

Default: the data record has a length of 0 bytes.

ststrle: <var: int: 1>

Name of a field defined with FL. This field contains an integer with a length of 1 byte which is interpreted as the length of the data record.

ststrle: <integer 1..60>

The data record length is specified as an integer of from 1 to 60.

secttyp: *UNCHANGED / *CHAR / <var: bit:1> / *HEXA

Section identification type.

secttyp: *UNCHANGED

Default: the section identification is not changed.

secttyp: *CHAR

The section identification consists of characters.

secttyp: <var: bit:1>

The section identification type is specified in a field.

secttyp: *HEXA

The section identification is specified in hexadecimal notation.

ststrpo: *UNCHANGED / <var: int: 2> / <integer 1..2047> / *STD

Defines as of which position (as of which byte after the record length field) the specified string in the SECTION record starts.

ststrpo: *UNCHANGED

Operand default value. The current setting is not changed.

ststrpo: <var: int: 2>

Name of a field defined with FL. This field contains an integer with a length of 2 bytes which is interpreted as the position of the data record.

ststrpo: <integer 1..2047>

Specifies an integer value for the position of the string in the SECTION record.

ststrpo: *STD

The relevant string begins by default at the start of the record, i.e.:

- in a SAM file: at the first byte after the record length field
- in an ISAM file with KEY-POS=5: at the first byte after the key
- in an ISAM file with KEY-POS > 5: at the first byte after the record length field.

ststroc: *UNCHANGED / <var: int: 2> / <integer 1..32767>

Specifies the record (containing the FIRST-RECORD string) as of which output is to start.

ststroc: *UNCHANGED

Operand default value. The current setting is not changed.

ststroc: <var: int: 2>

Name of a field defined with FL. This field contains an integer with a length of 2 bytes which is interpreted as the number of occurrences.

ststroc: <integer 1..32767>

Integer value for the occurrences of the data record.

LASTREC=(enpos, enstrid, enstrle, secttyp, enstrpo, enstroc)

Defines the last data record up to which the file is to be processed. The last data record can be defined via a record number, a section number or any string.

"enpos" can be used to specify a last data record up to which the file is to be processed.

"enstrid" can be used to select a data record in which the specified string occurs as the last data record.

The data record length can be specified with "enstrle".

The string type can be specified with "secttyp".

"enstrpo" is used to specify the position within the data record as of which the specified string starts.

"enstroc" can be used to specify the number of occurrences of the data record containing the string as of which the output is to finish.

enpos: <var: int: 4> / <integer 1..2147483647> / *END_OF_FILE / *BY_STRING_ID Specifies the last data record up to which the file is to be processed. The last data record in the file, the number of a data record, a section mark or any string in a data record can be selected for this.

enpos: <var: int: 4>

Name of a field defined with FL. This field contains an integer with a length of 4 bytes which is interpreted as the number of the last data record.

enpos: <integer 1..2147483647>

The last data record is specified as an integer of from 1 to 2147483647.

enpos: *END_OF_FILE

Output ends with the last data record in the file, even if SECTION records are specified.

enpos: *BY_STRING_ID

A data record containing a string, which must be specified subsequently, is selected as the last data record.

enstrid: *UNCHANGED / <var: char: 60> / <c-string 1..60: c-string_1..60>

Output ends with the record in which the specified string of printable or hexadecimal characters is found at a specific position in the file.

enstrid: *UNCHANGED

Operand default value. The current setting is not changed.

enstrid: <var: char: 60>

Name of a field defined with CL. This field contains a string with a length of 60 bytes which is interpreted as a string.

enstrid: <c-string 1..60: c-string_1..60>

Output is to end with the data record in which the specified string of printable or hexadecimal characters is found at a specific position in the file.

enstrle: 0 / <var: int: 1> / <integer 1..60>

Specifies the data record length.

enstrle: 0

Default: The data record length is 0 bytes.

enstrle: <var: int: 1>

Name of a field defined with FL. This field contains an integer with a length of 1 byte which is interpreted as the data record length.

enstrle: <integer 1..60>

The data record length is specified as an integer of from 1 to 60.

secttyp: *UNCHANGED / *CHAR / <var: bit:1> / *HEXA

Section identification type.

secttyp: *UNCHANGED

Default: the section identification is not changed.

secttyp: *CHAR

The section identification consists of characters.

secttyp: <var: bit:1>

The section identification type is specified in a field.

secttyp: *HEXA

The section identification is specified in hexadecimal notation.

enstrpo: *UNCHANGED / <var: int: 2> / <integer 1..2047> / *STD

Defines as of which position (as of which byte after the record length field) the specified string in the SECTION record starts.

enstrpo: *UNCHANGED

Operand default value. The current setting is not changed.

enstrpo: <var: int: 2>

Name of a field defined with FL. This field contains an integer with a length of 2 bytes which is interpreted as the position of the data record.

enstrpo: <integer 1..2047>

Specifies an integer value for the position of the string in the SECTION record.

enstrpo: *STD

The relevant string begins by default at the start of the record, i.e.:

- in a SAM file: at the first byte after the record length field
- in an ISAM file with KEY-POS=5: at the first byte after the key
- in an ISAM file with KEY-POS > 5: at the first byte after the record length field.

enstroc: *UNCHANGED / <var: int: 2> / <integer 1..32767>

Specifies the occurrence of the record containing the LAST-RECORD string at which output is to stop.

enstroc: *UNCHANGED

Operand default value. The current setting is not changed.

enstroc: <var: int: 2>

Name of a field defined with FL. This field contains an integer with a length of 2 bytes which is interpreted as the number of occurrences.

enstroc: <integer 1..32767>

Integer value for the number of times the data record occurs.

RECPART=(firstch, lastch)

Specifies whether only a specific part of each of the records selected by means of the above operands is to be processed.

firstch: *UNCHANGED / <var: int: 2> / <integer 1..32767>

Allows a byte number (record column) to be specified indicating the point as of which the records of a file are to be output. (The bytes of a record are numbered consecutively from left to right starting with 1; ISAM keys and control characters are components of a record).

What happens depends on whether or not one of the values *BY_EBCDIC_CONTROL, *BY_IBM_CONTROL or *BY_ASA_CONTROL is specified in the LINESP operand.

If one of these values is specified:

Output starts with the data byte following the specified byte number. The feed control character is interpreted irrespective of the entry for *first*, provided the value for *first* is less than the length of the record. If the specified value is greater than the length of the record, it is ignored (i.e. printing does not take place, nor does line feed).

If none of these values is specified:

Output starts with the data byte corresponding to the specified byte number.

firstch: *UNCHANGED

Operand default value. The current setting is not changed.

firstch: <var: int: 2>

Name of a field defined with FL or a register containing the value. This field or register contains an integer with a length of 2 bytes which is interpreted as the record column.

firstch: <integer 1..32767>

Integer value for the byte number (record column) as of which the data records in a file are to be output.

lastch: *UNCHANGED / <var: int: 2> / <integer 1..32767>

Specifies the byte indicating the point at which printing of each record is to stop.

What happens depends on whether or not one of the values *BY_EBCDIC_CONTROL,

*BY_IBM_CONTROL or *BY_ASA_CONTROL is specified in the LINESP operand.

If one of these values is specified:

Output ends with the data byte following the specified byte number (unless FIRST-CHARACTER is specified for an ISAM file with KEY-POSITION=5, in which case output ends with the data byte corresponding to the specified byte number).

If none of these values is specified:

Output ends with the data byte corresponding to the specified byte number.

If the records are longer than the form definition allows, printing continues on the next line.

lastch: *UNCHANGED

Operand default value. The current setting is not changed.

lastch: <var: int: 2>

Name of a field defined with FL. This field contains an integer with a length of 2 bytes which is interpreted as the last byte.

lastch: <integer 1..32767>

Integer value for the last byte which is to be printed from each data record.

OUTPART=(from, to, dim)

Specifies that the whole input file is to be edited for printing but that output is to be limited to a subset of all logical print pages.

from: *UNCHANGED / <var: int: 4> / <integer -2147483647..2147483647> / *BEGIN OF FILE

Allows a page or line number in the print file to be specified as of which output is to start, as specified by the *dim* operand.

from: *UNCHANGED

Operand default value. The current setting is not changed.

from: <var: int: 4>

Name of a field defined with FL. This field contains an integer with a length of 4 bytes which is interpreted as the page or line number.

from: <integer -2147483647..2147483647>

Integer value for a page or line number of the print file. If the value is < 0, the number of pages or lines to be output is calculated from the end of the file. The DIMENSION operand is used to define whether the entry is interpreted as a page or line number.

from: *BEGIN_OF_FILE

Output starts at the beginning of the file. The pages of a print file are defined as described for the LINEPP operand (as long as no line feed occurs due to prior control characters).

to: *UNCHANGED / <var: int: 4> / <integer 1..2147483647> / *END_OF_FILE

Allows a page or line number in the print file to be specified at which output is to end, as specified by the *dim* operand.

The value specified here must be greater than that specified in the *from* operand.

to: *UNCHANGED

Operand default value. The current setting is not changed.

to: <var: int: 4>

Name of a field defined with FL. This field contains an integer with a length of with a length of 4 bytes which is interpreted as a page or line number.

to: <integer 1..2147483647>

Integer value for a page or line number in the print file.

to: *END OF FILE

Output ends with the end of the file. The pages of a print file are defined as described for the LINEPP operand (as long as no line feed occurs due to prior control characters).

dim: *UNCHANGED / <var: enum-of _layout_part_dim_set: 1> / *PAGES / *LINES Specifies whether the values in the *from* and *to* operands are to be interpreted as page or line numbers.

dim: *UNCHANGED

Operand default value. The current setting is not changed.

dim: <var: enum-of _layout_part_dim_set: 1>

The dimension is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field.

dim: *PAGES

The values specified for the *from* and *to* operands are to be interpreted as page numbers.

dim: *LINES

The values specified for the from and to operands are to be interpreted as line numbers.

DOCFORM=<u>*UNCHANGED</u> / <var: enum-of _docform_set: 1> / *TEXT / *PAGE_FORMAT / *SPECIAL_FORMAT

Specifies the type of the document contents, i.e. the format of the file to be printed as regards the interpretation of feed control characters, printer control characters, font identifiers, RENO commands and/or VTSU codes.

DOCFORM=*UNCHANGED

Operand default value. The current setting is not changed.

DOCFORM=<var: enum-of _docform_set: 1>

The type of document contents is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*TEXT
1	*PAGE_FORMAT
2	*SPECIAL_FORMAT
See the parameter list description as of page 272.	

DOCFORM=*TEXT

Except for feed control characters, the file has no printer-specific control characters. The position of the feed control character in the record can be specified in the substructure of the LINE-SPACING operand.

The data is sent to the printer unchanged. Since only the data to be printed is sent to the printer, records can be truncated.

The following applies to HP and HP90 printers:

When the operand TO-PRINTER=*PAR(PRINTER-TYPE=*ANY) is specified, the character X'FF' is replaced by the character X'1F' in order to be compatible with the PRM statement CONVERT-PRINT-RESOURCES.

The following applies to LP65 printers:

Spoolout jobs for which DOCFORM=*TEXT is specified can contain any LP65 control characters and printer control characters. Only records longer than 8192 characters are truncated.

The following applies to RSO:

Records of type A-1 or A-2 are to be printed (no control characters in the data stream); in other words, except for feed control characters in the first column of the records, no control characters are interpreted. This is also the reason why a font change is not possible. The form's default font or the first font specified for CHARSET, if specified, is used. Non-printable characters, i.e. characters with a hexadecimal value less than X'40', are output as blanks. If the record length exceeds the maximum line length, the record is truncated in the printout. The maximum line length depends on the character spacing, which is defined by means of the font used (see the CHARSET operand).

DOCFORM=*PAGE FORMAT

The file contains control characters specific to laser printers. The CONTMOD operand can be used to define how the control characters are to be interpreted.

DOCFORM=*SPECIAL_FORMAT

A printer-specific language is used in the document. In this case, the SPOOL and Dprint subsystems provide transparent control for the document, i.e. the file is transferred to an RSO or Xprint printer without evaluation.

LINESP=(spacing, ccpos)

Specifies the number of line feeds or the type of control character interpretation.

spacing: *UNCHANGED / <var: enum-of _space_set:1> / *SPACE_1 /

*SPACE_2 / *SPACE_3 / *BY_ASA_CONTROL /

*BY_EBCDIC_CONTROL / *BY_IBM_CONTROL / *NO

The following can be defined according to the operand value:

The records are to be printed with 1-, 2- or 3-line spacing (*SPACE_n).

The contents of the first byte of each record are to be interpreted as ASA feed control characters (*BY ASA CONTROL).

The contents of the first byte of each record are to be interpreted as EBCDIC feed control characters (*BY EBCDIC CONTROL).

The contents of the first byte of each record are to be interpreted as IBM feed control characters (*BY IBM CONTROL).

The contents of the first byte of each record are not to be interpreted as line feed control characters (*NO)..

The standard value (*STD) depends on the control character evaluation mode (CONTMOD operand).

spacing: *UNCHANGED

Operand default value. The current setting is not changed.

spacing: <var: enum-of _space_set:1>

The number of line feeds is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*STD
1	*SPACE_1
2	*SPACE_2
4	*SPACE_3
8	*BY_EBCDIC_CONTROL
16	*BY_ASA_CONTROL
32	*BY_IBM_CONTROL
144	*NO
See the parameter list description as of page 272.	

ccpos: *UNCHANGED / <var: int: 2> / <integer 1..2040> / *STD

Number of the data byte in which SPOOL finds the feed control character. In the case of records of variable length, the fields containing the length are not counted as part of the data.

ccpos: *UNCHANGED

Operand default value. The current setting is not changed.

ccpos: <var: int: 2>

Name of a field defined with FL. This field contains an integer with a length of 2 bytes which is interpreted as the number of the data byte.

ccpos: <integer 1..2040>

Integer value for the number of the data byte.

ccpos: *STD

The standard value (*STD) depends on the control character evaluation mode (CONTMOD operand).

CONTMOD=(*mode*, *pcc*, *ctltype*)

Defines how the control characters are to be interpreted (mode), whether the control character list must be present at the start of each page (pcc) and whether the control characters are suitable for use on type HP or HP90 printers (type).

mode: *UNCHANGED / <var: enum-of _control_mode_set: 1> / *STD /
*PHYSICAL / *LINE_MODE / *LOGICAL / *APA / *PAGE_MODE

Defines how the control characters are to be interpreted..

mode: *UNCHANGED

Operand default value. The current setting is not changed.

mode: <var: enum-of _control_mode_set: 1>

How the control characters are to be interpreted is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

1	*STD
2	*PHYSICAL
4	*PAGE_MODE
8	*LOGICAL
16	*LINE_MODE
64	*APA
See the parameter list description as of page 272.	

mode: *STD

The operand is to be ignored.

mode: *PHYSICAL Only for RSO.

LINE-SPACING=*NO is set automatically, which means that you must implement page and line feeds with LINE-MODE control characters (i.e. VTSU codes, printer control characters and RENO commands) in the file itself.

If you specify LINE-SPACING=*BY-EBCDIC-CONTROL for records of type D-2, the feed control character in the first byte is interpreted as a line or page feed control character. Non-printable characters are also transferred to the printer (in contrast to CONTMOD=*LINE_MODE).

You are responsible for the correctness of the control characters in the file (including feed control).

mode: *LINE MODE

Only for RSO.

Records of type C (see appendix) are printed out. The records may contain data mixed with LINE-MODE control characters. LINE-MODE control characters are (in any combination):

- printer control characters (i.e. physical control character beginning with X'27' or X'3C')
- RENO commands
- VTSU codes

Control over record and file formats (including page and line feed with LINE-MODE control characters) rests solely with the user. With the 9025/9026 RENO page printer, users must also ensure that the printer is set to the correct start position on the paper. A loop is not interpreted. Non-printable characters, i.e. characters with a hexadecimal value < X'40' are output as blanks.

The spoolout job is executed with the default font of the form used until you change the font in the file with the aid of LINE-MODE control characters.

mode: *LOGICAL

Only for RSO printers.

Specifies that records of type B-1 or B-2 are to be printed out, i.e. records which, in addition to a feed control character in the first byte, may contain data mixed with font identifiers, printer control characters, RENO commands and VTSU codes (see the "RSO (BS2000/OSD)" manual, section "Record type B-1/B-2"). With the exception of the VTSU codes VPA, NP, VT, NL and CR, which are output as blanks, the abovementioned control characters are interpreted. A character set identifier, a VTSU code or a RENO command remains valid until a new control character is specified. If the font identifier is omitted, the default font for the form is used.

Since page feed is implemented via a loop or constant line feed when CONTROL-MODE=LOGICAL is specified, the file should not contain the RENO commands \LF, \FF and \CR. Setting the form height is likewise not permitted.

mode: *APA

Specifies that the file to be printed contains APA control characters which have to be evaluated.

mode: *PAGE MODE

The control characters are suitable for processing on page printers.

pcc: *UNCHANGED / <var enum-of _pcc_set:1> / *YES / *NO

Specifies whether the control character list must be at the beginning of each page (i.e. always after branching to the vertical tab "channel 1" in the loop). The operand is only evaluated with output to HP and HP90 printers (setting CONTMODE=*PAGE MODE).

pcc: *UNCHANGED

Operand default value. The current setting is not changed.

pcc: <var enum-of _pcc_set:1>

The control character list is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*YES
1	*NO
2	*DUMMY
See the parameter list description as of page 272.	

pcc: *YES

The control character list must be there.

pcc: *NO

No control character list at the beginning of the page. However, this means that the following functions cannot be controlled in the case of output to HP printers:

- Film overlays on individual pages in the file; specified in the MODPJAT macro, a film overlay is used on each page of the print file of the spoolout job.
- Page copies of individual pages of the file; all pages of the print file are output with as many copies as were specified in the in the MODPJAT macro.
- Column-oriented indentation on individual pages; the value specified in the MODPJAT macro is valid for all the pages in the print file.
- FOB data overlay on individual pages in the file; specified in the MODPJAT macro, an FOB data overlay is used on each page in the print file of the spoolout job.
- A copy reference number cannot be specified.
- Page rotation control for individual pages in the print file; all the pages are output in either portrait or landscape format, as specified in the MODPJAT macro.

ctltype: *UNCHANGED / <var: enum-of _control_type_set:1> / *COMPATIBLE / *HP
Specifies whether the control characters are suitable for processing on HP or HP90 printers or whether they have to be converted.

ctltype: *UNCHANGED

Operand default value. The current setting is not changed.

ctltype: <var: enum-of _control_type_set:1>

The control character translation is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*DUMMY
1	*COMPATIBLE
2	*HP
See the parameter list description as of page 272.	

ctltype: *COMPATIBLE

The file does not contain HP or HP90 printer-specific control characters. SPOOL must convert the control characters.

ctltype: *HP

The file contains HP or HP90 printer-specific control characters that can only be processed by these printers.

ADDCOP=*UNCHANGED / <var: int: 1> / <integer 0..255>

Specifies how many additional times the file is to be printed. Each additional copy contains its own header page. The value of this operand can be specified as an integer from 0 to 255, the name of a field defined with FL with a length of 1 byte or a register containing the value. Default: 0 (no additional printouts).

ADDCOP=*UNCHANGED

Operand default value. The current setting is not changed.

ADDCOP=<var: int: 1>

Name of a field defined with FL. This field contains an integer with a length of 1 byte which is interpreted as the number of additional copies.

ADDCOP=<integer 0..255>

Number of additional copies.

CHECKP=*UNCHANGED / <var: enum-of _checkp_set:1> /

*ON_PAGES / *ON_SECTION_RECORDS

Specifies whether checkpoint processing is to be performed by the controller on the basis of pages or SECTIONs.

CHECKP=*UNCHANGED

Operand default value. The current setting is not changed.

CHECKP=<var: enum-of _checkp_set:1>

The restart is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*ON_PAGES
1	*ON_SECTION_RECORDS
See the parameter list description as of page 272.	

CHECKP=*ON PAGES

Default restart mechanism.

When an interrupted job is restarted, processing is resumed from a point a given number of pages back.

CHECKP=*ON SECTION RECORDS

The operand value can be specified for all printer types but offers advantages particularly in the case of output to HP90 printers using the TWO-UP procedure and LP65 printers. With these types of printer, a physical page can comprise a number of logical pages without SPOOL detecting it (the information is in the PCL file); in other words, the default restart mechanism, which is geared to logical pages, is highly prone to errors with this type of printer.

SECTION records are used here as restart markers. You divide your files into sections with the aid of SECTION records. These SECTION records must contain the printer commands needed to ensure correct data processing. If a physical page contains a number of logical pages, the start of a physical page must also be clearly indicated in the SECTION record. If an error occurs, processing is resumed with HOLD-PRINT-JOB and RESUME-PRINT-JOB a given number of sections further on in the file; i.e. RESTART-POSITION = PAGE(...) or BACK(...) refers to sections rather than pages. Similarly, the values shown in error messages are not pages but sections.

To arrive at a correct result, a section must correspond to at least one physical page (ideally to precisely one page). If the operand value CONTMOD=*PAGE_MODE is specified together with CHECKP=*ON_SECTION_RECORDS, in the case of output to a laser printer, you must ensure that the SECTION records are located immediately ahead of records which contain the control character line at the start of the print page. As soon as repositioning takes place (PRNTDOC or restart of an interrupted job), the first record that SPOOL prints must contain the control character line.

PRJNAME=*UNCHANGED / <var: char: 8> / <c-string 1..8: c-string_with-lower_case_1..8>

Job name for the spoolout job.

The job name can be formed from a maximum of 8 characters from the set (A,...Z,0,...9,@, #,\$,..,-) but must not start with a hyphen or end with a period. It may only start with a period if this is followed by an alpha character; in this case, the period is not printed as part of the

job name on the header page. The special character string period and hyphen '.-' may only be specified in quotes.

The job name is printed on the header page in the third uppercase line and also appears in the output of the SHOW-PRINT-JOB-STATUS command. If this operand is omitted, the job name from the SET-LOGON-PARAMETERS command is used instead.

PRJNAME=*UNCHANGED

Operand default value. The current setting is not changed.

PRJNAME=<var: char: 8>

Name of a field defined with CL. This field contains a string with a length of 8 bytes which ia interpreted as the job name.

PRJNAME=<c-string 1..8: c-string_with-lower_case_1..8>

Job name which is to be assigned to the spoolout job (and which overwrites a job name assigned to the job issuing the command).

PRJPRIO=*UNCHANGED / <var: int: 1> / <integer 30..255>

Defines the urgency with which this spoolout job is started relative to other spoolout jobs.

PRJPRIO=*UNCHANGED

Operand default value. The current setting is not changed.

PRJPRIO=<var: int: 1>

Name of a field defined with FL. This field contains an integer with a length of 1 byte which is interpreted as the priority.

PRJPRIO=<integer 30..255>

Integer value for the priority to be assigned to the spoolout job.

The highest priority you can assign here is defined in the user catalog and can be displayed with the SHOW-USER-ATTRIBUTES command. If an invalid priority is entered (or no priority at all), the spoolout job is given the same priority as the job issuing the command.

TRUNC=*UNCHANGED / <var: enum-of _truncation_set:1> / *STD / *DELETE_FILE / *KEEP_FILE

Specifies what happens if lines are truncated.

TRUNC=*UNCHANGED

Operand default value. The current setting is not changed.

TRUNC=<var: enum-of _truncation_set:1>

The spoolout job continuation is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*STD
1	*DELETE_FILE
2	*KEEP_FILE
See the parameter list description as of page 272.	

TRUNC=*STD

The default value from the SPOOL parameter is valid. You can display this value with the SHOW-SPOOL-PARAMETERS command or SPSERVE statement (field: ERROR-PR=(TRUNC=)).

TRUNC=*DELETE FILE

Processing of the spoolout job continues (i.e. DELF=*YES is executed).

An appropriate message is printed on the trailer page.

TRUNC=*KEEP FILE

Processing of the spoolout job is continued, but the file is not subsequently deleted.

LOCKF=*UNCHANGED / <var: enum-of lockf set:1> / *YES / *NO

Specifies whether the file is to be protected as long as the spoolout job is in the wait state (TYPE 4, see output of the SHOW-PRINT-JOB-STATUS command). During this time the file can only be read. As a rule, tape files are never locked. During processing of the spoolout job (TYPE 5, see output of the SHOW-PRINT-JOB-STATUS command), the file is protected irrespective of the setting of this operand.

A spoolout job is also generated if the file to be output is reserved by a SECURE-

RESOURCE-ALLOCATION command. This reservation must, however, be canceled by the time the spoolout job is processed; otherwise the job is not executed.

The file to be output is locked until the end of the session if the LOC-F=*YES operand is specified in the MODPJAT macro and the job cannot be executed owing to reservation.

LOCKF=*UNCHANGED

Operand default value. The current setting is not changed.

LOCKF=<var: enum-of _lockf_set:1>

File protection is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

LOCKF=*YES

The file is protected while the spoolout job is in a wait state.

LOCKF=*YES is ignored if either *EAM or *SYSTEM_FILE is simultaneously specified in the DTYPE operand.

An MODPJAT macro to a library element is rejected if LOCKF=*YES is specified. The LOCKF=*YES file protection also remains effective if the spoolout job is processed in the next system run.

LOCKF=*NO

The file is not protected while the spoolout job is in the wait state. The file can be deleted or modified before processing of the spoolout job commences. LOCK-FILE=*NO is ignored for temporary files.

DELF=<u>*UNCHANGED</u> / <var: enum-of _delf_set: 1> / *NO / *YES / *DESTROY

Specifies whether the file is to be deleted at the end of output and, if so, whether its catalog entry and the data are to be overwritten with X'00..0'. By default, the file is not deleted after printing and not overwritten with binary zeros. You must have write access to the file. If the file to be output belongs to a file generation group, the DELF operand is ignored. If a spoolout job is used to print several elements of a PLAM library with SECTREC=..., the DELF operand is set to *NO (i.e. suppressed). The operand must not be specified together with *SYSLST, SYSLSTnn or *SYSOUT.

DELF=*UNCHANGED

Operand default value. The current setting is not changed.

DELF=<var: enum-of _delf_set: 1>

File deletion is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

1	*YES
2	*DESTROY
3	*NO
See the parameter list description as of page 272.	

DELF=*NO

The file is not to be deleted after printing (exceptions: EAM and system files).

DELF=*YES

The file is to be deleted as soon as output has been completed.

DELF=*DESTROY

Not for EAM and cataloged system files.

Specifies that once the file has been printed both its catalog entry and its data are to be overwritten with binary zeros.

PAGECOP=*UNCHANGED / <var: int: 2> / <integer 0..255> / *STD

Specifies how many times each individual page is to be repeated. The ADDITIONAL-COPIES operand allows additional printouts of the whole file to be requested.

A PRINT-DOCUMENT command with the PAGE-COPIES operand is rejected if the line number specified in the LINE-PER-PAGE operand is greater than the number of lines in the loop record minus the number of lines before the line on which the vertical tab "channel 1" is defined.

On an HP or HP90 printer or on a 2050-APA or 2090-APA printer (or twin printer) a maximum of 255 copies of a page can be printed in succession.

PAGE-COPIES=255 has the same effect as PAGE-COPIES=254: one original and 254 copies are printed.

Im lokalen SPOOL-Betrieb nur für die Druckertypen HP, HP90, LP65.

PAGECOP=*UNCHANGED

Operand default value. The current setting is not changed.

PAGECOP=<var: int: 2>

Name of a field defined with FL. This field contains an integer with a length of 2 bytes which is interpreted as the number of page copies.

PAGECOP=<integer 0..255>

Integer value for the number of page copies.

PAGECOP=*STD

For LP65 printers:

The number of page copies is as specified in the PCL file.

For all other printer types:

PAGE-COPIES=0.

LEFTMAR=*UNCHANGED / <var: int: 1> / <integer 0..31> / *STD

Specifies whether the output text is to be indented.

The LEFTMAR operand is ignored if CONTMOD=*PHYSICAL is also specified.

LEFTMAR=*UNCHANGED

Operand default value. The current setting is not changed.

LEFTMAR=<var: int: 1>

Name of a field defined with FL. This field contains an integer with a length of 1 byte which is interpreted as the number of columns.

LEFTMAR=<integer 0..31>

For all printers apart from RSO printers: The output text is to be indented by the specified number of columns.

For all RSO printers:

The output text is to be indented by the specified number of * 1/10 inches.

LEFTMAR=*STD

The default is stored in the SPOOL parameter file and can be retrieved with the SHOW-SPOOL-PARAMETERS command or SPSERVE statement. The information can be found in the PRINT-CMD-DEFAULTS:...LEFT-MARGIN=... output field.

LINEPP=*UNCHANGED / <var: int: 2> / <integer 1..32767> / *STD

Specifies how many lines (including the header and blank lines) are to be printed on a page.

LINEPP=*UNCHANGED

Operand default value. The current setting is not changed.

LINEPP=<var: int: 2>

Name of a field defined with FL. This field contains an integer with a length of 2 bytes which is interpreted as the number of lines on a page.

LINEPP=<integer 1..32767>

Number of lines on a page.

LINEPP=*STD

If the operand is omitted, the number of lines per print page is calculated using the following formula, regardless of what has been specified for the HEADER-LINE operand:

Number of lines = P * L - N - 6

Where:

P = paper size in inches

L = line density

N = number of lines before the first channel 1

Printers with a loadable vertical format buffer

- The vertical tab "channel 1" controls the line on which printing is to start. Unless otherwise specified, 2 blank lines are set before printing starts; i.e. channel 1 (CHANNEL 01) is in the third line of the loop.
- If the value specified for the LINE-PER-PAGE operand is greater than the specified number of lines in the loop, the value in the loop is used.
- If the LINE-PER-PAGE operand is specified together with the HEADER-LINE and LINE-SPACING operands, a value specified here must be at least three times that of the line feed specified for LINE-SPACING=1/2/3.

HEADLIN=*UNCHANGED / <var: enum-of headline set:1> / *NO /

*STD / *DATE / *FRECORD / *PAGE / *DATE_FRECORD /

*DATE_PAGE / *FRECORD_PAGE / *DATE_FRECORD_PAGE

Specifies whether a header line is to be printed on every page (except the header and trailer pages).

HEADLIN=*UNCHANGED

Operand default value. The current setting is not changed.

HEADLIN=<var: enum-of headline set:1>

The header line is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

1	*NO	
2	*STD	
4	*FRECORD	
8	*DATE	
12	*DATE_FRECORD	
16	*PAGE	
20	*FRECORD_PAGE	
24	*DATE_PAGE	
28	*DATE_FRECORD_PAGE	
See th	See the parameter list description as of page 346.	

HEADLIN=*NO

No header line is printed.

HEADLIN=*STD

The header has the following format:

Header	DATE yyyy-mm-dd	userid	file	PAGE nnnn
Column	1	41(11,11)	60(21,67)	124 (77)

The first value in parentheses applies in the case of a form definition with a line length < 132 and ≥ 80 characters (with differing values). The second value in parentheses applies in the case of a form definition with a line length < 80 and ≥ 70 characters (with differing values).

yy-mm-dd year-month-day userid user identification

file file name

nnnn page number

Unless otherwise specified, the header line is followed by a blank line. The header is shifted in accordance with the LEFT-MARGIN operand, but the header line will be truncated at column 132. DATE and PAGE are only present in the header line if the line size \geq 032.

HEADLIN=*DATE / *FRECORD / *PAGE / *DATE_FRECORD / *DATE_PAGE / *FRECORD_PAGE / *DATE_FRECORD_PAGE

The header has the following format:

HEADER- LINE=	DATE	FIRST-RECORD	PAGE-NUMBER
Header	DATEyyyy-mm-dd (yyyy-mm-dd)	First record	PAGE nnnn (nnnn, nnnn)
Column	1	21	124 (77,67)

*DATE:

The value in parentheses applies in the case of a form definition with a line length < 132 and ≥ 70 characters (with differing values).

*PAGE-NUMBER:

The first value in parentheses applies in the case of a form definition with a line length < 132 and ≥ 80 characters (with differing values). The second value in parentheses applies in the case of a form definition with a line length < 80 and ≥ 70 characters (with differing values).

yyyy-mm-dd year-month-day

First record first logical record of the file

nnnn page number

If *DATE, *FRECORD or *PAGE is omitted, the appropriate section is filled with blanks.

Unless otherwise specified, the header line is followed by a blank line.

The header is shifted in accordance with the value in the LEFTMAR operand, but the header line will be truncated at column 132.

*DATE and *PAGE are only present in the header line if the line size ≥ 032.

If *FRECORD is specified, the first record is regarded as not belonging to the date.

The entries can be linked with the "_ " (underscore) character.

OUTFORM=<u>*UNCHANGED</u> / <var: enum-of _outform_set:1> / *CHARACTER / *HEXADECIMAL

Indicates whether the output format is character format only or character format and hexadecimal format.

OUTFORM=*UNCHANGED

Operand default value. The current setting is not changed.

OUTFORM=<var: enum-of outform set:1>

Character output is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

1	*CHARACTER
2	*HEXADECIMAL
See the parameter list description as of page 272.	

OUTFORM=*CHARACTER

Outputs in character format only. Records which exceed the length of a print line are truncated.

OUTFORM=*HEXADECIMAL

Outputs the data records in character format and in hexadecimal format.

Output format

Each output line starts with an 8-byte prefix followed by data 50 bytes in length.

Each output line is first printed using the appropriate font and then repeated in hexadecimal format.

Format of the output line

Column	Contents
1-4	Number of the byte in the record which is output first in this line
5-8	Blanks
as of 9	Characters in the input record using the appropriate font; the individual characters are separated by a blank. The characters are repeated in hexadecimal format in the next line

TWOSIDE=*UNCHANGED / <var: enum-of _twoside_set:1> / *STD / *NO / *YES / *TUMBLE / *IGNORE

Specifies whether the LP65 printer, PCL printer, APA printer or the 4830 printer is to print the paper on one side or two.

TWOSIDE=*UNCHANGED

Operand default value. The current setting is not changed.

TWOSIDE=<var: enum-of _twoside_set:1>

Single-sided or double-sided printing is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*STD	
1	*YES	
2	*TUMBLE	
4	*NO	
See the parameter list description as of page 272.		

Note

The functions TWO-SIDED, ELECTRONIC-OVERLAY and PAGE-COPIES are linked and part of one and the same printer command. If only one of these operands is specified, default values are generated for the others.

TWOSIDE=*STD

The paper is printed on one side (simplex mode) or two (duplex mode), as defined in the PCL file.

TWOSIDE=*NO

The paper is printed on one side (simplex mode).

TWOSIDE=*YES

The paper is printed on two sides (duplex mode).

TWOSIDE=*TUMBLE

The paper is printed on two sides (duplex mode), and the pages are turned over from top to bottom rather than left to right.

TWOSIDE=*IGNORE

The setting for the TWOSIDE operand is to be ignored.

ROT=*UNCHANGED / <var: enum-of _rotation_set: 1> / *NO / *ROT_0 / *ROT_90 / *ROT_180 / *ROT_270 / *ROT_0_180 / *ROT_90_270 / *ROT_180_0 / *ROT_270_90 / *BY_CONTROL_CODES

Specifies whether the pages to be printed from the spoolout job are to be rotated, and if so by how many degrees.

Only the values 0, 90, 180 and 270 may be specified for printer types 4830-PCL and 4850-PCL.

The print page set up in the printer is rotated (clockwise) by a certain number of degrees and printed on the form; for example, paper inserted in the printer in portrait format can be printed in landscape format. A separate loop is needed for pages rotated through 90°/270° (see the LOOP-NAME operand). Unless ROT=*NO is specified, output is directed (automatically) to HP and HP90 printers or PCL printers. For HP and HP90 printers SHOW-SPOOL-PARAMETERS indicates whether or not an HP or HP90 printer, PCL printer or 4830-PCL and 4850-PCL printer with a page rotation module is available in the current SPOOL configuration: output field DEVICE-TYPE:..., ROT=YES/NO.

Spoolout jobs with page rotation are displayed in the outputs for the SHOW-USER-STATUS and SHOW-PRINT-JOB-STATUS JOB-IDENTIFICATION=TSN(TSN=...) commands.

The feed for rotated pages is generally controlled via a separate loop (ROTATION-LOOP-NAME). If you have specified neither a ROTATION-LOOP-NAME for PRINT-DOCUMENT nor a form with a defined ROTATION-LOOP (see ADD-SPOOL-FORM), the default rotation loop R06 in the default form handles feed control for the rotated pages. Header and trailer

ROT=*UNCHANGED

Operand default value. The current setting is not changed.

ROT=<var: enum-of _rotation_set: 1>

pages are not printed out in rotated format.

Page rotation is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

213	*NO	
232	*BY_CONTROL_CODES	
241	*ROT_90	
242	*ROT_180	
243	*ROT_270	
244	*ROT_0_180	
245	*ROT_90_270	
246	*ROT_180_0	
247	*ROT_270_90	
See th	See the parameter list description as of page 272.	

ROT=*NO

Page rotation is not performed. Any control characters for page rotation in the file are not interpreted.

ROT=*ROT_0 / *ROT_90 / *ROT_180 / *ROT_270

Each print page is rotated by 900 / 1800 / 270 0 (clockwise) and printed out. Control characters for page rotation contained in the file are not interpreted. A separate loop is needed for pages rotated through 900 / 2700. You must check that output with the specified loop does not lead to errors. If an error occurs, the job is rejected.

ROT=*ROT_0_180 / *ROT_90_270 / *ROT_180_0 / *ROT_270_90

The odd pages (number before the hyphen) and even pages (number after the hyphen) are to be printed at different angles.

ROT=*BY CONTROL CODES

Page rotation control characters in the file are to be evaluated even if CONTMOD=*PHYSICAL is also specified.

INTRAY=<u>*UNCHANGED</u> / <var: int: 1> / <integer 1..99> / *STD / *IGNORE / *BY_FORMAT(...)

Specifies the paper input tray for LP65 printers, PCL printers, APA printers, and for the RSO printers 2030-PCL, 4011, 4812, 4813, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL, 4830-PCL, 4850-PCL, 9000-EPLQ, 9000-EPSQ, 9000-PCL, 9004, 9011, 9012, 9013, 9014, 9015, 9021, 9022, 9022-200, 9025, 9026-PCL, 9026-RENO, 9097 and DJET. Only input tray numbers 1 to 3 can be used with printer type LP65 and input tray numbers 1 to 99 for RSO printers.

INTRAY=*UNCHANGED

Operand default value. The current setting is not changed.

INTRAY=<var: int: 1>

Name of a field defined with FL. This field contains an integer with a length of 1 byte which is interpreted as the number of the paper input tray.

INTRAY=<integer 1..99>

Number of the input tray from which the paper is taken for printing the file itself as well as the header page and trailer page.

RSO does not check the specified value. Possible values: 1...99.

The following table contains the printer types and the maximum number of input trays in each case (column "Max.").

Printer	Max.	Printer	Max.	Printer	Max.	Printer	Max.
2030-PCL	2	8121	0	9002	0	9022	2
4011	2	9000	0	9003	0	9022-200	2
4812	2	9000-PCL	2	9004	3	9025	2
4813	2	9000-PRO	0	9011	2	9026-PCL	4
4818-PCL	2	9000-PS	0	9012	2	9026-RENO	4
4821-PCL	2	9000-EPFX	0	9013	3	9045-ANSI	0
4822-PCL	3	9000-EPLQ	2	9014	3	9046	0
4824-PCL	2	9000-EPSQ	2	9015	2	9645	0
4825-PCL	3	9001	0	9021	2	DJET	1
4830-PCL	3	9001-31	0		1		•
4850-PCL	2	PCL-Drucker	3				

INTRAY=*STD

If a print job is addressed to the local spool or to a special printer pool, this operand is ignored. But if the print job is addressed to a specific RSO printer, for instance, paper is taken from the input tray defined in the device record of the LP65 printer in the SPOOL parameter file. The input tray selection is valid for the entire spoolout process, i.e. including processing of the header and trailer pages.

INTRAY=*IGNORE

Only for RSO.

The paper input tray is selected via a PCL file or an input on the printer console.

The PCL file itself can also select a paper input tray.

The order of priorities for selecting a paper input tray is as follows:

- 1. the paper input tray number specified in the MODPJAT macro
- 2. any possible selection via the PCL file which was specified in the MODPJAT macro
- 3. any possible selection via the PCL file which was specified in the form definition in the SPOOL parameter file
- 4. the default values defined in the device entry.

INTRAY=*BY_FORMAT (...)

The paper input tray is defined via a keyword for the format in the INTRAYF operand.

INTRAYF=<var: enum-of _intray_format_set:1> / *MANUAL / *A3 / *A4 / *A5 / *B4 / *B5 / *DOUBLE_LETTER / *EXEC / *FOLIO / *INVOICE / *LEGAL / *LETTER / *MONARCH / *COMMERCIAL_10 / *DL / *C5 / *A3_UNCUT / *A4_UNCUT / *LEDGER

Specifies the paper input tray via a keyword for the format.

In order to ensure that the values specified for paper size and line length in the INTRAYF operand do not exceed the maximum values permitted for the selected paper format, the values of the INTRAYF operand are compared with the maximum values permitted for the paper format. The maximum values permitted for each paper format are listed in the following table.

Paper format	Maximum page size	Maximum line size
А3	165	116
A4	116	82
A5	32	58
B4	143	101
B5	101	71
FOLIO	129	85
INVOICE	85	55
EXEC	105	722
LEGAL	140	85
LETTER	110	85
DOUBLE-LETTER	150	117
MONARCH	75	38
COMERCIAL-10	95	41
DL	86	43
C5	90	63
a3-uncut	120	120
a4-uncut	169	84
ledger	170	110

INTRAYF=<var: enum-of _intray_format_set:1>

The paper input tray is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

128	MANUAL
129	A3
130	A4
131	A5
132	B4
133	B5
134	DOUBLE-LETTER
135	EXEC
136	FOLIO
137	INVOICE
138	LEGAL
139	LETTER
140	MONARCH
141	COMMERCIAL-10
142	DL
143	C5
144	A3_UNCUT
145	A4_UNCUT
146	LEDGER
See th	e parameter list description as of page 272.

These values are defined in the ASSEMBLER, C and COBOL interface. For example, the following is in the C interface:

```
/* intray_format_set
                                                                          */
/* ENUM _intray_format_set
                                                                          */
<< The input tray is not specified explicitly, but rather via>>
<< the following constant definition>>
<< (the integer value stored in the constant can therefore be used>>
<< in the parameter list of the function call): >>
#define PRNTDOCintray format manual 128
                                            /* intray format = *manual
#define PRNTDOCintray_format_a3 129
                                             /* intray format = *a3
                                                                           */
#define PRNTDOCintray format a4 130
                                              /* intray format = *a4
                                                                           */
```

```
#define PRNTDOCintray format a5 131
                                               /* intray format = *a5
                                                                             */
#define PRNTDOCintray format b4 132
                                               /* intray format = *b4
                                                                             */
#define PRNTDOCintray format b5 133
                                               /* intray format = *b5
                                                                             */
#define PRNTDOCintray format double-letter 134
/* intray format = *double-letter*/
#define PRNTDOCintray format exec 135
                                                                             */
                                               /* intray format = *exec
#define PRNTDOCintray_format_folio 136
                                               /* intray format = *folio
                                                                             */
#define PRNTDOCintray format invoice 137
                                               /* intray format = *invoicel
                                                                             */
#define PRNTDOCintray format legal 138
                                               /* intray format = *legal
                                                                             */
#define PRNTDOCintray_format_letter 139
                                               /* intray format = *letter
                                                                             */
#define PRNTDOCintray format monarch 140
                                               /* intray format = *monarch
                                                                             */
#define PRNTDOCintray_format_commercial-10 141
/* intray_format = *commercial-10 */
#define PRNTDOCintray_format_dl 142
                                               /* intray format = *dl
                                                                             */
                                               /* intray format = *c5
#define PRNTDOCintray_format_c5 143
                                                                             */
```

INTRAYF=*MANUAL

The paper is fed in manually and you should therefore insert a page each time a new page is to be printed or you are requested to do so by the printer.

The following printers support manual paper feed: 9014, 9021, 9022, 9022-200, 9026-Reno, 9026-PCL, 4812, DJET, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL and 9000-PCL.

INTRAYF=*A3

The paper is taken from the first input tray containing A3 pages.

The value *A3 can be used for the printers 4830-PCL, 9026-PCL and 9026-RENO.

INTRAYF=*A4

The paper is taken from the first input tray containing A4 pages.

The value *A4 can be used for the printers 9021, 9000-PCL, 9026-PCL, 9026-RENO, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL and 4830-PCL.

INTRAYF=*A5

The paper is taken from the first input tray containing A5 pages.

The value *A5 can be used for the printers 4830-PCL, 9026-PCL and 9026-RENO.

INTRAYF=*B4

The paper is taken from the first input tray containing B4 pages.

The value *B4 can be used for the printers 4830-PCL and 9026-RENO.

INTRAYF=*B5

The paper is taken from the first input tray containing B5 pages.

The value *B5 can be used for the printers 4830-PCL, 9026-PCL and 9026-RENO.

INTRAYF=*FOLIO

The paper is taken from the first input tray containing FOLIO pages.

The value *FOLIO can be used for the printer 9026-RENO.

INTRAYF=*INVOICE

The paper is taken from the first input tray containing INVOICE pages. The value *INVOICE can be used for the printers 9026-PCL and 9026-RENO.

INTRAYF=*EXEC

The paper is taken from the first input tray containing EXEC pages.

The value *EXEC can be used for the printers 9021, 9000-PCL, 9026-PCL, 9026-RENO, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL and 4830-PCL.

INTRAYF=*LEGAL

The paper is taken from the first input tray containing LEGAL pages.

The value *LEGAL can be used for the printers 9021, 9000-PCL, 9026-PCL, 9026-RENO, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL and 4830-PCL.

INTRAYF=*LETTER

The paper is taken from the first input tray containing LETTER pages.

The value *LETTER can be used for the printers 9021, 9000-PCL, 9026-PCL, 9026-RENO, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL and 4830-PCL.

INTRAYF=*DOUBLE-LETTER

The paper is taken from the first input tray containing DOUBLE-LETTER pages.

The value *DOUBLE-LETTER can be used for the printer 9026-RENO.

INTRAYF=*MONARCH

The paper is taken from the first input tray containing MONARCH pages. The value *MONARCH can be used for the printers 9021, 9000-PCL, 9026-PCL, 9026-RENO, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL and 4825-PCL.

INTRAYF=*COMMERCIAL-10

The paper is taken from the first input tray containing COMMERCIAL-10 pages. The value *COMMERCIAL-10 can be used for the printers 9021, 9000-PCL, 9026-PCL, 9026-RENO, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL and 4825-PCL.

INTRAYF=*DL

The paper is taken from the first input tray containing International-DL envelopes. The value *DL can be used for the printers 9021, 9000-PCL, 9026-PCL, 9026-RENO, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL and 4830-PCL.

INTRAYF=*C5

The paper is taken from the first input tray containing International-C5 envelopes. The value *C5 can be used for the printers 9021, 9000-PCL, 9026-PCL, 9026-RENO, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL and 4830-PCL.

INTRAYF=*A3-UNCUT

The paper is taken from the first input tray containing A3-UNCUT pages.

The value *A3-UNCUT can only be used for PCL printers.

INTRAYF=*A4-UNCUT

The paper is taken from the first input tray containing A4-UNCUT pages.

The value *A4-UNCUT can only be used for PCL printers.

INTRAYF=*LEDGER

The paper is taken from the first input tray containing LEDGER pages.

The value *LEDGER can only be used for PCL printers.

OUTTRAY=*UNCHANGED/<var: int: 1>/<integer 1..99>/*STD/*IGNORE/*SORTER

Specifies the paper output tray for LP65 printers, APA-printers, and for RSO printers 2030-PCL, 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL, 4830-PCL,

4850-PCL, 9000-PCL, 9014, 9015, 9026-PCL and 9026-RENO.

The job can only be executed on single page printers.

OUTTRAY=*UNCHANGED

Operand default value. The current setting is not changed.

OUTTRAY=<var: int: 1>

Name of a field defined with FL. This field contains an integer with a length of 1 byte which is interpreted as the number of the paper output tray.

OUTTRAY=<integer 1..99>

Specifies the paper output tray to be used for the current job.

OUTTRAY=*STD

Output goes into the paper output tray defined as the standard output tray in the device entry (SPOOL parameter file).

The paper output tray selection applies for the entire spoolout process, i.e. including the header and trailer pages.

The paper output tray selection by BS2000 can be prevented via a PCL file or an input at the printer console. The OUTTRAY entries have no effect in these cases. A paper output tray can also be specified in the PCL file.

The order of priority for selecting an output tray is as follows:

- 1. The number of the paper output tray specified in the MODPJAT macro, provided this is not prevented by a PCL file or an entry at the printer console.
- 2. Any selection via the PCL file specified in the MODPJAT macro.
- 3. Any selection via the PCL file specified in the form definition in the SPOOL parameter file.
- 4. The default values defined in the device entry.

OUTTRAY=*IGNORE

If you specify this value, the print control does not send a code to the printer for selecting the output tray. This allows you to define the output tray in the prolog file.

OUTTRAY=*SORTER

Specifies that the sort mechanism is to be used for the current job, using the SRTMODE operand. The sort mechanism is permitted for printers 9026-RENO, 4818-PCL, 4822-PCL and 4825-PCL and can be employed for up to 20 paper output trays. It cannot be controlled via the default values defined in the device entry.

Note

Print resources such as PROLOG, EPILOG, DIA, MEMBER and font character files are sent before the paper output tray is selected.

COVPAGE=(headtxt, headexi, traiexi)

headtxt: <var: char: 32> / <c-string 1..32: c-string_1..32> / *NONE Specifies whether a string is to be printed on the header and trailer pages.

headtxt: <var: char: 32>

Name of a field defined with CL. This field contains a string with a length of 32 bytes which is interpreted as information to be stored.

headtxt: <c-string 1..32: c-string_1..32>

The specified information (maximum of 32 characters) is stored in the SCB for processing the system exits. The first 8 characters are printed on the header page as an uppercase line under the mailing box. Only alpha characters, digits and a number of special characters are printed on the header page (in outsize type). All other character codes are automatically replaced by the printable character '?'.

Format of the header page

- 1. User ID in outsize letters (10 lines + 2 blank lines)
- 2. Account number in outsize letters (10 lines + 2 blank lines)
- 3. Job name in outsize letters (10 lines + 2 blank lines)
- 4. Mailing box (address and identification field: 12 lines + 2 blank lines)
- 5. 'text' in outsize letters (10 lines + 2 blank lines)

Priority sequence

- 1. Mailing box (address and identification field: 12 lines + 2 blank lines)
- 2. 'text' in outsize letters (10 lines + 2 blank lines)
- 3. Job name in outsize letters (10 lines + 2 blank lines)
- 4. User ID in outsize letters (10 lines + 2 blank lines)
- 5. Account number in outsize letters (10 lines + 2 blank lines)

headtxt: *NONE

No string is to be printed on the header and trailer pages.

headexi: *UNCHANGED / <var: int: 4> / <integer 0..2147483639> / *NO

Number of desired header pages.

The precise significance of this operand depends on the definitions made for the computer center.

headexi: *UNCHANGED

Operand default value. The current setting is not changed.

headexi: <var: int: 4>

Name of a field defined with FL. This field contains an integer with a length of 4 bytes which is interpreted as the number of header pages.

traiexi: *UNCHANGED / <var: int: 4> / <integer 0..2147483639> / *NO

Number of desired trailer pages.

The precise significance of this operand depends on the definitions made for the computer center.

traiexi: *UNCHANGED

Operand default value. The current setting is not changed.

traiexi: <var: int: 4>

Name of a field defined with FL. This field contains an integer with a length of 4 bytes which is interpreted as the number of trailer pages.

FORM=*UNCHANGED / <var: char: 6> / <c-string 1..6: alphanum_name_1..6> / *STD Specifies the paper (form) to be used for output (e.g. STD, STDSF1, STDWA4)

FORM=*UNCHANGED

Operand default value. The current setting is not changed.

FORM=<var: char: 6>

Name of a field defined with CL. This field contains a string with a length of 6 bytes which is interpreted as the name of the form.

FORM=<c-string 1..6: alphanum_name_1..6>

Name of the form with which the spoolout job is to be processed. A loop (or a page and format definition for APA printers) is implicitly named when the form is specified. The associated loop (or the PAGEDEF and FORMDEF) must be in a printer control file. The following table shows which printer control file must contain this loop (or the PAGEDEF and FORMDEF) for the relevant printer type:

Output device	Printer control file with the loop for the specified form
Printers 3337, 3338, 3339, 3348, 3349, 3365, 2090-LIP, 2140-LIP, LP-EMULATED	\$SYSSPOOL.PRFILE
Printers: 3351, 3353, 2090, 2140, 2240, PCL printer	\$SYSSPOOL.PRFILE or the user PRFILE specified in the USER-RESOURCES-FILE operand
APA printers	\$SYSSPOOL.SYSPRT.SPS.021 or the user SPSLIB specified in the USER-RESOURCES-FILE operand

The loop named implicitly via the FORM operand is ignored if the LOOP operand is specified at the same time. If the FORM and LOOP operands are omitted, the file is printed out using the default form entered for the printer type.

A loop explicitly specified in the LOOP operand must have the same length as the loop record assigned to the form used.

No loops can be specified for APA printers. If page and format specifications are made in the FORM operand, these are used for printing header, trailer and message pages.

FORM=*STD

Default form.

LOOP=*UNCHANGED / <var: char: 3> / <c-string 1..3: alphanum_name_1..3> / *STD Name of the loop to be loaded into the feed information buffer (VFB/FCB). The loop name must not include any of the characters '\$', '&' or '@'.

LOOP=*UNCHANGED

Operand default value. The current setting is not changed.

LOOP=<var: char: 3>

Name of the loop which is to control line feed. The length of the specified loop must match the length of the default loop of the form used. A loop for feed control is needed for the HP and HP90 printers (3351, 3353, 2090, 2140) and printers of types 3337, 3338, 3339, 3348, 3349, 3365, 2140-LIP and 2090-LIP. Loops are stored in the PRFILE resource library. If no loop is specified, the implicit entries in the FORM operand are used. If the FORM or LOOP operand is omitted, default values apply.

LOOP=<c-string 1..3: alphanum_name_1..3>

Name of a field defined with CL. This field contains a string with a length of 3 bytes which is interpreted as the name of the loop.

LOOP=/ *STD

The spoolout job line feed is to be controlled with the default loop of the form used.

ROTLOOP=<u>*UNCHANGED</u> / <var: char: 3> / <c-string 1..3: alphanum_name_1..3> / *STD

Only for HP and HP90 printers with a page rotation module and for RSO and PCL printers. Specifies the loop to control output in landscape format. The loop name must not include any of the characters '\$', '&' or '@'.

ROTLOOP=*UNCHANGED

Operand default value. The current setting is not changed.

ROTLOOP=<var: char: 3>

Name of a field defined with CL. This field contains a string with a length of 3 bytes which is interpreted as the name of the loop.

ROTLOOP=<c-string 1..3: alphanum_name_1..3>

A string enclosed in quotes for the name of the loop to control the feed of rotated pages.

ROTLOOP=*STD

Output of rotated pages is to be controlled by the default rotation loop of the specified form or the default rotation loop R06 of the default form (if no form was specified in the MODPJAT macro).

CHARSET=*UNCHANGED / array(16): <var: char: 3> /

<c-string 1..3: alphanum_name_1..3> / *STD

Names of the fonts to be used for output.

Up to 4 fonts may be specified in the list for local SPOOL, up to 16 for RSO. For local SPOOL, the fonts must be contained in the \$SYSSPOOL.PRFILE resource library or in a user PRFILE (specified by means of the USERRES operand). If more than four fonts are to be used, a font pool must be specified (POOLNAM operand).

If more than one font is to be used in a spoolout job, CONTMOD=*PAGE_MODE must be specified.

For DOCFORM=*TEXT, only the first specified font is used for printing the (entire) file. The name of the first font and the number of specified fonts are shown in the output for the command SHOW-PRINT-JOB-STATUS JOB-IDENTIFICATION=TSN(TSN=...).

Only FUJITSU-SIEMENS standard fonts may have names consisting of three numerical characters.

If the HEADLIN operand is specified, the first font specified is used for the header line. The default value for CHARSET for the form used can be displayed with SHOW-SPOOL-FORMS; the information is given in the C-S output field.

The following point applies to HP printers:

Systems support can specify whether the header page is printed with the default font or with the font specified in CHARSET. This default setting can be displayed using the SHOW-SPOOL-PARAMETERS command or SPSERVE statement; the information is given in the HEADER-PAGE: CHARACTER-SET=... output field.

CHARSET=*UNCHANGED

Operand default value. The current setting is not changed.

CHARSET=<var: char: 3>

Name of a field defined with CL. This field contains a string with a length of 3 bytes which is interpreted as the name of the font.

CHARSET=<c-string 1..3: alphanum_name_1..3>

Names of the fonts with which the spoolout job is to be processed. For DOCFORM=*TEXT (default), only the first specified font is used for printing.

For laser printers in local SPOOL mode, a maximum of four fonts may be specified. The printer control characters for changing fonts are only interpreted if

CONTMODE=*PAGE_MODE is also specified.

In the case of RSO printers, a maximum of 16 fonts may be specified. Font identifiers (CSIs) in the text are only interpreted if CONTROL-MODE=*LOGICAL is specified at the same time. The command is rejected if a list of fonts is specified in conjunction with DOCFORM=*TEXT.

CHARSET=*STD

The default font is selected for this printer type from the desired form. It can be displayed by means of SHOW-SPOOL-FORMS.

POOLNAM=*UNCHANGED / <var: char: 4> / <c-string 1..4: name_1..4> / *NONE Name of the font pool (only for HP and HP90 printers) to be used for the output.

POOLNAM=*UNCHANGED

Operand default value. The current setting is not changed.

POOLNAM=<var: char: 4>

Name of a field defined with CL or a register containing the value. This field or register contains a string with a length of 4 bytes which is interpreted as the name of the font pool.

POOLNAM=<c-string 1..4: name_1..4>

Specifies a font pool (with a maximum of 64 fonts for an HP printer or 46 fonts for an HP90 printer) from which one or more fonts are to be used for output. All the fonts of this font pool are loaded into the font buffer when the spoolout job is executed.

POOLNAM=*NONE

No font pool is required for the output.

POOLIND=*UNCHANGED / <var: int: 1> / <integer 0..64>

Number of the font from the font pool with which the spoolout job is to be processed. The number of the font is determined by its position in the definition of the font pool. The specified font is used if POOL-INDEX is specified together with CONTROL-MODE=*PAGE-MODE.

POOLIND=*UNCHANGED

Operand default value. The current setting is not changed.

POOLIND=<var: int: 1>

Name of a field defined with FL. This field contains an integer with a length of 1 byte which is interpreted as the font number.

POOLIND=<integer 0..64>

Integer value for the font number from the font pool.

EFO=<u>*UNCHANGED</u> / <var: char: 2> / <c-string 2..2: c-string_2..2> / *NONE

Specifies whether a film overlay - in the case of HP90 and 3365 printers an EFO data overlay - is to be used for processing the spoolout job.

EFO=*UNCHANGED

Operand default value. The current setting is not changed.

EFO=<var: char: 2>

Name of a field defined with CL. This field contains a string with a length of 2 bytes which is interpreted as the name of the film overlay.

EFO=<c-string 2..2: c-string_2..2>

Name of the film overlay (HP90 and 3365: EFO data overlays) to be used for processing the spoolout job (the name must be approved by systems support).

EFO=*NONE

No film overlay (HP90 and 3365: no EFO data overlay) is used for output.

OVERLAY=(face, reverse)

Specifies for LP65 printers whether EFO data overlays are to be used on the recto and/or verso. They must be stored in printer memory.

The OVERLAY, TWOSIDE and PAGECOP operands are part of one and the same LP65 printer command and are consequently linked to one another. If only the OVERLAY operand is specified, SPOOL will generate default values for the other two operands. These default values are overwritten by each value specified in a PCL file or at the printer control console. For an overview of the possible combinations of these three linked functions, see the description of the printer type LP65 in the "SPOOL (BS2000/OSD)" manual.

face: <var: int: 1> / <integer 1..127> / *STD / *NONE

Identification number of the overlay to be used on the recto.

face: <var: int: 1>

Name of a field defined with FL. This field contains an integer with a length of 1 byte which is interpreted as the identification number.

face: <integer 1..127>

Integer value for the overlay identification number to be used on the recto.

face: *STD

The default value for the overlay identification number (or *NONE) is stored in the SPOOL parameter file.

face: *NONE

No overlay is to be used on the recto.

reverse: <var: int: 1> / <integer 1..127> / *STD / *NONE Identification number of the overlay to be used on the verso.

reverse: <var: int: 1>

Name of a field defined with FL. This field contains an integer with a length of 1 byte which is interpreted as the identification number.

reverse: <integer 1..127>

Integer value for overlay identification number to be used on the verso.

reverse: *STD

The default value for the overlay identification number (or *NONE) is stored in the SPOOL parameter file.

reverse: *NONE

No overlay is to be used on the verso.

FOB=*UNCHANGED / <var: char: 4> /

<c-string 1..4: alphanumname_1..4, c-string_1..4> / *NONE

Defines whether an FOB data overlay is to be used for processing the spoolout job.

FOB=*UNCHANGED

Operand default value. The current setting is not changed.

FOB=<var: char: 4>

Name of a field defined with CL or a register containing the value. This field or register contains a string with a length of 4 bytes which is interpreted as the overlay name.

FOB=<c-string 1..4: alphanumname_1..4, c-string_1..4>

Name of the overlay to be used for processing the spoolout job. If an overlay is specified, the file is printed out on an HP or HP90 printer. Use of an overlay for the spoolout job is indicated in the output of the command SHOW-PRINT-JOB-STATUS JOB-IDENTIFICATION=TSN(TSN=...).

FOB=*NONE

No FOB data overlay is used for output.

PAGEPCL=*UNCHANGED / <var: int: 4> / <integer 1..50000> / *STD

Specifies which page definition is to be used for output on LP65 or APA printers.

PAGEPCL=*UNCHANGED

Operand default value. The current setting is not changed.

PAGEPCL=<var: int: 4>

Name of a field defined with FL. This field contains an integer with a length of 4 bytes which is interpreted as the number of the PCL file.

PAGEPCL=<integer 1..50000>

Only for LP65 printers:

Integer value for the number of the PCL file with which the print file is to be output.

PAGEPCL=*STD

For APA printers:

The default definition specified in the SPSLIB is to be used (see the "PRISMAproduction/BS2000 - SPS" manual).

For LP65 printers:

Number of the PCL file with which the print file is to be output. Only the print file itself is output with this PCL file. The header and trailer pages are controlled by the PCL file defined in the SPOOL parameter file form.

Note

A PCL file that makes it impossible to select another PCL file by means of a channel command must not be used.

After the specified PCL file has started, the operating mode of the printer is checked. If EXCCW mode is active, and an error occurs during the print process, the restart point is the last SECTION record; if no value was entered under the SECTION operand in the PRINT-DOCUMENT command, the file is printed out again from the start.

USERRES=*UNCHANGED / <var: char: 44> /

<c-string 1..44: filename_without-userid_1..44, c-string_1..44> / *STD

Specifies a user file containing all the resources required for output on different printer types: user-defined loops, fonts, overlays, font pools, code translation tables and SPS data stream definitions. The following can be specified:

- a user PRFILE containing loops, fonts, overlay entries (FORMS-OVERLAY-BUFFER operand) and font pool entries (the CHARACTER-SETS operand)
- a user SPSLIB containing the PAGEDEFs (PAGE-DEFINITION operand),
 FORMDEFs (FORM-DEFINITION operand), fonts, page segments, overlays and raster image data
- a user RSOFILE (only for RSO) containing loops

If no user PRFILE, SPSLIB or RSOFILE is specified, the information is taken from the following files: \$SYSSPOOL.PRFILE, \$SYSSPOOL.SYSPRT.SPS.021 or \$SYSSPOOL.RSOFILE, respectively.

USERRES=*UNCHANGED

Operand default value. The current setting is not changed.

USERRES=<var: char: 44>

Name of a field defined with CL or a register containing the value. This field or register contains a string with a length of 44 bytes which is interpreted as the name of the resources file.

USERRES=<c-string 1..44: filename_without-userid_1..44, c-string_1..44>

Name of a user PRFILE, SPSLIB or RSOFILE, which may contain a catalog ID and a user ID. SPOOL uses this file name with the suffix .PRFILE, .SPSLIB or .RSOFILE.

The string can contain up to 28 characters without the catalog ID and user ID, to ensure that this user file can be called from any ID.

If the file name is specified without a user ID, the file is searched for under the user ID of the caller first, then under SYSSPOOL. If it is not found, the command is rejected. If a user ID is specified, the file is searched for under this ID only.

Example 1

PRINT-DOCUMENT FILE, USER-RESOURCES-FILE=\$XX.XX

A search is carried out for the \$XX.XX.PRFILE file. If the file is not found, the command is rejected.

Example 2

If the catalog ID is specified, the search is limited to the specified pubset:

PRINT-DOCUMENT FILE, USER-RESOURCES-FILE=:A:XXXXX

The file :A:\$userid.XXXXX.PRFILE is searched for. If it is not found, a search is carried out for the file :A:\$SYSSPOOL.XXXXX.PRFILE. If this file is not found either, the command is rejected.

Example 3

PRINT-DOCUMENT FILE, USER-RESOURCES-FILE=:A:\$XX.XXXXX

The file: A:\$XX.XXXX.PRFILE is searched for. If it is not found, the command is rejected.

If the file is on an exported PVS (EXPORT-PUBSET command), all spoolout jobs that require this PVS are placed in the KEEP queue. When the PVS becomes available again (IMPORT-PUBSET command), the spoolout jobs are restarted.

USERRES=*STD

The required resources are taken from \$SYSSPOOL.PRFILE, \$SYSSPOOL.SYSPRT.SPS.021 or \$SYSSPOOL.RSOFILE.

TRANTAB=(name, file)

Specifies whether a code translation table is to be used for processing the spoolout job. The code translation table is necessary if the default escape character 'FF' is to be replaced by a random character.

name: <var: char: 8> / <c-string 1..8: alphanum_name_1..8, c-string_1..8> / *NONE Defines whether a code translation table is to be used for processing the spoolout job.

name: <var: char: 8>

Name of a field defined with CL. This field contains a string with a length of 8 bytes which is interpreted as the name of the code translation table.

name: <c-string 1..8: alphanum_name_1..8, c-string_1..8>

Name of the code translation table to be used for processing the spoolout job.

name: *NONE

No code translation table is used.

file: <var: char: 44> / <c-string 1..44: filename_without-userid_1..44,

c-string_1..44> / *SYSTEM

Specifies the file from which the code translation table is to be taken.

file: <var: char: 44>

Name of a field defined with CL. This field contains a string with a length of 44 bytes which is interpreted as the name of the file.

file: <c-string 1..44: filename_without-userid_1..44, c-string_1..44>

The name of the file containing the specified code translation table is specified with a string enclosed in quotes.

file: *SYSTEM

The specified code translation table is taken by default from the \$SYSSPOOL.PRFILE file; a user RSOFILE can be specified for RSO printers.

PRNAME=*<u>UNCHANGED</u> / <var: char 1..8> / <c-string: c-string_1..8> / *STD / *IPP Specifies the requested target printer of the print job. You can specify a distributed local printer pool, a non-distributed local printer pool, an RSO printer pool or an RSO printer. You can use Distributed Print Services to specify further target printers (see the "Distributed Print Services (BS2000/OSD)" manual for details).

If you specify a printer pool

The job is output on any printer in the specified device pool. The pool must be defined in the SPOOL parameter file and can contain up to 16 RSO devices or 16 local SPOOL devices, but not both at the same time. Device pools are managed by means of the SPSERVE statements ADD-, MODIFY-, REMOVE-, and SHOW-PRINTER-POOL (see the "SPSERVE (BS2000/OSD)" manual).

The spoolout job is rejected if:

- no printer type from the device pool is assigned in the PRINT-DOCUMENT command (see the SHOW-SPOOL-FORMS command or SPSERVE statement)
- the ELECTRONIC-OVERLAY operand is specified.

Please bear in mind that the pool can contain various types of devices: if the file to be printed contains control characters that are only interpreted by a specific printer type, you should specify a printer of this type in the MODPJAT macro. One option is to specify a form in the MODPJAT macro that is only defined for the desired printer type (ADD-SPOOL-FORM).

PRNAME=*UNCHANGED

Operand default value. The current setting is not changed.

PRNAME=<var: char 1..8>

Name of a field defined with CL. This field contains a string with a length of 8 bytes which is interpreted as the name of the printer.

PRNAME=<c-string: c-string_1..8>

Only for RSO.

Symbolic name of the device on which the spoolout job is to be processed.

PRNAME=*STD

The spoolout job is to be processed on the default device type specified in the SPOOL parameters (PRINT-CMD-DEFAULTS).

PRNAME=*IPP

Only for RSO.

Indicates that the spoolout job is to be processed on an IPP printer addressed by its URL.

PRTYPE=*UNCHANGED / <var: enum-of _prtype_set:1> / *ANY / *LP65 PRINTER / *HP PRINTER / *APA PRINTER

Specifies which printer type is to process the print job. Only local printer types can be specified.

PRTYPE=*UNCHANGED

Operand default value. The current setting is not changed.

PRTYPE=<var: enum-of _prtype_set:1>

The printer type is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*ANY	
1	*LP65_PRINTER	
2	*APA_PRINTER	
3	*HP_PRINTER	
See the parameter list description as of page 272.		

PRTYPE=*ANY

A specific printer type is not requested. In this case, the SPOOL subsystem automatically determines the permitted printer types that can process the user request. *ANY must be specified for output on RSO printers.

PRTYPE=*LP65 PRINTER

The spoolout job is to be processed on an LP65 printer.

PRTYPE=*HP PRINTER

The spoolout job is to be processed on an HP or HP90 printer. *HP_PRINTER includes the following printer types: 3351/3353/2090/2140/2240(HP90). The control characters for HP and HP90 laser printers are identical; a spoolout job for an HP printer can be processed on an HP90 printer and vice versa.

Selection of the printer type is affected by whether or not the CONTMOD operand is specified at the same time:

Printer selection	CONTMOD = *PAGE_MODE (default)	CONTMOD =(*PAGE_MODE,,*HP)	
PRINTER- TYPE = *ANY	Output is possible on all printer types	Output is possible on HP and HP90 printers. Other control characters are converted to HP/HP90 control characters.	
	The following applies to HP/HP90 pri The OVERPRINT function is converte		
PRINTER-TYPE = *HP-PRINTER	Output is possible only on HP and HP90 printers. The OVERPRINT function is converted to the LINE-MERGE function.		
	PRFILE restrictions: The file must not contain the X'FF' character.	HP-specific control characters are supported. The following restriction applies: The file can contain the X'FF' character only as an escape character if control character interpretation is active.	

Note for HP/HP90 printers

If the entries in MODPJAT specify output to an HP or HP90 laser printer, a PRFILE must be available, otherwise the command is rejected. If there is no HP/HP90 available in an installation, or only devices with insufficient configurations (not enough fonts, no graphics buffer for FOBs or no page rotation module, for example), the jobs can only be output to replay tape. Systems support can run the SHOW-SPOOL-JOB-STATUS command to obtain information on these jobs.

PRTYPE=*APA_PRINTER

The spoolout job is to be processed on a 2050-APA-PRINTER, 2090-APA-PRINTER or 2090-TWIN-PRINTER.

SRTMODE=*UNCHANGED / <var: enum-of _srtmode_set:1> / *NO / *GROUP / *COLLATE / *STACKER / *AUTOMATIC
Sets the printer sort mode.

SRTMODE=*UNCHANGED

Operand default value. The current setting is not changed.

You use this entry to specify that the sort mechanism is used for the current job. The sort mechanism is permitted for printers 9026-RENO, 4818-PCL, 4822-PCL and 4825-PCL and can be used for up to 20 paper output trays. It cannot be controlled via the default values defined in the device entry.

SRTMODE=*NO

All pages of the document are stored in the sort trays from bottom to top, with the exception of 9026-RENO: here, pages are stored in the most accessible paper tray.

SRTMODE=*GROUP

Each copy of a specific page - as many as were specified under PAGE-COPIES - is stored individually in a sort tray, i.e. the next page to be stored in each sort tray is always the next one in the printing sequence. When the job has finished printing, each sort tray used will contain a copy of the printed document.

Example

The file contains three pages, and PAGE-COPIES=2 was specified:

tray 1	tray 2	tray 3
page 1	page 1	
page 2	page 2	
page 3	page 3	

SRTMODE=*COLLATE

All copies of a page - as many as were specified under PAGE-COPIES - are collected in one sort tray. The copies of the next page are stored in the next sort tray, and so on. The sort trays are used from bottom to top.

Example

The file contains three pages, and PAGE-COPIES=2 was specified:

tray 1	tray 2	tray 3
page 1	page 2	page 3
page 1	page 2	page 3

SRTMODE=*STACKER

Cannot be used for 9026 printers.

All printed pages are stored in the stacker of the sort mechanism up to a maximum of 500 pages. This mode is useful if a single copy of a very long document is to be printed.

SRTMODE=*AUTOMATIC

For 9026 Printers only.

The storage behavior is selected automatically, depending on the number of copies per page requested in PAGE-COPIES and the number of sort trays. The printed pages are stored unsorted, as with *NO, if the number of copies specified in PAGE-COPIES is the same as the number of sort trays.

The printed pages are sorted by document, as with *GROUP, if the number of copies specified in PAGE-COPIES is less than the number of sort trays. Or the printed pages are stored in the sort trays sorted by page, as with *COLLATE, if the number of copies specified in PAGE-COPIES is greater than the number of sort trays.

SRTMODE=<var: enum-of _srtmode_set:1>

The sort mechanism is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

1	NO	
2	GROUP	
3	COLLATE	
4	STACKER	
5	AUTOMATIC	
See the parameter list description as of page 272.		

These values are defined in the ASSEMBLER, C and COBOL interface. For example, the following is in the C interface:

```
/* srtmode set
                                                                 */
/* ENUM _srtmode_set
                                                                 */
<< The sort mechanism is not specified explicitly, but rather via>>
<< the following constant definition>>
<< (the integer value stored in the constant can therefore be used>>
<< in the parameter list of the function call): >>
*/
                                                                 */
                                                                 */
#define PRNTDOCsrtmode_stacker 4 /* srtmode =*stacker
                                                                 */
#define PRNTDOCsrtmode automatic 5 /* srtmode =*automatic
                                                                 */
```

VIRTUAL=<u>*UNCHANGED</u> / <var: enum-of _virtual_set:1> / *ALLOWED / *NOT ALLOWED / *STD / *MUST

This operand allows the print job to be passed on to an application for processing via a virtual printer.

VIRTUAL=*UNCHANGED

Operand default value. The current setting is not changed.

VIRTUAL=<var: enum-of _virtual_set:1>

Whether and how a print job is passed to an application via a virtual printer is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	ALLOWED	
1	NOT-ALLOWED	
See the parameter list description as of page 272.		

These values are defined in the ASSEMBLER, C and COBOL interface. For example, the following is in the C interface:

VIRTUAL=*ALLOWED

The print job is preferably transferred to a virtual printer. If no virtual printer is active when the print job is entered in the list of print jobs, it is assigned to a real printer.

VIRTUAL=*NOT ALLOWED

This print job cannot be passed to an application program via a virtual printer.

VIRTUAL=*STD

The virtual printer which is to receive the print job is determined from the SPOOL parameter file.

The print job is preferably transferred to a virtual printer. If no virtual printer is active when the print job is entered in the list of print jobs, it is assigned to a real printer.

VIRTUAL=*MUST

The print job is transferred to a specific virtual printer which is defined by the parameters in the brackets. If the supervisor task of this virtual printer is inactive, the print job remains in the wait state.

PRJCLAS=*UNCHANGED / <var: int: 1> / <integer 1..255>

For the SPOOL administrator only.

Defines the job class of the spoolout job.

PRJCLAS=*UNCHANGED

Operand default value. The current setting is not changed.

PRJCLAS=<var: int: 1>

Name of a field defined with FL. This field contains an integer with a length of 1 byte which is interpreted as the job class number.

PRJCLAS=<integer 1..255>

The specified job class is assigned to the spoolout job. For reasons of compatibility the value 0 (zero) is not permitted.

PROGRAM=(name, string)

Specifies the device name of a virtual printer which is to process the spoolout job.

name: *UNCHANGED / <var: char: 8> /

<c-string 1..8: alphanum_name_1..8, c-string_1..8> / *ANY

Specifies the device name of a virtual printer which is to process the spoolout job.

name: *UNCHANGED

Operand default value. The current setting is not changed.

name: <var: char: 8>

Name of a field defined with CL. This field contains a string with a length of 8 bytes which is interpreted as the name of the virtual printer.

name: <c-string 1..8: alphanum_name_1..8, c-string_1..8>

The name of the virtual printer which is to process the spoolout job is specified with a string enclosed in quotes.

name: *ANY

Any virtual printer may process the spoolout job.

string: *NONE / <var: char: 32> /

<c-string 1..32: alphanum_name_1..32, c-string_1..32>

Specifies supplementary information for the virtual printer which is to process the spoolout job.

string: *NONE

Operand default value. No supplementary information for the virtual printer.

string: <var: char: 32>

Name of a field defined with CL. This field contains a string with a length of 32 bytes which is interpreted as supplementary information for the virtual printer.

string: <c-string 1..32: alphanum_name_1..32, c-string_1..32>

The supplementary information for the virtual printer which is to process the spoolout job is specified with a string enclosed in quotes.

FRMNAME=*UNCHANGED / *STD / <c-string: c-string 1..63> / <var: char: 63> / (<reg: char:63>)

Specifies the format to be processed. The document is processed in transparent mode. Name of the format in which the output data is passed to the printer. If no value was specified for FRMNAME, a default value is derived automatically from the value of the CONTMODE operand. This results in the following:

CONTMODE=*PAGE-MODE(...) FRMNAME=*HP
CONTMODE=*APA(...) FRMNAME=*SPDS
CONT-MODE=*LOGICAL / *PHYSICAL / *LINEMODE FRMNAME=*STD

The value FRMNAME=*STD means that the contents of the file to be printed are irrelevant.

FRMNAME=*UNCHANGED

Operand default value. The current setting is not changed.

FRMNAME=*STD

The value of FRMNAME is set to *STD.

FRMNAME=<c-string: c-string 1..63>

The format name is output directly.

FRMNAME=<var: char: 63>

The format name is output indirectly via a field.

FRMNAME=(<reg: char:63>)

A register points to a field containing the format name.

TOPOFF=*UNCHANGED / *IGNORE / <integer -255..255>

Defines the offset of the top margin of the print page from the top edge of the physical page in millimeters. First the print page is positioned on the paper and then the writing is rotated and positioned within the print page. This means that when the print page is moved around on the paper, the orientation of the text within it is ignored.

TOPOFF=*UNCHANGED

Operand default value. The current setting is not changed.

TOPOFF=*IGNORE

The printer controller does not set an offset between the print page and the top edge of the sheet.

TOPOFF=<integer -255..255>

The print page is shifted vertically by the specified offset on the paper.

LEFTOFF=*UNCHANGED / *IGNORE / <integer -255..255>

Defines the offset of the left margin of the print page from the left edge of the physical page in millimeters. First the print page is positioned on the paper and then the writing is rotated and positioned within the print page. This means that when the print page is moved around on the paper, the orientation of the text within it is ignored.

LEFTOFF=*UNCHANGED

Operand default value. The current setting is not changed.

LEFTOFF=*IGNORE

The printer controller does not set an offset between the print page and the left edge of the sheet.

LEFTOFF=<integer -255..255>

The print page is shifted horizontally by the specified offset on the paper.

SCHEDTIME=(*schdate*, *schtime*)

Specifies the scheduling time, i.e. the time at which the print job can be started at the earliest. The print job will be started from the specified point in time on, depending on printer availability and the processing regulations. If there are jobs with the status WFT, data transfer will be started from the specified time on. The operand may not be used in conjunction with UNIX systems.

schdate: *UNCHANGED / *TODAY / <var: char 1..10> / <c-string 10..10>

Specifies the point in time at which a print job will be started.

schdate: *UNCHANGED

Operand default value. The value remains unchanged.

schdate: *TODAY

The print job will be started today.

schdate: <var: char 1..10> / <c-string 10..10>

schtime: *UNCHANGED / *NOW / <var: char 1..5> / <c-string 5..5>

Specifies the point in time at which a print job will be started.

schtime: *UNCHANGED

Operand default value. The value remains unchanged.

schtime: *NOW

The print job will be started immediately.

schtime: <var: char 1..5> / <c-string 5..5>

Specifies the time in the form <hour>:<minutes>. The value consists of two unsigned integers separated by a colon. The values allowed are 0 to 23 for the hour and 0 to 59 for the minutes. A zero can precede one-position values.

Example:

SCHEDTIME = (SCHDATE=*TODAY,SCHTIME=*NOW)

No scheduling time was specified. The print job will be started in accordance with printer availability and the processing rules for print jobs.

Incompatible operands

If the VIRTUAL operand is set to *NOT_ALLOWED, the PROGRAM operand must not be assigned a value, otherwise an error is reported.

Return codes

(SC2)	SC1	Maincode	Meaning
00	00	0000	Processing completed without errors
00	00	0048	Warning: DELETE ignored for system file
00	01	0001	Invalid file name
00	01	0002	Invalid PL access
00	01	8000	Inconsistent parameter
00	01	0024	Invalid parameter value
00	01	0030	RECFORM=F in conjunction with RECSIZE=0 not permitted
00	01	0078	Printing of *SYSOUTPUT not permitted
00	01	007C	I-spac=ebcdic and RECFORM=U
00	01	0800	Printing of *SYSOUT not permitted
00	01	0090	Invalid or missing parameter for HP-HP90
00	01	0098	System file assigned to *DUMMY
00	01	009C	System file assigned to *TAPE
00	01	00A0	System file assigned to a temporary file
00	01	00A4	System file not assigned to SAM file
00	01	00DC	No PLAM library
00	01	FFFF	Interface version not supported
00	20	001C	Memory request error
00	20	0020	Memory release error
00	20	0004	Wait queue routine error
00	20	00D4	No information on the user job

(SC2)	SC1	Maincode	Meaning
00	20	00FC	Internal error
00	40	000C	DVS error
00	40	0010	EAM processing error
00	40	0018	Invalid or missing password
00	40	002C	Invalid file access
00	40	0034	Creation/expiration data not available
00	40	0040	EAM file not present
00	40	0044	System file does not contain any data
00	40	0054	DELETE-FILE command active
00	40	006C	Error while closing a system file
00	40	0070	File currently in use
00	40	0088	Output destination does not exist
01	40	8A00	File access not available
02	40	00b4	Form does not exist
03	40	00B8	SAM error in EAM processing
04	40	00BC	Rotation not supported
05	40	00C0	CHAR-SET-NUM or FOB size exceeded
06	40	00C4	CHAR-SET-NUM in pool exceeded
07	40	00C8	No ROTATION-LOOP defined
80	40	00CC	New TSN cannot be assigned
09	40	00E0	PLAM library not present
04	40	00E4	Library not found
05	40	00E8	MONJV error
06	40	00EC	Authorization error
07	40	00F0	Distributed processing not permitted
08	80	0094	SPOOL subsystem halted
09	80	FFFF	S&P subsystems not loaded

Notes

- The MODPJAT macro returns the above codes. The return codes from HDRCHECK and \$VALID may additionally be returned.
 If no data area was provided for the header, the value X'0000003C' is returned in register 15.
- The MODPJAT macro outputs information specific to the following subsystems and indicating whether the subsystem is loaded or not:
 - RSO
 - Dprint
 - SPS

Parameter list description

* para	meter	list description	
SMAPL	DS	0F	
SMAHDR		MF=(C,SMA),EQUATES=NO	Standard header
		n codes	
SMAOK	EQU	0	successful processing
SMAW048	EQU	72	warning
SMAE001	EQU	1	invalid filename
SMAE002	EQU	2	invalid pl access
	EQU	8	inconsistency between param
SMAE024	EQU	36	invalid parameter spec
SMAE030	EQU	48	recform=F not allowed with
*			recsize=0
SMAE078	EQU	120	print *sysout not allowed
SMAE07C	EQU	124	l-spac=ebcdic and recform U
SMAE080	EQU	128	fgg not allowed
SMAE090 *	EQU	144	invalid or missing parameters for HP-HP90
SMAE098	EQU	152	system file assigned to
*			*dummy
SMAE09C	EQU	156	system file assigned to tape
SMAE0A0	EQU	160	system file assigned to
*			temporary one
SMAE0A4	EQU	164	system file not assigned to
*			SAM file
SMAEODC	EQU	220	not a plam library
SMAWVER	EQU	65535	interface version not
*			supported
SMAE01C	EQU	28	memory request error
SMAE020	EQU	32	memory release error
SMAE004	EQU	4	queue routine error
	EQU	212	no information on user job
SMAEOFC	EQU	252	internal error
SMAE00C	EQU	12	DMS error
SMAE010	EQU	16	error in EAM processing
SMAE018	EQU	24	invalid or missing password
SMAE02C	EQU	44	invalid file access
SMAE034	EQU	52	creation-expiration date not
*			available
SMAE040	EQU	64	EAM file not available
SMAE044	EQU	68	syst file contains no data
SMAE054	EQU	84	DELETE-FILE command active
SMAE06C	EQU	108	syst file close error
	EQU	112	file currently in use
SMAE088	EQU	136	destination does not exist
SMAE0A8	EQU	168	file access not available
SMAE0B4	EQU	180	form does not exist

SMAE0B8	EQU	184	SAM error in EAM processing
SMAE0BC	EQU	188	no rotation supported
SMAEOCO *	EQU	192	exceeding char-set-num or FOB size
SMAE0C4	EQU	196	exceeding char-set-num in
SMAE0C8	EQU	200	<pre>pool no rotation-loop defined</pre>
SMAEOCC	EQU	204	new TSN can not be assigned
SMAE0CC SMAE0E0	EQU	224	PLAM library not available
SMAE0E4	EQU	228	library element not found
SMAE0E8	EQU	232	monjv error
SMAEOEC	EQU	236	authorization error
	EQU	240	distribution not allowed
SMAE094	EQU	148	SPOOL subsystem pended
SMAFFFF	EQU	65535	S&P subsystems not available
*	LQU	00000	Jai Subsystems not avairable
SMARSOP	DS	A	address of rso_pl
SMASPSP	DS	A	address of sps_pl
SMADPP	DS	Α	address of dprint_pl
SMASEMP	DS	A	address of dsem_pl
SMARES1	DS	A	
SMARES2	DS	Α	
SMARES3	DS	A	
SMARES4	DS	A	
SMATSN	DS	CL4	client or server TSN
SMACLUS	DS	CL8	cluster name
SMASERV	DS	CL8	server name
SMAMJV	DS	CL54	MONJV name
SMAUN1	DS	XL14	
SMAMOPT	DS	FL1	modify option
* _opt	ion_se	et	
SMAOPTS	EQU	1	option = tsn
SMAOPSR	EQU	2	option = server-tsn
SMAOPCL	EQU	3	option = client-tsn
SMA0PM0	EQU	4	option = monjv
*			
SMAUN2 *	DS	XL147	
SMASECR	DS	0XL64	input-section
SMASEID	DS	CL60	section-identifier
SMASEP0	DS	Н	position
SMASELE	DS	Χ	section-id length
SMASEIN	DS	AL1	sect_ind
SMASEGI	EQU	X.80,	sect-id given
SMASEIC	EQU	X'40'	character = true or
*			hexadecimal = false
SMASEWF	EQU	X'20'	sect-id = *whole-file
SMASEPS	EQU	X'10'	position = *std

```
SMASEST EQU
               X'08'
                                         on : sect-id is given in
                                          char. off : sect-id is given
                                          in hexa
SMASETG EQU
               X'04'
                                          section type has been given
SMASERF
         EQU
               X'03'
SMAFREC DS
               0XL72
                                          input-part(first-record)
SMASPOS DS
                                          record number
SMASSID DS
               CI 60
                                          string
SMASSPO DS
                                          string position
SMASTOC DS
               Н
                                          string occurrence
SMASSLE DS
                                          string length
               Χ
SMAFIL6 DS
               CL1
SMASSII DS
                                          strid_ind
SMASSIC EQU
               X,8000,
                                         character = true or
                                         hexadecimal = false
SMASSBO EQU
               X'4000'
                                         first-rec = *begin-of-file
               X'2000'
SMASSIT EQU
                                          first-rec = integer
SMASSBS FOU
               X'1000'
                                          first-rec = *by-string
SMASSPS EQU
               X,0800,
                                         position = *std
SMASSST EQU
               X'0400'
                                         on: first-rec given in char,
                                         off: first-rec is given in
                                         hexa
SMASSTG EQU
               X'0200'
                                          section type has been given
SMASSIF EQU
               X'01FF'
SMALREC DS
               0XI 72
                                          input-part(last-record)
SMAEPOS DS
               F
                                          record number
SMAFSID DS
               CI 60
                                          string
SMAESPO DS
               Н
                                          string position
SMAETOC DS
               Н
                                          string occurrence
SMAFSLE DS
               Χ
                                          string length
SMAFIL7 DS
               CL1
SMAESII DS
                                          strid ind
SMAFFIC FOU
               X'8000'
                                         character = true or
                                         hexadecimal = false
SMAESEF
         EQU
               X'4000'
                                          last-rec = *end-of-file
SMAFSIT FOU
               X'2000'
                                          last-rec = integer
SMAFSBS FOU
               X'1000'
                                         first-rec = *by-string
SMAESPS EQU
               X,0800,
                                         position = *std
SMAFSST FOU
               X'0400'
                                         on : last-rec given in char,
                                         off: last-rec is given in
                                         hexa
SMAFSTG FOU
               X'0200'
                                          section type has been given
SMAESIF EQU
               X'01FF'
```

```
SMARECP DS 0XL4
                                      record-part
SMARPFC DS
             Н
                                      first character
SMARPLC DS
            Н
                                      last character
SMALISP DS 0XL4
                                      line-spacing
SMACCPO DS
                                      control char position
SMASPAC DS
              FI1
                                      line spacing
* _space_set
SMASPA1 EOU 1
                                      spacing = *space_1
                                      spacing = *space 2
SMASPA2 EQU
              2
SMASPA3 EQU 4
                                      spacing = *space_3
SMASPAE EOU
                                      spacing = *by_ebcdic_control
SMASPAA EQU 16
                                      spacing = *by_asa_control
SMASPAI EQU 32
                                      spacing = *by_ibm_control
SMASPAN EOU 144
                                      spacing = *no
SMASPAS EQU 0
                                      spacing = *std
SMALSUN DS
              X I 1
                                      unused
SMADOCF DS
              FL1
                                      document format
* docform set
SMADCFT EQU 0
                                      docform = *text
SMADCFP EQU
              1
                                      docform = *page-format
SMADCFS EOU
                                      docform = *special-format
SMACONM DS
             0XI3
                                      control mode
SMACTLM DS
             FL1
                                      control mode
* _control_mode_set
SMACTNO EQU 0
                                      mode = *none
SMACTMS EQU 1
                                      mode = *std
SMACTPH EQU 2
                                      mode = *physical
SMACTMP EOU 4
                                      mode = *page-mode
                                      mode = *logical
SMACTML EQU 8
SMACTLI FOU 16
                                      mode = *line-mode
SMACTMA EOU 64
                                      mode = *apa
SMAPCC DS
              FI1
                                      page control character
* _pcc_set
SMATOPY EQU
            0
                                      pcc = *yes
SMATOPN FOU
                                      pcc = *no
SMATOPD EQU
                                      pcc = *dummy
SMACTLT DS FL1
                                      control type
* _control_type_set
SMACTDU EQU 0
                                      ctltype = *dummy
```

```
SMACTHC EQU
              1
                                        ctltype = *compatible
                                        ctltype = *hp
SMACTHP EQU
SMAUN4
        DS
              XL1
SMACHKP DS
              FL1
                                        checkpoint
* _checkp_set
SMACKPP EQU 0
                                        checkp = *on-page
SMACKPR EQU
              1
                                        checkp = *on-section-record
SMADOCN DS
                                        Number of files
SMAUN5
        DS
              XL5
SMALOCK DS
              FL1
                                        lock file
* _lockf_set
SMALOCS EQU
                                        lockf = *std
              0
SMALOCN EQU
                                        lockf = *no
SMALOCY FOU
                                        lockf = *yes
SMADELF DS
              FL1
                                        delete-after-print
* delf set
SMADELY EQU
              1
                                        delf = *yes
SMADELD EQU
              2
                                        delf = *destroy
SMADELN EQU
              3
                                        delf = *no
SMACOPY DS
                                        additional-copies
SMAUN6 DS
             CL58
SMAPRJN DS
             CL 8
                                        print-job-name
SMAPRIO DS
              Χ
                                        print-job-priority
SMAUN7 DS
              XI 24
SMATRNC DS
              FI1
                                        Line-truncation
* _truncation_set
SMATRUS EOU
                                        trunc = *std
             0
SMATRUD EQU
              1
                                        trunc = *delete file
SMATRUK FOU
                                        trunc = *keep_file
SMAPART DS
              0XI 12
                                        output-part
SMARCER DS
                                        from
SMARCTO DS
              F
                                        to
SMALAYD DS
              FL1
                                        dimension
* _layout_part_dim_set
SMALAYN EQU 0
                                        dim = *none
SMALAYP FOU
                                        dim = *pages
              1
SMALAYL EQU
              2
                                        dim = *line
SMAOPSW DS
              AL1
                                        sect_ind
SMAOIFB EQU
              X'80'
                                        sect-id given
              X'40'
SMAOITE EQU
                                        character = true or
```

```
hexadecimal = false
SMAOIFR EOU X'3F'
SMAFIL5 DS
             CL2
SMACLAS DS
                                       print-job-class
SMALFTM DS
              χ
                                       left-margin
SMALIPP DS
              Н
                                       Lines-per-page
SMAHDLI DS
              FL1
                                       header-line
* headline set
SMAHLNO EQU
              1
                                       headlin = *no
SMAHLST EQU
                                       headlin = *std
SMAHLR EQU
                                       headlin = *frecord
              4
SMAHLD EOU
              8
                                       headlin = *date
SMAHLDR EOU
             12
                                       headlin = *date frecord
SMAHLP EQU
            16
                                       headlin = *page
SMAHLRP EQU
              20
                                       headlin = *frecord page
SMAHLDP FOU
              24
                                       headlin = *date_page
SMAHDRP EQU
              28
                                       headlin = *date_frecord_page
SMAOUTF DS
             FI1
                                       output-format
* _outform_set
SMAOUTC EQU
                                       outform = *character
SMAOUTH EQU
                                       outform = *hexadecimal
SMATWSD DS
              FL1
                                       two-sided
* twoside set
SMATWOS EQU 0
                                       twoside = *std
              1
SMATWOY EQU
                                       twoside = *yes
SMATWOT EOU 2
                                       twoside = *tumble
SMATWON EQU
              4
                                       twoside = *no
SMATWOI FOU
                                       twoside = *ignore
        DS
SMAROT
              FL1
                                       rotation
   _rotation_set
                                       rot = *no
SMAROTN EQU 213
SMAROTO EOU
              240
                                       rot = *rot 0
SMAROT1 FOU
              241
                                       rot = *rot 90
SMAROT2 EOU 242
                                       rot = *rot 180
SMAROT3 EOU 243
                                       rot = *rot 270
SMAROT4 FOU 244
                                       rot = *rot_0_180
SMAROT5 FOU 245
                                       rot = *rot_90_270
SMAROT6 EOU 246
                                       rot = *rot 180 0
                                       rot = *rot_270_90
SMAROT7 EQU
              247
SMAROTY EQU
              232
                                       rot = *by_control_codes
SMAROTD EQU
              255
                                       rot = *no for modify
SMAIT#
        DS
              0XL1
                                       intray
        DS
SMAINT
              Χ
                                       input-tray-number
```

```
ORG
               SMAIT#
        DS
SMAITS
               FL1
                                         input-tray indicator
* _intray_set
SMAITBF EQU
               0
                                         intray = *by-format
SMAITIG EOU
               255
                                         intrav = *ignore
         ORG
               SMAIT#
SMAITFS DS
               FL1
                                         input-tray-format
    _intray_format_set
SMAITMA EQU
               128
                                         intray_format = *manual
                                         intray_format = *a3
SMAITA3 EOU
               129
SMAITA4 EQU
              130
                                         intray_format = *a4
SMAITA5 EQU
              131
                                         intray format = *a5
                                         intray_format = *b4
SMAITB4 EQU
              132
              133
                                         intray_format = *b5
SMAITB5 EQU
SMAITDB EQU
               134
                                         intray format =
                                         *double-letter
SMAITEX EQU
               135
                                         intray format = *exec
SMAITFO EQU
              136
                                         intray format = *folio
SMAITIV FOU
               137
                                         intray format = *invoice
              138
SMAITLG EQU
                                         intray_format = *legal
SMAITLT EQU
              139
                                         intray format = *letter
SMAITMO EQU
               140
                                         intray format = *monarch
SMAIT10 EQU
              141
                                         intray_format =
                                         *commercial-10
SMAITDL EOU
               142
                                         intray format = *d1
SMAITC5 EQU
              143
                                         intray_format = *c5
SMAIT3U EQU
               144
                                         intray format = *A3-uncut
SMAIT4U EOU
               145
                                         intray format = *A4-uncut
SMAITLD EQU
              146
                                         intray_format = *ledger
         ORG
               SMAIT#+1
         DS
SMAOT#
               0XL1
                                         outtray
SMAOUTT
        DS
                                         output-tray-number )
         ORG
               SMAOT#
        DS
SMAOTS
               FL1
                                         output-tray indicator
* _outtray_set
SMAOUTI EQU
                                         outtray = *ignore
               128
SMAOUTS EOU
               129
                                         outtray = *sorter
         ORG
               SMAOT#+1
SMAPGCP DS
               Н
                                         page-copies
SMACVPG DS
              0XL40
                                         cover-pages
SMAHTXT DS
              CL32
                                         header page text
               F
SMAHDEX DS
                                         header exit number
SMATREX DS
               F
                                         trailer exit number
```

SMAFORM SMALOOP SMARTLP SMACHS SMACHS#	DS DS DS DS EOU	CL6 CL3 CL3 16CL3	form-name loop-name rotation-loop name character-sets
SMAPLNM	DS	CL4	pool-name
SMAPLIN	DS	X	pool-index
SMAEF0	DS	CL2	electronic-overlay
*			
SMAOVLY	DS	0XL2	overlay
SMAOVFA	DS	X	face side
SMAOVRE *	DS	X	reverse side
* SMAFOB	DS	CL4	forms-overlay-buffer
SMAVIRT	DS	FL1	virtual-printer
	tual_s		VII baar printeer
SMAMUST	EQU	4	virtual = *must
SMAALLO	EQU	2	virtual = *allowed
SMANOTA	EQU	1	<pre>virtual = *not_allowed</pre>
SMASTDA	EQU	0	virtual = *std
*			
SMAFIL2	DS	CL2	
SMAPGPC	DS	F	page-definition
SMAUSRR	DS	CL44	user-resources-file
*	D.C	0.71.50	
SMATRTB	DS DS	OXL52 CL8	translation-table translation table name
SMATRNA SMATRFI	DS DS	CL44	translation table file
*	DЗ	CL44	cranstacton cable file
SMAPRNM	DS	CL8	printer-name
SMAPRTY	DS	FL1	printer-type
* _prt	ype_se	t	
SMAPTAN		0	prtype = *any
SMAPT65	EQU	1	prtype = *lp65_printer
SMAPTAP	EQU	2	prtype = *apa_printer
SMAPTHP	EQU	3	prtype = *hp_printer
*	5.0	=. 4	
SMASRTM	DS	FL1	sorter mode
	mode_s		cotmodo - tos
SMASMNO	EQU	1	srtmode = *no
SMASMGR		2 3	<pre>srtmode = *group srtmode = *collate</pre>
SMASMCL SMASMST	EQU EQU	4	srtmode = *stacker
SMASMAU	EQU	5	srtmode = *stacker srtmode = *automatic
*	LŲŪ	5	31 cilique — autolliatic

SMAFIL3	DS	CL2	
SMATSNO	DS	CL4	OUTPUT : returned tsn of the
*	ВЗ	OL I	modified job
SMAFIL4	DS	CL8	
SMASP1	DS	AL1	specified1
SMASP11	EQU	X'FO'	5,550
SMASP15	EQU	X.08.	section record
SMASP16	EQU	X'04'	first record
SMASP17	EQU	X'02'	last record
SMASP18	EQU	X'01'	record part
SMASP2	DS	AL1	specified2
SMASP21	EQU	X.80.	line spacing
SMASP22	EQU	X'40'	document format
SMASP23	EQU	X'20'	control mode
SMASP24	EQU	X'10'	additional copies
SMASP25	EQU	X.08.	familly
SMASP26	EQU	X'04'	checkpoint
SMASP27	EQU	X'02'	monjv
SMASP28	EQU	X'01'	jv-password
SMASP3	DS	AL1	specified3
SMASP31	EQU	X,80,	print-job-name
SMASP32	EQU	X'40'	print-job-priority
SMASP33	EQU	X'20'	processing admission
SMASP34	EQU	X'10'	line-truncation
SMASP35	EQU	X'08'	lock-file
SMASP36	EQU	X'04'	delete-after-print
SMASP37	EQU	X'02'	output part
SMASP38	EQU	X'01'	page-copies
SMASP4	DS	AL1	specified4
SMASP41	EQU	X,80,	left-margin
SMASP42	EQU	X'40'	line-per-page
SMASP43	EQU	X'20'	header-line
SMASP44	EQU	X'10'	output-format
SMASP45	EQU	X,08,	two-sided
SMASP46	EQU	X'04'	rotation
SMASP47	EQU	X'02'	input-tray
SMASP48	EQU	X'01'	output-tray
SMASP5	DS	AL1	specified5
SMASP51	EQU	X'80'	cover-page
SMASP52	EQU	X'40'	form-name
SMASP53	EQU	X'20'	loop-name_
SMASP54	EQU	X'10'	rotation-loop-name
SMASP55	EQU	X'08'	character-set
SMASP56	EQU	X'04'	pool-name
SMASP57	EQU	X'02'	pool-index
SMASP58	EQU	X'01'	electronic-form-overlay
SMASP6	DS	AL1	specified6
SMASP61	EQU	X,80,	overlay

SMASP62 SMASP63 SMASP64 SMASP65 SMASP66 SMASP67 SMASP71 SMASP71 SMASP72 SMASP73 SMASP74 SMASP74	EQU	X'40' X'20' X'10' X'08' X'04' X'02' X'01' AL1 X'80' X'40' X'20' X'10' X'0F'	fob lp65-pagedef user resources file translation table printer-name printer-type tsn specified7 rso parameters sps parameters dprint parameters dsem parameters
SMASP8	DS	AL1	specified8
SMASP81	EQU	X,80,	reserved for start-processing
SMASP82	EQU	X'40'	sorter mode virtual control
SMASP83 SMASP84	EQU EQU	X'20' X'10'	job-class
SMASP85	EQU	X,08,	prog. name
SMASP86	EQU	X'04'	string field
SMASP87	EQU	X,05,	format name
SMASP88	EQU	X'01'	scheduling time
*			
SMAVALL	DS	OXL40	virtual device synchrone
SMAPVNA	DS	CL8	virtual program name
SMAPVST *	DS	CL32	virtual program string
SMAFRMT	DS	CL63	format name
SMASPM1	DS	AL1	specified_m1
SMASPM11	EQU	X'80'	header exit number
SMASPM12	EQU	X'40'	trailer exit number
SMASPM13	EQU	X'20'	
SMASPM14	EQU	X'1F'	
SMASPM2	DS	AL1	specified_m2
SMASPM24		X'FF'	
SMASP9	DS	AL1	specified9
SMASP91	EQU	X,80,	top-offset
SMASP92	EQU	X'40'	left-offset
SMASP93	EQU	X'20'	scheduling time
SMASP9F	EQU	X'3F'	
SMATOPO *	DS	Н	Specifies the length of the white band that must be
*			inserted
^ SMALEFO	DS	Н	Specifies the width of the
*	טט	11	white band that must be
*			inserted

SMAUN9 DS XL11
SMADATE CL 10

SMATIME CL 5

SMA# EQU *-SMAHDR

unused specifies the date from which the job can be started specifies the time from which the job can be started

SHOPJAT - show print job attributes

User group: Nonprivileged users **Programming language:** Assembler

Macro type: C, D, E, L, M

This macro outputs the attributes of a print job. The macro corresponds to the SHOW-PRINT-JOB-ATTRIBUTES command.

Format

Format	
Operation	Operands
SHOPJAT	XPAND=PARAM / OUTPUT
	VARIANT= <u>001</u> / <c-string 33=""></c-string>
	CALLER= <u>*USER</u> / *SYSTEM
	TSN=(tsn1, cluste1) tsn1: <var: 4="" char:=""> / <c-string 44=""> cluste1: *LOCAL_CLUSTER / <var: 8="" char:=""> / <c-string 18:="" name_18=""></c-string></var:></c-string></var:>
	SERVTSN=(tsn2, server) tsn2: <var: 4="" char:=""> / <c-string 44=""> server: <var: 8="" char:=""> / <c-string 18:="" name_18=""></c-string></var:></c-string></var:>
	MONJV=*NONE / <var: 54="" char:=""> /</var:>
	FOREIGN=(identif, cluste2) identif: *NONE / <var: 4="" int:=""> / <integer 12147483647=""> cluste2: <var: 8="" char:=""> / <c-string 18:="" name_18=""></c-string></var:></integer></var:>
	BUFFER= <var: pointer=""></var:>
	BUFLEN= <var: 4="" int:=""> / <integer 18388607=""></integer></var:>
	INFO=list-poss(7): *ALL / *PRINTER_SELECTION /

Description of the operands

XPAND=PARAM / OUTPUT

Specifies which structure is to be expanded (created). Entries for this operand are ignored for other MF values.

XPAND=PARAM

The layout of the input parameter list is expanded

XPAND=OUTPUT

All parameter list descriptions (DSECTs) for describing the output information blocks are generated.

VARIANT=001 / <c-string 3..3>

Specifies the variant of the parameter list.

CALLER=*USER / *SYSTEM

Caller of the macro.

CALLER=*USER

Operand default value. A user calls the macro.

CALLER=*SYSTEM

The macro is called by the system.

Note

The TSN, SERVTSN, MONJV and FOREIGN operands are mutually exclusive and only one of them may be used at any one time.

TSN=(tsn1, cluste1)

Identifies the print job via its references on the client computer.

tsn1: <var: char: 4> / <c-string 4..4>

TSN of the job to be displayed.

tsn1: <var: char: 4>

The print job TSN is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field.

tsn1: <c-string 4..4>

The print job TSN is specified directly.

cluste1: *LOCAL CLUSTER / <var: char: 8> / <c-string 1..8: name 1..8>

Specifies the cluster in which the print job is processed. Only BS2000 clusters can be specified. The TSN is the local TSN at the gateway of the addressed BS2000 cluster.

cluste1: *LOCAL CLUSTER

Default value for the name of the cluster in which the print job is to be processed.

cluste1: <var: char: 8>

The cluster name is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field.

cluste1: <c-string 1..8: name_1..8>

The cluster name is specified directly.

SERVTSN=(tsn2, server)

Identifies the print job via its references on the server computer. Only print jobs in the local cluster can be addressed in this way.

tsn2: <var: char: 4> / <c-string 4..4: alphanum-name_4..4>

Task sequence number of the job to be displayed on the server.

tsn2: <var: char: 4>

The TSN of the print job is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field.

tsn2: <c-string 4..4: alphanum-name_4..4>

The TSN of the print job is specified directly.

server: <var: char: 8> / <c-string 1..8: name_1..8>

Name of the server on which the print job can be addressed by its TSN.

server: <var: char: 8>

The server computer name is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field.

server: <c-string 1..8: name_1..8>

The server computer name is specified directly.

MONJV=*NONE / <var: char: 54> / <c-string1..54: filename_1..54_with-catid>

This operand allows a print job to be addressed by its monitoring job variable, provided the specified MONJV is accessible on the host at which the command is issued.

MONJV=*NONE

Default value for the MONJV operand. No monitoring job variable was specified.

MONJV=<var: char: 54>

The MONJV name is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field.

MONJV=<c-string1..54: filename_1..54_with-catid>

The MONJV name is specified directly.

FOREIGN=(*identif*, *cluste2*):

The print job is addressed via a non-BS2000 identification.

identif: *NONE / <var: int: 4> / <integer 1..2147483647> Identification of the non-BS2000 reference to the print job.

identif: *NONE

Default value, no identification specified.

identif: <var: int: 4>

The identification is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field.

identif: <integer 1..2147483647>

The identification is specified directly.

cluste2: <var: char: 8> /<c-string 1..8: name_1..8>
Name of the cluster in which the print job is processed.

cluste2: <var: char: 8>

The cluster name is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field.

cluste2: <c-string 1..8: name_1..8>
The cluster name is specified directly.

BUFFER=<var: pointer>

Pointer to the output buffer.

BUFLEN=<var: int: 4> / <integer 1..8388607>

Output buffer size.

BUFLEN=<var: int: 4>

The output buffer size is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field.

BUFLEN=<var: int: 4> / <integer 1..8388607>

The output buffer size is specified directly.

INFO=list-poss(7): *ALL / *PRINTER_SELECTION / *DOCUMENT_DESCRIPTION / *PRINT_JOB_CONTROL / *LAYOUT_CONTROL / *RESOURCES_DESCRIPTION / *TO PRINTER

This operand allows you to select the information to be output.

Return codes

(SC2)	SC1	Maincode	Meaning
00	00	0000	No errors
02	00	0000	Print job not found or processing not allowed
01	01	FFFF	Invalid parameter
02	01	FFFF	Invalid address / length
03	01	FFFF	Output area too short
00	20	FFFF	System error
02	40	FFFF	No authorization
04	40	FFFF	Operand value error
05	40	FFFF	P/L version not supported
00	80	FFFF	Subsystem not loaded
01	80	FFFF	SPOOL not loaded
02	80	FFFF	RSO not loaded
03	80	FFFF	PLAM not loaded
04	80	FFFF	JV not loaded
05	80	FFFF	SPS not loaded
06	80	FFFF	DPRINTCL not loaded
07	80	FFFF	DPRINTCM not loaded
80	80	FFFF	DPRINTSV not loaded
09	80	FFFF	DSEM not loaded

Notes

- The SHOPJAT macro returns the above codes. The return codes from HDRCHECK and \$VALID may additionally be returned.
 If no data area was provided for the header, the value X'0000003C' is returned in register 15.
- The information on the print job is stored in an output buffer that must be made available by the user. If the buffer is defined too short, it is filled to the available length. The _main_header_model and _header_block_mdl sections are always written to the buffer. As long as they do not contain 0, the two 2-byte fields in the _main_header_model section contain the offset from the start of the output buffer to the relevant data block.

Length used	Total number of print jobs	Reported number of print jobs	Total number of PAM pages	Address of the first block
		· · · · · · · · · · · · · · · · · · ·		
Block	Block			
length	1			
Block length	Block 2			
Block length	Block 3			
	1			

- The SHOPJAT macro outputs information specific to the following subsystems and indicating whether the subsystem is loaded or not:
 - RSO
 - Dprint
 - SPS

Input parameter list description for the assembler programming language

<pre>* parameter list description</pre>				
SJA_PL_MDL DS OF	=			
SJAHDR FHDR MF=(C,SJA), EQUA	ATES=NO Standard header			
<pre>* main return codes</pre>				
SJARCNO EQU O	subcode1 = 00 : no error			
SJAWARN EQU 2	job not found or processing			
*	not permitted			
SJAPARM EQU 1	<pre>subcode1 = 01 : wrong</pre>			
*	parameter			
SJAVALD EQU 2	invalid address/length			
SJASHRT EQU 3	output area too short			
SJASYSE EQU 0	<pre>subcode1 = 20 : system error</pre>			
SJAPRIV EQU 2	subcode1 = 40 : no			
*	authorization			
SJAVERR EQU 4	operand value error			
SJAVERE EQU 5	p/l version not supported			
SJANOSS EQU 0	<pre>subcode1 = 80 : subsystem not</pre>			
*	loaded			

SJANSPO SJANRSO SJANPLM SJANJV SJANSPS SJANDCL SJANDCM SJANDSV SJANDSM	EQU EQU EQU EQU EQU EQU EQU EQU	1 2 3 4 5 6 7 8	spool not loaded rso not loaded plam not loaded jv not loaded sps not loaded dprintcl not loaded dprintcm not loaded dprintsv not loaded dsem not loaded
SJATSN SJACLUS SJASERV SJAMJV SJAUN2 SJAIDEN * SJABUF@ SJABUFL	DS DS DS DS DS DS	CL4 CL8 CL54 XL2 F A F	client or server TSN cluster name server name monjv name job identification on non-BS2000 cluster user buffer address user buffer length
SJAOPT * _option_set SJAOTSN SJAOSVT SJAOMJV SJAOFRG SJAOSYS *	EQU EQU EQU EQU EQU	0 1 2 3 4	option *TSN *SERVER-TSN *MONJV *FOREIGN *SYSFILE
SJAINFO SJADOPA SJADOFO SJAPRJC SJALACO SJAREDE SJATOPR SJAINFF SJAUN3 SJA#	DS EQU	AL1 X'80' X'40' X'20' X'10' X'08' X'04' X'03' XL14 *-SJAHDR	information document part info document format info print job control info layout control info resource description info to printer info

Output parameter list description for the assembler programming language

<pre>* _main_header_mode</pre>	1		
SJA_MAIN_HEADER_MODEL	DS	0F	
SJAHULE	DS	F	length used in the block
SJAHDRB	DS	F	displacement to header block
SJADOPB *	DS	F	displacement to document part block
SJADOFP	DS	F	displacement to document format block
SJAPJCB	DS	F	displacement to print job control block
SJALCOB	DS	F	displacement to layout
SJAREDB *	DS	F	displacement to resource description block
SJATOPB	DS	F	displacement to to printer
SJARSOB *	DS	F	displacement to RSO main
SJASPSB *	DS	F	displacement to SPS main
SJADSEB *	DS	F	displacement to DSEM main block
SJADPRB *	DS	F	displacement to DPRINT main block
SJAOUTPUT#	EQU	*-SJAHULE	

SHOPJST - show print job status

User group: Nonprivileged users **Programming language:** Assembler

Macro type: C, D, E, L, M

This macro outputs the status of a print job. The macro corresponds to the SHOW-PRINT-JOB-STATUS command.

Format

Operation	Operands
SHOPJST	XPAND=PARAM / OUTPUT
	,VARIANT= <u>001</u> / <c-string 33=""></c-string>
	,CALLER= <u>*USER</u> / *SYSTEM
	,RSOPAR=*NONE / <var: pointer=""></var:>
	,CLUSTER=*LOCAL_CLUSTER / <var: 8="" char:=""> /</var:>
	<pre>INFO = <var: _flag_set:1="" enum-of=""> / *ALL / *ORIGIN / *DESTINATION / *TRAITS / *DISTRIBUTED / *SUMMARY / *SPOOL-FILTER / *RSO-FILTER / default: *ORIGIN</var:></pre>
	,BUFFER= <var: pointer=""></var:>
	,BUFLEN= <var: 4="" int:=""> / <integer 18388607=""></integer></var:>
	,S_SPN_T= *ALL / <var: _flag_set:1="" enum-of=""> / *NONE / *WILD / *LIST /</var:>
	,S_SPNAM=*ALL / array(16): <var: 8="" char:=""> / <c-string 18=""></c-string></var:>
	,S_SPN_W=*ALL / <var: 24="" char:=""> / <c-string 124=""></c-string></var:>

(part 1 of 6)

Operation	Operands
SHOPJST	,S_USR_T= <u>*STD</u> / <var: _flag_set:1="" enum-of=""> / *LIST / *WILD / *OWN / *ALL</var:>
	,S_USER= <u>*ALL</u> / array(16): <var: 8="" char:=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	,S_USERW=*ALL / <var: 24="" char:=""> / <c-string 1-24="" 124:="" c-string=""></c-string></var:>
	,S_HST_T= <u>*HOME</u> / <var: _flag_set:1="" enum-of=""> / *LIST / *WILD / ALL</var:>
	,S_HOST= <u>*ALL</u> / array(16): <var: 8="" char:=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	,S_HOSTW=*ALL / <var: 24="" char:=""> / <c-string 1-24="" 124:="" c-string=""></c-string></var:>
	,S_SRV_T= <u>*STD</u> / <var: enum-of_flag_set:1="">/*LIST/*WILD/*STD/*ALL/ *HOME</var:>
	,S_SERV=*ALL / array(16): <var: 8="" char:=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	,S_SERVW=*ALL / <var: 24="" char:=""> / <c-string 1-24="" 124:="" c-string=""></c-string></var:>
	,S_FRM_T= <u>*ALL</u> / <var: _flag_set:1="" enum-of=""> / *LIST / *WILD</var:>
	,S_FORM= <u>*ALL</u> / array(16): <var: 6="" char:=""> / <c-string 1-6="" 16:="" c-string=""></c-string></var:>
	,S_FORMW= <u>*ALL</u> / <var: 24="" char:=""> / <c-string 1-24="" 124:="" c-string=""></c-string></var:>
	,S_TSN_T= <u>*ALL</u> / <var: _flag_set:1="" enum-of=""> / *LIST</var:>
	,S_TSN=*ALL / array(16): <var: 4="" char:=""> / <c-string 1-4="" 14:="" c-string=""></c-string></var:>
	,S_IDENT= <u>*ALL</u> / <var: 4="" int:=""> / <integer 12147483647=""></integer></var:>
	,S_SRT_T= <u>*ALL</u> / <var: 1="" _flag_set:="" enum-of=""> / *LIST</var:>
	,S_SVTSN=*ALL / array(16): <var: 4="" char:=""> / <c-string 1-4="" 14:="" c-string=""></c-string></var:>
	,S_OVL_T= <u>*ALL</u> / <var: _flag_set:1="" enum-of=""> / *LIST</var:>

(part 2 of 6)

Operation	Operands
SHOPJST	,S_OVER= <u>*ALL</u> / array(16): <var: 2="" char:=""> / <c-string 2-2="" 22:="" c-string=""></c-string></var:>
	,S_SPC_T= <u>*ALL</u> / <var: _flag_set:1="" enum-of=""> / *LIST</var:>
	,S_CLASS= <u>*ALL</u> / array(16): <var: 1="" int:=""> / <integer 1255=""></integer></var:>
	,S_ACC_T= <u>*ALL</u> / <var: 1="" _flag_set:="" enum-of=""> / *LIST / *NONE</var:>
	,S_ACC=*ALL / array(16): <var: 8="" char:=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	,S_FRT_T=*ALL / <var:enum-of flag_set:1="">/*LIST</var:enum-of>
	,S_FRT_N=*ALL / array(16): <var: 63="" char:=""> /</var:>
	,S_DESTY= <u>*ALL</u> / list-poss(8): *ALL / *DEVICE / *REMOTE / *MANAGED_DEV / *CENTRAL / *LOCAL / *DESTINATION
	,S_SELTY= <u>*MAY</u> / *MUST
	,S_SPDVT= <u>*ALL</u> / array(16): *ALL / *HP / *HP90 / *LP /
	,S_DEV_T= <u>*OWN</u> / <var: 1="" _flag_set:="" enum-of=""> / *LIST / *WILD / *ALL</var:>
	,S_DVNAM=*ALL / array(16): <var: 8="" char:=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	,S_DVNMW=*ALL / <var: 24="" char:=""> / <c-string 1-24="" 124:="" c-string=""></c-string></var:>
	,S_DES_T= <u>*ALL</u> / <var: _flag_set:1="" enum-of=""> / *LIST / *WILD</var:>
	,S_DSNAM=*ALL / array(16): <var: 8="" char:=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	,S_DSNMW= <u>*ALL</u> / <var: 24="" char:=""> / <c-string 1-24="" 124:="" c-string=""></c-string></var:>

(part 3 of 6)

Operation	Operands
SHOPJST	,S_ERR_T= <u>*ANY</u> / <var: _flag_set:1="" enum-of=""> / *LIST / *WILD / *ALL / *NONE</var:>
	,S_ERMSG = <u>*ANY</u> / array(16): <var: 18="" <c-string="" char=""> / *ALL / *NONE</var:>
	,S_ERRMW =*ANY / <var: 124="" <c-string="" char=""> / *ALL / *NONE</var:>
	,S_FOB=(type, low, high):
	,S_CSN=(type, low, high):
	,S_PRIO=(type, low, high):
	,S_ROT= <u>*ANY</u> / list-poss(4): *ANY / *YES / *NO
	,S_JTYP= <u>*ALL</u> / array(16): *ALL / *WAIT / *ACTIVE / *KEEP / *REPLAY /
	,S_VPR_T=*ALL / *LIST / <var: _flag_set:1="" enum-of=""></var:>
	,S_VPR_N=*ALL / <c-string 1-8="" 18:="" c-string=""></c-string>
	,E_SPN_T= <u>*NONE</u> / <var: _flag_set:1="" enum-of=""> / *LIST / *WILD</var:>
	,E_SPNAM=*NONE / array(16): <var: 8="" char:=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	,E_SPNAW=*NONE / <var: 24="" char:=""> / <c-string 1-24="" 124:="" c-string=""></c-string></var:>

(part 4 of 6)

Operation	Operands
SHOPJST	,E_USR_T=*NONE / <var: 1="" _flag_set:="" enum-of=""> / *LIST / *WILD</var:>
	,E_USER=*NONE / array(16): <var: 8="" char:=""> / <c-string 1-8="" 18="" c-string=""></c-string></var:>
	,E_USERW=*NONE / <var: 24="" char:=""> / <c-string 1-24="" 124:="" c-string=""></c-string></var:>
	,E_HST_T= <u>*NONE</u> / <var: 1="" _flag_set:="" enum-of=""> / *LIST / *WILD</var:>
	,E_HOST=*NONE / array(16): <var: 8="" char:=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	,E_HOSTW=*NONE / <var: 24="" char:=""> / <c-string 1-24="" 124:="" c-string=""></c-string></var:>
	,E_SRV_T= <u>*NONE</u> / <var: _flag_set:1="" enum-of=""> / *LIST / *WILD</var:>
	,E_SERV=*NONE / array(16): <var: 8="" char:=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	,E_SERVW=*NONE / <var: 24="" char:=""> / <c-string 1-24="" 124:="" c-string=""></c-string></var:>
	,E_FRM_T= <u>*NONE</u> / <var: _flag_set:1="" enum-of=""> / *LIST / *WILD</var:>
	,E_FORM=*NONE / array(16): <var: 6="" char:=""> <c-string 1-6="" 16:="" c-string=""></c-string></var:>
	,E_FORMW=*NONE / <var: 24="" char:=""> / <c-string 1-24="" 124:="" c-string=""></c-string></var:>
	,E_TSN_T= <u>*NONE</u> / <var: _flag_set:1="" enum-of=""> / *LIST</var:>
	,E_TSN=*NONE / array(16): <var: 4="" char:=""> / <c-string 1-4="" 14:="" c-string=""></c-string></var:>
	,E_SRT_T= <u>*NONE</u> / <var: _flag_set:1="" enum-of=""> / *LIST</var:>
	,E_SVTSN= <u>*NONE</u> / array(16): <var: 4="" char:=""> / <c-string 1-4="" 14:="" c-string=""></c-string></var:>
	,E_OVL_T= <u>*NONE</u> / <var: 1="" _flag_set:="" enum-of=""> *LIST</var:>
	,E_OVER=*NONE / array(16): <var: 2="" char:=""> / <c-string 2-2="" 22:="" c-string=""></c-string></var:>

(part 5 of 6)

Operation	Operands
SHOPJST	,E_SPC_T=*NONE / <var: 1="" _flag_set:="" enum-of=""> / *LIST</var:>
	,E_CLASS= <u>*NONE</u> / array(16): <var: 1="" int:=""> / <integer 1255=""></integer></var:>
	,E_ACC_T= <u>*NONE</u> / <var: 1="" _flag_set:="" enum-of=""> / *LIST</var:>
	,E_ACC=*NONE / array(16): <var: 8="" char:=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	,E_DESTY= <u>*NONE</u> / list-poss(4): *NONE / *DEVICE / *CENTRAL
	,E_DEV_T= <u>*NONE</u> / <var: 1="" _flag_set:="" enum-of=""> / *LIST / *WILD</var:>
	,E_DVNAM=*NONE / array(16): <var: 8="" char:=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	,E_DVNAW=*NONE / <var: 24="" char:=""> / <c-string 1-24="" 124:="" c-string=""></c-string></var:>
	,E_FRT_T= <u>*NONE</u> / <var: 1="" _flag_set:="" enum-of=""> / *LIST</var:>
	,E_FRT_N=*NONE / array(16): <var: 63="" char:=""> /</var:>
	E,_DES_T=*NONE / <var: 1="" _flag_set:="" enum-of=""> / *LIST / *WILD</var:>
	,E_DSNAM=*NONE / array(16): <var: 8="" char:=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	,E_DSNMW=*NONE / <var: 24="" char:=""> / <c-string 1-24="" 124:="" c-string=""></c-string></var:>
	,E_VPR_T= <u>*NONE</u> / <var: 1="" _flag_set:="" enum-of=""> / *LIST</var:>
	,E_VPR_N=*NONE / array(16): <var: 8="" char:=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>

(part 6 of 6)

Description of the operands

XPAND=PARAM / OUTPUT

Specifies which structure is to be expanded (created).

XPAND=PARAM

The layout of the input parameter list is expanded.

XPAND=OUTPUT

All data descriptions (DSECTs) for describing the output information blocks are generated.

VARIANT=<u>001</u> / <c-string 3..3>

Specifies the variant of the parameter list.

CALLER=*USER / *SYSTEM

Caller of the macro.

CALLER=*USER

Operand default value. A user calls the macro.

CALLER=*SYSTEM

The macro is called by the system.

RSOPAR=*NONE / <var: pointer>

The address of a parameter list containing RSO-specific values for calling the macro can be specified.

RSOPAR=*NONE

The values are not to be supplied via a parameter list.

RSOPAR=<var: pointer>

Defines a pointer, i.e. the contents of the variable or field is not the RSO parameter itself but rather the address of a memory location containing the RSO parameter list (A(field)). The macro is called by the system.

CLUSTER=*LOCAL_CLUSTER /<var: char: 8> / <c-string 1..8: name_1..8>

Name of the cluster containing the print jobs.

CLUSTER=*LOCAL CLUSTER

Operand default value. The print jobs are searched for in the local cluster.

CLUSTER=<var: char: 8>

The cluster name is passed in a field.

CLUSTER=<c-string 1..8: name_1..8>

The cluster name is specified directly.

INFO=*ORIGIN / <var: enum-of _info_set: 1> / *DESTINATION / *TRAITS / *SUMMARY / *DISTRIBUTED / *SPOOL-FILTER / *RSO-FILTER / *ALL

This operand allows you to select which information is to be output.

INFO=*ORIGIN

Outputs the following information on the origin of the spoolout jobs:

- TSN
- name of the server on which the job is processed
- user ID
- account number of the user who generated the spoolout job
- number of specified page copies
- number of specified file copies
- spoolout name of the job
- type and size of the file
- name of the client host
- TSN of the job issuing the job

This information is primarily of interest to the nonprivileged user.

INFO=<var: enum-of _info_set: 1>

The required information is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired information:

1	*ORIGIN
2	*DESTINATION
3	*TRAITS
4	*DEVICE_TYPE
5	*ALL
6	*SUMMARY
7	*DISTRIBUTED
8	*SPOOL-FILTER
9	*RSO-FILTER
See the parameter list description as of page 331.	

INFO=*DESTINATION

Outputs information on the output medium of the spoolout job:

- TSN
- name of the server on which the job is processed
- job type (RSO or local SPOOL)
- name of the device
- name of the device pool
- device types
- status and type of job (WAIT,ACTIVE,KEEP,...)
- RSO error code

This information is primarily of interest to the device administrator.

INFO=*TRAITS

Outputs information on the following attributes of the spoolout jobs:

- priority
- form name
- spoolout class
- number of character sets required
- size of the FOB data overlay required
- name of the film overlay required
- use of the page rotation module
- output control (CONTROL)
- name of the server on which the job is processed
- location of the resources

This information is primarily of interest to systems support, since it relates to the attributes (traits) of the local SPOOL devices.

INFO=*SUMMARY

Outputs the number of spoolout jobs selected and the number of PAM pages provided for these jobs.

INFO=*DISTRIBUTED

Global information on print jobs is output:

- TSN on the local and remote host (client and server TSN)
- name of the client host
- name of the server on which the job is processed
- job type (local, distributed, between clusters)
- information on the requesting user ID

INFO = *SPOOL-FILTER / *RSO-FILTER

Information on the destination devices of a print job can be displayed using these values. The device list contains all device types on which the print job can be processed after it has been filtered. Depending on the filters, the print jobs can be processed by SPOOL (if INFO=*SPOOL-FILTER was specified) or RSO (if INFO=*RSO-FILTER was specified). Both values can be specified in conjunction with INFO=DESTINATION, if the field ERMSG (error message) contains 'FILTER' or 'DEL/FLT'.

Output information:

- TSN
- Name of the server on which the print job is being processed
- Job type (RSO or local SPOOL)
- Device name
- Printer pool name
- Device type
- Status and type of job (WAIT,ACTIVE,KEEP,...)
- Error code
- Error message

Note

This information is basically of interest to the device administrator and the systems support staff (PRINT SERVICE ADMINISTRATION).

BUFFER=<var: pointer>

Pointer to the output buffer.

BUFLEN=<var: int: 4> / <integer 1..8388607>

Output buffer size.

BUFLEN=<var: int: 4>

The output buffer size is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field.

BUFLEN=<integer 1..8388607>

The output buffer size is specified directly.

S_SPN_T=*ALL / <var: enum-of _flag_set:1> / *NONE / *LIST / *WILD

Defines the type of entry for the print job spoolout name.

S_SPN_T=<var: enum-of _flag_set:1>

The type of entry for S_SPN_T and S_SPN_W is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions type:

0	*ALL/*ANY
1	*NONE
2	*LOCAL
3	*REMOTE
4	*CENTRAL
64	*DEVICE
128	*MANAG_DEV
5	*WILDCARD
6	*LIST
7	*STD
8	*OWN
9	*HOME
10	*IDENT
11	*ALL/*ANY
12	*ONLY
13	*RANGE
14	*ONE
15	*NO
16	*DESTINATION
See tl	ne parameter list description as of page 331.

S SPN T=*NONE

Only print jobs without spoolout names are selected.

S_SPN_T=<u>*ALL</u>

All print job spoolout names are selected.

S_SPN_T=*LIST

With *LIST, you define that the S_SPNAM operand contains a list of spoolout names.

S_SPN_T=*WILD

With *WILD, you define that the S_SPN_W operand contains a partially qualified name with wildcards. In this case, values in S_SPNAM create an error.

S_SPNAM=<u>*ALL</u> / array(16): <var: char: 8> / <c-string 1..8>

Spoolout names of the spoolout jobs on which information is to be output.

S SPNAM=*ALL

All spoolout names of the print jobs for which the information is to be output.

S_SPNAM=array(16): <var: char: 8>

List of spoolout names for the print jobs for which information is to be output.

S SPN T=*LIST must be set for this.

S_SPNAM=<c-string 1..8>

Spoolout name of the print job for which information is to be output.

S_SPN_W=*ALL / <var: char: 24> / <c-string 1..24>

Spoolout names of the print jobs for which information is to be output. The names are specified partially qualified with wildcards. S_SPN_T=*WILD must be set for this.

S_SPN_W=*ALL

All spoolout names of the print jobs for which the information is to be output.

S_SPN_W=<var: char: 24> / <c-string 1..24>

Spoolout names of the print jobs for which the information is to be output. The names are specified partially qualified with wildcards.

S_USR_T=<u>*STD</u> / <var: enum-of _flag_set:1> / *LIST / *WILD / *OWN / *ALL

Defines the type of entry for the user IDs of the print jobs.

S USR T=*STD

*STD means *ALL for operator tasks and *OWN for all other tasks.

S_USR_T=<var: enum-of _flag_set:1>

The type of entry for S_USER and S_USERW is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

S USR T=*LIST

With *LIST, you define that the S_USER operand contains a list of user IDs.

S_USR_T=*WILD

With *WILD, you define that the S_USERW operand contains a partially qualified name with wildcards. In this case, values in S_USER create an error.

S USR T=*OWN

Only user print jobs are selected.

S_USR_T=*ALL

All print jobs are selected.

S_USER=*ALL / array(16): <var: char: 8> / <c-string 1..8: c-string 1-8>

User IDs of the print jobs on which information is to be output.

S USER=*ALL

All user IDs of the print jobs for which information is to be output.

S_USER=array(16): <var: char: 8>

List of user IDs of the print jobs for which information is to be output.

S_USER=<c-string 1..8: c-string 1-8>

User ID of the print job for which information is to be output.

S_USERW=<u>*ALL</u> / <var: char: 24> / <c-string 1..24: c-string 1-24>

User IDs of the print jobs for which information is to be output. The names are specified partially qualified with wildcards.

S USERW=*ALL

All user IDs of the print jobs for which information is to be output.

S_USERW=<var: char: 24> / <c-string 1..24: c-string 1-24>

User IDs of the print jobs, partially qualified with wildcards, for which information is to be output.

S_HST_T=*HOME / <var: enum-of _flag_set: 1> / *LIST / *WILD / *ALL

Defines the type of entry for the name of the host computer.

S_HST_T=*HOME

With *HOME, you specify that the query applies to the local computer.

S_HST_T=<var: enum-of _flag_set: 1>

The type of entry for S_HOST and S_HOSTW is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions. See the table on page 301 for the relationships between the values and the desired types.

S_HST_T=*LIST

With *LIST, you define that the S_HOST operand contains a list of computer names.

S HST T=*WILD

With *WILD, you define that the S_HOSTW operand contains a partially qualified name with wildcards. In this case, values in S_HOST create an error.

S_HST_T=*ALL

All names of the host computers are selected.

S_HOST=*ALL / array(16): <var: char: 8> / <c-string 1..8: c-string 1-8>

Information is output on print jobs started from the specified hosts. Nonprivileged users do not receive any information if they specify a remote host; they can only obtain information on their own print jobs.

S_HOST=*ALL

All names of the host computer are selected.

S HOST=array(16): <var: char: 8>

Outputs information on print jobs started by the hosts specified in the list.

S HOST=<c-string 1..8: c-string 1-8>

Outputs information on print jobs started by the specified host.

S HOSTW=*ALL / <var: char: 24> / <c-string 1..24: c-string 1-24>

Information is output on print jobs started from the specified hosts. Nonprivileged users do not receive any information if they specify a remote host; they can only obtain information on their own print jobs.

The computer names are specified partially qualified with wildcards.

S HOSTW=*ALL

All host computer names are selected.

S_HOSTW=<var: char: 24> / <c-string 1..24: c-string 1-24>

The computer names are specified partially qualified with wildcards.

S_SRV_T=*HOME / <var: enum-of _flag_set: 1> / *LIST / *WILD / *ALL

Defines the type of entry for the names of the servers.

S_SRV_TYP=<u>*STD</u>/<var: enum-of _flag_set:1>/*LIST/*WILD/*ALL/*HOME

Defines the type of entry for the names of the servers.

S_SRV_TYP=*STD

*STD means *ALL for inter-cluster and *HOME for intra-cluster queries.

S_SRV_T=*HOME

With *HOME, you specify that the query applies to the local server.

S_SRV_T=<var: enum-of _flag_set: 1>

The type of entry for S_SERV and S_SERVW is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions. See the table on page 301 for the relationships between the values and the desired types.

S SRV T=*LIST

With *LIST, you define that the S_SERV operand contains a list of computer names.

S_SRV_T=*WILD

With *WILD, you define that the S_SERVW operand contains a partially qualified name with wildcards. In this case, values in S_SERV create an error.

S SRV T=*ALL

All server computer names are selected.

S_SERV=*ALL / array(16): <var: char: 8> / <c-string 1..8: c-string 1-8>

Information is output on print jobs processed on the specified server.

S_SERV=*ALL

All print jobs on all available servers are selected.

S_SERV=array(16): <var: char: 8>

Information is output on print jobs processed on the servers in the list.

S SERV=<c-string 1..8: c-string 1-8>

Information is output on print jobs processed on the specified server.

S SERVW=*ALL / <var: char: 24> / <c-string 1..24: c-string 1-24>

Information is output on print jobs processed on the specified server. The names are specified partially qualified with wildcards.

S_FRM_T=*ALL / <var: enum-of _flag_set: 1> / *LIST / *WILD

Defines the type of entry for the form names.

With *LIST, you define that the S_FORM operand contains a list of form names.

With *WILD, you define that the S_FORMW operand contains a partially qualified name with wildcards. In this case, values in in S_FORM create an error.

S_FRM_T=*ALL

All form names are selected.

S_FRM_T=<var: enum-of _flag_set: 1>

The type of entry for S_FORM and S_FORMW is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions. See the table on page 301 for the relationships between the values and the desired types.

S FRM T=*LIST

With *LIST, you define that the S_FORM operand contains a list of form names.

S FRM T=*WILD

With *WILD, you define that the S_FORMW operand contains a partially qualified name with wildcards. In this case, values in S_FORM create an error.

S_FORM=<u>*ALL</u> / array(16): <var: char: 6> / <c-string 1..6: c-string 1-6>

Form names of the spoolout jobs on which information is to be output.

S_FORM=<u>*ALL</u>

All form names are selected.

S_FORM=array(16): <var: char: 6>

List of the form names for the print jobs for which the information is to be output.

S_FORM=<c-string 1..6: c-string 1-6>

Form name of the spoolout jobs on which information is to be output.

S FORMW=*ALL / <var: char: 24> / <c-string 1..24: c-string 1-24>

Form names of the spoolout jobs on which information is to be output. The names are specified partially qualified with wildcards.

S FORMW=*ALL

All form names are selected.

S_FORMW=<var: char: 24> / <c-string 1..24: c-string 1-24>

Partially qualified form names with wildcards for print jobs about which the information is to be output.

S_TSN_T=*ALL / <var: enum-of _flag_set: 1> / *LIST

Defines the type of entry for the print job numbers.

S_TSN_T=*ALL

All job numbers are selected.

S_TSN_T=<var: enum-of _flag_set: 1>

The type of entry for S_TSN is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions. See the table on page 301 for the relationships between the values and the desired types.

S TSN T=*LIST

With *LIST, you define that the S_TSN operand contains a list of job numbers.

S_TSN=*ALL / array(16): <var: char: 4> / <c-string 1..4: c-string 1-4>

Task sequence numbers of the spoolout jobs on which information is to be output.

S TSN=*ALL

All job numbers are selected.

S_TSN=array(16): <var: char: 4>

List of job numbers for print jobs on which information is to be output.

S_TSN=<c-string 1..4: c-string 1-4>

Task sequence number of the spoolout job on which information is to be output.

S_IDENT=*ALL / <var: int: 4> / <integer 1..2147483647>

Print jobs in UNIX systems can be addressed via their identification. An Xprint domain in a UNIX system must be selected in this case. Refer to the manual "Distributed Print Services (BS2000/OSD)" for details.

S IDENT=*ALL

All print jobs of the UNIX system are addressed.

S_IDENT=<var: int: 4> / <integer 1..2147483647>

Print jobs of a UNIX system are addressed via their identification. An Xprint domain in a UNIX system must be selected in this case.

S SRT T=*ALL / <var: enum-of flag set: 1> / *LIST

Defines the type of entry for the server TSN of the print jobs.

S_SRT_T=*ALL

All server TSNs are selected.

S_SRT_T=<var: enum-of _flag_set: 1>

The type of entry for S_SVTSN is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions. See the table on page 301 for the relationships between the values and the desired types.

S SRT T=*LIST

With *LIST, you define that the S_SVTSN operand contains a list of server TSNs.

S_SVTSN=*ALL / array(16): <var: char: 4> / <c-string 1..4: c-string 1-4> Server TSN of the print jobs for which information is to be output.

S SVTSN=*ALL

All server TSNs are selected.

S_SVTSN=array(16): <var: char: 4>

List of server TSNs of the print jobs for which information is to be output.

S_SVTSN=<c-string 1..4: c-string 1-4>

Server TSN of the print jobs for which information is to be output.

S_OVL_T=*ALL / <var: enum-of _flag_set: 1> / *LIST

Defines the type of entry for the film overlays of the print jobs.

S_OVL_T=*ALL

All film overlays are selected.

S_OVL_T=<var: enum-of _flag_set: 1>

The type of entry for S_OVER is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

S_OVL_T=*LIST

With *LIST, you define that the S_OVER operand contains a list of film overlays.

S_OVER=*ALL / array(16): <var: char: 2> / <c-string 2..2: c-string 2-2>

Information is to be output on spoolout jobs for which the named physical forms overlays have been specified.

S OVER=*ALL

All film overlays are selected.

S OVER=array(16): <var: char: 2>

Information on print jobs for the film overlays specified in the list is to be output.

S OVER=<c-string 2..2: c-string 2-2>

Information on print jobs for the specified film overlay is to be output.

S SPC T=*ALL / <var: enum-of flag set: 1> / *LIST

Defines the type of entry for the spoolout classes of the print jobs.

S SPC T=*ALL

All spoolout classes are selected.

S_SPC_T=<var: enum-of _flag_set: 1>

The type of entry for S_CLASS is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

S SPC T=*LIST

With *LIST, you define that the S_CLASS operand contains a list of film overlay spoolout classes.

S_CLASS=*ALL / array(16): <var: int: 1> / <integer 1..255>

Spoolout classes of the print jobs for which information is to be output.

S CLASS=*ALL

All spoolout classes are selected.

S_CLASS=array(16): <var: int: 1>

List of spoolout classes of the print jobs for which information is to be output.

S_CLASS=<integer 1..255>

Spoolout class of the print jobs for which information is to be output.

S_ACC_T=*ALL / <var: enum-of _flag_set: 1> / *LIST / *NONE

Defines the type of entry for the print job account numbers.

S_ACC_T=*ALL

All account numbers are selected.

S_ACC_T=<var: enum-of _flag_set: 1>

The type of entry for S_ACC is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

S ACC T=*LIST

With *LIST, you define that the S_ACC operand contains a list of account numbers.

S_ACC_T=*NONE

No account numbers are selected.

S_ACC=*ALL / array(16): <var: char: 8> / <c-string 1..8: c-string 1-8>

Account numbers of the spoolout jobs on which information is to be output.

S_ACC=*ALL

All account numbers are selected.

S_ACC=array(16): <var: char: 8>

List of account numbers for print jobs on which information is to be output.

S ACC=<c-string 1..8: c-string 1-8>

Account number of the spoolout jobs on which information is to be output.

S_FRT_T=*ALL / <var: enum-of _flag_set: 1> / *LIST

Defines the type of entry for the format names of the print jobs.

S FRT T=*ALL

All format names are selected.

S_FRT_T=<var: enum-of _flag_set: 1>

The type of entry for S_FRT is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

S FRT T=*LIST

With *LIST, you define that the S_FRT_N operand contains a list of format names.

S_FRT_N=*ALL / array(16): <var: char: 63> / <c-string 1..63: c-string 1-63>

Print jobs whose document format names are on the specified list are included in the selection.

S_FRT_N=*ALL

All format names are selected.

S_FRT_N=array(16): <var: char : 63>

List of format names for the print jobs on which information is to be output.

S_FRT_N=<c-string 1..63: c-string 1-63>

Format name of the print job on which information is to be output.

S_DESTY=*ALL / list-poss(8): *ALL / *DEVICE / *REMOTE / *MANAGED_DEV / *CENTRAL / *LOCAL / *DESTINATION

Information is to be output on the print jobs processed on the specified output media. Information can be requested on:

- all jobs (*ALL)
- jobs for local SPOOL devices (*LOCAL)
- jobs for RSO devices (*REMOTE)
- jobs whose destination is the central printer pool (*CENTRAL)
- jobs for devices managed by the user (*MANAGED-DEVICES)
- jobs for specific devices (*DEVICE)

S SELTY=*MAY / list-poss(3): *MAY / *MUST

Selection of local spoolout jobs is restricted to those that may be output on at least one of the specified device types (*MAY) or to those that must be output on one of the specified device types and must not be output on any device type other than the ones specified (*MUST).

S_SPDVT=*ALL / array(16): *ALL / *HP / *HP90 / *LP / *LP_EMUL / *LP48 / *LP65 / *TAPE / *2050_APA / *2090_APA / *2090_TWIN / *VIRTUAL

Device types for local spoolout jobs on which information is requested.

S_DEV_T=*OWN/ <var: enum-of _flag_set: 1> / *LIST / *WILD / *ALL

Defines the type of entry for the device names.

S DEV T=*OWN

Only user device names are specified.

S_DEV_T=<var: enum-of _flag_set: 1>

The type of entry for S_DVNAM and S_DVNMW is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

S DEV T=*LIST

A list of device names is specified.

S DEV T=*WILD

The device names are specified partially qualified with wildcards.

S_DEV_T=*ALL

All device names are valid.

S_DVNAM=*ALL / array(16): <var: char: 8> / <c-string 1..8: c-string 1-8>

Names of the devices on which information is requested.

S_DVNAM=*ALL

Information is requested on all devices.

S_DVNAM=array(16): <var: char: 8>

List of device names on which information is requested.

S_DVNAM=<c-string 1..8: c-string 1-8>

Name of the device on which information is requested.

S_DVNMW=*ALL/ <var: char: 24> / <c-string 1..24: c-string 1-24>

Names of the devices on which information is requested. The names are specified partially qualified with wildcards.

S DVNMW=*ALL

Information is requested on all devices.

S_DVNMW=<var: char: 24> / <c-string 1..24: c-string 1-24>

Names of the devices on which information is requested. The names are specified partially qualified with wildcards.

S_DES_T=*ALL / <var: enum-of _flag_set: 1> / *LIST / *WILD

Defines the type of entry for the device names.

S DES T=*ALL

All device names are valid.

S_DES_T=<var: enum-of _flag_set: 1>

The type of entry for S_DSNAM and S_DSNMW is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

S DES T=*LIST

A list of device names is specified.

S DES T=*WILD

The device names are specified partially qualified with wildcards.

S_DSNAM=*ALL / array(16): <var: char: 8> / <c-string 1..8: c-string 1-8>

Names of the device pools on which information is requested.

S DSNAM=*ALL

Information is requested on all device pools.

S_DSNAM=array(16): <var: char: 8>

List of names of device pools on which information is requested.

S_DSNAM=<c-string 1..8: c-string 1-8>

Name of the device pool on which information is requested.

S_DSNMW=*ALL / <var: char: 24> / <c-string 1..24: c-string 1-24>

Names of the device pools on which information is requested. The names are specified partially qualified with wildcards.

S_DSNMW=*ALL

Information is requested on all device pools.

S_DSNMW=<var: char: 24> / <c-string 1..24: c-string 1-24>

Names of the device pool on which information is requested. The names are specified partially qualified with wildcards.

S_ERR_T=*ANY / <var: enum-of_FLAG_SET:1 > / *LIST / *WILD / *ALL / *NONE

Specifies the type of entry for selecting the error message operand.

S_ERR_T=*ANY

All print jobs are selected, regardless of whether they have an error message or not.

S ERR T=*WILD

Specifies that the operand S_ERRMW is partially qualified by wildcards. If a value is entered in the field S-ERMSG, an error occurs.

S ERR T=*LIST

Specifies that S ERMSG contains a list of error messages.

S ERR T=*ALL

All current print jobs with an error message are selected.

S ERR T=*NONE

All current print jobs with an error message are selected.

S_ERMSG=*ANY / *ALL / *NONE / array(16) <var: char 1..8> / <c-string 1..8>

Selects all print jobs affected by the specified error messages.

S ERMSG=*ANY

All print jobs are selected, regardles of whether they are affected by an error message or not.

S ERMSG=*ALL

All current print jobs with an error message are selected.

S ERMSG=*NONE

All current print jobs without an error message are selected.

S_ERMSG=array(16) <var: char 1..8> / <c-string 1..8>

Selects the print jobs affected by the error messages specified in the list.

S_ERRMW=*ANY/<var: char 1..24>/<c-string 1..24>/*ALL/*NONE

Selects the print jobs affected by the specified error messages. The error messages can be partially qualified using wildcards, however, S_ERR_T=*WILD must then also be specified.

S ERRMW=*ANY

All print jobs are selected, regardless of whether they are affected by an error message or not.

S ERRMW=*ALL

Selects all current print jobs with an error message.

S ERRMW=*NONE

Selects all current print jobs without an error message.

S_ERRMW=<var: char 1..24>/<c-string 1..24>

Selects error messages concerning the print job. The error messages may be partially qualified using wildcards, however, S_ERR_T=*WILD must then also be specified.

$S_FOB=(type, low, high)$:

Information is to be output on print jobs for which

- any FOB data overlay was specified (*ONLY)
- no FOB data overlay was specified (*NO)
- FOB data overlays within the specified size range were specified (*RANGE).

The operand is ignored as a selection criterion if *ANY is specified.

type: *ANY / *ONLY / *NO / *RANGE

Defines the type of entry for selecting an FOB data overlay.

low: *MIN / <var: int: 2> / <integer 0..32767>

Defines the minimum size of the FOB data overlay. *MIN corresponds to the value 0.

high: *MAX / <var: int: 2> / <integer 0..32767>

Defines the maximum size of the FOB data overlay. *MAX corresponds to the value 32767.

S_CSN=(type, low, high):

Information is to be output about the print jobs

- which only need one character set (*ONE)
- whose number of required character sets is within the specified range (*RANGE).

type: *ALL / *ONE / *RANGE

Defines the type of entry for selecting the character set.

low: *MIN / <var: int: 2> / <integer 1..32767>

Defines the minimum number of character sets. *MIN corresponds to the value 1.

high: *MAX / <var: int: 2> / <integer 1..32767>

Defines the maximum number of character sets. *MAX corresponds to the value 32767.

S_PRIO=(type, low, high):

Information is to be output on print jobs whose priority is within the specified range (*RANGE). The operand is ignored as a selection criterion if *ANY is specified.

type: *ALL / *RANGE

Defines the type of entry for the priority.

low: *MIN / <var: int: 1> / <integer 30..255>

Defines the lowest priority value.

high: *MAX / <var: int: 1> / <integer 30..255>

Defines the highest priority value.

S_ROT=*ANY / list-poss(4): *ANY / *YES / *NO

Information is to be output on print jobs for which

- rotation was requested in PRINT-DOCUMENT (*YES),
- no rotation was requested in PRINT-DOCUMENT (*NO)

The operand is ignored as a selection criterion if *ANY is specified.

S_JTYP=<u>*ALL</u> / array(16): *ALL / *WAIT / *ACTIVE / *KEEP / *REPLAY / *WAIT_PRE / *PREPROC / *BEF APA PR / *AFT APA PR / *WAIT FT / *FT

Information is to be output on the following types of spoolout job:

- waiting spoolout jobs: *WAIT
- active spoolout jobs: *ACTIVE
- suspended spoolout jobs: *KEEP
- spoolout jobs to be output to replay tape: *REPLAY
- waiting SECTION spoolout jobs (output section by section):
 *WAIT-PREPROCESSING
- active SECTION spoolout jobs (output section by section):
 *PREPROCESSING
- *BEFORE-APA (APA printers: job status = "TRANSFERRED", see also the "SPOOL (BS2000/OSD)" manual)
- *AFTER-APA (APA printers: job status = "TRANSIT", see also the "SPOOL (BS2000/OSD)" manual)
- jobs waiting for file transfer: *WAIT-FILE-TRANSFER
- jobs currently undergoing file transfer: *FILE-TRANSFER

E_SPN_T=*NONE / <var: enum-of _flag_set: 1> / *LIST / *WILD

Defines the type of entry for the spoolout names of the print jobs which are to be excluded from information output.

If you specify *NONE, the operand does not act as a selection criterion.

E SPN T=*NONE

No spoolout names are excluded from the information output.

E_SPN_T=<var: enum-of _flag_set: 1>

The type of entry for E_SPNAM and E_SPNAW is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

E SPN T=*LIST

With *LIST, you define that the E SPNAM operand contains a list of spoolout names.

E_SPN_T=*WILD

The spoolout names are specified partially qualified with wildcards.

E_SPNAM=*NONE / array(16): <var: char: 8> / <c-string 1..8: c-string 1-8>

Spoolout names of the spoolout jobs to be excluded from the information output.

If you specify *NONE, the operand does not act as a selection criterion.

E SPNAM=*NONE

No spoolout names are excluded from the information output.

E_SPNAM=array(16): <var: char: 8>

List of spoolout names for which no information is to be output. S_SPN_T=*LIST must be set for this.

E SPNAM=<c-string 1..8: c-string 1-8>

Spoolout name to be excluded from the information output.

E_SPNAW=*NONE / <var: char: 24> / <c-string 1..24: c-string 1-24>

Spoolout names of print jobs to be excluded from the information output. The names are specified partially qualified with wildcards.

If you specify *NONE, the operand does not act as a selection criterion.

E SPNAW=*NONE

No spoolout names are excluded from the information output.

E SPNAW=<var: char: 24> / <c-string 1..24: c-string 1-24>

The spoolout names are specified partially qualified with wildcards.

E_USR_T=*NONE / <var: enum-of _flag_set: 1> / *LIST / *WILD

Defines the type of entry for the user IDs of print jobs to be excluded from the information output.

If you specify *NONE, the operand does not act as a selection criterion.

E USR T=*NONE

No user IDs are excluded from the information output.

E_USR_T=<var: enum-of _flag_set: 1>

The type of entry for E_USER and E_USERW is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

E USR T=*LIST

With *LIST, you define that the E-USER operand contains a list of user IDs.

E USR T=*WILD

The user IDs are specified partially qualified with wildcards.

E_USER=*NONE / array(16): <var: char: 8> / <c-string 1..8: c-string 1-8>

User IDs of the print jobs to be excluded from the information output.

If you specify *NONE, the operand does not act as a selection criterion.

E USER=*NONE

No user IDs are excluded from the information output.

E_USER=array(16): <var: char: 8>

List of spoolout names for the print jobs for which no information is to be output.

E_USER_T=*LIST must be set for this.

E_USER=<c-string 1..8: c-string 1-8>

User ID to be excluded from the information output.

E_USERW=*NONE / <var: char: 24> / <c-string 1..24: c-string 1-24>

User IDs of print jobs to be excluded from the information output. The user IDs are specified partially qualified with wildcards.

If you specify *NONE, the operand does not act as a selection criterion.

E USERW=*NONE

No user IDs are excluded from the information output.

E USERW=<var: char: 24> / <c-string 1..24: c-string 1-24>

The user IDs are specified partially qualified with wildcards.

E_HST_T=*NONE / <var: enum-of _flag_set: 1> / *LIST / *WILD

Defines the type of entry for hosts from which print jobs are started which are to be excluded from the information output. If you specify *NONE, the operand does not act as a selection criterion.

E_HST_T=*NONE

No hosts are excluded from the information output.

E_HST_T=<var: enum-of _flag_set: 1>

The type of entry for E_HOST and E_HOSTW is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

E HST T=*LIST

With *LIST, you define that the E_HOST operand contains a list of user IDs.

E HST T=*WILD

The hosts are specified partially qualified with wildcards.

E_HOST=*NONE / array(16): <var: char: 8> / <c-string 1..8: c-string 1-8>

Print jobs started from the specified hosts are excluded from the information output. The specified hosts must belong to the local cluster.

If you specify *NONE, the operand does not act as a selection criterion.

E HOST=*NONE

No hosts are excluded from the information output.

E_HOST=array(16): <var: char: 8>

List of spoolout hosts for which no information is to be output. E_HST_T=*LIST must be set for this.

E_HOST=<c-string 1..8: c-string 1-8>

Host which is to be excluded from the information output.

E_HOSTW=*NONE / <var: char: 24> / <c-string 1..24: c-string 1-24>

Print jobs started from the specified hosts are excluded from the information output. The specified hosts must belong to the local cluster. The host names are specified partially qualified with wildcards.

If you specify *NONE, the operand does not act as a selection criterion.

E HOSTW=*NONE

No hosts are excluded from the information output.

E_HOSTW=<var: char: 24> / <c-string 1..24: c-string 1-24>

The hosts are specified partially qualified with wildcards.

E_SRV_T=*NONE / <var: enum-of _flag_set: 1> / *LIST / *WILD

Defines the type of entry for servers from which print jobs are to be processed which are to be excluded from the information output.

If you specify *NONE, the operand does not act as a selection criterion.

E SRV T=*NONE

No servers are excluded from the information output.

E_SRV_T=<var: enum-of _flag_set: 1>

The type of entry for E_SERV and E_SERVW is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

E_SRV_T=*LIST

With *LIST, you define that the E_SERV operand contains a list of user IDs.

E_SRV_T=*WILD

The servers are specified partially qualified with wildcards.

E_SERV=*NONE / array(16): <var: char: 8> / <c-string 1..8: c-string 1-8>

Print jobs to be processed on the specified server are excluded from the information output. If you specify *NONE, the operand does not act as a selection criterion.

E_SERV=*NONE

No servers are excluded from the information output.

E_SERV=array(16): <var: char: 8>

List of spoolout names for the print jobs for which no information is to be output.

E_SRV_T=*LIST must be set for this.

E_SERV=<c-string 1..8: c-string 1-8>

Server which is to be excluded from the information output.

E_SERVW=*NONE / <var: char: 24> / <c-string 1..24: c-string 1-24>

Print jobs to be processed on the specified server are excluded from the information output.

The server names are specified partially qualified with wildcards.

If you specify *NONE, the operand does not act as a selection criterion.

E SERVW=*NONE

No servers are excluded from the information output.

E_SERVW=<var: char: 24> / <c-string 1..24: c-string 1-24>

The servers are specified partially qualified with wildcards.

E FRM T=*NONE/<var: enum-of flag set: 1>/*LIST/*WILD

Defines the type of entry for form names of print jobs which are to be excluded from the information output.

If you specify *NONE, the operand does not act as a selection criterion.

E FRM T=*NONE

No form names are excluded from the information output.

E_FRM_T=<var: enum-of _flag_set: 1>

The type of entry for E_FORM and E_FORMW is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

E FRM T=*LIST

With *LIST, you define that the E FORM operand contains a list of user IDs.

E FRM T=*WILD

The form names are specified partially qualified with wildcards.

E_FORM=*NONE / array(16): <var: char: 6> / <c-string 1..6: c-string 1-6>

Form names of the spoolout jobs to be excluded from the information output. If you specify *NONE, the operand does not act as a selection criterion.

E FORM=*NONE

No form names are excluded from the information output.

E_FORM=array(16): <var: char: 6>

List of spoolout names for the print jobs for which no information is to be output.

E FRM T=*LIST must be set for this.

E_FORM=<c-string 1..6: c-string 1-6>

Form name which is to be excluded from the information output.

E_FORMW=*NONE / <var: char: 24> / <c-string 1..24: c-string 1-24>

Form names of the spoolout jobs to be excluded from the information output. The form names are specified partially qualified with wildcards.

If you specify *NONE, the operand does not act as a selection criterion.

E FORMW=*NONE

No form names are excluded from the information output.

E_FORMW=<var: char: 24> / <c-string 1..24: c-string 1-24>

The form names are specified partially qualified with wildcards.

E_TSN_T=*NONE / <var: enum-of _flag_set: 1> / *LIST

Defines the type of entry for job numbers of print jobs which are to be excluded from the information output.

If you specify *NONE, the operand does not act as a selection criterion.

E TSN T=*NONE

No job numbers are excluded from the information output.

E_TSN_T=<var: enum-of _flag_set: 1>

The type of entry for E_TSN is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

E TSN T=*LIST

With *LIST, you define that the E_TSN operand contains a list of user IDs.

E_TSN=*NONE / array(16): <var: char: 4> / <c-string 1..4: c-string 1-4>

Task sequence numbers of the spoolout jobs to be excluded from the information output. If you specify *NONE, the operand does not act as a selection criterion.

E TSN=*NONE

No job numbers are excluded from the information output.

E_TSN=array(16): <var: char: 4>

List of spoolout names for the print jobs for which no information is to be output. E TSN T=*LIST must be set for this.

E_TSN=<c-string 1..4: c-string 1-4>

Job number which is to be excluded from the information output.

E_SRT_T=*NONE / <var: enum-of _flag_set: 1> / *LIST

Defines the type of entry for the server TSN.

If you specify *NONE, the operand does not act as a selection criterion.

E_SRT_T=*NONE

No server TSNs are excluded from the information output.

E_SRT_T=<var: enum-of _flag_set: 1>

The type of entry for E_SVTSN is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

E_SRT_T=*LIST

With *LIST, you define that the E_SVTSN operand contains a list of user IDs.

E_SVTSN=*NONE / array(16): <var: char: 4> / <c-string 1..4: c-string 1-4>

Print jobs with the specified server TSN are excluded from the information output. If you specify *NONE, the operand does not act as a selection criterion.

E SVTSN=*NONE

No server TSNs are excluded from the information output.

E SVTSN=array(16): <var: char: 4>

List of server TSNs for which no information is to be output.

E SRT T=*LIST must be set for this.

E SVTSN=<c-string 1..4: c-string 1-4>

Server TSN which is to be excluded from the information output.

E_OVL_T=*NONE / <var: enum-of _flag_set: 1> / *LIST

Defines the type of entry for film overlays.

E OVL T=*NONE

No film overlays are excluded from the information output.

E_OVL_T=<var: enum-of _flag_set: 1>

The type of entry for E_OVER is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

E OVL T=*LIST

With *LIST, you define that the E OVER operand contains a list of user IDs.

E_OVER=*NONE / array(16): <var: char: 2> / <c-string 2..2: c-string 2-2>

Spoolout jobs for which these physical forms overlays have been specified are to be excluded from the information output.

E OVER=*NONE

No film overlays are excluded from the information output.

E_OVER=array(16): <var: char: 2>

List of film overlays for the print jobs for which no information is to be output.

E_OVL_T=*LIST must be set for this.

E_OVER=<c-string 2..2: c-string 2-2>

Film overlay which is to be excluded from the information output.

E_SPC_T=*NONE / <var: enum-of _flag_set: 1> / *LIST

Defines the type of entry for spoolout classes of print jobs which are to be excluded from the information output.

If you specify *NONE, the operand does not act as a selection criterion.

E_SPC_T=*NONE

No spoolout classes are excluded from the information output.

E_SPC_T=<var: enum-of _flag_set: 1>

The type of entry for E_CLASS is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

E SPC T=*LIST

With *LIST, you define that the E CLASS operand contains a list of user IDs.

E_CLASS=*NONE / array(16): <var: int: 1> / <integer 1..255>

Spoolout classes of the spoolout jobs to be excluded from the information output. If you specify *NONE, the operand does not act as a selection criterion.

E CLASS=*NONE

No spoolout classes are excluded from the information output.

E_CLASS=array(16): <var: int: 1>

List of spoolout classes for which no information is to be output. E_SPC_T=*LIST must be set for this.

E_CLASS=<integer 1..255>

Spoolout class which is to be excluded from the information output.

E_ACC_T=*NONE / <var: enum-of _flag_set: 1> / *LIST

Defines the type of entry for account numbers of print jobs which are to be excluded from the information output. If you specify *NONE, the operand does not act as a selection criterion.

E_ACC_T=*NONE

No account numbers are excluded from the information output.

E_ACC_T=<var: enum-of _flag_set: 1>

The type of entry for E_ACC is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

E ACC T= *LIST

With *LIST you define that the E_ACC operand contains a list of user IDs.

E_ACC=*NONE / array(16): <var: char: 8> / <c-string 1..8: c-string 1-8>

Account numbers of the spoolout jobs to be excluded from the information output. If you specify *NONE, the operand does not act as a selection criterion.

E_ACC=*NONE

No account numbers are excluded from the information output.

E_ACC=array(16): <var: char: 8>

List of account numbers for which no information is to be output. E_ACC_T=*LIST must be set for this.

E ACC=<c-string 1..8: c-string 1-8>

Account number which is to be excluded from the information output.

E_DESTY=*NONE / list-poss(4): *NONE / *DEVICE / *CENTRAL

Print jobs to be processed on the specified output media (pool or devices) are excluded from the information output.

If you specify *NONE, the operand does not act as a selection criterion.

E DESTY=*NONE

Since no output medium is specified, no print jobs are excluded from the information output.

E DESTY=*DEVICE

Information on print jobs for the devices specified in E_DEV_T, E_DVNAM and E_DVNAW are excluded from the information output.

E DESTY=*CENTRAL

Information on print jobs which are to be processed on SPOOL devices in the central printer pool are excluded from the information output.

E_DEV_T=*NONE / <var: enum-of _flag_set: 1> / *LIST / *WILD

Defines the type of entry for the names of devices which are to be excluded from the information output.

If you specify *NONE, the operand does not act as a selection criterion.

E_DEV_T=*NONE

No devices are excluded from the information output.

E_DEV_T=<var: enum-of _flag_set: 1>

The type of entry for E_DVNAM and E_DVNAW is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

E DEV T=*LIST

With *LIST, you define that the E_DVNAM operand contains a list of user IDs.

E_DEV_T=*WILD

The devices are specified partially qualified with wildcards.

E_DVNAM=*NONE / array(16): <var: char: 8> / <c-string 1..8: c-string 1-8>

Names of the devices which are to be excluded from the information output.

If you specify *NONE, the operand does not act as a selection criterion.

E_DVNAM=*NONE

No devices are excluded from the information output.

E_DVNAM=array(16): <var: char: 8>

List of devices for which no information is to be output.

E_DEV_T=*LIST must be set for this.

E_DVNAM=<c-string 1..8: c-string 1-8>

Device which is to be excluded from the information output.

E_DVNAW=*NONE / <var: char: 24> / <c-string 1..24: c-string 1-24>

Names of devices which are to be excluded from the information output. The names are specified partially qualified with wildcards.

If you specify *NONE, the operand does not act as a selection criterion.

E DVNAW=*NONE

No devices are excluded from the information output.

E_DVNAW=<var: char: 24> / <c-string 1..24: c-string 1-24>

The devices are specified partially qualified with wildcards.

E_FRT_T=*NONE / <var: enum-of _flag_set: 1> / *LIST

Defines the type of entry for the format names of print jobs which are to be excluded from the information output.

If you specify *NONE, the operand does not act as a selection criterion.

E_FRT_T=*NONE

No format names are excluded from the information output.

E_FRT_T=<var: enum-of _flag_set: 1>

The type of entry for E_FRT_N is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

E FRT T=*LIST

With *LIST, you define that the E_FRT_N operand contains a list of format names.

E_FRT_N=*NONE / array(16): <var: char: 63> / <c-string 1..63: c-string 1-63>

Format names of print jobs for which no information is to be provided.

E_FRT_N=*NONE

No format names are excluded from the information output.

E_FRT_N=array(16): <var: char : 63>

List of format names of print jobs which are to be excluded from the information output. E_FRT_T=*LIST must be set for this.

E_FRT_N=<c-string 1..63: c-string 1-63>

Format name of the print job which is to be excluded from the information output.

E_DES_T=*NONE / <var: enum-of _flag_set: 1> / *LIST / *WILD

Defines the type of entry for the names of device pools which are to be excluded from the information output.

If you specify *NONE, the operand does not act as a selection criterion.

E DES T=*NONE

No device pools are excluded from the information output.

E DES T=<var: enum-of flag set: 1>

The type of entry for E_DSNAM and E_DSNMW is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

E DES T= *LIST

With *LIST you define that the E_DSNAM operand contains a list of user IDs.

E DES T=*WILD

The device pools are specified partially qualified with wildcards.

E_DSNAM=*NONE / array(16): <var: char: 8> / <c-string 1..8: c-string 1-8>

Names of the device pools which are to be excluded from the information output.

If you specify *NONE, the operand does not act as a selection criterion.

E_DSNAM=*NONE

No device pools are excluded from the information output.

E_DSNAM=array(16): <var: char: 8>

List of device pools for which no information is to be output.

E_DES_T=*LIST must be set for this.

E_DSNAM=<c-string 1..8: c-string 1-8>

Device pool which is to be excluded from the information output.

E_DSNMW=*NONE / <var: char: 24> / <c-string 1..24: c-string 1-24>

Names of device pools which are to be excluded from the information output. The names are specified partially qualified with wildcards.

If you specify *NONE, the operand does not act as a selection criterion.

E_DSNMW=*NONE

No device pools are excluded from the information output.

E_DSNMW=<var: char: 24> / <c-string 1..24: c-string 1-24>

The device pools are specified partially qualified with wildcards.

E_VPR_T=*NONE / <var: enum-of _flag_set: 1> / *LIST

Defines the type of entry for the name of the virtual printer of the print jobs which are to be excluded from the information output.

If you specify *NONE, the operand does not act as a selection criterion.

E_VPR_T=*NONE

No virtual printer names are excluded from the information output.

E_VPR_T=<var: enum-of _flag_set: 1>

The type of entry for E_VPR_N is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. See the table on page 301 for the relationships between the values and the desired types.

E VPR T=*LIST

With *LIST, you define that the E_VPR_N operand contains a list of virtual printer names.

E_VPR_N=*NONE / array(16): <var: char : 8> / <c-string 1..8: c-string 1-8>

Name of the virtual printer of print jobs for which no information is to be output.

E_VPR_N=*NONE

No virtual printers are excluded from the information output.

E_VPR_N=array(16): <var: char: 8>

List of virtual printers for the print jobs which are to be excluded from the information output. E_VPR_T=*LIST must be set for this.

E_VPR_N=<c-string 1..8: c-string 1-8>

Virtual printer of the print jobs which is to be excluded from the information output.

Long form of operand names

Short form in macro	Long form (do not use in the macro)
S_SPN_T	select-SPOOLout-name-type
S_SPNAM	select-SPOOLout-name
S_SPN_W	select-SPOOLout-name-wildcard
S_USR_T	select-user-id-type
S_USER	select-user-identification
S_USERW	select-user-id-wildcard
S_HST_T	select-hostname-type
S_HOST	select-hostname
S_HOSTW	select-hostname_wildcard
S_SRV_T	select-server-name-type
S_SERV	select-server_name
S_SERVW	select-server_name_wildcard
S_FRM_T	select-form-name-type
S_FORM	select-form-name
S_FORMW	select-form_name_wildcard
(part 1 of 3)	
S_TSN_T	select-tsn-type
S_TSN	select-tsn
S_IDENT	select-ident
S_SRT_T	select-server_tsn-type
S_SVTSN	select-server_tsn
S_OVL_T	select-overlay-type
S_OVER	select-overlay
S_SPC_T	select-spoolout_class-type
S_CLASS	select-spoolout_class
S_ACC _T	select-account-type
S_ACC	select-account
S_FRT_T	select-format_name-type
S_FRT_N	select-format-name
S_DESTY	select-dest_type
S_SELTY	select-s_type
S_SPDVT	select-SPOOL-dev-type

Short form in macro	Long form (do not use in the macro)
S_DEV_T	select-device-name-type
S_DVNAM	select-dev-name
S_DVNMW	select-dev_name_wildcard
S_DES_T	select-dest-name-type
S_DSNAM	select-dest-name
S_DSNMW	select-dest-name-wildcard
S_ERR_T	select-error-type
S-ERR-MSG	select-error-message
S-ERRMW	select-error-message_wildcard
S_FOB	select-fob
S_CSN	select-csn
S_PRIO	select-priority
S_ROT	select-rotation
S_JTYP	select-job-type
E_SPN_T	except-SPOOLout-name-type
E_SPNAM	except-SPOOLout-name
E_SPNAW	except-SPOOLout-name-wildcard
E_USR_T	except-user-id-type
E_USER	except-user-identification
E_USERW	except-user-id-wildcard
E_HST_T	except-hostname-type
E_HOST	except-hostname
E_HOSTW	except-hostname_wildcard
E_SRV_T	except-server-name-type
E_SERV	except-server_name
E_SERVW	except-server_name_wildcard
E_FRM_T	except-form-name-type
E_FORM	except-form_name
E_FORMW	except-form_name_wildcard
E_TSN_T	except-tsn-type specify
E_TSN	except-tsn
E_SRT_T	except-server-tsn-type

(part 2 of 3)

Short form in macro	Long form (do not use in the macro)
E_SVTSN	except-server_tsn
E_OVL_T	except-overlay-type
E_OVER	except-overlay
E_SPC_T	except-SPOOLout-class-type
E_CLASS	except-spclass
E_ACC_T	except-account-type
E_ACC	except-account
E_DESTY	except-dest_type
E_DEV_T	except-device-name-type
E_DVNAM	except-dev-name
E_DVNAW	except-dev_name_wildcard
E_FRT_T	except-format_name-type
E_FRT_N	except-format-name
E_DES_T	except-dest-name-type
E_DSNAM	except-dest-name
E_DSNMW	except-dest-name-wildcard
S-ERR-T	select-error-message-type
S-ERMSG	select-error-message
S-ERRMW	select-error-message-wildcard
E_VPR_T	except-virtual_printer-type
E_VPR_N	except-virtual_printer-name

(part 3 of 3)

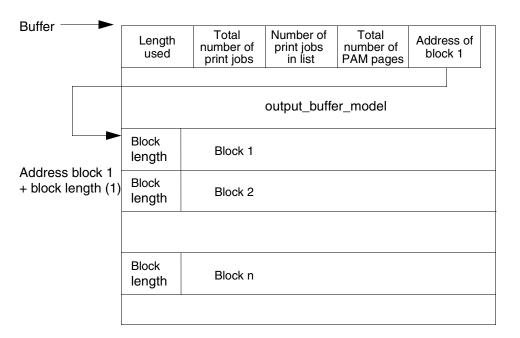
Return codes

(SC2)	SC1	Maincode	Meaning
00	00	0000	No errors
02	00	0000	Print job not found or processing not allowed
01	01	FFFF	Invalid parameter
02	01	FFFF	Invalid address / length
03	01	FFFF	Output area too short
04	01	FFFF	error during cross check of parameters
00	20	FFFF	System error
02	40	FFFF	No authorization
04	40	FFFF	Operand value error
05	40	FFFF	P/L version not supported
00	80	FFFF	Subsystem not loaded
01	80	FFFF	SPOOL not loaded
02	80	FFFF	RSO not loaded
03	80	FFFF	PLAM not loaded
04	80	FFFF	JV not loaded
05	80	FFFF	SPS not loaded
06	80	FFFF	DPRINTCL not loaded
07	80	FFFF	DPRINTCM not loaded
80	80	FFFF	DPRINTSV not loaded
09	80	FFFF	DSEM not loaded

Notes

- The SHOPJST macro returns the above codes. The return codes from HDRCHECK and \$VALID may additionally be returned.
 If no data area was provided for the header, the value X'0000003C' is returned in register 15.
- The information on the print job is stored in an output buffer which must be provided by the user. If the buffer is defined too short, the macro call is terminated with the error X'03 01 FFFF' "Output area too short". A buffer long enough for the complete number of print jobs must be requested in order to receive the complete information. An additional call to SHOPJAT then returns the desired information.

The section _output_buffer_model is always written to the buffer. It contains the total number of print jobs, the address of the first block containing information about a print job and the length of the block concerned. The address of each additional information block can be calculated from the address of the first informations block and its length.



Information on the RSO subsystem is provided by RSO if this subsystem is loaded.
 However, this information only applies to devices controlled by RSO.

Parameter list description

* parameter list description					
SSS_MDL	DS	0F			
SSSHDR FHDR MF=(C,SS	S),EQU	ATES=NO	Standard header		
<pre>* main return codes</pre>					
SSSRCNO	EQU	0	subcode1=00 : no error		
SSSWARN	EQU	2	job not found or processing		
*			not permitted		
SSSPARM	EQU	1	<pre>subcode1 = 01 : wrong</pre>		
*			parameter		
SSSVALD	EQU	2	invalid address/length		
SSSSHRT	EQU	3	output area too short		
SSSSYSE	EQU	0	subcode1 = 20 : system error		
SSSPRIV	EQU	2	subcode1 = 40 : no		
*			authorization		
SSSVERR	EQU	4	operand value error		
SSSVERE	EQU	5	p/l version not supported		
SSSNOSS	EQU	0	<pre>subcode1 = 80 : subsystem not</pre>		
*	_40		loaded		
SSSNSPO	EQU	1	spool not loaded		
SSSNRSO	EQU	2	rso not loaded		
SSSNPLM	EQU	3	plam not loaded		
SSSNJV	EQU	4	jv not loaded		
SSSNSPS	EQU	5	sps not loaded		
SSSNDCL	EQU	6	dprintcl not loaded		
SSSNDCM	EQU	7	dprinter not loaded		
SSSNDSV	EQU	8	dprintsv not loaded		
SSSNDSM	EQU	9	dsem not loaded		
*	LŲŪ	9	uselli flot Todueu		
SSSRSOP	DS	Α	address of rso_pl		
SSSRES1	DS	A	address of sps_pl		
SSSRES2	DS	A	address of dprint_pl		
SSSRES3	DS	A	address of dsem_pl		
SSSRES4	DS	A	address or dselli_pr		
SSSRES5	DS	A			
SSSRES6	DS	A			
SSSRES7	DS DS	A			
SSSCLUS	DS DS	CL8	This operand specifies the		
*	DS	CLO	cluster in which the		
*			print-job is running. When a		
*			BS2000 cluster is specified,		
*			the TSN represents the		
*			gateway host int the remote		
*			specified cluster. Only a		
*			BS2000 cluster may be		
*			specified.		
			apaaau.		

SSSBUF@ * *	DS	А	Address of a user-specified area where the output must
	5.0	_	be returned.
SSSBUFL	DS	F	user buffer length.
SSSINFO	DS	FL1	
<pre>* _info_set</pre>			
SSSORIG	EQU	1	info = *origin
SSSDEST	EQU	2	info = *destination
SSSTRTS	EQU	3	info = *traits
SSSDEVT	EQU	4	info = *device_type
SSSDVAL	EQU	5	info = *device_type
SSSSUMM	EQU	6	info = *summary
SSSDIST	EQU	7	info = *distributed
*	-,-		
SSSUN2	DS	XL3	
*			
SSSSEL	DS	0XL2640	select
*			
SSSASSPN	DS	0XL156	
SSSAFC8	DS	FL1	
* _flag_set			
SSSFALL	EQU	0	_type = *all
SSSFNON	EQU	1	_type = *none
SSSFLOC	EQU	2	_type = *local
SSSFREM	EQU	3	_type = *remote
SSSFCEN	EQU	4	_type = *central
SSSFDEV	EQU	64	_type = *device
SSSFMDV	EQU	128	_type = *manag_dev
SSSFWLD	EQU	5	_type = *wildcard
SSSFLST	EQU	6	_type = *list
SSSFSTD	EQU	7	_type = *std
SSSFOWN	EQU	8	_type = *own
SSSFHOM	EQU	9	_type = *home
SSSFIDE	EQU	10	_type = *ident
SSSFANY	EQU	11	_type = *any
SSSFONL	EQU	12	_type = *only
SSSFRNG	EQU	13	
			_type = *range
SSSFONE	EQU	14	_type = *one
SSSFN0	EQU	15	_type = *no
SSSFDES	EQU	16	_type=*destination
*	DC	C1 04	
SSSAWC8	DS	CL24	
SSSAUC8	DS	XL3	
SSSAEC8	DS	16CL8	
SSSAEC8#	EQU	16	
*			
*	DC	0.71.15.6	
SSSBSUID	DS	0XL156	

SSSBFC8 SSSBUC8 SSSBEC8 SSSBEC8	DS DS DS DS EQU	FL1 CL24 XL3 16CL8 16
SSSCSHOS SSSCFC8 SSSCWC8 SSSCUC8 SSSCEC8 SSSCEC8	DS DS DS DS DS EQU	0XL156 FL1 CL24 XL3 16CL8
*		
SSSDSSER SSSDFC8 SSSDWC8 SSSDEC8 SSSDEC8# *	DS DS DS DS DS EQU	OXL156 FL1 CL24 XL3 16CL8 16
* SSSESFOR SSSEFC6 SSSEWC6 SSSEUC6 SSSEEC6 \$SSEEC6	DS DS DS DS DS EQU	0XL124 FL1 CL24 XL3 16CL6 16
* SSSFSTSN SSSFFC4 SSSFWC4 SSSFUC4 SSSFEC4 \$SSFEC4 *	DS DS DS DS DS EQU	0XL92 FL1 CL24 XL3 16CL4
* SSSGSIDE SSSGFI4 SSSGUN1 SSSGEI4 *	DS DS DS DS	OXL68 FL1 XL3 16F
* SSSHSSTS SSSHFC4 SSSHWC4	DS DS DS	OXL92 FL1 CL24

SSSHUC4 SSSHEC4 SSSHEC4#	DS DS EQU	XL3 16CL4 16	
* SSSISOVE SSSIFC2 SSSIWC2 SSSIUC2 SSSIEC2 SSSIEC2 *	DS DS DS DS DS		
* SSSJSSPC SSSJFI1 SSSJUI2 SSSJEI1 *	DS DS DS DS	0XL20 FL1 XL3 16X	
* SSSKSACC SSSKFC8 SSSKWC8 SSSKUC8 SSSKEC8 SSSKEC8	DS DS DS DS DS EQU	0XL156 FL1 CL24 XL3 16CL8	
* SSSQSFTN SSSQFC63 SSSQWC63 SSSQUC63 SSSQEC63 SSSQEC63# *	DS DS DS DS DS	0XL1036 FL1 CL24 XL3 16CL63	
SSSLSDES SSSLFLDS SSSLSTDS	DS DS DS	0XL169 FL1 FL1	
<pre>* sel_type_set SSSSTMA SSSSTMU</pre>	EQU EQU	128 64	<pre>sel-type = *may sel-type = *must</pre>
* SSSLU1DS SSSLDTDS SSSLNADS SSSLU2DS SSSLDSEL SSSLDSEL#	DS DS DS DS DS EQU	XL2 FL1 CL24 XL3 16CL8 16	

*			
SSSLDVDS	DS	0XL8	
SSSLDTB1	DS	AL1	device type bytel
SSSLLP	EQU	X'80'	LP printer
SSSLND	EQU	X'40'	ND printer
SSSLHP	EQU	X'20'	HP printer
SSSLLP48	EQU	X'10'	LP48 printer
SSSLLP65	EQU	X'08'	LP65 printer
SSSLTAPE	EQU	X'04'	TAPE
SSSLLPEM	EQU	X'01'	LP-EMUL printer
SSSLDTB2	DS	AL1	device type byte2
SSSLDSU1	EQU	X'CO'	device type bytez
SSSLHP90	EQU	X'20'	HP90 printer
SSSLAP50	EQU	X'10'	APA-2050 printer
SSSLAP90	EQU	X'08'	APA-2090 printer
SSSLTW90	EQU	X'04'	TWIN-2090 printer
SSSLDSU2	EQU	X'03'	TWIN 2000 PITTICE
SSSLDTB3	DS	AL1	device type byte3
SSSLDSU3	EQU	X'FF'	devise type system
SSSLDTB4	DS	AL1	device type byte4
SSSLPCL	EQU	X'80'	PCL printer
SSSLDSU4	EQU	X'7F'	P
SSSLDTB5	DS	AL1	device type byte5
SSSLDSU5	EQU	X'FF'	
SSSLDTB6	DS	AL1	device type byte6
SSSLDSU6	EQU	X'FF'	
SSSLDTB7	DS	AL1	device type byte7
SSSLDSU7	EQU	X'FF'	
SSSLDTB8	DS	AL1	device type byte8
SSSLDSU8	EQU	X'F0'	
SSSLVIRT	EQU	X'08'	VIRTUAL printer
SSSLDSU9	EQU	X'07'	
*			
SSSLUDS	DS	XL1	
*			
SSSUN4	DS	XL3	
*			
SSSMSFOB	DS	0XL8	
SSSMFR2	DS	FL1	
SSSMUN1	DS	XL1	
SSSMLR2	DS	Н	
SSSMHR2	DS	Н	
SSSMUR2	DS	XL2	
*			
*	D.C	0.71.0	
SSSNSCSN	DS	0XL8	
SSSNFR2	DS	FL1	
SSSNUN1	DS	XL1	

SSSNLR2 SSSNHR2 SSSNUR2 *	DS DS DS	H H XL2	
* SSSOSPRI SSSOFR1 SSSOLR1 SSSOHR1 SSSOUR1	DS DS DS DS	OXL4 FL1 X X XL1	
* SSSSROT	DS	FL1	
<pre>* rotation_set SSSROTA SSSROTY SSSROTN *</pre>	EQU EQU EQU	0 1 2	<pre>rotation = *any rotation = *yes rotation = *no</pre>
SSSSJTY	DS	16X	
* SSSRSVPN SSSRFC8 SSSRWC8 SSSRUC8 SSSREC8 SSSREC8	DS DS DS DS DS	0XL156 FL1 CL24 XL3 16CL8	
* SSSUN5 *	DS	XL3	
* SSSEXC *	DS	0XL2529	_except
SSSOESPN SSSOFC8 SSSOWC8 SSSOUC8 SSSOEC8 *	DS DS DS DS EQU	16CL8	
SSS1EUID SSS1FC8 SSS1WC8 SSS1UC8 SSS1EC8 SSS1EC8#	DS DS DS DS EQU	0XL156 FL1 CL24 XL3 16CL8	
* SSS2EHOS	DS	0XL156	

SSS2FC8 SSS2WC8 SSS2UC8 SSS2EC8 *	DS DS DS DS EQU	FL1 CL24 XL3 16CL8 16
* SSS3ESER SSS3FC8 SSS3WC8 SSS3UC8 SSS3EC8 SSS3EC8	DS DS DS DS DS EQU	0XL156 FL1 CL24 XL3 16CL8
SSS4EFOR SSS4FC6 SSS4WC6 SSS4UC6 SSS4EC6 SSS4EC6# *	DS DS DS DS DS EQU	0XL124 FL1 CL24 XL3 16CL6
* SSS5ETSN SSS5FC4 SSS5WC4 SSS5UC4 SSS5EC4 SSS5EC4	DS DS DS DS DS EQU	0XL92 FL1 CL24 XL3 16CL4
* SSS6ESTS SSS6FC4 SSS6WC4 SSS6UC4 SSS6EC4 SSS6EC4# *	DS DS DS DS DS EQU	0XL92 FL1 CL24 XL3 16CL4 16
* SSS7E0VE SSS7FC2 SSS7WC2 SSS7UC2 SSS7EC2 SSS7EC2# *	DS DS DS DS DS EQU	0XL60 FL1 CL24 XL3 16CL2
* SSS8ESPC	DS	0XL20

SSS8FI1 SSS8UI2 SSS8EI1	DS DS DS	FL1 XL3 16X	
* SSS9EACC SSS9FC8 SSS9WC8 SSS9UC8 SSS9EC8 SSS9EC8	DS DS DS DS EQU	0XL156 FL1 CL24 XL3 16CL8	
* SSSSSFTN SSSSFC63 SSSSWC63 SSSSUC63 SSSSEC63 SSSSEC63	DS DS DS DS DS EQU	0XL1036 FL1 CL24 XL3 16CL63	
SSSZEDES SSSZFLDS SSSZSTDS SSSZU1DS SSSZDTDS SSSZDTDS SSSZNADS SSSZNADS SSSZU2DS SSSZDSEL SSSZDSEL#	DS DS DS DS DS DS DS DS EQU	0XL169 FL1 FL1 XL2 FL1 CL24 XL3 16CL8	
* SSSZDVDS SSSZDTB1 SSSZLP SSSZND SSSZHP SSSZLP48 SSSZLP65 SSSZLP65 SSSZTAPE SSSZDTB2 SSSZDTB2 SSSZDSU1 SSSZHP90 SSSZAP50 SSSZAP50 SSSZAP90 SSSZTW90 SSSZDSU2 SSSZDSU2	DS DS EQU	0XL8 AL1 X'80' X'40' X'20' X'10' X'08' X'04' X'01' AL1 X'C0' X'20' X'10' X'08' X'04' X'03' AL1	device type byte1 LP printer ND printer HP printer LP48 printer LP65 printer TAPE LP-EMUL printer device type byte2 HP90 printer APA-2050 printer APA-2090 printer TWIN-2090 printer

SSSZDSU3	EQU	X'FF'	
SSSZDTB4 SSSZPCL SSSZDSU4	DS EQU EQU	AL1 X'80' X'7F'	device type byte4 PCL printer
SSSZDTB5 SSSZDSU5	DS EQU	AL1 X'FF'	device type byte5
SSSZDTB6 SSSZDSU6	DS EOU	AL1 X'FF'	device type byte6
SSSZDSUU SSSZDTB7 SSSZDSU7	DS EQU	AL1 X'FF'	device type byte7
SSSZDSU7 SSSZDTB8 SSSZDSU8	DS EOU	AL1 X'FO'	device type byte8
SSSZVIRT SSSZDSU9	EQU EQU	X'08' X'07'	VIRTUAL printer
SSSZUDS *	DS	XL1	
*			
SSSTSVPN SSSTFC8 SSSTWC8 SSSTUC8 SSSTEC8 SSSTEC8#	DS DS DS DS EQU	OXL156 FL1 CL24 XL3 16CL8	
SSSUN6 SSS#	DS EQU	XL15 *-SSSHDR	

4.1 Support macros for MODPJAT

The following macros support using the MODPJAT macro. They are used to set subsystemspecific parameters for the MODPJAT macro.

"MODJRSO - define RSO parameters for MODPJAT macro" on page 341 This macro is used to set parameters for the RSO subsystem.

"MPJADPC - define Dprint parameters for MODPJAT macro" on page 348 This macro is used to set parameters for the Dprint subsystem.

"MPJASPS - define SPS parameters for MODPJAT macro" on page 350 This macro is used to set parameters for the SPS subsystem.

MODJRSO - define RSO parameters for MODPJAT macro

User group: Nonprivileged users **Programming language**: Assembler

Macro type: S

The MODJRSO macro writes RSO-specific values for calling the MODPJAT macro into a parameter list.

Format

Operation	Operands			
MODJRSO	,VARIANT= <u>001</u> / <c-string 33=""></c-string>			
	,CHARATT= <u>*STD</u> / *ALL / *RESTRICTED / <var: _char_att_type_set:1="" enum-of=""></var:>			
	,REDIR= <u>*STD</u> / *YES / *NO / <var: _redir_type_set:1="" enum-of=""></var:>			
	TOPOFF=*UNCHANGED / *IGNORE / <integer -255255=""></integer>			
	LEFTOFF=*UNCHANGED / *IGNORE / <integer -255255=""></integer>			
	URL=(urlptr,urllen,prtype) urlptr: <var: pointer=""> / *NONE / *UNCHANGED urllen: <var: 065535="" int="">/ <integer 11023=""> / *NONE prtype: <var: _prtype_set="" enum-of:1=""> / *NONE /*UNCHANGED /</var:></integer></var:></var:>			

Description of the operands

VARIANT=001 / <c-string 3..3>

Specifies the variant of the parameter list.

CHARATT=*STD / *ALL / *RESTRICTED / <var: enum-of char att type set:1>

Specifies whether all or only specific character set properties are to apply for the RSO job. The properties include, for example, character type, NLQ, color, etc. (see the SHOW-SPOOL-CHARACTER-SETS command or SPSERVE statement). This operand is ignored for header and trailer pages.

CHARATT=*STD

The properties set as default for the character set used for printing are applied to the current RSO job.

CHARATT=*ALL

All properties for the character set used for printing are applied to the current RSO job.

CHARATT=*RESTRICTED

Only the three properties CHARACTER TYPE, LANGUAGE and NLQ (NEAR-LETTER-QUALITY) are applied.

CHARATT=<var: enum-of char att type set:1>

The character set properties to be used are not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*STD	
1	*ALL	
2	*RESTRICTED	
See the parameter list description as of page 346.		

REDIR=*STD / *YES / *NO / <var: enum-of redir type set:1>

Specifies whether a device administrator may redirect the RSO job to another printer. Job redirection by the user or systems support are not affected.

REDIR=*STD / *YES

A device administrator may redirect the RSO job to another printer.

REDIR=*NO

A device administrator may not redirect the RSO job to another printer.

REDIR=<var: enum-of _redir_type_set:1>

RSO job redirection is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0	*STD	
1	*YES	
2	*NO	
See the parameter list description as of page 346.		

TOPOFF=*UNCHANGED / *IGNORE / <integer -255..255>

Defines the offset of the top edge of the print page from the top edge of the physical page in millimeters. First the print page is positioned on the paper and then the writing is rotated and positioned within the print page. This means that when the print page is moved around on the paper, the orientation of the text within it is ignored.

This operand is only permitted for printers 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL, 9000-PCL, 9021, 9022-200 and 2030-PCL.

Note

Only TOPOFF=*IGNORE can be used in conjunction with DOCFOR=*SPECIAL_FORMAT.

TOPOFF=*IGNORE

Print control leaves no spacing between the print page and the top edge of the page. The offset which is effective when printing the document is defined by either the printer default or the prolog file setting.

TOPOFF=<integer -255..255>

The permitted value range is from -255 to +255. Positive values shift the print page down while negative values shift it up.

LEFTOFF=<u>*UNCHANGED</u> / *IGNORE/ <integer -255..255>

Defines the offset of the left edge of the print page from the left edge of the physical page in millimeters. First the print page is positioned on the paper and then the writing is rotated and positioned within the print page. This means that when the print page is moved around on the paper, the orientation of the text within it is ignored.

This operand is only permitted for printers 4818-PCL, 4821-PCL, 4822-PCL, 4824-PCL, 4825-PCL, 9000-PCL, 9021, 9022-200, 9026-PCL and 20930-PCL.

Note

Only LEFTOFF=*IGNORE can be used in conjunction with DOCUMENT_FORMAT=*SPECIAL_FORMAT.

LEFTOFF=*IGNORE

Print control leaves no spacing between the print page and the left edge of the page. The offset which is effective when printing the document is defined by either the printer default or the prolog file setting.

LEFTOFF=<integer -255..255>

The permitted value range is from -255 to +255. Positive values shift the print page right while negative values shift it left.

URL=(urlptr,urllen,prtype)

Specifies the IPP printer addressed by its URL

urlptr: <var: pointer>

Specifies an area containing the printer URL.

urlptr: *NONE

No printer URL has been specified.

urlptr: *UNCHANGED

Operand default value. The current setting is not changed.

urllen: <var: int 0..65535>Specifies the area length.urllen: <integer 1..1023>Specifies the area length.

urllen: *NONE

Operand default value. No URL has been specified.

prtype: <var: enum-of:1 _PRTYPE_SET>

The printer type is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationship exists between the values and the desired functions:

value	printer type
72	*4850_PCL
73	*4830_PCL
74	*4818_PCL
75	*9000_EPSQ
76	*9000_EPLQ
77	*9000_EPFX
78	*9000_PRO
79	*9000_PS
7B	*9046

value	printer type
7D	*9045_ANSI
7E	*9015
7F	*4825_PCL
80	*4822_PCL
84	*9001
88	*9025
8A	*9013
8B	*9000
8C	*9001_31
8D	*9011
8E	*9012
8F	*9022
A0	*9014
A1	*9021
A2	*9022_200
A4	*9000_PCL
A6	*4824_PCL
A7	*9026_RENO
A8	*9026_PCL
A9	*4812
AA	*9097
AB	*4011
AC	*4813
AD	*4821_PCL
AE	*2030_PCL

prtype: *NONE

No URL has been specified.

prtype: *UNCHANGED

Operand default value. The current setting is not changed.

```
prtype: *4850_PCL / *4830_PCL / *4818_PCL / *9000_EPSQ / *9000_EPLQ /
*9000_EPFX / *9000_PRO / *9000_PS / *9046 / *9045_ANSI / *9015 / *4825_PCL /
*4822_PCL / *9001 / *9025 / *9013 / *9000 / *9001_31 / *9011 / *9012 / *9022 / *9014 /
*9021 / *9022_200 / *9000_PCL / *4824_PCL / *9026_RENO / *9026_PCL / *4812 /
*9097 / *4011 / *4813 / *4821_PCL / *2030_PCL
Specifies the RSO device type of the target printer
```

Parameter list description

```
parameter list description
SROAMDI
        DS
               0F
SROAHDR FHDR MF=(C,SROA), EQUATES=NO
                                         Standard header
                FI1
SROACHRA DS
                                         Specifies whether all or only
                                         specific character set
                                         attributes are to be used
                                         for the print-job. These
                                         attributes include character
                                         type, near letter quality,
                                         color, etc. This operand
                                         does not apply to header and
                                         trailer pages.
   _char_att_type_set
        AIF ('&EOUATES' NE 'YES').001001
SROACSTD EOU
SROACALL FOU
               1
SROACRST EOU
.Q01001 ANOP
SROARDIR DS
               FL1
                                         Specifies whether a device
                                         administrator may redirect
                                         print-job to another
                                         printer. Redirection of
                                         print-job by users or
                                         system-administration is not
                                         affected by this operand.
    redir type set
               ('&EQUATES' NE 'YES').Q01002
         ATF
SROARSTD EQU
               0
SROARYES EOU
               1
SROARNO
          FOU
.Q01002 ANOP
SROASPEC DS
               AL1
                                         specified; for compatibility
                                         with MODJRSO
         ATF
               ('&EOUATES' NE 'YES').001003
SROACSPC
        EQU
              X'80'
SROARSPC EQU
              X'40'
```

```
SROATSPC EQU
              X'20'
SROALSPC EQU
             X'10'
SROAUNUS EQU
              X'0F'
.Q01003 ANOP
SROAFIL1 DS
               CL1
                                        slack byte
SROATOPO DS
                                        Specifies the length of the
                                        white band that must be
                                        inserted
SROALEFO DS
                                        Specifies the width of the
               Н
                                        white band that must be
                                        inserted
SROAUNS1
        DS
               XL8
                                        unused
        AIF
              ('&EQUATES' NE 'YES').Q01004
SROA#
         EOU
              *-SROAHDR
```

MPJADPC - define Dprint parameters for MODPJAT macro

User group: Nonprivileged users **Programming language**: Assembler

Macro type: S

The Dprint parameters for the MODPJAT macro are defined with this macro.

Format

Operation	Operands		
MPJADPC	VARIANT=001 / <c-string 33=""></c-string>		
	,FORMAT= <u>*STD</u> / <var: 63="" char:=""> / <c-string 163=""></c-string></var:>		
	.RESLOC=*STD / *HOME / *SERVER / <var: enum-of_resloc_set:1=""></var:>		

Description of the operands

VARIANT=001 / <c-string 3..3>

Specifies the variant of the parameter list.

FORMAT=*STD / <var: char: 63> / <c-string 1..63>

Type of document contents.

FORMAT=*STD

By default, no special format name is assumed.

FORMAT=<var: char: 63>

Name of the field containing the format name.

FORMAT=<c-string 1..63>

The format name is specified directly.

RESLOC=*STD / *HOME / *SERVER / <var: enum-of_resloc_set:1>

Location of the resources used for printing.

RESLOC=*STD

By default, the values from the SPOOL parameter file are assumed.

RESLOC=*HOME

The client resources are used.

RESLOC=*SERVER

The server resources are used.

RESLOC=<var: enum-of resloc set:1>

Name of the equate for the operand value. Only possible in conjunction with MF=M.

Parameter list description

```
parameter list description
        DS
DMAPI
              0F
DMAHDR
        FHDR MF=(C,DMA), EQUATES=NO
                                        Standard header
DMAUNU1 DS
              XL8
                                         reserved for compatibility
                                        with print command
DMAFMT DS
              CI 63
                                        document content type
DMARESL DS
              FL1
                                         location of the resources
                                        used to process the
                                        print-job is submitted
   _resloc_set
        AIF
             ('&EQUATES' NE 'YES').Q01001
DMARSTD EQU
              0
                                         resource-location = *std
DMARHOM FOU
              1
                                         resource-location = *home
DMARSRV EQU
                                         resource-location = *server
.Q01001 ANOP
DMASP1
        DS
              AL1
                                        specified
        ATF
              ('&EQUATES' NE 'YES').Q01002
DMASP11 FOU
              X'80'
DMASPUN1 EOU
              X'40'
                                         reserved for compatibility
                                        with print command
DMASP13 FOU
              X'20'
DMASP14 EOU
              X'10'
DMASP1F EQU
              X'OF'
.001002 ANOP
DMAUNU2 DS
              XL63
                                         reserved for compatibility
                                        with print command
DMAFIL1 DS
              CI 4
                                         slack bytes
        AIF
              ('&EQUATES' NE 'YES').Q01003
        EQU
DMA#
              *-DMAHDR
.001003 ANOP
```

MPJASPS - define SPS parameters for MODPJAT macro

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

The SPS parameters for the MODPJAT macro are defined with this macro.

Format

Operation	Operands
MPJASPS	VARIANT= <u>001</u> / <c-string 33=""></c-string>
	FONTS=*UNCHANGED / array(4): <var: 8:="" alphanum_name_18="" char:=""></var:>
	PAGEDEF=*UNCHANGED / <var: 8:="" alphanum_name_18="" char:=""></var:>
	FORMDEF= <u>*UNCHANGED</u> / <var: 8:="" alphanum_name_18="" char:=""></var:>
	TRC=*UNCHANGED / <var: 1="" _trc_set:="" enum-of=""> / *YES / *NO</var:>
	MSGPAGE= <u>*UNCHANGED</u> / <var: 1="" _msgpage_set:="" enum-of=""> / *YES / *NO</var:>
	FRONTOVL=*UNCHANGED / <var: 8:alphanum_name_18="" char:=""> / *STD</var:>
	BACKOVL=* <u>UNCHANGED</u> / <var: 8:alphanum_name_18="" char:=""> / *STD</var:>

Description of the operands

VARIANT=<u>001</u> / <c-string 3..3>

Specifies the variant of the parameter list.

FONTS=*UNCHANGED / array(4): <var: char: 8: alphanum_name_1..8>

Defines the fonts which are to be evaluated by SPS if the TABLE-REFERENCE-CHAR operand of the PRINT-DOCUMENT and MODIFY-PRINT-JOB-ATTRIBUTES commands or the TRC operand of the PRNTDOC and MPJASPS macros are used.

FONTS=*UNCHANGED

The setting defined when the print job is issued remains effective.

FONTS=array(4): <var: char: 8: alphanum_name_1..8>

Name of the fonts which are to be linked with the TRC value in the print file (which each represents a specific font). Regardless of the number of different TRC values in the file, a maximum of four fonts, which are to be specified in the form of a list, can be used for printing the data records. TRC values greater than X'03' (corresponds to the fourth element in the list) automatically refer back to the first font in the list.

PAGEDEF=*UNCHANGED / <var: char: 8: alphanum_name_1..8>

Specifies the page definition to be used for printing on APA printers.

PAGEDEF=*UNCHANGED

The setting defined when the print job is issued remains effective.

PAGEDEF=<var: char: 8: alphanum_name_1..8>

Only for APA printers:

The page definition with the specified name is to be used. It must be in the SPSLIB.

The first two characters of the specified name must be "P1", otherwise the print job modification is rejected.

FORMDEF=*UNCHANGED / <var: char: 8: alphanum_name_1..8>

Specifies the format definition to be used for printing on APA printers.

FORMDEF=*UNCHANGED

The setting defined when the print job is issued remains effective.

FORMDEF=<var: char: 8: alphanum_name_1..8>

The format definition with the specified name is to be used. It must be in the SPSLIB.

The first two characters of the specified name must be "F1", otherwise the command is rejected.

TRC=<u>*UNCHANGED</u> / <var: enum-of _trc_set:1> / *YES / *NO

Specifies whether the user selects the fonts to be used for print page layout via control characters in the text. This selection can be made with the TRC control characters (Table Reference Character) which refer within the data records to a list of fonts.

The list can either be part of the page definition PAGEDEF or be made known using the FONTS operand. A detailed description of using the TRC can be found in the "SPOOL (BS2000/OSD)" manual.

TRC=*UNCHANGED

The setting defined when the print job is issued remains effective.

TRC=<var: enum-of trc set:1>

Whether TRC control characters are used is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

1	*YES
2	*NO
4	*UNCHANGED

TRC=*YES

The print file contains font selection control characters (TRC) which are to be evaluated by SPS. Each TRC value represents a specific font with which the data record concerned is to be printed. The separate values (from X'00' to X'0F') are to be linked according to their order either with the entries in the FONTLIST (font list in the page definition PAGEDEF) or explicitly with the fonts specified in the CHARACTER SETS operand. Regardless of the number of different TRC values in the file, a maximum of four fonts, which are to be specified in the form of a list, can be used for printing the data records. TRC values greater than X'03' (corresponds to the fourth element in the list) automatically refer back to the first font in the list.

TRC=*NO

Either no font selection control characters are in the print file or, if there are, they are to be ignored.

MSGPAGE=<u>*UNCHANGED</u> / <var: enum-of _msgpage_set:1> / *YES / *NO

Specifies whether the APA message page is to be printed or not. The APA message page contains error messages and warnings. The entry is ignored if an APA printer is not in use.

MSGPAGE=*UNCHANGED

The setting defined when the print job is issued remains effective.

MSGPAGE=<var: enum-of _msgpage_set:1>

Whether the APA message page is printed is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

1	*YES
2	*NO
4	*UNCHANGED

MSGPAGE=*YES

The APA message page is printed. The entry is ignored if an APA printer is not in use.

MSGPAGE=*NO

The APA message page is not printed.

FRONTOVL=*UNCHANGED / <var: char: 8:alphanum name 1..8> / *STD

Specifies whether an APA overlay is to be used for the front page.

FRONTOVL=*UNCHANGED

The current setting remains effective.

FRONTOVL=<var: char: 8:alphanum_name_1..8>

Name of the APA overlay that is to be used for the front page.

FRONTOVL=*STD

No APA overlays are used for the output, except when an APA overlay is specified in the corresponding form definition.

BACKOVL=*UNCHANGED / <var: char: 8:alphanum_name_1..8> / *STD

Specifies whether an APA overlay is to be used for the back page.

BACKOVL=*UNCHANGED

The current setting remains effective.

BACKOVL=<var: char: 8:alphanum_name_1..8>

Name of the APA overlay that is to be used for the back page.

BACKOVL=*STD

No APA overlays are used for the output, except when an APA overlay is specified in the corresponding form definition.

Parameter list description

```
parameter list description
        DS
               0F
SPSMPL
SPSMHDR FHDR MF=(C,SPSM), EQUATES=NO
                                       Standard header
                                       When TRCs are used, allows to
SPSMFNT DS
              4CL8
                                        reference a list of fonts
SPSMFNT# EQU
                                       Element name of the page
SPSMPAGD DS
             CL8
                                       definition to be used.
              CL8
                                       Element name of the format
SPSMFRMD DS
                                       definition to be used.
SPSMTRC DS
               FL1
                                       Specifies whether the file to
                                       print contains TRCs.
   trc set
SPSMTRCY EQU
               1
SPSMTRCN FOU
SPSMTRCS EQU
SPSMMSGP DS
               FI1
   _msgpage_set
SPSMMSGY EQU
              1
SPSMMSGN EQU
               2
SPSMMSGS EOU
SPSMSP1
         DS
               AL1
                                       specified
SPSMSP11 EOU
             X,80,
SPSMSP12 EQU
             X'40'
SPSMSP13 EQU
             X'20'
SPSMSP14 EOU
             X'10'
SPSMSP15
        EQU
             Х,08,
SPSMSP1F EQU
               X'07'
SPSMFIL1 DS
               CL1
                                       slack byte
SPSM#
         EQU
             *-SPSMHDR
```

4.2 Output structures of the SHOPJAT macro

The following macros describe the structures with which the SHOPJAT macro output buffer is filled after a status query.

- "SJAODOF output structure for INFO=*DOCUMENT_FORMAT" on page 356 Describes the structure with which the SHOPJAT macro output buffer is filled after a query with INFO=*DOCUMENT_FORMAT.
- "SJAODOP output structure for INFO=*DOCUMENT_PART" on page 358

 Describes the structure with which the SHOPJAT macro output buffer is filled after a query with INFO=*DOCUMENT_PART.
- "SJAOLCT output structure for INFO=*LAYOUT_CONTROL" on page 361
 Describes the structure with which the SHOPJAT macro output buffer is filled after a query with INFO=*LAYOUT_CONTROL.
- "SJAOPJC output structure for INFO=*PRINT_JOB_CONTROL" on page 364
 Describes the structure with which the SHOPJAT macro output buffer is filled after a query with INFO=*PRINT_JOB_CONTROL.
- "SJAORSD output structure for INFO=*RESOURCES_DESCRIPTION" on page 365 Describes the structure with which the SHOPJAT macro output buffer is filled after a query with INFO=*RESOURCES_DESCRIPTION.
- "SJAOTOP output structure for INFO=*TO_PRINTER" on page 367
 Describes the structure with which the SHOPJAT macro output buffer is filled after a query with INFO=*TO_PRINTER.
- "SPJARSO RSO output structure for SHOPJAT macro" on page 368

 Describes the structure with which the SHOPJST macro output buffer is supplemented if a query concerned a print job to be executed with RSO.
- "SPJADPC Dprint output structure for SHOPJAT macro" on page 369

 Describes the structure with which the SHOPJST macro output buffer is supplemented if a query concerned a print job to be executed with Dprint.
- "SPJASPS SPS output structure for SHOPJAT macro" on page 370

 Describes the structure with which the SHOPJST macro output buffer is supplemented if a query concerned a print job to be executed with SPS.

SJAODOF - output structure for INFO=*DOCUMENT_FORMAT

User group: Nonprivileged users **Programming language**: Assembler

Macro type: S

This macro describes the form in which the SHOPJAT macro outputs information addressed under the DOCUMENT_FORMAT option.

Format

Operation	Operands
SJAODOF	VARIANT=001 / <c-string 33=""></c-string>

Description of the operands

VARIANT=001 / <c-string 3..3>

Specifies the variant of the parameter list.

Parameter list description

* _doc_format_	_block_	_mdl	
SAFDPBM	DS	OH	
SAFDFDF	DS	FL1	document format
* _doc_format_	_set		
SAFDFTE	EQU	0	text
SAFDFPG	EQU	1	page
SAFDFSP	EQU	2	special
*			
*			
SAFCONTROL_MODE	DS	0XL3	_control_mode
SAFD0F0	DS	FL1	
* _contmod_typ	be_set		
SAFCTMP	EQU	0	page mode
SAFCTNO	EQU	1	no
SAFCTPH	EQU	2	physical
SAFCTML	EQU	4	logical
SAFCTLI	EQU	16	line mode
SAFCTTR	EQU	32	transparent
SAFCTMA	EQU	64	apa
*			
SAFPCC	DS	FL1	

* _contmod_p		0				
SAFTOPY SAFTOPN	EQU EQU	0 1	yes no			
*						
SAFCTLT	DS	FL1				
<pre>* _contmod_c SAFCCNO</pre>	EQU	_set O	none			
SAFCCHC	EQU	1	hp compatible			
SAFCCHP	EQU	2	hp			
*						
*	DC	C1 C2				
SAFDFFN *	DS	CL62				
SAFLINESP	DS	0XL4	_linesp			
SAFCCP0	DS	Н	_ '			
SAFSPAC	DS	FL1				
* _line_space_set						
SAFSPA1 SAFSPA2	EQU	1 2	space=1			
SAFSPA3	EQU EQU	3	space=2 space=3			
SAFSPAN	EQU	4	space=no			
SAFSPAE	EQU	5	space=ebcdic			
SAFSPAI	EQU	6	space=ibm			
SAFSPAA *	EQU	7	space=asa			
SAFFIL4	DS	CL1				
*	55	021				
SAFDFLP	DS	Н				
SAFDFCC	DS	H				
SAFDFHL * _headline_	DS	FL1				
SAFHLNO	EQU	1	no			
SAFHLP	EQU	4	page number			
SAFHLD	EQU	64	date			
SAFHLFR *	EQU	128	first record			
SAFDF0F	DS	FL1				
* _outform_s	set					
SAFOUTC	EQU	0	character			
SAFOUTH *	EQU	2	hexadecimal			
* SAFOUTPUT#	EQU	*-SAFDFDF				
5 55 5.,,	_45	5 5. 5.				

..AX;

SJAODOP - output structure for INFO=*DOCUMENT_PART

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

This macro describes the form in which the SHOPJAT macro outputs information addressed under the DOCUMENT_PART option.

Format

Operation	Operands
SJAODOP	VARIANT=001 / <c-string 33=""></c-string>

Description of the operands

VARIANT=001 / <c-string 3..3>

Specifies the variant of the parameter list.

Parameter list description

```
_doc_part_block_mdl
SAPDPBL DS OF
SAPSDPSC DS 0XL64
                                       Identifier of the optional
                                       section records
SAPSSEID DS
             CL60
SAPSSEPO DS
SAPSSEIN DS
             AL1
                                       sect ind
SAPSSEGV EQU X'80'
                                       ON = given, OFF = not given.
SAPSSEU1 EQU X'7F'
SAPSSEIT DS
              AL1
                                       id_type
SAPSSETY EQU X'80'
                                       ON = character, OFF =
                                       hexadecimal
SAPSSEU2 EQU X'7F'
SAPDPIP DS
              0XI 156
                                       _inpart
SAPDPDL DS
              FL1
* select set
SAPSEAL EQU 1
SAPDPU1 DS
              XL3
```

SAPFRCH DS SAPFRCBF EC SAPFRCBS EC SAPFRCB EC SAPFRCIT EC SAPFRCU1 EC SAPFRCS DS SAPFRCS DS SAPFRCS DS	S QU QU QU QU QU S S	0XL76 AL1 X'80' X'40' X'20' X'10' X'0F' CL60 AL1 X'80'	first record type ind1
SAPFRCST EC * SAPFRCU2 EC SAPFRCU3 DS SAPFRCVA DS SAPFRCOC DS SAPFRCOC DS SAPFRCU4 DS * *	QU S S S	X'7F' XL2 F F H XL2	ON = character, OFF = hexadecimal
SAPLDPLR DS SAPLRCTY DS SAPLRCBF EC SAPLRCBF EC SAPLRCBS EC SAPLRCIT EC SAPLRCU1 EC SAPLRCSI DS	S QU QU QU QU	0XL76 AL1 X'80' X'40' X'20' X'10' X'0F' CL60	last record type
SAPLRCIN DS SAPLRCST EC * SAPLRCU2 EC SAPLRCU3 DS SAPLRCVA DS SAPLRCOC DS SAPLRCOC DS SAPLRCU4 DS * *	S QU QU S S S S	AL1 X'80' X'7F' XL2 F F H	<pre>ind1 ON = character, OFF = hexadecimal</pre>
SAPRDPRP DS SAPRRPFC DS SAPRRPLC DS SAPRRPSL DS SAPRRPU1 DS *	S S S	OXL12 F F FL1 XL3	record part
SAPODPOP DS		OXL16 F	output part from

```
SAPOOPTO DS
                                       to
SAPOOPLA DS
                                       to
SAPOOPDI DS
             FL1
* _outpart_dim_set
SAPDIMN EQU 0
                                       none
SAPDIMP EQU
             1
                                       pages
SAPDIML EQU
              2
                                       lines
SAPOOPSL DS
             FL1
* _outpart_select_set
SAPOSAL EQU 0
                                       none
SAPOSRN EQU
              1
                                       yes
              2
SAPOSLA EQU
                                       no
SAPOOPU1 DS XL2
SAPOUTPUT# EQU *-SAPSSEID
```

SJAOLCT - output structure for INFO=*LAYOUT_CONTROL

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

This macro describes the form in which the SHOPJAT macro outputs information on the page layout.

Format

Operation	Operands
SJAOLCT	VARIANT= <u>001</u> / <c-string 33=""></c-string>

Description of the operands

VARIANT=001 / <c-string 3..3>

Specifies the variant of the parameter list.

Parameter list description

* _layout_ctl_md	1		
SAL_LAYOUT_CTL_MDL	DS	0F	
SALLPGC	DS	Χ	
SALLLMG	DS	Χ	
SALLROT	DS	FL1	
<pre>* _rotation_set</pre>			
SALROTN	EQU	0	rotation=no
SALROTO	EQU	240	rotation=0
SALROT1	EQU	241	rotation=90
SALROT2	EQU	242	rotation=180
SALROT3	EQU	243	rotation=270
SALROT4	EQU	244	rotation=0-180
SALROT5	EQU	245	rotation=90-270
SALROT6	EQU	246	rotation=180-0
SALROT7	EQU	247	rotation=270-90
SALROTY	EQU	232	rotation=by-control-codes
*			
SALLTSS	DS	FL1	
<pre>* _twoside_set</pre>			
SALTWYS	EQU	1	yes
SALTWTM	EQU	2	tumble
SALTWNO	EQU	4	no
*			

SALLITR	DS	0XL1	in_tray
SALLIIN	DS	Χ	
ORG SAL	LITR		
SALLIIS	DS	FL1	
<pre>* _intray_set</pre>			
SALITBF	EQU	0	by-format
SALITIG	EQU	1	ignore
*	LQU	1	rgnore
	LITR+1		
		E. 1	
SALLIFS	DS .	FL1	
* _intray_format	_		
SALITMA	EQU	128	intray_format = *manual
SALITA3	EQU	129	intray_format = *a3
SALITA4	EQU	130	intray_format = *a4
SALITA5	EQU	131	intray_format = *a5
SALITB4	EQU	132	intray_format = *b4
SALITB5	EQU	133	intray_format = *b5
SALITDB	EQU	134	intray format =
*	LQO	101	*double-letter
SALITEX	EQU	135	intray_format = *exec
SALITEN SALITFO			
	EQU	136	intray_format = *folio
SALITIV	EQU	137	<pre>intray_format = *invoice</pre>
SALITLG	EQU	138	<pre>intray_format = *legal</pre>
SALITLT	EQU	139	intray_format = *letter
SALITMO	EQU	140	intray_format = *monarch
SALIT10	EQU	141	intray_format =
*			*commercial-10
SALITDL	EQU	142	intray_format = *dl
SALITC5	EQU	143	intray_format = *c5
SALIT3U	EQU	144	intray_format = *A3-uncut
SALIT4U	EQU	145	intray_format = *A4-uncut
SALITLD	EQU	146	intray_format = *ledger
*	LQU	140	Theray_rormat - reager
	DC	0 V I 1	0.1.++
SALLOT#	DS	0XL1	outtray
SALLOUT	DS.	Χ	
	LOT#		
SALLOTS	DS	FL1	
<pre>* _outtray_set</pre>			
SALOUTI	EQU	128	outtray = *ignore
SALOUTS	EQU	129	outtray = *sorter
*			
ORG SAL	LOT#+1		
SALLSRM	DS DS	FL1	Sorter mode
* _srtmode_set		·=	
SALSMNO	EQU	1	srtmode = *no
SALSMGR	EQU	2	srtmode = *group
SALSMCL	EQU	3	<pre>srtmode = *collate</pre>
SALSMST	EQU	4	<pre>srtmode = *stacker</pre>

SALSMAU	EQU	5	<pre>srtmode = *automatic</pre>
*			
*			
SALCOVPAGE	DS	0XL40	_covpage
SALLCHP	DS	CL32	
SALLCHE	DS	F	
SALLCTE	DS	F	
*			
SALTOIN	DS	Н	
SALLOIN	DS	Н	
SAL#	EQU	*-SALLPGC	

SJAOPJC - output structure for INFO=*PRINT_JOB_CONTROL

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

This macro describes the form in which the SHOPJAT macro outputs information on the print job control.

Format

Operation	Operands
SJAOPJC	VARIANT=001 / <c-string 33=""></c-string>

Description of the operands

VARIANT=001 / <c-string 3..3>

Specifies the variant of the parameter list.

<pre>* _print_job_ctl_md</pre>	1		
SAC_PRINT_JOB_CTL_MDL	DS	OX	
SACPJJN	DS	CL8	
SACPJPR	DS	Χ	
SACPJCL	DS	Χ	
SACPJMJ	DS	CL54	
SACPJCH	DS	FL1	
<pre>* _checkp_set</pre>			
SACCKPP	EQU	0	on pages
SACCKPR	EQU	1	on section records
*			
SACPJMP	DS	FL1	
<pre>* _yes_no_set</pre>			
SACYNN	EQU	0	
SACYNY	EQU	1	
&P.PJSD	DS	CL10	date(YYYY-MM-DD)
&P.PJST	DS	CL5	time(HH:MM)
&P.PJSS	DS	CL1	season(SUMMER/WINTER)
*			
SAC#	EQU	*-SACPJJN	

SJAORSD - output structure for INFO=*RESOURCES_DESCRIPTION

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

This macro describes the form in which the SHOPJAT macro outputs information on the resources.

Format

Operation	Operands
SJAORSD	VARIANT= <u>001</u> / <c-string 33=""></c-string>

Description of the operands

VARIANT=<u>001</u> / <c-string 3..3>

Specifies the variant of the parameter list.

Parameter list description

* _resources_descr_md	1		
SAR_RESOURCES_DESCR_MDL	DS	0 X	
SARRDFN	DS	CL6	
SARRDLN	DS	CL3	
SARRDRL	DS	CL3	
*			
SARRDCS	DS	0XL53	char_set
SARRDCN	DS	16CL3	
SARRDCN#	EQU	16	
SARRDPN	DS	CL4	
SARRDPI	DS	Χ	
*			
*			
SARRDOR	DS	0XL8	overlay_res
SARREFO	DS	CL2	
SARRFOB	DS	CL4	
*			
SARRDOV	DS	0XL2	overlay
SARRDFA	DS	Χ	
SARRDRE	DS	Χ	
4			

*			
SARPGDF	DS	CL8	
SARFMDF	DS	CL8	
SARUSRR	DS	CL44	
*			
SARTRANSL_TAB	DS	0XL52	_transl_tab
SARTRNA	DS	CL8	
SARTRFI	DS	CL44	
*			
SAR#	EQU	*-SARRDFN	

SJAOTOP - output structure for INFO=*TO_PRINTER

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

This macro describes the form in which the SHOPJAT macro outputs information on the output devices.

Format

Operation	Operands
SJAOTOP	VARIANT= <u>001</u> / <c-string 33=""></c-string>

Description of the operands

VARIANT=001 / <c-string 3..3>

Specifies the variant of the parameter list.

Parameter list description

* _to_printer_mdl		
SAT_TO_PRINTER_MDL	DS	OX
SATPRNA	DS	CL8
SATPRTY	DS	FL1
<pre>* _prtype_set</pre>		
SATPTAN	EQU	0
SATPTHP	EQU	1
SATPTAP	EQU	2
SATPT65	EQU	3
*		
SATPRVI	DS	FL1
<pre>* _virtual_set</pre>		
SATVNAL	EQU	0
SATVALL	EQU	1
SATVMUS	EQU	2
*		
SATPRPN	DS	CL8
SATPRSF	DS	CL32
SATPRTG	DS	CL8
SATUN4	DS	XL2
SAT#	EQU	*-SATPRNA

SPJARSO - RSO output structure for SHOPJAT macro

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

The SPJARSO macro describes the form in which the SHOPJAT macro outputs RSO-specific values.

Format

Operation	Operands
SPJARSO	VARIANT=001 / <c-string 33=""></c-string>

Description of the operands

VARIANT=<u>001</u> / <c-string 3..3>

Specifies the variant of the parameter list.

<pre>* _main_header_model</pre>			
RSSA_MAIN_HEADER_MODEL	DS	0F	
RSSADOPB	DS	F	displacement to document part
*			block
RSSADOFP	DS	F	displacement to document
*			format block
RSSAPJCB	DS	F	displacement to print job
*			control block
RSSALCOB	DS	F	displacement to layout
*			control block
RSSAREDB	DS	F	displacement to resource
*			description block
RSSATOPB	DS	F	displacement to printer
*			block
RSSAOUTPUT#	EQU	*-RSSADOPB	

SPJADPC - Dprint output structure for SHOPJAT macro

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

This macro describes the form in which the SHOPJAT macro outputs Dprint-specific values.

Format

Operation	Operands
SPJADPC	VARIANT=001 / <c-string 33=""></c-string>

Description of the operands

VARIANT=001 / <c-string 3..3>

Specifies the variant of the parameter list.

Parameter list description

<pre>* _main_header_model</pre>			
DPSA_MAIN_HEADER_MODEL	DS	0F	
DPSADOPB	DS	F	displacement to document part
*			block
DPSADOFP	DS	F	displacement to document
*			format block
DPSAPJCB	DS	F	displacement to print job
*			control block
DPSALCOB	DS	F	displacement to layout
*			control block
DPSAREDB	DS	F	displacement to resource
*			description block
DPSATOPB	DS	F	displacement to printer
*			block
DPSAOUTPUT#	EQU	*-DPSADOPB	

SPJASPS - SPS output structure for SHOPJAT macro

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

This macro describes the form in which the SHOPJAT macro outputs SPS-specific values.

Format

Operation	Operands
SPJASPS	VARIANT=001 / <c-string 33=""></c-string>

Description of the operands

VARIANT=001 / <c-string 3..3>

Specifies the variant of the parameter list.

* _main_header_model			
SPJA_MAIN_HEADER_MODEL	DS	0F	
SPJAD0PB	DS	F	displacement to document part
*			block
SPJAD0FP	DS	F	displacement to document
*			format block
SPJAPJCB	DS	F	displacement to print job
*			control block
SPJALCOB	DS	F	displacement to layout
*			control block
SPJAREDB	DS	F	displacement to resource
*			description block
SPJATOPB	DS	F	displacement to printer
*			block
SPJAOUTPUT#	EQU	*-SPJADOPB	

4.3 Support macro for the SHOPJST macro

The SPJSRSO macro supports use of the SHOJPST macro. It is used for setting RSO parameters for the SHOJPST macro.

SPJSRSO - define RSO parameters for SHOPJST macro

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

The SPJSRSO macro writes RSO-specific values for calling the SHOPJST macro into a parameter list.

Format

Operation	Operands
SPJSRSO	XPAND=PARAM / OUTPUT
	,VARIANT= <u>001</u> / <c-string 33=""></c-string>
	,S_RSDVT= <u>*ALL</u> / array(16): *ALL / *DJET / *2030-PCL / *4011 /

Description of the operands

XPAND=PARAM / OUTPUT

Defines the structure to be expanded (created).

XPAND=PARAM

The input parameter list layout is expanded.

XPAND=OUTPUT

All DSECTs for describing the output information blocks are generated.

VARIANT=001 / <c-string 3..3>

Specifies the variant of the parameter list.

S_RSDVT=*ALL / array(16): *ALL / *DJET / *2030-PCL / *4011 / *4813 / *4818_PCL / *4821_PCL / *4822-PCL / *4824_PCL / *4825_PCL / *4830-PCL / *4850-PCL / *8121 / *9000 / *9000-EPFX / *9000_EPLQ / *9000_EPSQ / *9000-PCL / *9000_PRO / *9000_PS / *9001 / *9001-31 / *9002 / *9003 / *9004 / *9011 / *9012 / *9013 / *9014 / *9015 / *9021 / *9022 / *9025 / *9026-PCL / *9026_RENO / *9046 / *9097 / *9645

Device types for print jobs executed via RSO, for which information is requested. Long form of the operand name "select-RSO-dev-type".

<pre>* parameter list desc RSS_MDL *</pre>	riptic DS	on OF	
RSSSELECT RSSHDR FHDR MF=(C,RS *	DS S),EQU	OXL32 JATES=NO	select standard header
RSSPRDT RSSDTB1 RSSDSU1 RSS8121 RSS8122 RSS9001 RSS9002 RSS9003	DS DS EQU EQU EQU EQU EQU EQU EQU	0XL8 AL1 X'80' X'40' X'20' X'10' X'08' X'04'	device type bytel 8121 8122 9001 9002 9003
RSS9004 RSS9025 RSSDTB2 RSS9645 RSS9031 RSS9000 RSS0131 RSS9011 RSS9012 RSS9022	EQU EQU DS EQU EQU EQU EQU EQU EQU	X'02' X'01' AL1 X'80' X'40' X'20' X'10' X'08' X'04' X'02'	9004 9025 device type byte2 9645 9013 9000 9001-31 9011 9012 9022
RSS4822 RSSDTB3 RSS4825 RSS9015 RSS9045 RSSDSU2 RSS9046 RSSDSU3 RSS90PS RSS9PRO	EQU DS EQU EQU EQU EQU EQU EQU EQU	X'01' AL1 X'80' X'40' X'20' X'10' X'08' X'04' X'02' X'01'	4822-PCL device type byte3 4825-PCL 9015 9045-ANSI 9046 9000-PS 9000-PRO

RSS#

RSSDTB4	DS	AL1	device type byte4
RSS9PFX	EQU	X'80'	9000-EPFX
RSS9PLQ	EQU	X'40'	9000-EPLQ
RSS9PSQ	EQU	X'20'	9000-EPSQ
RSS4818	EQU	X'10'	4818-PCL
RSSDSU4	EQU	X'OE'	
RSS9014	EQU	X'01'	9014
RSSDTB5	DS	AL1	device type byte5
RSS9021	EQU	X'80'	9021
RSS9222	EQU	X'40'	9022-200
RSS9223	EQU	X'20'	9022-300
RSS9PCL	EQU	X'10'	9000-PCL
RSSDJET	EQU	X'08'	DJET
RSS4824	EQU	X'04'	4824-PCL
RSSRENO	EQU	X'02'	9026-RENO
RSS6PCL	EQU	X'01'	9026-PCL
RSSDTB6	DS	AL1	device type byte6
RSS4812	EQU	X'80'	4812
RSS9097	EQU	X'40'	9097
RSS4011	EQU	X'20'	4011
RSS4813	EQU	X'10'	4813
RSS4821	EQU	X'08'	4821-PCL
RSS2030	EQU	X'04'	2030-PCL
RSSDSU5	EQU	X'03'	
RSS4850	EQU	x'01'	4850-PCL
RSSDTB7	DS	AL1	device type byte7
RSSD4830	EQU	x'80'	4830-PCL
RSSDSU7	EQU	X'FF'	
RSSDTB8	DS	AL1	device type byte8
RSSDSU8	EQU	X'FF'	
*			
RSSUN1	DS	XL16	
*			

U41003-J-Z125-3-76

EQU *-RSSHDR

4.4 Output structures of the SHOPJST macro

The following macros describe the structures with which the SHOPJST macro output buffer is filled after a status query.

"SJSODES - output structure for DESTINATION" on page 375

Describes the structure with which the SHOPJST macro output buffer is filled after a query with INFO=*DESTINATION.

"SJSODES - output structure for SPOOL-FILTER" on page 378

Describes the structure with which the SHOPJST macro output buffer is filled after a query with INFO=*SPOOL-FILTER.

"SJSODES - output structure for RSO-FILTER" on page 381

Describes the structure with which the SHOPJST macro output buffer is filled after a query with INFO=*RSO-FILTER.

"SJSODIS - output structure for DISTRIBUTED" on page 384

Describes the structure with which the SHOPJST macro output buffer is filled after a query with INFO=*DISTRIBUTED.

"SJSODSS - output structure for DESTINATION and UNIX systems" on page 386

Describes the structure with which the SHOPJST macro is supplemented if a query with INFO=*DESTINATION concerned a UNIX system based cluster.

"SJSOORI - output structure for ORIGIN" on page 388

Describes the structure with which the SHOPJST macro output buffer is filled after a query with INFO=*ORIGIN.

"SJSOORS - output structure for ORIGIN and UNIX systems" on page 390

Describes the structure with which the SHOPJST macro is supplemented if a query with INFO=*ORIGIN concerned a UNIX system based cluster.

"SJSOTRA - output structure for TRAITS" on page 391

Describes the structure with which the SHOPJST macro output buffer is filled after a query with INFO=*TRAITS.

"SJSOTRS - output structure for TRAITS and UNIX systems" on page 393

Describes the structure with which the SHOPJST macro is supplemented if a query with INFO=*TRAITS concerned a UNIX system based cluster.

SJSODES - output structure for DESTINATION

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

This macro describes the structure of the SHOPJST macro output buffer if the status of print jobs is queried with the INFO=*DESTINATION operand. If the query concerns a UNIX system based cluster, the output buffer is supplemented with the SJSODSS structure. See "SJSODSS - output structure for DESTINATION and UNIX systems" on page 386.

Note

For print jobs, for which a scheduling time was specified, a new display box will inform users of a delay. If the print job is filtered, the display box will tell users, that the job can be converted.

Format

Operation	Operands
SJSODES	VARIANT= <u>001</u> / <c-string 33=""></c-string>

Description of the operands

VARIANT=<u>001</u> / <c-string 3..3>

Specifies the variant of the parameter list.

* _out_destination_mc SSD_OUT_DESTINATION_MDL SSDELEN SSDETSN SSDESRV SSDEMOD * _out_mode_set SSDOMLO		OH H CL4 CL8 FL1	block length tsn server name
SSDOMRS *	EQU	2	
<pre>SSDESTA * _out_job_status_set</pre>	DS ;	FL1	
SSDSWTF SSDSFT SSDSWTP SSDSPRE SSDSWT SSDSTRD SSDSACT SSDSTRT SSDSTRT SSDSKP SSDSDIR SSDSREL *	EQU EQU EQU EQU EQU EQU EQU EQU EQU EQU	1 2 3 4 5 6 7 8 9 10 255	replay in direct mode
SSDERPL SSDERPC SSDEUN1 SSDEDVN SSDEDST SSDEECD SSDEEMS *	DS EQU EQU DS DS DS DS	AL1 X'80' X'7F' CL8 CL8 CL8 CL8	sect_ind ON if replay in copy mode not used device name destination name
SSDDESDT SSDDTB1 SSDDLP SSDDND SSDDHP SSDDLP48 SSDDLP65 SSDDTAPE SSDDLPEM SSDDTB2 SSDDDSU1 SSDDHP90 SSDDAP50 SSDDAP90	DS DS EQU EQU EQU EQU EQU EQU EQU EQU EQU DS EQU EQU EQU EQU EQU	OXL8 AL1 X'80' X'40' X'20' X'10' X'08' X'04' X'01' AL1 X'C0' X'20' X'10' X'10'	device type byte1 LP printer ND printer HP printer LP48 printer LP65 printer TAPE LP-EMUL printer device type byte2 HP90 printer APA-2050 printer APA-2090 printer

SSDDTW90	EQU	X'04'	TWIN-2090 printer
SSDDDSU2	EQU	X'03'	
SSDDDTB3	DS	AL1	device type byte3
SSDDDSU3	EQU	X'FF'	
SSDDDTB4	DS	AL1	device type byte4
SSDDPCL	EQU	X'80'	PCL printer
SSDDDSU4	EQU	X'7F'	
SSDDDTB5	DS	AL1	device type byte5
SSDDDSU5	EQU	X'FF'	
SSDDDTB6	DS	AL1	device type byte6
SSDDDSU6	EQU	X'FF'	
SSDDDTB7	DS	AL1	device type byte7
SSDDDSU7	EQU	X'FF'	
SSDDDTB8	DS	AL1	device type byte8
SSDDDSU8	EQU	X'F0'	
SSDDVIRT	EQU	X'08'	VIRTUAL printer
SSDDDSU9	EQU	X'07'	
*			
SSDEIFG	DS	AL1	info flag
SSDEDEL	EQU	X'80'	delayed job
SSDEFLT	EQU	X'40'	filtrable job
SSD#	EQU	*-SSSLEN*	

SJSODES - output structure for SPOOL-FILTER

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

This macro describes the structure of the SHOPJST macro output buffer if the status of print jobs is queried with the INFO=*SPOOL-FILTER operand.

Note

For print jobs, for which a scheduling time was specified, a new display box will inform users of a delay. If the print job is filtered, the display box will tell users, that the job can be converted.

Format

Operation	Operands
SJSODES	VARIANT= <u>001</u> / <c-string 33=""></c-string>

Description of the operands

VARIANT=001 / <c-string 3..3>

Describes the variant of the generated parameter list.

Parameter list description

* _out_destination_mc SSD_OUT_DESTINATION_MDL SSDELEN SSDETSN SSDESRV SSDEMOD * _out_mode_set SSDOMLO	DS DS DS DS DS DS	0H H CL4 CL8 FL1	block length tsn server name
SSDOMRS * SSDESTA	EQU DS	2 FL1	
* _out_job_status_set SSDSWTF SSDSFT SSDSWTP SSDSPRE SSDSWT SSDSTRD SSDSACT SSDSTRT SSDSKP SSDSDIR SSDSREL *	EQU EQU EQU EQU EQU EQU EQU EQU EQU EQU	1 2 3 4 5 6 7 8 9 10 255	replay in direct mode
SSDERPL SSDERPC SSDEUN1 SSDEDVN SSDEDST SSDEECD SSDEEMS	DS EQU EQU DS DS DS	AL1 X'80' X'7F' CL8 CL8 CL8	sect_ind ON if replay in copy mode not used device name destination name
SSDDESDT SSDDDTB1 SSDDLP SSDDND SSDDHP SSDDLP48 SSDDLP65 SSDDTAPE SSDDLPEM SSDDDTB2 SSDDDSU1 SSDDHP90 SSDDAP50 SSDDAP90	DS DS EQU	OXL8 AL1 X'80' X'40' X'20' X'10' X'08' X'04' X'01' AL1 X'C0' X'20' X'10' X'20'	device type byte1 LP printer ND printer HP printer LP48 printer LP65 printer TAPE LP-EMUL printer device type byte2 HP90 printer APA-2050 printer APA-2090 printer

SSDDTW90	EQU	X'04'	TWIN-2090 printer
SSDDDSU2	EQU	X'03'	
SSDDDTB3	DS	AL1	device type byte3
SSDDDSU3	EQU	X'FF'	
SSDDDTB4	DS	AL1	device type byte4
SSDDPCL	EQU	X'80'	PCL printer
SSDDDSU4	EQU	X'7F'	
SSDDDTB5	DS	AL1	device type byte5
SSDDDSU5	EQU	X'FF'	
SSDDDTB6	DS	AL1	device type byte6
SSDDDSU6	EQU	X'FF'	
SSDDDTB7	DS	AL1	device type byte7
SSDDDSU7	EQU	X'FF'	
SSDDDTB8	DS	AL1	device type byte8
SSDDDSU8	EQU	X'F0'	
SSDDVIRT	EQU	X'08'	VIRTUAL printer
SSDDDSU9	EQU	X'07'	
*			
SSDEIFG	DS	AL1	info flag
SSDEDEL	EQU	X'80'	delayed job
SSDEFLT	EQU	X'40'	filtrable job
SSD#	EQU	*-SSSLEN*	

SJSODES - output structure for RSO-FILTER

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

This macro describes the structure of the SHOPJST macro output buffer if the status of print jobs is queried with the INFO=*RSO-FILTER operand.

Note

For print jobs, for which a scheduling time was specified, a new display box will inform users of a delay. If the print job is filtered, the display box will tell users, that the job can be converted.

Format

Operation	Operands
SJSODES	VARIANT=001 / <c-string 33=""></c-string>

Description of the operands

VARIANT=001 / <c-string 3..3>

Describes the variant of the generated parameter list.

* _out_destination_m SSD_OUT_DESTINATION_MDL SSDELEN SSDETSN SSDESRV SSDEMOD * _out_mode_set	DS DS DS DS DS	OH H CL4 CL8 FL1	block length tsn server name
SSDOMLO SSDOMRS *	EQU EQU	1	
<pre>SSDESTA * _out_job_status_set</pre>	DS	FL1	
SSDSWTF SSDSFT SSDSWTP SSDSPRE SSDSWT SSDSTRD SSDSACT SSDSTRT SSDSTRT SSDSKP SSDSDIR SSDSREL *	EQU EQU EQU EQU EQU EQU EQU EQU EQU EQU	1 2 3 4 5 6 7 8 9 10 255	replay in direct mode
SSDERPL SSDERPC SSDEUN1 SSDEDVN SSDEDST SSDEECD SSDEEMS *	DS EQU EQU DS DS DS DS	AL1 X'80' X'7F' CL8 CL8 CL8 CL8	sect_ind ON if replay in copy mode not used device name destination name
SSDDESDT SSDDDTB1 SSDDLP SSDDND SSDDHP SSDDLP48 SSDDLP65 SSDDTAPE SSDDLPEM SSDDTB2 SSDDDSU1 SSDDHP90 SSDDAP50 SSDDAP90	DS EQU	OXL8 AL1 X'80' X'40' X'20' X'10' X'08' X'04' X'01' AL1 X'C0' X'20' X'10' X'10' X'20'	device type byte1 LP printer ND printer HP printer LP48 printer LP65 printer TAPE LP-EMUL printer device type byte2 HP90 printer APA-2050 printer APA-2090 printer

SSDDTW90	EQU	X'04'	TWIN-2090 printer
SSDDDSU2	EQU	X'03'	
SSDDDTB3	DS	AL1	device type byte3
SSDDDSU3	EQU	X'FF'	
SSDDDTB4	DS	AL1	device type byte4
SSDDPCL	EQU	X'80'	PCL printer
SSDDDSU4	EQU	X'7F'	
SSDDDTB5	DS	AL1	device type byte5
SSDDDSU5	EQU	X'FF'	
SSDDDTB6	DS	AL1	device type byte6
SSDDDSU6	EQU	X'FF'	
SSDDDTB7	DS	AL1	device type byte7
SSDDDSU7	EQU	X'FF'	
SSDDDTB8	DS	AL1	device type byte8
SSDDDSU8	EQU	X'F0'	
SSDDVIRT	EQU	X'08'	VIRTUAL printer
SSDDDSU9	EQU	X'07'	
*			
SSDEIFG	DS	AL1	info flag
SSDEDEL	EQU	X'80'	delayed job
SSDEFLT	EQU	X'40'	filtrable job
SSD#	EQU	*-SSSLEN	

SJSODIS - output structure for DISTRIBUTED

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

This macro describes the structure of the SHOPJST macro output buffer if the status of print jobs is queried with the operand INFO=*DISTRIBUTED.

Note

For print jobs, for which a scheduling time was specified, a new display box will inform users of a delay. If the print job is filtered, the display box will tell users, that the job can be converted.

Format

Operation	Operands
SJSODIS	VARIANT=001 / <c-string 33=""></c-string>

Description of the operands

VARIANT=<u>001</u> / <c-string 3..3>

Specifies the variant of the parameter list.

Parameter list description

<pre>* _out_distributed_md</pre>	1		
SSI_OUT_DISTRIBUTED_MDL	DS	OH	
SSIILEN	DS	Н	block length
SSIITSN	DS	CL4	tsn
SSIISRV	DS	CL8	server name
SSIIPTS	DS	CL4	tsn
SSIIJTY	DS	FL1	
<pre>* _dist_job_type_set</pre>			
SSIIJTL	EQU	1	job-type = local
SSIIJTD	EQU	2	job-type = dist
SSIIJTI	EQU	3	job-type = inter
*			
SSIIHST	DS	CL8	client hostname
SSIIUID	DS	CL8	user identification
SSIIACC	DS	CL8	account number
SSIIECD	DS	CL8	
SSIIEMS	DS	CL8	
*			
SSIIIFG	DS	AL1	info flag
SSIIDEL	EQU	X'80'	delayed job
SSIIFLT	EQU	X'40'	filtrable job
SSD#	EQU	*-SSSLEN	

SJSODSS - output structure for DESTINATION and UNIX systems

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

This macro describes the structure with which the SHOPJST macro is extended if the query with the operand INFO=*DESTINATION concerns a cluster in a UNIX system.

Format

Operation	Operands
SJSODSS	VARIANT= <u>001</u> / <c-string 33=""></c-string>

Description of the operands

VARIANT=001 / <c-string 3..3>

Specifies the variant of the parameter list.

Parameter list description

<pre>* _out_destination_sinix_md</pre>	1		
SID_OUT_DESTINATION_SINIX_MDL	DS	OH	
SIDELEN	DS	Н	block length
SIDESSN	DS	CL8	server name
SIDESST	DS	FL1	
<pre>* _out_job_status_set</pre>			
SIDSWTF	EQU	1	
SIDSFT	EQU	2	
SIDSWTP	EQU	3	
SIDSPRE	EQU	4	
SIDSWT	EQU	5	
SIDSTRD	EQU	6	
SIDSACT	EQU	7	
SIDSTRT	EQU	8	
SIDSKP	EQU	9	
SIDSDIR	EQU	10	replay in direct mode
SIDSREL	EQU	255	
*			
SIDESDN	DS	CL8	device name
SIDESID	DS	CL255	
*			identification
SIDESUN	DS	XL2	
SID#	EQU	*-SID	DELEN

SJSOORI - output structure for ORIGIN

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

This macro describes the structure of the SHOPJST macro output buffer if the status of print jobs is queried with the operand INFO=*ORIGIN. If the query concerns a cluster in a UNIX system, the output buffer is supplemented with the SJSOORS structure. See "SJSOORS output structure for ORIGIN and UNIX systems" on page 390.

Format

Operation	Operands
SJSOORI	VARIANT=001 / <c-string 33=""></c-string>

Description of the operands

VARIANT=001 / <c-string 3..3>

Specifies the variant of the parameter list.

<pre>* _out_origin_md</pre>	1		
SSO_OUT_ORIGIN_MDL	DS	OH	
SS00LEN	DS	Н	block length
SS00TSN	DS	CL4	tsn
SS00SRV	DS	CL8	server name
SS00SPN	DS	CL8	spoolout name
SS00RTS	DS	CL4	tsn of the requestor
SS00HST	DS	CL8	client hostname
SS00UID	DS	CL8	user identification
SS00ACC	DS	CL8	account number
SS00FC0	DS	Χ	number of copies asked
SS00PC0	DS	Χ	number of page copies asked
SS00FTY	DS	FL1	file type
<pre>* _out_file_type</pre>	_set		
SSODMS	EQU	1	
SS0EAMF	EQU	2	
SS00MF	EQU	3	
SSOPLAM	EQU	4	
SSOLST	EQU	5	
SS00PT	EQU	6	
SS00UT	EQU	7	

SSOTMP	EQU	8	
SSOUFS	EQU	9	
*			
SS00FCB	DS	FL1	FCB type
<pre>* _out_file_type</pre>	_set		
SSOSAM	EQU	0	
SSOTAPE	EQU	1	replay file
SSOEAM	EQU	2	
SSOISAM	EQU	64	
SSOBTAM	EQU	128	
SSOPAM	EQU	192	
*			
SS00FSZ	DS	Н	file size in PAM-pages
SSO#	EQU	*-SSOOLEN	

SJSOORS - output structure for ORIGIN and UNIX systems

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

This macro describes the structure with which the SHOPJST is extended if the query with the operand INFO=*ORIGIN concerns a cluster in a UNIX system.

Format

Operation	Operands
SJSOORS	VARIANT= <u>001</u> / <c-string 33=""></c-string>

Description of the operands

VARIANT=<u>001</u> / <c-string 3..3>

Specifies the variant of the parameter list.

<pre>* _out_origin_sinix_md</pre>	1		
SIO_OUT_ORIGIN_SINIX_MDL	DS	0H	
SIOOLEN	DS	Н	block length
SIOOSRV	DS	CL8	server name
SIOOSSN	DS	CL8	spoolout name
SIOOSHN	DS	CL8	client hostname
SIOOSUI	DS	CL8	user identification
SIOOSFC	DS	Χ	number of copies asked
SIOOSPC	DS	Χ	number of page copies asked
SIOOSID	DS	CL255	identification
SIOUN1	DS	XL1	
SIO#	EQU	*-SIOOLEN	

SJSOTRA - output structure for TRAITS

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

This macro describes the structure of the SHOPJST macro output buffer if the status of print jobs is queried with the operand INFO=*TRAITS. If the query concerns a cluster in a UNIX system, the output buffer is supplemented with the SJSOTRS structure. See "SJSOTRS output structure for TRAITS and UNIX systems" on page 393.

Format

Operation	Operands
SJSOTRA	VARIANT= <u>001</u> / <c-string 33=""></c-string>

Description of the operands

VARIANT=001 / <c-string 3..3>

Specifies the variant of the parameter list.

Parameter list description

<pre>* _out_traits_md</pre>	17		
SST_OUT_TRAITS_MDL	DS	OH	
SSTTLEN	DS	Н	block length
SSTTTSN	DS	CL4	tsn
SSTTSRV	DS	CL8	server name
SSTTPRI	DS	Χ	priority
SSTTFRM	DS	CL8	form name
SSTTCLA	DS	Χ	job class
SSTTCSN	DS	Χ	character set number
SSTTUN1	DS	XL1	
SSTTFOB	DS	Н	fob size
SSTTEF0	DS	CL2	form overlay
SSTTROT	DS	FL1	rotation
<pre>* _out_rotation_</pre>	set		
SSTOR0	EQU	0	rotation = 0
SSTOR90	EQU	1	rotation = 90
SSTOR18	EQU	2	rotation = 180
SSTOR27	EQU	3	rotation = 270
SSTOR01	EQU	4	rotation = 0
SSTOR92	EQU	5	rotation = 90
SSTOR10	EQU	6	rotation = 180

SSTOR29	EQU	7	rotation = 270
SSTORY	EQU	8	rotation = *yes
SSTORNO	EQU	9	rotation = *no
*			
SSTTCTL	DS	FL1	control output
* _out_control	l_set		
SSTOCAP	EQU	2	control = apa
SSTOCTR	EQU	4	control=transparent
SSTOCLI	EQU	8	control = line
SSTPCLG	EQU	32	control = logical
SSTOCPH	EQU	64	control = physical
SSTOCNO	EQU	128	control = no
*			
SSTTRES	DS	AL1	resource_location
SSTTRSV	EQU	X'80'	ON : resource on server - OFF
*			: resource on client
SSTTUN2	EQU	X'7F'	not used
SSTTUN3	DS	XL1	
SST#	EQU	*-SSTTLEN	

SJSOTRS - output structure for TRAITS and UNIX systems

User group: Nonprivileged users **Programming language:** Assembler

Macro type: S

This macro describes the structure with which the SHOPJST is extended if the query with the operandINFO=*TRAITS concerns a cluster in a UNIX system.

Format

Operation	Operands
SJSOTRS	VARIANT= <u>001</u> / <c-string 33=""></c-string>

Description of the operands

VARIANT=001 / <c-string 3..3>

Specifies the variant of the parameter list.

Parameter list description

<pre>* _out_traits_sinix_md</pre>	1		
SIT_OUT_TRAITS_SINIX_MDL	DS	OH	
SITTSLE	DS	Н	block length
SITTSSN	DS	CL8	server name
SITTSPR	DS	Χ	number of copies asked
SITTSFN	DS	CL8	form name
SITTSRT	DS	FL1	rotation
<pre>* _out_rotation_set</pre>			
SITORO	EQU	0	rotation = 0
SITOR90	EQU	1	rotation = 90
SITOR18	EQU	2	rotation = 180
SITOR27	EQU	3	rotation = 270
SITORO1	EQU	4	rotation = 0
SITOR92	EQU	5	rotation = 90
SITOR10	EQU	6	rotation = 180
SITOR29	EQU	7	rotation = 270
SITORY	EQU	8	rotation = *yes
SITORNO	EQU	9	rotation = *no
*			
SITTSID	DS	CL255	identification
SITTUN3	DS	XL1	
SIT#	EQU	*-SITTSLE	

5 Macros for managing printers

A program interface comprising five macros is provided for managing printers. The macros offer the same functionalities as the corresponding commands:

- SASDSPO(x) corresponds to the SHOW-ACTIVE-SPOOL-DEVICES command.
 See "SASDSPO requesting information on active spool devices" on page 396
- STPSPPR(x) corresponds to the STOP-PRINTER-OUTPUT command.
 See "STPSPPR cancel printer allocation" on page 435
- STRSPPR(x) corresponds to the START-PRINTER-OUTPUT command.
 See "STRSPPR allocate print jobs to printers" on page 440
- STRSRSO(x) corresponds to the RSO-specific operands used with the STRSPPR macro. See "STRSRSO - allocate printers with RSO-specific parameters" on page 457
- SUPSPPR(x) corresponds to the MODIFY-PRINTER-OUTPUT-STATUS command.
 See "SUPSPPR modify administration parameters for printers" on page 460

(x) is the supplement for the programming language used (a blank stands for Assembler, "C" for CPP and "H" for C).

SASDSPO - requesting information on active spool devices

User group: Nonprivileged users, RSO device administrators, spool administrators, cluster administrators

Programming language: Assembler, CPP, C

Macro type: M, S

This macro supplies information on the printers specified (e.g. on printers for which a START-PRINTER-OUTPUT command was issued). The printer for which information is requested may be located both within or outside the cluster configuration to which the user belongs.

Format

Operation	Operands
SASDSPO	ENTRY = <u>YES</u> / NO
	XPAND = PARAM / OUTPUT
	VARIANT = <c-string-without-quotes 33=""> / <u>001</u></c-string-without-quotes>
	CALLER = *USER / *SYSTEM
	INFO = <var: _information_set:1="" enum-of=""> / *SUMMARY / *ALL</var:>
	DPCPAR = <var: pointer=""> / *NONE</var:>
	RSOPAR = <var: pointer=""> / *NONE</var:>
	SPSPAR = <var: pointer=""> / *NONE</var:>
	DSEMPAR = <var: pointer=""> / *NONE</var:>
	BUFFER = <var: pointer=""> / *NONE</var:>
	BUFLEN = <var: int:4=""> / <integer 18388607=""> / *NONE</integer></var:>

(part 1 of 4)

Operation	Operands
SASDSPO	SCHEDST = <var: _sched_state_set:1="" enum-of=""> / *NEXT / *CURRENT</var:>
	ROT = *ANY / *YES / *NO / *MANUAL
	TWO_UP = *ANY / *YES / *NO / *MODE1 / *MODE2
	EXITR = *ANY / *YES / *NO
	STATUS = *ALL / *ATTACHED / *IDLE / *TRANSFER / *RUN / *MESSAGE / *STOPPED / *WORKED / *DETACHED / *UNKNOWN
	ERR_TYP = <var: _flag_set:1="" enum-of=""> / *ALL / *LIST / *WILD / *ANY / *NONE / *CLIST</var:>
	ERR_MSG = array(1): <var: char:8=""> / <c-string 1-8="" 18:="" c-string=""> / *ALL</c-string></var:>
	EMSG_W = <var: char:24=""> / <c-string 1-24="" 124:="" c-string=""> / *ALL</c-string></var:>
	DVC_TYP = <var: _flag_set:1="" enum-of=""> / *LIST / *WILD / <u>*ALL</u> / *CLIST</var:>
	DVCNAME = array(8): <var: char:8=""> / <c-string 1-8="" 18:="" c-string=""> / *ALL</c-string></var:>
	DVCN_W = <var: char:24=""> / <c-string 1-24="" 124:="" c-string=""> / *ALL</c-string></var:>
	DES_TYP = <var: _flag_set:1="" enum-of=""> / *LIST / *LOCAL / *REMOTE / *PUBLIC / *ALL</var:>
	DESNAME = array(16): <var: char:8=""> 7 <c-string 1-8="" 18:="" c-string=""> / *ALL</c-string></var:>
	FRM_TYP = <var: _flag_set:1="" enum-of=""> / *LIST / *STD 7 *EXCEPT / *ALL</var:>

(part 2 of 4)

Operation	Operands
SASDSPO	FRMNAME = array(16): <var: char:6=""> / <c-string 1-6="" 16:="" c-string=""> / *ALL</c-string></var:>
	USR_TYP = <var: _flag_set:1="" enum-of=""> / *LIST / *EXCEPT / *ALL / *CLIST</var:>
	USRNAME = array(16): <var: char:8=""> / <c-string 1-8="" 18:="" c-string=""> / *ALL</c-string></var:>
	CLS_TYP = <var: _flag_set:1="" enum-of=""> / *LIST / *EXCEPT / *ALL</var:>
	CLASS = array(16): <var: int:2=""> / <integer 1255=""> / *ALL</integer></var:>
	SPN_TYP = <var: _flag_set:1="" enum-of=""> / *LIST / *EXCEPT / *ALL / *CLIST</var:>
	SPNAME = array(16): <var: char:8=""> / <c-string 1-8="" 18:="" c-string=""> / *ALL</c-string></var:>
	ACC_TYP = <var: _flag_set:1="" enum-of=""> / *LIST / *EXCEPT / *ALL</var:>
	ACCOUNT = array(16): <var: char:8=""> / <c-string 1-8="" 18:="" c-string=""> / *ALL</c-string></var:>
	OVL_TYP = <var: _flag_set:1="" enum-of=""> / *LIST / *NONE / *ONLY / *EXCEPT / *ALL</var:>
	OVERLAY = array(16): <var: char:2=""> / <c-string 2-2="" 22:="" c-string=""> / *ALL</c-string></var:>
	FOB = (type, low, high) type: *ANY / *ONLY / *NO / *RANGE low: <var: int:2=""> / <integer 032767=""> / *MIN high: <var: int:2=""> / <integer 032767=""> / *STD</integer></var:></integer></var:>

(part 3 of 4)

Operation	Operands
SASDSPO	PRIO = (type, low, high) type: *ALL / *RANGE from: <var: int:1=""> / <integer 30255=""> / *MIN to: <var: int:1=""> / <integer 30255=""> / *MAX</integer></var:></integer></var:>
	CSN = (type, low, high) type: *ALL / *ONE / *RANGE low: <var: int:2=""> / <integer 132767=""> / *MIN high: <var: int:2=""> / <integer 132767=""> / *MAX</integer></var:></integer></var:>
	SRV_TYP = <var: _flag_set:1="" enum-of=""> / *LIST / *WILD / *ALL /</var:>
	SRW_WLD = <var: char:24=""> / <c-string 1-24="" 124:="" c-string=""> / *ALL HST_TYP = <var: _flag_set:1="" enum-of=""> / *LIST / *LOCAL_CLUSTER / *ALL / *HOME / *EXCEPT / *CLIST</var:></c-string></var:>
	HSTNAME = array(16): <var: char:8=""> / <c-string 1-8="" 18:="" c-string=""> / *ALL</c-string></var:>
	CLUSTER = <var: char:8=""> / <c-string 18:="" name_18=""> / *LOCAL_CLUSTER</c-string></var:>

(part 4 of 4)

Description of the operands

In the following text, a spool devices can either be a local device (distributed or central) or an RSO device.

ENTRY=YES/NO

Specifies whether the relevant entries are generated or not.

ENTRY=YES

Corresponding entries are generated.

ENTRY=NO

Corresponding entries are not generated.

XPAND=PARAM / OUTPUT

Specifies which structure is to be expanded (generated).

XPAND=PARAM

The layout of the input parameter list is expanded.

XPAND=OUTPUT

All data descriptions (DSECTs) for the output information blocks are generated.

VARIANT=001 / <c-string without quotes 3..3>

The variant of the generated parameter list.

CALLER=*USER / *SYSTEM

Macro caller.

CALLER=*USER

Operand default value. The macro is called by a user.

CALLER=*SYSTEM

Operand default value. The macro is called by the system.

INFO=<var: enum-of _information_set:1> / *SUMMARY / *ALL

Selects the information to be output.

INFO=<var: enum-of information set:1>

The information is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

0 * SUMMARY

1 *ALL

INFO=*SUMMARY

Only one line is output for each element.

INFO=*ALL

All lines are output for each element.

DPCPAR=*NONE/<var: pointer>

The address of a parameter list can be specified which contains DPRINT-specific values for calling the macro.

DPCPAR=*NONE

Operand default value. The values are not to be supplied from a parameter list.

DPCPAR=<var: pointer>

A pointer is defined, i.e. the contents of the variable or field is not the DPRINT parameter list itself, but the address of the location at which the DPRINT parameter list has been stored (A(field)).

RSOPAR=*NONE / <var: pointer>

The address of a parameter list can be specified, in which the RSO-specific values for calling the macro are stored.

RSOPAR=*NONE

Operand default value. The values are not to be supplied from a parameter list.

RSOPAR=<var: pointer>

A pointer is defined, i.e. the contents of the variable or field is not the RSO parameter list itself, but the address of the location at which the RSO parameter list has been stored (A(field)).

SPSPAR=*NONE/<var: pointer>

The address of a parameter list can be specified, in which the SPS-specific values for calling the macro are stored.

SPSPAR=*NONE

Operand default value. The values are not to be supplied from a parameter list.

SPSPAR=<var: pointer>

A pointer is defined, i.e. the contents of the variable or field is not the SPS parameter list itself, but the address of the location at which the SPS parameter list has been stored (A(field)).

DSEMPAR=*NONE/<var: pointer>

The address of a parameter list can be specified, in which the DSEM-specific values for calling the macro are stored.

DSEMPAR=*NONE

Operand default value. The values are not to be supplied from a parameter list.

DSEMPAR=<var: pointer>

A pointer is defined, i.e. the contents of the variable or field is not the DSEM parameter list itself, but the address of the location at which the DSEM parameter list has been stored (A(field)).

BUFFER=<var: pointer>

Specifies a pointer to the output buffer. The size of the buffer also has to be specified (see BUFLEN). It the buffer is too small, an error message will be returned.

BUFFER=*NONE

Operand default value. No output buffer is specified, therefor *NONE has to be specified for BUFLEN. In this case, Spool & Print generates a buffer in a class 6 memory. The buffer address is returned to the input parameter list, as is also the buffer length. In this case the buffer has to be released by the caller.

BUFLEN=<var: int: 4> / <integer 1..8388607>

Output buffer size.

BUFLEN=<var: int: 4>

The size of the output buffer is not specified directly through the operand value, but rather indirectly through a field.

BUFLEN=<integer 1..8388607>

The size of the output buffer is specified directly.

BUFLEN=*NONE

Operand default value. The size of the output buffer is not specified.

SCHEDST=*NEXT/*CURRENT/<var: enum-of _sched_state_set:1>

Defines the entry type for the scheduling status.

SCHEDST=*NEXT

Operand default value. The next scheduling operation is selected.

SCHEDST=*CURRENT

The current scheduling operation is selected.

SCHEDST=<var: enum-of _sched_state_set:1>

The scheduling status is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. The following relationships exist between the values and the desired functions:

- 1 *CURRENT
- 2 *NEXT

ROT=*ANY/*YES/*NO/*MANUAL

Information on an active spool device is output according to the entry made for the ROTATION operand of the START-PRINTER-OUTPUT, START-TAPE-OUTPUT or START-TAPE-REPLAY command.

ROT=*ANY

Operand default value. Information on active spool devices is issued, regardless of whether print jobs use the page rotation module or not.

ROT=*YES

Information on active spool devices is issued, at which print jobs use the page rotation module.

ROT=*NO

Information on active spool devices is issued, at which print jobs do not use the page rotation module.

ROT=*MANUAL

Information on active spool devices is issued, at which print jobs access the page rotation module manually.

TWO UP=*ANY/*YES/*NO/*MODE1/*MODE2

Outputs information on printers for which the relevant TWO-UP-PROCESSING value is specified in the START-PRINTER-OUTPUT command.

For further information refer to the "SPOOL (BS2000/OSD)" manual.

EXITR=*ANY/*YES/*NO

Information on active spool devices is issued according to the entry made for the EXIT-ROUTINE operand in the START-PRINTER-OUTPUT command.

EXITR=*ANY

Operand default value. Information on active spool devices is issued, regardless of whether they use the exit routine or not.

EXITR=*YES

Information on active spool devices which use the exit routine is issued.

EXITR=*NO

Information on active spool devices which do not use the exit routine is issued.

STATUS=*ALL/*ATTACHED/*IDLE/*TRANSFER/*RUN/*MESSAGE/*STOPPED/ *WORKED/*DETACHED/*UNKNOWN

Outputs information on active spool devices where the status has the specified value.

ERR_TYP=<var: enum-of _flag_set:1> / *ALL / *LIST / *WILD / *ANY / *NONE / *CLIST Defines the entry type for selecting the error message operand.

ERR_TYP=<var: enum-of _flag_set:1>

The entry type for ERR_MSG and EMSG_W is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. Refer to the table under DVC_TYP for the meaning of the values.

ERR TYP=*ALL

All error messages are valid.

ERR TYP=*LIST

Specifies a list of error messages.

ERR TYP=*WILD

The error messages are partially qualified using wildcards.

ERR_TYP=*ANY

Operand default value. All active spool devices are selected, regardless of whether they have an error message or not.

ERR TYP=*NONE

Outputs information on active spool devices for which there is no error message.

ERR TYP=*CLIST

All the elements of the error message list must be of type c-string, with a maximum length of 8 characters (length estimated without quotes).

ERR_MSG=*ALL/array(1): <var: char:8>/<c-string 1..8: c-string 1-8>

Outputs information on active spool devices for which there is an error message.

ERR MSG=*ALL

Operand default value. Outputs information on active spool devices for all error messages.

ERR_MSG=array(1): <var: char:8>

Outputs information on active spool devices for the error messages specified in the list.

ERR_MSG=<c-string 1..8: c-string 1-8>

Displays information on active spool devices for the error message specified.

EMSG_W=*ALL/<var: char:24>/<c-string 1..24: c-string 1-24>

Displays information on active spool devices for the error messages specified. Wildcards may be used.

EMSG_W=*ALL

Operand default value. Outputs information on active spool devices for all error messages.

EMSG_W=<var: char:24>/<c-string 1..24: c-string 1-24>

The error messages are partially qualified using wildcards.

DVC_TYP=<var: enum-of _flag_set:1> / *LIST / *WILD /<u>*ALL</u> / *CLIST

Specifies the entry type for the device name.

DVC_TYP=<var: enum-of _flag_set:1>

The entry type for DVCN_NAME and DVCN_W is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. Refer to the table under DVC_TYP for the meaning of the values.

DVC TYP=*LIST

Specifies a device type list.

DVC TYP=*WILD

The device names are partially qualified using wildcards.

DVC_TYP=*ALL

Operand default value. All device names are valid.

DVC_TYP=*CLIST

All the elements of the device type list must be of type *c-string*, with a maximum length of 8 characters (length estimated without quotes).

DVCNAME=*ALL/array(8): <var: char:8>/<c-string 1..8: c-string 1-8>

Names of the active spool devices for which information is output.

DVCNAME=*ALL

Operand default value. (Brief) information on all devices to which the user has access is requested.

DVCNAME=array(8): <var: char:8>

Requests information on all devices in the list.

DVCNAME=<c-string 1..8: c-string 1-8>

Requests information on the specified device.

DVCN_W=*ALL/<var: char:24>/<c-string 1..24: c-string 1-24>

Names of the active spool devices on which information is requested. Wildcards may be used.

DVCN_W=*ALL

Operand default value. Requests information on all devices.

DVCN W=<var: char:24>/<c-string 1..24: c-string 1-24>

The device names are partially qualified using wildcards.

DES_TYP=*LOCAL/<var: enum-of _flag_set:1>/*LIST/*REMOTE/*PUBLIC/*ALL

Outputs information on active spool devices according to the entry made for the DESTINATION operand of the START-PRINTER-OUTPUT command or the specified active printer.

DES TYP=*LOCAL

Operand default value. Local devices are selected (both distributed and central).

DES_TYP=<var: enum-of _flag_set:1>

DES_TYP is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. Refer to the table under DVC_TYP for the meaning of the values.

DES TYP=*LIST

Specifies a list of allocated devices.

DES_TYP=*REMOTE

Selects active RSO devices. This operand may only be used in conjunction with CLUSTER-NAME=*LOCAL-CLUSTER and SERVER-NAME=*ALL or *HOME.

DES_TYP=*PUBLIC

Selects active public RSO devices. This operand may only be used in conjunction with CLUSTER-NAME=*LOCAL-CLUSTER and SERVER-NAME=*ALL or *HOME.

DES TYP=*ALL

Selects all device types, regardless of whether they are local or RSO devices. If the RSO devices are not distributed, only active RSO devices from the local spool parameter file are selected.

DESNAME=*ALL/array (16): <var: char: 8>/<c-string 1..8: c-string 1-8>

Specifies the name of the device pool on which information is requested.

DESNAME=*ALL

Operand default value. Requests information on all device pools.

DESNAME=array(16): <var: char:8>

List of device pools on which information is requested.

DESNAME=<c-string 1..8: c-string 1-8>

Name of the device pool on which information is requested.

FRM_TYP=*ALL/<var: enum-of _flag_set:1>/*LIST/*STD/*EXCEPT

Specifies the entry type for form names.

FRM TYP=*ALL

Operand default value. All form names are allowed.

FRM_TYP=<var: enum-of _flag_set:1>

FRM_TYP is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. Refer to the table under DVC_TYP for the meaning of the values.

FRM TYP=*LIST

Selects a list of form names.

FRM TYP=*STD

Requests information on active devices to which the specified forms have been allocated or for which they have been excluded (see START-PRINTER-OUTPUT).

FRM TYP=*EXCEPT

Requests information on active devices for which the specified forms have been excluded.

FRMNAME=*ALL/array(16): <var: char:6>/<c-string 1..6: c-string 1-6>

Requests information on active devices to which the specified forms have been allocated or for which they have been excluded (see START-PRINTER-OUTPUT).

FRMNAME=*ALL

Operand default value. All form names are selected.

FRMNAME=array(16): <var: char:6>

A list of the form names allocated to or excluded for the devices on which information is to be output.

FRMNAME=<c-string 1..6: c-string 1-6>

Form name (allocated or excluded) for the active devices on which information is requested.

USR_TYP=<var: enum-of _flag_set:1> / *LIST / *EXCEPT / *ALL / *CLIST

Specifies the entry type for the user ID of the print jobs.

USR_TYP=<var: enum-of _flag_set:1>

USR_TYP is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. Refer to the table under DVC_TYP for the meaning of the values.

USR TYP=*LIST

List of print job user IDs for which information is to be output.

USR TYP=*EXCEPT

Requests information on active devices for which the specified user IDs are excluded.

USR_TYP=*ALL

Operand default value. All user IDs are selected.

USR TYP=*CLIST

All the elements of the user ID list must be of type *c-string*, with a maximum length of 8 characters (length estimated without quotes).

USRNAME=*ALL/array(16): <var: char:8>/<c-string 1..8: c-string 1-8>

Requests information on the active devices to which the specified user IDs are allocated or for which they are excluded (see START-PRINTER-OUTPUT).

USRNAME=*ALL

Operand default value. All print jobs are selected.

USRNAME=array(16): <var: char:8>

List of the user IDs for the active devices for which information is to be output.

USRNAME=<c-string 1..8: c-string 1-8>

User ID (allocated or excluded) for the active devices on which information is requested.

CLS_TYP=<u>*ALL</u>/<var: enum-of _flag_set:1>/*LIST/*EXCEPT

Specifies the entry type for the spoolout classes.

CLS_TYP=*ALL

Operand default value. All spoolout classes are selected.

CLS_TYP=<var: enum-of _flag_set:1>

CLS_TYP is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. Refer to the table under DVC_TYP for the meaning of the values.

CLS TYP=*LIST

List of the spoolout classes for the active devices for which information is to be output.

CLS TYP=*EXCEPT

Requests information on the active devices for which the specified spoolout classes are excluded.

CLASS=*ALL/array (16): <var: int: 2>/<integer 1..255>

Requests information on active devices to which the specified spoolout classes are allocated or for which they are excluded (see START-PRINTER-OUTPUT).

CLASS=*ALL

Operand default value. All spoolout classes are selected.

CLASS=array (16): <var: int: 2>

List of the spoolout classes (allocated or excluded) for the active devices on which information is to be output.

CLASS=<integer 1..255>

Spoolout class (allocated or excluded) for the active devices on which information is requested.

SPN_TYP=<var: enum-of _flag_set:1> / *LIST / *EXCEPT / *ALL / *CLIST Specifies the entry type for the job names.

SPN TYP=<var: enum-of flag set:1>

SPN_TYP is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. Refer to the table under DVC_TYP for the meaning of the values.

SPN TYP=*LIST

List of job names for the active devices on which information is requested.

SPN_TYP=*EXCEPT

Requests information on active devices for which the specified job names are excluded.

SPN_TYP=*ALL

Operand default value. All job names are selected.

SPN_TYP=*CLIST

All the elements of the list of job names must be of type c-string, with a maximum length of 8 characters (length estimated without quotes).

SPNAME=*ALL/array(16): <var: char:8>/<c-string 1..8: c-string 1-8>

Requests information on active devices to which the specified job names are allocated or for which they are excluded (see START-PRINTER-OUTPUT).

SPNAME=*ALL

Operand default value. All job names are selected.

SPNAME=array(16): <var: char:8>

List of the job names (allocated or excluded) for the active devices on which information is to be output.

SPNAME=<c-string 1..8: c-string 1-8>

Job name (allocated or excluded) for the active devices on which information is to be output.

ACC TYP=*ALL/<var: enum-of flag set:1>/*LIST/*EXCEPT

Specifies the entry type for the account number.

ACC_TYP=*ALL

Operand default value. All account numbers are selected.

ACC_TYP=<var: enum-of _flag_set:1>

ACC_TYP is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. Refer to the table under DVC_TYP for the meaning of the values.

ACC_TYP=*LIST

List of account numbers for the active devices on which information is requested.

ACC TYP=*EXCEPT

Requests information on active devices for which the specified account numbers are excluded.

ACCOUNT=*ALL/array (16): <var: char: 8>/<c-string 1..8: c-string 1-8>

Requests information on active devices to which the specified account numbers are allocated or for which they are excluded (see START-PRINTER-OUTPUT).

ACCOUNT=*ALL

Operand default value. All account numbers are selected.

ACCOUNT=array(16): <var: char:8>

List of the account numbers (allocated or excluded) for the active devices on which information is to be output.

ACCOUNT=<c-string 1..8: c-string 1-8>

Account number (allocated or excluded) for the active devices on which information is to be output.

OVL_TYP=*ALL/<var: enum-of _flag_set:1>/*LIST/*NONE/*ONLY/*EXCEPT Specifies the entry type for the overlays.

OVL TYP=*ALL

Operand default value. All overlays are selected.

OVL_TYP=<var: enum-of _flag_set:1>

OVL_TYP is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. Refer to the table under DVC_TYP for the meaning of the values.

OVL TYP=*LIST

List of overlays for the active devices on which information is requested.

OVL TYP=*NONE

Requests information on the active devices which do not support overlays.

OVL TYP=*ONLY

Requests information on the active devices which support overlays.

OVL TYP=*EXCEPT

Requests information on the active devices for which the specified overlays are excluded.

OVERLAY=*ALL/array (16): <var: char: 2>/<c-string 1..2: c-string 1-2>

Requests information on active devices to which the specified overlays are allocated or for which they are excluded (see START-PRINTER-OUTPUT).

OVERLAY=*ALL

Operand default value. All overlays are selected.

OVERLAY=array(16): <var: char:2>

List of the overlays (allocated or excluded) for the active devices on which information is to be output.

OVERLAY=<c-string 2..2: c-string 2-2>

Overlay (allocated or excluded) for the active devices on which information is to be output.

FOB=(type, low, high)

Information on active devices is to be output

- for which an optional FOB data overlay was specified (*ONLY)
- or which no FOB data overlay was specified (*NO)
- for which FOB data overlays within the specified size range were specified (*RANGE).

If *ANY is specified, the operand does not function as a selection criterion.

The entries made here must correspond to those made in the FORMS-OVERLAY-BUFFER operand of the START-PRINTER-OUTPUT command.

Information on printers without FOB are only issued if FOB=*ANY is specified.

type: *ANY / *ONLY / *NO / *RANGE

Specifies the entry type for selecting an FOB data overlay.

type: *ANY

Information is output regardless of the FOB data overlay.

type: *ONLY

Information is to be output on active devices for which an optional FOB data overlay was specified.

type: *NO

Information is to be output on active devices for which no FOB data overlay was specified.

type: *RANGE

Information is to be output on active devices for which FOB data overlays in the specified size range were specified.

low: *MIN /<var: int: 2> / <integer 0..32767>

Defines the minimum size of the FOB data overlay. *MIN is the value 0.

high: *STD / <var: int: 2> / <integer 0..32767>

Defines the maximum size of the FOB data overlay. *STD is the value 4032.

PRIO=(type, from, to)

Information is to be output on active devices whose priority is within the specified range (*RANGE). The entries made here have to match those in the relevant operands of the START-PRINTER-OUTPUT command.

type: *ALL / *RANGE

Printer types on which information is to be output.

type: *ALL

Information is output regardless of the device priority.

type: *RANGE

Information is to be output on devices whose priority is within the specified range.

from: *MIN / <var: int: 1> / <integer 30..255>
Defines the lowest priority. *MIN is the value 30.

to: *MAX / <var: int: 1> / <integer 30..255>

Defines the highest priority.*MAX is the value 255.

CSN=(type, low, high)

Information is to be output on active devices which

- need only one character set (*ONE)
- need a number of character sets within the specified range (*RANGE).

The entries made here have to match those in the relevant operands of the START-PRINTER-OUTPUT command.

Information on printers other than HP printers are only output if CHARACTER-SET-NUMBER=ALL was specified.

type: *ANY / *ONE / *RANGE

Defines the type of device on which information is to be output.

type: *ANY

Information is to be output on active devices, regardless of the number of character sets used.

type: *ONE

Information is to be output on active devices which require only one character set.

type: *RANGE

Information is to be output on active devices which require a number of character sets within the specified range.

low: *MIN/<var: int: 2>/<integer 1..32767>

Defines the lowest number of character sets. *MIN is the value 1.

high: *MAX/<var: int: 2>/<integer 1..32767>

Defines the highest number of character sets. *MAX is the value 64.

SRV_TYP=*ALL/<var: enum-of _flag_set:1>/*LIST/*WILD/*HOME

Specifies the entry type for server names.

SRV TYP=*ALL

Operand default value. All server names are selected.

SRV_TYP=<var: enum-of _flag_set:1>

SRV_TYP is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. Refer to the table under DVC_TYP for the meaning of the values.

SRV TYP=*LIST

List of server names for the active devices on which information is to be output.

SRV TYP=*WILD

The server names are partially qualified using wildcards.

SRV TYP=*HOME

The request is directed to the local server. The printers can be started either locally or remotely.

SERVER=*ALL/array (16): <var: char: 8>/<c-string 1..8: c-string 1-8>

The spool device is selected by the server that manages it. Active printers can only execute print jobs if the server of the computer to which the printer is connected allows it. If a remote cluster was selected, the only value permitted here is *ALL.

SERVER=*ALL

Operand default value. The spool devices managed by all servers on all computers in the addressed cluster are selected.

SERVER=array (16): <var: char: 8>

The spool devices managed by all the specified servers on the host in the addressed cluster are selected.

SERVER=<c-string 1..8: c-string 1-8>

The spool devices managed by the specified server on the hosts in the addressed cluster are selected.

SRW_WLD=*ALL/<var: char:24>/<c-string 1..24: c-string 1-24>

The spool devices managed by the specified servers on the hosts in the addressed cluster are selected. The servers are partially qualified using wildcards.

SRW_WLD=*ALL

Operand default value. The spool devices managed by all servers on all hosts in the addressed cluster are selected.

SRW WLD=<var: char: 24>/<c-string 1..24: c-string 1-24>

The servers are partially qualified using wildcards.

HST_TYP=<var: enum-of _flag_set:1> / *LIST / *LOCAL_CLUSTER / *ALL / *HOME / *EXCEPT / *CLIST

Defines the entry type for the host names.

HST_TYP=<var: enum-of _flag_set:1>

HST_TYP is not specified directly by means of an operand value; instead, it is specified indirectly by means of a field with constant contents (equate). An integer can be stored in the constant or the corresponding field. Refer to the table under DVC_TYP for the meaning of the values.

HST TYP=*LIST

List of host names for the active devices on which information is to be output.

HST_TYP=*LOCAL_CLUSTER

Information is output on active devices on which print jobs can be executed by all hosts in the local cluster.

HST_TYP=*ALL

Operand default value. All host names are selected.

HST TYP=*HOME

Information is output on active devices on which print jobs can be executed by the local host.

HST_TYP=*EXCEPT

Information is output on active devices on which print jobs can be executed by any host with the exception of those listed here.

HST_TYP=*CLIST

All the elements of the list of host names must be of type c-string, with a maximum length of 8 characters (length estimated without quotes).

HSTNAME=*ALL/array (16): <var: char: 8>/<c-string 1..8: c-string 1-8>

Information is requested on active devices on which print jobs can or cannot be executed by the specified hosts.

HSTNAME=*ALL

Operand default value. Information is output on active devices on which print jobs can be executed by all hosts in the clusters.

HSTNAME=array (16): <var: char: 8>

Information is output on active devices on which print jobs can or cannot be executed by the specified hosts.

HSTNAME=<c-string 1..8: c-string 1-8>

Information is output on active devices on which print jobs can or cannot be executed by the specified host.

CLUSTER=*LOCAL CLUSTER/<var: char: 8>/<c-string 1..8: c-string 1-8>

The active device is selected by the cluster containing the server that manages the device.

CLUSTER=*LOCAL CLUSTER

Operand default value. Information is output on active devices on which print jobs can be executed by all hosts in the local cluster.

CLUSTER=<var: char: 8>

The active device is selected by the cluster containing the server that manages the device.

CLUSTER=<c-string 1..8: c-string 1-8>

The active device is selected by the cluster containing the server that manages the device.

Note

New printer statuses are being introduced in conjunction with the interoperability between BS2000 and UNIX systems. These are based on ISO references. The SASDSPO macro shows the following statuses:

- I (IDLE): the printer is ready; no print job is currently being performed.
- R (RUN): the printer is active.
- M (Message): there is a console message concerning the printer.
- D (DETACHED): this is either the status between the entry of the START-PRINTER-OUTPUT command and its execution by the controller task or the status after the entry of the STOP-PRINTER-OUTPUT command.
- T (TRANSFER): a spoolout job that has been started is waiting for a transfer confirmation.
- U (UNKNOWN): the status is unknown.

Refer to the following table for the an overview of device statuses and the corresponding ISO status:

Displayed status	Print jobs	for printer ty	ISO status	
	Spool	APA	RSO	
A (ATTACHED)			Х	idle
I (IDLE)	Х	Х	Х	idle
T (TRANSFER)	Х		Х	running
R (RUN)	Х	Х	Х	running
M (MESSAGE)	Х	X	Х	needs attention
S (STOPPED)			Х	needs attention
W (WORKING FOR ADM)			Х	needs attention
D (DETACHED)	Х	X	Х	shutdown
U (UNKNOWN)	Х	Х	Х	shutdown

Return codes

(SC2)	SC1	Maincode	Meaning	
00	00	0000	No error	
02	00	XXXX	Warning - printer not found or processing not permitted	
03	01	FFFF	Insufficient buffer length	
01	01	XXXX	Invalid DEVICE operand	
02	01	XXXX	Invalid DESTINATION operand	
03	01	xxxx	Invalid FORM-NAME operand	
04	01	XXXX	Invalid OVERLAY operand	
05	01	XXXX	Invalid HOST-NAME operand	
0A	01	xxxx	Invalid USERID operand	
0B	01	xxxx	xx Invalid SPOOLOUT-CLASS operand	
0C	01	XXXX	Invalid JOB-NAME operand	
0D	01	XXXX	Invalid ACCOUNT operand	
0E	01	xxxx	Invalid ROTATION operand	
0F	01	xxxx	Invalid TWO-UP operand	
10	01	xxxx	Invalid FORM-OVERLAY-BUFFER operand	
11	01	xxxx	Invalid PRIORITY operand	
12	01	xxxx	Invalid CHARACTER-SET-NUMBER operand	
13	01	xxxx	Invalid address of RSO parameter list extension	

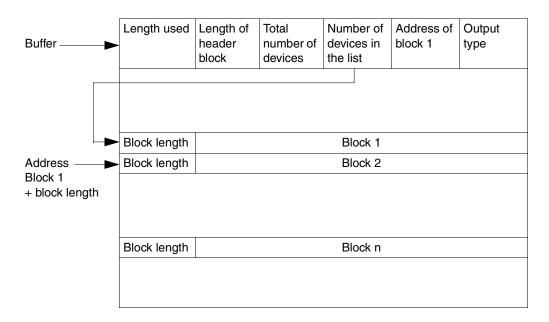
(SC2)	SC1	Maincode	Meaning	
15	01	XXXX	Invalid EXIT operand	
17	01	xxxx	Invalid address of Dprint parameter list extension	
18	01	xxxx	Invalid address of SPS parameter list extension	
1A	01	xxxx	Invalid address of DSEM parameter list extension	
1B	01	xxxx	Invalid BUFFER operand	
1C	01	xxxx	Invalid BUFFER LENGTH operand	
1D	01	xxxx	Invalid SCHEDULING-STATE operand	
1E	01	xxxx	Invalid STATUS operand	
1F	01	xxxx	Invalid CLUSTER operand	
20	01	xxxx	Invalid SERVER-NAME operand	
21	01	xxxx	Invalid ERROR-MESSAGE operand	
00	20	xxxx	System error	
02	40	xxxx	No authorization	
04	40	xxxx	Wrong operand value	
05	40	xxxx	This parameter list version is not supported	
02	80	xxxx	RSO subsystem not loaded	
05	80	xxxx	SPS subsystem not loaded	
06	80	xxxx	Dprintcl subsystem not loaded	
07	80	xxxx	Dprintcm subsystem not loaded	
08	80	xxxx	Dprintsv subsystem not loaded	
FF	FF	FFFF	SPOOL subsystem not loaded	

Note

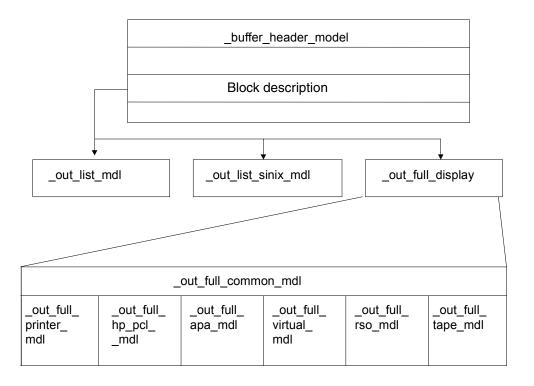
The characters xxxx in the Maincode column represent the message number. Enter the command HELP-MSG SCPxxxx or SPSxxxx in SYSTEM MODE in order to obtain detailed information on an error.

- The SASDSPO macro returns the code shown above. The HDRCHECK and \$VALID return codes can also be returned.
 If there is no data for the header, X'0000003C' will be returned in register 15.
- The information on the device is stored in an output buffer which has to be supplied by the user. If the buffer is not large enough, the macro will be terminated and the error message X'03 01 FFFF', meaning that the output area is too small, will be issued. A buffer in a sufficient size for all devices needs to be requested in order to obtain the entire information. Calling SASDSPO will then issue the information required.

 The section _output_buffer_model is always written into the buffer. It contains the exact number of active spool devices, the address of the first block containing the information on the active device, the block length and the output type. The address of each additional information block can be calculated from the address of the first information block and its length.



The block description differs in length and in layout, depending on the output type.
 The entire display layout consists of a general section and a section that refers to the device type.



- Information on the RSO subsystem is supplied by RSO itself, provided the subsystem was loaded. However, the information refers only to devices managed by RSO.
- The caller can specify the address of the reserved buffer and its length in the input parameter list using the BUFFER and BUFLEN operands. A warning that the buffer is not large enough can then be returned during a print process. Since the total number of printers and the number of printers actually returned to the buffer is specified in the buffer header, the user can calculate the required buffer size and call up the interface again.

The caller can also call the interface with BUFFER=*NONE and BUFLEN=*NONE (default values). The process will then request a sufficiently large buffer in the class 6 memory itself. The buffer pointer and buffer length are added to the input parameter list. The caller has to release the buffer.

Parameter list description

<pre>* parameter list des SSS_MDL</pre>	cripti DS	on 0F	
SSSHDR FHDR MF=(C,S	SS),EQ	UATES=NO	Standard header
<pre>* main return codes</pre>			
AIF ('&EQUATES' NE '	YES').	Q01001	
SSSRCNO	EQU	0	subcode1 = 00 : no error
SSSWARN	EQU	2	job not found or processing
*			not permitted
SSSVALM	EQU	2	<pre>subcode1 = 01 : pl area</pre>
*			invalid - maincode = FFFF
SSSPARM	EQU	1	subcode1 = 01 : wrong
*			parameter
SSSDEVI	EQU	1	subcode1 = 01 : device
SSSDEST	EQU	2	subcode1 = 01 : destination
SSSFONA	EQU	3	<pre>subcode1 = 01 : form name</pre>
SSS0VRL	EQU	4	<pre>subcode1 = 01 : overlay</pre>
SSSHOST	EQU	5	subcode1 = 01 : host
SSSPPOU	EQU	6	<pre>subcode1 = 01 : page printer</pre>
*			output
SSSPRSA	EQU	7	<pre>subcode1 = 01 : print sample</pre>
SSSTRAE	EQU	8	subcode1 = 01 : trace
SSSACCE	EQU	9	<pre>subcode1 = 01 : access</pre>
SSSUSER	EQU	10	subcode1 = 01 : userid
SSSSPCL	EQU	11	<pre>subcode1 = 01 : spoolout</pre>
*			class
SSSJNAM	EQU	12	<pre>subcode1 = 01 : job name</pre>
SSSACCN	EQU	13	subcode1 = 01 : account
SSSROTT	EQU	14	subcode1 = 01 : rotation
SSSTWOU	EQU	15	<pre>subcode1 = 01 : two-up</pre>
SSSFOB	EQU	16	subcode1 = 01 : fob
SSSPRIO	EQU	17	<pre>subcode1 = 01 : priority</pre>
SSSCSNU	EQU	18	<pre>subcode1 = 01 : char_set_num</pre>
SSSRSOL	EQU	19	subcode1 = 01 : addr of rso
*			pl
SSSREVI	EQU	20	<pre>subcode1 = 01 : revision</pre>
SSSEXTT	EQU	21	subcode1 = 01 : exit
SSSDPPL	EQU	23	subcode1 = 01 : addr of
*			dprintpl
SSSSPSL	EQU	24	subcode1 = 01 : addr of sps
*			p]
SSSSCSL	EQU	25	subcode1 = 01 : addr of scs
*			ρŢ
SSSDSEM	EQU	26	subcode1 = 01 : addr of dsem
*	-, -		pl
SSSBUFF	EQU	27	subcode1 = 01 : addr of
*			buffer

SSSBULE	EQU	28	<pre>subcode1 = 01 : buffer length</pre>
SSSSCHE	EQU	29	subcode1 = 01 : scheduling
*			state
SSSSTAU	EQU	30	<pre>subcode1 = 01 : status</pre>
SSSCLUT	EQU	31	<pre>subcode1 = 01 : cluster</pre>
SSSSERN	EQU	32	subcode1 = 01 : server
SSSERRM	EQU	33	<pre>subcode1 = 01 : error message</pre>
SSSEINF	EQU	34	<pre>subcode1 = 01 : information</pre>
SSSSHRT	EQU	3	output area too short - main
*			code=FFFF
SSSSYSE	EQU	0	<pre>subcode1 = 20 : system error</pre>
SSSPRIV	EQU	2	subcode1 = 40 : no
*	-,-		authorization
SSSVERR	EQU	4	operand value error
SSSVERE	EQU	5	p/l version not supported
SSSNOSS	EQU	0	<pre>subcode1 = 80 : subsystem not</pre>
*	240	Ü	loaded
SSSNRS0	EQU	2	rso not loaded
SSSNDCL	EQU	6	dprintcl not loaded
SSSNDCM	EQU	7	dprintcm not loaded
SSSNDSV	EQU	8	dprintsv not loaded
SSSNSP0	EQU	255	subcode1 = 255: spool not
*	240	200	loaded
.Q01001 ANOP			
*			
SSSDPCP	DS	Α	address of dprint_pl
SSSSPSP	DS	Α	address of sps_pl
SSSRSOP	DS	Α	address of rso_pl
SSSSCSP	DS	Α	address of scs_pl
SSSDSEP	DS	Α	address of dsem_pl
SSSRES5	DS	Α	
SSSRES6	DS	Α	
SSSRES7	DS	Α	
SSSBUF@	DS	Α	Address of a user-specified
*			area where the output must
*			be returned.
SSSBUFL	DS	F	user buffer length
SSSSCHS	DS	FL1	scheduling state
<pre>* _sched_state_set</pre>			
AIF ('&EQUATES' NE	'YES').	Q01002	
SSSSSNE	EQU	2	scheduling state = *next
SSSSSCU	EQU	1	scheduling state = *current
.Q01002 ANOP			- -
*			
SSSROTA	DS	FL1	rotation
<pre>* rotation_set</pre>			
AIF ('&EQUATES' NE	'YES').	Q01003	
SSSROSA	EQU	1	rotation = *any
			•

```
SSSROSY
                        EQU
                              2
                                           rotation = *yes
SSSROSN
                        EQU
                              4
                                           rotation = *no
SSSROSM
                        EQU
                              8
                                           rotation = *manual
.001003 ANOP
SSSTWUP
                              FL1
                        DS
                                           two-up
    two up set
    ('&EOUATES' NE 'YES').001004
SSSTUPA
                        FOU
                              131
                                           two-up = *any
SSSTUPY
                        EQU
                              3
                                           two-up = *yes
SSSTUPN
                        EQU
                              128
                                           two-up = *no
SSSTUM1
                        EQU
                              1
                                           two-up = *mode-1
SSSTUM2
                        EQU
                              2
                                           two-up = *mode-2
SSSTUDF
                        EQU
                              8
                                           two-up = not specified
.Q01004
        ANOP
SSSEXIT
                        DS
                              FI1
                                           exit.
    exit_set
AIF ('&EQUATES' NE 'YES').Q01005
SSSEXSA
                        FOU
                              4
                                           exit = *any
SSSEXSY
                        EQU
                              1
                                           exit = *yes
SSSEXSN
                        EQU
                              0
                                           exit = *no
.001005 ANOP
SSSSTAT
                        DS
                              FL1
                                           printer status
    _printer_status_set
AIF ('&EQUATES' NE 'YES').Q01006
SSSPSAT
                        FOU
                              193
                                           printer state = *attached
SSSPSID
                        FOU
                              201
                                           printer state = *idle
SSSPSTR
                        FOU
                              227
                                           printer state = *transfer
SSSPSRU
                        FOU
                              217
                                           printer state = *run
SSSPSME
                        EQU
                              212
                                           printer state = *message
SSSPSST
                        EQU
                              226
                                           printer state = *stopped
                              230
SSSPSW0
                        FOU
                                           printer state = *worked
SSSPSDE
                        EQU
                              196
                                           printer state = *detached
SSSPSUN
                        EQU
                              228
                                           printer state = *unknown
SSSPSAL
                        FOU
                              1
                                           printer state = *all
.001006 ANOP
SSSSINE
                        DS
                              FI1
                                           information
    _information_set
AIF ('&EQUATES' NE 'YES').Q01007
SSSSINS
                        FOU
                              0
                                           information = *summary
SSSSINA
                        EQU
                              1
                                           information = *all
*.001007
         ANOP
SSSUN2
                        DS
                              XL2
SSSCLUS
                        DS
                              CL8
                                           cluster
```

```
SSSADEVN
                         DS
                                0XL156
                                             device
SSSAFC8
                         DS
                                FL1
     _flag_set
     ('&EOUATES' NE 'YES').001008
SSSFALL
                         EQU
                                0
                                             _{type} = *all
SSSFNON
                         EQU
                                1
                                             _{type} = *none
SSSFLOC
                         EQU
                                2
                                             _{type} = *local
SSSFREM
                         EQU
                                3
                                             _{type} = *remote
SSSFDEV
                         EQU
                                5
                                             _type = *public_remote
SSSFLCL
                         EQU
                                6
                                             _type = *local_cluster
SSSFWLD
                         EQU
                                7
                                             _type = *wildcard
SSSFLST
                         EQU
                                8
                                             _{type} = *list
SSSFSTD
                         EQU
                                9
                                             _{type} = *std
SSSFHOM
                         EQU
                                10
                                             _{type} = *home
SSSFONL
                         EQU
                                11
                                             _type = *only
SSSFRNG
                         EQU
                                12
                                             _{type} = *range
SSSFONE
                         EQU
                                13
                                             _{type} = *one
SSSFEXC
                         EQU
                                14
                                             _type= *except
SSSFANY
                         EQU
                                15
                                             _type= *any
SSSFN0
                         EQU
                                16
                                             _type= *no
.001008
         ANOP
SSSAWC8
                         DS
                                CL24
SSSAUC8
                         DS
                                XL3
SSSAEC8
                         DS
                                8CL8
                ('&EQUATES' NE 'YES').Q01009
         ATF
SSSAEC8#
                         EQU
                               8
.001009
         ANOP
SSSAFU8
                         DS
                               8CL8
                ('&FOUATES' NF 'YES').001010
         ATF
SSSAEU8#
                         EQU
.001010 ANOP
SSSBDEST
                                0XL156
                         DS
                                             destination
SSSBFC8
                         DS
                                FI1
SSSBWC8
                         DS
                                CL24
SSSBUC8
                         DS
                                XL3
SSSBEC8
                         DS
                                16CL8
         AIF
                ('&EQUATES' NE 'YES').Q01011
SSSBEC8#
                         EQU
                                16
.Q01011
         ANOP
SSSCSRVN
                         DS
                                0XL156
                                             server name
SSSCFC8
                         DS
                                FL1
SSSCWC8
                         DS
                                CL24
```

```
SSSCUC8
                      DS
                            XL3
SSSCEC8
                      DS
                            16CL8
        AIF ('&EQUATES' NE 'YES').Q01012
SSSCEC8#
                      EQU
                            16
.001012 ANOP
SSSDHOST
                      DS
                            0XL156 host name
SSSDFC8
                      DS
                            FL1
SSSDWC8
                      DS
                            CL24
SSSDUC8
                      DS
                            XL3
SSSDEC8
                      DS
                            16CL8
              ('&EQUATES' NE 'YES').Q01013
        AIF
SSSDEC8#
                       EQU
                            16
.Q01013 ANOP
SSSEEMSG
                      DS
                            OXL156 error messae
                            FL1
SSSEFC8
                      DS
SSSEWC8
                      DS
                            CL24
SSSEUC8
                      DS
                            XL3
SSSEEC8
                      DS
                            1CL8
        AIF
               ('&EQUATES' NE 'YES').Q01014
SSSEEC8#
                      EQU 1
.Q01014 ANOP
SSSEEU8
                      DS
                            15CL8
        AIF ('&EQUATES' NE 'YES').Q01015
SSSEEU8#
                      EQU
                            15
.Q01015 ANOP
SSSFFORM
                      DS
                            0XL124
                                       form name
SSSFFC6
                      DS
                            FL1
SSSFWC6
                      DS
                            CL24
SSSFUC6
                      DS
                            XL3
SSSFEC6
                      DS
                            16CL6
        ATF
               ('&EQUATES' NE 'YES').Q01016
SSSFEC6#
                       EQU
                            16
.Q01016 ANOP
SSSGUSER
                      DS
                             0XL156 userid
SSSGFC8
                      DS
                             FL1
SSSGWC8
                      DS
                            CL24
SSSGUC8
                      DS
                            XL3
SSSGFC8
                             16CL8
                      DS
        AIF
               ('&EQUATES' NE 'YES').Q01017
SSSGEC8#
                       EQU
                             16
```

```
.001017 ANOP
SSSHSCLA
                        DS
                               0XL20
                                            spoolout class
                               FL1
SSSHFI1
                        DS
SSSHUI2
                        DS
                               XL3
SSSHEI1
                        DS
                               16X
SSSISNAM
                        DS
                               0XL156
                                            job name
SSSIFC8
                        DS
                               FL1
SSSIWC8
                        DS
                               CL24
SSSIUC8
                        DS
                               XL3
SSSIEC8
                        DS
                               16CL8
         AIF
                ('&EQUATES' NE 'YES').Q01018
SSSIEC8#
                               16
                        EQU
.Q01018 ANOP
SSSJACC0
                        DS
                               0XL156
                                            account
SSSJFC8
                        DS
                               FL1
SSSJWC8
                        DS
                               CL24
SSSJUC8
                        DS
                               XL3
SSSJEC8
                        DS
                               16CL8
                ('&EQUATES' NE 'YES').Q01019
         AIF
SSSJEC8#
                         EOU
                               16
.Q01019 ANOP
*
SSSKOVER
                        DS
                               0XL60
                                            overlay
SSSKFC2
                         DS
                               FI1
                               CL24
SSSKWC2
                        DS
SSSKUC2
                        DS
                               XL3
                               16CL2
SSSKEC2
                         DS
         AIF
                ('&EQUATES' NE 'YES').Q01020
SSSKEC2#
                         EQU
                               16
.Q01020
         ANOP
SSSLF0B
                        DS
                               0XL8
                                            fob
SSSLFR2
                        DS
                               FL1
                               XL1
SSSLUN1
                        DS
SSSLLR2
                         DS
                               Н
SSSLHR2
                        DS
                               Н
                               XL2
SSSLUR2
                        DS
SSSMCSN
                        DS
                               0XL8
                                            char set number
```

```
SSSMFR2
                        DS
                               FL1
SSSMUN1
                        DS
                               XL1
SSSMLR2
                        DS
                               Н
SSSMHR2
                        DS
                              Н
SSSMUR2
                        DS
                               XL2
SSSNPRIO
                        DS
                               0XL4
                                           priority
SSSNFR1
                        DS
                               FL1
SSSNLR1
                        DS
                               Χ
SSSNHR1
                        DS
                               Χ
SSSNUR1
                        DS
                               XL1
SSSUN6
                        DS
                              XL16
         AIF ('&EQUATES' NE 'YES').Q01021
SSS#
                              *-SSSHDR
                        EQU
.Q01021 ANOP
* _out_mdl
SSS_OMDL
                               0F
                        DS
SSSBHDR
                        DS
                               0XL24
                                            buffer header
SSSHUSL
                        DS
                               F
                                            total used length
SSSHBLL
                        DS
                                            header block length
SSSHTDV
                        DS
                               F
                                            total dvc count
SSSHRDV
                        DS
                               F
                                            return dvc count
SSSHFBD
                        DS
                                            displacement to first block
SSSHOTY
                        DS
                               FL1
                                            output type
    _output_type_set
         AIF ('&EQUATES' NE 'YES').Q01022
SSS0TLI
                        EQU
                               1
                                            output type = list
SSS0TFU
                               2
                        FOU
                                            output type = full
SSS0TSI
                        EQU
                               4
                                            output type = sinix
.001022 ANOP
SSSHDU2
                        DS
                               XL3
SSS_DISPLAY_ENTRY
                        DS
                               0XL1628
                                            _display_entry
SSSFULD
                        DS
                               0XL1628
                                            full display
SSSFCOM
                        DS
                               0XL1416
                                            common part
SSSFCBL
                        DS
                               Н
                                            block length
SSSFCDN
                        DS
                               CL8
                                            device name
SSSFCDT
                        DS
                               CL16
                                            device type
SSSFCUI
                        DS
                               CL8
                                            userid of the current job
                        DS
                               CI 4
SSSFCTS
                                            tsn of the current job
SSSFCLP
                        DS
                               Χ
                                            range of priority : low value
                        DS
                               Χ
                                            range of priority: high
SSSFCHP
```

```
value
SSSFCRV
                       DS
                             Χ
                                          revision number
SSSUNOV
                       DS
                             Χ
SSSFCUS
                       DS
                            FL1
                                          type of device activity
* _out_use_set
         AIF ('&EQUATES' NE 'YES').Q01023
SSSSINU
                       EQU
                             1
                                          use = input
SSSSPGU
                       EQU
                             2
                                          use = page printer
SSSSNOU
                       EQU
                             4
                                          use = no
SSSS0UU
                       EQU
                             8
                                          use = output
.001023 ANOP
SSSFCSS
                             FL1
                       DS
                                          scheduling state
   _out_sched_state_set
         AIF ('&EQUATES' NE 'YES').Q01024
SSSOSCC
                             1
                       EQU
                                          scheduling state = current
SSSOSCN
                       EQU
                              2
                                          scheduling state = next
.Q01024 ANOP
SSSFCAI
                       DS
                             FL1
                                          additional info type
* _add_info_type_set
         AIF ('&EQUATES' NE 'YES').Q01025
SSSATIN
                       FOU
                                          info = no
SSSATLI
                       FOU
                              1
                                          info = line printer
SSSAIHP
                       EQU
                              2
                                          info = hp/hp90 printer
SSSAIAP
                       EQU
                             3
                                          info = apa printer
SSSAIVI
                       EQU
                             4
                                          info = virtual printer
SSSAIRS
                             5
                       EQU
                                          info = rso printer
SSSAITA
                       EQU
                                          info = tape printer
                             6
.Q01025 ANOP
SSSFUSD
                       DS
                             XL3
                       DS
SSSSPCI ASS
                             0XI20
                                          class
SSSFCCN
                                          number of elements in list
                       DS
SSSFCCI
                       DS
                             FL1
                                          indicator
    _out_indicator_set
             ('&EQUATES' NE 'YES').Q01026
SSSSINZ
                                          no value set
                       EOU
                             0
SSSSINE
                       FOU
                              1
                                          negative list
SSSSIP0
                       EQU
                             16
                                          positive list
SSSSIAL
                       EQU
                             20
                                          all or any
.Q01026 ANOP
SSSFCCU
                       DS
                             XL1
SSSECCI
                       DS
                              16X
                                          list of spoolout class
```

```
SSSFORM
                       DS
                             0XL100
                                         form
SSSFCFN
                       DS
                                         number of elements in list
SSSFCFI
                       DS
                             FI1
                                         indicator
   _out_ind_form_set
         AIF ('&EQUATES' NE 'YES').Q01027
SSSSFNG
                       EQU
                             1
                                         negative list
SSSSFNE
                       EQU
                             19
                                         negative equivalent
SSSSFAL
                       EQU
                             4
                                         a11
SSSSFAE
                       EQU
                             22
                                         all equivalent
SSSSFP0
                       EQU
                             16
                                         positive list
SSSSFPE
                       EQU
                             18
                                         positive equivalent
.Q01027 ANOP
SSSFCFU
                       DS
                             XL1
SSSFCFE
                       DS
                             16CL6
                                        list of form name
               ('&EQUATES' NE 'YES').Q01028
         AIF
SSSECEF#
                       FOU
                             16
.Q01028 ANOP
*
SSSPNAME
                       DS
                             0XL132
                                         pname
SSSFCPN
                       DS
                                         number of elements in list
SSSFCPI
                       DS
                             FL1
                                         indicator
SSSECPU
                       DS
                             XL1
SSSFCPE
                       DS
                             16CL8
                                     list of job name
               ('&EQUATES' NE 'YES').Q01029
         AIF
                             16
SSSFCPE#
                       FOU
.Q01029 ANOP
                       DS
SSSUSFRID
                             0XI 132
                                         userid
SSSFCUN
                       DS
                                         number of elements in list
                       DS
                             FL1
                                         indicator
SSSFCUD
SSSECUU
                       DS
                             X I 1
SSSFCUE
                       DS
                             16CL8
                                         list of user ids
               ('&EQUATES' NE 'YES').Q01030
         AIF
SSSFCUF#
                       FOU
                             16
.001030 ANOP
SSSACCOUNT
                       DS
                             0XL132
                                         account
                                         number of elements in list
SSSFCAN
                       DS
SSSFCAD
                       DS
                             FI1
                                         indicator
SSSFCAU
                       DS
                             XL1
SSSFCAE
                       DS
                             16CL8
                                         list of account number
               ('&EQUATES' NE 'YES').Q01031
         ATF
SSSFCAE#
                       EQU 16
.001031 ANOP
```

```
*
SSSDESTIN
                         DS
                               0XL812
                                            destin
SSSFCLN
                         DS
                               Н
                                            number of elements in list
SSSFCDI
                         DS
                               FL1
                                            indicator
SSSFCDU
                         DS
                               X I 1
SSSFCDE
                         DS
                               101CL8
                                            list of device pools
         AIF
                ('&EOUATES' NE 'YES').001032
SSSFCDE#
                         FOU
                               101
.001032
         ANOP
SSSFCDA
                         DS
                               CL10
                                            last scheduling date
SSSFSTI
                         DS
                               CL5
                                            last scheduling time
SSSFSSE
                         DS
                               CL1
                                            season (S/W)
SSSFEMS
                         DS
                               CL8
                                            error msg
                               XL16
SSSFRSX
                         DS
SSS_OUT_FULL_DEDICATED DS
                               0XL212
                                            _out_full_dedicated
SSSEHPC
                         DS
                               0XI 212
SSSFHBL
                         DS
                               Н
                                            block length
SSSFHFN
                         DS
                               CL6
                                            current form used
SSSFHSN
                         DS
                               CL8
                                            server name
SSSEHHN
                         DS
                               CL 8
                                            host name
SSSFHLF
                         DS
                               Н
                                            range of fob : low value
SSSFHHF
                         DS
                               Н
                                            range of fob : high value
SSSFHLC
                         DS
                               Н
                                            range of char set num : low
                                            value
SSSFHHC
                         DS
                               Н
                                            range of char set num : high
                                            value
SSSFHTR
                         DS
                               FI1
                                            trace file to be generated
    _yes_no_set
                ('&EQUATES' NE 'YES').Q01033
                               1
SSSSTYF
                         FOU
                               2
SSSSTN0
                         EQU
.001033
         ANOP
SSSFHEX
                         DS
                               FL1
                                            exit routines activated
SSSFHSA
                         DS
                               FL1
                                            sample printout
SSSFHST
                         DS
                               FL1
                                            printer status
    _out_prt_status_set
                ('&EQUATES' NE 'YES').Q01034
SSSOPSA
                               193
                                            C'A'
                         FOU
SSSOPSI
                         EQU
                               201
                                            C'I'
SSSOPST
                                            C'T'
                         EQU
                               227
SSSOPSR
                               217
                                            C'R'
                         FOU
SSSOPSM
                         EQU
                               212
                                            C'M'
                               226
                                            C'S'
SSSOPSS
                         EOU
```

```
SSSOPSW
                      EQU
                           230
                                       C'W'
SSSOPSD
                      EQU
                           196
                                       C'D'
SSSOPSU
                      EQU
                           228
                                       C'U'
.001034 ANOP
SSSFHSC
                      DS
                           FL1 printer scope
   _scope_set
        AIF ('&EOUATES' NE 'YES').001035
SSSSCNO
                      EQU
                           0
                                       scope = no
SSSSCLO
                      EQU
                           1
                                     scope = local
SSSSCDI
                      EQU
                           2
                                     scope = distributed
.Q01035 ANOP
SSSFHET
                     DS
                           FL1 class of error
* _out_der_type_set
        AIF ('&EQUATES' NE 'YES').Q01036
SSSDERN
                      FOU
                                       device erro type = no
SSSDERO
                      EQU
                           1
                                       device erro type = oper
SSSDERD
                      EQU 2
                                       device erro type = data
SSSDERL
                      EQU 4
                                       device erro type = load
SSSDERJ
                      EQU 8
                                       device erro type = job
.Q01036 ANOP
SSSFHTU
                      DS
                           FL1 two-up processing
   _out_two_up_set
        AIF ('&EQUATES' NE 'YES').Q01037
SSSTUM1
                      FOU
                           1
                                      two-up = mode-1
SSSTUM2
                      EQU
                           2
                                      two-up = mode-2
SSSTUPY
                      EOU 7
                                     two-up = yes
SSSTUPN
                      EQU 128
                                      two-up = no
SSSTUPA
                      FOU
                           135
                                       two-up = any
.001037 ANOP
SSSFHR0
                      DS
                           FI1
                                       rotation
   out rotation set
        AIF ('&EQUATES' NE 'YES').Q01038
SSSSRAN
                           1
                      FOU
                                       rotation = *any
SSSSRYE
                      EQU
                           2
                                       rotation = *yes
SSSSRNO
                      EQU
                           4
                                       rotation = *no
SSSSRMA
                      FOU
                                       rotation = *manual
.Q01038 ANOP
SSSFHCD
                      DS
                           CL2
                                      current dia
                      DS
SSSHOST HP PCL
                           0XL132
                                       host hp pcl
SSSFHIN
                                       number of elements in list
                      DS
                           Н
SSSFHHI
                      DS
                           FL1
                                       indicator
* _out_ind_host_set
```

```
AIF
                ('&EQUATES' NE 'YES').Q01039
SSSSHP0
                        EQU
                               16
                                            positive
SSSSHNG
                        EQU
                               1
                                            negative
SSSSHAL
                        EQU
                               20
                                            all cluster
SSSSHH0
                        EQU
                               48
                                            home
SSSSHL0
                        EQU
                               80
                                            local
.001039 ANOP
SSSFHHU
                        DS
                               XL1
SSSFHHE
                        DS
                               16CL8
                                           list of host name
         AIF
                ('&EQUATES' NE 'YES').Q01040
SSSFHHE#
                               16
                        EQU
.Q01040 ANOP
                        DS
SSSDIA_HP_PCL
                               0XL36
                                            dia_hp_pcl
SSSFHDN
                        DS
                                            number of elements in list
                        DS
                               FL1
                                            indicator
SSSFHDI
    _out_ind_dia_set
              ('&EQUATES' NE 'YES').Q01041
SSSSDP0
                        EQU
                               16
                                            positive
SSSSDNG
                        EQU
                               1
                                            negative
SSSSDAL
                        EQU
                               20
                                            all
SSSSDNO
                        EQU
                               32
                                            none
SSSSDON
                        EQU
                               64
                                            only
.001041
        ANOP
SSSFHDU
                        DS
                               XL1
SSSFHDE
                        DS
                               16CL2
                                            list of overlay
         ATF
                ('&EQUATES' NE 'YES').Q01042
SSSFHDF#
                               16
                        FOU
.Q01042 ANOP
SSSFUS2
                        DS
                               XI2
         ORG
                SSS OUT FULL DEDICATED
SSSFLIN
                        DS
                               0XL164
SSSFLBL
                        DS
                                            block length
                               Н
SSSFL FN
                        DS
                               CL6
                                            current form used
SSSFLSN
                        DS
                               CL8
                                            name of the server
SSSFLHN
                        DS
                               CL8
                                            name of the host
                        DS
                               FL1
                                            trace file to be generated
SSSFLTR
SSSFLEX
                        DS
                               FL1
                                            exit routines activated
                        DS
                               FL1
                                            sample printout
SSSFLSA
SSSFLST
                        DS
                               FL1
                                            printer status
SSSFLSC
                        DS
                               FL1
                                            printer scope :
                                            local/distributed
```

```
SSSFLET
                        DS
                              FL1
                                           class of error
SSSHOST_PRINTER
                        DS
                              0XL132
                                           host_printer
                                           number of elements in list
SSSFLLN
                        DS
                              Н
                                           indicator
SSSFLHI
                        DS
                              FL1
SSSFLHU
                        DS
                              X I 1
SSSFLHE
                        DS
                              16CL8
                                           list of hostname
         AIF
               ('&EOUATES' NE 'YES').001043
SSSFLHE#
                        FOU
                              16
.Q01043 ANOP
SSSFUS1
                        DS
                              XL2
         ORG
               SSS_OUT_FULL_DEDICATED
SSSFAPA
                        DS
                              0XL164
SSSEABL
                        DS
                                           block length
SSSFAFN
                        DS
                                           current form used
                              CL6
SSSFASN
                        DS
                              CL8
                                           server name
SSSFAHN
                        DS
                              CL8
                                           host name
SSSFATR
                        DS
                              FL1
                                           trace file to be generated
SSSFALV
                        DS
                              FL1
                                           trace file to be generated
    out level set
         AIF ('&EQUATES' NE 'YES').Q01044
SSSLEVN
                        EQU
                              40
                                           trace level = not spec.
                              226
                                           trace level = status
SSSLEVS
                        EQU
                        EQU
                                           trace level = blk-cont.
SSSI FVB
                              194
SSSLEVC
                        EQU
                              195
                                           trace level = complete
.001044 ANOP
                                           exit routines activated
                        DS
                              FI1
SSSFAFX
                                           printer status
SSSFAST
                        DS
                              FL1
                        DS
                              FL1
                                           printer scope
SSSFASC
                                           class of error
                              FI1
SSSFAFT
                        DS
SSSHOST APA
                        DS
                              0XL132
                                           host apa
SSSFALN
                        DS
                                           number of elements in list
SSSFAHI
                        DS
                              FL1
                                           indicator
SSSFAHU
                        DS
                              XL1
SSSEAHE
                        DS
                              16CL8
                                           list of host name
         ATF
               ('&EQUATES' NE 'YES').Q01045
SSSFAHE#
                        EQU
                              16
.Q01045 ANOP
SSSFAUN
                        DS
                              XL2
         ORG
               SSS_OUT_FULL_DEDICATED
```

SSSFVIR SSSFVBL SSSFVSN SSSFVPN SSSFVST SSSFVSC SSSFVET SSSFVU1			DS DS DS DS DS DS DS DS	0XL160 H CL8 CL8 CL4 FL1 FL1 XL1	block length server name host name program task printer status printer scope class of error
SSSHOST_ SSSFVLN SSSFVHI SSSFVHU SSSFVHE	VIRT AIF	('	DS DS DS DS DS	0XL132 H FL1 XL1 16CL8 E 'YES').Q01	host_virt number of elements in list indicator list of host name
SSSFVHE# .Q01046 *		CALQUA	EQU	16	040
SSSFVU2 *			DS	XL2	
	ORG	SSS_OUT	_FULL_I	DEDICATED	
*					
SSSFRS0			DS	0XL82	
SSSFRBL			DS	H	block length
SSSFRFN			DS	CL6	current form used
SSSFRTR			DS	FL1	trace file to be generated
SSSFREX			DS	FL1	exit routines activated
SSSFRSA			DS	FL1	sample printout
SSSFRST SSSFRHE			DS DS	FL1 4CL4	printer status list of accesses
SSSEKHE	AIF	(' 8. EOII).		4014 E 'YES').Q010	
SSSFRHE#		(algun	EQU	4	047
.Q01047	ANOP		LQU	7	
SSSFJVN *	7.1101		DS	CL54	MONJV name
	ORG	SSS_OUT	_FULL_I	DEDICATED	
*			DC	0.71.100	
SSSFTAP			DS	0XL182	blook longth
SSSFTBL SSSFTLF			DS	Н	block length range of fob : low value
SSSFTHF			DS DS	H H	range of fob: fow value
SSSFTLC			DS	Н	range of the set num: low
*			در	11	value
SSSFTHC *			DS	Н	range of char set num : high value
SSSFTRP			DS	Н	retention period
SSSFTIM			DS	CL6	vsn of directory tape
					·

```
SSSFTTD
                       DS
                             CL8
                                          density of tape
                       DS
                              FL1
                                          processing mode
SSSFTRM
* _out_rmode_set
         AIF ('&EQUATES' NE 'YES').Q01048
SSSSRMZ
                       EQU
                             0
                                          no value set
SSSSRMD
                       EQU
                              2
                                          rmode = direct
SSSSRMC
                       EQU
                             8
                                          rmode = copy
.001048 ANOP
SSSFTTU
                       DS
                              FL1
                                          two-up processing
SSSFTR0
                       DS
                              FL1
                                          rotation
SSSFTTY
                       DS
                              FL1
                                          selection type
    _tape_type_flag_set
         AIF ('&EQUATES' NE 'YES').Q01049
SSSTTFY
                       EQU
                              1
                                         type = may
                       EQU
                              2
SSSTTFM
                                          type = must
.001049 ANOP
SSSFTVA
                       DS
                              14X
                                          list of type
                       DS
SSSDIA_TAPE
                             0XL36
                                          dia_tape
SSSFTDN
                       DS
                                          number of elements in list
SSSFTDI
                       DS
                             FL1
                                          indicator
SSSFTDU
                       DS
                             XL1
SSSFTDE
                       DS
                              16CL2
                                          list of overlay
         AIF
               ('&EQUATES' NE 'YES').Q01050
SSSFTDE#
                       FOU
                             16
.Q01050 ANOP
                       DS
SSSVSN
                              0XI 100
SSSFTVN
                       DS
                                          number of elements in list
SSSFTVI
                       DS
                             FL1
                                          indicator
SSSFTVU
                       DS
                             X I 1
SSSFTVE
                       DS
                             16CL6
                                          list of vsn
               ('&EQUATES' NE 'YES').Q01051
         AIF
SSSFTVF#
                       FOU
                              16
.001051 ANOP
SSSFTU2
                       DS
                             XI2
         ORG
               SSS OUT FULL DEDICATED+212
         ORG
               SSS_DISPLAY_ENTRY
SSSFLID
                       DS
                              0XL92
                                          list display
SSSLBLK
                       DS
                             Н
                                          block length
                              CL8
SSSLDVN
                       DS
                                          device name
```

SSSLDVT SSSLUID SSSLTSN SSSLFRM SSSLSPC *		DS DS DS DS DS	CL16 CL8 CL4 CL6 X	device type userid of the current job tsn of the current job form of the current job spoolout class of the current job
SSSRSOC		DS	Χ	
SSSLEXI		DS	FL1	exit routine activated
SSSLSTA		DS	FL1	printer status
SSSLSCO *		DS	FL1	<pre>printer scope : local/distributed</pre>
SSSLSTP		DS	FL1	stop command issued
SSSLCAL		DS	FL1	caller is administrator
SSSLCRI		DS	FL1	explicite criteria given
SSSLCDA		DS	CL10	last scheduling date
SSSLSTI		DS	CL5	last scheduling time
SSSLSSE		DS	CL1	season (S/W)
SSSLEMS		DS	CL8	error msg
SSSLRSX *		DS	XL16	
	ORG	SSS_DISPLAY_E	NTRY	
*				
SSSFSLI		DS	0XL18	sinix display
SSSLSBL		DS	Н	block length
SSSLSFN		DS	CL6	current form used
SSSLSDN		DS	CL8	device name
SSSLSST		DS	FL1	printer status
SSSLSUN *		DS	XL1	
	ORG	SSS_DISPLAY_E	NTRY+1628	
	AIF	('&EQUATES' N	E 'YES').Q01	052
SSSOUTPUT#		EQU	*-SSSHUSL	

.Q01052 ANOP

STPSPPR - cancel printer allocation

User group: Systems support staff, RSO device administrator

Programming language: Assembler, CPP, C

Macro type: M, S

Printer allocations are cancelled again with the STPSPPR macro.

Format

Operation	Operands
STPSPPR	ENTRY = YES / NO
	VARIANT = <c-string-without-quotes 33=""> / <u>001</u></c-string-without-quotes>
	DVNAME = array(8): <var: char:8=""> / <c-string 18=""></c-string></var:>
	FORCE = *STD / *IMMEDIATE

Description of the operands

ENTRY=YES/NO

Specifies whether the relevant entries are generated or not.

ENTRY=YES

The entries are generated.

ENTRY=NO

The entries are not generated.

VARIANT=001 / <c-string-without-quotes 3..3>

Specifies the variant of the generated parameter list.

DVNAME:=list-poss(8): <alphanum-name 1..8> /identifier

Names of the printers that are to be removed.

FORCE:=*STD / *IMMEDIATE

Specifies whether an active print job is to be terminated immediately (*IMMEDIATE) or whether processing is to be concluded first (*STD).

Return code

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
00	00	0000	No error
02	00	XXXX	Warning - printer does not exist or processing not allowed
02	01	FFFF	Invalid parameter list area
01	01	XXXX	Invalid DEVICE operand
16	01	XXXX	Invalid FORCE operand
00	20	XXXX	System error
02	40	XXXX	No authorization
04	40	XXXX	Operand value error
05	40	XXXX	Parameter list version not supported
FF	FF	FFFF	Spool subsystem not loaded

Note

The characters xxxx in the Maincode column represent the message number. Enter the command HELP-MSG SCPxxxx or SPSxxxx in SYSTEM MODE in order to obtain detailed information on an error.

Parameter list description

```
parameter list description
STP MDL
          DS
                0F
STPHDR
         FHDR MF=(C,STP), EQUATES=NO Standard header
    main return codes
         ATF
              ('&EQUATES' NE 'YES').Q01001
STPRCNO
          FOU
                0
                                           subcode1 = 00 : no error
          EQU
STPPARM
               1
                                           subcode1 = 01 : wrong
                                           parameter
                2
                                           subcode1 = 20 : system error
STPSYSE
          EQU
                                           subcode1 = 40 : no
STPPRIV
          EQU
                2
                                           authorization
STPVFRR
          FOU
                4
                                           operand value error
STPVERE
          EQU
                5
                                           p/l version not supported
STPNOSS
          FOU
                ()
                                           subcode1 = 80 : subsystem not
*
                                           loaded
STPNSP0
          FOU
                1
                                           spool not loaded
          EQU
                2
                                           rso not loaded
STPNRS0
STPDPPL
          FOU
                23
                                           subcode1 = 01 : addr of
                                           dprintpl
STPSPSL
          FOU
                24
                                           subcode1 = 01 : addr of sps
                                           subcode1 = 01 : addr of rso
STPRSOL
          EQU
                19
STPSCSL
          FOU
                25
                                           subcode1 = 01 : addr of scs
          EQU
                26
STPDSEM
                                           subcode1 = 01 : addr of dsem
STPBUFF
          EQU
                27
                                           subcode1 = 01 : addr of
                                           buffer
STPBUFL
          FOU
                28
                                           subcode1 = 01 : buffer length
STPSCHE
          EQU
                29
                                           subcode1 = 01 : scheduling
STPROTA
          FOU
                14
                                           subcode1 = 01 : rotation
STPTWOU
          EQU
                15
                                           subcode1 = 01 : two-up
STPEXTT
          EQU
                21
                                           subcode1 = 01 : exit
STPSTAT
          FOU
                30
                                           subcode1 = 01 : status
STPCLUS
          EQU
                31
                                           subcode1 = 01 : cluster
                                           subcode1 = 01 : device
STPDEVI
          EQU
                1
STPDEST
          FOU
                2
                                           subcode1 = 01 : destination
                32
                                           subcode1 = 01 : server
STPSERN
          FOU
STPHOST
          EQU
                5
                                           subcode1 = 01 : host
STPERRM
          EQU
                33
                                           subcode1 = 01 : error message
STPFONA
          EQU
                3
                                           subcode1 = 01 : form name
STPUSER
          FOU
                10
                                           subcode1 = 01 : userid
STPSPCL
          EQU
                11
                                           subcode1 = 01 : spoolout
                                          class
```

```
STPJNAM
           EQU
                 12
                                             subcode1 = 01 : job name
           EQU
                 13
                                             subcode1 = 01 : account
STPACCN
STPOVRI
           FOU
                 4
                                             subcode1 = 01 : overlay
STPF0B
           EQU
                 16
                                             subcode1 = 01 : fob
STPCSNU
           EQU
                 18
                                             subcode1 = 01 : char set num
STPPRIO.
           FOU
                 17
                                             subcode1 = 01 : priority
STPPPOU
           EQU
                 6
                                             subcode1 = 01 : page printer
*
                                             output
STPPRSA
           FOU
                 7
                                             subcode1 = 01 : print sample
STPTRAF
           FOU
                 8
                                             subcode1 = 01 : trace
STPACCE
           EQU
                 9
                                             subcode1 = 01 : access
           EQU
                 20
                                             subcode1 = 01 : revision
STPREVI
STPFORC
           FOU
                 22
                                             subcode1 = 01 : force
.001001
         ANOP
STPFORCE DS
                 FL1
                                             type of use no
    flag set
                ('&EQUATES' NE 'YES').Q01002
         AIF
STPFALL
           FOU
                 0
                                             _{type} = *all
STPFNON
           FOU
                 1
                                             type = *none
                 2
                                             _{type} = *local
STPFLOC
           EQU
STPFREM
                 3
                                             _{type} = *remote
           EQU
STPFCEN
           FOU
                 4
                                             type = *central
STPFDFV
           FOU
                 64
                                             _type = *device
STPFMDV
           EQU
                 128
                                             _type = *manag_dev
STPFWLD
           FOU
                 5
                                             type = *wildcard
                                             _{type} = *list
STPFL ST
           FOU
                 6
                 7
                                             _{type} = *std
STPFSTD
           EQU
STPFOWN
           FOU
                 8
                                             type = *own
STPFHOM
           FOU
                 9
                                             _{type} = *home
                 10
                                             _{type} = *ident
STPFIDE
           FOU
STPFANY
           EQU
                 11
                                             _{type} = *any
                 12
STPFONL
           EQU
                                             _{type} = *only
                 13
                                             _type = *range
STPFRNG
           FOU
STPFONE
           EQU
                 14
                                             _{type} = *one
                 15
STPFNO
           EQU
                                             _{type} = *no
STPFOU
           FOU
                 16
                                             _type=*equivalent
           EQU
                 17
                                             _type=*equivalent_ex
STPEQUX
STPEXCE
           EQU
                 18
                                             type=*except
STPALLO
           FOU
                 19
                                             _type=*allowed
STPFYFS
           FOU
                 20
                                             _{type} = *_{yes}
                 21
STPFCOM
           EQU
                                             _{type} = *complete
STPFSTA
           FOU
                 22
                                             _type = *status
STPFBLK
           FOU
                 23
                                             _type = *block_control
                 24
                                             _type = *manual
STPFMAN
           EQU
                 25
                                             _{type} = *mode-1
STPFM01
           FOU
STPFM02
           EQU
                 26
                                             _{type} = *mode2
           EQU
                 27
STPFACT
                                             _type = *active
```

```
EQU
               28
                                       _type = *no-active
STPFNOA
             29
                                       _{type} = *unchanged
STPFUNC EQU
STPFADD EQU
               30
                                       _{type} = *add
                                       _{type} = *remove
STPFREV EOU
               31
               32
STPFUPD EQU
                                       _type = *update
STPFIMD
         EQU
               33
                                       _type = *immediate
.Q01002 ANOP
STPUN1
        DS
              XL15
STPADVOUT DS
              0XL132
                                       device name
STPAFCB DS
               FL1
STPAUCB
       DS
              XL3
STPAECB
        DS
               8CL8
        AIF
             ('&EQUATES' NE 'YES').Q01003
STPAECB# EQU
.001003 ANOP
STPAEC2
        DS
               8CL8
        AIF
              ('&EQUATES' NE 'YES').Q01004
STPAEC2# EOU
.Q01004 ANOP
        AIF
             ('&EQUATES' NE 'YES').Q01005
STP#
        EQU
             *-STPHDR
.Q01005 ANOP
```

STRSPPR - allocate print jobs to printers

User group: Systems support staff

Programming language: Assembler, CPP, C

Macro type: M, S

Print jobs are allocated to printers using the STRSPPR macro.

Format

Operation	Operands
STRSPPR	ENTRY = <u>YES /</u> = NO
	VARIANT = <c-string-without-quotes 33=""> / <u>001</u></c-string-without-quotes>
	DEVTYP = *ANY-LOCAL / *RSO / *ANY-LOCAL
	DEVNAM = array(8): <var: char:8=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	RSOPAR = <var: pointer=""> / *NONE</var:>
	DESTTYP = *NONE / *STD / *LIST
	DESTNAM = array(16): <var: char:8=""> / *CENTRAL / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	FORMTYP = *ALL / *STD / *LIST / *EQUIVALENT / *EXCEPT / *EQUIVALENT_EX
	FORMNAM = array(16): <var: char:6=""> / <c-string 1-6="" 16:="" c-string=""></c-string></var:>
	OVLTYP = *ALL / *NONE / *ONLY / *EXCEPT / *LIST
	OVLNAM = array(16): <var: char:2=""> / <c-string 1-2="" 12:="" c-string=""></c-string></var:>
	HOSTTYP = *ALL-CLUSTERS / *HOME / *LOCAL-CLUSTER / *EXCEPT / *LIST/ *CLIST

(part 1 of 3)

Operation	Operands
STRSPPR	HOSTNAM = array(16): <var: char:8=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	PAPROUT = <u>*NO</u> / *ALLOWED
	PSAMPLE = <u>*NO</u> / *YES
	TRACE = *NO / *COMPLETE / *STATUS / *BLOCK_CONTROL /
	USERTYP = *ALL /*EXCEPT / *LIST / *CLIST
	USERNAM = array(16): <var: char:8=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	CLASTYP = *ALL / *EXCEPT / *LIST
	CLASNAM = array(16): <var: int:1=""> / <integer 1255=""></integer></var:>
	SPNATYP = *ALL / *EXCEPT / *LIST / *CLIST
	SPNANAM = array(16): <var: char:8=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	ACCTYP = *ALL / *EXCEPT / *LIST
	ACCNAM = array(16): <var: char:8=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	ROT = *NO / *YES / <u>*ANY</u> / *MANUAL
	TWO_UP = *NO / *YES / <u>*ANY</u> / *MODE-1 / *MODE-2
	F_O_B = (type, low, high): type: *ANY / *ONLY / *NO / *RANGE low: <var: int:2=""> / <integer 032767=""> / *STD high: <var: int:2=""> / <integer 032767=""> / *STD</integer></var:></integer></var:>

(part 2 of 3)

Operation	Operands
STRSPPR	PRIO = (type, low, high):
	CHARSN = (type, low, high): type: *ALL / *ONE / *RANGE low: <var: int:2=""> / <integer 132767=""> / *MIN high: <var: int:2=""> / <integer 132767=""> / *MAX</integer></var:></integer></var:>
	EXITR = *ACTIVE / *NOT-ACTIVE

(part 3 of 3)

Description of the operands

ENTRY=YES/NO

Specifies whether the relevant entries are generated or not.

ENTRY=YES

The relevant entries are generated.

ENTRY=NO

The relevant entries are not generated.

VARIANT=001 / <c-string-without-quotes 3..3>

Specifies the variant of the generated parameter list.

DEVTYP=*ANY-LOCAL / *RSO

Specifies the device type.

DEVTYP=*ANY-LOCAL

Local printers are allocated.

DEVTYP=*RSO

RSO printers are to be allocated.

DEVNAM=array(8): <alphanum-name 1..8> / <c-string 1..8: c-string 1..8>

Logical device name specified in the spool parameter file (up to 8 alphanumeric characters). When a device with a specific name is activated (START-PRINTER-OUTPUT or MODIFY-PRINTER-OUTPUT), it can only be addressed by that name until it is deactivated again (STOP-PRINTER-OUTPUT command or STPSPPR macro).

RSOPAR=*NONE / <var:pointer>

Address of the RSO parameter list. It is only specified when specific RSO parameters are used.

RSOPAR=*NONE

No RSO parameter list address.

RSOPAR=<var:pointer>

A pointer is defined, i.e. the variable or the field do not contain the RSO parameter list itself, rather the address of the storage location at which the RSO parameter list is stored (A(field)).

DESTTYP=*NONE / *STD /*LIST

Specifies the output device pool type.

DESTTYP=*NONE

There is no device pool, e.g. because the list of local device pools was deleted for a specific printer and spoolout jobs for that pool cannot therefore be processed.

DESTTYP=*STD

The list of local pools is replaced by a list of standard pools (spool parameter file) containing the printers specified.

DESTTYP=*LIST

List of pools to which the spoolout jobs for the specified printers can be directed.

DESTNAM=array(16): <var:char:8> / *CENTRAL / <c-string 1..8: c-string 1..8>

Specifies the name of or a list of names for the output device pools. Spoolout jobs for the specified printers can be directed there.

DESTNAM=*CENTRAL

The printer can process local spoolout jobs without an output device pool being specified.

FORMTYP = *ALL/*STD/*LIST/*EQUIVALENT/*EXCEPT/*EQUIVALENT-EXCEPT

Specified the entry type for form names.

FORMTYP=*ALL

Spoolout jobs can be carried out, regardless of the form requested by the specified printer.

FORMTYP=*STD

Spoolout jobs can be performed with all forms allowed for the specified printer in the spool parameter file.

FORMTYP=*LIST

Spoolout jobs can be performed with all forms specified in the FORMNAM operand.

FORMTYP=*EQUIVALENT

Equivalent forms with which the spoolout jobs can be performed on the specified printers. The printout is not interrupted when the form changes.

FORMTYP=*EXCEPT

Forms with which spoolout jobs cannot be processed on the specified printers.

FORMTYP=*EQUIVALENT-EXCEPT

Equivalent forms with which the spoolout jobs cannot be performed on the specified printers.

FORMNAM=array(16): <var:char:6> / <c-string 1..6: c-string 1..6>

List of forms with which the spoolout jobs can be processed on the specified printers.

OVLTYP=*ALL/*NONE/*ONLY/*EXCEPT/*LIST

Specifies the overlay types with which the spoolout jobs can be processed on the specified laser printers.

OVLTYP=*ALL

All spoolout jobs can be processed on the specified printers using an optional overlay.

OVLTYP=*NONE

Only print jobs that do not use an overlay can be processed on the specified printers.

OVLTYP=*ONLY

Only spoolout jobs that use overlays can be processed on the specified printers.

OVLTYP=*EXCEPT

Name of the overlays with which spoolout jobs cannot be processed on the specified printers.

OVLTYP=*LIST

List of the overlays with which spoolout jobs can be processed on the specified printers.

OVLNAM=array(16): <var:char:2> / <c-string 1..2: c-string 1..2>

List of the overlays with which spoolout jobs can be processed on the specified printers.

HOSTTYP=*ALL-CLUSTERS / *HOME / *LOCAL-CLUSTER / *EXCEPT / *LIST / *CLIST

Specifies that the printer can process all spoolout jobs from a specific host.

HOSTTYP=*ALL-CLUSTERS

Spoolout jobs from all hosts in all clusters are processed on the specified printer.

HOSTTYP=*HOME

Only spoolout jobs from the local host are processed on the specified printer.

HOSTTYP=*LOCAL-CLUSTER

Spoolout jobs from all hosts in the local cluster are processed on the specified printer.

HOSTTYP=*EXCEPT

Spoolout jobs from all hosts, with the exception of the hosts specified in the HOSTNAM list, are processed on the specified printer.

HOSTTYP=*LIST

Spoolout jobs from the hosts listed in the HOSTNAM operand are processed.

HOSTTYP=*CLIST

Only print jobs from the hosts in this list are processed. All the elements of the list must be of type c-string, with a maximum length of 8 characters (length estimated without quotes).

HOSTNAM=array(16): <var:char:8> / <c-string 1..8: c-string 1..8>

Print jobs from the hosts in this list only are processed by the specified printer.

PAPROUT=*NO / *ALLOWED

Specifies whether the spoolout jobs for laser printers can also be processed on line printers.

PSAMPLE=*NO / *YES

Specified whether a sample printout is to be made for spoolout jobs that are processed on the basis of the allocation operands. PRINT-SAMPLE=*YES is rejected for LP65, PCL and SPS printers.

For sample printouts on HP or HP90 printers, character sets with proportional spacing may not be used. The characters in the file are replaced by '*' and '0' and have different widths in proportional spacing.

TRACE=*NO / COMPLETE / STATUS / *BLOCK_CONTROL

Activates or deactivates a trace. Name of the trace file:

- for SPOOL:
 - \$\$Y\$\$POOL.\$Y\$TRC.\$POOL.\$
- for SPS:
 - \$SYSSPOOL.SYSTRC.SPS.<dvcname>.<yyyy-mm-dd>.<hhmmss>.<W/S> (W for Winter/S for Summer)
- for RSO:
- \$SYSPOOL.SYSTRC.RSO.<dvcname>.yyyy-mm-dd.ss.mm

TRACE=*NO

No trace is to be recorded.

TRACE=COMPLETE

A complete trace is to be recorded.

TRACE=STATUS

For SPS printers only: trace and status data is recorded.

TRACE=*BLOCK CONTROL

For SPS printers only: trace, status and block control data is recorded.

USERTYP=*ALL / *EXCEPT / *LIST / *CLIST

Specifies the user type under which the spoolout jobs can be processed on the specified printers.

USERTYP=*ALL

Spoolout jobs can be processed on the specified printers, regardless of the user type.

USERTYP= *EXCEPT

Spoolout jobs cannot be processed on the specified printers under the specified user types.

USERTYP= *LIST

List of user types specified in the USRNAM operand under which spoolout jobs can be processed on the specified printers.

USERTYP=*CLIST

List of user types under which the spoolout jobs can be processed on the specified printers. All the elements of the list must be of type c-string, with a maximum length of 8 characters (length estimated without quotes).

USERNAM:=array(16): <var:char:8> / <c-string 1..8: c-string 1..8>

List of user types under which spoolout jobs can be processed on the specified printers.

CLASTYP=*ALL/*EXCEPT/*LIST

Spoolout classes from which spoolout jobs can be processed on the specified printers.

CLASTYP=*ALL

Spoolout jobs can be processed on the specified printers, regardless of the spoolout class.

CLASTYP=*EXCEPT

Spoolout class from which spoolout jobs cannot be processed on the specified printers.

CLASTYP=*LIST

List of spoolout classes from which spoolout jobs can be processed on the specified printers.

CLASNAM=array(16): <var:int:1>/<integer 1..255>

List of spoolout classes from which spoolout jobs can be processed on the specified printers.

SPNATYP=*ALL / *EXCEPT / *LIST / *CLIST

Job names under which spoolout jobs can be processed on the specified printers.

SPNATYP=*ALL

Spoolout jobs can be processed on the specified printers, regardless of the job name.

SPNATYP=*EXCEPT

Spoolout jobs cannot be processed on the specified printers under the specified job names.

SPNATYP =*LIST

List of job names under which spoolout jobs can be processed on the specified printers.

SPNATYP=*CLIST

List if job names under which spoolout jobs can be processed on the specified printers. All the elements of the list must be of type c-string, with a maximum length of 8 characters (length estimated without quotes).

SPNANAM=array(16): <var:char:8> / <c-string 1..8: c-string 1..8>

List of job names under which spoolout jobs can be processed on the specified printers.

ACCTYP=*ALL/*EXCEPT/*LIST

Account numbers of the spoolout jobs that can be processed on the specified printers.

ACCTYP=*ALL

Spoolout jobs can be processed on the specified printers, regardless of the account number.

ACCTYP=*EXCEPT

Spoolout jobs cannot be processed on the specified printers with the specified account numbers.

ACCTYP=*LIST

List of account numbers with which spoolout jobs can be processed on the specified printers.

ACCNAM=array(16): <var:char:8> / <c-string 1..8: c-string 1..8>

List of account numbers under which spoolout jobs can be processed on the specified printers.

ROT= *ANY/*NO/*YES/*MANUAL

Specifies whether spoolout jobs that require the page rotation module can be processed on the specified printers.

ROT=*ANY

Any spoolout jobs can be processed on the specified printers, regardless of whether they require the page rotation module or not.

ROT=*NO

Only spoolout jobs that do not require the page rotation module can be processed on the specified printers.

ROT= *YES

Only spoolout jobs that require the page rotation module can be processed on the specified printers.

ROT=*MANUAL

As opposed to the value *NO, the operator can access the page rotation module through a hardware switch. All pages are rotated.

TWO_UP=<u>*ANY</u> / *YES / *NO / *MODE-1 / *MODE-2

For local spool only.

Specifies the printer-specific TWO-UP-PROCESSING feature for spoolout jobs on HP90 printers:

Specifies whether and in which order two parallel pages are to be output. The "SPOOL (BS2000/OSD)" manual contains a detailed description of the various 'TWO-UP' modes.

FOB=(type, low, high)

Specifies whether print jobs using an FOB overlay can be processed on the specified printers.

type: *ANY/*ONLY/*NO/*RANGE

Print job types which can be processed on the specified printers.

type:*ANY

Both print jobs with and without FOB overlays can be processed on the specified printers. The highest possible FORMS-OVERLAY-BUFFER is contained in the relevant default device entry.

type:*ONLY

Only print jobs that require an FOB overlay can be processed on the specified printers. The highest possible FORMS-OVERLAY-BUFFER is contained in the relevant default device entry.

type:*NO

Print jobs that require an FOB overlay cannot be processed on the specified printers.

type:*RANGE

Only print jobs that require an FOB overlay that is within the specified size range can be processed on the specified printers.

low:*STD/ <var:int:2> / <integer 0..32767>/

Lower range limit (specified in sublines). *STD is the value 0.

high: *STD / <var:int:2> / <integer 0..32767>

Upper range limit for HP90 printers: 4032 sublines. Upper range limit for HP-PRINTER: 672 sublines.*STD is the value 4032.

PRIO=(type, low, high)

Priority of the print jobs that can be processed on the specified printers.

type:*ALL/*RANGE

Print job types that can be processed on the specified printers.

type:<u>*ALL</u>

Print jobs can be processed on the specified printers, regardless of their priority.

type:*RANGE

Print jobs with a priority within the range specified can be processed on the specified printers.

low: *MIN/<var: int: 1>/<integer 30..255> Lower range limit. *MIN is the value 30.

high: *MAX/<var: int: 1>/<integer 30..255>
Upper range limit.*MAX is the value 255.

CHARSN=(type, low, high)

Number of character sets required with which print jobs can be processed on the specified printers.

type:*ALL /*ONE / *RANGE

Character sets with which print jobs can be processed on the specified printers.

type:*ALL

Print jobs can be processed on the specified printers, regardless of the number of character sets required. The maximum number of character sets required is contained in the default device entry.

type:*ONE

Only print jobs that require only one character set can be processed on the specified printers.

type: *RANGE

Print jobs whose number of required character sets is within the specified range can be processed on the specified printers. While a device is being initialized, the system checks how many character sets can be loaded.

If the value is smaller than one of the values specified, a message is issued to the operator console and the value for high is adjusted (if $low \le$ number of character sets that can be loaded on the device) or the command is rejected (if low > number of character sets that can be loaded on the device).

low:*MIN / <var:int:2> / <integer 1..32767>

Lower range limit. *MIN is the value 1.

high:*MAX / <var:int:2> / <integer 1..32767>

Upper range limit. *MAX is the value 64.

EXITR=*ACTIVE / *NOT-ACTIVE

Specifies whether EXIT routines are to be called in the spoolout.

Return codes

(SC2)	SC1	Maincode	Meaning
00	00	0000	No error
02	00	xxxx	Warning - printer does not exist or processing not allowed
02	01	FFFF	Invalid parameter list area
01	01	xxxx	Invalid DEVICE operand
02	01	xxxx	Invalid DESTINATION operand
03	01	xxxx	Invalid FORM-NAME operand
04	01	xxxx	Invalid OVERLAY operand
05	01	xxxx	Invalid HOST-NAME operand
06	01	xxxx	Invalid PAGE-PRINTER-OUTPUT operand

(SC2)	SC1	Maincode	Meaning
07	01	XXXX	Invalid PRINT-SAMPLE operand
80	01	xxxx	Invalid TRACE operand
09	01	xxxx	Invalid ACCESS operand
0A	01	xxxx	Invalid USERID operand
0B	01	xxxx	Invalid SPOOLOUT-CLASS operand
0C	01	xxxx	Invalid JOB-NAME operand
0D	01	xxxx	Invalid ACCOUNT operand
0E	01	xxxx	Invalid ROTATION operand
0F	01	xxxx	Invalid TWO-UP operand
10	01	xxxx	Invalid FORM-OVERLAY-BUFFER operand
11	01	xxxx	Invalid PRIORITY operand
12	01	xxxx	Invalid CHARACTER-SET-NUMBER operand
13	01	xxxx	Invalid RSO parameter list extension address
14	01	xxxx	Invalid REVISION operand
15	01	xxxx	Invalid EXIT operand
17	01	xxxx	Invalid dprintpl address
18	01	xxxx	Invalid sps pl address
1A	01	xxxx	Invalid dsem pl address
1B	01	xxxx	Invalid buffer address
1C	01	xxxx	Invalid buffer length
1D	01	xxxx	Invalid SCHEDULING-STATE operand
1E	01	xxxx	Invalid STATUS operand
1F	01	xxxx	Invalid CLUSTER operand
20	01	xxxx	Invalid SERVER operand
21	01	xxxx	Invalid ERROR-MESSAGE operand
00	20	xxxx	System error
02	40	XXXX	No authorization
04	40	xxxx	Operand value error
05	40	XXXX	Parameter list version is not supported
02	80	xxxx	RSO subsystem not loaded
05	80	xxxx	SPS subsystem not loaded
06	80	xxxx	Dprintcl subsystem not loaded
07	80	xxxx	Dprintcm subsystem not loaded
08	80	xxxx	Dprintsv subsystem not loaded
FF	FF	FFFF	Spool subsystem not loaded

Note

The characters xxxx in the Maincode column represent the message number. Enter the command HELP-MSG SCPxxxx or SPSxxxx in SYSTEM MODE in order to obtain detailed information on an error.

Parameter list description

```
dev type set
         AIF ('&EQUATES' NE 'YES').Q01001
STRANYL
           FOU
                 1
                                          local device
           FOU
                 2
                                          RSO device
STRRSO
.Q01001 ANOP
    parameter list description
STR MDL
           DS
                 0 F
         FHDR MF=(C,STR),EQUATES=NO
STRHDR
                                         Standard header
   main return codes
         ATF
             ('&EQUATES' NE 'YES').Q01002
STRRCNO
           FOU
                 0
                                          subcode1 = 00 : no error
STRWARN
          EQU
                 2
                                          printer not found or
                                          processing not permitted
STRVALM FOU
                 2
                                          subcode1 = 01 : pl area
                                          invalid - maincode = FFFF
          EQU
                 1
                                          subcode1 = 01 : wrong
STRPARM
                                          parameter
                                          subcode1 = 01 : device
STRDEVI
           EQU
                 1
STRDEST
           EQU
                 2
                                          subcode1 = 01 : destination
STRFONA
           FOU
                 3
                                          subcode1 = 01 : form name
STROVRL
           EQU
                 4
                                          subcode1 = 01 : overlay
           EQU
                 5
                                          subcode1 = 01 : host
STRHOST
STRPPOU
           FOU
                 6
                                          subcode1 = 01 : page printer
                                          output
STRPRSA
           EQU
                 7
                                          subcode1 = 01 : print sample
STRTRAF
           FOU
                 8
                                          subcode1 = 01 : trace
STRACCE
           EQU
                 9
                                          subcode1 = 01 : access
STRUSER
           EQU
                 10
                                          subcode1 = 01 : userid
STRSPCI
           FOU
                 11
                                          subcode1 = 01 : spoolout
                                          class
STRJNAM
           EQU
                 12
                                          subcode1 = 01 : job name
STRACCN
           FOU
                 13
                                          subcode1 = 01 : account
STRROTA
           EQU
                 14
                                          subcode1 = 01 : rotation
                                          subcode1 = 01 : two-up
STRTWOU
           EQU
                 15
           EQU
                 16
                                          subcode1 = 01 : fob
STRFOB
STRPRIO
           EQU
                 17
                                          subcode1 = 01 : priority
STRCSNU
           EQU
                 18
                                          subcode1 = 01 : char set num
STRRSOL
           EQU
                 19
                                          subcode1 = 01 : addr of rso
                                          ٦q
STRREVI
           EQU
                 20
                                          subcode1 = 01 : revision
           EQU
                 21
                                          subcode1 = 01 : exit
STREXTT
STRSYSE
           EQU
                                          subcode1 = 20 : system error
                 0
STRPRIV
           EQU
                 2
                                          subcode1 = 40 : no
                                          authorization
STRVERR
           EQU
                 4
                                          operand value error
```

```
STRVERE
            EQU
                  5
                                             p/l version not supported
            EQU
                  0
                                             subcode1 = 80 : subsystem not
STRNOSS
                                             loaded
STRNRS0
            EQU
                  2
                                             rso not loaded
STRNDCL
            EQU
                  6
                                             dprintcl not loaded
STRNDCM
                  7
            EQU
                                             dprintcm not loaded
STRNDSV
            EQU
                  8
                                             dprintsv not loaded
STRNSP0
            EQU
                   255
                                             subcode1 = 255: spool not
                                             loaded
.Q01002
        ANOP
            DS
                  0XL132
STRADVNM
                                             device name
STRAFC8B
            DS
                  FL1
    flag set
         AIF
                ('&EQUATES' NE 'YES').Q01003
STREALL
            FOU
                                             _{type} = *all
STRFNON
            EQU
                  1
                                             _{type} = *none
                  2
STRFL OC
            EQU
                                             _{type} = *local
STRFRFM
            EQU
                  3
                                             type = *remote
STRFCEN
            EQU
                  4
                                             _{type} = *central
STRFDFV
            EQU
                  64
                                             _type = *device
STRFMDV
            EQU
                  128
                                             type = *manag dev
STRFWLD
            FOU
                  5
                                             _type = *wildcard
STRFI ST
            EQU
                  6
                                             _{type} = *list
STRESTD
            EQU
                  7
                                             type = *std
STRFOWN
            EQU
                  8
                                             _{type} = *own
STRFHOM
            EQU
                  9
                                             _{type} = *home
STRFIDE
            EQU
                  10
                                             type = *ident
STRFANY
            EQU
                  11
                                             _{type} = *any
            FOU
                  12
                                             _type = *only
STRFONI
STRFRNG
            EQU
                  13
                                             _type = *range
            EOU
                  14
STRFONE
                                             type = *one
                                             _{type} = *no
STRFN0
            FOU
                  15
STREQU
            EQU
                  16
                                             type=*equivalent
                  17
STREOUX
            EQU
                                             type=*equivalent ex
STRFXCF
            FOU
                  18
                                             _type=*except
STRALL0
            EQU
                  19
                                             _type=*allowed
STRFYES
            EQU
                  20
                                             _{type} = *_{yes}
STRFCOM
            FOU
                  21
                                             _type = *complete
STRESTA
            EQU
                  22
                                             _type = *status
STRFBLK
            EQU
                  23
                                             type = *block control
            FOU
                  24
STRFMAN
                                             _{type} = *manual
STRFM01
            EQU
                  25
                                             _{type} = *mode-1
            EQU
                  26
                                             _{type} = *mode2
STRFM02
                  27
                                             _type = *active
STRFACT
            FOU
STRFNOA
            EQU
                  28
                                             _{type} = *no-active
            EQU
                  29
                                             _{type} = *unchanged
STRFUNC
```

```
STRFADD EOU
               30
                                     _{type} = *add
                                     _{type} = *remove
STRFREV
        EQU
               31
STRFUPD EQU
               32
                                     _{type} = *update
                                     _type = *minprio
STRFMPR EOU
               30
STRFMAP EOU
               255
                                     _type = *maxprio
STRFMCH
        EQU
                                     _type = *minchars
               1
STRFMXC EOU
               64
                                     _type = *maxchars
.001003 ANOP
STRAUC8B
         DS
               XL3
STRAEC8B
        DS
               8CL8
        AIF ('&EQUATES' NE 'YES').Q01004
STRAEC8B# EQU
.001004 ANOP
STRAEC82 DS
              8CL8
        AIF ('&EQUATES' NE 'YES').Q01005
STRAEC82# EOU
.Q01005 ANOP
*
STRDDEST DS 0XL132
                                     destination
STRDFC8 DS
              FL1
STRDUC8
        DS
              XI3
STRDFC8
        DS
              16CL8
        AIF ('&EQUATES' NE 'YES').Q01006
STRDEC8# EOU
.Q01006 ANOP
STRAFRMN DS 0XL100
                                     form name
STRAFC6 DS
              FI1
              XL3
STRAUC6
        DS
STRAEC6
        DS
              16CL6
             ('&EQUATES' NE 'YES').Q01007
        ATF
STRAEC6# EOU 16
.001007 ANOP
STRAFOVL DS
               0XL36
                                     form overlay
STRAFC2 DS
               FI1
STRAUC2
        DS
              XL3
STRAEC2
        DS
              16CL2
        ATF
             ('&EQUATES' NE 'YES').Q01008
STRAEC2# EQU 16
.001008 ANOP
               0XL132
STRBHOST DS
                                     host name
```

```
STRBFC8
            DS
                  FL1
STRBUC8
            DS
                  XL3
STRBEC8
            DS
                  16CL8
         AIF
                ('&EQUATES' NE 'YES').Q01009
STRBEC8#
            EQU
                  16
.Q01009 ANOP
STRPP0
            DS
                  FL1
                                            page printer output
STRPSAM
            DS
                  FL1
                                            print sample
STRTRAC
            DS
                  FL1
                                            trace
STRUN1
            DS
                  XL1
STREACCS
            DS
                  0XL20
                                            allowed accesses
STREFC4
            DS
                  FL1
STREUC4
            DS
                  XL3
                  4CL4
STREEC4
            DS
         AIF
                ('&EQUATES' NE 'YES').Q01010
STREEC4#
            EQU
.Q01010 ANOP
STRCUSRD
            DS
                  0XL132
                                            userid
STRCFC8
            DS
                  FL1
STRCUC8
            DS
                  XL3
STRCEC8
            DS
                  16CL8
                ('&EQUATES' NE 'YES').Q01011
         AIF
STRCEC8#
            FOU
                  16
.Q01011
         ANOP
STRASPCI
            DS
                  0XI20
                                            spoolout class
STRAFI1
            DS
                  FL1
            DS
                  XL3
STRAUC1
STRAFI1
            DS
                  16X
STRESPNA
            DS
                  0XI 132
                                            spoolout name
STREFC8
            DS
                  FL1
STREUC8
            DS
                  XL3
STREEC8
            DS
                  16CL8
         AIF
                ('&EQUATES' NE 'YES').Q01012
            EQU
STREEC8#
                  16
.Q01012
         ANOP
STRFACNT
            DS
                  0XL132
                                            account
STRFFC8
            DS
                  FL1
            DS
                  XL3
STRFUC8
```

```
STRFEC8
           DS
                  16CL8
                ('&EOUATES' NE 'YES').001013
         AIF
STRFEC8#
            FOU
                  16
.001013 ANOP
STRROT
            DS
                  FI1
                                             rotation
STRTUP
            DS
                  FL1
                                             two-up processing
STRUN2
            DS
                  XL2
STRMFOB
            DS
                  0XL8
                                             form-overlay-buffer
STRMFR2
            DS
                  FL1
STRMUN1
            DS
                  X I 1
STRMLR2
            DS
                  Н
STRMHR2
            DS
STRMUR2
            DS
                  XL2
*
STROPRIO
            DS
                  0XL4
                                             priority
                  FL1
STROFR1
            DS
STROLR1
            DS
                  χ
STROHR1
            DS
                  Χ
STROUR1
            DS
                  XL1
            DS
STRACHAR
                  0XL8
                                             character-set-number
STRAFR2
            DS
                  FI1
                  X I 1
STRAUN1
            DS
STRALR2
            DS
                  Н
STRAHR2
            DS
                  Н
STRAUR2
            DS
                  XL2
STRRSOP
            DS
                  Α
                                             address of rso pl
STRPTRRES1 DS
                                             address of reserved1
                  Α
STRPTRRFS2 DS
                                             address of reserved2
                  Α
STRPTRRES3 DS
                                             address of reserved3
                  Α
            DS
                                             revision number
STRREVIS
                  Χ
STRUN4
            DS
                  XI3
STREXIT
            DS
                  FL1
                                             exit-processing
STRSP1
            DS
                  AL1
                                             specified1
         ATF
                ('&EQUATES' NE 'YES').Q01014
STRSP11
            EQU
                  X'80'
                                             device type
STRSP12
            EQU
                  X'40'
                                             rso param
STRSP13
            EQU
                  X'20'
                                             destination type
STRSP14
            EQU
                  X'10'
                                             form type
STRSP15
            EQU
                  X'08'
                                             overlay type
STRSP16
                  X'07'
            EQU
                                             unused
.Q01014 ANOP
STRSP2
            DS
                  AL1
                                             specified2
```

```
AIF
                ('&EOUATES' NE 'YES').001015
           EQU
STRSP21
                  X'80'
                                            host type
STRSP22
           EQU
                  X'40'
                                            page pr. output
STRSP23
           EQU
                  X'20'
                                            print sample
STRSP24
           EQU
                  X'10'
                                            trace
STRSP25
           EQU
                  X'08'
                                            userid type
STRSP26
           EQU
                  X'07'
                                            unused
.001015 ANOP
STRSP3
           DS
                  AL1
                                            specified3
         ATF
                ('&EQUATES' NE 'YES').Q01016
STRSP31
           EQU
                  X'80'
                                            spoolout class type
STRSP32
           EQU
                  X'40'
                                            spoolout name type
STRSP33
           EQU
                  X'20'
                                            spoolout name list
STRSP34
           EQU
                  X'10'
                                            account type
STRSP35
           EQU
                  X'08'
                                            rotation
STRSP36
           EQU
                  X'04'
                                            two-up processing
STRSP37
           EQU
                  X'02'
                                            charsetnum low
STRSP38
           EQU
                  X'01'
                                            charsetnum high
         ANOP
.Q01016
STRSP4
           DS
                  AI1
                                            specified4
         ATF
                ('&EQUATES' NE 'YES').Q01017
STRSP41
           EQU
                  X'80'
                                            fob type
STRSP42
           EOU
                  X'40'
                                            fob range low
STRSP43
           FOU
                  X'20'
                                            fob range high
                  X'10'
STRSP44
           EQU
                                            priority
STRSP45
           EQU
                  X'08'
                                            priority from
STRSP46
                  X'04'
           EQU
                                            priority to
STRSP47
           EQU
                  X'02'
                                            charsetnum
STRSP48
           EQU
                  X'01'
                                            exit routines
.Q01017
        ANOP
STRUN3
           DS
                  XI 43
         ATF
                ('&EQUATES' NE 'YES').Q01018
STR#
           EQU
                  *-STRHDR
```

.001018 ANOP

STRSRSO - allocate printers with RSO-specific parameters

User group: Systems support staff, RSO device administrator

Programming language: Assembler, CPP, C

Macro type: M, S

The STRSRSO macro is used to allocate RSO print jobs to printers.

Format

Operation	Operands
STRSRSO	VARIANT = <u>001</u> / <c-string-without-quotes 33=""></c-string-without-quotes>
	ACSTYP = *STD / *LIST
	ACSNAM = array(4): <var: char:4=""> / <c-string 1-4="" 18:="" c-string=""></c-string></var:>

Description of the operands

VARIANT=001 / <c-string-without-quotes 3..3>

Specifies the variant of the generated parameter list.

ACSTYP=*STD / *LIST

This operand specifies which access types are allowed. It can be used for any RSO printer.

ACSTYP=*STD

Only RSO jobs can be processed on the printer.

ACSTYP=*LIST

List of allowed access types specified in the ACSNAM operand.

ACSNAM=array(4): <var:char:4> / <c-string 1..8: c-string 1..4>

Name of allowed access type. The following entries can be made:

C`RSO` RSO print jobs can be allocated to the printer when it is active. C`UTM` UTM print jobs can be allocated to the printer when it is active.

Parameter list description

```
parameter list description
SSR_MDL DS
              0F
SSRHDR FHDR MF=(C,SSR), EQUATES=NO Standard header
   main return codes
        ATF
             ('&EQUATES' NE 'YES').Q01001
SSRRCNO FOU
              0
                                         subcode1 = 00 : no error
SSRWARN EOU 2
                                         job not found or processing
                                         not permitted
SSRPARM EQU
               1
                                         subcode1 = 01 : wrong
                                         parameter
SSRVALD EQU
               2
                                         invalid address/length
               3
SSRSHRT FOU
                                         output area too short
SSRSYSE EQU
              0
                                         subcode1 = 20 : system error
SSRPRIV FOU
               2
                                         subcode1 = 40 : no
                                         authorization
SSRVERR EOU
               4
                                         operand value error
SSRVFRF FOU
               5
                                         p/l version not supported
SSRNOSS FOU
                                         subcode1 = 80 : subsystem not
                                         loaded
SSRNSPO EQU
                                         spool not loaded
              1
SSRNRSO FOU
                                         rso not loaded
SSRNPLM EOU
               3
                                         plam not loaded
              4
SSRNJV EQU
                                         iv not loaded
SSRNSPS FOU
              5
                                         sps not loaded
SSRNDCL EOU
              6
                                         dprintcl not loaded
SSRNDCM EQU 7
                                         dprintcm not loaded
SSRNDSV FOU
              8
                                         dprintsv not loaded
SSRNDSM EQU
               9
                                         dsem not loaded
.Q01001 ANOP
SSRAACCS DS
              0XL20
                                         allowed access
SSRAFC4 DS
               FI1
   flag set
        AIF
              ('&EOUATES' NE 'YES').001002
SSRFALL FOU
               \cap
                                         type = *all
SSRFNON EQU
              1
                                         _{type} = *none
                                         _{type} = *local
SSRFLOC EOU
               2
SSRFRFM FOU
               3
                                         _{type} = *remote
SSRECEN FOU
                                         _{type} = *central
              4
                                         _type = *device
SSRFDEV EQU
              64
SSRFMDV EQU
              128
                                         _type = *manag dev
SSRFWLD EQU
             5
                                         _type = *wildcard
SSRELST FOU
              6
                                         _{type} = *list
SSRFSTD EQU
               7
                                         _{type} = *std
SSRFOWN EQU
               8
                                         _{type} = *own
```

```
SSRFHOM EOU
               9
                                         _{type} = *home
                                         _{type} = *ident
SSRFIDE EQU
               10
SSRFANY FOU
              11
                                         _{type} = *any
SSRFONL EQU
               12
                                         _{type} = *only
SSRFRNG EQU
               13
                                         _type = *range
SSRFONE EQU
              14
                                         _{type} = *one
SSRFNO
       EQU
              15
                                         _{type} = *no
SSREQU
         EQU
               16
                                         _type=*equivalent
SSREQUX EQU
              17
                                         _type=*equivalent_ex
SSREXCE EQU
              18
                                         _type=*except
SSRALLO EOU
              19
                                         _type=*allowed
SSRFYES EQU
              20
                                         _{type} = *yes
SSRFCOM EQU
               21
                                         _{type} = *complete
SSRFSTA EQU
               22
                                         _type = *status
SSRFBLK EQU
               23
                                         _type = *block_control
               24
SSRFMAN EQU
                                         _{type} = *manual
SSRFM01 EOU
               25
                                         _{type} = *mode-1
SSRFM02 EQU
               26
                                         _{type} = *mode2
               27
SSRFACT EQU
                                         _type = *active
SSRFNOA EQU
               28
                                          type = *no-active
.Q01002 ANOP
SSRAUC4 DS
               XL3
SSRAEC4 DS
               4CI 4
               ('&EQUATES' NE 'YES').Q01003
         AIF
SSRAEC4# EQU
.Q01003 ANOP
SSRUN1
         DS
               XL40
         ATF
              ('&EQUATES' NE 'YES').Q01004
SSR#
         FOU
               *-SSRHDR
.001004 ANOP
```

SUPSPPR - modify administration parameters for printers

User group: Systems support staff, RSO device administrator

Programming language: Assembler, CPP, C

Macro type: M, S

A modification can be made for a printer previously started with STRSPPR(X) or with the START-PRINTER-OUTPUT command using the SUPSPPR macro.

Format

Operation	Operands
SUPSPPR	ENTRY = <u>YES</u> / = NO
	VARIANT = <c-string-without-quotes 33=""> / <u>001</u></c-string-without-quotes>
	DEVTYP = *ANY-LOCAL / *RSO / <u>*UNCHANGED</u>
	RSOPAR = <var: pointer=""> / *NONE / *UNCHANGED</var:>
	DEVNAM = array(8): <var: char:8=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	DESTTYP = *NONE / *STD / *ADD / *REMOVE / *LIST / *UNCHANGED
	DESTNAME = array(16): <var: char:8=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	FORMTYP = *ALL / *STD / *LIST / *EQUIVALENT / *EXCEPT / *EQUIVALENT_EX / *ADD / *REMOVE / *UNCHANGED
	FORMNAM = array(16): <var: char:6=""> / <c-string 1-6="" 16:="" c-string=""></c-string></var:>
	OVLTYP = *ALL / *NONE / *ONLY / *EXCEPT / *LIST / *ADD / *REMOVE / *UNCHANGED
	OVLNAM = array(16): <var: char:2=""> / <c-string 1-2="" 12:="" c-string=""></c-string></var:>

(part 1 of 3)

Operation	Operands
SUPSPPR	HOSTYP = *ALL-CLUSTER / *HOME / *LOCAL-CLUSTER /*EXCEPT / *LIST / *ADD / *REMOVE / *UNCHANGED / *CLIST
	HOSTNAM = array(16): <var: char:8=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	PSAMPLE = *NO / *YES / <u>*UNCHANGED</u>
	TRACE = *NO / *COMPLETE / *STATUS / *BLOCK_CONTROL / *UNCHANGED
	USERTYP = *ALL / *EXCEPT / *LIST / *ADD / *REMOVE / *UNCHANGED / *CLIST
	USERNAM = array(16): <var: char:8=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	CLASTYP = *ALL / *EXCEPT / *LIST / *ADD / *REMOVE / *UNCHANGED
	CLASNAM = array(16): <var: int:1=""> / <integer 1255=""></integer></var:>
	SPNATYP = *ALL / *EXCEPT / *LIST / *ADD / *REMOVE / *UNCHANGED / *CLIST
	SPNANAM = array(16): <var: char:8=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	ACCTYP = *ALL / *EXCEPT / *LIST / *ADD / *REMOVE / *UNCHANGED
	ACCNAM = array(16): <var: char:8=""> / <c-string 1-8="" 18:="" c-string=""></c-string></var:>
	ROT = *NO / *YES / *ANY / *MANUAL / *UNCHANGED
	TWO_UP = *NO / *YES / *ANY / *MODE-1 / *MODE-2 / *UNCHANGED

(part 2 of 3)

Operation	Operands
SUPSPPR	F_O_B = (type, low, high) type: *ANY / *ONLY / *NO / *RANGE / *UNCHANGED low: *STD / <var: int:2=""> / <integer 032767=""> / *UNCHANGED high: *STD / <var: int:2=""> / <integer 032767=""> / *UNCHANGED</integer></var:></integer></var:>
	PRIO = (type, from, to) type: *ALL / *RANGE / *UNCHANGED from: *MIN / <var: int:1=""> / <integer 30255=""> / *UNCHANGED to: *MAX /<var: int:1=""> / <integer 30255=""> / *UNCHANGED</integer></var:></integer></var:>
	CHARSN = (type, low, high): type: *ALL / *ONE / *RANGE / :*UNCHANGED low: <var: int:2=""> / <integer 132767=""> / *MIN / *UNCHANGED high: <var: int:2=""> / <integer 132767=""> / *MAX / *UNCHANGED</integer></var:></integer></var:>
	EXITR = *ACTIVE / *NOT-ACTIVE / *UNCHANGED
	RVSION = <var: int:1=""> / <integer 1255=""></integer></var:>

(part 3 of 3)

Description of the operands

ENTRY=YES/NO

Specifies whether the relevant entries are generated or not.

ENTRY=YES

The relevant entries are generated.

ENTRY=NO

The relevant entries are not generated.

VARIANT=001 / <c-string-without-quotes 3..3>

Specifies the variant of the generated parameter list.

DEVTYP=*ANY-LOCAL / *RSO / *UNCHANGED

Specifies the device type.

DEVTYP=*ANY-LOCAL

Administration parameters for local printers that will be specified subsequently are to be modified.

DEVTYP=*RSO

Administration parameters for RSO printers that will be specified subsequently are to be modified.

DEVTYP=*UNCHANGED

Operand default value. The current setting is not modified.

DEVNAM=array(8): <alphanum-name 1..8> / <c-string 1..8: c-string 1..8>

Logical device name defined in the spool parameter file (up to 8 alphanumeric characters). When a device with a specific name is activated (START-PRINTER-OUTPUT or MODIFY-PRINTER-OUTPUT), it can only be addressed by that name until it is deactivated again (STOP-PRINTER-OUTPUT command or STPSPPR macro).

RSOPAR=*NONE / <var:pointer> / *UNCHANGED

RSO parameter list address. It is only specified when specific RSO parameters are required.

RSOPAR=*NONE

No RSO parameter list address.

RSOPAR=<var:pointer>

A pointer is defined, i.e. the variable or the field do not contain the RSO parameter list itself, rather the address of the storage location at which the RSO parameter list is stored (A(field)).

RSOPAR=*UNCHANGED

Operand default value. The current setting is not modified.

DESTTYP=*NONE / *STD /*LIST / *ADD / *REMOVE / *UNCHANGED

Specifies the output device pool type.

DESTTYP=*NONE

There is no device pool, e.g. because the list of local device pools was deleted for a specific printer and print jobs for that pool cannot therefore be processed.

DESTTYP=*STD

The list of local pools is replaced by a list of standard pools (spool parameter file) containing the printers specified.

DESTTYP=*LIST

List of pools to which the spoolout jobs for the specified printers can be directed.

DESTTYP=*ADD

Pools that are to be added to an existing list of pools to which the print jobs for the specified printers can be directed.

DESTTYP=*REMOVE

Pools that are to be removed from an existing list of pools to which the print jobs for the specified printers can be directed.

DESTTYP=*UNCHANGED

Operand default value. The current setting is not modified.

DESTNAM=array(16): <var:char:8> / <c-string 1..8: c-string 1..8>

Specifies the name of or a list of names for the output device pools. Spoolout jobs for the specified printers can be directed there.

FORMTYP=*ALL_/ *STD / *LIST / *EQUIVALENT / *EXCEPT / *EQUIVALENT-EXCEPT/ *ADD /*REMOVE / *UNCHANGED

Specifies the form type for the print job.

FORMTYP=*ALL

Spoolout jobs can be carried out, regardless of the form requested by the specified printer.

FORMTYP=*STD

Spoolout jobs can be processed with all forms allowed for the specified printers in the spool parameter file.

FORMTYP=*LIST

Spoolout jobs can be processed with all forms specified in the FORMTYP operand.

FORMTYP=*EQUIVALENT

Equivalent forms with which the spoolout jobs can be processed on the specified printers. The printout is not interrupted when the form changes.

FORMTYP=*EXCEPT

Forms with which spoolout jobs cannot be processed on the specified printers.

FORMTYP=*EQUIVALENT-EXCEPT

Equivalent forms with which the spoolout jobs cannot be processed on the specified printers.

FORMTYP=*ADD

Forms that are to be added to an existing list of forms with which the spoolout jobs can be processed on the specified printers.

FORMTYP=*REMOVE

Forms that are to be removed from an existing list of forms with which the spoolout jobs can be processed on the specified printers.

FORMTYP=*UNCHANGED

Operand default value. The current setting is not modified.

FORMNAM=array(16): <var:char:6> / <c-string 1..6: c-string 1..6>

List of forms with which spoolout jobs can be processed on the specified printers. If a (positive) list of forms for the specified printers already exists it will be replaced by this new list. Printout will be interrupted and a message issued at the operator console each time the form changes.

OVLTYP=*ALL / *NONE / *ONLY / *EXCEPT / *LIST / *ADD / *REMOVE / *UNCHANGED

Specifies the overlay types with which the spoolout jobs can be processed on the specified laser printers.

OVLTYP=*ALL

All spoolout jobs can be processed on the specified printers using an optional overlay.

OVLTYP=*NONE

Only print jobs that do not use an overlay can be processed on the specified printers.

OVLTYP=*ONLY

Only print jobs that use an overlay can be processed on the specified printers.

OVLTYP=*EXCEPT

Name of the overlays with which spoolout jobs cannot be processed on the specified printers.

OVLTYP=*LIST

List of the overlays with which spoolout jobs can be processed on the specified printers.

OVLTYP=*ADD

Overlays that are to be added to an existing list of overlays with which the spoolout jobs can be processed on the specified laser printers.

OVLTYP=*REMOVE

Overlays that are to be removed from an existing list of overlays with which the spoolout jobs can be processed on the specified laser printers.

OVLTYP=*UNCHANGED

Operand default value. The current setting is not modified.

OVLNAM=array(16): <var:char:2> / <c-string 1..2: c-string 1..2>

List of the overlays with which spoolout jobs can be processed on the specified printers.

HOSTYP=*ALL-CLUSTERS / *HOME / *LOCAL-CLUSTER / *EXCEPT / *LIST / *ADD / *REMOVE / *UNCHANGED / *CLIST

Specifies that the printer can process all spoolout jobs started from a specific host.

HOSTTYP=*ALL-CLUSTERS

Spoolout jobs from all hosts in all clusters are processed on the specified printer.

HOSTTYP=*HOME

Spoolout jobs from the local host are processed on the specified printer.

HOSTTYP=*LOCAL-CLUSTER

Spoolout jobs from all hosts in the local cluster are processed on the specified printer.

HOSTTYP=*EXCEPT

Spoolout jobs from all hosts, with the exception of the hosts specified in the HOSTNAM list, are processed on the specified printer.

HOSTTYP=*LIST

Spoolout jobs from the hosts listed in the HOSTNAM operand are processed.

HOSTTYP=*ADD

The specified hosts are added to the existing ACCEPT list for the printer which names the hosts from which print jobs are processed.

HOSTTYP=*REMOVE

The specified hosts are removed from the existing ACCEPT list for the printer which names the hosts from which print jobs are processed.

HOSTTYP=*UNCHANGED

Operand default value. The current setting is not modified.

USERTYP=*CLIST

List of user types under which spoolout jobs can be processed on the specified printers. All the elements of the list must be of type c-string, with a maximum length of 8 characters (length estimated without quotes).

HOSTNAM=array(16): <var:char:8> / <c-string 1..8: c-string 1..8>

Print jobs from the hosts in this ACCEPT list only are processed by the specified printer.

PSAMPLE=*NO / *YES / *UNCHANGED

Specified whether a sample printout is to be made for spoolout jobs that are processed on the basis of the allocation operands. PRINT-SAMPLE=*YES is rejected for LP65, PCL and SPS printers.

For sample printouts on HP or HP90 printers, character sets with proportional spacing may not be used. The characters in the file are replaced by '*' and '0' and have different widths in proportional spacing.

PSAMPLE=*NO

No sample printout.

PSAMPLE=*YES

A sample printout is made.

PSAMPLE=*UNCHANGED

Operand default value. The current setting is not modified.

TRACE=*NO / COMPLETE / STATUS / *BLOCK_CONTROL / *UNCHANGED

Activates or deactivates a trace. Name of the trace file:

- for SPOOL:
 - \$SYSSPOOL.SYSTRC.SPOOL.<dev-mnemonic>.<yyyy-mm-dd>.<hh-mm>
- for SPS:
 - \$SYSSPOOL.SYSTRC.SPS.<dvcname>.<yyyy-mm-dd>.<hhmmss>.<W/S> (W for Winter/S for Summer)
- for RSO:
- \$SYSPOOL.SYSTRC.RSO.<dvcname>.yyyy-mm-dd.ss.mm

TRACE=*NO

No trace is to be recorded.

TRACE=COMPLETE

A complete trace is to be recorded.

TRACE=STATUS

For SPS printers only: trace and status data is recorded.

TRACE=*BLOCK CONTROL

For SPS printers only: trace, status and block control data is to be recorded.

TRACE=*UNCHANGED

Operand default value. The current setting is not modified.

USERTYP=*ALL / *EXCEPT / *LIST / *ADD / *REMOVE / *UNCHANGED / *CLIST

Specifies the user type under which the spoolout jobs can be processed on the specified printers.

USERTYP=*ALL

Spoolout jobs can be processed on the specified printers, regardless of the user type.

USERTYP=*EXCEPT

Spoolout jobs can be processed on the specified printers under the specified user types.

USERTYP=*LIST

List of user types specified in the USRNAM operand under which spoolout jobs can be processed on the specified printers.

USERTYP=*ADD

User types that are to be added to an existing list of user types with which the spoolout jobs can be processed on the specified printers.

USERTYP=*REMOVE

User types that are to be removed from an existing list of user types with which the spoolout jobs can be processed on the specified printers.

USERTYP=*UNCHANGED

Operand default value. The current setting is not modified.

USERTYP=*CLIST

List of user types under which spoolout jobs can be processed on the specified printers. All the elements of the list must be of type c-string, with a maximum length of 8 characters (length estimated without quotes).

USERNAM:=array(16): <var:char:8> / <c-string 1..8: c-string 1..8>

List of user types under which spoolout jobs can be processed on the specified printers.

CLASTYP=*ALL / *EXCEPT / *LIST / *ADD / *REMOVE / *UNCHANGED

Spoolout classes from which spoolout jobs can be processed on the specified printers.

CLASTYP=*ALL

Spoolout jobs can be processed on the specified printers, regardless of the spoolout class.

CLASTYP=*EXCEPT

Spoolout classes from which spoolout jobs cannot be processed on the specified printers.

CLASTYP=*LIST

List of spoolout classes from which spoolout jobs can be processed on the specified printers.

CLASTYP=*ADD

Spoolout classes that are to be added to an existing list of spoolout classes with which the spoolout jobs can be processed on the specified printers.

CLASTYP=*REMOVE

Spoolout classes that are to be removed from an existing list of spoolout classes with which the spoolout jobs can be processed on the specified printers.

CLASTYP=*UNCHANGED

Operand default value. The current setting is not modified.

CLASNAM=array(16): <var:int:1>/<integer 1..255>

List of spoolout classes from which spoolout jobs can be processed on the specified printers.

SPNATYP=*ALL / *EXCEPT / *LIST / *ADD / *REMOVE / *UNCHANGED / *CLIST Job names under which spoolout jobs can be processed on the specified printers.

SPNATYP=*ALL

Spoolout jobs can be processed on the specified printers, regardless of the job name.

SPNATYP=*EXCEPT

Spoolout jobs cannot be processed on the specified printers under the specified job names.

SPNATYP=*LIST

List of job names under which spoolout jobs can be processed on the specified printers.

SPNATYP=*ADD

Job names that are to be added to an existing list of job names with which the spoolout jobs can be processed on the specified printers.

SPNATYP=*REMOVE

Job names that are to be removed from an existing list of job names with which the spoolout jobs can be processed on the specified printers.

SPNATYP=*UNCHANGED

Operand default value. The current setting is not modified.

SPNATYP=*CLIST

List of job names under which spoolout jobs can be processed on the specified printers. All the elements of the list must be of type c-string, with a maximum length of 8 characters (length estimated without quotes).

SPNANAM=array(16): <var:char:8> / <c-string 1..8: c-string 1..8>

List of job names under which spoolout jobs can be processed on the specified printers.

ACCTYP=*ALL / *EXCEPT / *LIST / *ADD / *REMOVE / *UNCHANGED

Account numbers of the spoolout jobs that can be processed on the specified printers.

ACCTYP=*ALL

Spoolout jobs can be processed on the specified printers, regardless of the account number.

ACCTYP=*EXCEPT

Spoolout jobs cannot be processed on the specified printers with the specified account numbers.

ACCTYP=*LIST

List of account numbers with which spoolout jobs can be processed on the specified printers.

ACCTYP=*ADD

Account numbers that are to be added to an existing list of account numbers with which the spoolout jobs can be processed on the specified printers.

ACCTYP=*REMOVE

Account numbers that are to be removed from an existing list of account numbers with which the spoolout jobs can be processed on the specified printers.

ACCTYP=*UNCHANGED

Operand default value. The current setting is not modified.

ACCNAM=array(16): <var:char:8> / <c-string 1..8: c-string 1..8>

List of account numbers under which spoolout jobs can be processed on the specified printers.

ROT=*ANY / *NO / *YES / *MANUAL / *UNCHANGED

Specifies whether spoolout jobs that require the page rotation module can be processed on the specified printers.

ROT=*ANY

Any spoolout jobs can be processed on the specified printers, regardless of whether they require the page rotation module or not.

ROT=*NO

Only spoolout jobs that do not require the page rotation module can be processed on the specified printers.

ROT=*YES

Only spoolout jobs that require the page rotation module can be processed on the specified printers.

ROT=*MANUAL

As opposed to the value *NO, the operator can access the page rotation module through a hardware switch. All pages are rotated.

ROT=*UNCHANGED

Operand default value. The current setting is not modified.

TWO UP=*ANY / *YES / *NO / *MODE-1 / *MODE-2 / *UNCHANGED

For local spool only.

Specifies the printer-specific TWO-UP-PROCESSING feature for spoolout jobs on HP90 printers:

Specifies whether and in which order two parallel pages are to be output. The "SPOOL (BS2000/OSD)" manual contains a detailed description of the various 'TWO-UP' modes.

FOB=(type, low, high)

Specifies whether print jobs using an FOB overlay can be processed on the specified printers.

type: *ANY / *ONLY / *NO / *RANGE / *UNCHANGED

Print job types which can be processed on the specified printers.

type: *ANY

Both print jobs with and without FOB overlays can be processed on the specified printers. The highest possible FORMS-OVERLAY-BUFFER is contained in the relevant default device entry.

type: *ONLY

Only print jobs that require an FOB overlay can be processed on the specified printers. The highest possible FORMS-OVERLAY-BUFFER is contained in the relevant default device entry.

type: *NO

Print jobs that require an FOB overlay cannot be processed on the specified printers.

type: *RANGE

Only print jobs that require an FOB overlay that is within the specified size range can be processed on the specified printers.

type: *UNCHANGED

Operand default value. The current setting is not modified.

low: *STD / <var:int:2> / <integer 0..32767> / *UNCHANGED Lower range limit (specified in sublines). *STD is the value 0.

high: *STD / <var:int:2> / <integer 0..32767> / *UNCHANGED

Upper range limit for HP90 printers: 4032 sublines. Upper range limit for HP-PRINTER: 672 sublines.*STD is the value 4032.

PRIO=(type, from, to)

Priority of the print jobs that can be processed on the specified printers.

type: *ALL / *RANGE / *UNCHANGED

Print job types that can be processed on the specified printers.

type: *ALL

Print jobs can be processed on the specified printers, regardless of their priority.

type: *RANGE

Print jobs with a priority within the range specified can be processed on the specified printers.

type: *UNCHANGED

Operand default value. The current setting is not modified.

from: *MIN / <var: int: 1> / <integer 30..255> / *UNCHANGED

Lower range limit. *MIN is the value 30.

to: *MAX / <var: int: 1> / <integer 30..255> / *UNCHANGED

Upper range limit.*MAX is the value 255.

CHARSN=(type, low, high)

Number of character sets required with which print jobs can be processed on the specified printers.

type: *ALL /*ONE / *RANGE / *UNCHANGED

Character sets with which print jobs can be processed on the specified printers.

type: *ALL

Print jobs can be processed on the specified printers, regardless of the number of character sets required. The maximum number of character sets required is contained in the default device entry.

type: *ONE

Only print jobs that require only one character set can be processed on the specified printers.

type: *RANGE

Print jobs whose number of required character sets is within the specified range can be processed on the specified printers. While a device is being initialized, the system checks how many character sets can be loaded.

If the value is smaller than one of the values specified, a message is issued to the operator console and the value for high is adjusted (if $low \le$ number of character sets that can be loaded on the device) or the command is rejected (if low > number of character sets that can be loaded on the device).

type: *UNCHANGED

Operand default value. The current setting is not modified.

low: *MIN / <var:int:2> / <integer 1..32767> / *UNCHANGED

Lower range limit. *MIN is the value 1.

high: *MAX / <var:int:2> / <integer 1..32767> / *UNCHANGED

Upper range limit.*MAX is the value 64.

EXITR=*ACTIVE / *NOT-ACTIVE / *UNCHANGED

Specifies whether EXIT routines are to be called in the spoolout.

EXITR=*ACTIVE

EXIT routines are to be called in the spoolout.

EXITR=*NOT-ACTIVE

EXIT routines are not to be called in the spoolout.

EXITR=*UNCHANGED

Operand default value. The current setting is not modified.

RVSION=<var: int:1> / <integer 1..255>

Number of the processing status. The number specified must be that of the current processing status.

Return codes

(SC2)	SC1	Maincode	Meaning
00	00	0000	No error
02	00	xxxx	Warning - printer does not exist or processing not allowed
02	01	FFFF	Invalid parameter list area
01	01	xxxx	Invalid DEVICE operand
02	01	xxxx	Invalid DESTINATION operand
03	01	xxxx	Invalid FORM-NAME operand
04	01	xxxx	Invalid OVERLAY operand
05	01	xxxx	Invalid HOST-NAME operand
06	01	xxxx	Invalid PAGE-PRINTER-OUTPUT operand
07	01	XXXX	Invalid PRINT-SAMPLE operand
08	01	XXXX	Invalid TRACE operand
09	01	xxxx	Invalid ACCESS operand
0A	01	XXXX	Invalid USERID operand
0B	01	XXXX	Invalid SPOOLOUT-CLASS operand
0C	01	XXXX	Invalid JOB-NAME operand
0D	01	XXXX	Invalid ACCOUNT operand
0E	01	XXXX	Invalid ROTATION operand
0F	01	XXXX	Invalid TWO-UP operand
10	01	XXXX	Invalid FORM-OVERLAY-BUFFER operand
11	01	XXXX	Invalid PRIORITY operand
12	01	xxxx	Invalid CHARACTER-SET-NUMBER operand
13	01	XXXX	Invalid RSO parameter list extension address
14	01	XXXX	Invalid REVISION operand
15	01	XXXX	Invalid EXIT operand
00	20	XXXX	System error
02	40	XXXX	No authorization
04	40	XXXX	Operand value error
05	40	XXXX	Parameter list version not supported
02	80	XXXX	RSO subsystem not loaded
05	80	XXXX	SPS subsystem not loaded
06	80	XXXX	Dprintcl subsystem not loaded
07	80	XXXX	Dprintcm subsystem not loaded
08	80	XXXX	Dprintsv subsystem not loaded
FF	FF	FFFF	Spool subsystem not loaded

Note

The characters xxxx in the Maincode column represent the message number. Enter the command HELP-MSG SCPxxxx or SPSxxxx in SYSTEM MODE in order to obtain detailed information on an error.

Parameter list description

```
dev type set
         AIF ('&EQUATES' NE 'YES').Q01001
SUPANYL
           FOU
                 1
                                           local device
           FOU
                 2
                                           RSO device
SUPRSO
.Q01001 ANOP
    parameter list description
SUP MDL
           DS
                 0 F
SUPHDR
         FHDR MF=(C,SUP), EQUATES=NO
                                          Standard header
   main return codes
         ATF
             ('&EQUATES' NE 'YES').Q01002
SUPRCNO
           FOU
                 0
                                           subcode1 = 00 : no error
SUPWARN
          EQU
                 2
                                           printer not found or
                                           processing not permitted
SUPVALM
        FOU
                 2
                                           subcode1 = 01 : pl area
                                           invalid - maincode = FFFF
SUPPARM
           EQU
                 1
                                           subcode1 = 01 : wrong
                                           parameter
                                           subcode1 = 01 : device
SUPDEVI
           EQU
                 1
SUPDEST
           EQU
                 2
                                           subcode1 = 01 : destination
SUPFONA
           FOU
                 3
                                           subcode1 = 01 : form name
SUPOVRL
           EQU
                 4
                                           subcode1 = 01 : overlay
           EQU
                 5
                                           subcode1 = 01 : host
SUPHOST
SUPPPOU
           FOU
                 6
                                           subcode1 = 01 : page printer
                                           output
SUPPRSA
           EQU
                 7
                                           subcode1 = 01 : print sample
SUPTRAF
           FOU
                 8
                                           subcode1 = 01 : trace
SUPACCE
           EQU
                 9
                                           subcode1 = 01 : access
SUPUSER
           EQU
                 10
                                           subcode1 = 01 : userid
SUPSPCI
           FOU
                 11
                                           subcode1 = 01 : spoolout
                                          class
SUPJNAM
           EQU
                 12
                                           subcode1 = 01 : job name
SUPACCN
           FOU
                 13
                                           subcode1 = 01 : account
SUPROTA
           EQU
                 14
                                           subcode1 = 01 : rotation
                                           subcode1 = 01 : two-up
SUPTWOU
           EQU
                 15
           FOU
                 16
                                           subcode1 = 01 : fob
SUPFOB
SUPPRIO
           EQU
                 17
                                           subcode1 = 01 : priority
                                           subcode1 = 01 : char set num
SUPCSNU
           EQU
                 18
SUPRSOL
           EQU
                 19
                                           subcode1 = 01 : addr of rso
                                           pΊ
SUPREVI
           EQU
                 20
                                           subcode1 = 01 : revision
           EQU
                 21
                                           subcode1 = 01 : exit
SUPEXTT
SUPSYSE
           EQU
                                           subcode1 = 20 : system error
                 ()
SUPPRIV
           EQU
                 2
                                           subcode1 = 40 : no
                                           authorization
SUPVERR
           EQU
                 4
                                           operand value error
```

```
SUPVERE
            EQU
                  5
                                             p/l version not supported
SUPNOSS
            EQU
                  0
                                             subcode1 = 80 : subsystem not
                                             loaded
SUPNRSO
            EQU
                  2
                                             rso not loaded
SUPNDCL
            EQU
                  6
                                             dprintcl not loaded
SUPNDCM
            EQU
                  7
                                             dprintcm not loaded
SUPNDSV
            EQU
                  8
                                             dprintsv not loaded
SUPNSPO
            EQU
                   255
                                             subcode1 = 255: spool not
                                             loaded
.Q01002
        ANOP
            DS
                  0XL132
SUPADVNM
                                             device name
SUPAFC8B
            DS
                  FL1
    flag set
                ('&EQUATES' NE 'YES').Q01003
         AIF
SUPFALL
            FOU
                                             _{type} = *all
SUPFNON
            EQU
                  1
                                             _{type} = *none
                  2
SUPFL OC
            EQU
                                             _{type} = *local
SUPEREM
            EQU
                  3
                                             _{type} = *remote
            EQU
                                             _{type} = *central
SUPFCEN
                  4
SUPFDEV
            EQU
                  64
                                             _type = *device
SUPFMDV
            EQU
                  128
                                             type = *manag dev
SUPFWI D
            FOU
                  5
                                             _type = *wildcard
SUPFL ST
            EQU
                  6
                                             _{type} = *list
SUPESTD
            EQU
                  7
                                             type = *std
SUPFOWN
            FOU
                  8
                                             _{type} = *own
SUPFHOM
            EQU
                  9
                                             _{type} = *home
SUPFIDE
            EQU
                  10
                                             type = *ident
SUPFANY
            FOU
                  11
                                             _{type} = *any
SUPFONL
            FOU
                  12
                                             _{type} = *only
SUPERNG
            EQU
                  13
                                             _type = *range
SUPFONE
            EOU
                  14
                                             type = *one
                  15
                                             _{type} = *no
SUPFNO
            FOU
SUPEQU
            EQU
                  16
                                             type=*equivalent
            EOU
                  17
SUPEQUX
                                             type=*equivalent ex
SUPFXCF
            FOU
                  18
                                             _type=*except
SUPALLO
            EQU
                  19
                                             _type=*allowed
SUPFYES
            EQU
                  20
                                             _{type} = *_{yes}
SUPECOM
            FOU
                  21
                                             _type = *complete
SUPFSTA
            EQU
                  22
                                             _type = *status
SUPFBLK
            EQU
                  23
                                             type = *block control
            FOU
                  24
                                             _type = *manual
SUPFMAN
SUPFM01
            EQU
                  25
                                             _{type} = *mode-1
SUPFM02
            EQU
                  26
                                             _{type} = *mode2
            FOU
                  27
SUPFACT
                                             _{type} = *active
SUPFNOA
            EQU
                  28
                                             _type = *no-active
            EQU
                  29
SUPFUNC
                                             _type = *unchanged
```

```
SUPFADD
         EQU
                 30
                                         _{type} = *add
           EQU
SUPFREV
                 31
                                         _type = *remove
SUPFUPD
           EQU
                 32
                                         _{type} = *update
SUPFMPR
         EQU
                 30
                                         _type = *minprio
                 255
SUPFMAP
         EQU
                                         _type = *maxprio
SUPFMCH
           EQU
                                         _type = *minchars
                 1
SUPFMXC
           EQU
                 64
                                         _type = *maxchars
.001003 ANOP
SUPAUC8B
           DS
                XL3
SUPAEC8B
           DS
                 8CL8
         AIF
              ('&EQUATES' NE 'YES').Q01004
SUPAEC8B# EQU
.001004 ANOP
SUPAEC82
           DS
                8CL8
         ATF
               ('&EQUATES' NE 'YES').Q01005
SUPAEC82# EOU
.Q01005 ANOP
*
SUPDDEST
           DS
                0XL132
                                         destination
SUPDFC8
         DS
                FL1
SUPDUC8
           DS
                X13
SUPDEC8
           DS
                16CL8
               ('&EQUATES' NE 'YES').Q01006
         AIF
SUPDEC8#
           EQU
.Q01006 ANOP
SUPAFRMN
           DS
                0XL100
                                         form name
                FI1
SUPAFC6
         DS
                XL3
SUPAUC6
           DS
SUPAEC6
          DS
                16CL6
               ('&EQUATES' NE 'YES').Q01007
         ATF
SUPAEC6#
           EQU
.001007 ANOP
SUPAFOVL
           DS
                0XL36
                                         form overlay
SUPAFC2
           DS
                 FI1
SUPAUC2
           DS
                XI3
SUPAEC2
           DS
                16CL2
         ATF
               ('&EQUATES' NE 'YES').Q01008
SUPAEC2#
           FOU
                16
.001008 ANOP
                0XL132
SUPBHOST DS
                                         host name
```

```
DS
SUPBFC8
                  FL1
SUPBUC8
           DS
                  XL3
SUPBEC8
           DS
                  16CL8
         AIF
                ('&EQUATES' NE 'YES').Q01009
SUPBEC8#
           EQU
                  16
.Q01009 ANOP
SUPPP0
           DS
                  FL1
                                            page printer output
SUPPSAM
           DS
                  FL1
                                            print sample
SUPTRAC
           DS
                  FL1
                                            trace
SUPUN1
           DS
                  XL1
SUPEACCS
           DS
                  0XL20
                                            allowed accesses
SUPEFC4
           DS
                  FL1
SUPEUC4
           DS
                  XL3
                  4CL4
SUPEEC4
           DS
         AIF
                ('&EQUATES' NE 'YES').Q01010
SUPEEC4#
           EQU
.Q01010 ANOP
SUPCUSRD
           DS
                  0XL132
                                            userid
SUPCFC8
           DS
                  FL1
SUPCUC8
           DS
                  XL3
SUPCEC8
           DS
                  16CL8
                ('&EQUATES' NE 'YES').Q01011
         AIF
SUPCEC8#
           FOU
                  16
.Q01011 ANOP
SUPASPCI
           DS
                  0XI20
                                            spoolout class
SUPAFI1
           DS
                  FL1
SUPAUC1
           DS
                  XL3
SUPAFI1
           DS
                  16X
SUPESPNA
           DS
                  0XI 132
                                            spoolout name
SUPEFC8
           DS
                  FL1
SUPEUC8
           DS
                  XL3
SUPFFC8
           DS
                  16CL8
         ATF
                ('&EQUATES' NE 'YES').Q01012
SUPEEC8#
           EOU
                  16
.Q01012 ANOP
SUPFACNT
           DS
                  0XL132
                                            account
SUPFFC8
           DS
                  FL1
           DS
                  XL3
SUPFUC8
```

```
SUPFEC8
            DS
                  16CL8
                ('&EOUATES' NE 'YES').001013
         AIF
SUPFFC8#
            FOU
                  16
.001013 ANOP
SUPROT
            DS
                  FI1
                                             rotation
SUPTUP
            DS
                  FL1
                                             two-up processing
SUPUN2
            DS
                  XL2
SUPMFOB
            DS
                  0XL8
                                             form-overlay-buffer
SUPMFR2
            DS
                  FL1
SUPMUN1
            DS
                  X I 1
SUPMLR2
            DS
                  Н
SUPMHR2
            DS
                  Н
SUPMUR2
            DS
                  XL2
*
SUPOPRIO
            DS
                  0XL4
                                             priority
            DS
                  FL1
SUPOFR1
SUPOLR1
            DS
                  Χ
SUPOHR1
            DS
                  Χ
SUPOUR1
            DS
                  XL1
            DS
SUPACHAR
                  0XL8
                                             character-set-number
SUPAFR2
            DS
                  FI1
                  X I 1
SUPAUN1
            DS
SUPALR2
            DS
                  Н
SUPAHR2
            DS
                  Н
SUPAUR2
            DS
                  XL2
SUPRSOP
            DS
                  Α
                                             address of rso pl
SUPPTRRES1 DS
                                             address of reserved1
                  Α
SUPPTRRES2 DS
                                             address of reserved2
                  Α
SUPPTRRES3 DS
                                             address of reserved3
                  Α
            DS
                                             revision number
SUPREVIS
                  χ
SUPUN4
            DS
                  XI3
SUPEXIT
            DS
                  FL1
                                             exit-processing
SUPSP1
            DS
                  AL1
                                             specified1
         ATF
                ('&EQUATES' NE 'YES').Q01014
SUPSP11
            EQU
                  X'80'
                                             device type
SUPSP12
            EQU
                  X'40'
                                             rso param
SUPSP13
            FOU
                  X'20'
                                             destination type
SUPSP14
            EQU
                  X'10'
                                             form type
                  X'08'
SUPSP15
            EQU
                                             overlay type
SUPSP16
                  X'07'
            FOU
                                             unused
.Q01014
        ANOP
SUPSP2
            DS
                  AL1
                                             specified2
```

```
AIF
                ('&EOUATES' NE 'YES').001015
           EQU
SUPSP21
                  X'80'
                                           host type
SUPSP22
           FOU
                 X'40'
                                           page pr. output
SUPSP23
           EQU
                 X'20'
                                           print sample
SUPSP24
           EQU
                 X'10'
                                           trace
SUPSP25
           EQU
                 X'08'
                                           userid type
SUPSP26
           EQU
                 X'07'
                                           unused
.001015 ANOP
SUPSP3
           DS
                  AL1
                                            specified3
         ATF
               ('&EQUATES' NE 'YES').Q01016
SUPSP31
           EQU
                 X'80'
                                            spoolout class type
SUPSP32
           EQU
                 X'40'
                                            spoolout name type
SUPSP33
           EQU
                 X'20'
                                            spoolout name list
SUPSP34
           EQU
                 X'10'
                                            account type
SUPSP35
           EQU
                 X'08'
                                            rotation
SUPSP36
           EQU
                 X'04'
                                           two-up processing
SUPSP37
           FOU
                 X'02'
                                           charsetnum low
SUPSP38
           EQU
                 X'01'
                                           charsetnum high
        ANOP
.001016
SUPSP4
           DS
                  AI1
                                           specified4
         ATF
               ('&EQUATES' NE 'YES').Q01017
SUPSP41
           EQU
                 X'80'
                                            fob type
SUPSP42
           FOU
                 X'40'
                                            fob range low
SUPSP43
           FOU
                 X'20'
                                            fob range high
                 X'10'
SUPSP44
           EQU
                                            priority
SUPSP45
           EQU
                 X'08'
                                           priority from
SUPSP46
           EQU
                 X'04'
                                           priority to
SUPSP47
           EQU
                  X'02'
                                           charsetnum
SUPSP48
           EQU
                 X'01'
                                           exit routines
.Q01017
        ANOP
           DS
SUPUN3
                  XI 43
         ATF
                ('&EQUATES' NE 'YES').Q01018
SUP#
           EQU
                  *-SUPHDR
.001018 ANOP
```

6 Macros for virtual printers

Virtual printers are objects in the Spool & Print Services which are handled as real devices but only exist as an application which is addressed via a device name. A virtual printer can accept print jobs from SPOOL in the same way as a real device. A virtual printer either forwards these jobs immediately to a synchronously running application or makes the job data accessible to an asynchronously running application.

A programming interface is provided to enable the application to communicate synchronously with the virtual printer. This programming interface comprises the following macros:

"SPVDINI - initialize dialog with virtual printer" on page 485

Initializes the dialog mechanism between the virtual printer and the user application. The dialog mechanism must be initialized once during application startup with SPVDINI.

"SPVDRCV - set application task to ready" on page 487

Receives data from the virtual printer belonging to the application. SPVDRCV is a synchronous interface, which means that SPVDRCV waits for data from the connected virtual printer as soon as the application calls this function. The dialog with the virtual printer must be initialized with SPVDINI before SPVDRCV can be called.

"SPVDRET - send information on current job to virtual printer" on page 489

Returns the processing result of the last query to the connected virtual printer The prior SPVDRCV call must have been concluded successfully for this. SPVDRET must also be called as soon as the application has completely processed a job.

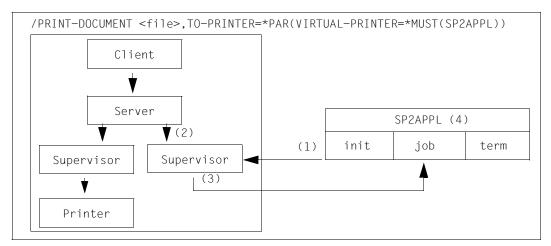
"SPVDEND - release dialog connection" on page 491

Terminates the link to the connected virtual printer.

6.1 Use of virtual printers in synchronous operation

An example of using a virtual printer in synchronous operation is shown below.

A printer SP2APPL has been entered in the SPOOL parameter file with the device type *VIRTUAL in connection with a program. This program implements special handling of the jobs and the interface for the dialog with the supervisor of the virtual printer.



(1) When the virtual device starts, the supervisor task is created. During initialization, the supervisor task checks the definition of SP2APPL and reads the description of this program. An ENTER job of this file is being executed. This task runs under \$SYSSPOOL with the job class \$SYSJC (if no user ID has been specified in the procedure name), the preassigned priority and the job name SP2APPL, the unique ID for the logon procedure.

The user can program and structure the logon procedure as he/she wishes, except that during batch processing a TU program must be loaded and started in order to establish the dialog with the supervisor via the program interface..

- (2) Every /PR-DOC VIRTUAL-PRINTER=*MUST(SP2APPL) command or every print job that matches the selection criteria specified when the virtual device was started becomes a job for the supervisor of the SP2APPL device.
- (3) The supervisor forwards the job to the application with an action code "job", the job attributes and optionally a character string which was generated at job allocation.

(4) This information enables the application to process the job. The application then informs the supervisor that job processing is finished and how it is to proceed with the job.

When the virtual printer is terminated, the application receives the action code "term". For the dialog between the application and the supervisor, a special program interface is available, which uses an FITC mechanism for the exchange of messages between the partners.

6.2 Program interface for synchronous operation

In synchronous mode, a virtual printer communicates with a user application. To facilitate this communication, a program interface is available which uses an FITC (Fast Inter-Task Communication) mechanism for the exchange of messages. This interface handles four functions:

SPVDINI initializes the dialog mechanism between the virtual printer and the user application. The dialog mechanism must be initialized once with SPVDINI

during initialization of the application.

SPVDRCV receives data from the virtual printer belonging to the application.

SPVDRCV is a synchronous interface, which means that SPVDRCV waits for data from the connected virtual printer as soon as the application has called this function. The dialog with the virtual printer must have been

initialized with SPVDINI before SPVDRCV can be called.

SPVDRET returns the result of processing the last query to the connected virtual

printer. For this to happen, the previous SPVDRCV call must have been terminated successfully. Moreover, SPVDRET must be called as soon as

the application has finished processing a job.

SPVDEND cancels the connection to the virtual printer.

An example of an user application which communicates with a virtual printer in synchronous mode using the described program interface can be found on page 498.

SPVDINI - initialize dialog with virtual printer

This macro initializes the dialog mechanism between the virtual printer and the user application. The dialog mechanism must be initialized once during application startup with SPVDINI.

Format

Operation	Operands
SPVDINI	MF=C / D / E / L ,VARIANT= <u>001</u> / <c-string 33=""> [,MACID=<u>PDI</u> / macid] [,PREFIX=<u>S</u> / p]</c-string>

Description of the operands

MF=C/D/E/L

The forms of the MF operand are described in detail in "Macro types" on page 582. The version operand must have the same value in all macro calls in which the MF operand differs (MF=L/E/D/C/M).

VARIANT=001 / <c-string 3..3>

Designates the variant of the generated parameter list.

MACID=PDI / macid

Only evaluated in conjunction with MF=C/D/M and defines the second and third characters of the field names and equates which are generated in the data area when the macro is resolved.

MACID=PDI

Default value for this operand.

MACID=macid

One- or two-character string which defines the second and third characters of the generated field names and equates.

PREFIX=S/p

Only evaluated in conjunction with MF=C/D/M and defines the first character of the field names and equates which are generated in the data area when the macro is resolved...

PREFIX=S

The default prefix with which the field names and equates generated by the assembler begin.

PREFIX=p

"p" is a one-character prefix with which the field names and equates generated by the assembler are to begin.

PREFIX=*

No prefix is generated.

Return codes

SRC2	SRC1	MRC	Meaning
00	00	0000	Executed successfully
00	00	0001	Dialog not possible
00	00	0002	Only one call via the task

SPVDRCV - set application task to ready

This smacro receives data from the virtual printer belonging to the application. SPVDRCV is a synchronous interface, which means that SPVDRCV waits for data from the connected virtual printer as soon as the application calls this function. The dialog with the virtual printer must be initialized with SPVDINI before SPVDRCV can be called.

Format

Operation	Operands	
SPVDRCV	MF = C/D/E/L/M	
	,VARIANT = <u>001</u> / <c-string 33=""></c-string>	
	,OUTADDR = <var:pointer></var:pointer>	
	,OUTLGT = <var:int:2> / <integer 132000=""></integer></var:int:2>	
	,MACID = <u>PDR</u> / <name 13=""></name>	
	,PREFIX= <u>S</u> / <name 11=""></name>	
	,PARAM = <var:pointer></var:pointer>	

Description of the operands

MF=C/D/E/L/M

The forms of the MF operand are described in detail in "Macro types" on page 582. The version operand must have the same value in all macro calls in which the MF operand differs (MF=L/E/D/C/M).

VARIANT=<u>001</u> / <c-string 3..3>

Designates the variant of the generated parameter list.

OUTADDR=<var: pointer>

Address of output buffer for data from supervisor task.

OUTLGT=<var: int:2> / <integer 1..32000 >

Length of output buffer for data from supervisor task.

MACID=PDR / <name 1..3>

Only evaluated in conjunction with MF=C/D/M and defines the second and third characters of the field names and equates which are generated in the data area when the macro is resolved.

MACID=PDR

Operand default.

MACID=<name 1..3>

One to three character string which defines the second, third and fourth characters of the generated field names and equates.

PREFIX=S / <name 1..1>

Only evaluated in conjunction with MF=C/D/M and defines the first character of the field names and equates which are generated in the data area when the macro is resolved...

PREFIX=S

The default prefix with which the field names and equates generated by the assembler begin.

PREFIX=<name 1..1>

<name 1..1> is a one-character prefix with which the field names and equates generated by the assembler are to begin.

PARAM=<var:pointer>

Address of the parameter list

Register usage

Register 1 is used for addressing the parameter list.

R15 may be modified only when the macro is called while the SPOOL subsystem is not present.

Return codes

SRC2	SRC1	MRC	Meaning
00	00	0000	Executed successfully
00	00	0001	Dialog not possible
00	00	0002	SPVDINI call expected - no data received
00	00	0004	SPVDRET call expected - no data received
00	00	8000	Work already completed - no data received
00	00	000a	Error in parameter list

SPVDRET - send information on current job to virtual printer

This macro returns the processing result of the last query to the connected virtual printer. The prior SPVDRCV call must have been concluded successfully for this. SPVDRET must also be called as soon as the application has completely processed a job.

Format

Operation	Operands
SPVDRET	MF=C/D/E/L/M
	,VARIANT=001 / <c-string 33=""></c-string>
	,PACTION= <var: _post_action_set="" enum-of=""></var:>
	,APPLERR= <var: char:8=""> / <c-string 88=""></c-string></var:>
	,OUTADD= <var: pointer=""></var:>
	,OUTLGT= <var: int:2=""> / <integer 132000=""></integer></var:>
	,MACID=PDT / <name 13=""></name>
	,PREFIX= <u>S</u> / <name 11=""> / *</name>
	,PARAM = <var:pointer></var:pointer>

Description of the operands

MF=C/D/E/L/M

The forms of the MF operand are described in detail in "Macro types" on page 582. The version operand must have the same value in all macro calls in which the MF operand differs (MF=L/E/D/C/M).

VARIANT=001 / <c-string 3..3>

Designates the variant of the generated parameter list.

PACTION=<var: enum-of _post_action_set>

Action to be applied by the virtual printer to the current job.

APPLERR=<var: char:8> / <c-string 8..8 >

Error code in printable form describing an error which the application encountered during processing of the current job.

OUTADD=<var: pointer>

Address of the output buffer containing a description of the print job. This description is an SSVSCBE parameter list, the SPOOL Control Block.

OUTLGT=<var: int:2> / <integer 1..32000 >

Length of output buffer for data from the supervisor task.

MACID=PDT / <name 1..3>

Only evaluated in conjunction with MF=C/D/M and defines the second and third characters of the field names and equates which are generated in the data area when the macro is resolved.

MACID=PDT

Operand default.

MACID = <name 1..3>

One to three character string which defines the second, third and fourth characters of the generated field names and equates.

PREFIX=S / <name 1..1>

Only evaluated in conjunction with MF=C/D/M and defines the first character of the field names and equates which are generated in the data area when the macro is resolved.

PREFIX=S

The default prefix with which the field names and equates generated by the assembler begin.

PREFIX=*

No prefix is generated.

PREFIX=<name 1..1>

<name 1..1> is a one-character prefix with which the field names and equates generated by the assembler are to begin.

PARAM = <var:pointer>

Address of the parameter list

Register usage

Register 1 is used for addressing the parameter list.

R15 may be modified only when the macro is called while the SPOOL subsystem is not present.

Return codes

SRC2	SRC1	MRC	Meaning
00	00	0000	Executed successfully
00	00	0001	Error during transfer of data to the virtual printer

SPVDEND - release dialog connection

This macro terminates the link to the connected virtual printer.

Format

Operation	Operands
SPVDEND	MF=C / D / E / L ,VARIANT=001 / <c-string 33=""> [,MACID=PDE / macid] [,PREFIX=S / p]</c-string>

Description of the operands

MF=C/D/E/L

The forms of the MF operand are described in detail in "Macro types" on page 582. The version operand must have the same value in all macro calls in which the MF operand differs (MF=L/E/D/C/M).

VARIANT=001 / <c-string 3..3>

Designates the variant of the generated parameter list.

MACID=PDE / macid

Only evaluated in conjunction with MF=C/D/M and defines the second and third characters of the field names and equates which are generated in the data area when the macro is resolved.

MACID=PDE

Operand default.

MACID=macid

One- or two-character string which defines the second and third characters of the generated field names and equates.

PREFIX=S / p

Only evaluated in conjunction with MF=C/D/M and defines the first character of the field names and equates which are generated in the data area when the macro is resolved.

PREFIX=S

The default prefix with which the field names and equates generated by the assembler begin.

PREFIX=p

"p" is a one-character prefix with which the field names and equates generated by the assembler are to begin.

PREFIX=*

No prefix is generated.

Return codes

SRC2	SRC1	MRC	Meaning
00	00	0000	Executed successfully
00	00	0001	Internal error

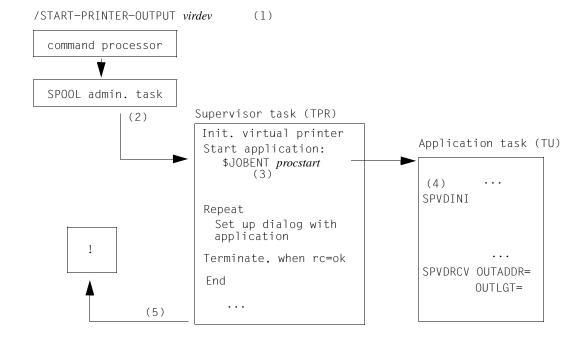
6.3 Communication between virtual printer and application

The virtual printer runs under the control of a TPR task (control task) while the application runs under the control of a TU task (application task). Data is exchanged between the two tasks.

6.3.1 Dialog initialization

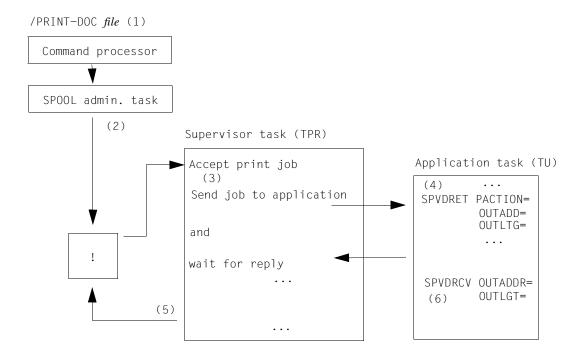
A virtual printer *virdev* was defined with the following SPSERVE statement: /ADD-SPOOL-DEVICE *virdev*, DEVICE-TYPE=*VIRTUAL, DEVICE-ACCESS=*LOCAL-ACCESS(PROGRAM-NAME=*procstart*)

procstart is the name of the procedure file which contains at least the START-PROGRAM command for the application which is connected to the virtual printer.



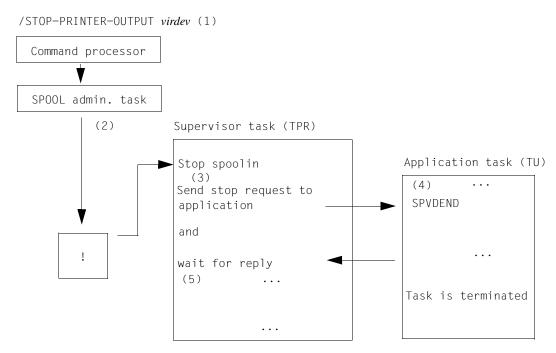
- With START-PRINTER-OUTPUT, the SPOOL administrator starts the virtual printer.
- (2) The SPOOL administrator task creates the supervisor task belonging to the virtual printer (TPR task).
- (3) The supervisor task creates an ENTER job with the procedure name it finds in the device definition, and tries to set up a dialog with the corresponding application task.
- (4) The application must initialize the dialog with SPVDINI and set itself to ready with SPVDRCV. In this call the application defines the address and the size of a storage area for receiving data from the virtual printer.
- (5) Once the dialog has been successfully established, the virtual printer waits for events which come from the SPOOL administrator task.

6.3.2 Processing print jobs



- (1) A user issues a print job.
- (2) The SPOOL administrator task forwards the print job to the virtual printer according to the rules for distribution.
- (3) The print job is transferred to the application. The supervisor task then waits for the response from the application.
- (4) The application receives the description of the print job (return of SPVDRCV), processes the print job and sends the result to the virtual printer. The application returns error codes, the processed print job and its new status to the virtual printer via the SPVDRET interface.
- (5) The virtual printer waits again for results which come from the SPOOL administrator task.
- (6) The application sets itself to ready and waits for events to come from the virtual printer.

6.3.3 Stopping processing



- (1) The SPOOL administrator issues the /STOP-PRINTER-OUTPUT command.
- (2) The SPOOL administrator task informs the supervisor task that processing is terminating.
- (3) The supervisor task passes on the stop command to the application.
- (4) The application stops its work, releases the dialog environment with SPVDEND and terminates the task.
- (5) The virtual printer stops its work and the supervisor task is terminated.

6.3.4 Error recovery

1. Error during initialization of the virtual printer

The virtual printer is forcibly terminated. An error message is output to the console.

2. Error during connection to the application task

As soon as the virtual printer has created the application task, it attempts to connect itself to the application task. After it has tried unsuccessfully for one minute, a message is output to the console, the virtual printer is stopped.

3. Errors which are returned by the application during job processing

An error message is output to the console. The print job is set to the wait state.

6.3.5 Restarting after errors

1. Failure of the virtual printer

If the virtual printer fails, SPOOL's restart mechanism creates a new supervisor task which must produce the connection to the application task.

2. Problems with the application task

If the TU application fails, the application task terminates processing. In this case the inter-task communication mechanism informs the partner tasks of the end of the TU application, whereupon the virtual printer is terminated.

If the TU application no longer responds to the queries of the supervisor task and the virtual printer is waiting for a reply, the /STOP-PRINTER-OUTPUT command can be issued. This cancels the application task and stops the virtual printer as soon as a timeout of maximum 10 minutes is reached.

6.4 Example: Communication with a virtual printer in synchronous operation

```
***************************
     COPYRIGHT (C) SIEMENS AG 1998
                  ALL RIGHTS RESERVED
************************
*
  This application communicates with the virtual printer control task
  to request print jobs.
  It checks the parameters input in the /PRINT-DOCUMENT-STRING command
  and processes the print job according to its different values:
  "CANCEL", the print job is terminated.
  "KEEP", the print job is inserted into the jobs wait queue
     with the status KEEP.
  "TRANS", the print job is assigned the code translation table
     whose name follows in the STRING operand. A MODIFY-PRINT-JOB-
     ATTRIBUTES is executed for this. The parameter VIRTUAL is
     set to *NOT-ALLOWED to prevent the print job being subsequently
     assigned to the virtual printer again.
  If an error occurs during the modification, the job is added
  to the KEEP queue, otherwise it is set to the status WAIT.
        TITLE 'MACRO DEFINITION'
        MACRO
        SAVE2 &LABEL1
             R10.&LABEL1
        ST
        ST
              R14.&LABEL1+4
        I R
              R10.R15
        MEND
        MACRO
        EXIT &LABEL2
             R10,&LABEL2
              R14.&LABEL2+4
        @FXIT
        MEND
        TITLE 'VIRTUAL DEVICE APPLICATION'
VIRTUAL @ENTR TYP=M, TITLE=NO, ENV=SPLSPEC, LOCAL=WORKA
VIRTUAL AMODE ANY
VIRTUAL RMODE ANY
```

```
@DATA CLASS=B, BASE=R8, DSECT=CONSTANT
         SPACE 1
               R8.CONSTADD
*
*
         Initialize dialog between application and virtual printer
*
         @PASS NAME=VIRTINI
         SETUS ON=1
         Set application to ready
         @PASS NAME=VIRTRCV
*
         Terminate dialog between application and virtual printer
*
         @PASS NAME=VIRTEND
         SETUS OFF=1
         @FXIT
         @END
CONSTADD DC
               A(CONSTANT)
VIRTINI
         @ENTR TYP=L,TITLE=NO,BASE=R10,LOADSB=YES
         @DATA CLASS=B, BASE=R8, DSECT=CONSTANT
         @DATA CLASS=B, BASE=R13, DSECT=WORKA
         SAVE2 VIRTINIS
*
         Initialize dialog between application and virtual printer
*
         MVC APVINI(SPDI#), PLVINI
                                        INIT P/L
         SPVDINI MF=E, PARAM=APVINI
         @TF
              ΝE
         CLC
               SPDIMRET, = AL2(SPDIRCOK)
                                          SUCCESSFULLY
         @THFN
         @TF
               ΕQ
         CLC
               SPDIMRET, = AL2(SPDIRCER) DIALOG NOT POSSIBLE
         @THEN
         TERM
         @BEND
         @BEND
         Request OUTPUT AREA
*
         REQM 1,16,PARMOD=24
         ST
               R1, OUTAREAA
         MVC
               OUTAREAL, C4096
         @TF
               NΕ
         LTR
               R15,R15
         @THEN
```

```
TERM
         @BEND
         EXIT VIRTINIS
         @FND
VIRTEND @ENTR TYP=L,TITLE=NO,BASE=R10,LOADSB=YES
         @DATA CLASS=B, BASE=R8, DSECT=CONSTANT
         @DATA CLASS=B, BASE=R13, DSECT=WORKA
         SAVE2 VIRTENDS
*
         Terminate dialog between application and virtual printer
*
         MVC APVEND(SPVE#), PLVEND
                                        INIT P/I
         SPVDEND MF=E, PARAM=APVEND
         FXIT VIRTENDS
         @FND
VIRTRCV
         @ENTR TYP=L,TITLE=NO,BASE=R10,LOADSB=YES
         @DATA CLASS=B, BASE=R8, DSECT=CONSTANT
         @DATA CLASS=B.BASE=R13.DSECT=WORKA
         SAVE2 VIRTRCVS
*
*
         Ready to receive
         @CYCL
         L
               R6, OUTAREAA
         L
               R7, OUTAREAL
         MVC
               APDRCV(SPVR#), PLDRCV
         SPVDRCV MF=M,OUTADDR=(R6),OUTLGT=(R7)
         SPVDRCV MF=E, PARAM=APDRCV
         @CAS2 SPVRMRET, COMP=CLC
*
         CHECK STATUS
         @0F
               =AL2(SPVRRCOK)
         WORK RECEIVED
         @CAS2 SPVRDAT, COMP=CLI
         WHICH KIND OF WORK
         @0F
               SPVRJOB
         CONTROL TO VIRTUAL PROCESS
         @PASS NAME=VIRTDEV
         @PASS NAME=VIRTRET
         @0F
               SPVRST0P
*
         USE=NO OF THE VIRTUAL PRINTER
         EXIT VIRTRCVS
         @OFRE
         @BEND
         @0F
              =AL2(SPVRINIE)
```

```
INIT CALL EXPECTED
         @PASS NAME=VIRTINI
         @0F
              =AL2(SPVRREPX)
*
         RETRUN EXPECTED
         @PASS NAME=VIRTRET
             =AL2(SPVRRTRM)
         TERMINATION DONE
         EXIT VIRTRCVS
         @OFRF
         @BEND
         @BFND
         EXIT VIRTRCVS
         @END
VIRTRET @ENTR TYP=L,TITLE=NO,BASE=R10,LOADSB=YES
         @DATA CLASS=B, BASE=R8, DSECT=CONSTANT
         @DATA CLASS=B.BASE=R13.DSECT=WORKA
         SAVE2 VIRTRETS
         Return print job and result to virtual printer
         SPVDRET MF=E, PARAM=APDRET
         FXIT VIRTRETS
         @FND
VIRTDEV @ENTR TYP=L.TITLE=NO.BASE=R10.LOADSB=YES
         @DATA CLASS=B, BASE=R3, DSECT=SCBE
         @DATA CLASS=B, BASE=R6, DSECT=STRINGDS
         @DATA CLASS=B, BASE=R8, DSECT=CONSTANT
         @DATA CLASS=B.BASE=R13.DSECT=WORKA
         SAVE2 VIRTDEVS
*
         Process print job as requested
               R3, SPVROUTA
         L
         LH
               R6.SCBEVIDS
         LA
               R6.0(R6.R3)
         LR
               R2, R6
         MVC APDRET(SPVT#),PLDRET
                                       INIT P/L
         @CAS2 ACTION.COMP=CLC
*
         CANCEL THE JOB
         @0F
              CAN
         SPVDRET MF=M, PACTION=*CANCEL
         KEEP THE JOB
         @OF KEEP
```

```
SPVDRET MF=M, PACTION=*KEEP
         @0F
               TRANS
         USE THE PRIVATE TRANSLATION TABLE
         ΙA
                R4.APMOD
         LA
                R6, PLMOD
         ΙA
                R5,SMA#
         LA
                R7,SMA#
         MVCL
                R4, R6
         LR
                R6, R2
         LA
                R4.TRTNAME
         MV T
               WORKFILE, X'40'
         MVC
               WORKFILE+1(L'WORKFILE-1),WORKFILE
         MVC
               WORKFILE(L'TRTLIB), TRTLIB
         ΙA
                R5.WORKFILE
         LA
                R2.SCBETSN
         MODPJAT MF=M, TRANTAB=((R4), (R5)),
                  VIRTUAL=*NOT_ALLOWED, TSN=((R2))
         MODPJAT MF=E, PARAM=APMOD
         @TF
               EQ
         CLC
                SMARET, = A(SMAOK)
         @THFN
         MVC APDRET(SPVT#), PLDRET
                                         INIT P/L
         SPVDRET MF=M, PACTION=*WAIT
         @FLSF
         MVC APDRET(SPVT#), PLDRET
                                         INIT P/I
         SPVDRET MF=M, PACTION=*KEEP
         @BFND
         @OFRE
         @BFND
         EXIT VIRTDEVS
         @END
         TITLE ' STATIC PARAMETER LISTS OF MACROS'
CONSTANT DS
                0F
OUTARFAA DC
               A(0)
OUTAREAL DC
               A(0)
C4096
         DC
                F'4096'
ZER0
         DC
                F'0'
REQMRCWO DS
               F
RCWORK
         DS
               CL10
SPTAB
         DC
               C'0123456789ABCDEF'
CAN
         DC
               CL6'CANCEL'
KEEP
         DC
               CL6'KEEP
MODIFY
         DC
               CL6'MODIFY'
         DC
               CL6'TRANS'
TRANS
WORKFILE DS
               CL44
PLVINI
         SPVDINI MF=L
                                          STATIC P/L
         SPVDRCV MF=L
                                          STATIC P/L
PLDRCV
PLDRET
         SPVDRET MF=L
                                         STATIC P/L
PLVEND
         SPVDEND MF=L
```

```
PLMOD
         MODPJAT MF=L
TITLE 'VIRTUAL - AUTOMATIC WORK AREA (ADF)'
*
*
    Dummy procedure creates an ADF at the end of the module
XXXXXXX @ENTR TYP=I, ENV=SPLSPEC, TITLE=NO, LOCAL=WORKA
         @END
              LTORG=NO
VIRTUAL
         CSECT .
                              BE SURE MAIN CSECT BEFORE @
WORKA
         @PAR D=YES
         PRINT GEN
WORKABEG EQU
               2F
VIRTINIS DS
               2F
VIRTENDS DS
VIRTRCVS DS
               2F
VIRTRETS DS
               2F
VIRTDEVS DS
               2F
APVINI
         SPVDINI MF=C
         ORG
               APVINI
APDRCV
         SPVDRCV MF=C
         ORG
               APVINI
APDRET
         SPVDRET MF=C
         ORG
               APVINI
         SPVDEND MF=C
APVEND
         ORG
               APVINI
APMOD
         MODPJAT MF=C
         ORG
               APVINI
WORKALEN EQU
               *-WORKABEG
WORKA
         @PAR LEND=YES
         SSVSCB D
         SSVSCBD D
STRINGDS DSECT
LENGTHST DS
               С
PRGNAME
         DS
               CL8
STRING
         DS
               0CL32
         DS
ACTION
               CL6
PAR
         DS
               0CL26
TRTNAME
         DS
               CL8
TRTLIB
         DS
               CL18
         ORG
               PAR
```

REFER DS CL8 VALUE DS CL8 END ,

7 SPOOL exits

An exit is a point in the operating system or a TU (Task Unprivileged) program at which the current processing can be interrupted by calling an exit routine.

An exit routine is a component which is generally created by BS2000 systems support and added to the operating system. This exit routine allows a normally unchangeable operating system function to be adapted to special computer center operation requirements. Exit routines can add functions to the normal operating system or modify or replace existing functions. They appear as integrated components of the operating system.

Exit routines can be loaded and activated or deactivated during the system run, i.e. administered dynamically.

This chapter briefly describes the SPOOL exits 090, 091, 092, 093, 094, 095, 096 and 097. A description of these exits, apart from exit 093, can be found in the manual "System Exits" together with a listing of the DSECTS and a general introduction to the exits.

7.1 General information on SPOOL exits

Overview of exit points

The exits are identified internally by their exit number. The exit points and the time of their calls are shown in the following table.

No.	Exit	Type/time of call
090	SPOOL exit	Before each record is output to the printer
091 092	SPOOL header page SPOOL trailer page	Before the header page is output Before the trailer page is output
093	Filter exit	When a file is transferred from BS2000 to a UNIX system and vice versa
094	SPOOL resources routine	Conversion of client / server resources
095	SPOOL exit	Before the channel program output
096	SPOOL exit	Server selection
097	SPOOL exit	Controls the jobs for file transfer

Whether SPOOL outputs to a device are processed by SPOOL exit routines can be controlled separately for each output device (see the START-PRINTER-OUTPUT command).

Requesting the current version of SPOOL

The SPOOL (and RSO) exit routines must request the SPOOL version number which is stored in the SCB (SPOOL control block).

Exit routines are loaded, activated and deactivated independently of the DSSM commands START-SUBSYSTEM, STOP-SUBSYSTEM, HOLD-SUBSYSTEM and RESUME-SUBSYSTEM. It is therefore possible to unload SPOOL while the exit routines remain active; these can be called later from the same SPOOL version or a different, newly loaded SPOOL version.

The exit routines must be able to recognize any change in the SPOOL version. The easiest way to do this is to check the SCB version number.

The version number is located in the SCBESCB field of the DSECT that can be created using the SSVSCB macro.

Using the SPOOL exits

If exits 90, 91 and 92 are active simultaneously, the call order is as follows:

EXIT 90: first call

EXIT 91: routine for header page

EXIT 90: processing routine

EXIT 92: routine for trailer page

EXIT 90: final call

The initial call is completely processed before the file is opened. Then the header page is generated.

Completion of the trailer page is followed by the final call, and then the file is closed.

System exit 95 may have been activated independently in order to monitor channel programs that are sent to the printers.

When families are processed, a header page can be produced either for the first file only or for each member file of the family. By issuing an SPSINF call (INFO=SPSGEN) in the exit routine you can read the GEN record and thus obtain the corresponding information: if GENFAM=YES is set in the GEN entry, a header page is printed for each file of the family printout; otherwise a header page is printed solely for the first file (i.e. if the field SCBESEQ#=0 is set in the SPOOL control block).

The layout of the trailer page for local SPOOL is shown on page 591.

Exit 093 can be used to convert the contents of files which are either to be printed in a UNIX system based cluster or originate in a UNIX system based cluster and are to be printed on a BS2000 printer.

Exit 094 can be used by systems support to create a user routine for controlling the conversion of resources within a client/server environment in conjunction with the Distributed Print Services subsystem.

Exit 095 can be used by systems support to link a user routine into the Spool & Print Services which is activated for each SPOOL output.

Exit 096 can be used by systems support in conjunction with the Distributed Print Services subsystem to make a server selection by means of combining certain selection criteria.

Exit 097 can be used to even out the work load for job assignment to Dprint DFTM tasks. This exit provides a means of controlling the work flow between the different Dprint hosts.

Note

The following descriptions are based on the new command PRINT-DOCUMENT and new macro PRNTDOC introduced with SPOOL V3.0. The exit functions can be used in the same way with the old command PRINT-FILE and the old PRNT macro.

7.2 Spoolout exit (090)

This exit routine is called after a record of the file to be output is read but before the record is processed further by SPOOL in accordance with the specifications in the PRINT-DOCUMENT command.

The exit can be used for output to line printers or laser printers (page printers). It cannot be used for output to magnetic tape.

The following system actions can be initiated by the exit routine:

- accept the record that has been read in (modified or unmodified)
- replace the record that has been read in
- skip the record that has been read in
- skip the record that has been read in and all subsequent records up to the end of the file.

The following information is passed to the exit routine:

R0 = Exit number "090"

R1 = A(EX090 parameter list, see below)

R12 = A(TPR program manager)

R13 = A(save area)

R14 = A(indirect return)

R15 = A(exit routine)

The return codes are entered in the standard header.

Standard header:



A return code on the execution of EX090 is passed in the standard header.

aa=Maincode1; bb=Maincode2; cc=Subcode1;
dd=Subcode2

X'aa'	Meaning
00	Accept read data record
04	Replace read data record with contents of output block
08	Skip read data record
0C	Skip read data record and all subsequent ones until end-of-file

SPOOL exits SPOOL exits

Notes

 The exit routine should be neither activated nor deactivated while a spoolout is being processed.

- The PRNT macro may be used in the exit routine, but not \$PRNT and \$SPRQ.
- Control is returned to the exit routine for every record read from the input file (including
 the sample printout of the first page) until the operator initiates printing. The same
 applies to RESPOOL processing (HOLD-SPOOLOUT-/RESUME-SPOOLOUT
 command), in which case the exit routine must also take account of the location of the
 start of RESPOOL.
- The information required for processing files with non-Siemens control characters can be found in the PRINT-FILE command and the SPOOL control block.
- If output blocks are provided, any memory class is permissible. However, it is advisable
 to select class 5 memory. The memory space is released by the exit as soon as the exit
 has regained control.

DSECT

A DSECT/CSECT for the parameter list (the address is transferred in register 1) can be generated by means of the EX090 macro. "prefix" must not be longer than three characters.

```
name EX090 D/C ,[prefix]
```

After the return has taken place, the input fields are not subjected to any more checks on the part of the calling SPOOL component. The first and last exit routine calls take place without a record from the file being made available (ICLTYP = X'02' or X'03'). The byte IPRTYP specifies the output device.

Values for the output fields are supplied by the exit routine. The byte IEXRC controls the function required by the exit routine. A more detailed analysis of the function follows the DSECT.

```
SPOOLEXO EXO90 D
1 *
1 *
      SPOOL INPUT RECORDS ROUTINE EXIT PARAMETER LIST
1 SPOOLEXO IDLKG ID=RC, P=, VER=851, SECT=D, ALIGN=F
                                                                     753
                 *, VERSION 851
2 SPOOLEXO DSECT
1 IBFG
          DS
                 OΗ
1
           FHDR MF=(C,I), EQUATES=NO
2
           DS
                 0A
2 IFHE
           DS
                 0XI8
                                      GENERAL PARAMETER AREA HEADER
2 *
2 IIFID
           DS
                 0.4
                                      INTERFACE IDENTIFIER
           DS
                 AL2
2 IFCTU
                                      FUNCTION UNIT NUMBER
```

```
2 *
                                       BIT 15
                                               HEADER FLAG BIT.
2 *
                                      MUST BE RESET UNTIL FURTHER NOTICE
2 *
                                       BIT 14-12 UNUSED. MUST BE RESET
2 *
                                      BIT 11-0 REAL FUNCTION UNIT NUMBER
2 IFCT
           DS
                 AL1
                                      FUNCTION NUMBER
                                  2
2 IFCTV
           DS
                 AI1
                                      FUNCTION INTERFACE VERSION NUMBER
2 *
           DS
                                      GENERAL RETURN CODE
2 IRET
                 0Α
                                  4
2 ISRET
           DS
                 0AL2
                                  4
                                      SUB RETURN CODE
2 ISR2
           DS
                                      SUB RETURN CODE 2
                 AL1
                                  4
2 ISR1
           DS
                 AI1
                                  5
                                      SUB RETURN CODE 1
2 IMRET
           DS
                                      MAIN RETURN CODE
                 0AL2
                                  6
2 IMR2
           DS
                 AL1
                                      MAIN RETURN CODE 2
2 IMR1
           DS
                 AI1
                                  7
                                      MAIN RETURN CODE 1
           EQU
2 IFHI
                                      GENERAL OPERAND LIST HEADER LENGTH
                 8
2 *
1 *
    EQUATES FOR EXIT RETURN INFORMATION
1 *
1 *
1 IACCPT
           EQU
                                            ACCEPT RECORD AS IT IS
                 0
1 IADD
           FOU
                 4
                                            ADD RECORD(S)
                 8
                                            DELETE (SKIP) RECORD
1 IDEL
           EQU
           EQU
                 12
                                            TERMINATE PRINT (SIMULATE EOF)
1 IEOF
      INPUT FIELDS FOR THE INPUT EXIT ROUTINE
1 *
1 IREC@
           DS
                 Α
                                            INPUT RECORD ADDRESS
1 ISCB@
           DS
                                            SCB ADDRESS
                 Α
                                            INPUT RECORD LENGTH
1 IRCLEN
           DS
                 Υ
1 IPRTYP
           DS
                 AL1
                                            TYPE OF PRINT
1 ILINES
           EQU
                  32
                                           LINES PRINTER TYPE
                                                                         751
                 33
                                           PAGE PRINTER TYPE
                                                                         751
1 IPAGES
           EQU
1 ILOADL
           EQU
                  34
                                           LOADABLE LINE PRINTER TYPE
                                                                         751
                 35
                                           RBP PRINTER TYPE
                                                                         751
1 IRBP
           EQU
                                                                         752
1 IRSOPB
           FOU
                  36
                                           PUBLIC RSO PRINTER
           FOU
                 37
                                                                         752
1 IRSOPR
                                           PRIVATE RSO PRINTER
1 IAPAPR
           FOU
                  38
                                           APA PRINTERS
1 IPUNCH
           FOU
                 48
                                           PUNCH TYPE
                                                                         751
1 ICLTYP
                                            CALL TYPE
           DS
                 AL1
1 ICLREC
           EQU
                  1
                                            RECORD CALL TYPE
1 ICLFST
           EQU
                  2
                                            NO RECORD FIRST TIME CALL
1 ICLLST
           EQU
                  3
                                            NO RECORD LAST TIME CALL
                 CL4
1
           DS
                                            RESERVED FOR EXTENSION
1 *
1 *
      OUTPUT FIELDS FROM THE INPUT EXIT ROUTINE
1 *
                                            DATA BLOCK ADDRESS
1 IBLK@
           DS
                 Α
                                            DATA BLOCK LENGTH
1 IBIKIF
           DS
                  Υ
                                            R-C FROM EXIT ROUTINE MAY BE
1 IEXRC
           DS
                 AL1
                                            STORED HERE IN ADDITION TO
1 *
1 *
                                            THE MAIN RETURN CODE
```

SPOOL exit 090 SPOOL exits

1	IBEXRC	DS	AL1	BASE R-C FROM EXIT ROUTINE	
1	*			MAY BE STORED HERE	
1	IBRCOK	EQU	0	BASE R-C O.K.	
1	INEXS	EQU	4	EXIT ROUTINE NOT EXISTENT	
1	INACT	EQU	8	EXIT ROUTINE NOT ACTIVE	
1	IPARER	EQU	12	PARAMETER ERROR	
1		DS	CL8	RESERVED FOR EXTENSION	
1	*				
1	* INPU	T FIEL	DS FOR THE INPUT E	EXIT ROUTINE	
1	*				
1	IPIR	DS	A	PRINTER INFORMATION RECORD ADDR.	755
1	IDVTYP	DS	AL1	DEVICE TYPE	755
1		DS	CL11	RESERVED FOR EXTENTION	755
1	TIFN	FOU	*-SP001 FX0	ROUTINE EXIT P/L LENGTH	

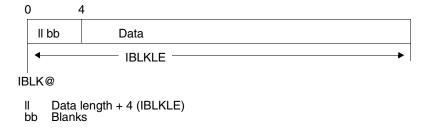
Exit routine functions

Return code X'00': accept the record that has been read

The address of the record and the record length are passed to the exit routine in the IREC@ or IRCLEN field. The record contains no record length field.

The exit routine can direct SPOOL to accept the record in unmodified form or it can modify it. If its length is not increased at this stage, the record can be modified directly in the input buffer. In this case, the fields IBLK@ and IBLKLE must be supplied with the address of the record from IREC@ and the correct record length. If the modification increases the length of the record, then the exit routine must make the record available in a special buffer, which is addressed via IBLK@.

Buffer format



Return code X'04': replace the record that has been read

SPOOL replaces the record that has been read in by the records provided by the exit routine in the output block (address in IBLK@):

II bb		Data 1				II bb		Data 2	
					Data 3				
	ll bb	bb Data 4						ll bb	
Data n-1 II bb					Data n				

The exit routine must supply values for fields IBLK@ and IBLKLE. If the record read is to be a component of the output block, the exit routine must transfer it to this block.

Return code X'08': skip the record that has been read

All the exit routine has to do is set the return code. No other action is necessary.

Return code X'OC': skip all records up to the end of the file

All the exit routine has to do is set the return code. No other action is necessary.

Operands in the PRINT-DOCUMENT command and PRNTDOC macro

The exit routine must take into consideration all operands specified in PRINT-FILE commands. The appropriate values can be derived from the SPOOL control block (see the appendix).

RECORD-PART=parameters(FIRST-CHARACTER = n):

All records inserted by the exit routine must be longer than n bytes, otherwise they will not be printed.

OUTPUT-PART = *RANGE(FROM = +n):

Records 1 to n are transferred to the exit routine.

OUTPUT-PART = *LAST(LAST = -n):

Initially, the exit routine receives control for all records from 1 to m (end-of-file, no output). Then SPOOL sets the FROM value to z = (m-n+1) and processes the file as for FROM = +z.

ADDITINAL-COPIES = n:

In the case of multiple output of a file, the exit routine cannot distinguish between the individual copies, i.e. there is no difference in the parameters or data records transferred.

SPOOL exits SPOOL exits

Example for exit 090

```
MACRO
               860
                        941109
                                32076095 SP00L
&L
        EX090 &D.&P
*******************
       COPYRIGHT (C) SIFMENS AG 1990
                    ALL RIGHTS RESERVED
LCLC &NAME
&NAME
        SETC '&P'
   SPOOL INPUT RECORDS ROUTINE EXIT PARAMETER LIST
        AIF ('&P' NE '').L5
&P
        SFTC
.L5
        ANOP
        ATF
             ('&P' NE '*').L6
&Р
        SETC
        ANOP
.L6
        AIF
             (&L NE '').LO1
&1
        SETC
             '&P.RC'
.L01
        ANOP
        MFCHK MF=&D, DMACID=E90, ALIGN=F, SUPPORT=(C,D), ENTRY=NO,
&1
             PREFIX=I, MACID=E90, DNAME=C
*
   VERSION 850 : ADAPTATION V3 : STANDARD HEADER FOR RETURN INFO.
   STANDARD HEADER RETURN INFORMATION
&P.BFG
        DS
             0H
        FHDR MF=(C, \&P), EQUATES=NO
   FOUATES FOR EXIT RETURN INFORMATION
&P.ACCPT EQU
             0
                                    ACCEPT RECORD AS IT IS
                                    ADD RECORD(S)
&P.ADD
      EQU
             4
&P.DEL
        EQU
                                    DELETE (SKIP) RECORD
             8
&P.EOF EQU
             12
                                    TERMINATE PRINT (SIMULATE EOF)
   INPUT FIELDS FOR THE INPUT EXIT ROUTINE
&P.REC@ DS
                                    INPUT RECORD ADDRESS
             Α
                                    SCB ADDRESS
&P.SCB@ DS
             Α
&P.RCLEN DS
                                    INPUT RECORD LENGTH
&P. PRTYP DS
             AI1
                                    TYPE OF PRINT
&P.LINES EQU
             32
                                   LINES PRINTER TYPE
                                                              751
&P.PAGES EQU
             33
                                   PAGE PRINTER TYPE
                                                              751
```

```
&P.LOADL EOU
               34
                                       LOADABLE LINE PRINTER TYPE
                                                                    751
&P.RBP EOU
               35
                                       RBP PRINTER TYPE
                                                                    751
&P.RSOPB EQU
               36
                                       PUBLIC RSO PRINTER
                                                                    752
&P.RSOPR EOU
               37
                                       PRIVATE RSO PRINTER
                                                                    752
&P.APAPR EOU
               38
                                       APA PRINTERS
&P.PUNCH EQU
              48
                                       PUNCH TYPE
                                                                    751
&P.CLTYP DS
              AL1
                                       CALL TYPE
&P.CLREC EOU
              1
                                        RECORD CALL TYPE
&P.CLFST EQU
               2
                                        NO RECORD FIRST TIME CALL
&P.CLLST EQU
               3
                                        NO RECORD LAST TIME CALL
        DS
              CL4
                                        RESERVED FOR EXTENSION
* OUTPUT FIELDS FROM THE INPUT EXIT ROUTINE
*
&P.BLK@ DS
                                        DATA BLOCK ADDRESS
              Α
                                        DATA BLOCK LENGTH
&P.BLKLE DS
              Υ
&P.FXRC DS
              AI1
                                       R-C FROM FXIT ROUTINE MAY BE
                                        STORED HERE IN ADDITION TO
                                        THE MAIN RETURN CODE
&P.BEXRC DS
              AI1
                                       BASE R-C FROM FXIT ROUTINE
                                       MAY BE STORED HERE
&P.BRCOK EQU
                                       BASE R-C O.K.
               0
&P.NEXS EQU
              4
                                       FXIT ROUTINE NOT FXISTENT
&P.NACT EQU
              8
                                        EXIT ROUTINE NOT ACTIVE
&P.PARER EQU
              12
                                        PARAMETER ERROR
         DS
              C18
                                        RESERVED FOR EXTENSION
* INPUT FIELDS FOR THE INPUT EXIT ROUTINE
&P.PIR
        DS
                                  PRINTER INFORMATION RECORD ADDR. 755
              Α
&P.DVTYP DS
                                  DEVICE TYPE
                                                                    755
              AI1
         DS
              CL11
                                  RESERVED FOR EXTENTION
                                                                    755
         AIF ('&L' EQ '').NOLAB
             *-&L
                                      ROUTINE FXIT P/I LENGTH
&P.IFN
         FOU
        MEXIT
        ANOP
.NOLAB
         ATF
               ('&NAME' NE '*').NOLAB1
&NAMF
        SETC
              1.1
.NOLAB1 ANOP
&P.IFN
         FOU
              *-&P.BEG
                                     ROUTINE EXIT P/I LENGTH
```

U41003-J-Z125-3-76 515

MEND

7.3 Exits for the header and trailer pages (091/092)

These system exits are used to control the layout of header/trailer pages for printer outputs by means of exit routines.

The exit routines can suppress the standard SPOOL system output, replace it by an alternative output or initiate an alternative output in addition to the standard output.

Alternative outputs for header or trailer pages are made available by system administration and are requested by the user with the aid of the HEADER-EXIT-NUMBER and TRAILER-EXIT-NUMBER operands in the PRINT-DOCUMENT command. The standard output can be modified by a text (HEADER-PAGE-TEXT operand in the PRINT-DOCUMENT command), independently of an exit routine.

Example

A standard name is to be defined,

HEADERPAGETEXT.n.

specifying a SAM/ISAM file or a job variable where you may store texts which can be referenced using the command

```
PRINT-DOCUMENT ..., LAYOUT-CONTROL=*PARAMETERS(COVER-PAGES= - *PARAMETERS(HEADER-EXIT-NUMBER=...))
```

The task of the exit routine is to access file

```
$userid.HEADERPAGETEXT.n
```

and write the text to the header page buffer. No selection is made until after the exit routine has been called. These exit routines save and restore all registers.

The routines are called:

- if no header/trailer page was produced
- in the case of an existing header/trailer page, to recreate the administrator buffer

Header page exit (091)

The following information is passed to the exit routine:

R0 = Exit number "090"

R1 = A(EX090 parameter list, see below)

R12 = A(TPR program manager)

R13 = A(save area) R14 = A(indirect return)

R15 = A(exit routine)

The return codes are entered in the standard header.

Standard header:



A return code on the execution of EX091 and EX092 is passed in the standard header. aa=Maincode1; bb=Maincode2; cc=Subcode1; dd=Subcode2

X'aa'	Meaning
00	No header or trailer page
01	Print the system header or trailer page without changes
02	Print the specified administrator header or trailer page
03	Print the system header or trailer page (without changes) and the specified administrator header or trailer page
04	Only for exit 92. No trailer page. The last feed to the "channel 1" vertical tab is suppressed

DSECT

A DSECT/CSECT for the parameter list (the address is transferred in register 1) can be generated by means of the EX090 macro. "prefix" must not be longer than three characters.

name EX091 D/C ,[prefix] SPOOLEX1 EX091 D 1 * 1 * SPOOL HEADER PAGE ROUTINE EXIT PARAMETER LIST 1 SPOOLEX1 IDLKG ID=HD, P=, VER=851, SECT=D, ALIGN=F 754 2 *.VERSION 851 2 SPOOLEX1 DSECT 1 IBEG DS 0Н 1 FHDR MF=(C,I), EQUATES=NO 2 DS 0A 2 IFHE DS 0XL8 GENERAL PARAMETER AREA HEADER 0 2 * 2 IIFID DS OΑ INTERFACE IDENTIFIER 2 IFCTU DS AI2 FUNCTION UNIT NUMBER 2 * HEADER FLAG BIT, 2 * MUST BE RESET UNTIL FURTHER NOTICE 2 * BIT 14-12 UNUSED, MUST BE RESET 2 * BIT 11-0 REAL FUNCTION UNIT NUMBER 2 IFCT DS AL1 FUNCTION NUMBER 2 IFCTV DS AL1 FUNCTION INTERFACE VERSION NUMBER 2 * 2 IRET DS 0A GENERAL RETURN CODE 2 ISRET DS 0AL2 4 SUB RETURN CODE 2 ISR2 DS AL1 SUB RETURN CODE 2 2 ISR1 DS AL1 5 SUB RETURN CODE 1 2 IMRET DS 0AL2 MAIN RETURN CODE 2 IMR2 DS AL1 6 MAIN RETURN CODE 2 2 IMR1 DS AL1 7 MAIN RETURN CODE 1 2 IFHL EQU GENERAL OPERAND LIST HEADER LENGTH 2 * 1 * 1 * EQUATES FOR EXIT RETURN INFORMATION IN &P.MR1 1 * 1 IEXNH EQU 0 NO HEADER PAGE OUTPUT 1 IEXSY EQU 1 SYSTEM HEADER UNALTERED 1 IEXAL FOU 2 ALTERNATIVE HEADER 1 IEXBO EQU 3 SYSTEM AND ALTERNATIVE HEADER INPUT FIELDS FOR THE INPUT EXIT ROUTINE 1 * 1 IPRTYP DS PRINTER TYPE X'20' LINES PRINTER TYPE 1 ILINES EQU 1 IPAGES EQU X'21' PAGE PRINTER TYPE 753

518

1	ILOADL	EQU	X'22'	LOADABLE LINE PRINTER TYPE	753
	IRBP	EQU	X'23'	RBP PRINTER TYPE	753
	IRSOPB	EQU	X '24'	PUBLIC RSO PRINTER	753
	IRSOPR	EQU	X'25'	PRIVATE RSO PRINTER	753
	TAPAPR	EQU	X'26'	APA PRINTER	, 00
_	IPUNCH	EQU	X,30,	PUNCH TYPE	753
	ICLTYP	DS	C	CALL TYPE	, 00
	INORM	EQU	X'01'	NORMAL CALL	754
	IRETRY	EQU	X'02'	RETRY	754
	INOBUF	EQU	X'04'	NO BUFFER	754
	ISMLHD	EQU	X.08.	SMALL HEADER	802
1		DS	CL2	RESERVED	
1	ISCB@	DS	A	SCB ADDRESS	
	ISYIB@	DS	A	SYSTEM BUFFER	
1	ILNPG	DS	F	# OF LINES PER PAGE	
1	ICHLN	DS	F	# OF CHAR. PER LINE	
1	*			" -	
1	* OUTPL	JT FIE	LDS FROM THE INPUT	EXIT ROUTINE	
1	*				
1	IUSOB@	DS	A	USER BUFFER ADDRESS	
1	IUSOBL	DS	F	USER BUFFER LENGTH	
	ISKIP#	DS	F	# SKIP RECORDS	
	ISYDF	DS	F	SYSTEM DUPL FACT	
1	IUSDF	DS	F	USER DUPL FACT	
1		DS	CL12	RESERVED FOR EXTENSION	
1	*				
1	* INPU	Γ FIEL	DS FOR THE INPUT EX	IT ROUTINE	
1	*				
1	IPIR	DS	A	PRINTER INFORMATION RECORD ADDR.	756
1	IDVTYP	DS	AL1	DEVICE TYPE	756
1		DS	CL11	RESERVED FOR EXTENTION	756
1	ILEN	EQU	*-SPOOLEX1	ROUTINE EXIT P/L LENG	TH

The following fields are set prior to calling the exit routine:

IPRTYP Output device

ILINES Line printer

IPAGES Laser printer / page printer ILOADL Printer with loadable VFB

IRSOPB RSO public printer IRSOPR RSO private printer

IAPAPR APA printer

ICLTYP Type of call

ISCB@

INORM Normal call

IRETRY Retry; required when the memory area for the alternative

header page has been destroyed or not set up correctly

INOBUF No system standard header page: no buffer address ISYIB@ is trans-

ferred; the exit routine can initiate an alternative output in accordance

with ILNPG and ICHLN (see below)

FSMLHD Short header page (after a job is aborted) Address of the SPOOL control block (SCB)

ISYIB@ Address of the system buffer

ILNPG Number of lines per header page

ICHLN Number of characters per header page line

The exit routine can supply values to the following fields:

IUSOB@ Class 3 memory address at which the alternative header page is available

IUSOBL Length of the buffer used for the alternative header page, corresponding to the buffer

area requested by the exit routine via \$GETMEM

ISKIP# Number of records in the alternative header page which are to be skipped, i.e. not

printed, in the event of a retry

ISYDF Iteration factor for output of the standard header page; default value: 1

IUSDF Iteration factor for output of the alternative header page; default value: 1

IPIR Address of the special printer information record.

For more detailed information see the relevant hardware description. (At present, only

LP65 printers are supported.)

IDVTYP Device type

For more detailed information see the description of the macro SPSINF (on page 189).

Trailer page exit (092)

The following information is passed to the exit routine:

```
R0 = Exit number "090"
```

R1 = A(EX090 parameter list, see below)

R12 = A(TPR program manager)

R13 = A(save area) R14 = A(indirect return) R15 = A(exit routine)

name EX092 D/C ,[prefix]

DSECT

2 IMRET

2 IMR2

2 IMR1 2 IFHL

2 * 1 * DS

DS

DS

FOU

0AL2

AI1

AL1

A DSECT or CSECT for the parameter list (address in R1) can be generated using the EX092 macro. "prefix" must not be longer than 3 characters.

```
SPOOLEX2 EX092 D
1 *
1 *
      SPOOL TRAILER PAGE ROUTINE EXIT PARAMETER LIST
1 *
                                                                     754
1 SPOOLEX2 IDLKG ID=TR, P=, VER=851, SECT=D, ALIGN=F
                 *.VERSION 851
2 SPOOLEX2 DSECT
1 IBEG
       DS
                 0H
           FHDR MF=(C,I), EQUATES=NO
2
           DS
2 IFHE
           DS
                 0XI8
                                      GENERAL PARAMETER AREA HEADER
2 *
2 IIFID
           DS
                 0Α
                                      INTERFACE IDENTIFIER
                                  0
2 IFCTU
           DS
                 AL2
                                      FUNCTION UNIT NUMBER
2 *
                                      BIT 15
                                                HEADER FLAG BIT.
2 *
                                      MUST BE RESET UNTIL FURTHER NOTICE
2 *
                                      BIT 14-12 UNUSED, MUST BE RESET
2 *
                                      BIT 11-0 REAL FUNCTION UNIT NUMBER
2 IFCT
           DS
                 AL1
                                      FUNCTION NUMBER
2 IFCTV
           DS
                 AI1
                                      FUNCTION INTERFACE VERSION NUMBER
2 *
2 IRET
           DS
                 0Α
                                  4
                                      GENERAL RETURN CODE
2 ISRET
           DS
                 0AL2
                                      SUB RETURN CODE
               AL1
2 ISR2
           DS
                                  4
                                      SUB RETURN CODE 2
2 ISR1
           DS
                 AL1
                                      SUB RETURN CODE 1
```

U41003-J-Z125-3-76 521

6

7

MAIN RETURN CODE

MAIN RETURN CODE 2

MAIN RETURN CODE 1

GENERAL OPERAND LIST HEADER LENGTH

1 * EQUA	ATES FO	OR EXIT RETURN INFO	RMATION IN &P.MR1	
1 IEXNT 1 IEXSY 1 IEXAL 1 IEXBO	EQU EQU EQU EQU JT FIEL	0 1 2 3 LDS FOR THE INPUT EX	NO TRAIL. PAGE OUTPUT SYSTEM TRAIL. UNALTERED ALTERNATIVE TRAILER PAGE SYSTEM AND ALTERNATIVE TRAI KIT ROUTINE	LER
1 IPRTYP	DS	С	PRINTER TYPE	
1 ILINES	EQU	X'20'	LINES PRINTER TYPE	
1 IPAGES	EQU	X'21'	PAGE PRINTER TYPE	753
1 ILOADL	EQU	X'22'	LOADABLE LINE PRINTER TYPE	753
1 IRBP	EQU	X'23'	RBP PRINTER TYPE	753
1 IRSOPB	EQU	X'24'	PUBLIC RSO PRINTER	753
1 IRSOPR	EQU	X'25'	PRIVATE RSO PRINTER	753
1 IAPAPR	EQU	X'26'	APA PRINTERS	750
1 IPUNCH	EQU	X,30,	PUNCH TYPE	753
1 ICLTYP 1 INORM	DS EQU	C X'01'	CALL TYPE NORMAL CALL	754
1 INORM 1 IRETRY	EQU	X.05.	RETRY	754
1 INOBUF	EQU	X'04'	NO BUFFER	754
1	DS	CL2	RESERVED	7 5 4
1 ISCB@	DS	A	SCB ADDRESS	
1 ISYIB@	DS	A	SYSTEM BUFFER	
1 ILNPG	DS	F	# OF LINES PER PAGE	
1 ICHLN	DS	F	# OF CHAR. PER LINE	
1 *				
1 * OUTF 1 *	PUT FIE	ELDS FROM THE INPUT	EXIT ROUTINE	
1 IUSOB@	DS	A	USER BUFFER ADDRESS	
1 IUSOBL	DS	F	USER BUFFER LENGTH # SKIP RECORDS SYSTEM DUPL FACT	
1 ISKIP#	DS	F	# SKIP RECORDS	
1 ISYDF	DS	F	SYSTEM DUPL FACT	
1 IUSDF	DS	F	USER DUPL FACT	
1	DS	CL12	RESERVED FOR EXTENSION	
1 *				
1 * INPU 1 *	JT FIEL	LDS FOR THE INPUT EX	KIT ROUTINE	
1 IPIR	DS	A	PRINTER INFORMATION RECORD ADDR.	756
1 IDVTYP	DS	AL1	DEVICE TYPE	756
1	DS	CL11	RESERVED FOR EXTENTION	756
1 ILEN	EQU	*-SP00LEX2	ROUTINE EXIT P/L LENG	HΤί

The following fields are set prior to calling the exit routine:

YDDXMAIN Indicator for the main case

YDDXAPPL main case' APP' (re. applications)

IPRTYP Output device

ILINES Line printer

IPAGES Laser printer / page printer

ILOADL Printer with loadable VFB IRSOPB RSO public printer

IRSOPR RSO private printer

IAPAPR APA printer

ICLTYP Type of call

INORM Normal call

IRETRY Retry; required when the memory area for the alternative

trailer page has been destroyed or not set up correctly

INOBUF No system standard trailer page: no buffer address ISYIB@ is trans-

ferred; the exit routine can initiate an alternative output in accordance

with ILNPG and ICHLN (see below)

FSMLHD Short trailer page (after a job is aborted)

ISCB@ Address of the SPOOL control block (SCB)

ISYIB@ Address of the system buffer ILNPG Number of lines per trailer page

ICHLN Number of characters per trailer page line

The exit routine can supply values to the following fields:

IUSOB@ Class 3 memory address at which the alternative trailer page is available

IUSOBL Length of the buffer used for the alternative trailer page, corresponding to the buffer

area requested by the exit routine via \$GETMEM

ISKIP# Number of records in the alternative trailer page which are to be skipped, i.e. not

printed, in the event of a retry

ISYDF Iteration factor for output of the standard trailer page; default value: 1

IUSDF Iteration factor for output of the alternative trailer page; default value: 1

IPIR Address of the special printer information record.

For more detailed information see the relevant hardware description. (At present, only

LP65 printers are supported.)

IDVTYP Device type

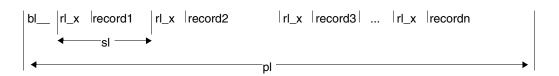
For more detailed information see the description of the SPOOL macro SPSINF

(page 189).

Structure of the buffer for header/trailer pages

Alternative output

In order to set up an alternative output, the exit routine has to reserve a buffer in class 3 memory. The header/trailer pages are made up of pages of variable length, i.e. prefixed by a 4-byte record length field. The first 4 bytes of the buffer contain, left-justified, the buffer length used (maximum length 32767 bytes = X'7FFF'.



bl Buffer length used

rl Record length / line length + 4

Blank (X' 40')

x Code character for type of output (only evaluated internally by SPOOL)

X' 0A' Printer

record for PRINT output: 1 byte print control character (see table) + text

Control character Meaning

X' 40' No feed before printing
X' 41' 1 line feed before printing
X' 42' 2 lines feed before printing
X' 43' 3 lines feed before printing

Other control characters may lead to undesired results, as they are not interpreted by every device type.

System header page

The format of the header page varies depending on the size of the paper and the character set used. The number of lines that can be output on one printed page is determined as follows:

```
Maximum number of lines = paper format in inches * lines / inches
```

Exception: in the case of laser printers, if the header page is output under LOOP control, the number of lines is preset by the VFB.

The lines of the header page are printed according to the order of priority described below (depending on the maximum number of lines).

Order of priority

```
Address and identification field (24 lines)

User text (10 lines) + 2 blank lines

Job name (10 lines) + 2 blank lines

User ID (10 lines) + 2 blank lines

Account number (10 lines) + 2 blank lines
```

If fewer than 24 lines are available, no standard header page is printed. However, the exit routine can take appropriate action.

Each printed line is stored as a variable-length record, i.e. with a prefixed record length field (4 bytes). Blank lines contain a blank (X'40') as the record contents.

Example: Format of a standard header page, contingent upon character and line density

Format (page size): 9 x 11 inches

Character set A: 15 characters/inch; 8 lines/inch (= 15 cpi / 8 lpi)

Character set B: 12 characters/inch; 6 lines/inch (= 12 cpi / 6 lpi)

	Character set A (15 cpi / 8 lpi)	Character set B (12 cpi / 6 lpi)
Number of characters per line	11 * 15 = 165	11 * 12 = 132
Number of lines per page	9 * 8 = 72	9 * 6 = 54
Header page format		
Address field (24 lines) Job name (10 + 2 lines) User ID (10 + 2 lines) Account number (10 + 2 lines)	Blank lines: 48 36 24 12	Blank lines: 30 18 6 -
Comments	The entire header page is printed out	Account number missing
length of asterisk lines	165 '*'	132 '*'

Paper feed

After the system trailer page has been printed, the next page is positioned to the first line. This also applies if the header/trailer page is not printed.

Notes on application

Effective use of the exits described above assumes a suitable degree of cooperation between system administration and the customers, e.g. a standard name for a SAM or ISAM file (or even a job variable). Both privileged and nonprivileged users can use this file to set up an alternative header page.

Example of the format of a standard name:

HEADERPAGETEXT.n, $0 \le n \le 2147.483.639$

The alternative header page can be printed using the PRINT-DOCUMENT command (operand HEADER-EXIT-NUMBER = n) or PRNTDOC macro (operand COVPAGE).

The exit routine must then access the file \$userid.HEADERPAGETEXT.n and provide the header page buffer with the alternative text.

Whether a header/trailer page is printed depends on the number of characters that can be printed per line.

If separate output blocks are provided by exits 091/092, class 3 memory is required. The memory space is released by SPOOL, not by the exits.

Example for exit 091

```
865
        MACRO
                                 960110
                                         32733508 SP00L
                                                              U
81
        EX091 &D.&P
********************
       COPYRIGHT (C) SIEMENS AG 1990
                    ALL RIGHTS RESERVED
********************
        LCLC
             &NAME
&NAME
        SETC '&P'
   SPOOL HEADER PAGE ROUTINE EXIT PARAMETER LIST
*
        ATF
             ('&P' NE '').L5
&Ρ
        SETC
             ' T '
.L5
        ANOP
             ('&P' NE '*').L6
        AIF
&Ρ
        SETC
.L6
        ANOP
        AIF
             (&L NE '').L01
&1
        SETC
             '&P.HD'
.101
        ANOP
٨1
        MFCHK MF=&D, DMACID=E91, ALIGN=F, SUPPORT=(C,D), ENTRY=NO,
             PREFIX=I, MACID=E91, DNAME=C
.*
   VERSION 850: ADAPTATION V3: STANDARD HEADER FOR RETURN INFO.
   STANDARD HEADER RETURN INFORMATION
*
&P.BEG
        DS
             0H
        FHDR MF=(C, &P), EQUATES=NO
   EQUATES FOR EXIT RETURN INFORMATION IN &P.MR1
&P.EXNH EQU
                                    NO HEADER PAGE OUTPUT
             0
&P.EXSY EQU
             1
                                    SYSTEM HEADER UNALTERED
             2
&P.EXAL EQU
                                    ALTERNATIVE HEADER
&P.EXBO EQU
             3
                                    SYSTEM AND ALTERNATIVE HEADER
   INPUT FIFIDS FOR THE INPUT FXIT ROUTINE
&P.PRTYP DS
             C
                                   PRINTER TYPE
```

```
&P.LINES EQU
               X'20'
                                      LINES PRINTER TYPE
&P.PAGES EOU
               X'21'
                                      PAGE PRINTER TYPE
                                                                     753
&P.LOADL EQU
               X'22'
                                      LOADABLE LINE PRINTER TYPE
                                                                     753
&P.RBP EOU
               X'23'
                                      RBP PRINTER TYPE
                                                                     753
&P.RSOPB EOU
               X'24'
                                      PUBLIC RSO PRINTER
                                                                     753
&P.RSOPR EQU
               X'25'
                                      PRIVATE RSO PRINTER
                                                                     753
&P.APAPR EQU
               X'26'
                                      APA PRINTER
&P.PUNCH EQU
               X'30'
                                      PUNCH TYPE
                                                                     753
&P.CLTYP DS
               С
                                      CALL TYPE
&P.NORM FOU
               X'01'
                                      NORMAL CALL
                                                                     754
&P.RETRY EOU
               X'02'
                                      RETRY
                                                                     754
               X'04'
                                      NO BUFFFR
                                                                     754
&P.NOBUF EQU
&P.SMLHD EQU
               X'08'
                                      SMALL HEADER
                                                                     802
         DS
               CL2
                                      RESERVED
&P.SCB@ DS
                                      SCB ADDRESS
               Α
&P.SYIB@ DS
                                      SYSTEM BUFFER
               Α
&P.INPG DS
                                      # OF LINES PER PAGE
               F
                                      # OF CHAR. PER LINE
&P.CHLN DS
  OUTPUT FIFIDS FROM THE INPUT EXIT ROUTINE
&P.USOB@ DS
                                      USER BUFFER ADDRESS
&P.USOBL DS
                                      USER BUFFER LENGTH
&P.SKIP# DS
               F
                                      # SKIP RECORDS
&P.SYDF DS
               F
                                      SYSTEM DUPL FACT
&P.USDF
         DS
                                      USER DUPL FACT
         DS
               CI 12
                                      RESERVED FOR EXTENSION
  INPUT FIFIDS FOR THE INPUT FXIT ROUTINE
&P.PIR
                                   PRINTER INFORMATION RECORD ADDR. 756
         DS
               Α
&P.DVTYP DS
               AI1
                                  DEVICE TYPE
                                                                    756
                                  HW ALIGNMENT
         DS
               XL1
                                  SEPARATOR PAGES FREQUENCE #
&P.FRFO# DS
                                                                    865
                                  FREO# IS INPUT / OUTPUT
                                                                    865
                                  ADDR OF 10 FREE FW FOR EXIT 091 865
&P.FWFR@ DS
               Α
                                   RESERVED - FREE FOR USE
                                                                    865
         DS
               ('&L' EO '').NOLAB
         ATF
&P.LEN
              *-&L
                                       ROUTINE EXIT P/L LENGTH
         EQU
         MEXIT
.NOLAB
         ANOP
         AIF
               ('&NAME' NE '*').NOLAB1
&NAMF
         SFTC
.NOLAB1 ANOP
&P.LEN
              *-&P.BEG
                                       ROUTINE EXIT P/L LENGTH
         EQU
         MEND
```

Example for exit 092

```
MACRO
                         860
                                   941109 32733509 SP00L
        EX092 &D.&P
&L
*********************
       COPYRIGHT (C) SIFMENS AG 1990
                     ALL RIGHTS RESERVED
        LCLC &NAME
&NAME
        SETC '&P'
   SPOOL TRAILER PAGE ROUTINE EXIT PARAMETER LIST
             ('&P' NE '').L5
        ATF
&Ρ
        SFTC
.L5
        ANOP
        ATF
              ('&P' NE '*').L6
&Р
        SETC
        ANOP
.L6
        AIF
              (&L NE '').LO1
&1
        SETC
              '&P.TR'
.L01
        ANOP
        MFCHK MF=&D,DMACID=E92,ALIGN=F,SUPPORT=(C,D),ENTRY=NO,
&1
              PREFIX=I, MACID=E92, DNAME=C
   VERSION 850 : ADAPTATION V3 : STANDARD HEADER FOR RETURN INFO.
   STANDARD HEADER RETURN INFORMATION
&P.BFG
        DS
              0Н
        FHDR MF=(C, \&P), EQUATES=NO
   FOUATES FOR EXIT RETURN INFORMATION IN &P.MR1
&P.EXNT EQU
              0
                                       NO TRAIL. PAGE OUTPUT
&P.EXSY EOU
                                       SYSTEM TRAIL. UNALTERED
              1
&P.EXAL EQU
              2
                                       ALTERNATIVE TRAILER PAGE
&P.EXBO EQU
              3
                                       SYSTEM AND ALTERNATIVE TRAILER
   INPUT FIELDS FOR THE INPUT EXIT ROUTINE
&P.PRTYP DS
              С
                                       PRINTER TYPE
&P.LINES EQU
                                     LINES PRINTER TYPE
              X'20'
&P.PAGES EQU
              X'21'
                                     PAGE PRINTER TYPE
                                                                  753
&P.IOADI FOU
              X'22'
                                     LOADABLE LINE PRINTER TYPE
                                                                  753
&P.RBP
       EQU
              X'23'
                                     RBP PRINTER TYPE
                                                                  753
&P.RSOPB EQU
              X'24'
                                     PUBLIC RSO PRINTER
                                                                  753
```

```
&P.RSOPR EOU X'25'
                                  PRIVATE RSO PRINTER
                                                               753
&P.APAPR EOU X'26'
                                   APA PRINTERS
&P.PUNCH EQU X'30'
                                   PUNCH TYPE
                                                               753
&P.CLTYP DS
            С
                                     CALL TYPE
&P.NORM EQU
            X'01'
                                  NORMAL CALL
                                                               754
&P.RETRY EQU X'02'
                                   RFTRY
                                                               754
&P.NOBUF EOU X'04'
                                   NO BUFFER
                                                               754
        DS
             CL2
                               RESERVED
&P.SCB@ DS
             Α
                                     SCB ADDRESS
&P.SYIB@ DS
            Α
                                     SYSTEM BUFFER
&P.LNPG DS
                               # OF LINES PER PAGE
&P.CHLN DS
            F
                                     # OF CHAR. PER LINE
* OUTPUT FIELDS FROM THE INPUT EXIT ROUTINE
&P.USOB@ DS
                                    USER BUFFER ADDRESS
&P.USOBL DS
                                     USER BUFFFR LENGTH
&P.SKIP# DS
            F
                               # SKIP RECORDS
&P.SYDF DS
            F
                               SYSTEM DUPL FACT
&P.USDE DS
                               USER DUPL FACT
        DS CL12
                                     RESERVED FOR EXTENSION
* INPUT FIFIDS FOR THE INPUT EXIT ROUTINE
&P.PIR DS
             Α
                               PRINTER INFORMATION RECORD ADDR. 756
&P.DVTYP DS
             AL1
                               DEVICE TYPE
                                                              756
        DS
             CL11
                               RESERVED FOR EXTENTION
                                                             756
        AIF ('&L' EQ '').NOLAB
&P.LEN EOU *-&L
                                    ROUTINE FXIT P/I LENGTH
        MEXIT
      ANOP
.NOLAB
       AIF ('&NAME' NE '*').NOLAB1
      SETC ''
&NAME
.NOLAB1 ANOP
&P.LEN EOU *-&P.BEG
                                    ROUTINE EXIT P/L LENGTH
```

530 U41003-J-Z125-3-76

MEND

SPOOL exits SPOOL exit 093

7.4 Exit for customer filters (093)

Exit 093 is an exit for Dprint operation (see the "Distributed Print Services (BS2000/OSD)" manual).

Customer filters can be defined with the exit 093 subsystem to implement customer wishes and to cope with different document formats which were generated by special applications or print servers. They are called customer filters because their construction is tailored to customer requirements.

Since customer filters have a higher priority than standard system filters, exit 093 is called by Dprint each time the filter mechanism is called (i.e. for the two different filter functions) in order to carry out any "customer" filtering that may be required. Standard document filtering is only carried out if

- the exit 093 subsystem is not loaded
- exit 093 requests this document filtering via its return code. This means that exit 093
 has not carried out any filtering and expects execution by Dprint.

The requirements, supported interfaces and restrictions for developing and using exit 093 are described in the following sections.

Exit 093 functionality

This exit routine is called each time document filtering is required to ensure print interoperability between BS2000 and SINIX/UNIX systems. The following actions are executed:

- When checking the print request:
 - Call the first filter function which checks the print job parameters.
 - Modify the parameters for print request acceptance.
 - Convert the document format originally specified in the print request into a format supported by the destination printer. For example, a BS2000 print request with DOCUMENT-FORMAT=*TEXT(...) to a SINIX SPOOL domain is modified by the standard filter into DOCUMENT-FORMAT=*SPECIAL-FORMAT(FORMAT-NAME=SIMPLE). The value SIMPLE can then be mapped to the '-ct' attribute of the SINIX SPOOL print request 'xpadd -iob'.

SPOOL exit 093 SPOOL exits

The \$DPFL parameter list, which is passed to exit 093, contains the following entries for these actions:

- filter action code (*MODIFY PRINT REQUEST)
- print job partner (*TO_GIP for BS2000→ UNIX systems and *FROM_GIP for UNIX systems → BS2000)
- address of the print job mapping (description of the specified print request operand values)
- CCSNAME field, which is filled by the filter for BS2000→ UNIX systems print interoperability and indicates whether the converted document is sent (with the second filter call) in EBCDIC-DF04 or ISO-8859-1 code.
- filter ID field in which a filter can enter its ID in order to check it with the second call.
- Once the print job has been accepted and before the document is transferred to the destination server:
 - convert the original document into a new format, depending on a few print request parameters of the original print job mapping (i.e. the mapping passed to the filter for executing the first action code).

The \$DPFL parameter list, which is passed to exit 093, contains the following entries for these actions:

- filter action code (*CONVERT-DOCUMENT)
- print job partner (*TO_GIP for BS2000→ UNIX systems and *FROM_GIP for UNIX systems→ BS2000)
- CCSNAME field, which is filled by the filter for BS2000→ UNIX systems print interoperability and indicates whether the converted document is sent (with the second filter call) in EBCDIC-DF04 or ISO-8859-1 code.
- filter ID field in which a filter can enter its ID in order to check it with the second call.
- address of the original print job mapping (description of the specified print request operand values before the filter for the first action is called)
- name of the converted file, which must be created by the filter and is sent back to the destination server as the filter return value.

The filter must therefore be implemented such that it can execute the two functions described above.

Using the registers

The following information is passed to exit routine 093:

R0 Exit number ,093'

R1 Address of the \$DPFL parameter list (see below)

R12 Address of the TPR program administrator

R13 Address of a save area (18 words)

R14 Address for indirect return jump

R15 Address of the exit routine

Return codes

The exit 093 return code to the calling system component is stored in the standard header of the \$DPFL parameter list.

(000)	001	Maincode	Magning
(SC2)	SC1	warncode	Meaning
	00	0000	Successful execution
00			Successful execution
01			Use standard system filter
	01	0001	Incorrect parameter
00			Unknown action code
01			Invalid print request address
02			Invalid output file name
03			Invalid protocol
	20	0002	Internal error
00			Filter error
01			\$GETMEM error
02			\$RETMEM error
03			DMS error
04			ILAM error
05			XHCS error
FF			Other error
00	40	0003	Filter rejects print request
00	81	0004	Not enough disk storage space for output file

SPOOL exit 093 SPOOL exits

\$DPFL parameter list

The \$DPFL interface is used to transfer the required information (e.g. output file name, print request mapping,...) to the called filter and to allow the filter to send return information to Dprint (e.g. modified print request mapping, return code,...).

The \$DPFL interface is available in assembler and C.

Interface name (ASS): \$DPFL
Interface name (C): \$DPFL.h

Interface title: DPRINTCL filter call interface

Interface area: DPRINTCL
Compilation area: Restricted

Interface type: CALL Function state: TPR

Interface purpose: Interface of interoperability filter

Interface comments: Only input/output parameter list (not MF=E)

Language: ASS, C

Prefix: D

Macid: PFL

Operands

Offset	Identification	Data type or value	Meaning
FW	\$DPFL_mdl	STRUCT:136	Parameter list layout
000	hdr	ESMFHDR:8	Standard header
800	eye_catcher	CHAR:8	Eye catcher '\$DPFL004'
010	action_code MODIFY_PRINT_REQUEST CONVERT_DOCUMENT	ENUM:1 0 255	Action code Modify the print request options Convert document
011	protocol FROM_GIP TO_GIP	ENUM:1 0 128	Protocol From GIP client To GIP server
012	reserved1	RES:2	Blank field
014	print_req	PTR:4	Print request address
018	output_file	CHAR:54	Name of output file
04E	ccs_name	CHAR:8	Coded Character Set
056	ID	CHAR:8	Filter ID
05E	reserved2	RES:42	Blank field

Data declarations for assembler

* Subcode 2 for SUCCESS DPFLSUCCESSFUL_FILTERING DPFLUSE_STANDARD_FILTER *	EQU	0	Successful filtering Use standard filter
* Subcode 2 for PARAMET	ΓER_ERI	ROR	
DPFLUNKNOWN_ACTION_CODE	EQU	0	Unknown action code
DPFLNULL_REQUEST_ADDRESS	EQU	1	Invalid print request address
DPFLINVALID_OUTPUT_FILE	EQU	2	Invalid output file name
DPFLINVALID_PROTOCOL	EQU	3	Invalid protocol
*			·
* Subcode 2 for INTERNA	AL_ERR	OR	
DPFLFILTER	EQU	0	Filter error
DPFLGETMEM	EQU	1	\$GETMEM error
DPFLRETMEM	EQU	2	\$RETMEM error
DPFLDMS	EQU	3	DMS error
DPFLILAM	EQU	4	ILAM error
DPFLXHCS	EQU	5	XHCS error
DPFLOTHER	EQU	255	Other error
*			

SPOOL exit 093 SPOOL exits

<pre>* Parameter list layout DPFL_MDL DPFLHDR FHDR MF=(C,DPFL)</pre>	DS ,EQUAT	OF ES=NO	Standard header
* Main-Returncodes DPFLSUCCESSFUL DPFLPARAMETER_ERROR DPFLINTERNAL_ERROR DPFLREQUEST_REJECTED DPFLNO_DISK_SPACE *	EQU EQU EQU EQU	0 1 2 3 4	Successful filtering Incorrect parameter Internal error Filter rejects print request Not enough disk storage space for output file
DPFLEYE_CATCHER DPFLACTION_CODE	DS DS	CL8 FL1	Eye catcher '\$DPFL004' Action code
* Supported action codes DPFLMODIFY_PRINT_REQUEST DPFLCONVERT_DOCUMENT * DPFLPROTOCOL	EQU EQU DS	0 255 FL1	Modify print request options Convert document Protocol
* Known protocols DPFLFROM_GIP DPFLTO_GIP EQU 128 *	EQU	0	From GIP client To GIP server
DPFLRESERVED1 DPFLPRINT_REQ DPFLOUTPUT_FILE DPFLCCS_NAME DPFLSIGNATURE DPFLRESERVED2	DS DS DS DS DS	XL2 A CL54 CL8 CL8 XL42	Blank field Print request address Name of output file Coded Character Set Filter ID Blank field

EQU *-DPFLHDR

DPFL#

SPOOL exit 093

Data declarations for C

```
/* Main return codes
                                                                             */
/* mc s
                                                                             */
#define sDPFLSUCCESSFUL
                                             /* Successful filtering
                                                                             */
                                   0
                                   1
                                             /* Incorrect parameter
#define sDPFLPARAMETER ERROR
                                                                             */
                                    2
                                              /* Internal error
                                                                             */
#define sDPFLINTERNAL ERROR
#define sDPFLREQUEST_REJECTED
                                   3
                                              /* Filter rejects print requ. */
#define sDPFLNO DISK SPACE
                                              /* Not enough disk storage
                                   4
                                                                             */
                                              /* space for output file
                                                                             */
/* Subcode 2 for SUCCESSFUL
                                                                             */
/* ENUM successful sc2 s
                                                                             */
#define sDPFLSUCCESSFUL_FILTERING
                                             /* Successful filtering
                                                                             */
#define sDPFLUSE_STANDARD_FILTER
                                    1
                                              /* Use standard filter
                                                                             */
                                                                             */
/* Subcode 2 for PARAMETER ERROR
/* ENUM parameter sc2 s
                                                                             */
#define sDPFLUNKNOWN ACTION CODE
                                              /* Unknown action code
                                                                             */
                                    0
#define sDPFLNULL REQUEST ADDRESS
                                              /* Invalid print request addr */
#define sDPFLINVALID OUTPUT FILE
                                              /* Invalid output file name
                                    2
                                                                             */
#define sDPFLINVALID_PROTOCOL
                                    3
                                              /* Invalid protocol
                                                                             */
/* Subcode 2 for INTERNAL_ERROR
                                                                             */
/* ENUM _internal_sc2_s
                                                                             */
#define sDPFLFILTER
                                    0
                                              /* Filter error
                                                                             */
#define sDPFLGETMEM
                                    1
                                              /* $GETMEM error
                                                                             */
#define sDPFLRETMEM
                                    2
                                              /* $RFTMFM error
                                                                             */
#define sDPFLDMS
                                    3
                                              /* DMS error
                                                                             */
#define sDPFLILAM
                                   4
                                              /* ILAM error
                                                                             */
#define sDPFLXHCS
                                    5
                                              /* XHCS error
                                                                             */
#define sDPFLOTHER
                                 255
                                              /* Other error
                                                                             */
/* Supported action codes
                                                                             */
/* ENUM action s
                                                                             */
#define sDPFLMODIFY_PRINT_REQUEST 0
                                             /* Modify print requ. options */
                                              /* Convert document
#define sDPFLCONVERT DOCUMENT
                                 255
                                                                             */
                                                                             */
/* Known protocols
/* ENUM _protocol_s
                                                                             */
#define sDPFLFROM GIP
                                   \cap
                                             /* From GIP client
                                                                             */
                                              /* To GIP server
                                                                             */
#define sDPFLTO GIP
                                 128
```

SPOOL exit 093 SPOOL exits

```
/* Parameter list layout
                                                                              */
struct sDPFL md1 {
        /*Standard header
        struct EMSFHDR hdr:
        char eye_catcher[8];
                                              /* Eye catcher '$DPFL004'
                                                                              */
        unsigned char action_code;
                                              /* Action code
                                                                              */
                                              /* Protocol
        unsigned char protocol;
                                                                              */
        char reserved1[2]:
                                              /* Blank field
                                                                              */
        void* print_req;
                                              /* Print request address
                                                                              */
        char output file[54];
                                              /* Output file name
                                                                              */
                                              /* Coded Character Set
        char ccs_name[8];
                                                                              */
                                               /* Filter ID
        char signature[8];
                                                                              */
                                               /* Blank field
        char reserved2[42]:
                                                                              */
}:
```

Administering exit 093 via DSSM

The subsystem corresponding to exit 093 must be loaded to ensure that any customer filters that are present are activated. Before this subsystem is loaded, it must be defined in the static subsystem catalog by systems support with SSCM. The name of the subsystem for running exit 093 is freely selectable (e.g. DPRINTFL). The subsystem can be administered dynamically with the START-SUBSYSTEM, HOLD-SUBSYSTEM, RESUME-SUBSYSTEM and STOP-SUBSYSTEM commands.

Further information on subsystem definition can be found in the "System Exits" manual.

Programming example for exit 093

Exit routine 093 is used to support customer filters. To allow more than one customer filter to be defined in exit 093 it is advisable to construct exit 093 in modules, e.g. comprising one "filter dispatch" base routine and several customer filter subroutines.

The "filter dispatcher" base routine can, for example, be used to check which customer filter is to be called. This depends on the document format attribute values of the print request and its internal tables containing the relevant entry names of the subroutines to be called.

Each customer filter subroutine should be configured such that it fulfills the dual functionality of the filter according to the entries in the \$DPFL parameter list (the corresponding function is initiated in the \$DPFL parameter list each time exit 093 is called):

- modify the print request and
- convert the document.

The following example gives an overview of implementing exit 093 in C. Using the C language requires implementing a small assembler to C module (called EXIT093 in the example) which is called by the exit interface \$SYSX. The user only needs to use the framework shown below and can extend it with the desired C functions. The HEADER file <msg7x.h> and the return transfer routine must be created by the user for returning error messages.

EXIT093 Assembler-to-C interface module

```
PRINT NOGEN
EXITO93 @ENTR TYP=E, ENV=SPLSPEC, RETURNS=NO, TITLE=NO, LOCAL=WADF
EXITO93 AMODE ANY
EXITO93 RMODE ANY
        ST R1.INPL@
                               Save $DPFL address
        LA R1,INPL@
L R15,=V(EXIT93C)
                                Fetch addresses of saved $DPFL address
        BASR R14,R15
                               Call C program EXIT93C
        @EXIT ,
                                Return to caller
        @FND
WADE
        @PAR D=YES
INPL@
        DS A
WADF
        @PAR LEND=YES
        FND
```

EXIT93C C function (called by EXIT093 module)

```
/*-----
/* #include AND LOCAL DEFINITIONS
/*------
#include <stdio.h>
#include <string.h>
#include <esmfhdr.h>
#include <prntdpc.h>
#include <prntdoc.h>
#include <sdpfl.h>
#include <msg7x.h>
.
.
. struct pl_mdl { struct sDPFL_mdl *dpfl;
```

```
};
/* INTERNAL CALL PROTOTYPES
/*----*/
static void to_gip_modify (struct pl_mdl *);
static void to_gip_convert (struct pl_mdl *);
static void dispatcher (struct pl_mdl *);
static vid send_reject (char, int);
/*-----*/
/* CODE PART
/* to_gip_modify
/* This routine executes the first call to the
                                         */
/* filter for the validity check and modifying */
/* BS2000 to UNIX systems print requests.
                                         */
void to_gip_modify (struct pl_mdl *pl)
 struct sDPFL mdl *dpfl = pl->dpfl;
 struct PRNTDOC_pl_mdl *prntdoc = dpfl->print_req;
 struct PRNTDPC_pl_mdl *prntdpc = prntdoc->dppar;
    Depending on the print request parameters,
    other functions (internal or external)
    are called to execute the appropriate
    checking and modification.
```

}

```
/* to_gip_convert
                                                           */
                                                           */
/*
                                                           */
/* This routine executes the second call to the
                                                           */
/* filter for processing the file and creating an
                                                           */
/* output file for BS2000 to UNIX sysrtems print requests. */
void to_gip_convert (struct pl_mdl *pl)
  struct sDPFL_mdl *dpfl = pl->dpfl;
  struct PRNTDOC_pl_mdl *prntdoc = dpfl->print_req;
  struct PRNTDPC_pl_mdl *prntdpc = prntdoc->dppar;
  /* _____
    Depending on the print request parameters.
    other functions (internal or external)
    are called to create the relevant output
    file.
/* Dispatching routine
                                                  */
/* Dispatch the print request, according to the */
/* protocol and action code and check for parameter */
/* errors.
                                                  */
void dispatcher (struct pl_mdl *pl)
  struct sDPFL_mdl *dpfl = pl->dpfl;
  if (dpfl->print reg == NULL) {
    dpfl->hdr.maincode = sDPFLPARAMETER ERROR;
    dpf1->hdr.subcode2 = sDPFLNULL_REQUEST_ADDRESS;
    send_reject (MSG7X_DEST_SYSOUT,NULL_REQUEST_ADDRESS);
    send_reject (MSG7X_DEST_SYSOUT,NULL_REQUEST_ADDRESS);
    return:
```

```
switch (dpfl->protocol) {
           sDPFLFROM_GIP:
   /* -----
      Dispatcher for the UNIX system to BS2000 print
      requests: the following example shows
      an exit 093 which does nothing other
      than use its return code to call the standard
      system filter supplied with DPRINTCL.
     dpf1->hdr.maincode = sDPFLSUCCESSFUL;
     dpf1->hdr.subcode2 =
              sDPFLUSE_STANDARD_FILTER;
     break:
          sDPFLTO GIP :
   case
      Dispatcher for the BS2000 to UNIX system print
      reguests: in the following example,
      the action code is checked and the
      corresponding functions are called.
     switch (dpfl->action_code) {
        case
               sDPFLMODIFY_PRINT_REQUEST:
          to_gip_modify (pl);
          break:
        case
               sDPFLCONVERT_DOCUMENT :
        to_gip_convert (pl);
        break:
      default:
        dpfl->hdr.maincode = sDPFLPARAMETER_ERROR;
        dpf1->hdr.subcode2 =
              sDPFLUNKNOWN ACTION CODE:
        send_reject (MSG7X_DEST_SYSOUT,INVALID_ACTION_CODE);
        break;
     break:
   default:
     dpfl->hdr.maincode = sDPFLPARAMETER_ERROR;
     dpfl->hdr.subcode2 = sDPFLINVALID_PROTOCOL;
     send_reject (MSG7X_DEST_SYSOUT,INVALID_PROTOCOL);
     break:
}
```

Security aspects

Exit routines run in privileged status, as do the calling system components. Great care must therefore be exercised when programming them. Particularly the use of TU macros or other programming errors can cause a task to abort and may even cause a system failure. Errors in exit routines are handled in the same way as system errors. Details on the rules and restrictions which must be adhered to can be found in the "System Exits" manual.

SPOOL exits SPOOL exits

7.5 Resource routine (094)

This SPOOL exit allows system administration to create a user routine that fulfills two functions in a client/server environment with the Distributed Print Services subsystem:

Conversion of client resources to server resources

This function modifies the values of some SCB fields (e.g. FORM, DIA). The input consists of the standard header, the address of the SCB and the function code F0.

The output consists of the return code in the standard header and the modified SCB.

Conversion of server resources to client resources

This function restores the original values of SCB fields (e.g. FORM, DIA). The input consists of the standard header, the address of the SCB and the function code F1.

The output consists of the return code in the standard header and the modified SCB.

Exit 094 must be started on the client and server machines. If it is not, the spoolout job is rejected with message SCP089.

The use of the exit routine with the function code F1 for restoring the original name in the SCB is optional. Only the return code must be set. However, the conversion performed by the server is also transparent in this case for the user on the client system. The output is modified appropriately for all SHOW-PRINT-JOB commands. The converted values are always displayed to the cluster administrator.

By default, the trailer page is printed with the values of the server. If the user wants to print his or her trailer page with the values of the client, he or she must create an appropriate trailer exit with an exit 094 call.

If an error occurs during processing of the function call F0 (return code not equal to 0 in the standard header), the spoolout job is rejected with error message SCP1090.

If an error occurs during processing of the function call F1 (return code not equal to 0 in the standard header), an error message is output at the operator terminal but the spoolout job is continued.

The following information is transferred to the exit routine:

R1 = A (EX094 parameter list)

The return codes are entered in the standard header.

Standard header:



A return code on the execution of EX094 is passed in the standard header. aa=Maincode1; bb=Maincode2; cc=Subcode1;

aa=Maincode1; bb=Maincode2; cc=Subcode1;
dd=Subcode2

X'aa'	Meaning
00	Exit executed without errors
01	Exit not executed without errors

DSECT

A DSECT/CSECT for the parameter list can be generated by means of the EX094 macro. "prefix" must not be longer than three characters.

name EX094 D/C ,[prefix]

1	SPOOLEX4 SPOOLEX4 E94EXBEG	DSECT	D , OH	RENAM	ING OF CLIENT RESOURCE EXIT P/L
1	LJ4LNDLU	FHDR	MF=(C,E94),EQU	JATES=I	NO
2		DS	0A		
2	E94FHE *	DS	0XL8	0	GENERAL PARAMETER AREA HEADER
2	E94IFID	DS	0A	0	INTERFACE IDENTIFIER
2	E94FCTU	DS	AL2	0	FUNCTION UNIT NUMBER
2	*				BIT 15 HEADER FLAG BIT,
2	*				MUST BE RESET UNTIL FURTHER NOTICE
2	*				BIT 14-12 UNUSED, MUST BE RESET
2	*				BIT 11-0 REAL FUNCTION UNIT NUMBER
2	E94FCT	DS	AL1	2	FUNCTION NUMBER
2	E94FCTV	DS	AL1	3	FUNCTION INTERFACE VERSION NUMBER
2	*				
2	E94RET	DS	0A	4	GENERAL RETURN CODE
2	E94SRET	DS	0AL2	4	SUB RETURN CODE
2	E94SR2	DS	AL1	4	SUB RETURN CODE 2
2	E94SR1	DS	AL1	5	SUB RETURN CODE 1
2	E94MRET	DS	0AL2	6	MAIN RETURN CODE
2	E94MR2	DS	AL1	6	MAIN RETURN CODE 2
2	E94MR1	DS	AL1	7	MAIN RETURN CODE 1
2	E94FHL	EQU	8	8	GENERAL OPERAND LIST HEADER LENGTH

SPOOL exit 094 SPOOL exits

```
2 *
1 *
1 *
      EQUATES FOR EXIT RETURN INFORMATION IN &P.RET
1 *
                                           EXIT NOK
1 E94EXNOK EQU
                 1
1 F94FX0K F0U
                 0
                                           FXIT OK
1 F94INFOR DC
                 X,00,
                                          INFORMATION GIVEN TO EXIT
1 E94CLTSV EQU
                 0.0,
                                          TRANS. CLIENT RESSOURCE TO SERV
1 E94SVTCL EQU
                 C'1'
                                          TRANS. SERVER RESSOURCE TO CLI.
1 E94UNUS DC
                 XL3,00,
                                                  UNUSED
                                          ADDRESS OF SCB
1 F94SCBAD DC
                 A(0)
           DS
                 CL20
                                          RESERVED FOR EXTENTION
1 E94PLLEN EQU
                                          LENGTH OF THE PARAMETER LIST
                 *-E94EXBEG
```

Example for exit 094

```
050
       MACRO
                             960829
                                            SPOOL
                                                       U
&NAME
       EX094 &D.&P
*******************
   BEGIN-INTERFACE
                 EX094
   TITLE
                  (/ ex094 /)
   NAME
                  EX094
   DOMAIN
                  SP00L
   LANGUAGE
                  ASS
   COPYRIGHT
                  (C) Siemens AG 1994
                     ALL RIGHTS RESERVED
   COMPILATION-SCOPE USER
   INTERFACE-TYPE
                 LAYOUT
   RUN-CONTEXT
                 TPR
   PURPOSE
                  (/ translation of resource (client or server) /)
   SYNTAX
                  (/ Syntax Variant 1:
                  &NAMF
                        EX094 &D.&P
                                    /)
   RFMARKS
                  (/ corresponding include : SSVEX094 /)
*******************
.*
   FND-INTERFACE
                  FX094.
***********************
       AIF
            ('&P' NE '').A0
&P
       SETC
            'E94'
.A0
       ANOP
       ATF
            ('&NAME' NE '').A1
```

```
&NAME
         SETC 'CLREXPL'
.A1
         ANOP
         ATF
              ('&D' EQ 'D').A2
&NAME
         DS
               0F
                             RENAMING OF CLIENT RESOURCE EXIT P/L
         AG0
               .A3
.A2
         ANOP
&NAME
         DSECT .
                             RENAMING OF CLIENT RESOURCE EXIT P/L
.A3
         ANOP
&P.EXBEG DS
               0Н
         FHDR MF=(C, &P), EQUATES=NO
*
  EQUATES FOR EXIT RETURN INFORMATION IN &P.RET
&P.EXNOK EQU
               1
                                        EXIT NOK
&P.EXOK EQU
                                        EXIT OK
               0
&P.INFOR DC
               X'00'
                                       INFORMATION GIVEN TO EXIT
&P.CLTSV EOU
              C'0'
                                       TRANS. CLIENT RESSOURCE TO SERV
&P.SVTCL EQU
              C'1'
                                       TRANS. SERVER RESSOURCE TO CLI.
&P.CALCL EQU
              C'2'
                                       TRANS. CALL ON CLIENT
&P.UNUS DC
              XL3'00'
                                       **** UNUSED ****
&P.SCBAD DC
                                       ADDRESS OF SCB
              A(0)
         DS
              CL20
                                       RESERVED FOR EXTENTION
&P.PLLEN EOU *-&P.EXBEG
                                       LENGTH OF THE PARAMETER LIST
         MEND
```

SPOOL exits SPOOL exits

7.6 Output routine (095)

This SPOOL exit allows system administration to generate a user routine which is invoked for every SPOOL output operation. Only in the case of error handling is no call made. The exit routine is called after a channel program is generated and before it is sent to the printer. The exit routine can be used for line and laser printers. APA printers, RSO, remote batch processing, magnetic tapes are not supported.

The following options exist for the exit routine:

- accept the channel program
- change the channel program
- suppress the channel program.

The following information is transferred to the exit routine:

R1 = A (EX095 parameter list)

The return codes are entered in the standard header.

Standard header:



A return code on the execution of EX090 is passed in the standard header.

aa=Maincode1; bb=Maincode2; cc=Subcode1;
dd=Subcode2

X'aa'	Meaning
00	Channel program output
04	Channel program not output
0C	Channel program not output and EOF

Notes

- The exit routine can access the SPOOL data structures. These data structures are version-specific.
- If the channel program is modified or is not output, the SPOOL account record is not updated.
- No code translation is performed on the data passed by the exit routine.

- If the channel program is modified, attention should be paid to the following rules:
 - The channel program and the data buffer must not be extended.
 - If the data addressed by a channel command is shortened, the data length specified for the channel command must also be corrected accordingly.
 - If the channel program is shortened, the "end-of-chain indicator" must also be corrected accordingly.
 - the data address can be modified but it must be an address within the SPOOL output range. The "start of output range" address is contained in the first channel command in the chain. The "end of output range" is calculated by adding the data length to the data address of the last channel command.

DSECT

A DSECT/CSECT for the parameter list can be generated by means of the EX095 macro. "prefix" must not be longer than three characters.

name EX095 D/C ,[prefix]

```
SPOOLEX5 EXO95 D
1 SPOOLEX5 DSECT .
                                          T/0
                                                 EXIT ROUTINE P/L
1 SPOEXBEG DS
                 0H
1
           FHDR MF=(C,SPO), EQUATES=NO
2
           DS
                 0A
2 SPOFHE
           DS
                 0XI8
                                      GENERAL PARAMETER AREA HEADER
2 *
2 SPOIFID DS
                 0A
                                      INTERFACE IDENTIFIER
2 SPOECTU DS
                 AL2
                                      FUNCTION UNIT NUMBER
2 *
                                                HEADER FLAG BIT,
2 *
                                      MUST BE RESET UNTIL FURTHER NOTICE
2 *
                                      BIT 14-12 UNUSED, MUST BE RESET
                                      BIT 11-0 REAL FUNCTION UNIT NUMBER
2 *
2 SPOFCT
           DS
                 AI1
                                      FUNCTION NUMBER
2 SPOFCTV
           DS
                 AL1
                                      FUNCTION INTERFACE VERSION NUMBER
2 *
2 SPORET
           DS
                 OΑ
                                 4
                                      GENERAL RETURN CODE
2 SPOSRET
           DS
                 0AL2
                                 4
                                      SUB RETURN CODE
2 SPOSR2
           DS
                 AI1
                                 4
                                      SUB RETURN CODE 2
2 SPOSR1
           DS
                 AI1
                                      SUB RETURN CODE 1
                 0AL2
2 SPOMRET
           DS
                                 6
                                     MAIN RETURN CODE
2 SPOMR2
           DS
                 AI1
                                     MAIN RETURN CODE 2
2 SPOMR1
           DS
                 AI1
                                 7
                                     MAIN RETURN CODE 1
2 SPOFHL
           FOU
                                      GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *
1 *
      FOUATES FOR EXIT RETURN INFORMATION IN &P.MR1
1 SPOFXIO
          FOU
                 0
                                           SEND I/O REQUESTED
1 SPOFXNI
          FOU
                 4
                                           DO NOT SEND I/O
```

SPOOL exit 095 SPOOL exits

1 SPOEXEO EQU	12	NO I/O DECLARE EOF	
1 SPOPRTYP DC	AL1(0)	PRINTER TYPE	
1 SPOLINES EQU	32	LINES PRINTER TYPE	751
1 SPOPAGES EQU	33	PAGE PRINTER TYPE	751
1 SPOLOADL EQU	34	LOADABLE LINE PRINTER TYPE	751
1 SPOINFOR DC	X,00,	INFORMATION GIVEN TO EXIT	
1 SPONOSCB EQU	X'01'	IND. SPOSCBAD NOT FILLED	
1 SPOUNUS DC	XL2'00'	**** UNUSED ****	
1 SPOCHADD DC	A(0)	CCW'S CHAIN OR BUFFER ADDRESS	
1 SPOSCBAD DC	A(0)	ADDRESS OF SCB	
1 SPODVTYP DS	AL1	DEVICE TYPE	
1 SPOPRTMN DS	CL2	PRINTER MNEMONIC	
1 DS	CL20	RESERVED FOR EXTENTION	
1 SPOPLLEN EQU	*-SPOEXBEG	LENGTH OF THE PARAMETER LIST	

The SPOCHADD field contains the address of the channel program. The channel command format is not dependent on the channel type or machine type.

ор	fl	count	address	
----	----	-------	---------	--

op = Operation code fl = Flag byte

count = Data length

address = Virtual data address

Note

The operation codes of the channel program are dependent on the channel type and printer type. They are not checked by SPOOL on reentry from the exit routine.

The &P.NOSCB field (see the parameter list) indicates that the &P.SCBAD field does not contain an SCB address and cannot therefore be used. This can occur if SPOOL carries out I/O operations that are not assigned to jobs.

The &P.DVTYP field contains the same detailed device type as used in the SPOOL parameter file. The SPSDTYP macro must be used to resolve it.

The &P.PRTMN field contains the mnemonic device name of the printer.

7.7 Exit for server selection (096)

This system exit allows SPOOL administration (TSOS or PRINT-SERVICE-ADM) to select a server by combining various selection criteria. A server can be selected in accordance with an individual decision table specifying, for example, that the jobs of user A that have paper format A are to be directed to server X, and the jobs of user A with paper format B are to be sent to server Y, and so on.

Exit 96 is not only called in a Dprint environment with a list of servers, but also (if necessary) for local SPOOL, whereby the server list contains just one element whose name consists of eight spaces.

Exit 096 must be started on the client computer. It is called during print job validation.

When changes are made to the server list, consistency must be maintained. The server selected by the SPOOL administrator for a job must be in the server list passed to the exit. To make sure this is the case, the SPOOL administrator can:

- place the desired server at the top of the list and retain the other servers
- place the desired server at the top of the list and set the number of available servers in the list to 1

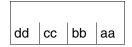
If a possibly distributed print job is to be redirected to local SPOOL, SERVER# must be set to zero in the exit routine call for the Dprint environment. The server list must then remain as it is (one element consisting of eight spaces) in the subsequent call for the local SPOOL, otherwise the print job is rejected completely.

The following information is passed to the exit routine:

R1 = A (EX096 parameter list)

The return codes are entered in the standard header.

Standard header:



A return code on the execution of EX090 is passed in the standard header.

aa=Maincode1; bb=Maincode2; cc=Subcode1;
dd=Subcode2

X'aa'	Meaning
00	Exit executed without errors
01	Exit not executed without errors

SPOOL exits SPOOL exits

DSECT

A DSECT/CSECT for the parameter list can be generated by means of the EX094 macro. "prefix" must not be longer than three characters.

name EX096 D/C ,[prefix]

```
SPOOLEX6 EXO96 D
1 SPOOLEX6 DSECT,
                               SERVER SELECTION EXIT P/L
1 E96EXBEG DS
                 0H
1
           FHDR MF=(C,E96),EQUATES=NO
2
           DS
                 0А
2 E96FHE
           DS
                 0XL8
                                     GENERAL PARAMETER AREA HEADER
2 *
2 E96IFID DS
                 OΑ
                                      INTERFACE IDENTIFIER
2 E96FCTU DS
                 AL2
                                     FUNCTION UNIT NUMBER
2 *
                                               HEADER FLAG BIT,
2 *
                                     MUST BE RESET UNTIL FURTHER NOTICE
2 *
                                     BIT 14-12 UNUSED, MUST BE RESET
2 *
                                     BIT 11-0 REAL FUNCTION UNIT NUMBER
                                     FUNCTION NUMBER
2 E96FCT
           DS
                 AL1
2 E96FCTV DS
                                     FUNCTION INTERFACE VERSION NUMBER
                 AL1
2 *
2 E96RET
           DS
                 0А
                                 4
                                     GENERAL RETURN CODE
2 E96SRET DS
               0AL2
                                 4
                                     SUB RETURN CODE
2 E96SR2
           DS
                 AL1
                                 4
                                     SUB RETURN CODE 2
              AL1
0AL2
2 E96SR1
          DS
                                     SUB RETURN CODE 1
2 E96MRET DS
                                 6
                                     MAIN RETURN CODE
               AL1
2 E96MR2 DS
                                     MAIN RETURN CODE 2
2 E96MR1
         DS
                 AL1
                                 7
                                     MAIN RETURN CODE 1
2 E96FHL
          EQU
                 8
                                     GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *
      EQUATES FOR EXIT RETURN INFORMATION IN &P.RET
1 *
1 *
1 E96EXNOK EQU
                 1
                                           EXIT NOK
1 E96EXOK EQU
                                           EXIT OK
1 E96SCBAD DC
                                         ADDRESS OF SCB
1 E96SVLAD DC
                 A(0)
                                         ADDRESS OF SERVER LIST
           DS
                 CL12
                                         RESERVED FOR EXTENTION
1 E96PLLEN EOU
                 *-E96EXBEG
                                         LENGTH OF THE PARAMETER LIST
1 E96SVL
           DSECT ,
                               SERVER LIST DESCRIPTION
1 E96SV#
           DS
                                         SERVERS#
1 E96SVSZ DS
                                          SIZE
1 E96SV
           DS
                                         SERVER(SERVERS#) (1 ENTRY)
                 0XL20
1 F96SVN
           DS
                 CL 8
                                         NAME
1 E96APR# DS
                 FL4
                                         PRINTERS#
1 F96DT
           DS
                 XI8
                                         DEVICE_TYPES
```

552

SPOOL exits SPOOL exit 097

7.8 Exit for controlling file transfer for Dprint requests (097)

Exit 097 is anchored in SPOOL. However, it is only relevant for a Dprint environment which is provided via the product Distributed Print Services. Exit 097 has the following tasks:

- to prevent server overloading by too many print jobs being in the file transfer phase simultaneously
- to prevent Dprint from using the FT resources on the client side exclusively for itself.

Exit 097 must be active on the server side to be able to fulfill these tasks. The exit regulates the number of print jobs transferred simultaneously to the server. A maximum number of print jobs which may be transferred simultaneously is defined for each computer in the cluster for this purpose. This sets the limit for the server to the sum of all client limits.

The maximum number of print jobs which may be transferred simultaneously from a particular client computer to the server is the sum of all limits which are defined on the servers in the cluster concerned for this client computer.

All these limits can be viewed and set with the SHOW-DPRINT-FT-LIMIT, MODIFY-DPRINT-FT-LIMIT, ADD-DPRINT-FT-LIMIT and REMOVE-DPRINT-FT-LIMIT commands which are provided for this purpose.

Background

If a Dprint job from a client is accepted by a remote server, it is placed into the SPOOL job queue of the server with the status WFT (wait file transfer). When requested by the Dprint FT unit, SPOOL selects a job from the queue and passes it to Dprint. The job selected is the one with the highest priority and whose original host (i.e. the client host which issued the request) is available and has resources free.

Once the Dprint FT unit has received the job, it informs the client host. This then initiates the transfer of the file to be printed and, if necessary, the file containing the resources for printing.

Since the Dprint FT unit requests print jobs continuously, all jobs accepted by the server which require FT are forwarded to FT immediately. If the number of jobs is very high and the server is not powerful enough, this leads to performance problems on the server side and exclusive utilization of the file transfer resources on the client side.

Exit 097 is localized in SPOOL on the server side and intervenes where a job is selected for the Dprint FT unit.

The exit 097 subsystem must be started on the server computer. It can even be loaded if SPOOL or Dprint have not been started. In this state, the subsystem then provides only the interface commands. The regulation mechanism starts when Dprint is loaded on the server. It is initiated every 30 seconds by the Dprint FT unit which requests a new print job for processing.

SPOOL exit 097 SPOOL exits

Exit processing

The exit subsystem calculates the number of print jobs waiting for file transfer and the number of print jobs being currently transferred, for each computer in the cluster. The status output by the SHOW-PRINT-JOB-STATUS command thereby differs from the status used internally by the exit 097 subsystem. A print job is in the WFT status as soon as it is accepted by the server, up to the moment when the server is informed that the client has correctly initiated the file transfer (FT NCOPY). The status of the print job changes from WFT to FT at this point.

For the exit 097 subsystem, a print job waits for the file transfer from the moment of acceptance by the server until processing has been initiated on the server by Dprint Server FT. The print job for the subsystem is transferred at this point.

The calculation result is stored in an internal host table. Each computer entry is assigned a number which specifies the maximum number of print jobs which can be transferred simultaneously by the computer concerned (FT limit).

If, during exit processing, the subsystem encounters a print job from a computer which is not entered in the table, a new entry is generated automatically. In this case, the computer is assigned a default value for the FT limit. After loading the subsystem for exit 097, this default value is 10.

A new entry can be added explicitly with the commands described below. The value for the FT limit must also be specified in this case.

Exit processing prevents file transfer initialization for print jobs from computers which are already simultaneously transferring their maximum number of print jobs.

Usage notes

- In section "Support commands for exit 097" on page 573 there is a description of commands with which you can administer the internal host table of a Dprint server. These commands are defined in a syntax file which you receive together with the subsystem for exit 097.
- The internal host table only exists as long as the exit subsystem is loaded on the server computer. This means that the host table disappears when the exit subsystem is stopped.
 - As soon as the exit subsystem is restarted, the host table must be rebuilt. The table is either recreated automatically by the exit and all FT limit values are reset to their defaults, thereby deleting all individual FT limit values, or you can write a procedure containing an ADD-DPRINT-FT-LIMIT command with an FT limit value for each computer. The exit subsystem must first be loaded in this case, then your procedure must fill the host table with data. The Dprint server can only be started with individual FT limit values after this.

EXIT specification

The following information is passed to the exit routine:

R1 = address of the EX097 parameter list

The parameter list contains the following fields:

E97HSTL@ = host list address

Address of a table containing the names of the available and non-saturated hosts

E97HST# = number of entries in the host table

E97HSTL = length of an entry in the host table

E97SVT@ = address of the SPOOL vector table

The following information is supplied by the exit routine:

Return code: The return code of exit 097 is stored in the standard header of the parameter list.

Standard header:



A return code on the execution of EX096 is passed in the standard header. aa=Maincode1; bb=Maincode2; cc=Subcode1;

dd=Subcode2

X'aa'	Meaning
00	Exit executed without errors
01	Exit not executed without errors

E97HOK# =	number of hosts remaining in the input host list
E97WFT# =	number of jobs with WFT status for the hosts remaining in the input host list
E97FT# =	number of jobs with FT status for the hosts remaining in the input host list
Input host list =	input list from which some entries have been removed (= filled with binary zeros)

SPOOL exit 097 SPOOL exits

DSECT

```
SPOOLEX7 EXO97 D
E97EXBEG DS
           OH
        FHDR MF=(C,E97),EQUATES=NO
* EQUATES FOR EXIT RETURN INFORMATION IN E97RET
E97EXNOK EOU
             1
                                       EXIT NOK
E97EXOK EQU
                                       EXIT OK
             XL4'00'
                                      **** UNUSED ****
E97UNUS DC
E97HSTL@ DS
              Α
                                      ACCESSIBLE HOST LIST
E97SVT@ DS
                                      SVT ADDRESS
              Α
E97HST# DS
              Н
                                      NBR OF ENTRIES IN THE LIST
E97HSTL DS
                                      LENGTH OF AN ENTRY IN THE LIST
                                      OUTPUT AREA = NBR OF CORRECT HST
E97HOK# DS
              Н
                                      TOTAL NBR OF JOB IN WFT
E97WFT# DS
              Н
                                      TOTAL NBR OF JOB IN FT
E97FT#
        DS
        DS
              CL20
                                      RESERVED FOR EXTENSION
E97PLLEN EQU *-E97EXBEG
                                     LENGTH OF THE PARAMETER LIST
        MEND
```

556

Explanation and example for exit 097

Exit 097 creates an internal host table and calculates the number of jobs with WFT and FT status for each host. A maximum value for simultaneous jobs in file transfer is set for each host. This maximum value is coded as a constant of 10 in the exit module. The value is the same for each host.

This value can be modified in the exit module with a REP. Example: for the maximum value 15, the following must be set in EXIT#097+X'210'

```
X'000F' (old code = X'000A').
```

The value can be modified during processing.

If a host from the internal host list reaches the maximum value and the host concerned is also in the input host list, the entry is removed from the input host list. The various return code counters are updated.

The following example shows the exit code for the processing described above:

```
********************
   COPYRIGHT (C) SIEMENS AG 1990
              ALL RIGHTS RESERVED
********************
 MODULE-DESCRIPTION
TITLE
            (/ EXIT#097 ASC MODULE /)
            EXIT#097
NAME
DOMAIN
            SPOOL
LANGUAGE
            ASC
COPYRIGHT (C) SIEMENS AG 1993
              ALL RIGHTS RESERVED
MEMORY-CLASS 4
 PROG-CONVENTS (/ ASS columbus conventions /)
 PROCEDURE
            BLDHSTI
   REMARKS
              (/ build host list table /)
   FNTRY
              BI DHSTI
                         FXTFRNAI
 DATA
            DCHDESC
   REMARKS
             (/ chain descriptor /)
   ENTRY
              CHDESC
                          INTERNAL
```

```
DATA SSVSVT
    REMARKS
             (/ SPOOL table /)
    ENTRY
              SVT EXTERNAL
           EXTERNAL GC USED
 DATA
   REMARKS (/ none /)
   ENTRY
              EXTERNAL_GC_USED EXTERNAL
 END-MODULE-DESCRIPTION EXIT#097.
*******************
* LEVO: EXIT#097
          Ţ
* LEV1: CHKHST
          !
* LEV2: BLDHST
* LEV3: ADD
* LEV4: SEARCH LOCK UNLOCK
* LEV0:
                    NLKRES34
* LEV1: SDFANAL
               SHOW ADD
* LEV3:
                            MODIFY REMOVE
* LEV4:
                SEARCH LOCK UNLOCK
EXIT#097 START
EXIT#097 RMODE ANY
EXIT#097 AMODE ANY
##BAL OPSYN ##BAS
##BALR OPSYN ##BASR
      ENTRY EXIT#097
```

558

```
EXIT#097 @ENTR TYP=I,LOCAL=STMWA,TITLE=NO,ENV=SPLSPEC
        @DATA CLASS=B, BASE=R11, DSECT=CLREXPL
        ΙR
             R11.R1
        @PASS NAME=LOCK
        @PASS NAMF=CHKHST
        X C
            E97RET.E97RET
        @PASS NAME=UNLOCK
        @FXIT
        @END
        FJFCT
        ENTRY NLKRES34
NLKRES34 DS
             0 F
NLKRS34 @ENTR TYP=I,LOCAL=STMWA,TITLE=NO,ENV=SPLSPEC
*********************
* UNIT-BODY FXIT#097.NIKRFS34
* TITLE (/
                NLKRES34 - main routine for cmd interface
* /)
* PURPOSE
* This function is the main entry used for the support of the
* interface commands.
*
                                                    FND-PURPOSF*
*
 REALIZATION
                                                              *
   CONTROL (/
       call SDFANAL for SDF transfer area analysis
       call LOCK
       case cmd.internal name
       = $SPSDP1 : call SHOW
       = $SPSDP2 : call ADD
       = $SPSDP3 : call MODIFY
       = $SPSDP4 : call REMOVE
       call UNLOCK
 /)
                                                END-REALIZATION*
 PARAMETERS
   ΙN
                 (/ SDF transfer area address /)
         NONE
                 (/ none /)
   OUT
*
                                                 FND-PARAMETERS*
                                                              *
* END-BODY EXIT#067.NLKRES34.
************************
```

SPOOL exit 097 SPOOL exits

```
@PASS NAME=SDFANAL
*
        @PASS NAME=LOCK
        @CAS2 CMDNAMX
        @0F
              =CL8'$SPSDP1 '
        @PASS NAMF=SHOW
        @0F
              =CL8'$SPSDP2 '
        @PASS NAME=ADD
        @0F
              =CL8'$SPSDP3 '
        @PASS NAME=MODIFY
        @0F
              =CL8'$SPSDP4 '
        @PASS NAME=REMOVE
        @BEND
        @PASS_NAMF=UNLOCK
        @FXIT
        @FND
        EJECT
        TITLE 'WORK AREA'
        @PAR D=YES
STMWA
LEV1
        DS
              16F
LEV2
        DS
              16F
LFV3
        DS
              16F
IFV4
        DS
              16F
JOBSTAT
        DS
              Χ
              X'01'
JOBWET
        FOU
JOBFT.
        EQU
              X'02'
                                                                 300
SEARCHH
        DS
              CL8
        DS
              XL3
LIMDEC
NBRDFC
        DS
              XL5
PACKED
        DS
              D
WROUTPL
        WROUT MF=C, PARMOD=31
********************
* Parameter list created by SDFANAL routine
* Input for SEARCH, ADD, MODIFY and REMOVE routines
CMDPL
        DS
              0F
                      Transfer area result and pl for subroutines
CMDNAMX
        DS
              CL8
                      command internal name
        DS
CMDHOST
              CL8
                      host name operand
CMDL TM
        DS
              Н
                       limit value
***********************
* Output from SEARCH routine
SEARCHRC DS
              Χ
                      RC from SEARCH routine
```

```
not_found EQU X'01' - entry not found found - entry found
FNTRY@
        DS
                       Address returned by SEARCH routine
*******************
* Input for SENDMSG routine
MSG#
                              id of the msg sent to sysout
         DS
              Χ
already_defined EQU
                       X'00'
                       X'01'
not defined
                EQU
                       X'02'
table_full
                 EQU
                       X'03'
                 EQU
no_entry
*
DISPBUF
        DS
              0F
DISPHDR DS
              Н
DISPRES DS
               3X
              0.X
DISPAREA DS
DISPREC DS
               0 X
DISPNAME DS
              CL8
DISPTEXT DS
              CL3
DISPLIM DS
              CL3
DISPTXT2 DS
              CL8
DISPWFT# DS
              CI 4
              CL7
DISPTXT3 DS
DISPFT# DS
              CL4
DISPCR
         DS
               Χ
DISPLEN EQU
              *-DISPREC
         ORG
               DISPREC
DISPTXTO DS
              CL16
DISPGLIM DS
              CL3
DISPLEN1 EQU
              *-DISPREC
         ORG
         PRINT NOGEN, CODE
STMWA
         @PAR LEND=YES
         EJECT
CHKHST
         @ENTR TYP=L,BASE=R10,TITLE=N0
         @DATA CLASS=B, BASE=R11, DSECT=CLREXPL
         @DATA CLASS=B.BASE=R13.DSECT=STMWA
*
         STM
               R14,R12,LEV1
         LR
               R10,R15
         ХC
               E97RET, E97RET
               R2.E97SVT@
         @PASS NAME=BLDHSTL
         @DATA CLASS=B, BASE=R5, DSECT=HOSTREC
```

```
L
               R3,E97HSTL@
         LH
               R4.E97HST#
         XR
               R6, R6
         XR
               R7, R7
         XR
               R8, R8
*
        @CYCL ,
               R5,HOSTTBL
        LA
         MVC
                CMDHOST, 0(R3)
         @PASS NAME=SEARCH
         @IF
                EQ
         CLI
                SEARCHRC, found
         @THEN
         R5, ENTRY@
         @TF
                GΕ
         CLC
                HOSTFT#,HOSTLIM
         @THEN
         ХC
                0(8,R3),0(R3)
         @ELSE
         ΑН
                R6,HOSTWFT#
         ΑН
                R7.HOSTFT#
         ΙA
                R8,1(,R8)
         @BEND
         @BEND
                R3.E97HSTL
         ΑН
         BCTR R4.0
         @WHEN ZE
         LTR
                R4, R4
         @BREA
         @BEND
         STH
                R6,E97WFT#
                R7,E97FT#
         STH
         STH
                R8,E97HOK#
         LM
                R14,R12,LEV1
         @EXIT
         @END
         EJECT
BLDHSTL
         @ENTR TYP=L,BASE=R10,TITLE=N0
         @DATA CLASS=B, BASE=R2, DSECT=SVT
         @DATA CLASS=B, BASE=R3, DSECT=CHDESC
         @DATA CLASS=B,BASE=R4,DSECT=SSLOT
         @DATA CLASS=B, BASE=R5, DSECT=HOSTREC
         @DATA CLASS=B, BASE=R13, DSECT=STMWA
         STM
                R14,R12,LEV2
```

562

```
LR
                R10,R15
*
         DROP R15
         LA
                R15,HOSTTBL
         @DATA CLASS=B, BASE=R15, DSECT=HOSTREC
*
      Clear counters
         @CYCL ,
         ХC
                HOSTWFT#(4),HOSTWFT#
         LA
                R15,HOSTLGT(,R15)
         @WHEN EQ
         CLC
               HOSTNAME, = C'*END
         @BREA
         @BEND
         DROP R15
                R3, SVTUNS1+2
                R4, CHDTABA
         L
         @WHIL NZ
         LTR
                R4, R4
         @D0
         ХC
                JOBSTAT, JOBSTAT
         @IF
         TM
                SSLJBTY, SSLJBDP+SSLCPSV
         @THEN
         @TF
                EQ
         CLI
                SSLSTAT, SSLFT
         @THEN
         MVI
                JOBSTAT, JOBFT
         @ELSE
         @IF
                ΕQ
                SSLSTAT, SSLWFT
         CLI
         @THEN
         @TF
                F0
         CLI
                SSLFTST, SSLFTRQ
         @THEN
         MV T
                JOBSTAT, JOBFT
         @ELSE
         MVI
               JOBSTAT, JOBWFT
         @BEND
         @BEND
         @BEND
         @BEND
*
         @TF
                NΖ
         CLI
                JOBSTAT,0
         @THEN
         @TF
                ΕQ
         CLI
                SSLCLTT, SSLCLHO
         @THEN
```

```
MVC
                SEARCHH, SSLOHST
         @ELSE
         MVC
                SEARCHH, SSLIHST
         @BFND
         MVC
                CMDHOST, SEARCHH
         @PASS NAME=SEARCH
         @TF
                EQ
         CLI
                SEARCHRC, not_found
         @THEN
         @IF
                NΖ
                ENTRY@, ENTRY@
         OC.
         @THFN
         R5, ENTRY@
         MVC
                HOSTNAME, SEARCHH
         MVC
               HOSTLIM, FTLIMIT
         @BEND
         @ELSE
         L
                R5.ENTRY@
         @BEND
*
         @IF
                EQ
         CLI
                JOBSTAT, JOBWFT
         @THEN
         LH
                R6,HOSTWFT#
         ΙA
                R6,1(,R6)
         STH
                R6,HOSTWFT#
         @ELSE
         LH
                R6,HOSTFT#
         LA
                R6.1(.R6)
         STH
                R6,HOSTFT#
         @BEND
         @BFND
                R6, CHDPTRS
         ΙH
         L
                R4,4(R6,R4)
         @BEND
         LM
                R14,R12,LEV2
         @EXIT
         @END
         EJECT
SDFANAL
         @ENTR TYP=L,BASE=R10,TITLE=N0
         @DATA CLASS=B, BASE=R1, DSECT=C$CMDINT
         @DATA CLASS=B.BASE=R2.DSECT=CMDHEAD
         @DATA CLASS=B, BASE=R3, DSECT=CMDODES
         @DATA CLASS=B, BASE=R4, DSECT=CMDOVAL
         @DATA CLASS=B,BASE=R13,DSECT=STMWA
         STM
                R14,R12,LEV1
         LR
                R10,R15
```

```
DROP R15
        ХC
             CMDLRET(4).CMDLRET
                                       Set rc OK
        ХC
              CMDLCRC(2), CMDLCRC
                                       Set command rc OK
        MVC
                                       Return msg CMD0001
             CMDLCRCM(7),=C'CMD0001'
                                       Get SDF transfer area address
        ΙA
              R1, CMDLTA
        @DATA CLASS=B, BASE=R1, DSECT=CMDSDES
             R2, CMDMAINO
        LA
                                       Get main operand address
        LR
              R3,R2
        MVC
              CMDNAMX, CMDINTN
                             internal name of the command
***********
       Host name operand
*********
        @TF
             GF
        CLC
             CMDNRMO,=AL2(1)
        @THEN ,
        @TF
             ON
        TM
              CMDGSTAT, CMDOCC
        @THFN
        ICM
              R4,15,CMDOPTR
        MVC
             CMDHOST, CMDAVAL
                              move input host name
        @FLSF
        ХC
             CMDHOST, CMDHOST
        @BEND
        @BFND
**********
       Limit operand
*********
        @TF
        CLC
              CMDNRMO.=AL2(2)
        @THEN .
              R2, CMDHEADL(,R2)
        ΙA
        LR
              R3, R2
        @TF
              ON
        TM
              CMDGSTAT, CMDOCC
        @THEN
        ICM
              R4,15,CMDOPTR
        ХC
              CMDLIM, CMDLIM
        MVC
              CMDLIM+1(1), CMDAVAL move input limit
        @BEND
        @BEND
        ΙM
             R14,R12,LEV1
        @EXIT
        @END
        EJECT
SHOW
        @ENTR TYP=L,BASE=R10,TITLE=N0
        @DATA CLASS=B, BASE=R3, DSECT=HOSTREC
```

```
@DATA CLASS=B, BASE=R13, DSECT=STMWA
*
                R14.R12.LEV3
         STM
         LR
                R10,R15
         DROP R15
         XR
               R5.R5
*
   Display first line: 'DEFAULT LIMIT: @@@'
         MVC
                DISPTXTO, = C'DEFAULT LIMIT : '
         XR
                R1,R1
         LH
                R1, FTLIMIT
         CVD
                R1, PACKED
         UNPK
                LIMDEC(3), PACKED+6(2)
         O T
                LIMDEC+2,X'F0'
         MVC
               DISPGLIM, LIMDEC
         LA
                R4, DISPLEN1+5
         STH
                R4.DISPHDR
         ХC
                DISPRES, DISPRES
         MVC
                WROUTPL(CUWL@WRO), WROUTL
         LA
                R15, DISPBUF
         ST
                R15.CUWMSGW
         ΙA
                R15, ERR001
         ST
                R15, CUWERRW
         WROUT PARMOD=31,MF=(E,WROUTPL)
*
   Display one line per host in the table
*
     'hostname : @@@ WFT# : @@@ FT# : @@@'
         LA
                R3.HOSTTBL
         @CYCL .
         @WHEN EQ
         CLC
               HOSTNAME, = C'*END
         @BRFA
         @TF
               NΖ
         OC.
               HOSTNAME, HOSTNAME
         @THEN
         LA
                R5.1(.R5)
         MVC
                DISPNAME, HOSTNAME
         MVC
                DISPTEXT,=C':
         XR
                R1, R1
                R1,HOSTLIM
         ΙH
         CVD
                R1, PACKED
         UNPK LIMDEC(3), PACKED+6(2)
                LIMDEC+2, X'F0'
         ΟI
         MVC
                DISPLIM, LIMDEC
```

566

```
MVC
                DISPTXT2,=C' WFT# : '
*
         LH
                R1.HOSTWFT#
         CVD
                R1, PACKED
         UNPK
                NBRDEC(5), PACKED+5(3)
         OΤ
                NBRDEC+4, X'F0'
         MVC
                DISPWFT#(4), NBRDEC+1
         MVC
                DISPTXT3,=C' FT# : '
         LH
                R1.HOSTFT#
         CVD
                R1, PACKED
                NBRDEC(5), PACKED+5(3)
         UNPK
         OΤ
                NBRDEC+4, X'F0'
         MVC
                DISPFT#(4), NBRDEC+1
*
         MV T
                DISPCR, CHRNL
         LA
                R4, DISPLEN+5
         STH
                R4, DISPHDR
         ХC
                DISPRES, DISPRES
         MVC
                WROUTPL(CUWL@WRO), WROUTL
         LA
                R15, DISPBUF
         ST
                R15.CUWMSGW
         ΙA
                R15, ERR001
         ST
                R15, CUWERRW
         WROUT PARMOD=31,MF=(E,WROUTPL)
         @BEND
         LA
                R3,HOSTLGT(,R3)
         @BEND
* No entry in the table => send error message
         @IF
                ZΕ
         I TR
                R5.R5
         @THEN
         MV I
                MSG#, no_entry
         @PASS NAME=SENDMSG
         @BEND
ERR001
         DS
                0Н
                R14,R12,LEV3
         LM
         @EXIT
         @END
         EJECT
ADD
         @ENTR TYP=L,BASE=R10,TITLE=N0
         @DATA CLASS=B, BASE=R3, DSECT=HOSTREC
         @DATA CLASS=B,BASE=R13,DSECT=STMWA
*
         STM
                R14,R12,LEV3
         LR
                R10,R15
```

```
DROP R15
* Search host name in the table
* if found then send error message
* if not found and not space in the table then send error message
* if not found and free space then create entry in the table
         @PASS NAME=SEARCH
         @CAS2 SEARCHRC, COMP=CLI
         @0F
              (found)
               MSG#,already_defined
         MV T
         @PASS NAME=SENDMSG
         @0F
              (not_found)
         @TF
               ZΕ
         0C
               ENTRY@, ENTRY@
         @THFN
         MVI
               MSG#,table_full
         @PASS NAME=SENDMSG
         @ELSE
               R3.ENTRY@
         MVC
               HOSTNAME, CMDHOST
         MVC
               HOSTLIM, CMDLIM
         @BFND
         @BFND
         LM
               R14,R12,LEV3
         @FXIT
         @END
         EJECT
MODIFY
         @ENTR TYP=L,BASE=R10,TITLE=N0
         @DATA CLASS=B.BASE=R3.DSECT=HOSTREC
         @DATA CLASS=B, BASE=R13, DSECT=STMWA
*
         STM
               R14,R12,LEV3
               R10,R15
         I R
         DROP R15
* If hostname not given (*NONE in cmd)
* then update default limit and return
         @TF
              7F
         00
              CMDHOST, CMDHOST
         @THEN
         MVC
              FTLIMIT, CMDLIM
         ΙM
              R14,R12,LEV3
         @FXIT
         @BEND
* Search host name in the table
```

```
* if not found then send error message
* if found then modify the limit
         @PASS NAMF=SFARCH
         @CAS2 SEARCHRC, COMP=CLI
         @0F
             (not_found)
         MVI
               MSG#, not defined
         @PASS NAME=SENDMSG
         @0F
              (found)
               R3.ENTRY@
         L
         MVC
               HOSTLIM, CMDLIM
         @BFND
         LM
               R14,R12,LEV3
         @FXIT
         @FND
         FJFCT
RFMOVF
         @ENTR TYP=L,BASE=R10,TITLE=N0
         @DATA CLASS=B, BASE=R3, DSECT=HOSTREC
         @DATA CLASS=B.BASE=R13.DSECT=STMWA
               R14,R12,LEV3
         STM
         LR
               R10,R15
         DROP R15
* Search host name in the table
* if not found then send error message
* if found then remove the entry in the host table
         @PASS NAME=SEARCH
         @CAS2 SEARCHRC, COMP=CLI
         @0F
             (not_found)
               MSG#, not defined
         MVI
         @PASS NAME=SENDMSG
         @0F
             (found)
         R3.ENTRY@
         ХC
               HOSTREC(HOSTLGT), HOSTREC
         @BFND
         LM
               R14.R12.LEV3
         @EXIT
         @END
         EJECT
SEARCH
         @ENTR TYP=L,BASE=R10,TITLE=N0
         @DATA CLASS=B, BASE=R13, DSECT=STMWA
         @DATA CLASS=B, BASE=R3, DSECT=HOSTREC
*
         STM
               R14,R12,LEV4
         LR
               R10,R15
         DROP R15
```

```
* search input host name in the table
* if found return its address in ENTRY@
* if not found then return rc + first free entry in the table
                     return ENTRY@ null if no free entry
*
         LA
                R3,HOSTTBL
         XR
                R6.R6
         @CYCL .
         MVI
                SEARCHRC, not_found
         ХC
                ENTRY@, ENTRY@
         @WHEN EQ
         CLC
               HOSTNAME, = C'*END
         @BRFA
*
         ST
                R3, ENTRY@
         MVI
                SEARCHRC, found
         @WHEN EQ
         CLC
               HOSTNAME, CMDHOST
         @BREA
         MVI
                SEARCHRC, not_found
         @IF
                ZΕ
         LTR
                R6.R6
         @AND
               ΕQ
         OC
               HOSTNAME, HOSTNAME
         @THFN
         LR
                R6.R3
         @BFND
                R3, HOSTLGT(,R3)
         LA
         @BEND
*
         @TF
                EQ
         CLI
                SEARCHRC, not_found
         @AND NZ
         LTR
                R6,R6
         @THEN
         ST
                R6, ENTRY@
         @BEND
         LM
                R14, R12, LEV4
         @EXIT
         @END
         EJECT
LOCK
         @ENTR TYP=L,BASE=R10,TITLE=N0
         @DATA CLASS=B,BASE=R13,DSECT=STMWA
         STM
                R14, R12, LEV4
         LR
                R10,R15
```

```
DROP R15
                R3.1
         LA
                R8.10
         LA
         @CYCL (R8),
                R2.R2
         XR
         @WHEN EQ
         CS
                R2,R3,LOCKZ
         @BREA
         VPASS 1
         @BEND
                R14, R12, LEV4
         LM
         @EXIT
         @END
         EJECT
UNI OCK
         @ENTR TYP=L,BASE=R15,TITLE=N0
         @DATA CLASS=B, BASE=R13, DSECT=STMWA
         ХC
               LOCKZ, LOCKZ
         @EXIT
         @END
         EJECT
SENDMSG
         @ENTR TYP=L,BASE=R10,TITLE=N0
         @DATA CLASS=B,BASE=R13,DSECT=STMWA
*
         STM
                R14,R12,LEV4
         LR
                R10,R15
         DROP R15
         @CAS2 MSG#, COMP=CLI
         @0F
                (already_defined)
         WROUT MSG001, ERROUT, PARMOD=31
         @0F
                (not_defined)
         WROUT MSG002, ERROUT, PARMOD=31
         @0F
                (table_full)
         WROUT MSG003, ERROUT, PARMOD=31
                (no entry)
         @0F
         WROUT MSG004, ERROUT, PARMOD=31
         @BEND
ERROUT
         DS
                0H
                R14, R12, LEV4
         LM
         @EXIT
         @FND
         TITLE 'STATIC
                          AREA'
WROUTL
         WROUT MSG001, MSG002, MF=L, PARMOD=31
* OUTPUT MESSAGES
```

```
MSG001
         DC
               Y(MSG001L)
         DS
                3 X
         DC
                'ERROR: HOST NAME ALREADY DEFINED'
MSG001L
         FOU
                *-MSG001
MSG002
         DC
               Y(MSG002L)
         DS
         DC
                'ERROR: HOST NAME NOT DEFINED'
MSG002L
         EQU
                *-MSG002
MSG003
         DC
               Y(MSG003L)
         DS
                3 X
         DC
                'ERROR: HOST TABLE FULL'
MSG003L
         FOU
               *-MSG003
MSG004
         DC
               Y(MSG0041)
         DS
                3 X
         DC
                'NO HOST DEFINED IN THE TABLE'
MSG004L
         EQU
               *-MSG004
FTLIMIT
         DC
               Y(10)
LOCKZ
         DC
               A(0)
         DS
                0F
H0STTBI
         DS
                XL4096
HOSTEND
                '*END
         DC
         TITLE 'DSECT AREA'
HOSTREC
         DSFCT
HOSTNAME DS
               CL8
HOSTWFT# DS
               Н
HOSTFT# DS
               Н
HOSTLIM
         DS
               Н
HOSTRES
         DS
               Н
HOSTLGT
         EQU
               *-HOSTREC
CLREXPL
         EX097 D
         DCHDESC D
CHDESC
         SSVSVT D
         DSLOT D
         $CMDINT MF=D
         CMDSTRU2
         VTCSET CHR
*
         TITLE 'EXIT#097- ETPND AND CROSS REFERENCE'
         ETPND EXIT#097, COMPNR=00000000, PATCH=200, CSECT=YES.
                DOMAIN=SPOOL,
                VER=002, DATE=960426
         FND
```

7.9 Support commands for exit 097

The contents of the internal host table can be viewed and modified with the following commands.

SHOW-DPRINT-FT-LIMIT

Domain: SPOOL-PRINT-ADMINISTRATION, SPOOL-PRINT-SERVICES

User group: Privileged users

Privileges: TSOS, OPERATING, PRINT-SERVICE-ADMINISTRATION

This command displays the contents of the internal host table. In the first line, SHOW-DPRINT-FT-LIMIT shows the default FT limit value assigned to new computers which were added implicitly to the table.

After this, SHOW-DPRINT-FT-LIMIT outputs a line containing the following information for each entry in the table:

- the computer name
- the FT limit value assigned to this computer
- the number of print jobs waiting for file transfer
- the number of print jobs currently being transferred

Format

SHOW-DPRINT-FT-LIMIT

Notes

- The information in the table is updated each time exit 097 is executed, i.e. approximately every 30 seconds.
- The values differ from those which can be queried with the SHOW-PRINT-JOB-STATUS command. The reason for this is that the job status is rated according to security considerations (see "Exit processing" on page 554).

MODIFY-DPRINT-FT-LIMIT

Domain: SPOOL-PRINT-ADMINISTRATION, SPOOL-PRINT-SERVICES

User group: Privileged users

Privileges: TSOS, OPERATING, PRINT-SERVICE-ADMINISTRATION

This command can be used to modify the default or current FT limit value of a computer.

Format

MODIFY-DPRINT-FT-LIMIT

HOSTNAME = *NONE / <alphanum_1..8>

,LIMIT = <integer_0..255>

Description of the operands

HOSTNAME=*NONE / <alphanum_1..8>

This operand is used to define whether the specified FT limit value is to apply for just one computer or be the default value for all computers that have not been entered into the host table yet.

HOSTNAME=*NONE

The specified value for the FT limit applies as the default value for all computers that have not been entered into the table yet.

HOSTNAME=<alphanum_1..8>

The specified value for the FT limit applies immediately for the specified computer.

LIMIT=<integer_0..255>

The FT limit value is specified with this operand.

ADD-DPRINT-FT-LIMIT

Domain: SPOOL-PRINT-ADMINISTRATION, SPOOL-PRINT-SERVICES

User group: Privileged users

Privileges: TSOS, OPERATING, PRINT-SERVICE-ADMINISTRATION

A new computer is entered into the host table with this command. The FT limit value to apply for this computer is specified together with the computer name.

Format

ADD-DPRINT-FT-LIMIT

HOSTNAME = <alphanum_1..8>

,LIMIT = <integer_0..255>

Description of the operands

HOSTNAME=<alphanum_1..8>

The specified computer is entered into the host table together with the FT limit value specified under LIMIT.

LIMIT=<integer_0..255>

The FT limit value is specified with this operand.

REMOVE-DPRINT-FT-LIMIT

Domain: SPOOL-PRINT-ADMINISTRATION, SPOOL-PRINT-SERVICES

User group: Privileged users

Privileges: TSOS, OPERATING, PRINT-SERVICE-ADMINISTRATION

The specified computer can be removed from the host table with this command.

Format

REMOVE-DPRINT-FT-LIMIT

HOSTNAME = <alphanum_1..8>

Description of the operands

HOSTNAME=<alphanum_1..8>

The specified computer is removed from the host table.

8 Appendix

8.1 Macro syntax

Macro format

A macro format consists of two columns; the first column contains the macro name while the second contains the possible operands.

Macro name Operands
<macroname> <operand 1>
,<operand 2>

When a macro is called, the macro name must be separated from the first operand by at least one blank. Where several operands are specified together, they must be separated by commas.

The macro formats are represented with the aid of certain notational symbols (metacharacters), which are explained in the following table.

Macro syntax Appendix

"New" metasyntax used for the macros

Elements of the metasyntax

Representation	Meaning	Examples
UPPERCASE LETTERS	Uppercase letters are used for keywords or constants, which the user must enter exactly as they are shown. Keywords must begin with * in case both keywords and names of constants or variables can be specified.	DIB FORCED=*YES
Lowercase letters	Lowercase letters denote data types of the values or variables which the user can specify.	DIB = <var: pointer=""></var:>
<>	Angle brackets denote variables whose range of values is described by the data types.	<var: pointer=""></var:>
Underscoring	Underscoring denotes the default value of an operand. If an operand does not have a default value, another value must be specified for it.	FORCED = *NO / *YES
=	The equals sign connects an operand name with the associated operand values.	DATA = <var: pointer=""></var:>
/	A slash serves to separate alternative operand values.	FORCED = *NO / *YES
list-poss(n)	The entry "list-poss" signifies that a list of operand values can be given at this point. If (n) is present, it means that the list must not have more than n elements. A list of more than one element must be enclosed in parentheses.	FLAG=list-poss(3): *SLI / *SKIP / *DC Specification: FLAG=*SKIP FLAG=(*SLI,*DC)

An operand is assigned an operand value from a defined range of values by means of the equals sign.

This value range is determined by a data type. The following table contains the data types of the operand values.

Data types of the operand values

Data type	Character set	Special rules
c-string	EBCDIC character	Must be enclosed within single quotes
integer	[+-] 02147483647	Is a decimal number
var:	Introduces a variable specification. The colon is followed by the type of the variable (see table "Data types for variables" below)	<var:var-type></var:var-type>
reg:	Registers 015	Specification: (<reg:var-type>)</reg:var-type>

Suffixes for data types

Suffix	Meaning	
nm	With data type "integer", nm means an interval specification; n: minimum value m: maximum value	
	With data type "c-string", nm means a length specification in bytes; n: minimum length m: maximum length with n < m	
n	With data type "c-string", n means a length specification in bytes; n <i>must</i> be adhered to.	

The operand values can be entered directly as a character string or integer (see data types "c-string" and "integer"), or indirectly via a variable (see data type "var:"). The following table contains the possible data types for variables.

Data types for variables

Data type	Description	Definition in program
char:n	The variable is a string of n characters. If the length specification is omitted, n=1 is assumed.	CLn
int:n	The variable is an integer occupying n bytes. If the length specification is omitted, n=1 is assumed. Condition: $n \le 4$	FLn
enum-of E:n	The variable is the list E, which occupies n bytes. If the length specification is omitted, $n=1$ is assumed. ($n \le 4$)	XLn
pointer	The variable is an address or an address value.	A

"Old" metasyntax used for the macros

The macro formats are represented with the aid of certain notational symbols (known as metacharacters) and conventions, which are explained in the following table:

Formal representation	Meaning	Example
UPPERCASE	Uppercase letters are used for constants, which the user must enter exactly as they are shown.	FSTAT ,LIST=(SYSLST) Enter: FSTAT ,LIST=(SYSLST)
lowercase	Lowercase letters are used for variables, which must be replaced by the user with the actual values, i.e. their contents will differ from case to case.	FILE filename Enter: FILE ACCOUNTING FILE XYZ, FILE A.B-1, etc.
{ }	Braces enclose alternatives; one of the possible values shown within the braces must be entered	FILE=pathname LINK=name Enter: FILE= or LINK=

Metasyntax (part 1 of 2)

Formal representation	Meaning	Example
[]	Square brackets enclose optional entries, i.e entries which may be omitted. If the comma is inside the square brackets, it is needed only if this optional entry is used and may be omitted for the first operand in a command. If, in contrast, the comma is outside the brackets, it must always be entered, even if no optional entry is specified. Note that normal (round) parentheses must always be entered.	F[REE]SIZE Enter: FREESIZE or the shortened form FSIZE
	Underscoring shows the default value, i.e. the value used by the system if the user omits the operand.	{ ISAM
	The dots indicate repetition, i.e. the preceding syntactical unit may be repeated several times.	(vsn,) Enter: (PVT003) or (PVT003,PVT456) or (XY00AB,XY0012,XY0005) etc.
_	This symbol represents a blank or space character (X'40').	STD_ Enter: 'STD'

Metasyntax (part 2 of 2)

Macro syntax Appendix

Macro types

Macros are divided into types on the basis of the manner in which their operands are passed. The various types are type 0, type R (where the operands are passed in registers) and type S (where the operands are passed in memory; S = storage).

0-type macros

Macros which are neither type R nor type S are known as 0-type macros.

Examples of this type are macros in which a register (often only R1) containing the start address of the operand list is specified in the operand field.

The operand list is defined in the data section of the program (using DC statements) and contains the operand values.

R-type macros

Operation	Operands
RTYP	<pre>foperand1</pre>

A macro belongs to type R if all necessary operand values can be loaded into the two registers (0 and 1) used for this purpose. An R-type macro does not generate an operand list.

The operands may be specified directly or placed in registers 0 and 1.

Address operands in R-type macros may be specified as explicit or implicit addresses.

S-type macros

Name	Operation	Operands
[opaddr]	macro	$ \begin{cases} S \\ L \\ C[,PREFIX=p][,MACID=mac] \\ (C,p) \\ D[,PREFIX=p] \\ (D,p) \\ M[,PREFIX=p][,MACID=mac] \end{cases} \} $
		$ MF = \begin{cases} E \\ (E = opraddr) \\ (E,(r)) \end{cases}], PARAM = addr] $

For S-type macros, the operand values specified in the macro are passed to the function module in the form of a data area which is part of the macro expansion. This is a suitably structured area which contains the data and memory definitions (DC and DS statements) necessary for passing the operand values.

The following applies to all macros that can be called with a specific macro version (e.g. via the VERSION or PARMOD operand): the version operand must have the same value in all calls with different values for the MF operand (MF=L/E/D/C).

With regard to MF, there are six different macro call forms: S form, E form, L form, D form, C form and M form.

```
S form = standard form
```

MF=S is the default value. The instruction section is generated first, followed by the data area with the operand values specified in the macro call. This data area contains no field names and no explanatory equates. The standard header is initialized.

E form = execute form

	Operation	Operands	
[label]	macro	MF=E,PARAM=	addr (r) (1)

The E form of the MF operand initiates a supervisor call (SVC): the contents of an operand list (see L form) are evaluated and the corresponding operations are executed. For this reason, the "execute" macro call must include the address of the operand list, either as a symbolic address (addr) or in a register (r/1). No other operands are evaluated.

"label" is the symbolic address of the macro in the Assembler program.

L form = LIST form

	Operation	Operands
label	macro	MF=L,operand-list

The list form uses the other operands specified in the macro to generate an operand list. This list does not contain symbolic addresses for the operands; these are generated by the C or D form of the macro. The address of the operand list must be specified in the macro (E form), which means that the symbolic address "label" must always be specified.

The operand list begins with the standard header (see page 586f), whose fields are loaded automatically when the list is created with MF=L. Even if an operand list is to be created dynamically with the D or C form, it must be initialized beforehand with MF=L in order to ensure that the header contains the correct information.

D form = DSECT form

	Operation	Operands	
[label]	macro	MF=D[,PREFIX=prefix]	

The D form generates a DSECT for the operand list of the macro. The first character of the generated names can be modified via the PREFIX operand. If a symbolic address is defined for the macro via "label", the DSECT receives this name. If "label" is not defined, the DSECT receives a macro-specific default name whose first character is likewise modified by PREFIX. The operand list should be initialized with the list form of the macro before the DSECT call in order to ensure that the standard header contains the correct information.

C form

	Operation	Operands
[label]	macro	MF=C[,PREFIX=prefix][,MACID=id]

Like the D form, the C form generates an operand list, but not in the form of a DSECT since no DSECT statement is generated. The operand list remains empty and should be initialized with the L form of the macro call in order to ensure, above all, that the standard header contains the correct information.

Using the PREFIX operand, the first character of the generated names can be changed, while the MACID operand can be used to change the second to fourth characters in these names (a string of up to three characters can be specified for MACID). If the macro is addressed symbolically via "label", this is also the address of the operand list; if "label" is not specified, the operand list cannot be addressed symbolically.

M form = modification form

	Operation	Operands
[label]	macro	MF=M[,PREFIX=prefix][,MACID=id],operand-list

Instructions (e.g. MVCs) are generated to overwrite the fields in an initialized data area (operand list) with the operand values specified in the macro call. The M form thus offers an easy way of dynamically matching the operand values with which a macro is called to the requirements of the program.

Since the instructions generated for this purpose use the addresses and equates of the C form or D form, care must be taken, when using the M form, that these names are available for addressing the operand list to be modified. In particular, care must be taken that, if specified, the operands PREFIX and MACID in a macro call with MF=M have the same values as in the associated MF=C or MF=D call.

Macro syntax Appendix

Standard header

All DMS macros use the standard header for BS2000 macros in their 31-bit interface. This standard header is an 8-byte field at the beginning of the operand list which contains the standardized designation of the interface and provides space for return codes. It is generated by each macro and should - wherever possible - be initialized with the list form of the MF operand.

Structure of the standard header

Field	Byte position	Meaning
UNIT	0-1	Specifies the function unit in which the desired function is implemented
FUNCTION	2	Specifies the function (within the function unit)
VERSION	3	Specifies the version number of the function
SUBCODE2	4	Contains subsidiary return code 2
SUBCODE1	5	Contains subsidiary return code 1
MAINCODE	6-7	Contains the main return code

Structure of the standard header

The fields SUBCODE2, SUBCODE1, MAINCODE contain the return code. The main return code indicates whether or not the operation was executed successfully. In the case of an error, the subsidiary return codes can be used to diagnose the reason for the error.

The following values for the return codes are conventions:

SUB- CODE2	SUB- CODE1	MAIN- CODE	Meaning
X'00'	X'00'	X'0000'	The function was executed successfully; there is no further information for this MAINCODE.
X'01'	X'00'	X'0000'	The function was executed successfully; no further action is necessary.
X'00'	X'01'	X'FFFF'	The requested function is not supported (invalid (entry for UNIT or FUNCTION in the standard header); unrecoverable error.
X'00'	X'02'	X'FFFF'	The specified function is not available; unrecoverable error.
X'00'	X'03'	X'FFFF'	The specified version of the interface is not supported (invalid version entry in the standard header); unrecoverable error.
X'00'	X'04'	X'FFFF'	The parameter list is not aligned on a word boundary.
X'00'	X'41'	X'FFFF'	The subsystem does not exist; it must be generated explicitly.
X'00'	X'42'	X'FFFF'	The calling process is not connected to this interface; it must be connected explicitly.
X'00'	X'81'	X'FFFF'	The subsystem is currently not available.
X'00'	X'82'	X'FFFF'	The subsystem is in the DELETE or HOLD state.

Return code conventions

MAINCODE shows the result of execution of the function. SUBCODE1 qualifies the main code. SUBCODE2 subdivides the errors into error classes.

The return code should be passed only in the standard header. However, for a transitional period the return code can also be passed in register R15 or in both the standard header and register R15. In order to check whether a return code has been passed in the standard header, the return code field should be filled with X'FFFFFFFF. The result of checking the standard header is also always returned in register R15:

X'00000000': standard header correctly initialized; error-free execution.

X'0001FFFF': invalid entry for UNIT or FUNCTION.

X'0003FFFF': invalid entry for VERSION.

Macro syntax Appendix

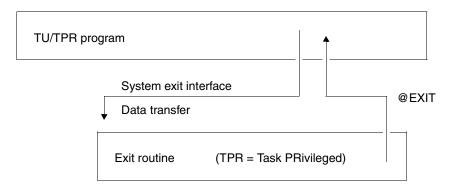
Wildcards

The user can employ wildcards for catalog ID and file names.

Wildcard	Meaning
*	Replaces a freely selectable character string, even an empty character string.
/	Replaces precisely one character.
<wildcard1,></wildcard1,>	Replaces all character strings which match at least one of the specified wildcards.
<wildcard1:wildcard2></wildcard1:wildcard2>	Replaces a character string which — is at least as long as the shortest specified wildcard; — is not longer than the longest specified wildcard; — lies in the alphabetical collating sequence between "wildcard1" and "wildcard2" (digits follow letters). "wildcard1" may also be an empty character string; this occupies the first position in the alphabetical collating sequence.
<wildcard1:wildcard2,></wildcard1:wildcard2,>	Wildcards of the type "wildcard1:wildcard2" may also be specified in the form of a list. The rules for each pair of wildcards are the same as described above. The system logically ORs the pairs, i.e. the wildcard list replaces any character string which matches one of the wildcard pairs. The length rule applies separately to each pair, not to all wildcards in the list.
-wildcard	Replaces any character string which does not match the specified wildcard. The minus sign may be specified only at the beginning of the wildcard string.

8.2 Functional principle of the exit mechanism

The functional principle of the exit mechanism is illustrated in the following diagram:



Calling exit routines

Via a TU or TPR macro interface, a module containing an exit point can branch to the corresponding exit routine, provided the routine exists and has been activated. Depending on what is required, this exit routine can check and (if necessary) modify the data, and it can reject or log a request. Control is then returned to the calling system module by means of the @EXIT macro in the program manager environment (DSL environment) or via BR R14 in pure Assembler routines. New exit routines should be developed in the DSL environment wherever possible.

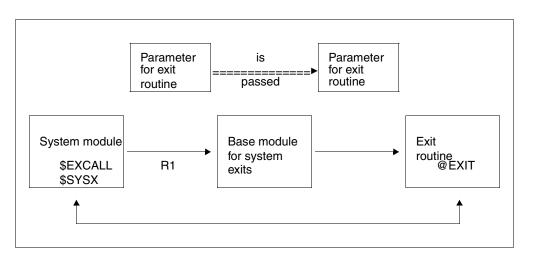
When control branches to an exit routine, certain TPR registers contain information. The register contents should be taken from the description of the relevant exit.

The contents of registers 12, 13 and 14 must not be destroyed by the exit routine; they must have been saved and loaded with the original contents before returning from the exit routine, i.e. they must have their original contents on returning.

Registers 0, 2 and 3 are now only defined for system exits which internally use an old system interface. For exits which have already been converted, registers 0, 2 and 3 are not defined. For details, see the descriptions of the exits.

The contents of the EXVT and the TCB must be accessed via the unbundled access functions.

As stipulated in the relevant exit description, the return code of the exit routine must be stored either in register 15 or in the specified parameter list with a standard header.



Exit mechanism

8.3 Trailer page for the local SPOOL

```
\mathsf{FIIF}: \mathsf{o} \mathsf{o
TYPF : @@@@@@@@
                                                           VFRS : @@@@@@@@@@@@@@@@@@@@@@@
                                                                                                        SPOOL TSN : ####
SPOOL DATE: ####-## HOST: @@@@@@@
START TIME : ##:##:##@ U-ID : @@@@@@@
                                                                                                        REO. TSN: ####
FND
              TIMF: ##:##:##@ DFV.: @@@@@@@@
                                                                                                    DFV. TYPF : @@@@@@@@
DIA
                    :@@ (@@) FAMILY:@@@
                                                                                         LOOP : (@@@,@@@) FORM :@@@@@@
STARTNO :####
                                            ENDNO :#####
                                                                                         FROM :#######TO
                                                                                                                                                 :###########
                  :###
                                            SPACE :@
                                                                                         BINARY:@@@
SHIFT
                                                                                                                                   CONTR:@@@@@@@@@@
DEL-FILE:@@@@@@ LINES :#####
                                                                                         COPIES:(###,###) PRIOR:@@@
HEADER :(@.@.@) HDRNUM:####### TRLNUM:#######CCPOS:#####
PNAME :@@@@@@@ TEXT
                                                           LOCK
                 : @@@
                                            DEST.: @@@@@@@@@@@@
                                                                                                                        TRUNCATION: @@@@@@@@@
TMAGE
                 MONJV
                    TRAY
                  :(###,###)
                                                                          MESSAGE=PAGE:@@@
                                                                                                                       CHECKPOINT:@@@@@@@
POSITION
                                                                                                                        : ####
NUMBER: #####
                                                                                                                        : ####
                                                                                         POSITION
NUMBER: #####
                                                                                                                        : ####
                                                                                         POSITION
LANG-EXT: @@@@@@@ (@@@@@@@@@@@@@)
PAGEDEF :@@@@@@@ LP65DIA(###,###)
                                                                                         DUPLEX
                                                                                                                        : @@@@@@
                    :(000,000,000,000)
                                                                                                                        : (@@@@,##)
CHARS
                                                                                         CHARPOOL
FOB
                    · @@@@
                                          FORMDFF:@@@@@@@@@
                                                                                         ROTATION
                                                                                                                        : ########
OVERLAY :(
                                                                   )
                                                                                         OFFSET
                                                                                                                        :(L:
                                                                                                                                         .T:
                                                                                                                                                        )
CLIFNT-H: @@@@@@@@
                                                                                                                        : @@@@@@
                                                                                         CLUSTER
FILE :SIZE=######.FCBTYPE=@@@@.RECFORM=@.RECSIZE=#####.BLKSIZE=######
               CCSNAME=@@@@@@@@
                                                                   FILE-CR-MODE=@@@@@@@@@@@@@
PRNT :#LINES=#######
                                                                                    COPY#=####
                                                                                                                          SPOOLOUT CLASS=@@@
```

Layout of the local SPOOL trailer page

Notes

- 1. The name of the host to which the print job was passed is output in the "CLIENT-H" field.
- 'LOCAL' is output in the "CLUSTER" field for local jobs or distributed jobs within the same cluster. 'SINIX' is output for distributed jobs between clusters if the print job was started from a UNIX system based cluster. 'BS2000' is output for distributed jobs between clusters if the print job was started from a BS2000 cluster.
- 3. The "START TIME" and "END TIME" fields have been updated to show summer and winter times.
- 4. The name of the coded character set used in the file is displayed in the "CCSNAME"field.
- 5. The file creation mode is displayed in the "FILE-CR-MODE" field if the print job requested the product DSEM.

8.4 Spoolout accounting record (SPLO)

The spoolout accounting record is written on termination of a spoolout job.

Information about the job requester, the print job and its processing is stored for each print job in this accounting record. The accounting record contains the following data in all cases:

- user ID, account number and group name
- the TSN of the controller task and of the spoolout job
- date and time with a four-digit year number for the start and end of job processing

Note

The layout of the accounting record can be generated using the ACSPLO macro.

The following attributes are used for each data field:

Field	Consecutive number of the data field in the part of the record described
Displace- ment	Relative distance of the data field to the beginning of the part of the record described
1	I could of the date field in hotes

Length Length of the data field in bytes

Format A = Alphanumeric

B = Binary number

B2 = Binary CPU time

C = Printable characters, including special characters

F = File name

X = Nonprintable characters

Z = Unpacked decimal number (*)

- * = Specified for the individual record types or extension elements
- Reserved for future extensions and contains either blanks or binary zero
- (*) The time is represented in the form hhmmss The date is represented in the form yymmdd

(A) Record definition

Record identifier: 'SPLO'

(B) Identification section

The identification section of the accounting records for the accounting of user tasks consists of a user identification.

Structure and contents:

Field	Displacement		Length	Format	Meaning	Notes
No.	Hex.	Dec.	(bytes)			
1	00	0	8	Α	User ID	
2	08	8	8	Α	Account number	
3	10	16	4	С	TSN	1)
4	14	20	8	Α	Group name	
5	1C	28	8	Α	Host name	2)

Notes

- 1) TSN of the controller task
- 2) Name of the computer from which the spoolout job is requested. This field is only filled when the product Distributed Print Services is used.

Length of the user identification: 36 bytes

(C) Basic information

Field	Displacem. Length		Format	Meaning	Notes	
no.	Hex.	Dec.	(bytes)			
1	00	0	6	Z	Date of spoolout start	1)
2	06	6	6	Z	Time of spoolout start	2)
3	0C	12	6	Z	Date of spoolout end	1)
4	12	18	6	Z	Time of spoolout end	2)
5	18	24	8	Α	Spoolout job name (PNAME)	
6	20	32	4	Z	TSN of the spoolout job	
7	24	36	2	В	Number of copies to be output	
8	26	38	1	В	Spoolout class	
9	27	39	1	В	Spoolout scheduling priority	
10	28	40	3	С	Type of spoolout file	3)
11	2B	43	1	-	- reserved -	
12	2C	44	2	Z	Century for spoolout start	4)
13	2E	46	1	Α	Time-of-the-year code for spoolout start	5)
14	2F	47	1	Α	Time-of-the-year code for spoolout end	5)
15	30	48	2	Z	Century for spoolout end	4)
16	32	50	4	Z	TSN of the partner print job	6)

Length of the basic information: 54 bytes

Notes

- 1. Date in the form yymmdd.
- 2. Time in the form hhmmss.
- 3. The following displays are possible:

'SYS' = logical system files (SYSLST, SYSOUT, SYSOPT)

'OMF' = EAM object module file

'EAM' = user EAM file

'PLM' = PLAM library element

' ' = other file

'TMP' = temporary file

4. E.g. Character string "19" as indicator for the year "1993".

- 5. "S" for summer time; "W" for winter time.
- 6. Two cases are discriminated for this TSN:
 - If the print job was issued on a remote computer and processed on the local server, it is then the TSN on the client or gateway computer.
 - If the print job was issued on the local computer and processed on the local server,
 it is then the same TSN as that of the print job.

(D) Variable Information

The variable information of the spoolout accounting record contains seven record extensions.

Structure and contents:

Field	Displacem. Length		Format	Meaning	Notes	
no.	Hex.	Dec.	(bytes)			
1	00	0	2	В	X'0007'	1)
2	02	2	2	В	Displacement of the 1st extension	2)
3	04	4	2	В	Displacement from the 1st to the 2nd extension	3)
4	06	6	2	В	Displacement from the 1st to the 3rd extension	
5	08	8	2	В	Displacement from the 1st to the 4th extension	
6	0A	10	2	В	Displacement from the 1st to the 5th extension	
7	0C	12	2	В	Displacement from the 1st to the 6th extension	
8	0E	14	2	В	Displacement from the 1st to the 7th extension	
9	10	16	2	В	Displacement from the 1st to the 8th extension	

- 1. Is 0 if there is no record extension available.
- 2. Refers to the displacement from the beginning of the extension record.
- 3. If an extension is not supported, it receives a displacement of 0.

Length of the extension header: 18 bytes

Extension 1: Spoolout termination cause

This record extension is a structure field extension comprising one element.

Field	Displacem. Length		Length	Format	Meaning	Notes
no.	Hex.	Dec.	(bytes)			
1	00	0	2	Α	Extension identifier 'OT'	
2	02	2	1	В	Number of elements (X'01')	
3	03	3	1	В	Element length (X'0C')	
4	04	4	2	Α	Termination indicator 'T' = normal termination 'A' = abnormal termination	
5	06	6	1	-	- reserved -	
6	07	7	1	Α	Termination request	1)
7	08	8	7	A/C	Termination code	1)
8	0F	15	1	-	- reserved -	

Length of the spoolout termination extension: 16 bytes

Note

Termination request	Terminat	ion code
'F'	NORM	normal termination
'E'	DMS	error during reading of file
' \$'	I/O	input/output error on output device
'X'	CAN	termination by /CANCEL-PRINT-JOB

Extension 2: Spoolout creation

This extension is a structure field extension containing one element.

Field	Displacem. Leng		Length	Format	Meaning	Notes
no.	Hex.	Dec.	(bytes)			
1	00	0	2	Α	Extension identifier '0C'	
2	02	2	1	В	Number of elements (X'01')	
3	03	3	1	В	Element length (X'1D')	
4	04	4	4	Z	TSN of creating task	
5	08	8	2	-	- reserved -	
6	0A	10	12	Z	Date and time of spoolout job creation	
7	18	22	2	Z	Century of the date	
8	1A	24	8	Α	Original user ID (in the case of replay jobs)	
9	20	32	1	Α	Time-of-the-year code for spoolout start	

Length of the spoolout creation extension: 33 bytes

Extension 3: Spoolout initiation

This extension is a case distinction extension.

It is present only if the spoolout process was triggered by way of the HOLD-PRINT-JOB command (RESUME-CONDITION operand).

Field	Displa	acem.	Length	Format	Meaning	Notes
no.	Hex.	Dec.	(bytes)			
1	00	0	2	Α	Extension identifier 'OI'	
2	02	2	1	В	Number of elements (X'01')	
3	03	3	1	В	Element length (X'13')	
4	04	4	2	Α	Case identifier 'RE'	
5	06	6	2	-	- reserved -	
6	08	8	12	-	- reserved -	
7	14	20	2	Z	- reserved -	
8	16	22	1	Α	- reserved -	

Length of the HOLD-PRINT-JOB command extension: 23 bytes

Extension 4: Spoolout input tape

This extension is a structure field extension containing one element.

It is generated only if the output file was read from a user tape or replay tape.

A user tape is generated using the WRITE-SPOOL-TAPE command.

A replay tape is a tape which is assigned for spoolout/spoolin by the operator or system administration using the START-TAPE-OUTPUT command.

Field	Displacem. L		Length	Format	Meaning	Notes
no.	Hex.	Dec.	(bytes)			
1	00	0	2	Α	Extension identifier 'IN'	
2	02	2	1	В	Number of elements (X'01')	
3	03	3	1	В	Element length (X'14')	
4	04	4	2	-	- reserved -	
5	06	6	2	Α	Mnemonic name of tape unit	
6	08	8	8	-	- reserved -	
7	10	16	6	-	- reserved -	
8	16	22	2	-	reserved -	

Length of the spoolout input tape extension: 24 bytes

Extension 5: Output medium

This extension is a case distinction extension.

The associated extension identifier is always 'OM'.

Format and contents depend on the case identifier, which also indicates in which manner spoolout has taken place.

a) Line printer extension

The spoolout file was printed on a SPOOL or RSO line printer.

Field	ld Displacem.		Length	Format	Meaning	Notes
no.	Hex.	Dec.	(bytes)			
1	00	0	2	Α	Extension identifier 'OM'	
2	02	2	1	В	Number of elements (X'01')	
3	03	3	1	В	Element length (X'20')	
4	04	4	2	Α	A Case identifier '	
5	06	6	2	Α	A Mnemonic device name of printer	
6	08	8	4	В	B Number of printed lines	
7	0C	12	4	В	Number of printed pages	1)
8	10	16	8	Α	Device name	
9	18	24	6	Α	Form name	
10	1E	30	2	-	reserved -	
11	20	32	3	-	- reserved -	
12	23	35	1	В	Component identifier	

Length of the line printer extension: 36 bytes

Notes

- 1. The values for printed lines/pages cannot be guaranteed during the spoolout in the event of printer problems. As far as type 3365 Printers in particular are concerned, the values are invalid in the following cases:
 - control characters in the input file
 - additional data passed to the printer via the PCL file
 - expanded mode is set to 'ON'
 - PCL sends special options to the printer (e.g. more than one logical page per physical page)

Generally speaking, it is also important that in all cases in which the printer does not behave like a line printer, the number of pages and lines cannot be complemented correctly.

2. Identifies whether output was on a local printer via a channel trunk or on an RSO printer via a data transmission link.

Permissible values:

- X'01' for local printer via a channel connection
- X'02' for RSO printer via data transmission link
- X'03' for printer on a bus channel
- X'05' for Dprint jobs within a cluster for local printer
- X'15' for Dprint jobs between clusters for local printer
- X'16' for Dprint jobs between clusters for RSO printer

b) APA page printer extension

The spoolout file was printed on an APA page printer.

Field	Displa	acem.	Length	Format	Meaning	Notes
no.	hex.	dec.	(bytes)			
1	00	0	2	Α	Extension id 'OM'	
2	02	2	1	В	Number of elements (X'01')	
3	03	3	1	В	Element length (X'50')	
4	04	4	2	Α	Case identifier 'AP'	
5	06	6	2	Α	Mnemonic device name	
6	08	8	4	-	reserved	
7	0C	12	4	-	reserved	
8	10	16	8	Α	Device name	
9	18	24	6	Α	A Form name	
10	1E	30	2	-	- reserved	
11	20	32	3	-	- reserved	
12	23	35	1	В	B Device access	
13	24	36	4	В	# of transmissions	
14	28	40	4	В	# of printed pages	
15	2C	44	4	В	# of printed page-sides	
16	30	48	4	В	Time used (.01 sec)	
17	34	52	4	В	# of extra pages printed	
18	38	56	4	В	# of extra page-sides printed	
19	3C	60	4	В	Extra time used (.01 sec)	
20	40	64	4	В	# of PAGEDEF's requested	
21	44	68	4	В	# of FORMDEF's requested	
22	48	72	4	В	# of FONTS requested	

Field	Displacem.		Displacem.		Length	Format	Meaning	Notes
no.	hex.	dec.	(bytes)					
23	4C	76	4	В	# of FONTS loaded			
24	50	80	4	В	# of overlays requested			
25	54	84	4	В	# of overlays loaded			
26	58	88	4	В	# of page segments requested			
27	5C	92	4	В	# of page segments loaded			
28	60	96	1	В	Input bin flag			
29	61	97	1	В	Output bin flag			
30	62	98	1	В	Duplex flag			
31	63	99	37	-	reserved			

Length of APA page printer extension: 136 bytes

Notes

- 1. This field defines whether the print job is local or distributed, i.e. whether the device is addressed over a channel or the network. The possible values are:
 - X'01' for a local printer over a channel
 - X'05' Dprint print jobs within a cluster for local printers
 - X'15' Dprint print jobs between clusters for local printers

Extension 6: Spoolout file name

This extension is a character string extension.

Field	Displacem.		Length	Format	Meaning	Notes
no.	Hex.	Dec.	(bytes)			
1	00	0	2	Α	Extension identifier 'FN'	
2	02	2	1	В	X'00'	
3	03	3	1	В	File name length (L)	
4	04	4	54	F/Z	Name of spoolout file	1)
5	ЗА	58	64	F/Z	Element name	
6	7A	122	24	С	Element version	
7	92	146	8	С	Element type	
8	9A	154	2	С	Number of records	2)

Maximum length of the file name extension: 156 bytes

Notes

- File name or name of the PLAM library according to the spoolout report. For object
 modules from the SYSEAM file, this field contains "*EAM-OMF"; for other temporary
 files from SYSEAM, this field contains a ten-digit file number.
 For cataloged files, the file name from the PRINT-DOCUMENT command is output.
- 2. This field corresponds to the data in the PRINT-DOCUMENT command with respect to the number of pages to be printed.

Extension 7: Account ID

This extension contains character strings.

Field	Displacem. Len		Length	Format	Meaning	Notes
no.	Hex.	Dec.	(byte)			
1	00	0	2	Α	Extension identifier 'ID'	
2	02	2	1	В	B X'00'	
3	03	3	1	В	B Length L of the account ID	
4	04	4	L	C/X	Spoolout file name	1)

Length of the account ID extension: 12 bytes

Note

 The account ID is entered by the user with either the WRITE-ACCOUNTING-RECORD command (USER-ACCOUNTING-STEP operand) or the AREC macro (ID operand). If no account ID was specified for the current print job, the field is filled with x'FEFFFFFFFFFFF.

Extension 8: Jobs from external users

This extension only applies if the spoolout process is started with a SPOOLOUT command from UNIX systems or WPRINT.

Field	Displacem. Length		Length	Format	Meaning	Notes
No.	Hex.	Dec.	(byte)			
1	00	0	2	Α	Extension identifier 'EU'	
2	02	2	1	В	Number of elements (X'00')	
3	03	3	1	В	Length of the element	
4	04	4	L	C/X	External user ID	1)

Maximum length of the extension for the account ID: 28 byte

Note

1. The maximum length of the external user ID is 24 bytes.

General note

Maximum length of the spoolout accounting record: 504 bytes

8.5 Application rules for C

Spool & Print - Macros support an interface for C programming in order to access various functions which are available as of SPOOL V4.1. However, the following application rules must be observed.

8.5.1 General rules

1. The default header has to be initialized with values that correspond to the interface. The table below contains the information required.

Macro name	Unit	Function	Version
STRSPPR	46	57	1
STRSRSO	46	57	1
STPSPPR	46	59	1
SUPSPPR	46	60	1
SASDSPO	46	24	1

- 2. Pointer fields must be initialized with ((voi *)-1) and not with *null*.
- 3. String fields must be initialized with blanks (X'40') and must not end with $\0$.
- 4. Queries with flag values can be used for a field. However, you should definitely refer to the interface description before doing so.

Example

The EXIT operand of the SASDSPO macro:

Operand value	Corresponding flag
EXIT=*ANY	SASDSPOexit_any
EXIT=*YES	SASDSPOexit_yes
EXIT=*NO	SASDSPOexit_no

5. List or wildcard operands

With the entry type *LIST, you can specify that the operand contains a list.

With the entry type *WILD, you can specify that the name is partially qualified with wildcards. The name operand is not set and the wildcard operand contains the current value.

Support of *LIST in C: flag_cx must be set to flag_list and the table element contains the list.

Support of *WILD in C: flag_cx must be set to flag_wildcard, wldname contains the current value and the table element is initialized with blanks.

The structured operands FOB, PRIORITY and CHARACTER-SET need to be specially pointed out with regard to structure initialization. In other languages, initialization is performed through MF support.

Syntax: *FIELD*=(*type*, *low*, *high*)

flag set must be used to set the value for the entry type.

In the case of an area type, the area should be selected according to the following table. It must be set to the current value. The use of *flag set* is strictly prohibited.

Operand name		STRSPPR	SUPSPPR	SASDSPO	Value to be set	
FOB	low	*STD	*STD/*UNCHANGED	*MIN	0	
	high	*STD	*STD/*UNCHANGED	*STD	4032	
PRIORITY	from	*MIN	*MIN/*UNCHANGED	*MIN	30	
	to	*MAX	*MAX/*UNCHANGED	*MAX	255	
CSN	low	*MIN	*MIN/*UNCHANGED	*MIN	1	
	high	*MAX	*MAX/*UNCHANGED	*MAX	64	

8.5.2 STRSPPR

The field 'specifiedX' must be set to false (=0).

8.5.3 SUPSPPR

- 1. The P/L layout allocated to RSO corresponds to the STRSRSO macro.
- 2. If 'unchanged' was selected for the flag type, the relevant field 'specifiedX' must be set to false (=0). This requires no further information for the parameter.
- 3. The field 'specifiedX' must be set to true (=1) if the relevant default value was chosen for the operand in STRSPPR.
- 4. The field for page printer output does not need to be initialized. The value is set internally by the macro. The field exists as an identical P/L layout for STRSPPR and SUPSPPR.
- Values for special flags must be defined for the default values for the operands PRIORITY and CHARACTER-SET of the STRSPPR macro to be set. You use SUPSPPRflag_minprio and SUPSPPRflag_maxprio for the PRIORITY operand SUPSPPRflag_minchars and SUPSPPRflag_maxchars for the CHARACTER-SET operand.
- 6. If ROSPAR=*NONE is selected, the pointer must be initialized with ((void *) -1) and not with flag *none*.

Related publications

Ordering manuals

The manuals are available as online manuals, see http://manuals.fujitsu-siemens.com, or in printed form which must be paid for and ordered separately at http://FSC-manualshop.com.

BS2000/OSD Softbooks English CD-ROM

Target group
BS2000/OSD users

Contents

The CD-ROM "BS2000/OSD SoftBooks English" contains almost all of the English manuals and README files for the BS2000 system software of the latest BS2000/OSD version and also of the previous versions, including the manuals listed here.

These Softbooks can also be found in the Internet on our manual server. You can browse in any of these manuals or download the entire manual.

Order number

U26175-J8-Z125-1-76

Internet address

http://manuals.fujitsu-siemens.com

BS2000/OSD-BC

Commands, Volumes 1 - 5

User Guide

Target group

This manual is addressed to nonprivileged users and systems support staff.

Contents

Volumes 1 through 5 contain the BS2000/OSD commands ADD-... to WRITE-... (basic configuration and selected products) with the functionality for all privileges. The command and operand functions are described in detail, supported by examples to aid understanding. An introductory overview provides information on all the commands described in Volumes 1 through 5.

The Appendix of Volume 1 includes information on command input, conditional job variable expressions, system files, job switches, and device and volume types.

The Appendix of Volumes 4 and 5 contains an overview of the output columns of the SHOW commands of the component NDM. The Appendix of Volume 5 contains additionally an overview of all START commands.

There is a comprehensive index covering all entries for Volumes 1 through 5.

BS2000/OSD-BC

Commands, Volume 6, Output in S Variables and SDF-P-BASYS

User Guide

Target group

This manual is addressed to programmers and users who write procedures.

Contents

Volume 6 contains tables of all S variables that are supplied with values by the SHOW commands in conjunction with structured output. Further chapters deal with:

- introduction to working with S variables
- SDF-P-BASYS

BS2000/OSD-BC

DMS Macros

User Guide

Target group

The manual addresses assembly language programmers.

Contents

The manual describes the DMS macro interface for the BS2000/OSD basic configuration.

There is a brief description of the access method-specific features relevant to programming, followed by a description of the macros in alphabetical order.

BS2000/OSD-BC Executive Macros

User Guide

Target group

The manual addresses all BS2000/OSD assembly language programmers.

Contents

The manual contains a summary of all Executive macros, detailed descriptions of each macro with notes and examples, including job variable macros, and a comprehensive general training section.

BS2000/OSD-BC Introductory Guide to DMS

User Guide

Target group

This manual is addressed to nonprivileged users and systems support staff.

Contents

It describes file management and processing in BS2000.

Attention is focused on the following topics:

- volumes and files
- file and catalog management
- file and data protection
- OPEN, CLOSE and EOV processing
- DMS access methods (SAM, ISAM,...)

BS2000/OSD-BC Introductory Guide to Systems Support

User Guide

Target group

This manual is addressed to BS2000/OSD systems support staff and operators.

Contents

The manual covers the following topics relating to the management and monitoring of the BS2000/OSD basic configuration: system initialization, parameter service, job and task control, memory/device/system time/user/file/pubset management, assignment of privileges, accounting and operator functions.

BS2000/OSD-BC **System Exits**

User Guide

Target group

The manual addresses systems support.

Contents

The manual contains an introduction to the system exits with a description of the base mechanism. The main section comprises descriptions of all system exits, e.g. exits for the BS2000/OSD basic configuration, SPOOL, DCAM, SDF and PLAM.

Distributed Print Services (BS2000/OSD)

Printing in Computer Networks

User Guide

Target group

This manual is intended for nonprivileged users, device administrators and systems support of BS2000/OSD.

Contents

The manual provides descriptions of the principles, use and administration of Distributed Print Services for each of these user groups. Possible uses of Distributed Print Services are illustrated by examples.

DSSM/SSCM

Subsystem Management in BS2000/OSD

User Guide

Target group

This manual addresses systems support staff and software consultants of BS2000/OSD.

Contents

The following are described: BS2000/OSD subsystem concept, dynamic subsystem management (DSSM), subsystem catalog management (SSCM) and the associated commands and statements.

IDOM (BS2000/OSD)

Integrated Document and Output Management User Guide

Target group

The manual addresses nonprivileged users and system administrators of BS2000/OSD. **Contents**

This manual describes the SPOOL subsystem IDOM, which offers document and output management in BS2000/OSD.

JV (BS2000/OSD)

Job Variables

User Guide

Target group

The manual addresses both nonprivileged users and systems support.

Contents

The manual describes management and possible uses of job variables. The command descriptions are divided according to function areas. The macro calls are described in a separate chapter.

PRISMAproduction/BS2000 SPS

Benutzerhandbuch

GERMAN ONLY

Target group

The manual addresses SPOOL users and systems support.

Contents

This manual describes printing on APA printers with the SPOOL subsystem SPS. Installation of SPS and APA printers, creation and provision of print file and print resources, printing and page presentation methods, and error handling are described.

PRM (BS2000/OSD)

User Guide

Target group

The manual addresses SPOOL users, systems support and RSO device administrators.

Contents

This manual describes the PRM utility routine for creating and managing print resources for BS2000 SPOOL. The manual deals with the description of the two PRM user interfaces: the SDF statements for interactive and batch mode, and the FHS-based menu interface for

interactive mode.

RSO (BS2000/OSD)

Remote SPOOL Output

User Guide

Target group

This manual is directed at nonprivileged users, RSO device administrators, SPOOL administrators and systems support of BS2000/OSD.

Contents

The manual describes the functions and options of the user groups with respect to utilizing and controlling decentralized printers (RSO printers) and deals with the technical characteristics of all RSO printers.

SDF (BS2000/OSD)

Introductory Guide to the SDF Dialog Interface

User Guide

Target group

BS2000/OSD users

Contents

This manual describes the interactive input of commands and statements in SDF format. A Getting Started chapter with easy-to-understand examples and further comprehensive examples facilitates use of SDF. SDF syntax files are discussed.

SDF-P (BS2000/OSD)

Programming in the Command Language

User Guide

Target group

The manual addresses BS2000/OSD users and systems support.

Contents

SDF-P is a structured procedure language in BS2000. The introduction is followed by a detailed description of commands, functions and macros.

SNS (BS2000/OSD)

SPOOL Notification Service

User Guide

Target group

The manual addresses nonprivileged users and systems support of BS2000/OSD.

Contents

This manual describes the SNS subsystem, which provides a tool for sending and managing notifications in the frame of BS2000/OSD.

SPCONV (BS2000/OSD)

User Guide

Target group

This manual is intended for systems support staff and RSO device administrators.

Contents

The manual describes the SPCONV subsystem. Use of SPCONV is mandatory when working with SPOOL as of Version 3.2A. The manual describes the structure of the filter system and the use of filters.

Spool & Print - Commands (BS2000/OSD)

User Guide

Target group

This manual is intended for nonprivileged users, device administrators, cluster administrators, SPOOL administrators and system support staff.

Contents

The commands available for SPOOL, Dprint, RSO and SPS are described, but not those for subsystem management and job control.

Spool & Print - Messages

User Guide

Target group

This manual addresses systems support, RSO device administrators and nonprivileged users.

Contents

Messages for SPOOL, RSO, SPSERVE, PRM, SPCONV, SPS, SPOOLSYS, IDOM and Distributed Print Services. The English message texts and meaning and response texts are included in the manual. Guaranteed messages are marked.

SPOOL (BS2000/OSD)

User Guide

Target group

This manual is intended for nonprivileged users, Spool & Print administrators, RSO device administrators and systems support staff.

Contents

The manual describes the operation of SPOOL.

SPSERVE (BS2000/OSD)

User Guide

Target group

This manual is addressed to nonprivileged users, RSO device administrators, Dprint cluster administrators and those responsible for BS2000/OSD system operation.

Contents

The manual describes the SPSERVE utility routine with all its statements. It takes account of all extensions to SPOOL, RSO, SPCONV, PRM, Distributed Print Services, and SPS.

Wprint (Windows])

Target group

Users who want to print from within Windows applications and system administrators of SINIX systems, UNIX systems and BS2000/OSD systems as well as SINIX Spool administrators.

Contents

This manual describes the operation and functions of the Wprint-Server and Wprint-Client components as well as their installation and configuration.

Xprint (Reliant UNIX systems)

Application Programming Interface (API)

User Guide

Target group

This manual is for programmers who want to make use of the Xprint functionality in their programs.

Contents

The manual contains a reference of all Xprint API functions and the corresponding macros. It describes how you can use the functions in your own programs.

Xprint (Reliant UNIX systems)

Menues

User Guide

Target group

Users and system administrators of Reliant UNIX systems, Xprint administrators Contents

This manual describes how to use Xprint via the menu interface.

Xprint (Reliant UNIX systems)

Reference Guide

Target group

Users and system administrators of Reliant UNIX systems, Xprint administrators Contents

This manual describes the Xprint commands in alphabetical order, and documents the Xprint messages, the configuration files for Xprint objects as well as the standard Xprint data formats. The manual also presents an overview of how the Xprint system works. In addition, basic information is provided about the Xprint system as well as concrete handling instructions using selected examples.

Xprint (Reliant UNIX systems) User and Administrator Guide

Target group

Users and system administrators of Reliant UNIX systems, Xprint administrators Contents

This manual provides information on the Xprint concept and documents the objects that make up to the Xprint. In addition, it describes how Xprint is installed as well as how it is used via the command interface.

Related publications			

Index

	PRNTSPS macro 177
\$DPFL parameter list	SHOPJAT macro 283
exit 093 534	SHOPJST macro 291
	SJAODOF macro 356
Α	SJAODOP macro 358
accounting record	SJAOLCT macro 361
SPLO 593	SJAOPJC macro 364
ADD-DPRINT-FT-LIMIT command 575	SJAORSD macro 365
administration	SJAORSO macro 368
utility routines 2	SJAOTOP macro 367
APA printer	SJSODES macro 375, 378, 381
format definition 179	SJSODIS macro 384
application notes	SJSODSS macro 386
exit 097 554	SJSOORI macro 388
Application rules for C 605	SJSOORS macro 390
ASA	SJSOTRA macro 391
feed control character 30	SJSOTRS macro 393
print control byte 30	SPJADPC macro 369
assignment	SPJASPS macro 370
font 31	SPJSRSO macro 371
D	SPSINF macro 189, 190
B	CANPJOB macro 205
barcode 7, 187	parameter list 209
С	return codes 208
C header file (PRNTDPC macro) 175	character set 31
call format	specify 31
CANPJOB macro 205	command
MODJRSO macro 341	ADD-DPRINT-FT-LIMIT 575
MODPJAT macro 210	MODIFY-DPRINT-FT-LIMIT 574
MPJADPC macro 348	REMOVE-DPRINT-FT-LIMIT 576
MPJASPS macro 350	SHOW-DPRINT-FT-LIMIT 573
PRNTDOC macro 41	contents
PRNTDPC macro 172	job variable for spoolout 36
PRNTRSO macro 8, 166	control characters
THINTISO MACIO 0, 100	interpret 33

controlling print job 27	exit 091
customer filter exit 093 531	parameter list 518 programming example 527
GAIL 000 SOT	return codes 517
D	exit 092
defining	parameter list 521
font 34	programming example 529
form 34	exit 093
loop 29, 34	\$DPFL parameter list 534
DELETE-FILE	programming example 538
after file output 54, 237	return codes 533
deleting file after output 54, 237	security aspects 543
Dprint	exit 094
brief description 1	parameter list 545 programming example 546
DSECT (SSVSCB macro) 506	return codes 545
duplex mode 100	exit 095
•	parameter list 549
E	return codes 548
EBCDIC feed control characters 29	exit 096
EBCDIC printer control byte 30	parameter list 552
error processing	return codes 551
job variable 35 error recovery for virtual printers 497	exit 097
exit 505	application notes 554
090 for spoolout 509	parameter list 556
091 for header pages 516	programming example 557 return codes 555
092 for trailer pages 516	exit mechanism 589
093 (customer filter) 531	exit routine 509
094 for resource conversion 544	information transfer 589
095 for printer output 548	
095 output routine 548	F
096 for server selection 551	feed control characters
097 (control File Transfer) 553	for output 29
for header page (091) 516, 551 for trailer page (092) 516, 551	file output
general information 506	to printer 27
exit 090	file protection in spoolout job in wait state 77, 236
parameter list 510	File Transfer control
programming example 514	exit 097 553
return codes 509	FOB data overlay 258

font 31 define 34 for print page layout 179 specify 31 font pool 32 form define 34	loop controlling paper feed 29 define 29, 34 LP65 printer paper input tray 65, 245 paper output tray 81, 251
specifying for printing 28	M
format definition	macro
APA printer 179	CANPJOB 205
.	MODJRSO 341
H	MODPJAT 210
header page exit (091) 551	MPJADPC 348
header pages	MPJASPS 350
exit 091 516	OLTPRQX 8
structure 524	PRNTDOC 27
	PRNTDPC 172
I	PRNTRSO 166
IDOM 1	PRNTSPS 177
incompatible operands	SHOPJAT 283
PRNTDOC macro 105	SHOPJST 291
information	SJAODOF 356
on SPOOL parameters 189	SJAODOP 358
interpreting	SJAOLCT 361
control characters 33	SJAOPJC 364
_	SJAORSD 365
J	SJAOTOP 367
job name for spoolout job 85, 234	SJSODES 375, 378, 381
job variable	SJSODIS 384
contents 36	SJSODSS 386
for spoolout 35	SJSOORI 388
SPOOL job 35	SJSOORS 390
spoolout, error processing 35	SJSOTRA 391
L	SJSOTRS 393
laser printers	SNPPRNT 182
examples 33	SPJADPC 369
layout control characters	SPJARSO 368
interpret 33	SPJASPS 370
library member	SPJSRSO 371
output to printer 27	SPSINF 189
LOCK-FILE 77, 236	SPVDEND 491
long form	SSVBARC 187
operand names 326	SSVSCB 506
	macro format 577

macro syntax 580 macro type call format 582 general description 582 type 0 582 type R 582 type S 583 macros	OLTPRQX (macro) 8 operand names long form 326 operands incompatibility, PRNTDOC macro 105 output feed control characters 29 file with PRNTDOC macro 27
metasyntax 578, 580 standard header 586 type O 582 type R 582 type S 583 wildcards 588	format of output line, hexadecimal 79, 242 SPOOL parameters 189 to printer 27 output routine exit 095 548
managing printers 395	P
manpage	page definition 178
printer command 616	page rotation 92, 244
nanual	paper feed
target group 4	controlling 29
metasyntax of the macros 578, 579	paper input tray 65, 245
minimum output requirements 28	LP65 printer 65, 245
minimum requirements	RSO printer 245
for printer 28	RSO printers 65
MODIFY-DPRINT-FT-LIMIT command 574	paper output tray 81, 251
MODJAT macro	LP65 printer 81, 251
return codes 270	RSO printer 81, 251
MODJRSO macro 341	sort mechanism 82, 95, 252, 264
parameter list 346	parameter list
MODPJAT macro 210	CANPJOB macro 209 exit 090 510
define values 341	exit 090 510 exit 091 518
parameter list 272	exit 091 516 exit 092 521
nonitoring	exit 092 521 exit 093 534
spoolout 35	exit 093 534 exit 094 545
monitoring job variable	exit 094 549
for spoolout 35	exit 095 549
SPOOL job 35	exit 030 552 exit 097 556
MONJV (SPOOL job) 35 MPJADPC macro 348	MODJRSO macro 346
parameter list 349	MODPJAT macro 272
MPJASPS macro 350	MPJADPC macro 349
parameter list 354	MPJASPS macro 354
parameter not out	PRNTDOC macro 109
)	PRNTDPC macro 174
OLTPRQX 8	PRNTRSO macro 171

parameter list (cont.)	printer output
PRNTSPS macro 181	exit 095 548
SHOPJAT macro 288	printer see Xprint
SHOPJST macro 331, 419, 437	printer selection
SJAODOF macro 356	implicit 28
SJAODOP macro 358	printing
SJAOLCT macro 361	controlling paper feed 29
SJAOPJC macro 364	specifying form 28
SJAORSD macro 365	private
SJAOTOP macro 367	define font 34
SJSODES macro 376, 379, 382	define form 34
SJSODIS macro 385	define loop 34
SJSODSS macro 387	private loop 29
SJSOORI macro 388	PRM
SJSOORS macro 390	brief description 2
SJSOTRA macro 391	PRNTDOC macro 27
SJSOTRS macro 393	define values 166
SPJADPC macro 369	incompatible operands 105
SPJARSO macro 368	parameter list 109
SPJASPS macro 370	return codes 107
SPJSRSO macro 372	PRNTDPC macro 172
pcl	parameter list 174
definitions 616	PRNTRSO macro 166
PCL file 178	parameter list 171
PostScript	PRNTSPS macro 177
print to PostScript printers 616	parameter list 181
print control byte	programming example
ASA 30	exit 090 514
print job	exit 091 527
control 27	exit 092 529
print page layout	exit 093 538
font 179	exit 094 546
print requirements 28	exit 097 557
PRINT-DOCUMENT command 27	
printer	Q
feed control characters 29	queue
format of output line, hexadecimal 242	spoolout 27
format of the output line, hexadecimal 79	_
output to 27	R
pcl definitions 616	REMOVE-DPRINT-FT-LIMIT command 576
print to PostScript printers 616	requirements
select 28	minimum, for printer 28
printer control byte 30	resource conversion
	exit 094 544

return codes	SJAOLCT macro 361
CANPJOB macro 208	parameter list 361
exit 090 509	SJAOPJC macro 364
exit 091 517	parameter list 364
exit 093 533	SJAORSD macro 365
exit 094 545	parameter list 365
exit 095 548	SJAOTOP macro 367
exit 096 551	parameter list 367
exit 097 555	SJSODES (macro) 378, 381
MODPJAT macro 270	SJSODES macro 375
PRNTDOC macro 107	parameter list 376, 379, 382
SHOPJAT macro 287	SJSODIS macro 384
SHOPJST macro 329	parameter list 385
SPSINF macro 195	SJSODSS macro 386
rotation 92, 244	parameter list 387
RSO	SJSOORI macro 388
brief description 1	parameter list 388
RSO printer	SJSOORS macro 390
paper input tray 65, 245	parameter list 390
paper output tray 81, 251	SJSOTRA macro 391
	parameter list 391
S	SJSOTRS macro 393
SASDSPO (macro) 396	parameter list 393
SCB 27, 506	SNPPRNT macro 182
security aspects (exit 093) 543	SNS
server selection	short description 2
exit 096 551	sort mechanism
SHOPJAT macro 283	paper output tray 82, 95, 252, 264
define values 368	SPCONV
parameter list 288	brief description 2
return codes 287	specifying
SHOPJST macro 291	character set 31
define values 371	SPJADPC macro 369
parameter list 331, 419, 437	parameter list 369
return codes 329	SPJARSO macro 368
SHOW-DPRINT-FT-LIMIT command 573	parameter list 368
Siemens	SPJASPS macro 370
feed control characters 30	parameter list 370
printer control byte 30	SPJSRSO macro 371
simplex mode 100	parameter list 372
SJAODOF macro 356	SPLO 594
parameter list 356	SPOOL
SJAODOP macro 358	brief description 1
parameter list 358	exits 505

Spool & Print Explorer brief description 2 Spool & Print services short description 1 SPOOL accounting record SPLO 593 SPOOL control block 27	trailer page exit (092) 516, 551 trailer pages exit 092 516 structure 524 TRC control characters 179, 352 TSN
SPOOL control block 27 SPOOL job variable 35	common 59
SPOOL parameter file	U
output contents 189	utility routines
spoolout	for administration 2
contents of the job variable 36	
control 28	V
error processing of the job variable 35	variable
job name 85, 234	contents during spoolout monitoring 36 error processing of the spoolout job
monitor 35	variable 35
queue 27	monitoring spoolout 35
spoolout accounting record	virtual print
SPLO 593	error recovery 497
spoolout exit (090) 509 SPOOLOUT-NAME 85, 234	stopping processing 496
SPS	virtual printer
brief description 1	communication with application 493
SPSERVE	initializing dialog 493
brief description 2	PRNTDOC macro 102
SPSINF macro 189	processing print jobs 495
return codes 195	SPVDEND macro 491
SPVDEND macro 491	W
SSVBARC (macro) 187	wait state
SSVBARC macro 187	spoolout job, file protection 77, 236
SSVSCB macro 506	wildcards 588
standard header 586	
standard header page 526	X
STPSPPR (macro) 435	Xprint
STRSPPR (macro) 440 structure layouts 13	add/modify/delete font 616
SUPSPPR (macro) 460	add/modify/delete form 616
system exit	configure/modify/delete object 616
for header page (091) 551	create /use resource 616
for trailer page (092) 551	getting to know the architecture 616 messages 616
system header pages 525	operate 616
_	supported printers 616
T	Xprint see printer
target group of the manual 4	•

Contents

1.1

1.2

1.3	Summary of contents
1.4	Changes to the macros and exits since the last edition
1.5	Notational conventions
1.6	README file
2	Macros for outputting files 7
	OLTPRQX - define UTM print parameters
	PRNTDOC (for SPOOL, SPS, RSO and Dprint) - output file
	Controlling SPOOLOUT
	Controlling paper feed
	Incompatible operands in the PRNTDOC macro
	Return codes
	Parameter list description and assembly language example
2.1	The C interface for the functions of the PRNTDOC macro
2.2	The COBOL interface for the functions of the PRNTDOC macro
2.3	Support macros for PRNTDOC
	PRNTRSO - define RSO parameters for PRNTDOC macro
	PRNTDPC - define Dprint parameters for PRNTDOC macro
	PRNTSPS - define SPS-APA parameters for PRNTDOC macro
	SNPPRNT - define notification parameters for PRNTDOC
	SSVBARC - translate data for barcodes
3	Macros for outputting information
	SPSINF - request information
4	Macros for managing print jobs
	CANPJOB - cancel print job

SHOPJST - show print job status

291

Preface Brief product description of the Spool & Print services

Target group of the manual

Contents

MODJRSO - define RSO parameters for MODPJAT macro MPJADPC - define Dprint parameters for MODPJAT macro MPJASPS - define SPS parameters for MODPJAT macro 0 Uptut structures of the SHOPJAT macro SJAODOF - output structure for INFO=*DOCUMENT_FORMAT SJAODOP - output structure for INFO=*DOCUMENT_PART SJAOLCT - output structure for INFO=*DOCUMENT_PART SJAOLCT - output structure for INFO=*PRINT_JOB_CONTROL SJAOPJC - output structure for INFO=*PRINT_JOB_CONTROL SJAORSD - output structure for INFO=*PRINT_JOB_CONTROL SJAORSD - output structure for INFO=*TO_PRINTER SPJARSO - RSO output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJST macro 4.3 Support macro for the SHOPJST macro SPJSRSO - define RSO parameters for SHOPJST macro 4.4 Output structures of the SHOPJST macro SJSODES - output structure for DESTINATION SJSODES - output structure for DESTINATION SJSODES - output structure for SPOOL-FILTER SJSODES - output structure for DISTRIBUTED SJSODSS - output structure for DESTINATION and UNIX systems SJSOORI - output structure for DRIGIN and UNIX systems SJSOTRA - output structure for TRAITS SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems STRSPPR - allocate printers SASDSPO - requesting information on active spool devices STPSPPR - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	
MPJASPS - define SPS parameters for MODPJAT macro Output structures of the SHOPJAT macro SJAODOF - output structure for INFO=*DOCUMENT_FORMAT SJAOLCT - output structure for INFO=*DOCUMENT_PART SJAOLCT - output structure for INFO=*DOCUMENT_PART SJAOPJC - output structure for INFO=*PRINT_JOB_CONTROL SJAORSD - output structure for INFO=*RESOURCES_DESCRIPTION SJAOTOP - output structure for INFO=*TO_PRINTER SPJARSO - RSO output structure for SHOPJAT macro SPJADPC - Dprint output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJAT macro 4.3 Support macro for the SHOPJST macro SPJSRSO - define RSO parameters for SHOPJST macro 4.4 Output structures of the SHOPJST macro SJSODES - output structure for DESTINATION SJSODES - output structure for RSO-FILTER SJSODES - output structure for RSO-FILTER SJSODIS - output structure for DISTRIBUTED SJSODS - output structure for ORIGIN SJSOORI - output structure for ORIGIN SJSOORS - output structure for TRAITS SJSOTRA - output structure for TRAITS and UNIX systems SJSOTRA - output structure for TRAITS and UNIX systems SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate printer swith RSO-specific parameters SUPSPPR - modify administration parameters for printers	341
4.2 Output structures of the SHOPJAT macro SJAODOF - output structure for INFO=*DOCUMENT_FORMAT SJAODOP - output structure for INFO=*DOCUMENT_PART SJAOLCT - output structure for INFO=*LAYOUT_CONTROL SJAORSD - output structure for INFO=*PRINT_JOB_CONTROL SJAORSD - output structure for INFO=*PRINT_JOB_CONTROL SJAORSD - output structure for INFO=*TO_PRINTER SPJARSO - RSO output structure for SHOPJAT macro SPJAPSC - Dprint output structure for SHOPJAT macro SPJAPS - SPS output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJAT macro SPJSRSO - define RSO parameters for SHOPJST macro 4.3 Support macro for the SHOPJST macro SPJSRSO - define RSO parameters for SHOPJST macro 4.4 Output structures of the SHOPJST macro SJSODES - output structure for DESTINATION SJSODES - output structure for PSOOL-FILTER SJSODES - output structure for DISTRIBUTED SJSODS - output structure for ORIGIN SJSOORS - output structure for ORIGIN SJSOORS - output structure for ORIGIN SJSOORS - output structure for TRAITS SJSOTRA - output structure for TRAITS SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS SJSOTRS - output structure for TRAITS SJSOTRS - allocate printers SASDSPO - requesting information on active spool devices STPSPPR - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	348
SJAODOF - output structure for INFO=*DOCUMENT_FORMAT SJAODOP - output structure for INFO=*DOCUMENT_PART SJAOLCT - output structure for INFO=*LAYOUT_CONTROL SJAOPJC - output structure for INFO=*PRINT_JOB_CONTROL SJAORSD - output structure for INFO=*RESOURCES_DESCRIPTION SJAOTOP - output structure for INFO=*TO_PRINTER SPJARSO - RSO output structure for SHOPJAT macro SPJADPC - Dprint output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJAT macro SPJSRSO - define RSO parameters for SHOPJST macro 4.3 Support macro for the SHOPJST macro SPJSRSO - define RSO parameters for SHOPJST macro 4.4 Output structures of the SHOPJST macro SJSODES - output structure for DESTINATION SJSODES - output structure for SPOOL-FILTER SJSODES - output structure for SPOOL-FILTER SJSODES - output structure for DISTRIBUTED SJSODSS - output structure for DISTRIBUTED SJSODSS - output structure for ORIGIN SJSOORS - output structure for ORIGIN SJSOORS - output structure for ORIGIN SJSOTRA - output structure for TRAITS SJSOTRA - output structure for TRAITS SJSOTRA - output structure for TRAITS and UNIX systems SJSOTRS - output structure for TRAITS and UNIX systems SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	350
SJAODOP - output structure for INFO=*DOCUMENT_PART SJAOLCT - output structure for INFO=*LAYOUT_CONTROL SJAOPJC - output structure for INFO=*PRINT_JOB_CONTROL SJAORSD - output structure for INFO=*PRINT_JOB_CONTROL SJAORSD - output structure for INFO=*RESOURCES_DESCRIPTION SJAOTOP - output structure for INFO=*TO_PRINTER SPJARSO - RSO output structure for SHOPJAT macro SPJADPC - Dprint output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJAT macro SPJSRSO - define RSO parameters for SHOPJST macro 4.3 Support macro for the SHOPJST macro SPJSRSO - define RSO parameters for SHOPJST macro Output structures of the SHOPJST macro SJSODES - output structure for DESTINATION SJSODES - output structure for SPOOL-FILTER SJSODES - output structure for SPOOL-FILTER SJSODIS - output structure for DISTRIBUTED SJSODSS - output structure for DESTINATION and UNIX systems SJSOORI - output structure for ORIGIN SJSOORS - output structure for ORIGIN and UNIX systems SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems 5 Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate printer swith RSO-specific parameters SUPSPPR - modify administration parameters for printers	
SJAOLCT - output structure for INFO=*LAYOUT_CONTROL SJAOPJC - output structure for INFO=*PRINT_JOB_CONTROL SJAORSD - output structure for INFO=*RESOURCES_DESCRIPTION SJAOTOP - output structure for INFO=*TO_PRINTER SPJARSO - RSO output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJAT macro SPJSRSO - define RSO parameters for SHOPJST macro SPJSRSO - define RSO parameters for SHOPJST macro 4.4 Output structures of the SHOPJST macro SJSODES - output structure for DESTINATION SJSODES - output structure for SPOOL-FILTER SJSODES - output structure for SPOOL-FILTER SJSODES - output structure for DISTRIBUTED SJSODSS - output structure for DESTINATION and UNIX systems SJSOORI - output structure for ORIGIN SJSOORS - output structure for ORIGIN and UNIX systems SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS SJSOTRS - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems 5 Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	356
SJAOPJC - output structure for INFO=*PRINT_JOB_CONTROL SJAORSD - output structure for INFO=*RESOURCES_DESCRIPTION SJAOTOP - output structure for INFO=*TO_PRINTER SPJARSO - RSO output structure for SHOPJAT macro SPJADPC - Dprint output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJAT macro 4.3 Support macro for the SHOPJST macro SPJSRSO - define RSO parameters for SHOPJST macro 4.4 Output structures of the SHOPJST macro SJSODES - output structure for DESTINATION SJSODES - output structure for SPOOL-FILTER SJSODES - output structure for SPOOL-FILTER SJSODIS - output structure for DISTRIBUTED SJSODSS - output structure for DESTINATION and UNIX systems SJSOORI - output structure for ORIGIN SJSOORS - output structure for ORIGIN and UNIX systems SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems 5 Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	358
SJAORSD - output structure for INFO=*RESOURCES_DESCRIPTION SJAOTOP - output structure for INFO=*TO_PRINTER SPJARSO - RSO output structure for SHOPJAT macro SPJADPC - Dprint output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJAT macro SPJSRSO - define RSO parameters for SHOPJST macro 4.4 Output structures of the SHOPJST macro SJSODES - output structure for DESTINATION SJSODES - output structure for DESTINATION SJSODES - output structure for SPOOL-FILTER SJSODIS - output structure for DISTRIBUTED SJSODS - output structure for DESTINATION and UNIX systems SJSOORI - output structure for ORIGIN SJSOORS - output structure for ORIGIN SJSOORS - output structure for TRAITS SJSOTRA - output structure for TRAITS and UNIX systems SJSOTRS - output structure for TRAITS and UNIX systems STRSPPR - cancel printers SASDSPO - requesting information on active spool devices STPSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	361
SJAOTOP - output structure for INFO=*TO_PRINTER SPJARSO - RSO output structure for SHOPJAT macro SPJADPC - Dprint output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJAT macro 4.3 Support macro for the SHOPJST macro SPJSRSO - define RSO parameters for SHOPJST macro 4.4 Output structures of the SHOPJST macro SJSODES - output structure for DESTINATION SJSODES - output structure for SPOOL-FILTER SJSODES - output structure for RSO-FILTER SJSODIS - output structure for DISTRIBUTED SJSODSS - output structure for DESTINATION and UNIX systems SJSOORI - output structure for ORIGIN SJSOORS - output structure for ORIGIN SJSOORS - output structure for TRAITS SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems 5 Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	364
SPJARSO - RSO output structure for SHOPJAT macro SPJADPC - Dprint output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJAT macro 4.3 Support macro for the SHOPJST macro SPJSRSO - define RSO parameters for SHOPJST macro 4.4 Output structures of the SHOPJST macro SJSODES - output structure for DESTINATION SJSODES - output structure for SPOOL-FILTER SJSODES - output structure for SPOOL-FILTER SJSODIS - output structure for DISTRIBUTED SJSODSS - output structure for DESTINATION and UNIX systems SJSOORI - output structure for ORIGIN SJSOORS - output structure for ORIGIN and UNIX systems SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS SJSOTRS - output structure for TRAITS SJSOTRPPR - allocate printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	365
SPJADPC - Dprint output structure for SHOPJAT macro SPJASPS - SPS output structure for SHOPJAT macro 4.3 Support macro for the SHOPJST macro SPJSRSO - define RSO parameters for SHOPJST macro 4.4 Output structures of the SHOPJST macro SJSODES - output structure for DESTINATION SJSODES - output structure for SPOOL-FILTER SJSODES - output structure for RSO-FILTER SJSODIS - output structure for DISTRIBUTED SJSODSS - output structure for DESTINATION and UNIX systems SJSOORI - output structure for ORIGIN SJSOORS - output structure for ORIGIN and UNIX systems SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems 5 Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	367
SPJASPS - SPS output structure for SHOPJAT macro Support macro for the SHOPJST macro SPJSRSO - define RSO parameters for SHOPJST macro 4.4 Output structures of the SHOPJST macro SJSODES - output structure for DESTINATION SJSODES - output structure for SPOOL-FILTER SJSODES - output structure for RSO-FILTER SJSODIS - output structure for DISTRIBUTED SJSODSS - output structure for DESTINATION and UNIX systems SJSOORI - output structure for ORIGIN SJSOORS - output structure for ORIGIN and UNIX systems SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems 5 Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	368
4.3 Support macro for the SHOPJST macro SPJSRSO - define RSO parameters for SHOPJST macro 4.4 Output structures of the SHOPJST macro SJSODES - output structure for DESTINATION SJSODES - output structure for SPOOL-FILTER SJSODES - output structure for RSO-FILTER SJSODIS - output structure for DISTRIBUTED SJSODSS - output structure for DESTINATION and UNIX systems SJSOORI - output structure for ORIGIN SJSOORS - output structure for ORIGIN and UNIX systems SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems 5 Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	369
SPJSRSO - define RSO parameters for SHOPJST macro 4.4 Output structures of the SHOPJST macro SJSODES - output structure for DESTINATION SJSODES - output structure for SPOOL-FILTER SJSODES - output structure for RSO-FILTER SJSODIS - output structure for DISTRIBUTED SJSODSS - output structure for DESTINATION and UNIX systems SJSOORI - output structure for ORIGIN SJSOORS - output structure for ORIGIN and UNIX systems SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems 5 Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	370
4.4 Output structures of the SHOPJST macro SJSODES - output structure for DESTINATION SJSODES - output structure for SPOOL-FILTER SJSODES - output structure for RSO-FILTER SJSODIS - output structure for DISTRIBUTED SJSODSS - output structure for DESTINATION and UNIX systems SJSOORI - output structure for ORIGIN SJSOORS - output structure for ORIGIN and UNIX systems SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems 5 Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	371
SJSODES - output structure for DESTINATION SJSODES - output structure for SPOOL-FILTER SJSODES - output structure for RSO-FILTER SJSODIS - output structure for DISTRIBUTED SJSODSS - output structure for DESTINATION and UNIX systems SJSOORI - output structure for ORIGIN SJSOORS - output structure for ORIGIN and UNIX systems SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems 5 Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	371
SJSODES - output structure for DESTINATION SJSODES - output structure for SPOOL-FILTER SJSODES - output structure for RSO-FILTER SJSODIS - output structure for DISTRIBUTED SJSODSS - output structure for DESTINATION and UNIX systems SJSOORI - output structure for ORIGIN SJSOORS - output structure for ORIGIN and UNIX systems SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems 5 Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	374
SJSODES - output structure for RSO-FILTER SJSODIS - output structure for DISTRIBUTED SJSODSS - output structure for DESTINATION and UNIX systems SJSOORI - output structure for ORIGIN SJSOORS - output structure for ORIGIN and UNIX systems SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems 5 Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	
SJSODES - output structure for RSO-FILTER SJSODIS - output structure for DISTRIBUTED SJSODSS - output structure for DESTINATION and UNIX systems SJSOORI - output structure for ORIGIN SJSOORS - output structure for ORIGIN and UNIX systems SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems 5 Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	378
SJSODIS - output structure for DISTRIBUTED SJSODSS - output structure for DESTINATION and UNIX systems SJSOORI - output structure for ORIGIN SJSOORS - output structure for ORIGIN and UNIX systems SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems 5 Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	
SJSOORI - output structure for ORIGIN SJSOORS - output structure for ORIGIN and UNIX systems SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems 5 Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	
SJSOORS - output structure for ORIGIN and UNIX systems SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	386
SJSOTRA - output structure for TRAITS SJSOTRS - output structure for TRAITS and UNIX systems Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	388
SJSOTRS - output structure for TRAITS and UNIX systems Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	390
SJSOTRS - output structure for TRAITS and UNIX systems Macros for managing printers SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	
SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	
SASDSPO - requesting information on active spool devices STPSPPR - cancel printer allocation STRSPPR - allocate print jobs to printers STRSRSO - allocate printers with RSO-specific parameters SUPSPPR - modify administration parameters for printers	395
STRSPPR - allocate print jobs to printers	396
STRSRSO - allocate printers with RSO-specific parameters	435
SUPSPPR - modify administration parameters for printers	440
	457
6 Macros for virtual printers	460
	481
6.1 Use of virtual printers in synchronous operation	
6.2 Program interface for synchronous operation	484
SPVDINI - initialize dialog with virtual printer	485
SPVDRCV - set application task to ready	
SPVDRET - send information on current job to virtual printer	
SPVDEND - release dialog connection	

607

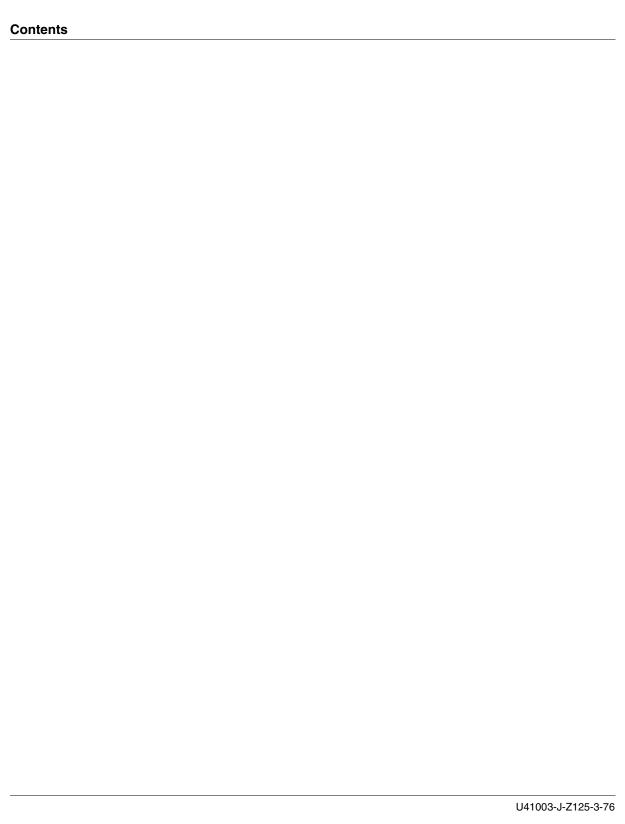
0.3	Communication between virtual printer and application493	
6.3.1	Dialog initialization	
6.3.2	Processing print jobs	
6.3.3	Stopping processing	
6.3.4	Error recovery	
6.3.5	Restarting after errors	
6.4	Example: Communication with a virtual printer in synchronous operation 498	
7	SPOOL exits	
7.1	General information on SPOOL exits	
7.2	Spoolout exit (090)	
7.3	Exits for the header and trailer pages (091/092)	
7.4	Exit for customer filters (093)	
7.5	Resource routine (094)	
7.6	Output routine (095)	
7.7	Exit for server selection (096)	
7.8	Exit for controlling file transfer for Dprint requests (097)	
7.9	Support commands for exit 097	
	SHOW-DPRINT-FT-LIMIT	
	MODIFY-DPRINT-FT-LIMIT	
	ADD-DPRINT-FT-LIMIT	
	REMOVE-DPRINT-FT-LIMIT	
_		
8	Appendix	
8.1	Macro syntax	
	Macro format	
	"New" metasyntax used for the macros	
	"Old" metasyntax used for the macros	
	Macro types	
	Standard header 586	
	Wildcards	
8.2	Functional principle of the exit mechanism 589	
8.3	Trailer page for the local SPOOL	
8.4	Spoolout accounting record (SPLO)	
8.5	Application rules for C	
8.5.1	General rules	
8.5.2	STRSPPR 607	

SUPSPPR

Related publications 609

Index 619

8.5.3



Spool & Print - Macros and Exits (BS2000/OSD)

User Guide

Target group

The manual is intended for programmers who wish to address the Spool & Print Services in their programs directly.

Contents

The manual describes the macros and exits of the Spool & Print Services, including the macros for virtual printers. The description of the macros is arranged according to functions.

Edition: May 2005

File: SPMAKROS.PDF

State of description:

- Distributed Print Services V1.1B
- RSO V3.4A
- SPCONV V1.0A
- SPOOL V4.6
- SPS V3.8
- SPSERVE V2.8

Copyright © Fujitsu Siemens Computers GmbH, 2005.

All rights reserved.

Delivery subject to availability; right of technical modifications reserved.

All hardware and software names used are trademarks of their respective manufacturers.

This manual was produced by cognitas. Gesellschaft für Technik-Dokumentation mbH www.cognitas.de

User Documentation 81730 Munich Germany Fax: (++49) 700 / 372 00001

Fujitsu Siemens computers GmbH

Corrections

Comments

Suggestions

http://manuals.fujitsu-siemens.com

Submitted by

manuals@fujitsu-siemens.com

Comments on Spool & Print



Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format ...@ts.fujitsu.com.

The Internet pages of Fujitsu Technology Solutions are available at http://ts.fujitsu.com/...

and the user documentation at http://manuals.ts.fujitsu.com.

Copyright Fujitsu Technology Solutions, 2009

Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form ...@ts.fujitsu.com.

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter http://de.ts.fujitsu.com/..., und unter http://manuals.ts.fujitsu.com finden Sie die Benutzerdokumentation.

Copyright Fujitsu Technology Solutions, 2009