SPS/MVS CIS V4.0

User's Guide

Beta Edition
13-Nov-00
Preface

This document is intended to provide information to the SPS user about the SPS/MVS product. It contains information that should help application programmers to develop powerful AFP applications to be processed by one or more of the SPS/MVS components.

It is assumed that the reader is familiar with the MVS environment and with Advanced Function Presentation (AFP) concepts.
Contents

1 SPS/MVS INTRODUCTION ............................................................................................................. 7
  1.1 DESCRIPTION ............................................................................................................................... 7
  1.2 BENEFITS ..................................................................................................................................... 8
  1.3 OPERATING SYSTEM REQUIREMENTS ...................................................................................... 11
  1.4 FEATURES SUMMARY .................................................................................................................. 11
    1.4.1 SPS/MVS-CIS ............................................................................................................... 11
    1.4.3 SPS/MVS-APA ............................................................................................................... 13
    1.4.4 SPS/MVS-LIP ............................................................................................................... 17

2 AUTOMATED DOCUMENT PROCESSING WITH SPS/MVS ....................................................... 21
  2.1 SPS/MVS-CIS FUNCTIONAL OVERVIEW ...................................................................................... 22
  2.3 SPS/MVS-APA FUNCTIONAL OVERVIEW ..................................................................................... 23
  2.4 SPS/MVS-LIP FUNCTIONAL OVERVIEW ....................................................................................... 25

3 SPS/MVS-CIS: CONVERTING, INDEXING AND SORTING PRINT DATA ................................... 27
  3.1 OVERVIEW .................................................................................................................................. 27
  3.2 UNDERSTANDING SPS/MVS-CIS CONVERSION AND INDEXING PROCESS ......................... 31
    3.2.1 Converting and Indexing print files containing S/370 Line formatted data ....................... 31
    3.2.2 Converting and Indexing print files containing AFP Mixed data ....................................... 34
    3.2.3 Converting and Indexing print files containing Unformatted ASCII data ......................... 34
    3.2.4 Normalizing print files containing AFP and MO:DCA-P data ............................................ 34
  3.3 USING SPS/MVS-CIS TO RETRIEVE AFP RESOURCE FILES ................................................ 37
  3.4 USING SPS/MVS-CIS TO GENERATE THE AFP INDEX OBJECT FILE ................................... 39
  3.5 USING SPS/MVS-CIS TO SORT AND REORDER OUTPUT PAGES ............................................. 42
    3.5.1 Contents sorting .......................................................................................................... 43
    3.5.2 Layout sorting ............................................................................................................ 47
  3.6 USING SPS/MVS-CIS TO CONSOLIDATE MAIL PIECES ........................................................... 55
  3.7 SPS/MVS-CIS JOB CONTROL REQUIREMENTS ................................................................. 57
  3.8 SPS/MVS-CIS PARAMETER FILE REFERENCE ........................................................................ 62
  3.9 SPS/MVS-CIS PERFORMANCE CONSIDERATIONS ............................................................... 80
    3.9.1 Memory requirements ....................................................................................................... 80
    DASD requirements .................................................................................................................. 80
    3.9.3 CPU requirements .......................................................................................................... 80

4 GLOSSARY ....................................................................................................................................... 81

5 BIBLIOGRAPHY ............................................................................................................................. 85
1 SPS/MVS Introduction

1.1 Description

SPS/MVS\(^1\) is a software product member of the PRISMAprofessional product family of Océ Printing Systems, which includes support for print applications, development tools, data stream transformers and high speed printer drivers under several platforms such as OS/390, VSE, BS2000, UNIX and Windows-NT. SPS/MVS is a key component of this family, providing powerful and sophisticated technologies that help a large number of high-volume production and mailing installations to satisfy their demands on faster and more flexible tools.

\(^1\) Smart Print Server for OS/390

---

Fig. 1. Automated Document Processing
SPS/MVS is a generic name used to refer to the following MVS products:

- SPS/MVS-CIS
- SPS/MVS-APA
- SPS/MVS-LIP

**SPS/MVS-CIS** is a high performance data manipulation tool that can be used to convert, index, sort and consolidate existing S/370 Line data and AFP print files, preparing the information for subsequent archiving, online browsing, network distribution or high performance printing. The sorting capabilities offered by SPS/MVS-CIS may substantially reduce mailing and other post-processing costs. The resource packaging function allows an installation to archive not just the print data, but also the resources it requires, enabling exact document reproduction even years later.

**SPS/MVS-APA** is a high performance printing subsystem that drives Océ and other IPDS compatible single and twin printers under MVS/SP2 (XA), MVS/SP3-SP4-SP5 (ESA), OS/390, and subsequent MVS systems. The SPS/MVS-APA driver accepts AFPDS and line data streams as input. It converts these input types (may be intermixed) to an IPDS data stream. The input data stream may be either JES spooled output or direct printing output from a program. Optionally library resource processing specifications may be overridden and special hardware features may be selected by control file requests.

**SPS/MVS-LIP** is the printer subsystem that drives Océ LIP printers under MVS/SP1 (SP), MVS/SP2 (XA), or MVS/SP3-SP4 (ESA), and subsequent MVS releases. As in the case of SPS/MVS-APA, the input data stream may be either JES spooled output or direct printing output from a program.

### 1.2 Benefits

Depending on the product features you are planning to use, SPS/MVS may help your installation to:

- **SPS/MVS-CIS**  
  Convert S/370 line data or mixed data from legacy applications into MO:DCA-P. The conversion process resolves all line data
SPS/MVS Introduction

relevant questions (i.e. conditional processing) and creates an AFP output file that is ready for online viewing, archiving, printing and/or downloading. Several AFP capable viewers are available in the market and may be used for this purpose.

- Extract variable text strings -part of the line data processed- and use them to build Index Tags. Indexing a print job enhances the ability to view, archive or just retrieve specific pages or group of pages contained in the file. With SPS/MVS-CIS, the index information can also be used to reorder the print job pages (i.e. by Postal code) and/or to consolidate several different print files into a single one.

- Create an Index file based on the Index Tags contained (or inserted) in the input file. The index file created by SPS/MVS-CIS may contain tags for page groups, pages, or both.

- Create a sequential file or a Partitioned Data Set with the resources required to print the application. Archiving the resources together with the print file guarantees an exact document reproduction even years later.

- Sort the input pages using a user specified criteria. The sorting criteria can be based on the Index information (i.e. post code, customer name, etc.) or on pre-defined page re-ordering layouts such as Booklet.

- A powerful parameter interface allows an installation to use one or more of SPS/MVS-CIS functions in a single run. Activating several functions at once simplifies job processing and reduces the overall elapsed time, causing SPS/MVS-CIS to convert, index, retrieve resources and sort a print file in one step. Activating only specific options allows an application programmer to tailor the SPS/MVS-CIS processing with special requirements, eliminating wasted run-time.

SPS/MVS-APA

- It is a robust, optimized driver for high performance IPDS printers. SPS/MVS-APA is designed to drive IPDS printers in all performance classes, from departmental printers (network attached, 20 ppm) up to high speed enterprise printers (Escon channel attached, 1000 ppm and more).

- Full error recovery and an exhaustive operator interface are also part of SPS/MVS-APA. When required, these error recovery facilities take into consideration the device
characteristics and advanced features of Océ printers.

- Uses the FSS interface available under JES to act as a normal JES printer. Job selection and command processing are done using JES' standard interfaces so that the MVS operators are able to work with SPS from the first day on.

- SPS/MVS-APA customers take advantage of the Forms Table concept to simplify JCL coding and to add flexibility to their printing process. Using this facility, the SPS administrator can establish a variety of predefined print formatting models to handle the standard print jobs in an installation. These predefined print characteristics are selected by form names associated with each of the models. This facility is used by providing a list of form names with associated output formats to end users who then only need to select the appropriate form name to generate the desired output format.

- The direct printing facility makes it possible for an application program to directly drive single and twin APA printers without spooling the output through JES. Multiple printers may be driven by an application in the same address space if required.

- The information pages facility (header, separator, message, and trailer pages) provide the capability of generating uniquely designed information pages for specific print models (forms). On cut sheet printers this provides clear output separation in all situations, allowing for offset stacking, colored separator forms and other features. In all situations it provides the capability of matching information page sizes and formats with special forms requirements.

- SPS/MVS-APA provides a user control file installation option which permits the identification of users or user groups who are permitted to provide their own predefined print models. These users have total flexibility in their processing options. They can build a set of personalized print models to handle their unique print job requirements. This option requires the end user have a knowledge of how to use and code print control files. This level of understanding requires very little time to master. Selected users or user groups may be given this authority as desired. This option may be used with the predefined forms option (forms control file).

- Full network printer support permits remote printers to be attached via SNA (LU1, LU 6.2) or TCP/IP protocols.

- Channel attached printers may use Escon channels as well as
SPS/MVS Introduction

S/370 channels. Most Channel extender devices are supported too.

SPS/MVS-LIP

- Is the Océ driver for LIP printers.
- Uses the FSS interface available under JES to act as a normal JES printer. Job selection and command processing are done using JES’ standard interfaces so that the MVS operators are able to work with SPS from the first day on.

1.3 Operating System Requirements

This version of SPS/MVS will run without restrictions on MVS systems at the release levels listed below. Any specifically required PTF’s are documented by the ++IF REQ statements of the supplied installation deck.

- MVS/XA Version 2.1.3 and above
- MVS/ESA Version 3, 4 and 5
- OS/390 Version 1 and 2

1.4 Features Summary

1.4.1 SPS/MVS-CIS

- Data processing
**Input Data Stream Processing:** Accept the following input data formats for conversion, indexing and sorting:

- S/370 Line format data, including Mixed-mode line data
- AFP and MO:DCA-P data
- SPDS data that conforms to the *SPDS Data Stream Reference Manual*.
- Unformatted ASCII data as defined in this manual. ANSI carriage controls using ASCII coding are supported.

**Resource Data Stream Processing:** These resources may be:

- Form definitions (FORMDEF)
- Page definitions (PAGEDEF)
- Page segments
- Overlays
- Fonts (Coded Fonts, Font characters sets, Code pages)
- Object Containers (Color Mapping table)

Only the resources required by an application are read and copied into the output resource data set using filtering parameters specified by the user. Storing the print document, together with the required resources, guarantees exact reprints on any platform where MO:DCA-P is supported even years later.

**Data conversion and normalization:** All data read is converted into MO:DCA-P structured fields. All S/370 line data features (i.e. conditional processing) and OPS extensions (i.e. page numbering) are fully processed so that the generated AFP data stream no longer requires a Pagedef in order to be printed.

**Index Tag insertion:** SPS/MVS-CIS inserts Index Tags into the MO:DCA-P data generated during the conversion process. Flexible data search and indexing rules allow a user to handle even the most complicated line data applications.

**Index file creation:** An Index Object File is created with the Index tags inserted (or already contained) in the input data stream. The Index Object file can be used by any MO:DCA-P capable browser to facilitate online document navigation.

**Sorting function:** SPS/MVS-CIS provides powerful sorting algorithms that allow an installation to sort an input print file based on the document contents. This unique
feature combined with the Layout sorting algorithms help reducing mailing and post-processing time and cost.

- **Document consolidation**: Consolidating several indexed documents is not a problem. In combination with the sorting function, an installation may merge two or more print files into a single one, consolidating the mail pieces addressed to a single destination.

- **Reliability, Availability, Serviceability (RAS)**
  
  - **Dumping**: Diagnostic dumps are automatically taken in the event of program or unexplained errors.
  
  - **Tracing**: A trace facility is available which permits the collection of information on all processing phases. The level of detail of the trace information collected may be selected by the installation and varies from flow trace to all control blocks, input and output data.
  
  - **Error information**: The AFP normalizer built into SPS/MVS-CIS does a comprehensive validation of the input data being processed. Extensive error information is provided to the user in the situations where the input data does not conform to the MO:DCA-P standards.

1.4.3 **SPS/MVS-APA**

SPS processing support for APA single and twin printers can be divided into three major categories: user input, installation specifications, and RAS facilities. The main elements of these functions are listed in the following sections.

- **User Input**

  - **Input Data Stream Processing**: Accepts from the spooling system or a user program any user generated data stream that conforms to the *SPDS Data Stream Reference Manual*. These data streams may include line data, composed text data, and any resource type.
• **Resource Data Stream Processing**: Reads from libraries any input data stream or JCL referenced resources. These resources may be:
  - Form definitions (FORMDEF)
  - Page definitions (PAGEDEF)
  - Page segments
  - Overlays
  - Fonts (Coded Fonts, Font characters sets, Code pages)
  - Object Containers (Color Mapping table)

• **Control File Support (extended JCL)**: SPS reads control statements from control files permitting the selection of features and control of individual print job processing through overriding FORMDEF and PAGEDEF specifications that is not possible through normal JCL. The following functions are currently supported for APA single and twin printers:
  - Selection of Simplex, Duplex normal and Duplex tumble
  - Selection of Portrait and Landscape presentations, with 0°, 90° and 180° rotation
  - Selection of input and output bins
  - Two up printing
  - Listing of active file options
  - Copies with from/to pages
  - Selection of eject to front facing function
  - Flip print on twin printers
  - Enhanced double byte character set usage
  - 3800 compatible media origin
  - Selection of presentation
  - Font pruning
  - Input bin substitution
  - Selection of chars, formdef, pagedef
  - Spooled message page

• **Job Control Language (JCL)**: SPS accepts all page printer related JCL
• **Message Generation:** SPS generates operator and user related messages concerning all phases of data stream processing. User messages are printed at the end of the user output or may be spun off to a spool file.

• **Checkpointing:** Processing checkpoints may be requested by the user or installation on a time or page count interval basis. This facility permits the restart of a data stream interrupted by operator, user error or hardware error.

• **Installation Specifications**

  • **System parameters:** SPS supports a variety of initialization parameters and processing defaults. All options specified under control file support above can be specified by an installation for a specific form or output class. The active setup and file parameters can be seen via the SDIS and FDIS SPS commands.

  • **Information Pages:** Flexible, easily defined installation header, trailer, data set separator and message pages. Different information page formats may be defined for each printer, output class or form name. PSF compatible user exits for information pages are also supported.

  • **Predefined Form Processing:** Any form can have installation defined print characteristics that will be used automatically when the form name is requested.

    Setup processing dependant on the number of input bins from a cutsheet printer is done.

  • **Accounting:** SPS generates the standard SMF type 6 accounting record for all input data streams processed. If MVS is not able to write an SMF record SPS informs the operator, allowing retry or ignore actions. Even when SPS abends or the MVS system crashes SPS writes SMF records for the printing files (either at clean-up or at reselection time) assuring a 100% reliable accounting system for the printing jobs.

  • **User exits:** The following exit points are defined:

    | Record | Modify, delete or insert records to be printed in the input data stream. Additional support for repositioning and paper jams is included in the input record exit interface. Each printer may have its own unique record processing exit. |
### SPS/MVS Introduction

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accounting</strong></td>
<td>Modify SMF type 6 accounting records generated by SPS. Each printer may have its own unique accounting exit.</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Perform special processing when any given message is issued. A different exit may be defined for each SPS message. Each printer may have its own unique set of message exits.</td>
</tr>
<tr>
<td><strong>Direct</strong></td>
<td>Direct printing compatibility exit. Provides full compatibility with IBM's direct printing exit interface and processing.</td>
</tr>
<tr>
<td><strong>D/S Allocation</strong></td>
<td>May change the output file characteristics or request SPS to purge or hold a file without printing it. Each printer may have its own unique data-set allocation exit.</td>
</tr>
<tr>
<td><strong>Resource read</strong></td>
<td>Modify, delete or insert resources read from system or user libraries.</td>
</tr>
<tr>
<td><strong>Page Segment</strong></td>
<td>Allows SPS to continue processing if a page segment missing condition is reached.</td>
</tr>
<tr>
<td><strong>Compatibility exits</strong></td>
<td>Full compatible PSF exit interface for exits 1, 2 and 3 is provided. A subset of PSF exit interface 7 is provided.</td>
</tr>
</tbody>
</table>

- **Reliability, Availability, Serviceability (RAS)**
  - **Dumping**: Diagnostic dumps are automatically taken in the event of program or unexplained errors. For direct printing applications, SPS must be RACF authorized in order to be able to write SYSUDUMPs.
  - **Tracing**: A trace facility is available which permits the collection of information on all phases of SPS processing. The level of detail of the trace information collected may be selected by the installation and varies from flow trace to all control blocks, input and output data.
  - **Error recovery**: SPS provides full error recovery for APA printers and guarantees that print data sets are not purged from the JES spool unless successfully printed.
  - **EREP**: In the event of a printer hardware error, SPS records an outboard type record in SYS1.LOGREC for analysis by EREP.
1.4.4 SPS/MVS-LIP

SPS processing support for LIP printers can be divided into three major categories: user input, installation specifications, and RAS facilities. The main elements of these functions are listed in the following sections.

- **User Input**

  - **Input Data Stream Processing:** Accepts from the spooling system any user generated data stream that conforms to the 2050/2075-1x and –9x Electronic Printing Systems, System Applications, Reference Manual. These data streams may include line data, LIP controls, 3211 carriage controls, and trc’s. All functions of either FCB2 or FCB3 may be used to control printing of data streams.

  - **Control File Support (extended JCL):** SPS reads control statements from control files permitting the selection of features and control of individual print job processing that are not possible through normal JCL. The following functions are currently supported for LIP printers:
    - Identify an LCL to be used
    - Selection of Simplex, Duplex, Tumble
    - Selection of Input and Output bins
    - Media selection (supported standard forms)
    - Right shifting of printout
    - Selection of overlays
    - Selection of initial font
    - Selection of embedded text control or line mode
    - SNI or IBM table reference characters (TRC’s)
    - Rotation of input page
    - Blocking of character, page position and FCB length checks
    - Selection of LCL contained print controls

  - **Job Control Language (JCL):** SPS processes the following printer related fields of the DD and OUTPUT statements:
SPS/MVS Introduction

- CKPTPAGE and CKPTSEC
- CLASS
- CONTROL
- COPIES
- DATACK
- DCB
- FCB
- FORMS
- PIMSG
- SYSOUT
- TRC

- **Message Generation:** SPS generates operator and user related messages concerning all phases of data stream processing. User messages are printed at the end of the user output or may be spun off to a spool file.

- **Checkpointing:** Processing checkpoints may be requested by the user or installation on a time or page count interval basis. This facility permits the restart of a data stream interrupted by operator, user error, or hardware error.

- **Installation Specifications**

  - **System parameters:** SPS supports a variety of initialization parameters and processing defaults. All options specified under control file support above can be specified by an installation for a specific form or output class. The active setup and file parameters can be seen via the SDIS and FDIS SPS commands.

  - **Information Pages:** Flexible, easily defined installation header, trailer, data set separator, and message pages. Different information page formats may be defined for each printer, output class or form name.

  - **Predefined Form Processing:** Any form can have installation defined print characteristics that will be used automatically when the form name is requested.
• **Accounting:** SPS generates the standard SMF type 6 accounting record for all input data streams processed. If MVS is not able to write an SMF record SPS informs the operator, allowing retry or ignore actions. Even when SPS abends or the MVS system crashes SPS writes SMF records for the printing files (either at clean-up or at reselection time) assuring a 100% reliable accounting system for the printing jobs.

• **User exits:** The following exit points are defined:

  - **Record**
    Modify, delete or insert records to be printed in the input data stream. Additional support for repositioning and paper jams is included in the input record exit interface. Each printer may have its own unique record processing exit.

  - **Accounting**
    Modify SMF type 6 accounting records generated by SPS. Each printer may have its own unique accounting exit.

  - **Message**
    Perform special processing when any given message is issued. A different exit may be defined for each SPS message. Each printer may have its own unique set of message exits.

• **Reliability, Availability, Serviceability (RAS)**

  - **Dumping:** Diagnostic dumps are automatically taken in the event of program or unexplained errors. For direct printing applications, SPS must be RACF authorized in order to be able to write SYSUDUMPs.

  - **Tracing:** An SPS trace facility is defined which permits the collection of information on all phases of SPS processing. The level of detail of the trace information collected may be selected by the installation and varies from flow trace to all control blocks, input and output data.

  - **Error recovery:** SPS provides full error recovery for LIP printers and guarantees that print data sets are not purged from the JES spool unless successfully printed.

  - **EREP:** In the event of a printer hardware error, SPS records an outboard type record in SYS1.LOGREC for analysis by EREP.
2 Automated Document Processing with SPS/MVS

During the 80's we have seen how the simple document printing process done in a data center evolved into a "mailroom" production environment. High speed printing devices and huge customer databases created an impressive demand on printed material. The evolution continued in the 90's and new concepts, such as the "Automated Document Factory" developed as visions.

But the business did not just grow in volume. Also the quality expected by the end users rose to new levels. From simple, tabular designed applications; invoices, bank statements and telephone bills, to mention just a few, become sophisticated ways of presenting the Corporate identity of an organization to their customers.

Fig. 2. SPS/MVS Overview
Implementing new format and document presentation strategies can be very difficult, specially when programming resources are scarce, program change cycles are long, and legacy applications and third party software cannot be changed. Even in the case where application changes are possible, they are not desirable as implementing such specialized formatting does not only increase the dependencies on the current presentation technology, but causes large amounts of redundant logic.

It is in this area where the powerful and flexible formatting capabilities available in SPS/MVS help application developers to optimize their development, leaving the formatting aspects up to SPS/MVS.

2.1 SPS/MVS-CIS functional overview

SPS/MVS-CIS is a sophisticated print stream processing and application development tool. It allows you to:

- Convert and normalize the output of existing legacy applications without having to change the applications program. In combination with its advanced Indexing feature, the SPS/MVS-CIS data converter is capable of building multilevel AFP Page Group structures.
- Generate and insert Index Tags based on data obtained during the conversion process.
- Generate an Index Object File using the Index Tags inserted (or already available) in the print file.
- Retrieve all the AFP resources used by the application and to copy them into a separate Resource data set (either sequential or partitioned).
- Sort the print file pages using either a contents oriented (index) or layout oriented algorithm.

Each of these SPS/MVS-CIS functions may be invoked separately or together in a single program invocation, adding flexibility to your applications and optimizing processing time. Fig. 3 shows the different tasks you can accomplish with SPS/MVS-CIS.
2.2 SPS/MVS-APA functional overview

SPS/MVS-APA is the printer subsystem that drives IPDS single and twin printers supporting the SNIPDS data stream (a data stream fully compatible with the IPDS data stream) under MVS.

SPS/MVS-APA, as distributed, includes the base SPS product and one driver supporting APA printers. Multiple executions of this driver may function concurrently under the same address space or in several address spaces within the system. The SPS/MVS-APA driver accepts AFP, MO:DCA-P and S/370 Line format data as input. It converts these input...
types (may be intermixed) to an SNIPDS data stream. The input data stream may be either JES spooled output or direct printing output from an application program. Optionally library resource processing specifications may be overridden and special hardware features may be selected using a powerful control file facility.

Fig. 4. SPS/MVS-APA Overview

APA printers use resources to control the formatting of input print data streams. These resources are read from libraries before the data stream is processed. SPS/MVS-APA provides the capability of controlling, adding to, or overriding resource processing from any combination of the following sources:
• Installation defined processing associated with form names
• User specified parameters in the Job Control (JCL)
• User specified parameters in the Control file
• Operator commands (JES2 or JES3)

Fig. 4 shows the different data sources you can process with SPS/MVS-APA.

2.4 SPS/MVS-LIP functional overview

SPS/MVS-LIP is the printer subsystem that drives Océ LIP printers under MVS. SPS/MVS-LIP, as distributed, includes one LIP driver module supporting locally attached LIP printers (i.e. 2050 and 2075). Multiple executions of this driver may function concurrently under the same address space or in several address spaces within the system.

The SPS/MVS-LIP driver accepts line mode input for LIP printers from JES2 and JES3 spool data sets generated to sysout and converts them to LIP compatible data streams. Optionally special text control characters controlling advanced functions may be embedded in the print data stream.
3 SPS/MVS-CIS: Converting, Indexing and Sorting print data

3.1 Overview

SPS/MVS-CIS is a high performance data manipulation tool that can be used to convert, index, sort and consolidate large volume print files, preparing the information for subsequent archiving, online browsing, network distribution or high speed printing. Installations may use the advanced indexing techniques available in SPS/MVS-CIS to transform existing legacy applications into hierarchically structured documents. The sorting capabilities offered by SPS/MVS-CIS may substantially reduce mailing and other post-processing costs. The resource packaging function it provides, allows an installation to archive not just the print documents but also the resources they use, enabling exact document reproduction even years later.

One single SPS/MVS-CIS run may invoke one or more of these sophisticated print stream conversion and processing features resulting in additional flexibility and reduced overhead as new applications are built. SPS/MVS-CIS allows installations to:

- Convert and normalize S/370 Line format data or mixed data generated by existing legacy applications into MO:DCA-P format without having to change the application programs. The conversion process resolves all line data relevant issues (i.e. conditional processing) and creates a print output file that is ready for online viewing, archiving, printing and/or downloading.

  Converted documents become platform independent and can be easily moved across different system platforms. SPS/MVS-CIS guarantees a conversion process that is identical to the process done by SPS/MVS-APA when S/370 Line format data is directly printed.

- Extract variable text included in the S/370 Line format data and use it to build Index Tags. Indexing a print job enhances the ability to view, archive or just retrieve specific pages or group of pages contained in the file. The powerful parameter interface available in SPS/MVS-CIS allows an installation to specify an unlimited number of indexing tags to be obtained from the print data.
The advanced indexing features available in SPS/MVS-CIS are capable of converting simple, “flat” S/370 Line format data files into hierarchically structured, multi-level, AFP Documents. A structured document is the base for any subsequent print stream processing to be done with SPS/MVS-CIS including functions such as data enrichment, sorting, post processing optimization, output segmentation and others.

- Generate an Index Object File using the Index Tags inserted during the conversion process. S/370 line data applications get their tags inserted by SPS/MVS-CIS. Input print files in MO:DCA-P format may include Index Tags that were inserted by the application program generating the file or by SPS/MVS-CIS in a previous run. Regardless of their origin, SPS/MVS-CIS uses these tags to generate the Index Object File. The index file may contain tags for page groups, pages or both.

- Retrieve the AFP resources used by an application and copy them into a separate Resource data set (sequential or partitioned). SPS/MVS-CIS not just identifies the resources required by a print job, but locates them, validates their contents and copies the data into the Resource file(s). Only the resources used by the print file get copied. Inline resources already present in the input data stream are also supported. Superfluous resources (if present) are removed. Missing resources (if any) are located in the Resource libraries and are copied into the output Resource file(s).

- Sort the pages, sheets or documents in the input print file using either a "contents based" or “layout based" sorting algorithm. Contents sorting uses the index values of pages or page groups to determine the new print file sequence. Layout sorting regroups the page or sheet sequence to one of several predefined layouts. Sorting a print file can save significant post-processing time, reducing postage and mailing costs and eliminating manual sorting. The SPS/MVS-CIS sort function may also be used to consolidate (merge) several print files, combining two or more documents, even from different applications, into a single mail piece.

In summary, there are many reasons for using SPS/MVS-CIS. Whether the goal is to archive a document and the associated resources, to remove platform dependencies and print a file in a distributed environment, to enhance the document's presentation or to optimize the post-processing work, SPS/MVS-CIS is a robust tool designed to achieve excellent performance, tuned to process large volumes of documents as quickly as possible. It is built for the tight processing window allowed in most installations. And all of these without having to change the original application.
SPS/MVS-CIS: Converting, Indexing and Sorting print data

Fig. 5. SPS/MVS-CIS application workflow
3.2
Understanding SPS/MVS-CIS conversion and indexing process

SPS/MVS-CIS may be used to convert any of the following print data stream formats into MO:DCA-P:

- S/370 Line format data, including Mixed-mode and Double-byte encoding
- SPDS data that conforms to the *SPDS Data Stream Reference Manual*.
- Unformatted ASCII data as described in this manual. ANSI carriage controls using ASCII coding are also supported.

In addition, SPS/MVS-CIS may be used to normalize existing AFP and MO:DCA-P applications.

The first part in the SPS/MVS-CIS processing is the Conversion and Normalization step (see Fig. 3). The Conversion process is required for data which is not in AFP format (i.e. S/370 Line format data). In this case, SPS/MVS-CIS converts the data into MO:DCA-P using the specifications contained in the Page definition. The conversion process is mandatory for S/370 Line data as this format is not supported as an output data stream. During the conversion process, SPS/MVS-CIS may be requested to scan the input data and generate Index tags. These Index tags are later used in the Index Object file generation and during contents sorting.

For data which is already in AFP format, SPS/MVS-CIS normalizes the data removing obsolete AFP definitions and some printer resolution dependencies. The normalization process also verifies the adherence of the print data to the MO:DCA-P rules, detecting inconsistencies between the print file and its resources. Due to its irregular format, AFP data cannot be Indexed by SPS/MVS-CIS. However, SPS/MVS-CIS uses Index tags already contained in such AFP data to generate an Index Object file and during contents sorting.

### 3.2.1 Converting and Indexing print files containing S/370 Line formatted data

S/370 line format data consists of records of text data, in a tabular format, mostly in EBCDIC coding, that usually begin with a carriage control character (CC) containing vertical line spacing commands (refer to *Fig. 7 Legacy application* on page 33). An optional table reference character (TRC) may be used to control font selection.

SPS/MVS-CIS uses a Page Definition resource (PAGEDEF) to convert the S/370 line data into MO:DCA-P. The conversion rules used are the same rules used by all AFP drivers (i.e. SPS/MVS-APA, PSF, etc.) so that the application's appearance is left unchanged. The conversion process is done in such a way that all PAGEDEF dependencies are removed, including such as conditional processing, font selection, bar code insertion, field
positioning, format control, etc. In the case of double-byte applications, the Shift Out-Shift In (SOSI) process is also resolved. Applications converted to MO:DCA-P become platform independent and may be transferred across different systems without risking formatting problems. An example showing the formatting capabilities available with SPS/MVS-CIS is shown in Fig. 8 Legacy application: converted document on page 35.

During the conversion process, SPS/MVS-CIS may be requested to scan the input data and to recognize the document's structure using the TRIGGER parameter defined for this purpose. This feature allows the user to specify "trigger" strings to be used to determine the beginning of each new structure within the input print file. Matching a trigger determines the level (depth) within the document hierarchy where the match occurs. The resulting output document may contain nested page-groups to identify each of these structures and levels. Hierarchically structured MO:DCA-P files add new options for sorting and data enrichment.

Example:

Using the application shown in Fig. 7 Legacy application on page 33, the following SPS/MVS-CIS parameters would request the generation of an Index Object file containing the ACCOUNT, NAME, ADDRESS and PLZ (Zip code) fields as indexing items:

```
FORMDEF ( PI_CIS02 )
PAGEDEF ( PI_CIS02 )

CC ( ASA )
TRCTYPE ( IBM )
INDEXOBJ ( ALL )

TRIGGER ( account, *, 1, '10Bank of Yesterday, Inc.', 7, 63, '00001' )

INDEX ( accountI, account, 'ACCOUNT', accountF )
INDEX ( nameI, account, 'NAME', nameF )
INDEX ( addressI, account, 'STREET', addressF )
INDEX ( plzI, account, 'PLZ', plzF )

FIELD ( accountF, 7, 3, 8 )
FIELD ( nameF, 13, 3, 30 )
FIELD ( addressF, 14, 3, 30 )
FIELD ( plzF, 15, 3, 30 )
```

Fig. 6. Legacy application: Indexing parameters
**SPS/MVS-CIS: Converting, Indexing and Sorting print data**

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>29.03.2000</td>
<td>29.03.2000</td>
<td>50020020</td>
</tr>
<tr>
<td>29.03.2000</td>
<td>29.03.2000</td>
<td>001-002</td>
</tr>
<tr>
<td>29.03.2000</td>
<td>29.03.2000</td>
<td>00005</td>
</tr>
</tbody>
</table>

---

Fig. 7. Legacy application
SPS/MVS-CIS: Converting, Indexing and Sorting print data

The Index Object file resulting from these indexing parameters is described later on page 38.

3.2.2 Converting and Indexing print files containing AFP Mixed data

AFP Mixed-mode data is a combination of S/370 Line format and AFP data. In most cases, AFP Mixed-mode data is the result of old legacy applications which where extended to include some AFP structured fields, composed text pages, images, bar codes, graphics, presentation text and other AFP objects. As in the case of S/370 line format data, AFP Mixed-mode data consists of records, in a tabular format, mostly in EBCDIC coding, that usually begin with a carriage control character (CC) containing vertical line spacing commands. A TRC byte may also be present.

SPS/MVS-CIS uses a Page Definition resource (PAGEDEF) to convert the AFP Mixed-mode data into MO:DCA-P. The conversion rules used are the same rules used by all AFP drivers (i.e. SPS/MVS-APA, PSF, etc.) so that the application appearance is left unchanged. The conversion process is done in such a way that all PAGEDEF dependencies are removed, including such as conditional processing, font selection, bar code insertion, format control, etc. In the case of double-byte applications, the Shift Out-Shift In (SOSI) process is also resolved. Applications converted to MO:DCA-P become platform independent and may be transferred across different systems without risking formatting problems.

SPS/MVS-CIS is able to scan and insert index tags on AFP Mixed-mode documents in a way similar to the indexing done for S/370 Line format data. The same parameters and features are available.

3.2.3 Converting and Indexing print files containing Unformatted ASCII data

SPS/MVS-CIS provides support for ANSI Carriage Control specification using ASCII codes, as well as indexing strings in hexadecimal. These two features, combined with ASCII code fonts allow an installation to process print files that originated in a workstation (Unix, Windows, etc.) under MVS. The support for ASCII coded data is very similar to the support provided for S/370 Line format data, with the exception that all string operations (i.e. indexing, conditional processing, etc.) has to be done specifying values coded using the hexadecimal notation.

3.2.4 Normalizing print files containing AFP and MO:DCA-P data

Advanced Function Presentation (AFP) formatted data uses a superset of the MO:DCA-P data stream including objects such as FOCA, GOCA, IOCA, PTOCA and BCBCA.
organized into resources such as fonts, overlays, page segments, form definitions, Color mapping tables and others. The data normalization process SPS/MVS-CIS supports for AFP data includes the following features:

- Conversion of IM images to IOCA to remove resolution dependencies
- Conversion of old fashion coded font resources to MCF-2 format
- Conversion of l-unit-per-unit-base values
- Renaming of Page Group and Page name structures

Fig. 8. Legacy application: converted document
SPS/MVS-CIS: Converting, Indexing and Sorting print data

- Insertion of Invoke Medium Map structured fields before every page group
- Removal / Insertion of x'5A' carriage control

SPS/MVS-CIS requires every AFP input record to contain a x'5A' character as carriage control only when the input DCB specifies that the dataset contains carriage control (i.e. VBA or VBM). This byte is not required for other DCB formats (i.e. VB, V, F, FB). Structured fields cannot span multiple input records.

The normalization process in SPS/MVS-CIS also verifies the adherence of the print file to the AFP standards, detecting and reporting inconsistencies within the print data, or between print data and resources. This feature allows a user to “certify” the quality of the print data generated by an application before it is further processed (i.e. online distributed or archived) increasing the overall reliability of such applications.

A user should note that normalizing an AFP file does not mean that it is printable under all circumstances. For example, missing hardware features could still stop an application from being printed.

3.3
Using SPS/MVS-CIS to Retrieve AFP Resource Files

An exact print file reproduction or viewing requires that the AFP resources used to print the file are also available to the reprint / browse process. This simple and basic consideration is not always easy to implement as installations, motivated by performance gains when the file is printed, tend to group all resources into large system libraries which may become repositories used by all sorts of applications. The result: it is very difficult to identify the resources used by each print file.

SPS/MVS-CIS helps installations having this problem by providing a Resource retrieval and Packaging feature. It can be used to locate and copy, into a separate Resource File, all the resources used by an application. The Resource Packaging feature can be activated selectively for each resource type (i.e. fonts, Page segments, Overlays, etc.).

The output file containing the resource data may be a sequential data set (also called Resource Object file) or it may be a partitioned data set which can then be used as a user library in order to print or distribute the application.

The SPS/MVS-CIS parameter interface allows installations to define different resource libraries to be used during the retrieval phase. The FDEFLIB, PDEFLIB, FONTLIB, OVLYLIB, PSEGLIB and OBJCONLIB parameters may be used to specify one or more resource libraries to be used for retrieving a particular resource type. The USERLIB parameter adds an extra level of library differentiation.

SPS/MVS-CIS also supports Inline Resources. Inline resources are not stored as external files, but included in the print file data passed to SPS/MVS-CIS for processing. With this feature, it is very simple to recreate a user resource library using the Inline resources present in a print file. The inline resource support provided by SPS/MVS-CIS is compatible with the AFP drivers (i.e. SPS/MVS-APA, PSF, etc.).

SPS/MVS-CIS may be used to include resources which are not part of the print data stream, but that are specified during print time or in the JCL. Example: Front / Back overlays, CHARS, etc. It may also be used to override some control options specified in the resources being packaged such as Input Bin selection, Output bin selection, Print direction, page offset and others. In all cases, the contents of the output Resource Object file created by SPS/MVS-CIS will include the extensions or modifications requested in the parameter file so that recreating an exact printout does not require the parameter file.

**Example:**

Using the application shown in *Fig. 7 Legacy application* on page 33, the following SPS/MVS-CIS resource packaging parameters would request the generation of a Resource Object file to be archived together with the document and index files:

```
RESTYPE ( ALL )
USERLIB ( CIS.AFPLIB )
```
Fig. 9. Resource Packaging parameters

The following picture shows an extract of the contents of the Resource Object file resulting from the parameters described above:

```
0000 00 D3A8C6  BRG  Begin resource group <.........>  1
0001 00 D3A8CE  BR  Begin resource <F1CIS02 >  2
0008 00 D3A8CD  BPM  Begin form map <.........>  3
0009 00 D3A8CC  BMM  Begin medium map <CIS2F1 > 10
... 0011 00 D3A9CD  BFM  End form map <........> 19
0012 00 D3A9CE  BFM  End form map <.........> 20
0000 00 D3A9CE  BR  End resource <F1CIS02 > 21
0000 00 D3A9CE  BR  Begin resource <S1CIS2F > 22
... 0000 00 D3A9CE  BR  End resource <S1CIS2F > 96
0000 00 D3A9CE  BR  Begin resource <S1CIS2B > 97
... 0000 00 D3A9CE  BR  End resource <S1CIS2B > 171
0000 00 D3A8CE  BR  Begin resource <S1CIS2B2> 172
... 0000 00 D3A8CE  BR  End resource <S1CIS2B2> 190
0000 00 D3A8CE  BR  Begin resource <S1CIS2F3> 191
... 0000 00 D3A8CE  BR  Begin resource <C0ARI10N> 652
... 0000 00 D3A9CE  BR  End resource <C0ARI10N> 666
0000 00 D3A8C6  BREG  End resource group <.........> 667
```

Fig. 10. Resource Object file

3.4
Using SPS/MVS-CIS to Generate the AFP Index Object file

SPS/MVS-CIS can be used to collect the index information being written into the output document file and to optionally generate an AFP Index Object File.

The index information collected by SPS/MVS-CIS during its normalization phase is used to control the subsequent processing steps, including page sorting and data enrichment. This information is also used to generate an Index Object file when requested. The format of the Index Object file is defined by the AFP architecture, and is supported by most AFP browsers available. AFP archiving systems such as PRISMAarchive also support this index information in order to build their own indexing tables.

As described on page 31 under Understanding SPS/MVS-CIS conversion and indexing process, SPS/MVS-CIS also inserts tag elements into the resulting output file. These tags are called Tagged Logical Element (TLE) and may be inserted at a Page or Page group level. SPS/MVS-CIS allows an installation to optimize the size of the resulting Index Object File by specifying the level of indexing information the file should contain using the INDEXOBJ parameter.

Example 1:

Using the application shown in Fig. 7 Legacy application on page 33 and the indexing parameters shown in Fig. 6 Legacy application: Indexing parameters, the following SPS/MVS-CIS Index Object file is created:

```
0000 00 D3AA7 BDI Begin document index
coded_graph_charset_global_id GCSSID = <FFFF>, CCSID = <01F4>
fully_qualified_name FQType = 'Begin Document Reference', FQName = <SPS4.CIS.CISOUT.INDEX>
Fully_qualified_name FQNType = 'Replace first GID name',
Fully_qualified_name FQNType = 'Begin Document Reference',
0000 00 D3B2A7 IEL index element
coded_graph_charset_global_id GCSSID = <FFFF>, CCSID = <01F4>
fully_qualified_name FQType = 'Index Element GID',
Direct_byte_offset_triplet DirByOff = <00000063>
fully_qualified_name FQNType = 'Begin Page Group Reference',
object_SF_offset_triplet SFOff = <00000001>
0000 00 D3A90 TLE tag logical element
coded_graph_charset_global_id GCSSID = <FFFF>, CCSID = <01F4>
fully_qualified_name FQType = 'Attribute GID',
attribute_value_triplet AttVal = <F8F4F4F3F6F0F2F3>
0000 00 D3A90 TLE tag logical element
coded_graph_charset_global_id GCSSID = <FFFF>, CCSID = <01F4>
fully_qualified_name FQType = 'Attribute GID',
attribute_value_triplet AttVal = <E2818289958540C2818388A2A3858995>
0000 00 D3A90 TLE tag logical element
coded_graph_charset_global_id GCSSID = <FFFF>, CCSID = <01F4>
fully_qualified_name FQType = 'Attribute GID',
attribute_value_triplet AttVal = <D99685948599A2A3994B40F6F7>
```
Example 2:
Using the same parameters as in Example 1 but INDEXOBJ, which is set to GROUP instead of ALL causes SPS/MVS-CIS to generate the following Index Object file:

```plaintext
0000 00 D3A900 TLE Tag logical element
coded_graph_charset_global_id GCCSID = <FFFF>, CCSID = <01F4>
fully_qualified_name FQName = <ACCOUNT>
attribute_value_triplet AttVal = <F1F4F0F3F4F3F5F0>

0000 00 D3A901 TLE Tag logical element
coded_graph_charset_global_id GCCSID = <FFFF>, CCSID = <01F4>
fully_qualified_name FQName = <NAME>
attribute_value_triplet AttVal = <E2818289958540C2818388A2A385895>
```

Fig. 11. Index Object file (example 1)
SPS/MVS-CIS: Converting, Indexing and Sorting print data

Fig. 12. Index Object file (example 2)
Using SPS/MVS-CIS to Sort and Reorder output pages

SPS/MVS-CIS does not only convert, index a print file and retrieve all resources required by an application; but also allows an installation to reorder the pages - contained in the print file - in a completely new sequence.

Two different kinds of sorting schemes are available:

1. Contents sorting
   Where documents (page groups, sheets) are regrouped based on their data contents (i.e. Postal code, Street name, Customer name, etc.), and

2. Layout sorting
   Where pages are regrouped into a completely new layout (order), replacing the previous document and sheet contents with a new one.

The sorting process implemented in SPS/MVS-CIS is the same for all sorting types and all sorting schemes. It consists of the following steps:

a) The Conversion and/or Normalization process is done for the complete input file. The resulting data is stored into the sorting work file (see CISSWAP on page 58).

b) The Index tags inserted or Collected during the Conversion/Normalization process are kept in storage. Resources, in case RESOBJ specified, are kept in storage too.

c) The input file structure is "simplified". This optional step is mostly used in connection with Layout sorting schemes such as SCHEME_1, 4UP_BOOKLET, etc.

d) Empty pages are added as required.

e) The Sorting algorithm is applied.

f) The output data is read from the CISSWAP work file and written to the Output file. The resource data, when requested, is written too.

The sorting methods available under SPS/MVS-CIS are explained in this chapter.
3.5.1 Contents sorting

Contents sorting takes advantage of the indexing information obtained during the normalization and conversion process done by SPS/MVS-CIS. How this process works is described in this manual under 3.2.1 Converting and Indexing print files containing S/370 Line formatted data, and under 3.2.4 Normalizing print files containing AFP and MO:DCA-P data.

Because Indexing information is independent of the data being presented in a page (index data does not need to be printed) it is possible for SPS/MVS-CIS to consolidate (merge) data from different applications into a single output file provided that the same indexing keys are used (see 3.6 Using SPS/MVS-CIS to Consolidate mail pieces on page 55 for more information).

Sorting input files using their page group level indexes causes SPS/MVS-CIS to leave the sheet/document structure unchanged. However, the input print file is required to have all page group boundaries matching the beginning of a new sheet.

```
SORT ( ASC, PLZ, STREET, NAME, ACCOUNT )
```

Fig. 13. Sort parameters

Contents sorting requires clean document structures. In case of legacy applications in S/370 Line format data, this structure is created by SPS/MVS-CIS during the Normalization process( see 3.2.1 Converting and Indexing print files containing S/370 Line formatted data on page 31).

In case of AFP applications, an installation should make sure that the report generator or transformer being used is capable of creating such a document structure and can generate the Index Tag Element (TLE) structured fields required by SPS/MVS-CIS for the contents sorting process.

Example:

Using the application shown in Fig. 7 Legacy application on page 33, the indexing parameters shown in Fig. 6 Legacy application: Indexing parameters, and the sorting parameters described in Fig. 13 Sort parameters the following process is done by SPS/MVS-CIS:
Contents sorting may be done in an ascending or descending sequence. The sorting algorithm implemented in this version of SPS/MVS-CIS is based on a simple comparison of the hexadecimal values of the fields being sorted. No considerations are made for specific data coding schemes (ASCII, EBCDIC), national characters or special symbols. Data alignment within the field (left or right justified), leading and/or trailing characters (i.e. leading zeroes, trailing blanks) and decimal separation characters (i.e. comma or dot) are handled as part of the data, without special considerations.

Fig. 14. Contents sorting
Example:

Fig. 15 shows the AFP document structured generated by SPS/MVS-CIS for the example shown in Fig. 7 Legacy application and used during the sorting process shown in Fig. 14 Contents sorting.

Requirements and considerations:

Following is a list of requirements and considerations that apply to Contents sorting. Deviations from this usage model are possible but not recommended as they may stop working in future SPS/MVS-CIS versions.

1. The sorting algorithms (ASC, DES) implemented in this version of SPS/MVS-CIS are based on a simple comparison of the hexadecimal values of the fields being sorted. No considerations are made for specific data coding schemes (ASCII, EBCDIC), national characters or special symbols. Data alignment within the field (left or right justified), leading and/or trailing characters (i.e. leading zeroes, trailing blanks) and
decimal separation characters (i.e. comma or dot) are handled as part of the data, without special considerations.

2. A mail piece, as defined in the Glossary on page 81, is the unit used when reordering the input print file. All pages and sheets within the mail piece are left in their original sequence. The print format is not modified. Nested page groups, if any, are also left unchanged.

3. Every mail piece must start on a new sheet.

4. The Page Numbering feature in the Page Definition (generated using SLE) is done before the sorting process. The pages numbers inserted reflect the original page sequence.
3.5.2 Layout sorting

SPS/MVS-CIS supports several layout sorting schemes allowing an installation to take advantage of fanfold-duplex and fanfold-2up printers, without having to change their existing applications. They are:

- **Fanfold listing duplex (SCHEME_1)**

  With the availability of continuous-forms duplex printing, installations can reduce paper costs by using the reverse side of most sheets. For short reports, most users prefer individual sheets that can be stapled and bind together. However, for long reports such as large program listings, storage dumps, console logs, EREP reports, GTF traces and others, stapling the output is unwieldy. At the same time, many installations deliver long reports as fan-fold output and their logistics are setup for this report format.

  The SCHEME_1 of Layout sorting available in SPS/MVS-CIS (also implemented by the SPS/MVS Page Sort utility) offers a solution for these long reports as it allows an installation to immediately realize the benefits of duplex printing, without using any post-processing machinery and without writing specific printing applications especially designed for duplex printers.

  Fig. 16 illustrates the resulting output. Black numbers represent the front-side page, gray-numbers are used to represent the back-side pages.

Requirements and considerations:

Following is a list of requirements and considerations that apply to this sorting scheme. Deviations from this usage model are possible but not recommended as they may stop working in future SPS/MVS-CIS versions.

1. The sorting scheme (and printing sequence) is: 1, n, 2, (n-1), 3, (n-2), 4, ..., (n:2)+3, (n:2)-1, (n:2)+2, (n:2), (n:2)+1. A blank page is printed after page 1 when \( n \) is an odd number.

2. This sorting scheme does a “simplification” of the input print file, removing existing Page Groups. A single Page Group containing all pages is created as output.

3. Index tags are also removed during the simplification process. A warning message is issued in this case.

4. In order to achieve the results expected with this sorting scheme, the Form Definition used should have the following characteristics:
- Duplex printing
- One-UP
- Medium Overlay(s), in case specified, will also be printed in the blank page inserted by SPS/MVS-CIS

5. Any attempt to control page/sheet grouping within the application (i.e using IMM to force a new sheet or using conditional processing) may cause unexpected print results.

6. The Page Numbering feature in the Page Definition (generated using SLE) is done before the sorting process. The page numbers inserted reflect the original page sequence.

Fig. 16. Layout sorting: SCHEME_1

- Fanfold listing 2-UP (SCHEME_2)

Fanfold 2-up sorting is often used on A3 simplex printers. Two A4 pages are printed side by side on each sheet. The postprocessing unit trenches the paper stream in the middle and stacks the two halves. Following graphic depicts the resulting output.
Requirements and considerations:

Following is a list of requirements and considerations that apply to this sorting scheme. Deviations from this usage model are possible but not recommended as they may stop working in future SPS/MVS-CIS versions.

1. The sorting scheme (and printing sequence) is: 1, (n:2)+1, 2, (n:2)+2, 3, ... (n:2)-1, n-1, n/2, n. A blank page is added at the end when $n$ is an odd number.

2. This sorting scheme does a “simplification” of the input print file, removing existing Page Groups. A single Page Group containing all pages is created as output.

3. Index tags are also removed during the simplification process. A warning message is issued in this case.

4. In order to achieve the results expected with this sorting scheme, the Form Definition used should have the following characteristics:
   - Simplex printing
   - Two-UP
   - Medium Overlay(s), in case specified, will also be printed in the blank page inserted by SPS/MVS-CIS
5. The Page Numbering feature in the Page Definition (generated using SLE) is done before the sorting process. The pages numbers inserted reflect the original page sequence.

- **Reverse printing sequence (SCHEME_3)**

For fanfold and cut-sheet listings, simplex, one-UP. The page sequence found in the print file is inverted, starting with the last page and ending with the first one.

**Fig. 18. Layout sorting: SCHEME_3**

**Requirements and considerations:**

Following is a list of requirements and considerations that apply to this sorting scheme. Deviations from this usage model are possible but not recommended as they may stop working in future SPS/MVS-CIS versions.

1. The sorting scheme (and printing sequence) is: n, (n-1), (n-2), (n-3), ..., 3, 2, 1.
2. This sorting scheme does a “simplification” of the input print file, removing existing Page Groups. A single Page Group containing all pages is created as output.

3. Index tags are also removed during the simplification process. A warning message is issued in this case.

4. In order to achieve the results expected with this sorting scheme, the Form Definition used should have the following characteristics:
   - Simplex printing, or
   - Duplex with only one side containing variable data
   - One-UP

   In case of cut-sheet output, a FormDef with a constant front and a variable back side may be used to obtain “face-up” stacked output with only one side containing variable data.

5. Any attempt to control page/sheet grouping within the application (i.e using IMM to force a new sheet or using conditional processing) may cause unexpected print results unless it is done for every sheet.

6. The Page Numbering feature in the Page Definition (generated using SLE) is done before the sorting process. The pages numbers inserted reflect the original page sequence.

- **4-UP booklet sorting (4UP_BOOKLET)**

4-up booklet sorting can be applied on A3 duplex printers to get A5 books. Following graphic depicts the result after post-processing (trenching and folding) the printers output of a 6 page job.

The booklet has 8 pages. The page group to be printed only consists off 6 pages, the last two are empty. Subsequent picture shows the sequence with which the trenched sheets are stacked (the numbers represent the order the sheets are stacked). After trenching the sheets are folded at the dashed lines.
Fig. 19. Layout sorting: 4UP_BOOKLET
Following graphic depicts the resulting printer output of a 9 page job. Again black numbers represent pages on the front, gray numbers represent pages on the back side. Pages marked with "X" represent empty pages generated by SPS/MVS-CIS. Pages marked with "P" represent pages of the next job.

Fig. 20. Layout sorting: 4UP_BOOKLET

Requirements and considerations:

• The sort scheme is: 2, 4, n-1, n-3, 1, 3, n, n-2, 6, 8, n-5, 7, 5, n-4, n-6, ...
• The number of pages is rounded up to the next value modulo 8 (in this case 16, this means 7 empty pages are generated, which is the maximum)

Fig. 21. Layout sorting: 4UP_BOOKLET
• Requirements to AFP-Formdef: duplex, 4-up, at most one MMO (recommended: no MMO), logical page positioning (the sequence the printer renders the pages on every sheet, shown in the next picture, black numbers represent pages on the front, gray numbers represent pages on the back side.)

3.6
Using SPS/MVS-CIS to Consolidate mail pieces

Many installations know this problem: two or more applications generate print output (mail pieces) to be sent to the same addressee but, because the applications run independently, the mail pieces are also printed and sent separately. These installations will like to consolidate the output in such a way that all the documents belonging to the same addressee are printed together and, more important, sent together in one larger mail piece saving shipping costs and improving the acceptance of the material being sent.

Fig. 22. Mail Piece Consolidation

It is not always possible and many times not even recommended to modify the applications for this kind of post optimization. With SPS/MVS-CIS it is also not necessary. Provided that some requirements (see below) are fulfilled, the Contents sorting facility of SPS/MVS-CIS can be used to re-order two or more input print files in such a way that all the mail pieces going to one addressee are printed together. The requirements are necessary in order to make it possible for SPS/MVS-CIS to identify which mail pieces belong together. Fig. 22 shows how Report Consolidation works.
Requirements and considerations:

Any application to be consolidated using the Contents sorting facility of SPS/MVS-CIS must fulfill the following requirements:

1. Indexing requirements
   - All the reports to be consolidated must include the Mail piece Index tag(s) to be used for sorting.
   - The index tags may be inserted by the application itself, or may have been inserted by SPS/MVS-CIS in a previous run (see 3.2 Understanding SPS/MVS-CIS conversion and indexing process on page 31).
   - An Index Object file may be generated but is not required.

2. Resource requirements
   The printing layout of the applications to be consolidated may be completely different (i.e. one application prints Portrait-Simplex, the next one Landscape-duplex and a third one Landscape-Nup). However, the resources used by the applications must fulfill the following requirements:
   - All Medium Maps used must be manually combined into a single Formdef. The name of this Formdef must be specified to the SPS/MVS-CIS Consolidation process.
   - Each Medium Map in the Formdef must have a unique name. Two different Medium Maps with the same name will cause incorrect printing results.
   - The resource libraries used by all the applications to be consolidated must be specified to the SPS/MVS-CIS Consolidation process. It is not possible to use the same name to refer to two or more different resources of the same type (i.e. having two different coded fonts called X0MYFONT will cause incorrect printing results).

3. The Page Numbering feature in the Page Definition (generated using SLE) is done before the sorting/consolidation process. The pages numbers inserted reflect the original page sequence within the original print files.

4. All other requirements and considerations applicable to Contents sorting also apply to the Mail Piece consolidation process. Refer to 3.5.1 Contents sorting on page 43 for more information.

3.7
SPS/MVS-CIS runs as a normal batch job and requires Job Control Language definitions. Fig. 23 shows an example of the JCL required. The statements enclosed in brackets ([ ]) are optional.

The purpose of each of these JCL statements required is described next:

**EXEC**
Specifies the SPS/MVS-CIS program name (SPSPCIS) and the region size to be used. The recommended size is 5Mb. Information on how to calculate the region size required by an application can be found under 3.9.1 Memory requirements on page 80.

**STEPLIB**
Specifies the DDNAME for the library where SPS/MVS-CIS is installed.

**SYSPRINT**
Indicates the destination where SPS/MVS-CIS will generate the information and error messages it issues.

**SYSTERM**
Indicates the destination where SPS/MVS-CIS will generate additional messages in case a severe runtime error occurs.

**SYSIN**
Specifies the DDNAME for the dataset containing the parameters for SPS/MVS-CIS. The syntax of the parameters supported is described on page 62 under SPS/MVS-CIS Parameter File reference. A data delimiter other than /* should be specified as this string is used in the CIS parameter syntax.

**INPUT**
Specifies the default DDNAME for the print file to be processed by SPS/MVS-CIS. It is possible to specify a different DDNAME using the parameter INPUTDD. Refer to SPS/MVS-CIS Parameter File reference for more information.

The data set used as input to SPS/MVS-CIS may include an Index and a Resource Object file (inline resources). However, if present, this Object files must be part of the same sequential data set where the document data is stored and not separate data sets which are concatenated via JCL.
Fig. 23. Sample JCL to invoke SPS/MVS-CIS

INDEX Specifies the default DDNAME for the Index Object file to be generated by SPS/MVS-CIS. It is possible to specify a different DDNAME using the parameter INDEXDD. Refer to SPS/MVS-CIS Parameter File reference for more information.
The Index Object file may have the following characteristics:

```
DSORG  PS
RECFM  V, VA, VM, VBA, VBM
LRECL  32756
BLKSIZE 32760
```

Do not specify carriage control (i.e. VB instead of VBA) if you want SPS/MVS-CIS to create a file that does not contain a carriage control byte (x'5A') at the beginning of every record. The record size should be large enough to contain the largest index record to be generated. Specifying a length of 32756 is recommended.

**OUTPUT**

Specifies the default DDNAME for the Output document file to be generated by SPS/MVS-CIS. It is possible to specify a different DDNAME using the parameter OUTPUTDD. Refer to *SPS/MVS-CIS Parameter File reference* for more information.

The output file may have the following characteristics:

```
DSORG  PS
RECFM  V, VA, VM, VBA, VBM
LRECL  32756
BLKSIZE 32760
```

Do not specify carriage control (i.e. VB instead of VBA) if you want SPS/MVS-CIS to create a file that does not contain a carriage control byte (x'5A') at the beginning of every record. The record size should be large enough to contain the largest document record to be generated. Specifying a length of 32756 is recommended.

**RESOBJ**

Specifies the default DDNAME for the Resource Object file to be generated by SPS/MVS-CIS. It is possible to specify a different DDNAME using the parameter RESOBJDD. Refer to *SPS/MVS-CIS Parameter File reference* for more information.

The Resource Object file may have the following characteristics:

```
DSORG  PS
RECFM  V, VA, VM, VBA, VBM
LRECL  32756
BLKSIZE 32760
```

Do not specify carriage control (i.e. VB instead of VBA) if you want SPS/MVS-CIS to create a file that does not contain a carriage control byte (x'5A') at the beginning of every record. The record size should be large enough to contain the largest resource record to be generated. Specifying a length of 32756 is recommended.
CISSWAP
Specifies the DDNAME for the sort work file to be used by SPS/MVS-CIS. This must be a VSAM file with the following characteristics:

```
DEFINE CLUSTER ( INDEXED REUSE -
   NAME (SPS4.CIS.CISOUT.WORK) -
   VOLUMES(SPSCIS) CYLINDERS(50,50) -
   KEYS(8 0) RECORDSIZE(1 8128) SPANNED -
   ) -
   DATA { NAME (SPS4.CIS.CISOUT.WORKDATA) } -
   INDEX { NAME (SPS4.CIS.CISOUT.WORKIDX) }
```

SPS1CIS
SPS2CIS
Specify the DDNAME for the trace datasets where SPS/MVS-CIS will write diagnostic information in case the TRACE parameter is specified. The two datasets specified are processed in a wrap-around manner.

They should only be specified when requested by your SMS/MVS software support, and should have the following characteristics:

```
DSORG PS
RECFM VBS
LRECL 32756
BLKSIZE 32760
```

SPS/MVS-CIS trace datasets may be transferred using TCP/IP FTP in binary format. The trace data sets may contain secondary extents. In this case, the dataset switching will occur when the last extent is full.

IFDEF
Can be used to control the execution of subsequent job steps based on the return code received. The return codes (decimal) set by SPS/MVS-CIS are:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful execution. No errors were found.</td>
</tr>
<tr>
<td>4</td>
<td>Successful execution with warning messages.</td>
</tr>
<tr>
<td>8</td>
<td>Data error detected during execution. One or more error messages were written to SYSPRINT describing the error found.</td>
</tr>
<tr>
<td>12</td>
<td>File related error detected during execution. One or more error messages were written to SYSPRINT describing the error found.</td>
</tr>
<tr>
<td>16</td>
<td>Program related error detected during execution. One or more error messages were written to SYSPRINT describing the error found.</td>
</tr>
</tbody>
</table>
IEBGENER  This standard MVS utility may be optionally used after an SPS/MVS-CIS step in order to concatenate the Index, Resource and Document output files generated by SPS/MVS-CIS into a self-contained AFP file which includes in-line resources.

The resource object file used in this case (SPS4.CIS.CISOUT.RES) must be generated using the parameter RESFILE( SEQ ) as described on page 74.
SPS/MVS-CIS Parameter File reference

- **Syntax rules:**

1. An SPS/MVS-CIS-parameter file is a text file (EBCDIC) that can consist of:
   - Keywords and their parameters as described below
   - Comments
   - Blank characters

2. In general, keywords and their parameters are not case sensitive, and are converted to uppercase before processing. The only exception are those parameters that are enclosed using single quotation marks (`'Case Sensitive'`).

3. Keywords and parameters may start in any column and may comprise one or more lines. No continuation mark is required. Multiple keywords may be specified in one line.

4. Keywords are not positional and may be specified in any sequence. The keyword parameters are positional and must be specified in the sequence indicated in this manual.

5. Blank characters may be used to improve the readability of the parameter file. They can appear anywhere between keywords and parameters, and are ignored unless they appear as part of a parameter enclosed using two single quotation marks (`'parameter with blanks'`). Blank lines are also allowed.

6. Comments (free form descriptive text) may consist of one or more partial or complete lines. The two-character string `/*` is used to indicate that a comment begins. The two-character string `*/` indicates the end of a comment. These strings may appear anywhere in the file and cause all the enclosed text to be ignored. Comments may not be nested.

7. The following description uses square brackets ([ ]) to indicate optional parameters. Underscore `_` is used to show the default values when a keyword or parameter is not specified.
# SPS/MVS-CIS Parameter Syntax

<table>
<thead>
<tr>
<th></th>
<th>Conversion</th>
<th>Indexing</th>
<th>Sorting</th>
<th>Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CC</strong> ( NO</td>
<td>SNI</td>
<td>ASA</td>
<td>IBM</td>
<td>ASAA</td>
</tr>
<tr>
<td><strong>CHARS</strong> ( fontname [&lt;mxm&gt;]</td>
<td>fontname [&lt;mxm&gt;]</td>
<td>[...] )</td>
<td>C</td>
<td>R</td>
</tr>
<tr>
<td><strong>COLORMAP</strong> ( name )</td>
<td></td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>FDEFLIB</strong> ( data set name [,data set name] [...] )</td>
<td></td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>FIELD</strong> ( fieldname, { record, column, length }</td>
<td>{ 'literal value'</td>
<td>`x' literal value'} )</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td><strong>FONTLIB</strong> ( data set name [,data set name] [...] )</td>
<td></td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>FORMDEF</strong> ( formdefname</td>
<td>DUMMY )</td>
<td></td>
<td></td>
<td>R</td>
</tr>
<tr>
<td><strong>GROUPNAME</strong> ( indexname )</td>
<td></td>
<td>I</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>IMAGEOUT</strong> ( ASIS</td>
<td>IOCA</td>
<td>IOCANOR )</td>
<td></td>
<td>C</td>
</tr>
<tr>
<td><strong>INBIN</strong> ( value )</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>INDEX</strong> ( indexname , triggername , `attribute name' , fieldname [,fieldname] [...] )</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>INDEXDD</strong> ( INDEX</td>
<td>ddname )</td>
<td>I</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>INDEXOBJ</strong> ( GROUP</td>
<td>ALL</td>
<td>NONE )</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td><strong>INDEXSTARTBY</strong> ( 1</td>
<td>value )</td>
<td>I</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>INPUTDD</strong> ( INPUT</td>
<td>ddname )</td>
<td>C</td>
<td>I</td>
<td>S</td>
</tr>
<tr>
<td><strong>LUPUB</strong> ( value )</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>OBJCONLIB</strong> ( data set name [,data set name] [...] )</td>
<td></td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>OFFSETXF</strong> ( nnnn [.mmm] unit )</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>OFFSETYF</strong> ( nnnn [.mmm] unit )</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>OFFSETXB</strong> ( nnnn [.mmm] unit )</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>OFFSETYB</strong> ( nnnn [.mmm] unit )</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>OUTBIN</strong> ( value )</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## SPS/MVS-CIS Parameter syntax

<table>
<thead>
<tr>
<th>SPS/MVS-CIS Parameter</th>
<th>CONVERSION</th>
<th>INDEXING</th>
<th>SORTING</th>
<th>RESOURCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTPUTDD (OUTPUT</td>
<td>ddname)</td>
<td>C</td>
<td>I</td>
<td>S</td>
</tr>
<tr>
<td>OVERLAYF (overlayname [,overlayname] [...])</td>
<td></td>
<td></td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>OVERLAYB (overlayname [,overlayname] [...])</td>
<td></td>
<td></td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>OVLYLIB (data set name [,data set name] [...])</td>
<td></td>
<td></td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>PAGEDEF (pagedefname</td>
<td>DUMMY)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PDEFLIB (data set name [,data set name] [...])</td>
<td></td>
<td></td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>PRESENTATION (ASIS</td>
<td>PORTRAIT</td>
<td>LANDSCAPE</td>
<td>C</td>
<td>PORTRAIT90</td>
</tr>
<tr>
<td>PRINTMODE (SOSI1</td>
<td>SOSI2</td>
<td>SOSI3)</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>PSEGLIB (data set name [,data set name] [...])</td>
<td></td>
<td></td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>RESFILE (SEQ</td>
<td>PDS)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESOBJDD (RESOBJ</td>
<td>ddname)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESTYPE (NONE</td>
<td>[ALL]</td>
<td>[FDEF]</td>
<td>[PSEG]</td>
<td>[OVLY]</td>
</tr>
<tr>
<td>SORT (SCHEME_1</td>
<td>SCHEME_2</td>
<td>SCHEME_3</td>
<td>SCHEME_4</td>
<td>4UP_BOOKLET</td>
</tr>
<tr>
<td>TRACEFLAGS (hex value)</td>
<td></td>
<td></td>
<td></td>
<td>C</td>
</tr>
<tr>
<td>TRACELEVEL (value [,value] [...])</td>
<td></td>
<td></td>
<td></td>
<td>C</td>
</tr>
<tr>
<td>TRCTYPE (NO</td>
<td>IBM</td>
<td>SNI)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRIGGER (triggername *, {column</td>
<td>*}, {value</td>
<td>x'value'}) [record, {column</td>
<td>*}, {value</td>
<td>x'value'}] [...])</td>
</tr>
<tr>
<td>UNIQUEBNGS (YES</td>
<td>NO)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USEPAGENAMES (YES</td>
<td>NO)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USERLIB (data set name [,data set name] [...])</td>
<td></td>
<td></td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>X2UP (OFF</td>
<td>ON</td>
<td>LEFT</td>
<td>RIGHT</td>
<td>ICOPIES)</td>
</tr>
</tbody>
</table>
CC ( NO | SNI | ASA | IBM | ASAA | SS | DS | TS )

Specifies the type of printer control characters to be used.

- NO: No printer control characters are used
- SNI: SNI control character
- ASA: ASA control character (EBCDIC)
- IBM: Machine control character
- ASAA: ASA control character (ASCII)
- SS: Force single spacing
- DS: Force double spacing
- TS: Force triple spacing

Example: CC ( ASA )


Specifies the member name of the coded font(s) to be used to process a file. Optionally, the matrix memory position <mxm> may be specified too. This list overrides any fonts specified in the Pagedef or in the data stream.

- fontname: A 1 to 8 character Coded font name(s) including the font prefix (i.e. X0GT10)
- mxm: A numeric value between 0 and 63.

The mxm position corresponds to the TRC value in the print data set that will select the particular font. At least one mxm must specify (or default to) the value 0 as this mxm is used as default for TRC’s which may not be in the list.

The mxm’s can be in any order. When the mxm position is not explicitly coded, mxm position starts at 0 and is incremented by 1 for each value in the parameter statement. When a mxm field is specified with the font, that font is loaded into the specified mxm position and the next font specified will be loaded into mxm location plus 1.

Example: CHARS ( MYFONT, X0GT12<23>, X0GT15 )

would cause MYFONT to be loaded at position 0; X0GT12 at position 23 and X0GT15 at position 24. All other positions (TRC’s) remain undefined and cause position 0 to be used.
SPS/MVS-CIS: Converting, Indexing and Sorting print data

**COLORMAP (name | DUMMY)**

Specifies the member name of a Color Mapping table. The value is:

*name*  
The name can be one to eight alphanumeric characters, including the two-character prefix, if there is one. Specifying DUMMY requires the print file to contain at least one inline Color mapping table. SPS/MVS-CIS uses the first color mapping table found and ignores all others.

Example:  

```
COLORMAP ( MYCOLMAP )
```

**FDEFLIB (data set name [, data set name ] [,...] )**

Specifies the data set names where SPS/MVS-CIS searches for the form definitions. CIS looks in the order the data set names are given. Any number of data set names can be specified.

The order SPS/MVS-CIS searches for Form definitions is:

- Inline resource group
- Userlibs if given
- Defined data set names

Example:  

```
FDEFLIB ( MY.FDEFLIB,YOUR.FDEFLIB )
```

**FIELD (fieldname, {record,column,length} | {'literal value' | x'literal value' })**

Specifies the data fields to be used to construct the indexing information, when processing S/370 line formatted data.

*fieldname*  
A 1 to 8 byte character name which is used as reference in the INDEX parameter.

*record*  
Specifies the relative record number from the indexing anchor record. Supported values are –255 to 255.

*column*  
Will be used as byte offset from the beginning of a record. If carriage control characters were used, column 1 refers to this. Supported values are 1 to 32756.

*length*  
Specifies the number of characters, starting by column
SPS/MVS-CIS: Converting, Indexing and Sorting print data

to compose this field. Supported values are 1 to 250.

Example: FIELD ( accountF, 7, 3, 8 )

'literal value'

x'literal value' Specifies a constant value. This value can consist of character data or hexadecimal data. The length has a range from 1 to 250 for character data and 1 – 500 for hexadecimal data. Be careful this is a case-sensitive field.

Example: FIELD ( myField, 'My constant string' )

FONTLIB ( data set name [, data set name ] [ ,... ] )

Specifies the data set names where SPS/MVS-CIS searches for the font definitions. CIS looks in the order the data set names are given. Any number of data set names can be specified.

The order SPS/MVS-CIS searches for font definitions is:

- Inline resource group
- Userlibs if given
- Defined data set names

Example: FONTLIB ( MY.FONTLIB, YOUR.FONTLIB )

FORMDEF ( formdefname | DUMMY )

A 1 to 8 character name of the form definition to be used in printing the print data stream. The complete name must be specified. No prefix is added by CIS.

Specifying DUMMY requires the print file to contain at least one inline Formdef. SPS/MVS-CIS uses the first Formdef found. All others are ignored.

Example: FORMDEF ( F1MYFORM )
GROUPNAME  

Specifies which of the index values should be used as the groupname for each index group. Using the most unique index value for the group name is recommended. The intent is to have a unique group name for every group CIS produces in the output file. The value includes the FIELD definitions from the INDEX parameter but does not include the attribute name. The maximum name length allowed in AFP is 250 characters.

Example:  

GROUPNAME( accountI )

This parameter is rejected when specified together with the sorting schemes SCHEME_1, SCHEME_2, SCHEME_3 or 4UP_BOOKLET.

IMAGEOUT  

Specifies the format of the image data produced by CIS in the output file.

ASIS  
CIS uses same image format as in input file.

IOCA  
CIS converts image data into uncompressed IOCA format using the replicate-and-trim option.

IOCANOR  
CIS converts image data into uncompressed IOCA format without using the replicate-and-trim option.

Example:  

IMAGEOUT ( IOCANOR )

INBIN  

Selection of the input bin from which to take the form on a page printer. The selection is done by physical media ID, and overrides all input media origins specified in the Form definition.

This parameter must be used together with RESTYPE( ALL ) or RESTYPE( FDEF ) otherwise it is ignored.

value  
A decimal character from 0 to 254
Example: INBIN ( 1 )

INDEX ( indexname , triggername , 'attribute name' , fieldname [,fieldname] [..] )

Specifies the content of the indexing tags for the entire file, when processing S/370 line formatted data. Each index may contain one or more field definitions.

indexname A 1 to 8 character name that specifies the index.

triggername Specifies a trigger.

attribute name Specifies a user-defined attribute name to be associated with the actual index value. This is a case-sensitive field. Supported range is from 1 to 250 byte.

fieldname Specifies one or more FIELD parameters that compose the index value. The total length of all fields used in an INDEX keyword may not exceed 250.

Example: INDEX ( account1, account, 'ACCOUNT', accountF )

INDEXDD ( INDEX | ddname )

Specifies the DDname for the index object file. It is an 1 to 8 byte character name. When CIS is indexing the file, it writes indexing information to this DDname.

Example: INDEXDD ( MYINDEX )

INDEXOBJ ( GROUP | ALL | NONE )

Specifies the amount of information CIS puts in the index object file. Selecting ALL may result in a very large Index Object file. A value other than NONE causes SPS/MVS-CIS to generate an Index Object file even if no index entries are available.

GROUP Only page-group-level entries

ALL Both page-group- and page-level entries

NONE No external indexing file will be written.
Example: INDEXOBJ ( NONE )

A value other than NONE is rejected when specified together with the sorting schemes SCHEME_1, SCHEME_2, SCHEME_3 or 4UP_BOOKLET.

INDEXSTARTBY (1 | value)

Specifies the output page number by which CIS must find an indexing field. Use this parameter to tell CIS to continue looking for an Indexing Trigger on a page other than the first one in the file.

Example: INDEXSTARTBY (3)

INPUTDD (INPUT | ddname)

Specifies the DDname for the file CIS will process. The name consists of 1 to 8 characters.

Example: INPUTDD (MYDD)

LUPUB (value)

Not all printers support the same l_units_per_units_base values. It is sometimes interesting to convert the value(s) used by an application to a value which is known as supported by the target presentation device (i.e. 2400). The LUPUB parameter specifies the l_units_per_unit_base value at which CIS should convert all currently specified values.

value Number of L-Units Per Unit Base supported by the printer for a unit base of 10 inches. A value of 1 to 32767 may be specified.

Example: LUPUB (2400)

OBJCONLIB (data set name [data set name] [...])

Specifies the data set names where CIS looks for the object container definitions such as the Color Mapping table. CIS looks in the order the
data set names are given. Any number of data set names can be specified. The order SPS/MVS-CIS searches for Object containers is:

- Inline resource group
- Userlibs if given
- Defined data set names

Example: OBJCONLIB (MY.CONTLIB, YOUR.CONTLIB)

Note: Color Mapping Table is the only Object container type supported in this SPS/MVS-CIS version

OFFSETXF ( nnnn [.mmm] unit )
OFFSETYF
OFFSETXB
OFFSETYB

Specify the offset in the X and/or Y direction of the logical page origin from the media origin for the Front and or the Back side of each sheet. If OFFSETxx isn’t specified the value specified in the form definition is used.

This parameter must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise it is ignored.

nnnn digit
mmm digit
unit one of the following values
IN specifies a unit of inches
CM specifies a unit of centimeters
MM specifies a unit of millimeters
PELS specifies a unit of picture elements (1/240 inch)
POINTS specifies a unit of points (1/72 inch)

If PELS or POINTS are specified, the mmm-values are ignored.

Example: OFFSETXF (1.5 IN)
OUTBIN (value)

Specifies the output bin number. The selection is done by physical media destination, and overrides all media destinations specified in the Form definition.

This parameter must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise it is ignored.

value A decimal character from 0 to 254

Example: OUTBIN (2)

OUTPUTDD (OUTPUT | ddname)

Specifies the DDname for the file where CIS puts the output. The name consists of 1 to 8 characters.

Example: OUTPUTDD (MYOUTDD)

OVERLAYF (overlayname [overlayname] [...])

OVERLAYB This parameter must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise it is ignored.

overlayname Specifies the name of a medium overlay to be placed on the Front or Back side of each sheet, in addition to overlays from other sources.

Example: OVERLAYF (MYOVLY1, MYOVLY2)

OVLYLIB (data set name [data set name] [...])

Specifies the data set names where CIS looks for the overlay definitions. CIS looks in the order the data set names are given. Any number of data set names can be specified.

The order CIS looks for definitions is:

- Inline resource group
- Userlibs if given
- Defined data set names
Example: OVLYLIB ( MY.OVLYLIB, YOUR.OVLYLIB )

PAGEDEF ( pagedefname | DUMMY )
A 1 to 8 character name of the page definition to be used in printing the print data stream. The complete name must be specified. No prefix is added by CIS.

Specifying DUMMY requires the print file to contain at least one inline Pagedef. SPS/MVS-CIS uses the first Pagedef found. All others are ignored.

Example: PAGEDEF ( P1TEST )

PDEFLIB ( data set name [ , data set name ] [ , ... ] )
Specifies the data set names where CIS looks for the page definitions. CIS looks in the order the data set names are given. Any number of data set names can be specified.

The order CIS looks for definitions is:
- Inline resource group
- Userlibs if given
- Defined data set names

Example: PDEFLIB ( MY.PDEFLIB, YOUR.PDEFLIB )

PRESENTATION ( ASIS | PORTRAIT | LANDSCAPE | PORTRAIT90 | LANDSCAPE90 )
Overrides any presentation specification specified in the Form definition. This parameter must be used together with RESTYPE( ALL ) or RESTYPE( FDEF ) otherwise it is ignored.

ASIS Presentation specified via file resources should be used by CIS
PORTRAIT Force presentation PORTRAIT.
LANDSCAPE Force presentation LANDSCAPE.
PRINTMODE ( SOSI1 | SOSI2 | SOSI3 )
Specifies the type of SOSI (shift-in shift-out) control to be used when processing double-byte data. Specifying a SOSI value that does not match the coding used in the data may cause unpredictable results.

SOSI1 Specifies that the SO or SI code invokes the font switch and causes a blank (X'40') to replace the SOSI code.
SOSI2 Specifies that the SO or SI code invokes the font switch only. No blank is inserted to replace the SOSI code.
SOSI3 Specifies that the SO or SI code invokes the font switch and causes two blanks (X'4040') to replace the SOSI code.

Example: PRINTMODE ( SOSI1 )

PSEGLIB ( data set name [ , data set name ] [ , ... ] )
Specifies the data set names where CIS looks for the page segment definitions. CIS looks in the order the data set names are given. Any number of data set names can be specified.
The order CIS looks for definitions is:
   Inline resource group
   Userlibs if given
   Defined data set names

Example: PSEGLIB ( MY.PSEGLIB, YOUR.PSEGLIB )

RESFILE ( SEQ | PDS )
Specifies in which format the used resources should be saved.
SEQ Creates a resource group that can be concatenated with the document file as inline resources. A sequential data set must
be allocated to the DDname specified in RESOBJDD.

PDS Stores each resource in a separate PDS member. A Partitioned data set must be allocated to the DDname specified in RESOBJDD.

Example: RESFILE ( PDS )

RESOBJDD ( RESOBJ | ddname )
Specifies the DDname for the file where CIS puts the resources. The name consists of 1 to 8 characters. A sequential or a Partitioned dataset may be specified (see RESFILE parameter).

Example: RESOBJDD ( MYDD )

Specifies the type of AFP print resources CIS should retrieve from the resource libraries for inclusion in the resource file (RESOBJDD).

NONE No resource file will be created.
ALL All resources will be saved in RESOBJDD.
FDEF Form definitions will be saved in RESOBJDD.
PSEG Page segments will be saved in RESOBJDD.
OVLY Overlays will be saved in RESOBJDD.
FONT Fonts will be saved in RESOBJDD.
OBJCON Object container will be saved in RESOBJDD.

Example: RESTYPE ( FDEF, PSEG, OVLY )

Note: Color Mapping Table is the only Object container type supported in this SPS/MVS-CIS version

SORT ( SCHEME_1 | SCHEME_2 | SCHEME_3 | 4UP_BOOKLET |
ASC, 'attribute name' ['attribute name'] [...] |
DES, 'attribute name' ['attribute name'] )
SPS/MVS-CIS: Converting, Indexing and Sorting print data

Specifies a sorting scheme (layout sorting), or a list containing the sort-direction and the user defined attribute names (contents sorting), associated to index values, specified in the INDEX parameter (only allowed for S/370 line formatted data) or to index values already present in the input file (AFP data).

```
SCHEME_1   1, n, 2, (n-1), 3, (n-2), ...
SCHEME_2   1, (n:2) + 1, 2, (n:2) + 2, 3, (n:3) + 3, ...
SCHEME_3   n, (n-1), (n-2), (n-3), ...
4UP_BOOKLET See Fig. 19 Layout sorting: 4UP_BOOKLET
ASC         Output will be sorted ascending.
DES         Output will be sorted descending.
attribute name Specifies a user-defined attribute name to be associated with the actual index value. This is a case-sensitive field. Supported range is from 1 to 250 byte.
```

If INDEX parameters are specified, `attribute name` must match with a `attribute name` from an index parameter.

Example: `SORT (ASC, 'PLZ', 'STREET', 'NAME')`

Refer to 3.5 Using SPS/MVS-CIS to Sort and Reorder output pages on page 42 for more information about sorting.

**TRACEFLAGS**

```
( hex value | FF, FF, FF, FF, FF, FF, FF )
```

An up to 16 character hex string used by a service representative to request predefined tracing options for diagnostic purposes. Every 2 hex characters must be separated by comma.

Example:

```
TRACEFLAGS(xx) will be expand to xFFFFFFFFFFFFFFFF,
TRACEFLAGS(..xx) will expand to FFFFFFFFFxFFFFFFFF and so on.
```

**TRACELEVEL**

```
( value [, value ] [... ] )
```
SPS/MVS-CIS: Converting, Indexing and Sorting print data

Specifies the level of trace information to be generated.

value Can be 1 to 5 numbers, where number has a range from 0 to 5.

Example: TRACELEVEL ( 1, 3, 5 )

TRCTYPE ( NO | IBM | SNI )

Specifies the kind of table reference characters used in the input file.

NO No table reference characters were used.
IBM IBM table reference characters were used.
SNI SNI table reference characters were used.

Example: TRCTYPE ( SNI )

TRIGGER ( triggername *, {column | *}, {‘value’ | x'value'} [, record, {column | *}, {‘value’ | x'value'}] [...])

Specifies the locations and values of data fields within the input file that are to be used to define indexing groups in the file, when processing S/370 line formatted data.

The trigger keyword is not allowed and causes a print file to be terminated if the file already contains Index Tag Element (TLE) structured fields.

triggername A 1 to 8 character name that specifies the trigger. This name will be used by the INDEX.

record | * Specifies the relative record number from the indexing anchor record. * indicates that every record should be checked. Supported range from 0 to 255. The first value has to be * and is used as anchor value. The other values mustn’t be *.

column | * Will be used as byte offset from the beginning of a record. If carriage control characters were used, column 1 refers to this. Supported values are 1 to 32756. * indicates that every column (starting with column 1) should be checked.
SPS/MVS-CIS: Converting, Indexing and Sorting print data

'value'

| x'value' | Specifies a constant value. This value can consist of character data or hexadecimal data. The length has a range from 1 to 250 for character data and 1 – 500 for hexadecimal data. Be careful this is a case-sensitive field.

Example: TRIGGER ( account, *, 1, '10Bank of', 7, 63, '00001' )

Refer to 3.2 Understanding SPS/MVS-CIS conversion and indexing process on page 31 for more information about the indexing process.

This parameter is rejected when specified together with the sorting schemes SCHEME_1, SCHEME_2, SCHEME_3 or 4UP_BOOKLET.

UNIQUEBNGS ( YES | NO )

Specifies whether CIS creates a unique group name or not.

YES CIS generates an 8-character string and appends it to the group name.

NO No string is appended.

Example: UNIQUEBNGS ( NO )

This parameter is ignored when specified together with the sorting schemes SCHEME_1, SCHEME_2, SCHEME_3 or 4UP_BOOKLET.

USEPAGENAMES ( YES | NO )

Specifies whether CIS generates page names using 8-byte counter or uses structured field tokens found in the input data stream. If the input data contains BPGs with FQNs, CIS does not generate page names.

YES CIS uses structured field tokens in the input data stream to generate page names.

NO CIS generates page names using an 8-byte counter.
Example: USEPAGENAMES ( YES )

**USERLIB ( data set name [ data set name ] [ ... ] )**

Specifies the data set names where CIS looks for the resources. CIS looks in the order the data set names are given. Any number of data set names can be specified.

The order CIS looks for definitions is:
- Inline resource group
- Userlibs if given
- Defined data set names

Example: USERLIB ( MY.USERLIB, YOUR.USERLIB )

**X2UP ( OFF | ON | LEFT | RIGHT | ICOPIES )**

Select whether the two up feature should be used. This feature is only available on Océ printers and is only supported by Océ software. Do not specify it unless the printer and driver to be used support it.

This parameter must be used together with RESTYPE( ALL ) or RESTYPE( FDEF ) otherwise it is ignored.

- OFF Set to one up.
- ON Use the two up feature.
- LEFT The same as ON.
- RIGHT Similar to ON except the page sequence on the paper is right -> left instead of left -> right.
- ICOPIES Use the two up feature to print two copies of each input page on each physical page.

Example: X2UP ( ON )

### 3.9
SPS/MVS-CIS Performance considerations

3.9.1 Memory requirements

- When converting line data
- When normalizing AFP data
- When indexing
- When retrieving resources
- When sorting
- When tracing

3.9.2 DASD requirements

- For index file
- For resource file
- For document file
- For SWAP file
- For trace files

3.9.3 CPU requirements

- When converting line data
- When normalizing AFP data
- When indexing
- When retrieving resources
- When sorting
- When tracing
4 Glossary

Some of the abbreviations and terms that appear in this glossary have been taken from other sources. They are provided as supporting information only.

A

Advanced Function Presentation (AFP). An IBM trademark. This term is used to refer to a presentation data stream. MO:DCA-P is the strategic AFP interchange data stream. IPDS is the strategic AFP printer data stream.

AFP. See Advanced Function Presentation.

AFPDS. See MO:DCA-P.

all points addressable (APA). The capability to address, reference, and position data elements at any addressable position in a presentation space or on a physical medium. An example of all points addressability is the positioning of text, graphics and images at any addressable point on the physical medium.

APA. See All point addressable printers.

C

CIS. See Conversion, Indexing and Sorting facility.

M

mailpiece. (1) A generic term used to refer to one or more sheets that have the same
addressed. (2) The "top level" page group(s) in a print file.


MO:DCA. See Mixed Object Contents Architecture.

MO:DCA-P. The subset of the MO:DCA architecture that defines presentation documents.

normalize. to return to a normal, usual or generally accepted condition. The AFP normalizer in SPS/MVS verifies the data quality of the input print file, and makes the necessary changes to guarantee the conformity of the output data to the MO:DCA standard.

physical medium. A physical entity on which information is presented. Examples of a physical medium are a sheet of paper, a roll of paper, an envelope, and a display screen.

S

sheet. A division of the physical medium; multiple sheets can exist on a physical medium. For example, a roll of paper might be divided by a printer into rectangular pieces of paper, each representing a sheet. Envelopes are an example of a physical medium that comprises only one sheet. A sheet has two sides, a front and a back side. Some sheets may only be printed on one side (i.e. overhead foils).

side. A physical surface of a sheet.

simplify. Make less complicated or less difficult. The optional AFP simplifier in SPS/MVS removes the existing AFP structures contained in the input print file, converting the file to a flat page sequence in preparation for a data enrichment or sorting process.

SPS. See Smart Printing Server.

Smart Printing Server (SPS). The Océ Printing Systems software product used to drive IPDS printers. It is part of the PRISMAproduction product family and includes other program features like data indexing and sorting.
5 Bibliography

- Printer Data Stream (SPDS) Model 4 Reference Manual
  U9737-J-Z247-X-7600

- SPS/MVS Installation Guide
  U20944-J-Z247-X-7600

- SPS/MVS-APA Messages Guide
  U21745-J-Z247-X-7600

- SPS/MVS-APA The PSORT User`s Guide
  U26433-J-Z247-1-7600

- SPS APA Type Fonts with 240 dpi Print Samples
  U23008-J-Z249-X-7400

- SPS APA Type Fonts with 300 dpi Print Samples
  U23009-J-Z249-X-7400

- Electronic Twin Printing System Operating Manual
  U7637-J-Z246-X-7600

- Electronic Twin Printing System Operator`s Reference Card
  U9739-J-Z248-X-7600

- Model 2,3 and 4 Electronic Printing System Operator`s Reference Card
  U8316-J-Z248-X-7600
Index

C
CC 65
CHARS 65
COLORMAP 66
concatenating CIS files 61

D
data flow
SPS/MVS-CIS 29

F
FDEFLIB 66
features
summary 11
FIELD 66
FONTLIB 67
FORMDEF 67

G
GROUPNAME 68

I
IEBGENER 61
IMAGEOUT 68
INBIN 68
INDEX 69
INDEXDD 69
INDEXOBJ 69
INDEXSTARTBY 70
INPUTDD 70

J
JCL requirements
CISSWAP dname 60
EXEC 57
IEBGENER 61
INDEX dname 58
INPUT dname 57
OUTPUT dname 59
RESOBJ dname 59
return code 60
SPSxCIS dname 60
STEPLIB dname 57
SYSIN dname 57
SYSPRINT dname 57
SYSTEM dname 57

L
LUPUB 70

O
OBJCONLIB 70
OFFSETXB 71
OFFSETXF 71
OFFSETYB 71
OFFSETYF 71
operating system requirements
MVS/XA 11
OUTBIN 72
OUTPUTDD 72
OVERLAYB 72
OVERLAYF 72
overview
SPS/MVS products 7
what is SPS/MVS 7
what is SPS/MVS-APA 8, 9
what is SPS/MVS-CIS 8, 27
what is SPS/MVS-LIP 8, 11
OVLYLIB 72

P
PAGEDEF 73
PDEFLIB 73
PRESENTATION 73
PRINTMODE 74
PSEGLIB 74
Index

R
RESFILE 74
RESOBJDD 75
resource packaging 37
RESTYPE 75
return code SPS/MVS-CIS 60

S
SORT 75
4-UP booklet sorting 51
4UP_BOOKLET 51
Fanfold listing 2-UP 48
Fanfold listing duplex 47
reverse printing sequence 50
SCHEME_1 47
SCHEME_2 48
SCHEME_3 50
SPS/MVS-APA 13
accounting 15
checkpointing 15
control file support 14
information pages 15
input data format 13
JCL support 14
RAS 16
resource processing 14
system parameters 15
trace function 16
user exits 15
SPS/MVS-CIS 11
consolidating documents 13
contents sorting 43
converting ASCII data 34
converting line data 12, 31
converting mixed data 34
creating an Index file 12
data flow 29
indexing 12
indexing ASCII data 34
indexing line data 32
indexing mixed data 34
input data formats 12
layout sorting 47
normalizing AFP data 34
RAS 13
report consolidation 55
resource data formats 12
resource retrieval 37
sort overview 12
sorting steps 42
trace function 13
SPS/MVS-LIP 17
accounting 19
checkpointing 18
control file support 17
information pages 18
input data format 17
JCL support 17
message generation 18
RAS 19
system parameters 18
trace function 19
user exits 19
system requirements 11

T
TRACEFLAGS 76
TRACELEVEL 76
TRCTYPE 77
TRIGGER 77

U
UNIQUEBNGS 78
USEPAGENAMES 78
USERLIB 79

X
X2UP 79
XA see operating system requirements