CIS
PRISMAproduction Host
5.0PRISMAproduction Server

Developer's Edition

User's Guide

18-May-09

...and Training?

Our training centers hold seminars
on the printing system described in
this manual. We can also conduct
operator workshops on your premises
or in Poing on request.

Information:
Phone +49 81 21 72-39 90
Fax   +49 81 21 77-39 50

Océ Printing Systems GmbH
Siemensallee 2
85581 Poing
Germany

Preface

This document is intended to provide information about the PRISMAproduction product family and in particular about the CIS product. It contains information that should help installations to convert existing print jobs into powerful and sophisticated AFP applications through the use of CIS.

This manual, and the reference documentation mentioned in this manual, provide a comprehensive, all-inclusive description of the functionality available in CIS. Any feature or function which is not explicitly documented in the PRISMAproduction Host literature is not supported.

It is assumed that the reader is familiar with the OS/390, z/OS or Linux environments and with Advanced Function Presentation (AFP) concepts.

Summary of Amendments

Summary of Amendments
for U28117-J-Z247-1-7600
for CIS-Module Version 4.00

- New product

Summary of Amendments
for CIS-Module Version 4.02

- Code Page support
- Insert BCOCA objects
- Insert PTOCA objects
- Data extraction
- Output segmentation
- Data Stream analyst

Summary of Amendments
for CIS-Module Version 4.04

- Support for Record-format Line data
- Support for Unicode (UCS-2) Line data
- Insert UP³I Finishing Operations such as Interposing, Folding, Stapling, Wrapping, Cutting and Punching
- Insert TIFF, JPEG, EPS, PCL, PDF and other paginated presentation objects as Object Containers
- Insert GOCA objects as lines, boxes, circles and ellipses
- New sorting algorithms for UNCOLLATED output stackers (face-up and face-down)
- Support for user defined COLORS and LEVELS

- Support for internal and external Resource Groups including Resource Consolidation
- Enhanced User Exit (plug-in) support
- Updated MO:DCA support

Summary of Amendments
for CIS-Module Version 4.06

- Support for XML data (XML print formatting via PAGEDEF)
- Support for
  - IBM 2D Bar codes (2DDATAMATRIX, 2DMAXICODE, 2DPDF417, 2DQRCODE)
  - CODE93 Bar codes
- UP^{3}I level 1.3 support
- CID font support
- Enhanced Indexing support
- Updated MO:DCA support
- New license level *Resource Packaging*

Summary of Amendments
for CIS-Module Version 5.00

1. **Updated MO:DCA** support.
   - **UTF-16 surrogate** support.
   - The following triplets and structured fields are now supported.
     - Structured fields: **BSG/ESG**: Begin/End Resource Environment
     - Structured field: **PPO**: Preprocess Presentation Object
     - Triplet **0x18**: Object Count in BDT structured field.
     - Triplet **0x1F**: Font Descriptor Specification in MCF2 structured field.
     - Triplet **0x20**: Font Coded Graphic Character Set Global Identifier in MCF, MDR structured fields.
     - Triplet **0x21**: Object Function Set Specification in BDT structured field.

- Triplet **0x27**: Line Data Object Position Migration in BBC, BGR, BII, BIM, IPS structured fields.
- Triplet **0x4E**: Color Specification in IOB, OBD, PGD structured fields.
- Triplet **0x50**: Encoding Scheme ID in MCF, MDR structured fields.
- Triplet **0x5D**: Font Horizontal Scale Factor in MCF structured field.
- Triplet **0x5E**: Object Count in BNG structured field.
- Triplet **0x62**: Local Date and Time Stamp in BBC, BDI, BFM, BGR, BIM, BMO, BOC, BPS, BPT, BRG structured fields.
- Triplet **0x63**: Object Checksum in BMO, BPS structured fields.
- Triplet **0x64**: Object Origin Identifier in BMO, BPS structured fields.
- Triplet **0x70**: Presentation Space Reset Mixing in IOB, OBD, PGD structured fields.
- Triplet **0x71**: Presentation Space Mixing Rules in IOB, OBD, PGD structured fields.
- Triplet **0x72**: Universal Date and Time Stamp in BBC, BDI, BDT, BFM, BGR, BIM, BMO, BOC, BPS, BPT, BRG structured fields.
- Triplet **0x74**: Toner Saver in PFC structured field.
- Triplet **0x80**: Attribute Qualifier in TLE structured field.
- Triplet **0x83**: Presentation Control in IEL, BNG, BPG structured fields.
- Triplet **0x86**: Text Fidelity in PFC structured field.
- Triplet **0x8B**: Data-Object Font Descriptor in MDR structured field.

- Support of **Multipage PDF/TIFF Container**
- Support for **OT/TT fonts**.
- Support for **CMR (RAT)**.

Restrictions:	Support for ObjectResources (RAT)

2. CIS **returncode changed**: If the inputfile contains only resources, CIS now returns rc=8 instead of rc=4

Return codes On termination, CIS passes a return code which summarises the errors detected during execution. The return codes (decimal) set by CIS are:

-4 Licence error detected.
PRISMAproduction server only !
On PRISMAproduction host return code 16 is set.

0 Successful execution. No errors were found.

4 Successful execution with warning messages.
8 Data error detected during execution. One or more error messages were written to `stdout` describing the problem found.
12 Keyword or File related error detected during execution. One or more error messages were written to `stdout` describing the problem found.
16 Program related error detected during execution. One or more error messages were written to `stdout` describing the problem found.

3. **Invalid TLEs** are ignored. TLE’s which are on a “wrong” place, f.e. between pages, are ignored.

4. Parameter **FONTMAP**: default *fontMapProcessing* behavior changed from ASIS to RESOLVE

   FONTMAP (*fontMapName* | <u>DUMMY</u> [, <u>RESOLVE</u> | ASIS])

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

   fontMapName The name can be one to eight alphanumeric characters, including the two-character prefix, if there is one. Specifying DUMMY (the default) requires the print file to contain at least one inline Font Mapping table. CIS uses the first Font Mapping table found and ignores all others.

   fontMapProcessing
   ASIS|<u>RESOLVE</u>

   ASIS Causes CIS to leave all fonts requests (including those made via GRID) unchanged. The Font Mapping table is also left unchanged.

   <u>RESOLVE</u> Causes CIS to process the Font Mapping table and to collect the substitution fonts. The font invocations are changed to use the substitution fonts.

5. Parameter **DEFINEBCOCA**: it is now checked whether the in the DEFINEBCOCA specified CodePageId is valid for the also in the DEFINEBCOCA specified Barcodetype.
   In case if not, a message will be sent and the processing is stopped.

6. **Location “END”** is allowed now in conjunction with **INSERTINDEX**
   examples: `DefineLevel( myLevel, DOCUMENT, END, ALL )`
   inserts a group level TLE in the last MP of a document
   `DefineLevel( myLevel, GROUPLEVEL1, END, ALL )`
   inserts a page level TLE in in the last sequential page of a mail piece.

```
DefineLevel( myLevel, SHEET, END, ALL )
```
inserts a page level TLE in the last sequential page of a sheet.

7. New parameter **CASECTRL**

CASECTRL ***(0, 1, 2, 3)***

Controls the spelling of resource names and extensions before searching the file in the attached resource libraries.

0 read the resource name and the extensions in lower case

1 read the resource name and the extensions in upper case
default on PRISMAproduction host

2 do not convert the spelling of the resource name and the extensions
default on PRISMAproduction server

3 at first the resource name and the extensions are read in upper case and afterwards in lower case (combination of "0" and "1")

Please note that directory names (specified with the library parameters USERLIB, FDEFLIB, FONTLIB, etc.) are always handled without converting the spelling.

Summary of Amendments
for CIS-Module Version 5.02 (PRISMAproduction server only !)

1. A **new licensing model** is served from CIS 5.02 on PRISMAproduction
2. **Reading resources via FTR** library is served from CIS 5.02 on PRISMAproduction

Therefore for the path given for the libraries where resources should be found (params: `FDEFLIB`, `FONTLIB`, `OBJCONLIB`, `OVLYLIB`, `PDEFLIB`, `PSEGLIB`, `USERLIB`) from CIS5.02. the PRISMAproduction path conventions must be observed.

Resource handling see chapter "APA Settings Overview" in the PRISMAproduction APA-Module Application Guide.

Crossing from FTRlib documentation:
fully qualified path, optionally preceded by sever name.
The files are searched under '/u/prismapro/' respectively in the structure beneath. The part '/u/prismapro(/)' at the given path is optional.
for socket port number: optionally preceeded by server name.

examples:	pc25:/u/prismapro/data/infile"	::= "/u/prismapro/data/infile" at pc25
	"/u/prismapro/data/infile"	::= "/u/prismapro/data/infile" local
	"exitdata/infile"	::= "/u/prismapro/exitdata/infile" local
	"9876"	::= Socket port number 9876 local
	"pc25:9876"	::= Socket port number 9876 at pc25

3. **Multiple USERLIB** commands are supported now. (> CIS_5.02.03)
4. Several **extensions** are done **in analyst information file**.
5. Various amendments due to **support of PRISMAproduction Preflight&MakeReady**

Contents

1 Introduction to PRISMAproduction

1.1 Automated Document Processing

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.

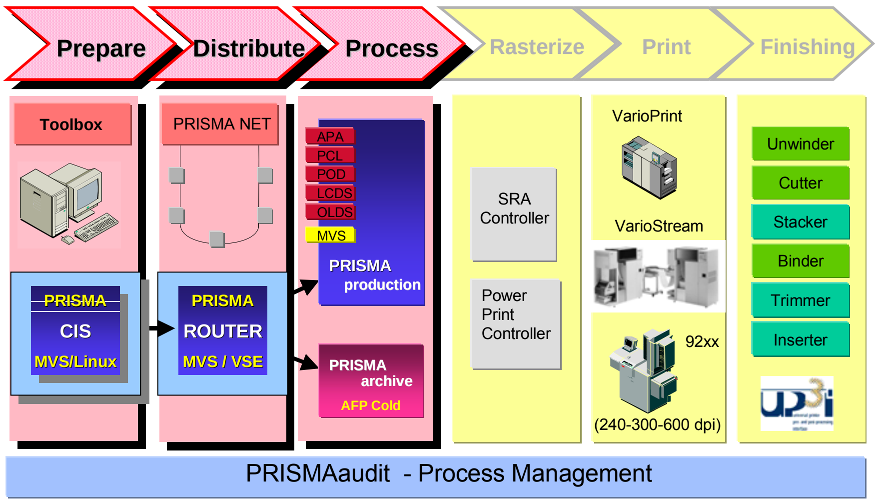


Fig. 1 Automated Document Processing

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.

Implementing new format and document presentation strategies can be very difficult, especially 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 CIS help application developers to optimize their work.

1.2 Component overview

PRISMAproduction Host is part of the PRISMAproduction software 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, Linux and Windows. It is a key component of this family, providing powerful and sophisticated technologies that help a large number of high-volume printing and mailing installations to satisfy their demands on faster and more flexible tools.

PRISMAproduction Host is a generic name used to refer to the following products:

- CIS
- Router
- SPS
- Model2-to-AFP

The **CIS** module is a high performance data manipulation tool that can be used to convert, index, sort, enrich 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 CIS to transform existing legacy applications into hierarchically structured documents. The sorting capabilities offered by CIS may substantially reduce mailing and other post-processing costs. The resource consolidation function allows an installation to archive not just the print documents but also the resources they use, enabling exact document reproduction even years later. The input selection, output segmentation and data enrichment functions provided by CIS make it the ideal tool whether the intention is to select specific pages for re-print, add marks and bar codes for controlling the post-processing equipment or to prepare the data for electronic distribution.

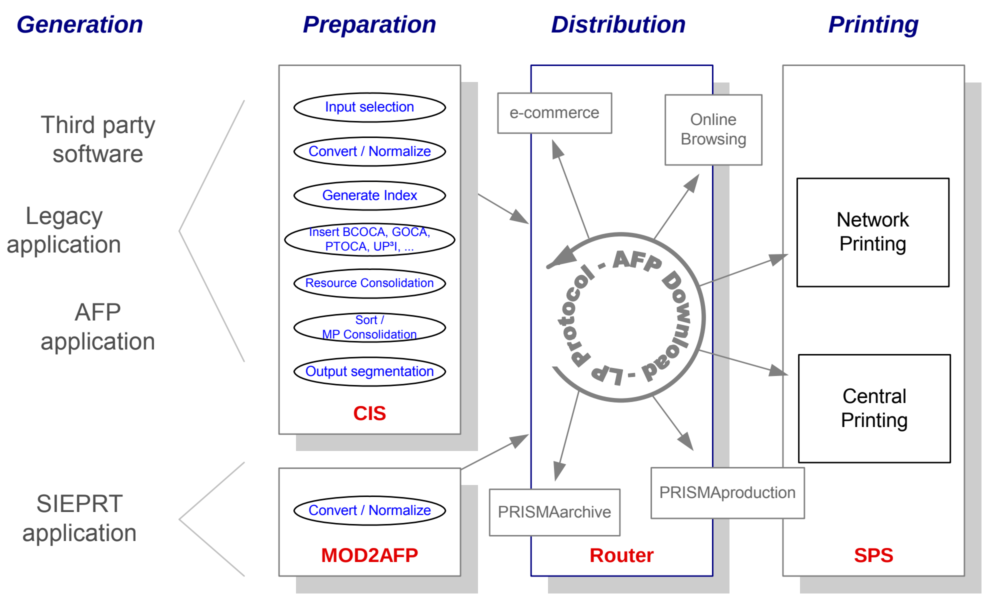


Fig. 2 Overview of PRISMAproduction Host

The **Router** module is a print data stream file transfer product that transmits print output from an OS/390 or z/OS system to a server system (e.g. PRISMAproduction or InfoPrint Manager) for printing, converting or archiving the data. The Router module supports print data stream transmission using the TCP/IP based “AFP Download” protocol. The transmission process is specially tuned for print data and includes facilities such as automatic submission of job parameters to the server, automatic checkpoint / restart, automatic file routing, user notification and job accounting. In combination with CIS, the PRISMAproduction Router can be configured to check for AFP data and to automatically pack all resources used by the application in the same file transfer step. In addition to AFP data, the Router module can be used to transfer any other print data stream supported by the spooling subsystem, including S/370 Line format data, Record format data, PCL, PostScript, etc.

The **SPS** (Smart Print Subsystem) module is a high performance printing subsystem that drives Océ and other IPDS compatible single and A-twin printers under OS/390 and z/OS systems. SPS accepts AFPDS and line format 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.

The **MOD2AFP** (Model 2 to AFP) module is a migration tool that helps customers converting from a SIEPRT/MVS configuration to AFP. Customers who want to move from the E-mode (Model 2) environment to AFP can use MOD2AFP in combination with SPS and be able to print legacy SIEPRT applications intermixed with modern AFP applications on the same printer. MOD2AFP in combination with Router will allow installations to send their output applications to distributed print centers such as PRISMAproduction Server.

2 Overview

2.1 Functional overview

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

- Select one or more portions of the print stream for processing. Input selection can be done based on page ranges, index contents and mail piece size among others.
- Convert and normalize the output of existing legacy applications without having to change the applications program.
- 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).
- Enhance the appearance and processing capabilities of the print stream by inserting and/or removing text blocks, bar codes, graphics, object containers, finishing operations, Index information or interposing extra sheets.
- Sort the print file pages using either a contents oriented (index), layout oriented or a collated algorithm.
- Segment a big print stream into multiple files by using different segment size criteria's such as number of bytes, pages, sheets, mail pieces or documents.
- Create a list with data extracted from every document, mail piece, sheet or page in the file.
- Invoke the Print Stream Analyzer and create a summary report containing statistics about the print file.

Each of these 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 CIS.

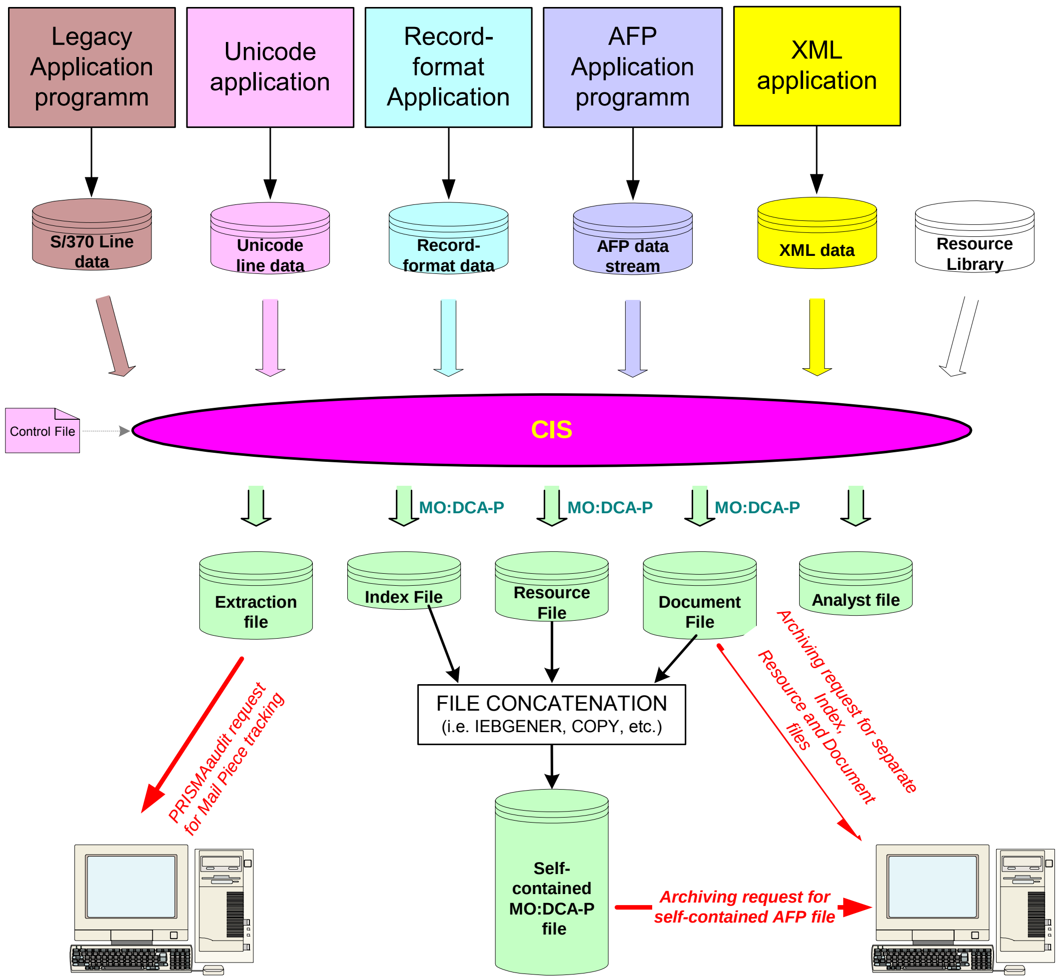


Fig. 3 CIS Overview

2.2 Benefits

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

- Convert S/370 line data, mixed data from legacy applications, Record-format line data, Unicode line data or XML data into MO:DCA-P. The conversion process resolves all line data relevant questions (e.g. 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.
- Select one or more portions of the print data stream for processing. Input selection can be done using criteria's such as page ranges, index contents and mail piece sizes. CIS restricts the processing of the print file to the selected pages, allowing installations to efficiently process (re-)print requests and at the same time restrict the scope of the resource retrieval process to the selected pages.
- Extract variable text strings -part of the 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 CIS, the index information can also be used to reorder the print job pages (e.g. by postal code) and/or to consolidate several different print files into a single one.
- Create an Index Object file based on the index tags contained (or inserted) in the input file. The index file created by 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.
- Enhance the appearance and processing capabilities of a print file by inserting and/or removing blocks of Text information and Bar codes. User defined strings and internal processing variables provided by CIS such as "current date" and "page x of y" can be used during this process.
- Enhance the appearance and processing capabilities of a print file by inserting and/or removing graphics in form of lines, curves, boxes, circles or ellipses. Enrich the original application by inserting OCA and non-OCA (JPEG, TIFF, EPS, etc.) object containers. Add UP³I Finishing operations such as stitching, cutting or binding.
- Extract data contained in the Index tags of the pages selected for processing and store them in a separate "data extraction" file.

- Sort the input pages using a user specified criteria. The sorting criteria can be based on the Index information (e.g. postal code, customer name, etc.) or on pre-defined page re-ordering layouts such as 2UP Booklet and Collated face-up.
- Split a big print file into multiple smaller files using different segmentation criteria's such as number of bytes, pages, sheets, mail pieces or documents. Each segment fulfils then special shipping requirements (e.g. weight or physical dimensions) or can be used as basis for electronic distribution (e.g. each segment contains one mail piece).
- Obtain the characteristics of an "unknown" print file. The Analyser function in CIS creates a summary report describing the attributes and printer requirements of the print file. The summary reports includes information such as file size (in pages, sheets and mail pieces), input bin requirements, output bin requirements, color requirements, etc.
- A powerful keyword interface allows an installation to use one or more of CIS functions in a single run. Activating several functions at once simplifies job processing and reduces the overall elapsed time, causing 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 CIS processing with special requirements, eliminating wasted run-time.
- A powerful user exit (plug-in) facility allow installations to customize the functionality provided by CIS beyond the options available in the Control file.

2.3 Features Summary

CIS print data stream processing can be divided into three major categories: data stream processing, installation specifications and RAS[1] facilities. The main elements of these functions are listed in the following section.

- DATA STREAM PROCESSING

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

[1] Reliability, Availability, Serviceability

    - S/370 Line format data, including Mixed-mode Line data
    - Record-format Line data
    - Unicode (UCS-2) Line data
    - XML data
    - AFP and MO:DCA-P data
    - SPDS data that conforms to the specifications described in [20].(see *8 Bibliography* on page 465).
    - Unformatted ASCII data as defined in this manual. ANSI carriage controls using ASCII coding are supported.

- **Resource Data Stream Processing:** Support the following AFP resource types:

    - Form definitions (FORMDEF)
    - Page definitions (PAGEDEF)
    - Page segments
    - Overlays
    - Fonts (Coded Fonts, Font characters sets, Code pages)
    - Object Containers (OCA and non-OCA objects)

  Only the resources required by a print file are read and copied into the output resource file using filtering keywords 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.

- **Resource consolidation:** Multiple input resource groups are supported. The resource consolidation function combines all the input resources in the different resource groups (internal and external) in a single external resource group. Naming conflicts are automatically avoided by renaming the resources.

- **Data selection:** Specific page, sheet or mail pieces may be selected for processing. The selection can be done based on ranges (e.g. page number) or size (e.g. all mail pieces with less than 10 sheets).

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

- **Index Tag generation:** 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 or MO:DCA-P 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.

- **Data enrichment:** The appearance and contents of a page can be enhanced using the insert/remove capabilities offered for Text blocks, Bar codes, Graphic images, Object Containers and Finishing Operations. User defined strings and a range of processing variables are available in order to give an installation full control over the print data stream without having to modify the application programs.

- **Sorting function:** 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 and Collating sorting algorithms help reducing mailing and post-processing time and cost.

- **Document consolidation:** Consolidating several indexed documents is not a problem. The Mail Piece and Resource consolidations features combined with the sorting function, allow an installation to merge two or more print files into a single one, consolidating the mail pieces addressed to a single destination.

- **Data extraction:** CIS allows an installation to extract information included in the print file as index tags (TLE's) and to store them into a separate sequential file.

- **Output segmentation:** CIS allows an installation to split an existing print file into two or more segments by using size criteria's such as number of sheets and/or mail pieces.

- **Data stream analysis:** The data stream analyzer function provided a summary report containing information about the print file such as size, AFP features in use, etc. This information can be very helpful when scheduling the file for print or later when retrieving it from an archiving system.

- <u>INSTALLATION SPECIFICATIONS</u>

  - **License Key:** CIS requires a License Key that is to be applied at installation time.

- **User exits (plug-in):** The following exit points are defined:
  - INPEXIT: Modify, delete or insert records to the Input data stream.
  - RESEXIT: Select which resources should be included in RESOBJDD
  - INDEXEXIT: Modify Index tags for clean-up and standardization patterns
  - SORT: Implement user specific sorting algorithms
  - INSERTBCOCA: Implement user specific BCOCA generation algorithms
  - INSERTINDEX: Implement user specific INDEX generation algorithms
  - INSERTPTOCA: Implement user specific PTOCA generation algorithms
  - INSERTCONTAINER: Implement user specific Object Container generation algorithms
  - INSERTFINISHINGOP Implement user specific UP^{3}I insertion algorithms

- **RELIABILITY, AVAILABILITY, SERVICEABILITY (RAS)**

  - **Dumping:** Under MVS, 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 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.

2.4 Operating System Requirements

2.4.1 OS/390 and z/OS

This version of CIS will run on MVS systems at the release levels listed below.

- **OS/390 Version 2.1.3 and above (not available for testing)**
- **z/OS Version 1.9 and above**

2.4.3 LINUX

This version of CIS will run on any of the LINUX platforms (Intel based) listed below.

- **SUSE Linux Enterprise Server 10 SP2**[1]

2.5 Restrictions

2.5.1 Unsupported MO:DCA functions

1. Segmented structured fields are not supported. The SegFlag bit in the Structured Field Introducer controls this function. Refer to MO:DCA Reference page 20 for more information.
2. The following triplets and structured fields are not supported. They cause CIS to issue an error message and to terminate the job.

    - BCA/ECA: Begin/End Color Attribute table
    - MCA: Map Color Attribute table
    - CAT: Color Attribute table
    - LLE: Link Logical Element
    - IPG: Include Page
    - MPG: Map Page
    - BDT/EDT: Begin/End Document when found inside a Resource Group
    - MCF: Map Code Font with a triplet x'02' (Fully Qualified Name) of type x'84' for Coded Font reference (GRID).

[1] Please refer to release note for exact kernel version number.

- MDR: Map Data Resource with a triplet x'02' (Fully Qualified Name) of types x'BE' and x'DE' for internal and external resource references in a data object.

3. The following triplets are not supported. They are not propagated (not forwarded) by CIS to the output data stream.

   - 0x56: Medium Map Page Number Triplet in structured fields BNG and BPG.
   - 0x63: Object Checksum triplet in all structured fields except BMO, BPS.
   - 0x64: Origin Identifier triplet in all structured fields except BMO, BPS.
   - 0x65: Comment Triplet in structured fields BAG, BDE, BDT, BFM, BMM, BNG, BPG and BRG.
   - 0x81: Page Position Information triplet in structured field BPG.

4. The No-Operation (NOP) structured field may or may not be propagated, depending on the kind of data being processed. More information can be found under *3.6 How to convert, normalize and index data*.

2.5.2 Differences with previous versions (before CIS V4.04)

1. The INDEXCONSOLIDATION keyword (available in CIS V4.02) has been replaced with the new MPCONSOLIDATION keyword.
2. The default DDnames for the INPUTDD, OUTPUTDD, RESOBJDD, INDEXDD and TRACEDD keywords have been changed to match the PRISMAproduction Host conventions.
3. The program name was changed from SPSPCIS to CIS. An alias with the name SPSPCIS is provided for compatibility reasons.
4. The name of the exit programs specified in the CIS control file must be enclosed in apostrophes. An exception is made under MVS in case of the INPEXIT keyword, where CIS also accepts a name without apostrophes. This is done for compatibility with version 4.00

2.6 Samples

CIS is distributed with the following set of samples in library hlv.qualifier.PPHSAMP:

CISIVP	Installation verification procedure

CISSKEL	Skeleton JCL for running CIS
CISENRCH	Data enrichment by inserting page numbers as ptoca (text) and bcoca (bar code)
CISESSEL	Entity sized based selection (only mailpieces with a defined number of sheet are written to output file)
CISISEL	Index based selection (only mailpieces with a defined index are written to output file)
CISOSEGM	Output Segmentation
CISRES	Resource consolidation and Analyst file
CISRSEL	Entity range based selection (select pages, mailpieces, sheet or documents which shall be written to output by number)

3 Exploring CIS

3.1 Internal Workflow

This section describes the CIS workflow and the sequence in which the different data manipulation phases are applied. In order to effectively use CIS, it is important to understand its internal workflow.

Fig. 4 on page 32 depicts the internal CIS workflow. One single CIS run may invoke one or more print stream conversion and processing features resulting in additional flexibility and reduced overhead as new applications are built. CIS allows installations to:

- Select one or more portions of a print file for processing. Whether the input print file contains S/370 Line format data, Record-format line data, XML data or it is already in MO:DCA-P format, the input selection criteria's available in CIS can be used to generate partial re-prints based in either ranges (from-to page, sheet, mail piece or document), index contents or even size (e.g. mail pieces with no more than 10 sheets).

  The selection process is done as early as possible within this workflow. All the processing steps that follow will only apply to the selected data. Index range based selection is done next, at the end of the normalization process. Size based selection requires a temporary work file and is done as last selection step.

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

  Normalized documents become platform independent and can be easily moved across different system platforms. CIS guarantees a normalization process that is identical to the process done by SPS when Line format data is directly printed.

- Extract variable text included in the input data and use it to build Index Tags. Indexing a print job enhances the ability to view, archive or retrieve specific pages (or group of pages) contained in the file. The powerful keyword interface available in CIS allows an installation to specify an unlimited number of indexing tags to be obtained from the print data.

The Mail Piece consolidation feature of CIS is capable of converting Line format data files into hierarchically structured, multilevel, AFP Documents. A structured document is the base for any subsequent print stream processing to be done with CIS including functions such as data enrichment, sorting, post processing optimization, output segmentation and others.

- Enhance the appearance and processing capabilities of a print file adding finishing control and document tracking information. This data enrichment option of CIS allows an installation to insert text blocks (PTOCA structures), index tags, bar codes (BCOCA structures), graphics (GOCA structures), images (Object containers) and finishing operations (UP³I extensions) at a page, sheet, mail piece or documentlevels. User defined strings and CIS processing variables are available during this process and can be used to build the data to be inserted. User specified levels can be used to control what is to be enriched, where is the enrichment operation to be applied and to which entities. All structure fields added by CIS contain a user specified name field. User exits (plug-ins) are also available and may be used for more flexibility.

  CIS also allows an installation to remove existing text blocks, index tags and bar codes provided that the data structures to be removed were clearly labeled by the data generation tool and that the labels are consistently used in the whole print file.

- Generate an Index Object File using the Index Tags inserted during the conversion or normalization process. S/370 line data, Record-format and XML applications get their tags inserted by 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 CIS in a previous run.

  Regardless of their origin, CIS uses these tags to generate the Index Object File. The Index Object 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). 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 mail pieces in the input print file using either a "contents based", "layout based" or "collating" 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. The collating function may be used in connection with special output stacking devices.

Sorting a print file can save significant post-processing time, reducing postage and mailing costs and eliminating manual sorting. The 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.

- Segment a large print file into smaller units to facilitate post-processing and/or shipment. A segment size of one mail piece is also supported and can be used to generate files that can be electronically distributed for on-line viewing.
- Analyze the data and generate a summary report describing the main characteristics of the print file. The information in the report can be used to schedule print jobs based on their size, complexity or special features (e.g. number of input/output bins or color).

  As all other CIS keywords, only the pages selected for processing are analyzed.

In summary, there are many reasons for using 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, 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.

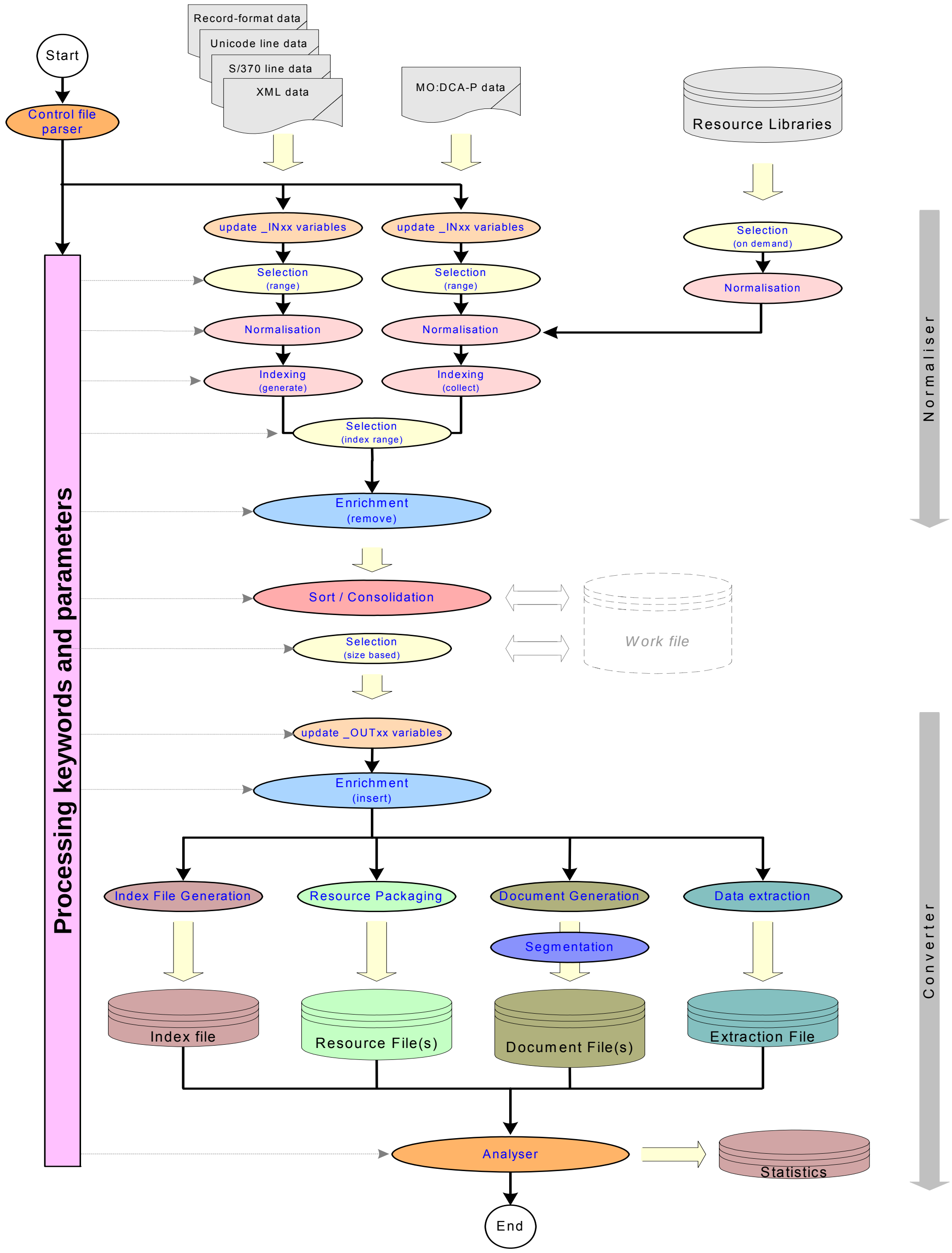


Fig. 4 CIS internal workflow

3.2 Code page support

Modern print solutions are often implemented using components that run under different system platforms, where the data may be generated under OS/390, printed on a remote location using a UNIX based server and archived or viewed under Windows.

In most cases, each of these platforms uses a different character encoding scheme, at least as default. EBCDIC, ASCII and UNICODE are just examples. Transferring the application files (print data and resources) in binary format guarantees that they remain printable through the whole process. Unfortunately, binary file transfer does not solve the problems associated with other data manipulation tasks that may be required to implement a flexible and integrated solution. Whether selecting print data, inserting text strings or bar codes, extracting data from a file, or even performing a contents based sort, all these processes require a good understanding of the character encoding used in the input data being processed and in the output data to be generated.

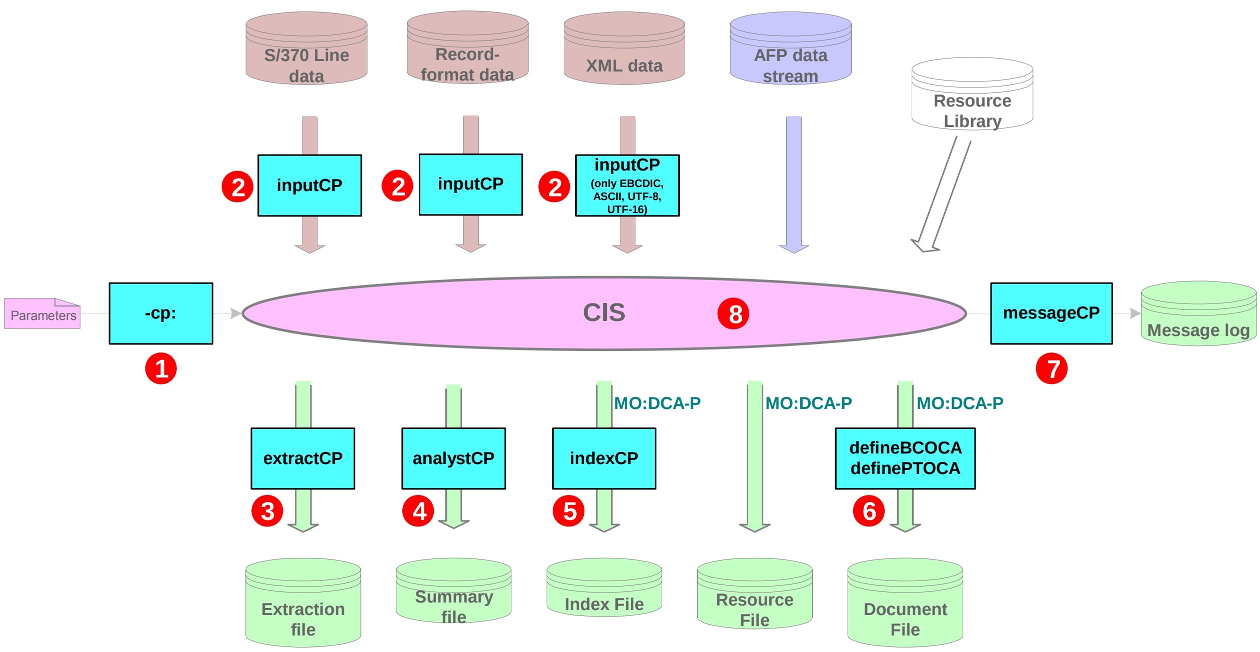


Fig. 5 CIS Code Page support

The Code Page support available in CIS is designed to solve this problem, offering a very high degree of flexibility regardless of the origin, transit and target platforms. A set of commonly used standard code pages is available (see also *6.4 Appendix D: Code Page Description* on page 362). Fig. 5 provides an overview of the Code Page translation options available in CIS; Fig. 6 shows the different places within the CIS workflow where a code point translation takes place. The Code page options available in CIS are:

1. **Code Page used in CIS Control file (CP command line keyword)**

   The **CP** command line keyword (see *4 Using CIS* on page 147) specifies the character encoding used to generate the CIS Control file. In most cases, the encoding will be the same as the CIS internal Code Page (default, see point 8 below) and there is no need to use this keyword. Any of the 8-bit encoding schemes (see *3.2.1 Supported Code Page IDs* on page 38) may be specified. The UTF16BE, UTF16LE and UNKNOWN code pages are not supported.

2. **inputCP keyword**

   The character encoding of an input file in S/370 Line format, Record-format data, XML data or AFP Mixed data format can be specified using the **inputCP** keyword in the CIS control file. This keyword must be used when inserting Index Tag Elements to input files generated on a different platform as where CIS is running, or where the input file contains special characters (such as €, ä, ö, ü, Ñ, ĉ, £, ß, etc.). The default is UNKNOWN which indicates that no code page conversion of the input data is to be done by CIS.

   In most cases CIS does not translate the print data itself when writing the output document file, and **the print data is written using the same character encoding as it was read.** The only exception is code page UTF16LE which is always converted to UTF16BE. CIS needs to know the character encoding scheme of the input data in order to convert the strings specified in the TRIGGER keyword to the appropriate code page or when converting the input data to other encoding schemes such as **indexCP, extractCP, defineBOCA** or **definePTOCA** keywords.

   XML data may only be encoded using

   - EBCDIC (Single-byte only)
   - ASCII (Single-byte only)
   - UTF-8
   - UTF-16
   - UTF16LE
   - UTF16BE

3. **extractCP keyword**

   The character encoding to be used in a Data Extraction file generated by CIS can be controlled using the **extractCP** keyword. Any data string written to the data extraction files will use this code page. Correct data extraction in CIS requires the correct specification of the character encoding scheme used in the Index Tag elements being processed. The default is to use the CIS internal Code Page (see point 8 below).

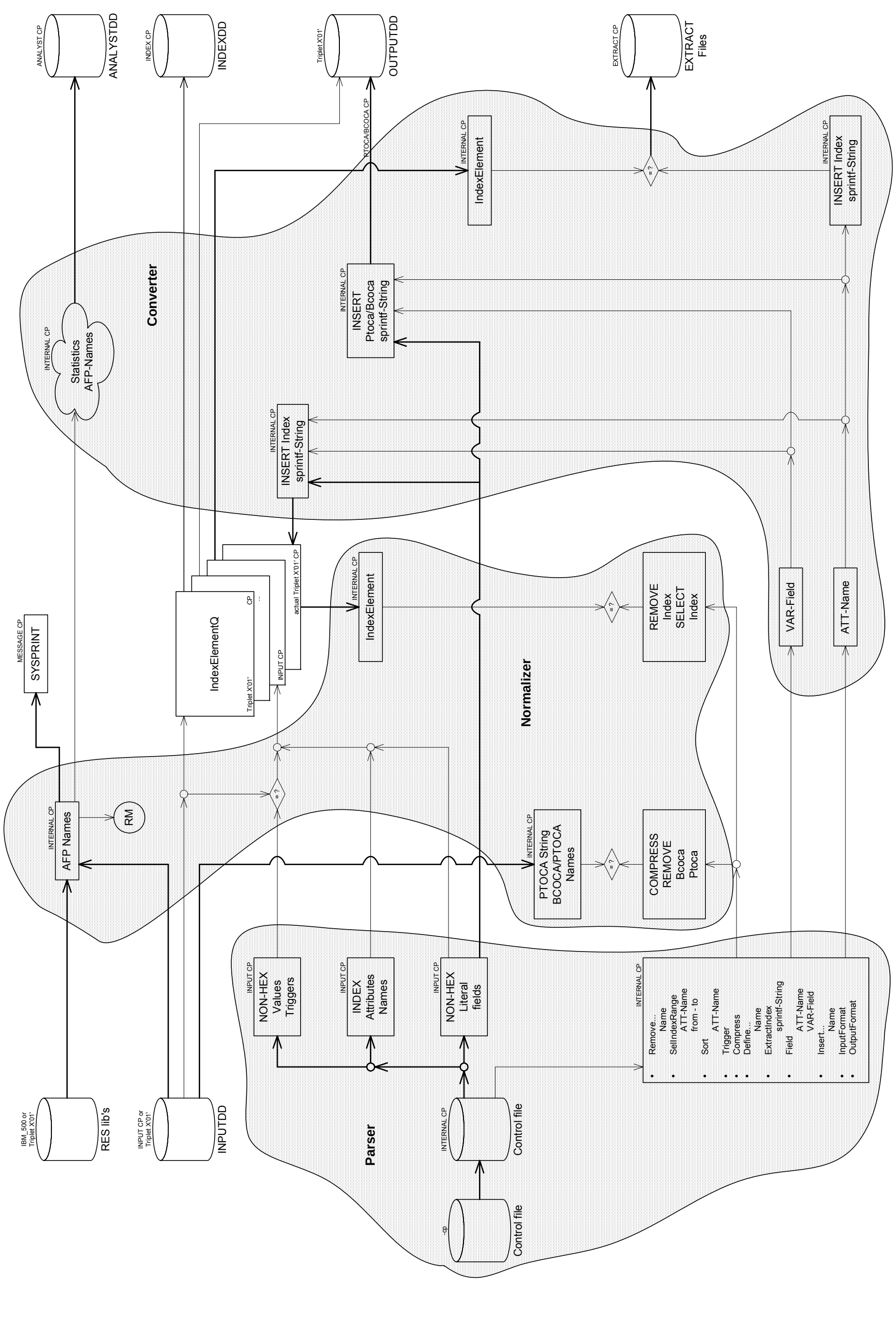


Fig. 6 CIS Transcoding operations

4. **analystCP keyword**

   The character encoding to be used in a Print Analyst file generated by CIS can be controlled using the **analystCP** keyword. The default is to use the CIS internal Code Page (see point 8 below).

5. **indexCP keyword**

   The character encoding to be used in the Index Tag Elements that CIS writes into the Index Object file can be controlled using the **indexCP** keyword. All index values and attribute names contained in the Tag Logical Elements (TLE) written by CIS in the Index Object File will be converted to use this code page. The default is to use the same Code Page as specified for the **inputCP** keyword. **This keyword does not affect the code page used in the TLE's written into the Output Document file.**

   A Coded Graphic Character Set Global Identifier Triplet 0x01 is added to each TLE and Index Element (IEL) structured field inserted in the current CIS run. Existing TLE's, if any, are only modified if they do not include a triplet 0x01, otherwise they are left unchanged.

6. **defineBCOCA and definePTOCA keywords**

   The character encoding to be used in the Text and Bar Code elements inserted by CIS can be controlled using the **defineBCOCA** and **definePTOCA** keywords.

   All BCOCA bar codes require a pre-defined encoding that cannot be changed by the user. By default, CIS will convert the data to be displayed into the correct code page as specified in the BCOCA architecture. The codePageID parameter of the defineBCOCA keyword allows an installation to override this default or to define the encoding to be used in case of a non-standard (numeric) bar code type. Correct Bar Code generation in CIS requires the correct specification of the character encoding scheme used in the input data being processed.

   In the case of PTOCA strings, the codePageID parameter of the definePTOCA keyword may be used to specify the code page to be used. The default is to use the CIS internal Code Page (see point 8 below).

7. **messageCP keyword**

   The messageCP keyword can be used to control the Code Page that CIS should use when generating the message file. The default is to use the CIS internal Code Page (see point 8 below).

8. **CIS internal Code Page**

   The CIS internal Code Page is platform dependent. It is the default for most Code Page keywords and parameters when nothing else is specified. The CIS internal Code Pages are:

- Under OS/390 and z/OS: IBM_1148
- Under Linux: ISO_8859-15

The CIS internal Code Page is hard coded within the product and is independent of any code page selection mechanism provided by the operating system platform.

3.2.1 Supported Code Page IDs

The standard code pages identifiers available in CIS are listed below. Please refer to on *6.4 Appendix D: Code Page Description* page 362 for a detailed description of each of these code pages.

Keyword	Description	CPGID
• IBM_0037	EBCDIC (USA and Canada)	00037
• IBM_0256	EBCDIC Netherlands	00256
• IBM_0273	EBCDIC Austria, Germany	00273
• IBM_0277	EBCDIC Denmark, Norway	00277
• IBM_0278	EBCDIC Finland, Sweden	00278
• IBM_0280	EBCDIC Italy	00280
• IBM_0284	EBCDIC Spain, Latin America (Spanish)	00284
• IBM_0285	EBCDIC United Kingdom	00285
• IBM_0290	EBCDIC Japanese Katakana	00290
• IBM_0297	EBCDIC France	00297
• IBM_0367	ANSI X3.4 ASCII Standard USA	00367
• IBM_0420	EBCDIC Arabic	00420
• IBM_0423	EBCDIC Greek	00423
• IBM_0424	EBCDIC Hebrew	00424
• IBM_0500	EBCDIC International Latin-1	00500
• IBM_0813	ISO 8859-7 Greek/Latin	00813
• IBM_0819	Same as ISO_8859-1	00819
• IBM_0833	EBCDIC Korean	00833
• IBM_0836	EBCDIC South-China	00836
• IBM_0838	EBCDIC Thai	00838
• IBM_0850	ASCII PC-Data-190: Latin Alphabet Number 1	00850
• IBM_0851	ASCII PC-Data Greek	00851
• IBM_0852	ASCII PC-Data Latin-2 Multilingual	00852
• IBM_0855	ASCII PC-Data Cyrillic	00855
• IBM_0856	ASCII PC-Data Hebrew	00856

	Keyword	Description	CPGID
•	IBM_0857	ASCII PC-Data Turkey Latin-5	00857
•	IBM_0860	ASCII PC-Data Portugal	00860
•	IBM_0861	ASCII PC-Data Iceland	00861
•	IBM_0862	ASCII PC-Data Hebrew (Migration)	00862
•	IBM_0863	ASCII PC-Data Canada	00863
•	IBM_0864	ASCII PC-Data Arabic	00864
•	IBM_0865	ASCII PC-Data Denmark, Norway	00865
•	IBM_0866	ASCII PC-Data Cyrillic, Russian	00866
•	IBM_0869	ASCII PC-Data Greek	00869
•	IBM_0870	EBCDIC Latin-2 Multilingual	00870
•	IBM_0871	EBCDIC Iceland	00871
•	IBM_0874	ASCII PC-Data Thai	00874
•	IBM_0875	EBCDIC Greek	00875
•	IBM_0880	EBCDIC Cyrillic Multilingual	00880
•	IBM_0893	OCR B	00893
•	IBM_0905	EBCDIC Turkey Latin 3 Multilingual	00905
•	IBM_0912	ASCII Latin 2 ISO 8859-2	00912
•	IBM_0915	ASCII Cyrillic ISO 8859-5	00915
•	IBM_0916	ASCII Hebrew ISO 8859-8	00916
•	IBM_0920	ASCII Turkey ISO 8859-9 Latin 5	00920
•	IBM_0924	same as ISO 8859-15	00924
•	IBM_1009	ASCII ISO-7 IRV (prior 1992)	01009
•	IBM_1010	ASCII ISO-7 France	01010
•	IBM_1011	ASCII ISO-7 Germany	01011
•	IBM_1012	ASCII ISO-7 Italy	01012
•	IBM_1013	ASCII ISO-7 United Kingdom	01013
•	IBM_1014	ASCII ISO-7 Spain	01014
•	IBM_1015	ASCII ISO-7 Portugal	01015
•	IBM_1016	ASCII ISO-7 Norway	01016
•	IBM_1017	ASCII ISO-7 Denmark	01017
•	IBM_1018	ASCII ISO-7 Finland and Sweden	01018
•	IBM_1019	ASCII ISO-7 Belgium and Netherlands	01019
•	IBM_1025	EBCDIC Cyrillic Multilingual	01025
•	IBM_1026	EBCDIC Turkey Latin 5	01026
•	IBM_1027	EBCDIC Japanese Latin	01027
•	IBM_1046	ASCII Arabic Windows	01046
•	IBM_1089	ASCII ISO-8859 Arabic	01089
•	IBM_1097	EBCDIC Farsi	01097

Keyword	Description	CPGID
• IBM_1098	ASCII Farsi Personal Computer	01098
• IBM_1140	same as IBM_0037 (Euro-Sign replaces Code Pos 9F)	01140
• IBM_1141	same as IBM_0237 (Euro-Sign replaces Code Pos 9F)	01141
• IBM_1142	same as IBM_0277 (Euro-Sign replaces Code Pos 5A)	01142
• IBM_1143	same as IBM_0278 (Euro-Sign replaces Code Pos 5A)	01143
• IBM_1144	same as IBM_0280 (Euro-Sign replaces Code Pos 9F)	01144
• IBM_1145	same as IBM_0284 (Euro-Sign replaces Code Pos 9F)	01145
• IBM_1146	same as IBM_0285 (Euro-Sign replaces Code Pos 9F)	01146
• IBM_1147	same as IBM_0297 (Euro-Sign replaces Code Pos 9F)	01147
• IBM_1148	same as IBM_0500 (Euro-Sign replaces Code Pos 9F)	01148
• IBM_1149	same as IBM_0871 (Euro-Sign replaces Code Pos 9F)	01149
• IBM_1303	Code 128	01303
• ISO_646	ISO 646 IRV:1991 (US ASCII, 0x00-0x7F)	
• ISO_8859-1	West European - Latin1 (Unicode 3.0, U+0000 - U+00FF)	00819
• ISO_8859-15	West European - Latin 15 (includes Euro sign)	00924
• PC_437	IBM PC (DOS Latin US)	00437
• PC_850	IBM PC (DOS Latin 1 Western Europe)	00850
• PC_1250	MS-Windows Latin-2	01250
• PC_1252	MS-Windows Latin-1	01252
• UNKNOWN		65285
• UTF16LE	Unicode Transformation Format 16b, little endian	65286
• UTF16BE	Unicode Transformation Format 16b, big endian	65287

3.2.2 Code Page Processing notes

The following notes apply to the code page processing done in CIS for the operations described in Fig. 5 and Fig. 6. They do not apply to the input data normalisation and conversion process, which is code page independent.

1. The input control file must be encoded in one of the 8-bit encoding schemes supported by CIS (see *Code Page used in CIS Control file (CP command line keyword)* on page 34).
2. The code page support currently implemented in CIS is intended to be used for converting the different LATIN encoding schemes available (EBCDIC, ASCII, US, Canada, Latin America, Western Europe).

Using other code pages is possible, however, the code points used must be restricted to those which have an equivalent representation in the CIS internal Code Page being used. Characters which do not have an equivalent representation in the CIS internal code page **cannot be translated**, and cause CIS to issue a message and stop its processing.

3. The translation process is restricted to single byte (8-bit) encoding schemes. Multi-byte encoding schemes such as Unicode Transformation Format-16 (UTF-16) or Unicode Transformation Format-32 (UTF-32) are not supported. For this reason, an INPUTCP() keyword with a parameter other than UNKNOWN, UTF16LE or UTF16BE is required when the name of a FIELD keyword of type other than LIT, VAR or ATT is specified in an INSERTBCOCA(), INSERTINDEX(), INSERTPTOCA(), INSERTPTOCANOP() or EXTRACTINDEX keyword. Further the keywords ANALYSTCP, DEFINEBCOCA, DEFINEPTOCA, MESSAGECP and EXTRACTCP require single byte (8 bit) encoding schemes.
4. Similarly, composite characters used to build single text elements (e.g. an Accent mark ´ combined with a Latin vowel **a** to build the character **á**) cannot be translated using the Code page support available in CIS and are treated as two separate characters. Pre-composed characters are supported (e.g. when the text editor used to create the CIS control file combines the composite characters into a single code point).
5. The above restriction only applies to the Code Page support used during data selection, insertion, extraction and when indexing. The double-byte S/370 Line format data conversion capability in CIS (SOSI parameter) is not affected as it does not require nor use the Code Page support functionality.

3.3 Support of Multiple-Up applications

The purpose of this section is to describe the support provided by CIS for Extended two-up (X2UP), Cut-Sheet Emulation (CSE) and N_UP applications. Many statement applications in production today were originally designed to generate groups of text records representing one statement per physical form. The introduction of printing systems supporting wide forms created a need to simplify the placement of multiple images on wider forms. X2UP, N_UP and CSE (Cut Sheet Emulation) are terms used to describe this ability of placing multiple images on a single form without re-coding the statement application. In both simplex or duplex mode, SPS supports X2UP[1], CSE[2] and all facets of N_UP.

X2UP, N_UP and CSE are mutually exclusive features and may not be used concurrently.

3.3.1 X2UP Printing

X2UP was developed by Océ Printing Systems to support their exclusive offering (at the time) of a continuous form device capable of imaging two DIN A4 images, side by side, on a printed form. With X2UP the AFP application pages are placed sequentially on the form. In simplex mode the first page is placed on the left and the second page is placed on the right side of the form. Fig. 7 shows how this is done. X2UP is not only available for Cut-Sheet or C-twin printers.

In duplex mode the first image is placed on the left front, the second image is placed on the back, so that when the form is split, it produces a page with print on both sides. The next two images produce two sided print on the right portion of the form.

The two portions of the physical media are treated as if they were two separate pieces of cut-sheet media (also called "sheetlets"). When a MO:DCA document is sent to a printer in X2UP mode, MO:DCA sheets and their content are mapped to each portion of the physical media at the printer. For this reason, a media collection for finishing operations such as UP3I may end at the left side and a new one may start at the right side.

X2UP may be specified using the X2UP keyword of the AFP Processing control file.. Additional X2UP parameters can be used to cause the first image to be placed on the right side of the form (RIGHT parameter) so that they are printed in a right-left sequence

[1] Not available in C-twin printers, not available in Cut-sheet printers

[2] Not available in A-twin printers, not available in Cut-sheet printers

(as opposed to left-right in normal X2UP), or that an identical copy of the left side of the form be produced on the right side of the form (ICOPIES parameter). Using X2UP does not require any changes to existing page definitions or overlays.

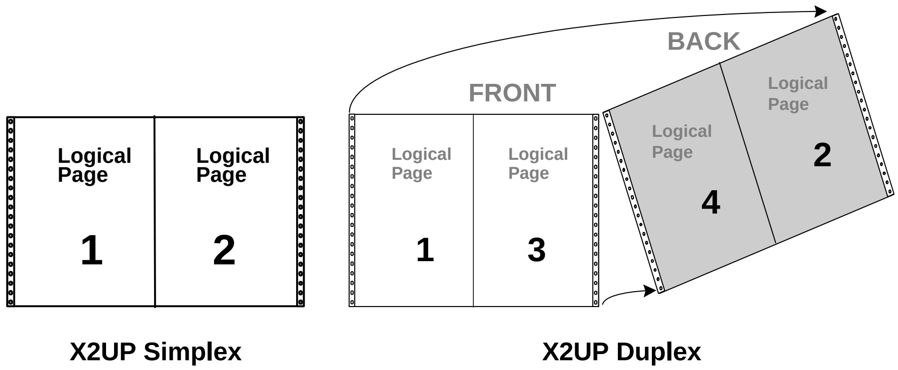


Fig. 7 X2UP Page Placement

3.3.2 N_UP Printing

N_UP has two levels of support, basic N_UP and Enhanced N_UP. Using a form definition, with N_UP one can place pages sequentially on a form, breaking the form up into logical partitions. N_UP printing is available for Continuous-Form and for Cut-Sheet printers.

The portions of the physical media (called "partitions" in case of N_UP) are **not** treated as if they were separate pieces of cut-sheet media. When a MO:DCA document is sent to a printer in N_UP mode, each MO:DCA sheet (including all the partitions it contains) is mapped to a single, wide, physical media at the printer. For this reason, a media collection for finishing operations such as UP3I **may not** end in one partition and a new one may start at the next partition on the same physical media (such a combination will cause the printer to eject to the next physical media, leaving all other partitions in that sheet unused).

- <u>**BASIC N_UP**</u>

Fig. 8 shows the default orientations for each logical page placement. Basic N_UP can also orient the logical pages for landscape orientation. N_UP 2 in the above illustration is equivalent to X2UP ON for simplex printing (as shown in Fig. 7). In duplex mode N_UP 2 does not modify image placement on the front of the form. In duplex mode, page three will be placed on the back of page two and page four will be placed on the back of page one (also shown in Fig. 7).

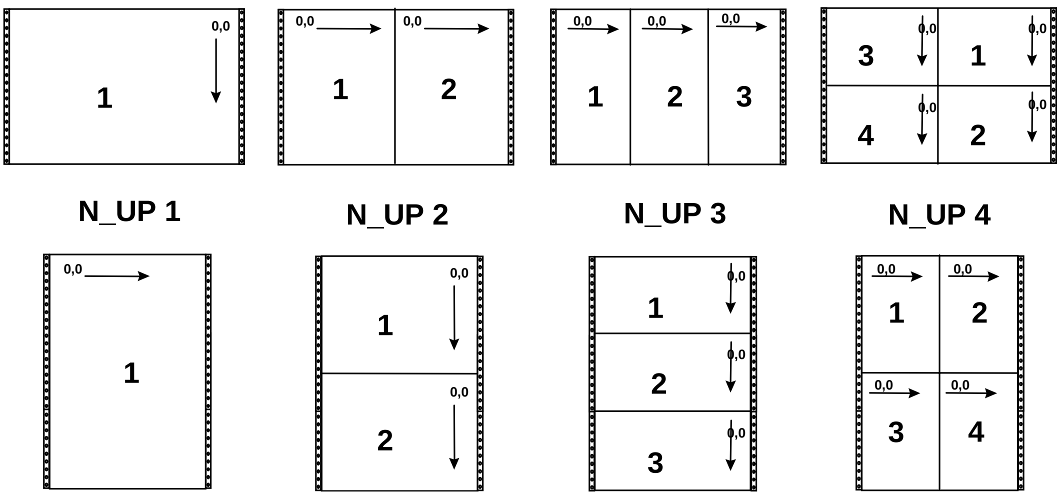


Fig. 8 Sample Portrait Across Partition Assignments for Basic N_UP

- **ENHANCED N_UP**

Enhanced N_UP uses the partition arrangement shown in Fig. 9. However, the sequence of each logical page placement can be modified for any possibility. The examples discussed thus far have involved the placement of two images on one form, as this is the most common usage of N_UP for existing high volume statement applications. Enhanced N_UP introduces some interesting coding deviations which need to be addressed.

The format shown in Fig. 9 can be accomplished using Enhanced N_UP because up to four partitions can be coded for each side of a form. In duplex mode, this means a maximum of eight logical pages can be placed per form. Logical page placement need not be symmetrical for each side of the form, (in this case seven on the front and one on the back). Partitions in the form definition can be coded so that no logical page placement takes place. This allows for an uneven numbers of pages per form, or an overlay placement in a particular partition without application data.

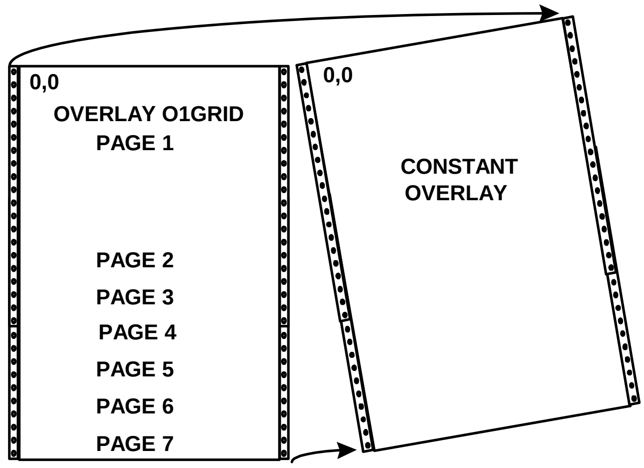


Fig. 9 Asymmetrical Page Placement Using Enhanced N_UP

3.3.3 Cut-Sheet Emulation (CSE)

Similar to X2UP, the "Cut-Sheet Emulation" feature can be used to print on continuous-forms media that, once cut and collated, emulates two sheets of cut-sheet output. In this mode, the printer logically divides the continuous-forms media in half parallel to the carrier strips and controls the placement of pages on either the left side or the right side of the physical media as defined by the operator in the printer control panel. The CSE feature requires manual operator intervention and may not be controlled using the SPS control files.

The two portions of the physical media are treated as if they were two separate pieces of cut-sheet media (called "sheetlets"). When a MO:DCA document is sent to a printer in X2UP mode, MO:DCA sheets and their content are mapped to each portion of the physical media at the printer. For this reason, a media collection for finishing operations such as UP3I may end at the left side and a new one may start at the right side.

3.4 Processing variables

This section describes the processing variables available in CIS. These variables are normally used in connection with the Data Enrichment parameters. Fig. 4 on page 32 shows the time at which the internal CIS variables are updated.

3.4.1 Processing notes

AFP data has a hierarchical organization that is processed in a sequential manner and that has a direct influence in the way the processing variables are updated. Before the variables are used, it is important to understand the way they are updated so that using undefined variables can be avoided. This is explained in the following lines:

1. An AFP print file may consist of one or more Documents (DOC), Page Groups and Pages. The formatting rules (FormDef, internal Medium Maps) that accompany the print file define which pages belong to a sheet (SH).
2. Pages identify the lowest entity in the variables hierarchy. Pages may not be nested nor may contain any of the other entities. At the time a page-begin is detected by CIS, the contents of all the PAG, SH, MP and DOC variables is known and up-to-date.
3. Sheets identify the next higher entity in the variables hierarchy. Sheets may not be nested and may only contain Pages. At the time a sheet-begin is processed by CIS, the contents of the SH, MP and DOC variables is known and up-to-date. The content of the PAG related variables is undefined.
4. CIS defines Mail Pieces (MP) as the top level Page Group in a print file. Mail Pieces identify the next higher entity in the variables hierarchy. Mail Pieces may not be nested and may only contain Sheets. At the time a mailpiece-begin is detected by CIS, the contents of the MP and DOC variables is known and up-to-date. The contents of the SH and PAG related variables is undefined.
5. Documents identify the highest entity in the variables hierarchy. Documents may not be nested and may only contain Mail Pieces. At the time the document-begin is process by CIS, the contents of the DOC variables is known and up-to-date. The contents of the MP, SH and PAG related variables is undefined.

In summary, when using any of the CIS Processing variables, it is important to know that the contents of all variables BELOW the entity level at which they are being used are undefined.

3.4.2 List of CIS processing variables

The following variable names are available and can be used in the FIELD and INITVARIABLES keywords of the control file. Their initial value and change conditions are described under *6.2 Appendix B: CIS processing variables* on page 331.

1. Input related variables
   - _inDocumentNumber
   - _inMailPieceNumber
   - _inSheetNumber
   - _inPageNumber
   - _inSheetNumberInMailPiece
   - _inPageNumberInMailPiece

2. Output related variables
   - _outSegmentNumber
   - _outDocumentNumber
   - _outMailPieceNumber
   - _outSheetNumber
   - _outPageNumber
   - _outMailPieceNumberInDocument
   - _outSheetNumberInMailPiece
   - _outPageNumberInMailPiece
   - _outPageNumberInSheet
   - _outNumberOfMailPiecesInDocument
   - _outNumberOfSheetsInMailPiece
   - _outNumberOfPagesInMailPiece
   - _outNumberOfPagesInSheet
   - _outPageSfOffset
   - _outPageByteOffset
   - _outSheetSfOffset
   - _outSheetByteOffset
   - _outMailPieceSfOffset
   - _outMailPieceByteOffset

3. Other processing variables
   - _dateYYYY
   - _dateYY
   - _dateMM
   - _dateDD

3.5 How to select and segment data in a print file

3.5.1 Segmenting Output data

There are situations where installations need to split large print applications into several smaller files. For example: to be able to print sections of a large file in parallel, to fulfill special shipping requirements for large fan-fold output (e.g. weight, pile size), to balance the load of post-processing equipment, to prepare the documents for electronic distribution, etc.

For all these cases, CIS offers a simple and powerful output segmentation facility. Using the **OUTPUTSEG** keyword, an installation may request CIS to split the input print file into one or more segments. The segmentation criteria are specified using three parameters: ENTITY, COUNT and BOUNDARY.

```
OUTPUTSEG( ENTITY, COUNT, BOUNDARY )
```

Fig. 10 Syntax OUTPUTSEG

The **OUTPUTSEGPSDEF** keyword may be used under MVS to specify the space and volume serial to be used for sequential data sets. Alternatively the user can provide a set of pre-allocated Data sets CIS uses for writing its output.

The ENTITY parameter may indicate Document, Mail Piece, Sheet, Page or Byte. It specifies the items to be counted for being written in separate output files. Byte segmentation causes CIS to generate segment files with the requested size so that a page may span over two or more segments. The optional BOUNDARY parameter may be used to specify an entity which CIS shall write completely to the actual output file before beginning the next one. The COUNT parameter indicates the number of entities (or megabytes in case of BYTE) each segment will contain. Only the last file may contain a different number of entities.

Fig. 4 on page 32 shows the CIS internal workflow. The Output Segmentation process can be combined with other features such as data selection and sorting. It cannot be used in connection with the generation of an Index Object File (INDEXOBJ must be NONE). Note that in case of resource packaging (parameter RESTYPE) the resource output is not valid before the CIS run is complete.

Example:

A z/OS installation wants to segment a very large output file in packages (files) of 200.000 sheets each with the last mailpiece not being separated over 2 files for parallel printing on several available printers. The user specifies the dataset definitions OUT001, OUT002, OUT003 and OUT004 in the JCL. These files shall be used by CIS for writing its output. In case of the job having more than 800.000 sheets the last file will contain the rest. In case of a smaller input file CIS will empty (OPEN – CLOSE) all files not needed for output. The JCL snippets and keywords shown in Fig. 11 can be used for this purpose. Note that the output is directed to SPOOL in this example and the usage of the JCL FREE and SPIN keywords for making the created SPOOL entries available for printing after CIS closed them.

```
//OUT001 DD  SYSOUT=8,DCB=(RECFM=VA,BLKSIZE=32760,LRECL=32756),
//    DEST=PRT1,FREE=CLOSE,SPIN=UNALLOC
//OUT002 DD  SYSOUT=8,DCB=(RECFM=VA,BLKSIZE=32760,LRECL=32756),
//    DEST=PRT2,FREE=CLOSE,SPIN=UNALLOC
//OUT003 DD  SYSOUT=8,DCB=(RECFM=VA,BLKSIZE=32760,LRECL=32756),
//    DEST=PRT3,FREE=CLOSE,SPIN=UNALLOC
//OUT004 DD  SYSOUT=8,DCB=(RECFM=VA,BLKSIZE=32760,LRECL=32756),
//    DEST=PRT4,FREE=CLOSE,SPIN=UNALLOC

...

OUTPUTSEG( SH, 200000, MP )
OUTPUTDD( OUT*** )
```

Fig. 11 Example: Output Segmentation

Example:

An installation wants to create a separate file for each Mail Piece in the print output in order to distribute them electronically. Here OUTFILE refers to a Partitioned Ordered Data set. Thus CIS will write each Mail Piece in its own member. A list with all the mail pieces found in the file should also be created. Fig. 12 shows the control file required in this case.

```
OUTPUTSEG( MP, 1 )
OUTPUTDD( OUTFILE )
```

Fig. 12 Example: Output Segmentation

Processing Notes:

1. Segmenting a job at an entity level lower than SHEET (e.g. PAGE) may cause the last SHEET to be split into different files. This will normally affect the way the segments are printed, especially in case of N-up and/or duplex applications.
2. Segmenting a job at a BYTE level will generally cause CIS to leave an incomplete AFP page at the end of each segment. Select PAGE or higher level in case you want each segment to contain complete AFP document structures that can be processed by an AFP tool or pass the BOUNDARY parameter to specify the entity level you want CIS to separate the output. When reaching the specified items CIS will complete the specified boundary entity writing it to the actual output file before beginning the next one.
3. Under z/OS CIS can be parameterized to write its output either to pre-allocated Data sets or to allocate the needed Data sets itself.

   - Passing a DD in OUTPUTDD referencing a Partitioned Data set (DSORG=PO) CIS will generate a different PDS member for each output segment and will name them D0000001, D0000002, etc.
   - Passing a DD referencing a Sequential Data set (DSORG=PS) CIS will allocate a different Sequential Data set for each output segment with the same DCB parameters and will name them <DSname>.D0000001, <DSname>.D0000002, etc.

     <DSname.D0000000> is used for writing the protocol. It will be allocated with:

     ```
     LRECL  = 80
     BLKSIZE = 6160,
     RECFM  = FB
     ```

     and will contain all Data set names used for this job. Example content of such a protocol file:

     ```
     HLQ.OUTPUT.D0000001
     HLQ.OUTPUT.D0000002
     HLQ.OUTPUT.D0000003
     ```

     The OUTPUTSEGPSDEF keyword can be used to specify the size and volume serial to be used for allocating these Data sets.
   - SYSOUT Data set. CIS will create a different SYSOUT data set for each output segment. The parameters specified in the output DD (see OUTPUTDD keyword) are used for all segments.
   - Passing a DD mask in the OUTPUTDD parameter referencing Data set definitions in the JCL step CIS will write its output to these Data sets. Following rules have to be kept in mind:

     The DD mask specified in OUTPUTDD must begin with at least one character followed by asterisks being overall not longer than 8 characters. CIS replaces the asterisks by numbers (beginning with 1) and will write its output to these Data sets. All Data sets from such a list must be of the same type and format. CIS uses such a set of Data sets until its sequence is interrupted or ends. If a Data set is full before the specified number of entities is written to it CIS will terminate its processing issuing an error message.

Examples:

1. Following parameter

```
OUTPUTDD( OUT** )
```

might refer to these Data set definitions in the step´s JCL:

```
//OUT01 DD DSN=HLQ.OUTPUT.FILE1
//OUT02 DD DSN=HLQ.OUTPUT.FILE2
//OUT03 DD DSN=HLQ.OUTPUT.FILE3
//OUT04 DD DSN=HLQ.OUTPUT.FILE4
//OUT05 DD DSN=HLQ.OUTPUT.REST
//OUT07 DD DSN=HLQ.OUTPUT.REST2
//OUT08 DD DSN=HLQ.OUTPUT.REST3
```

CIS will use OUT01 to OUT05 for its output. OUT07 and OUT08 are not used because they are not in continues sequence.

2. Following parameter

```
OUTPUTDD( P* )
```

might refer to these Data set definitions in the step's JCL:

```
//P1 DD SYSOUT=X,DCB=(RECFM=VA,BLKSIZE=32760,LRECL=32756),
//      DEST=PRT1,FREE=CLOSE,SPIN=UNALOC
//P2 DD SYSOUT=X,DCB=(RECFM=VA,BLKSIZE=32760,LRECL=32756),
//      DEST=PRT2,FREE=CLOSE,SPIN=UNALOC
//P3 DD SYSOUT=X,DCB=(RECFM=VA,BLKSIZE=32760,LRECL=32756),
//      DEST=PRT3,FREE=CLOSE,SPIN=UNALOC
//P4 DD SYSOUT=X,DCB=(RECFM=VA,BLKSIZE=32760,LRECL=32756),
//      DEST=PRT4,FREE=CLOSE,SPIN=UNALOC
```

CIS will use PRINT001 to PRINT004 for its output.

4. Under Linux the segment files are created using the following naming convention:

   <*fileName*>.D0000001, <*fileName*>.D0000002, etc.

   where *fileName* corresponds to the parameter specified in the OUTPUTDD keyword (Note: the naming conventions have changed since the last version).

3.5.2 Selecting Input data

Even though in most cases installations want to process complete print files, there are many situations where a partial or selective processing is desirable. For example:

- During job scheduling, when the mail pieces to be processed have to be selected (or excluded) based on their size because a post-processing device such as an Inserter cannot handle more than a given number of sheets per mail piece.
- During job scheduling, when a print file is to be distributed electronically based on its index entries.
- During re-prints, when some specific documents, mail pieces or sheets need to be reprinted.
- During testing or audit controls, when randomly extracted pages are used for this purpose.

For all these applications, CIS offers three flexible input data selection and exclusion options: Entity-range, Index-range and Entity-size based selection. A fourth method can be implemented in the form of a user exit program (see on page for more information).

Before any data selection process starts (Fig. 4 on page 32) CIS updates all its processing variables associated with the input data. As a result, the input related variables always reflect the original print stream characteristics regardless of whether data selection and/or exclusion is done. Output related variables never include pages which were excluded during the selection process. This means that, for a given input print file, different CIS runs with different selection and/or exclusion parameters will always generate the same input related variable values and will always generate different output related variable values. Please refer to *3.4 Processing variables* on page 46 for more information.

- **ENTITY-RANGE BASED SELECTION**

Entity-range selection works based on the absolute sequence number that each entity to be processed has within the print file. As shown in Fig. 4 on page 32, this is the first type of data selection done in CIS, even before completing the data normalization process. The data which is not selected may or may not be parsed and validated by CIS.

Entity-range selection is controlled using the SELRANGE keyword. With this parameter an installation specifies an entity and one or more ranges to be included or excluded, with the exclude parameter having a higher weight as an include. The example in Fig. 13 will select mail pieces 501 to 1499 and 1601 to 2000.

```
SELRANGE ( MP,1001-2000, !1500-1600, 501-1550 )
```

Fig. 13 Entity-range selection parameters

CIS makes sure that the AFP structure on the resulting output file matches the input structure. This means that every page or group of pages selected will be preceded by the corresponding Begin Document (BDT) and Begin Named Page Group (BNG) structured fields and will be followed by the equivalent END structures that were found in the input file.

The resources associated to pages excluded during the Entity-range selection process are also excluded from the resource packaging and consolidation process.

- **INDEX-RANGE BASED SELECTION**

Another type of input data selection done in CIS is the Index-range based selection, where the index values are used to determine if an entity is to be selected or excluded from processing.

As shown in Fig. 4 on page 32, this step is done after normalizing and converting the input data. All pages in the input file are parsed and validated. Only the items within the index range are selected for further processing. All others are discarded and do not get passed to the output files generated by CIS.

Index-Range selection is controlled using the SELINDEXRANGE keyword. With this keyword an installation specifies an entity, an index attribute name and one or more index ranges to be included or excluded, with the exclude parameters having a higher weight as the includes. The selection is based on a simple comparison of the hexadecimal values of the index field used as criteria. The example in Fig. 14 will select all mail pieces that contain an index attribute name called 'CUSTOMER NAME' where the index value starts with the character 'S', except those where the name starts with 'SMITH'.

```
SELINDEXRANGE ( MP, 'CUSTOMER NAME', 'S'   - 'S',
                   !'SMITH' - 'SMITH' )
```

Fig. 14 Index-Range selection parameters

Index values are binary compared, without making special considerations for specific data encoding schemes (ASCII, EBCDIC), national characters or special symbols. Data alignment within the field (left or right justified), leading and/or trailing characters (e.g. leading zeroes, trailing blanks) and decimal separation characters (e.g. comma or dot) are handled as part of the data and may influence the selection results. The resources associated to pages excluded during the Index-range selection process of CIS are also excluded from the resource packaging process.

Items with limiting index values are included (in case of include) resp. excluded (in case of exclude).

It is not possible to use the Index Tag Elements inserted with the insertINDEX keyword to control the selection process of the same CIS run. Splitting the index insertion and selection processes in two CIS runs will bypass this problem. Fig. 4 on page 32 shows the sequence used for the index insertion, sorting and extraction steps.

- <u>**ENTITY-SIZE BASED SELECTION**</u>

Entity-size selection allows an installation to select the items to be processed based on their physical size. It is the last selection step done in CIS as shown in Fig. 4 on page 32 and requires the use of a work file as intermediate storage. All pages in the input file are parsed and validated by CIS, and are written into the internal work file.

Entity-size selection is controlled using the SELSIZE keyword. With this parameter an installation specifies an entity and one or two size criteria's. Only the items matching the size criteria are selected for further processing. All others are discarded and do not get passed to the output files generated by CIS.

The resources associated to pages excluded during the Entity-Size selection process of CIS **are included** in the resource packaging and consolidation process. A second CIS run over the resulting document file may be used to generate a resource file which only includes the resources required by the selected pages.

<u>Example:</u>

The example in Fig. 15 will select all mail pieces containing more than 5 and not more than 10 (6 to 10) sheets.

```
SELSIZE ( MP, GT, 5, SH, LE, 10 )
```

Fig. 15 Size based selection parameters

3.6 How to convert, normalize and index data

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
- Record-format Line data including Mixed-mode and double-byte encoding.
- Unicode (UCS-2) Line data including Mixed-mode, Carriage control and TRC.
- XML data (No Mixed-mode, no TRC and no CC supported).
- SPDS data that conforms to the specifications described in [20] (see *8 Bibliography* on page 465).
- Unformatted ASCII data as described in this manual. ANSI carriage controls using ASCII coding are also supported.

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

Fig. 4 on page 15 shows the internal CIS workflow. The conversion process is required for data which is not in MO:DCA format. In this case, CIS converts the data using the specifications contained in the Page definition[1]. During the conversion or normalization process respectively, CIS may be requested to scan the input data and generate index tags which are then used during Index-range data selection or during contents sorting and which are written in the Index Object file. The LINEMERGE keyword may be used to control the way in which two or more input lines are merged into a single output line, so that the results obtained are compatible with the IBM 3800 and E-mode specifications.

CIS normalizes data which is already in AFP format by removing obsolete AFP definitions and some printer resolution dependencies. The normalization process also verifies the adherence to the MO:DCA-P rules, detecting and reporting any inconsistencies between the print data stream and its resources. Any Index tags already contained in the AFP data may be used during Index-range data selection, during contents sorting and are written in the Index Object file.

[1] The conversion process is mandatory for S/370 Line data, Unicode line data, XML data and Record-format Line data as these formats are not supported as output data streams.

3.6.1 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, usually in EBCDIC coding (could also be ASCII or a 16-bit based coding), that normally begin with a carriage control character (CC) containing vertical line spacing commands (refer to *Fig. 18 Legacy application* on page 60). An optional table reference character (TRC) may be used to control font selection. More information about S/370 Line formatted data may be found in [17](see *8 Bibliography* on page 465).

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 (e.g. PRISMAproduction, PSF, etc.) so that the application's appearance is always the same. 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 CIS is shown in *Fig. 20 Legacy application: converted document* on page 64.

```
  FORMDEF ( F1CIS02 )
  PAGEDEF ( P1CIS02 )

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

/*  Define the first level page groups - mail pieces -            */
/*  When a new page occurs and the customer page number is 000001 */

  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     )

/* 4 Fields at input data, in relation to TRIG1 */
    FIELD ( accountF,  7, 3,  8 )
    FIELD ( nameF   , 13, 3, 30 )
    FIELD ( addressF, 14, 3, 30 )
    FIELD ( plzF    , 15, 3, 30 )
```

Fig. 16 Legacy application: Indexing parameters

During the conversion process (but also through the normalization process for MO:DCA-P data) CIS may be requested to scan the input data and to recognize the document’s structure using the TRIGGER keyword defined for this purpose. This feature allows the user to specify one or more “trigger” strings to be used to determine the beginning of each new structure within the input print file. By specifying multiple triggers, an installation may use CIS to normalize and add index tags to line data files containing multiple page layouts, or may be even use CIS to consolidate multiple S/370 Line format print reports into a single one.

Example:

Using the application shown in *Fig. 18 Legacy application* on page 60, the following CIS keyword would request the generation of an Index Object file containing the ACCOUNT, NAME, ADDRESS and PLZ (Zip code) fields as indexing items:

```
EXTRACTINDEX( IndexObjectFile.txt, DOC,
              'ACCOUNT=%s, NAME=%s, ADDRESS=%s, PLZ=%s \r\n',
               accountI, nameI, addressI, plzI )
```

The specifications of a FIELD keyword must start within the record data (size) available. The length however may extend beyond the end of the record. In this case CIS will append blank characters (using the specifications from the INPUTCP keyword) to complete the requested field length.

The Index Object file resulting from the indexing keywords in Fig. 16 is described later on page 77. The INPUTFORMAT keyword specifies the criteria to be used for recognizing input records (lines). The OUTPUTFORMAT keyword controls the format used in the MO:DCA-P files generated by CIS.

CIS allows another more flexible way of defining the FIELD parameter which can be used when FIELDs are in no fixed relative position to the trigger. This usage may be more likely in the handling of MO:DCA-P input data (therefore see also the example in chapter *3.6.6*), but is explained here also with a quite far-fetched example.

Example:

Using the application shown in *Fig. 18 Legacy application* on page 60, let’s say the installation is interested in having the customers´ new account balance (“Neuer Kontostand”) in addition to the index value (“BALANCE”). As this FIELD is in no fixed relative position to the trigger you can use the following FIELD parameter setting:

```
INDEX ( balanceI, account, 'BALANCE', balanceF )

FIELD ( balanceF, ‘Neuer Kontostand EURO’, 35, 12 )
```

Fig. 17 Legacy application: Enhanced indexing parameter using a search field

3.6.2 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 were 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. Records containing AFP structured fields must include a CC byte with the value 0x5A. Line data records have a CC other than 0x5A. These records may also include a TRC byte. Please refer to [17] (see *8 Bibliography* on page 465) for more information about AFP Mixed data.

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 (e.g. PRISMAproduction, 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.

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 keywords and features are available. Any existing Document and Page-Group structured fields (BDT, EDT, BNG, ENG) will be removed and replaced with the structured fields resulting from the indexing process (TRIGGER keyword specified). In this case CIS also removes and/or replaces any Page Group level TLE's that may exist in the input file. Page level TLE's are left unchanged.

```
                                                                                                    1
         1         2         3         4         5         6         7         8         9         0
1...|....0....|....0....|....0....|....0....|....0....|....0....|....0....|....0....|....0....|....0....|

10Bank of Yesterday, Inc.
 0Postfach 12345, 20000 Berlin
 0
 0
 0Kontonummer/        Bankleitzahl       Laufende            Blatt-         DATUM               Betreff
 0Kontoinhaber                           Nummer              Nummer
 0
 014034350            50020010           001-002             00005          29.03.2000
 0Lieberwirt, William
 0
 0718/00079058
 0
 0Mr.
 0William Lieberwirt
 0Ruehmann Str. 78
 099091 Erfurt
 0
 0
 0Tag der  Wert-    Text/Verwendungszeck                        Zu Ihren Lasten     Zu Ihren Gusten
 0Buchung  stellung
 0
 0   5.03    04.03  DPT    Restaurant MY DAY                          780.00 DM
 0                         Invoice Nr. 2345-98                        398.81 EUR
 0                         Account nr. 567-34111014
 0
 0   7.03    05.03  TM     Silverwing Lottery                                             248.90 DM
 0                         Deposit Nr. 23456                                              127.26 EUR
 0                         Kontonummer 2345 12
 0
 0   9.03    08.03  TM     MOVEFAST International transport                               241.50 DM
 0                         Return                                                         123.48 EUR
 0                         St. Nr 12345 987 67
 0
 0  11.03    09.03  DPT    Silverwing Lottery                                             301.80 DM
 0                         Money transfer                                                 154.31 EUR
 0                         Account 23457865
 0
 0  13.03    11.03  ***    Mrs. Anna Tormann                          539.00 DM
 0                         Cheque Nr. 567 - 3451                      275.59 EUR
 0                         Kontonummer 456781
 0
 0  15.03    13.03  DPT    WELCOME Credit card                         99.80 DM
 0                         Cheque Nr. 567 - 3451                       51.03 EUR
 0                         Account nr. 567-34111014
 0
 0  17.03    15.03  CSH    Herr Thomas Hopf                                               268.30 DM
 0                         Deposit Nr. 23456                                              137.18 EUR
 0                         Kontonummer 456781
 0
 0  19.03    18.03  XFR    TAIKI bar, Tokyo                           983.40 DM
 0                         Invoice Nr. 32                             502.80 EUR
 0                         Card nr. 4005 3245 1344 9993
 0
 0
 1                         Neuer Kontostand DEM                       688.47 DM
 1                         Neuer Kontostand EURO                      352.01 EUR
```

Fig. 18 Legacy application

In case of AFP Mixed data, CIS propagates all No-Operation (NOP, 0xD3EEEE) structured fields which are part of the page data and writes them immediately after the MO:DCA-P Active Environment Group. The first record in the input file is not considered part of the page data and is therefore not propagated.

The INPUTFORMAT keyword specifies the criteria to be used for recognizing input records (lines). The OUTPUTFORMAT keyword controls the format used in the MO:DCA-P files generated by CIS.

For indexing MO:DCA-P data see a special example in chapter Normalizing and indexing print files containing AFP and MO:DCA-P data on page 65.

3.6.3 Converting and Indexing print files containing Record-format Line data

Record-format Line data consists of records of text data, where each record contains a 10-byte identifier which selects the Record Descriptor (RCD) in the Record-format Data Map in the PageDef used to format the line data. A carriage control (CC) byte is optional and, if present, is ignored. A table reference character (TRC) is not supported.

The Record Descriptor structured field contains information such as record position, text orientation, font selection, field selection, and conditional processing. The RCD must be part of the Data Map Transmission Subcase structured of the currently active data map.

Some of the functions that can be accomplished with Record-format line data include:

- Selecting different formatting for different type of Data records based on the Record ID.
- Defining page headers and trailers and column headers to be automatically printed on subsequent pages.
- Automatic page numbering.
- Automatic page eject when text reaches the bottom margin.
- Format database records created with field delimiters rather than fixed length fields.
- Aligning field output to the left or right side.

The Record Descriptor structured field contains information such as record position, text orientation, font selection, field selection, and conditional processing. For more information concerning Record-format Line data please refer to [17] (see *8 Bibliography* on page 465).

3.6.4 Converting and Indexing print files containing Unicode (UCS-2) Line data

This section describes the CIS support of UCS-2 Line data as specified in [17] (see *8 Bibliography* on page 465). Input AFP data (e.g. PTX structured field) in Unicode encoding is handled as any other double-byte coded data (see “Converting and Indexing print files containing AFP Mixed data”on page 66).

The Unicode Standard is a character encoding scheme for written characters and text. It defines a consistent way of encoding multilingual text that enables the exchange of text data internationally. The Unicode standard is a superset of all characters in widespread use today. It contains the characters from major international and national standards as well as prominent industry character sets. The Unicode standard is defined by the Unicode Consortium. For more information about the standard and/or the Unicode group please visit http://www.unicode.org.

Unicode (UCS-2) Line data

```
         CC ( NO)
    TRCTYPE ( NO )
    INPUTCP ( UTF16LE ) or INPUTCP ( UTF16BE )
INPUTFORMAT ( RDW ) or INPUTFORMAT ( FIXED,xx )
```

CC	Unicode (UCS-2) Line data

```
         CC ( IBM ) or CC ( ASA )
    TRCTYPE ( NO )
    INPUTCP ( UTF16LE ) or INPUTCP ( UTF16BE )
INPUTFORMAT ( RDW ) or INPUTFORMAT ( FIXED,xx )
```

CC	TRC	Unicode (UCS-2) Line data	LF

```
         CC ( IBM ) or CC ( ASA )
    TRCTYPE ( IBM )
    INPUTCP ( UTF16LE ) or INPUTCP ( UTF16BE )
INPUTFORMAT ( AFPSTREAM, 0x0A00 or 0x000A )
```

Fig. 19 Unicode (UCS-2) Line format data

Unicode Line data is a special kind of line data where the text portion of each line is encoded using the ISO Unicode standard (UCS-2). The level of support for Unicode Line data provided in CIS is described in [19] (see *8 Bibliography* on page 465) and includes:

<u>Input data stream</u>

- The UTF16BE (Unicode Transformation Format 16 bit, Big Endian) parameter of the INPUTCP keyword. It is used for Line data files encoded using the UTF16 (Big Endian) format. Fixed two-byte code points with surrogates are supported.
- The UTF16LE (Unicode Transformation Format 16 bit, Little Endian) parameter of the INPUTCP keyword. It is used for Line data files encoded using the UTF16 (Little Endian) format. Fixed two-byte code points with surrogates are supported.
- The INPUTFORMAT keyword to indicate whether the Unicode Line data is stored in record (S/390 file system) or in stream format. In case of stream format, the end-of-line delimiter (AFPSTREAM parameter) should be set to 0x000A.
- BOM's (Byte Order Marks), a 2 to 3 bytes string at the beginning of a Unicode file indicating the type of encoding being used (UTF16LE, UTF16BE, ...) are not allowed in connection with Unicode Line data.
- UTF32 is not supported.

<u>CIS processing</u>

- Unicode encoding is not supported for the CIS control file. This restriction is important in the following cases:

- TRIGGER keyword	The hexadecimal notation of the *value* parameter must be used. The value entered must match the endian used in the input file. The *column* parameter counts bytes and not characters.
- *attributeName* parameter	The hexadecimal notation of this parameter must be used in the following CIS keywords when the attribute name is to be in Unicode: INDEX, FIELD, INSERTINDEX, MPCONSOLIDATION, REMOVEINDEX, SELINDEXRANGE, SORT. The value entered must match the Endian used in the input file.
- COMPRESS keyword	The hexadecimal notation of the *value* parameter must be used. The value entered must match the Endian used in the input file.

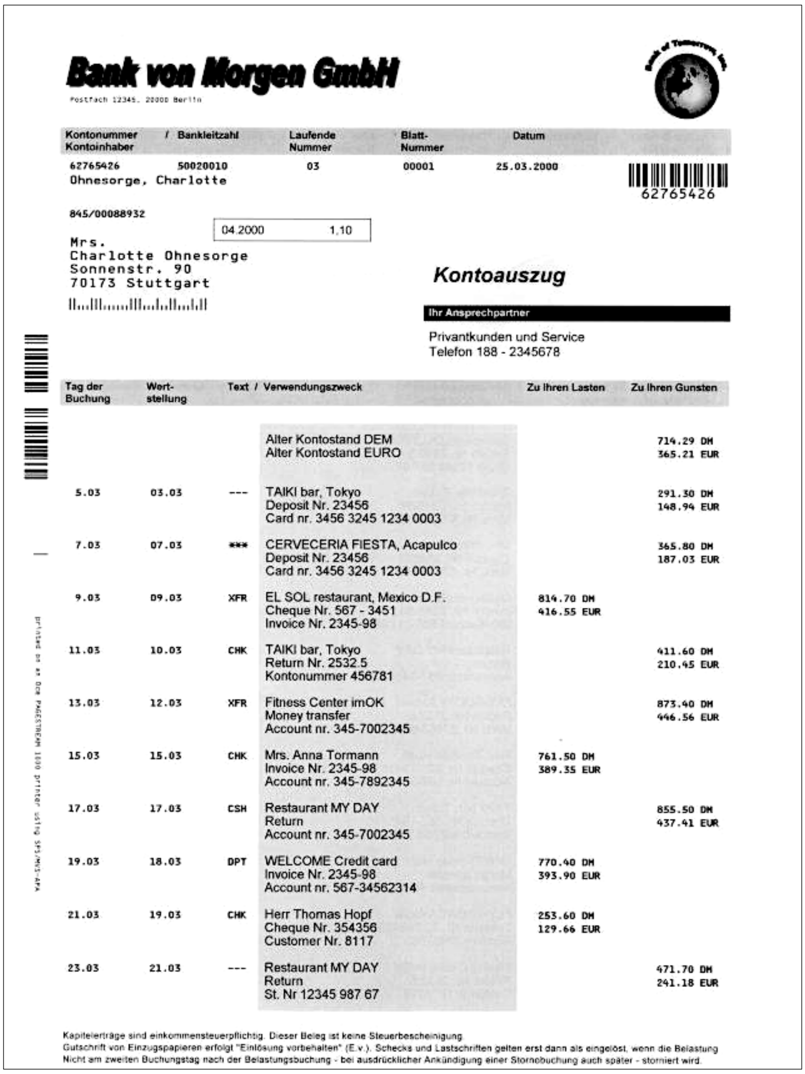
- The *fieldName* parameter (EXTRACTINDEX, INSERTBCOCA, INSERTINDEX, INSERTPTOCA or INSERTPTOCANOP keywords) may not be used to refer to a Unicode encoded string in case of a data enrichment or data extraction operation.
- The SOSI1, SOSI2 and SOSI3 parameters of the PRINTMODE keyword are not supported for Unicode Line data.

- The UTF16BE and UTF16LE code pages may be specified in the INPUTCP and INDEXCP keywords only. They are not allowed in ANALYSTCP, DEFINEBCOCA, DEFINEPTOCA, EXTRACTCP, MESSAGECP or in the CP command line keyword.
- Little Endian Unicode Line data is automatically converted to Big Endian during the CIS normalization process. A Conditional Processing text string in the PAGEDEF (if any) must be Big Endian encoded.

Output data stream

- Outline fonts (and not Raster fonts) are supported in AFP in connection with Unicode Line data. The fonts must be Big Endian encoded in order to print the MO:DCA data generated by CIS.

Fig. 19 shows some of the record formats for Unicode Line data and the CIS keywords to process them. Please refer to [19] (see *8 Bibliography* on page 465) for more detailed information about the valid Unicode Line data formats.



Bank von Morgen GmbH

Postfach 12345, 20000 Berlin

Kontonummer / Kontoinhaber	Bankleitzahl	Laufende Nummer	Blatt-Nummer	Datum
62765426	50020010	03	00001	25.03.2000
Ohnesorge, Charlotte				

62765426

845/00088932

04.2000 1,10

Mrs.
Charlotte Ohnesorge
Sonnenstr. 90
70173 Stuttgart

Kontoauszug

Ihr Ansprechpartner

Privantkunden und Service
Telefon 188 - 2345678

Tag der Buchung	Wert-stellung	Text	Verwendungszweck	Zu Ihren Lasten	Zu Ihren Gunsten
			Alter Kontostand DEM Alter Kontostand EURO		714.29 DM 365.21 EUR
5.03	03.03	---	TAIKI bar, Tokyo Deposit Nr. 23456 Card nr. 3456 3245 1234 0003		291.30 DM 148.94 EUR
7.03	07.03	***	CERVECERIA FIESTA, Acapulco Deposit Nr. 23456 Card nr. 3456 3245 1234 0003		365.80 DM 187.03 EUR
9.03	09.03	XFR	EL SOL restaurant, Mexico D.F. Cheque Nr. 567 - 3451 Invoice Nr. 2345-98	814.70 DM 416.55 EUR	
11.03	10.03	CHK	TAIKI bar, Tokyo Return Nr. 2532.5 Kontonummer 456781		411.60 DM 210.45 EUR
13.03	12.03	XFR	Fitness Center imOK Money transfer Account nr. 345-7002345		873.40 DM 446.56 EUR
15.03	15.03	CHK	Mrs. Anna Tormann Invoice Nr. 2345-98 Account nr. 345-7892345	761.50 DM 389.35 EUR	
17.03	17.03	CSH	Restaurant MY DAY Return Account nr. 345-7002345		855.50 DM 437.41 EUR
19.03	18.03	DPT	WELCOME Credit card Invoice Nr. 2345-98 Account nr. 567-34562314	770.40 DM 393.90 EUR	
21.03	19.03	CHK	Herr Thomas Hopf Cheque Nr. 354356 Customer Nr. 8117	253.60 DM 129.66 EUR	
23.03	21.03	---	Restaurant MY DAY Return St. Nr 12345 987 67		471.70 DM 241.18 EUR

Kapitelerträge sind einkommensteuerpflichtig. Dieser Beleg ist keine Steuerbescheinigung
Gutschrift von Einzugspapieren erfolgt "Einlösung vorbehalten" (E.v.). Schecks und Lastschriften gelten erst dann als eingelöst, wenn die Belastung
Nicht am zweiten Buchungstag nach der Belastungsbuchung - bei ausdrücklicher Ankündigung einer Stornobuchung auch später - storniert wird.

Fig. 20 Legacy application: converted document

3.6.5 Converting and Indexing print files containing Unformatted ASCII data

The code page support available in CIS makes it very easy for installations to process files coded in ASCII which use ANSI Carriage Control specifications (ASAA parameter of the CC keyword). These two features, combined with ASCII code fonts allow an installation to process print files that originated in a workstation (Unix, Windows, etc.) on an EBCDIC based platform such as MVS. The support for ASCII coded data is the same that is provided for S/370 Line format data coded in EBCDIC.

The INPUTFORMAT keyword specifies the criteria to be used for recognizing input records (lines). The OUTPUTFORMAT keyword controls the format used in the MO:DCA-P files generated by CIS.

3.6.6 Converting and indexing print files containing XML data

This section describes the CIS support of XML data as specified in [17] (see *8 Bibliography* on page 465).
Extensible Markup Language, or XML for short, is a new technology for designing text formats that let you structure your data. XML is a set of rules (you may also think of them as guidelines or conventions) defined and described by the World Wide Web Consortium (W3C) at http://www.w3.org. XML avoids common pitfalls in language design: it is extensible, platform-independent, and it supports internationalization and localization. The official XML specification forbids applications from trying to second-guess the creator of a broken XML file; if the file is broken, an application has to stop right there and report an error. This makes it a robust technology where sensitive data is handled.
XML data may be formatted using a Page Definition (with the XMD structured field), however there are following limitations:

- Carriage Control (CC) and Table Reference Characters (TRC) are not supported.
- The data is encoded using one of the following:
  - EBCDIC (Single-byte only)
  - ASCII (Single-byte only)
  - UTF-8
  - UTF-16
- MO:DCA data cannot be mixed with XML data.

3.6.7 Normalizing and indexing 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 BCOCA organized into resources such as fonts, overlays, page segments, form definitions, object containers and others. The data normalization process CIS supports for AFP data includes the following features:

- Conversion of IM images to IOCA to remove resolution dependencies.
- Automatic conversion of old fashion coded font resource invocation to MCF-2 format.
- Conversion of l-unit-per-unit-base values.
- Renaming of Page Group and Page name structures.
- Automatic insertion of Invoke Medium Map structured fields before every page group.
- Removal / Insertion of x'5A' carriage control.
- File reblocking.

The INPUTFORMAT keyword specifies the criteria to be used for recognizing input records (lines). The OUTPUTFORMAT keyword controls the format used in the MO:DCA-P files generated by CIS.

The normalization process in 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 (e.g. distributed in electronic form 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 or a mismatch between the fonts and data could still stop an application from being printed.

CIS is able to scan and insert index tags in AFP documents in a way similar to the indexing done for Line format data. The same keywords and features are available. Any existing Document and Page-Group structured fields (BDT, EDT, BNG, ENG, TLE) are removed and replaced with the structured fields resulting from the indexing process (TRIGGER keyword specified). In this case CIS also removes and/or replaces any Page Group level TLE's that may exist in the input file. Page level TLE's are left unchanged.

Parsing a MO:DCA-P data file for index information is easy using the file FIELD´s search parameter.

<u>Example</u>:

An installation wants to retrieve index information (the policy number, POLNUM) in its legacy MO:DCA-P data containing customer insurance polices. As this FIELD is in no fixed relative position to a possible trigger point you can use following parameter settings:

```
TRIGGER ( Trigger1, *, 4, x'D3EE9B', /* PTX SF */
          0, *, 'Policy Number: ' )

 FIELD ( PolNumF, 'Policy Number: ', 15, 12 )

 INDEX ( PolNumI, Trigger1, 'POLNUM', PolNumF )
```

Fig. 21 Legacy application: Enhanced indexing parameter using a search field

Where:

- `'Policy Number: '` is used as search string
- the 3rd FIELD parameter value (15) defines the start position of the designated data field, starting from at the beginning of the search string (`'Policy Number: '`)
- the 4th FIELD parameter value (12) defines the length of the designated data field.

Note: The FIELD-searchString is only searched for on the same page the trigger remains.

Using the method described in Fig. 21 you can also read information from AFP NOP structured fields.

In case of AFP data, CIS propagates all No-Operation (NOP, 0xD3EEEE) structured fields which are part of the page data (between BPG and EPG structured fields) and writes them immediately after the MO:DCA-P Active Environment Group in the output file. NOP's which are not part of the page data (outside BPG/EPG) are not propagated.

3.6.8 Compressing AFP and MO:DCA-P data

CIS allows an installation to reduce the size of the resulting output file without introducing a proprietary compression algorithm through the use of the PTOCA Repeat String (RPS) control sequence. The resulting output file is a standard AFP file where repeated characters are optimized using the RPS control sequence. Depending on the characteristics of the input file, a significant reduction in the size of the resulting output file may be achieved.

The compression feature in CIS is disabled by default as it introduces additional processing overhead. Using the COMPRESS keyword, an installation may activate this feature either in a generic way (all characters repeated more than 6 times consecutively are replaced with an RPS control sequence), or in a more specific way where one or more characters (or string of characters) are checked for repetition. Only the data characters contained in the Presentation Text Data (PTX) structured fields generated or selected by CIS are compressed. The information in other structured fields is not checked for compression. For more information about Presentation Text objects please refer to *Data Stream and Object Architectures: Presentation Text Object Content Architecture Reference*.

Fig. 22 shows an example on how to use the COMPRESS keyword to remove repeated blank characters, repeated strings consisting of a blank and an asterisk, and repeated strings with a given hexadecimal value.

```
COMPRESS ( STRING, ' ', ' *', X’4243’ )
```

Fig. 22 AFP PTOCA compression: example 1

Fig. 23 shows an example on how to use the COMPRESS keyword to optimise all strings where the same character is repeated more than 6 times consecutively.

```
COMPRESS ( CHARS )
```

Fig. 23 AFP PTOCA compression: example 2

3.7 How to Retrieve and Consolidate AFP Resources

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 requirement is not always easy to fulfill 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 on each print job.

CIS helps installations having these problems by providing a resource retrieval and consolidation feature. It can be used to locate and copy, into a separate Resource File, all the resources used by an application. The term "resource consolidation" is used to refer to the process where the AFP resources used in a print file are copied into a single output file or library, regardless of their library or resource group of origin. The feature can be activated selectively for each resource type (e.g. fonts, Page segments, Overlays, etc.).

CIS also includes a normalization step in case of non-OCA object resource files, which are automatically converted to AFP Object Containers before they are written to the Resource file.

The Resource consolidation process in CIS is done in parallel to the normalization, sorting and data enrichment steps (see Fig. 4 on page 32). Resources invoked as a result of a data enrichment operation (e.g. INSERTOBJECT or INSERTPTOCA) are also included in the external Resource group written to RESOBJDD.

In case of AFP resources, CIS propagates all No-Operation (NOP, 0xD3EEEE) structured fields which are part of an Overlay (between BMO and EMO structured fields). NOP's found in other resource types are not propagated.

The presentation (layout) of the print file is not affected. Please refer to [18] in *8 Bibliography* on page 465 for more information about the different resource grouping options available in MO:DCA.

3.7.1 Types of Resource Consolidation

The following types of resource consolidation are available in CIS:

Basic consolidation

A basic level of resource consolidation is part of the functionality available in CIS since version 4.00. This level of consolidation is the default and corresponds to the specification of RESCONSOLIDATION(BASIC). The following resource handling applies to basic resource consolidation:

- Only the first "external resource group" in the input print file (if any) is used during the resource search process. The resource group must be placed before any document data. External resource groups placed after the first document (or line of data) in the file are ignored. A resource, when not found in the first "external resource group", is read from an external library. The resources used are written to RESOBJDD under the control of the RESTYPE keyword.
- All "external resource groups" in the input print file (if any) are removed from the output file written to OUTPUTDD. These resources are also not copied to RESOBJDD.
- All "internal resource groups" in the print file (if any) are removed from the output file written to OUTPUTDD. These resources are also not copied to RESOBJDD.
- All "internal Medium Maps" in the print file (if any) are copied into the output file written to OUTPUTDD. They are not copied to RESOBJDD.

The output file (OUTPUTDD keyword) created by CIS when RESCONSOLIDATION(BASIC) is used **must** be printed using the resource file(s) created in the same CIS run (RESOBJDD keyword) **unless** the input print file did not include any external resource groups, in which case the original resource libraries may be used.

```
0000 00 D3A8C6 BRG Begin EXTERNAL resource group
0000 00 D3A8CE BR   Begin EXTERNAL resource
                ...
0000 00 D3A9CE ER   End resource <........>
0000 00 D3A8CE BR   Begin EXTERNAL resource
                ...
0000 00 D3A9CE ER   End resource <........>
0000 00 D3A9C6 ERG End EXTERNAL resource group

0000 00 D3A8A8 BDT Begin document
0000 00 D3A8AD BNG   Begin Named Page group
0000 00 D3A8AF BPG    Begin Page
0000 00 D3A8C6 BRG      Begin INTERNAL resource group (PAG level)
0000 00 D3A8CE BR        Begin INTERNAL resource
                    ...
0000 00 D3A9CE ER        End resource <........>
0000 00 D3A9C6 ERG      End INTERNAL resource group
0000 00 D3A8C9 BAG      Begin Active Environment group
                   ...
0000 00 D3A9C9 EAG      End Active Environment group
                 ...
0000 00 D3A9AF EPG    End Page
                ...
0000 00 D3A9AD BNG   End Named Page group
              ...
0000 00 D3A9A8 EDT End document
```

```
0000 00 D3A8C6 BRG Begin EXTERNAL resource group
0000 00 D3A8CE BR   Begin EXTERNAL resource
                   ...
0000 00 D3A9CE ER   End resource <........>
0000 00 D3A8CE BR   Begin EXTERNAL resource
                   ...
0000 00 D3A9CE ER   End resource <........>
0000 00 D3A9C6 ERG End EXTERNAL resource group

0000 00 D3A8A8 BDT Begin document
0000 00 D3A8AD BNG   Begin Named Page group
0000 00 D3A8AF BPG    Begin Page
0000 00 D3A8C6 BRG      Begin INTERNAL resource group (PAG level)
0000 00 D3A8CE BR        Begin INTERNAL resource
                       ...
0000 00 D3A9CE ER        End resource <........>
0000 00 D3A9C6 ERG      End INTERNAL resource group
0000 00 D3A8C9 BAG      Begin Active Environment group
                      ...
0000 00 D3A9C9 EAG      End Active Environment group
                    ...
0000 00 D3A9AF EPG    End Page
                   ...
0000 00 D3A9AD BNG   End Named Page group
                 ...
0000 00 D3A9A8 EDT End document
```

Fig. 24 AFP resource grouping supported by CIS

Consolidating External Resource Groups

The following resource handling applies when external resource consolidation is requested:

- All External Resource Groups in the input print file (if any) are used during the resource search process. Please refer to [18] in *8 Bibliography* on page 465 for more information about the different resource grouping options available in MO:DCA. Each group replaces a previous one, and must be placed before the associated document data. An External Resource Group stays valid for all the subsequent AFP documents, until the end of the file is reached or a BRG structured field is found. A resource, when not found in the corresponding "external resource group", is read from an external library. All resources used are written to RESOBJDD, regardless of their origin.
- Eventual naming conflicts are automatically solved by CIS by renaming the affected resources (except FORMDEF, see below) and the corresponding

resource invocations. Because of this renaming process, external resource group consolidation requires that the RESTYPE keyword specifies ALL.

Resources are renamed starting with the second external resource group. Resources are renamed when a resource with the same name was found in a previous "external resource group". The new name assigned to the resource is internally generated by CIS. **The renamed resources cannot be used for data enrichment operations.**

- In case of a FORMDEF, CIS merges the Medium Maps found in the multiple Form definitions in order to build a new, bigger, Form definition which contains all Medium Maps found in the external FORMDEF's used by the application.

  The Document Environment Groups (DEG) are also merged using the following rules:

  - The structured fields Map Medium Overlay (MMO), Medium Descriptor (MDD) and Page Position (PGP) are merged into the Medium Maps and are removed from the DEG.
  - The Map Suppression (MSU) structured fields of all DEG's are collected and become part of the new DEG. Duplicate suppressions (same ID and name) are removed. A maximum total of 127 suppressions is supported.
  - All Presentation Fidelity Control (PFC) structured fields are merged and become part of the new DEG. The less tolerant action takes precedence.
  - All Medium Finishing Control (MFC) structured fields are merged and become part of the new DEG. An Object Offset (x'5A') triplet is generated for each document within the scope of an MFC.

  Medium Map naming conflicts are automatically solved by CIS by renaming the affected Medium Maps and their invocations. Medium Maps are renamed starting with the second external resource group. Medium Maps are only renamed when a Medium Map with the same name was found in a previous "external resource group", or when a DEFINEMEDIUMMAP for that name was done (the rename is done after applying the parameters of the DEFINEMEDIUMMAP). The new names are internally generated by CIS (see also *3.8.8 Inserting and Removing additional Sheets* on page 96).

  Only one FORMDEF keyword may be specified to CIS. In order to use multiple Form Definitions, an application must either: a) name all FORMDEF's the same so that they match the name specified in the FORMDEF keyword, or b) include the desired FORMDEF as first Form definition in each of the external resource groups and specify FORMDEF (DUMMY).

- All External Resource Groups in the input print file (if any) are removed from the output file written to OUTPUTDD.

The output file (OUTPUTDD keyword) created by CIS when RESCONSOLIDATION(EXTERNALRESGROUP) is used **must** be printed using the resource file(s) created in the same CIS run (RESOBJDD keyword) **unless** the input print file did not include any external resource groups, in which case the original resource libraries may be used.

Consolidating Internal Resource Groups

The following resource handling applies when internal resource consolidation is requested:

- All Internal Resource Groups in the input print file (if any) are used during the resource search process. Please refer to [18] in *8 Bibliography* on page 465 for more information about the different resource grouping options available in MO:DCA. The scope of each resource group is specified by the MO:DCA architecture and which allows for nesting and inheritance. A resource, when not found in the current "internal resource group", is read from the external resource group currently active, or from an external library (in this sequence). All resources used are written to RESOBJDD, regardless of their origin.
- Name conflicts are avoided by CIS by renaming the affected resources and the corresponding resource invocations. Because of this renaming process, internal resource consolidation requires that the RESTYPE keyword specifies ALL.
- All "internal resource groups" in the input print file (if any) are removed from the output file written to OUTPUTDD.

The output file (OUTPUTDD keyword) created by CIS when RESCONSOLIDATION(INTERNALRESGROUP) is used **must** be printed using the resource file(s) created in the same CIS run (RESOBJDD keyword) **unless** the input print file did not include any internal **and** external resource groups, in which case the original resource libraries may be used.

Consolidating Internal Medium Maps

The following resource handling applies when resource consolidation for internal Medium Maps (also called "inline") is requested:

- All Internal Medium Maps in the input print file (if any) are activated as specified by the MO:DCA architecture. Please refer to [18] in *8 Bibliography* on page 465 for more information about Internal Medium Maps.
- Active Internal Medium Maps are removed from the input print file and are copied into the current FORMDEF. An Invoke Medium Map (IMM) structured field is inserted in the appropriate location within the print data stream. Eventual naming conflicts are automatically solved by CIS by renaming the affected components. Because of this process, internal Medium Map consolidation requires that the RESTYPE keyword specifies ALL or at least FDEF.

- The output file written to OUTPUTDD does not include any Internal Medium Maps.

The output file (OUTPUTDD keyword) created by CIS when RESCONSOLIDATION(INTERNALMEDIUMMAP) is used **must** be printed using the resource file(s) created in the same CIS run (RESOBJDD keyword) **unless** the input print file did not include any internal medium maps **and** external resource groups, in which case the original resource libraries may be used.

3.7.2 Input specifications when using External Resource libraries

The CIS keyword interface allows installations to define different resource libraries to be used during the retrieval phase. The FDEFLIB, PDEFLIB, FONTLIB, OVLYLIB, PSEGLIB and OBJCONLIB keywords may be used to specify one or more resource libraries to be used for retrieving a particular resource type. The USERLIB keyword adds an extra level of library differentiation. The RESTYPE keyword can be used to select which resource types are to be retrieved.

```
     RESTYPE ( ALL )
RESCONSOLIDATION ( BASIC )
     USERLIB ( CIS.AFPLIB )
     FONTLIB ( HLD.FONT300,
          HLD.FONT600 )
  CODEDFONTEXT ( ".300",
          ".600",
          "" )
      CHARS ( X0abc )
```

Fig. 25 Resource Packaging keywords

When the Operating system platform allows, CIS also provides a set of keywords that can be used to specify the suffix(es) or file extension(s) to be appended to the resource name during the resource file search process. The CODEDFONTEXT, CODEPAGEEXT, FONTCHARSETEXT, OUTLINEFONTEXT, FORMDEFEXT, OBJCONTEXT, OVERLAYEXT, PAGEDEFEXT and PAGESEGEXT keywords control this operation. Each directory specified in a xxxLIB keyword is searched for the first occurrence of a resource file with one of the extensions specified in the corresponding xxxEXT keyword before continuing with the next directory.

Example:

Using control file from Fig. 25 on page 74 when searching for a coded font named X0abc:

1. Library "HLD.FONT300" is searched for the following file names: X0abc.300, X0abc.600 and X0abc

2. Library "HLD.FONT600" is searched for the following file names: X0abc.300, X0abc.600 and X0abc
3. Library "CIS.AFPLIB" is searched for the following file names: X0abc.300, X0abc.600 and X0abc

The first occurrence will stop the search and that resource file will be used. Resource names are case sensitive under Linux. MVS resource names are not case sensitive and may cause a resource file with the name X0ABC to be selected.

CIS may also be used to include resources which are not part of the print data stream, but that are specified in the control file. (e.g. Overlays, CHARS, etc.). It may also be used to override some of the 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 CIS will include the extensions or modifications requested in the control file so that recreating an exact printout does not require the control file. For more information on this topic please refer to *3.8.11 Overriding other FormDef / PageDef parameters* on page 107.

3.7.3 Output specifications for the consolidated External Resource Group

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

In combination with inline resources, the RESFILE (PDS) keyword allows an installation to extract resources which are part of an input file and to write them into separate resource files using the specifications of the RESOBJDD keyword. Under MVS, RESFILE (PDS) requires that the DDname in RESOBJDD corresponds to a Partitioned Data set (DSORG=PO).

<u>Example</u>:

Using the application shown in *Fig. 18 Legacy application* on page 60, the following CIS resource packaging keywords would request the generation of a Resource Object file to be archived together with the document and index files:

```
 RESTYPE ( ALL )
 USERLIB ( CIS.AFPLIB )
 FONTLIB ( HLD.FONT300 )
 FORMDEF ( F1CIS02 )
 PAGEDEF ( P1CIS02 )

   CC ( ASA )
 TRCTYPE ( IBM )

RESOBJDD ( MYRESLIB )
```

```
RESCONSOLIDATION ( EXTERNALRESGROUP,
          INTERNALRESGROUP,
          INTERNALMEDIUMMAP )
```

Fig. 26 Resource Consolidation keywords

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

```
0000 00 D3A8C6 BRG Begin resource group <........>        1
0000 00 D3A8CE BR   Begin resource <F1CIS02 >          2
0001 00 D3A8CD BFM    Begin form map <........>        3
0008 00 D3A8CC BMM      Begin medium map <CIS2F1 >     10
...
0011 00 D3A9CC EMM      End medium map <CIS2F1 >      19
0012 00 D3A9CD EFM    End form map <........>         20
0000 00 D3A9CE ER   End resource <F1CIS02 >          21
0000 00 D3A8CE BR   Begin resource <S1CIS2F >         22
...
0000 00 D3A9CE ER   End resource <S1CIS2F >          96
0000 00 D3A8CE BR   Begin resource <S1CIS2B >         97

...

0000 00 D3A8CE BR   Begin resource <C0ARI10N>         652
...
0000 00 D3A9CE ER   End resource <C0ARI10N>          666
0000 00 D3A9C6 ERG End resource group <........>       667
```

Fig. 27 Resource Object file

3.8 How to Enrich Print Data streams

CIS may be used to enrich an existing print application by inserting different data objects at levels such as Page, Sheet, Mail piece or Document. The objects types that may be inserted during data enrichment operations include:

- Text strings and Marks (PTOCA)
- Bar Codes (BCOCA)
- Index tag elements (TLE)
- Graphics (GOCA)
- Object containers (OCA and non-OCA)
- Finishing operations (UP³I based)

Data enrichment consists of inserting (or removing) information stored in a print file, after the file was generated. With CIS, the application programs do not need to be changed; the application jobs do not need to be rerun; the print file does not need to be re-created. Data enrichment may be used in combination with other CIS features. As Fig. 4 on page 32 shows, the internal CIS data enrichment process is divided in two parts: a removal phase and an insertion phase. The objects selected for removal are deleted as the input data is converted and normalized. Object insertion occurs later in the workflow, after the normalization, selection and sorting steps are completed.

CIS requires some definitions to be done before an input print file can be enriched. These definitions identify the kind of data enrichment operation to be applied (e.g. a bar code or a graphic box), the location in the print file where the operation is to be inserted (e.g. in the front page of every document) and the data to be inserted (e.g. the bar code value). These definitions are done using one of the following keywords by specifying them in the control file:

1. Define the kind of data enrichment operation to be applied, including formatting keywords (if any):

define2DDataMatrix	define2DMaxiCode	define2DPDF417
define2DQRCode	defineBCOCA	defineBOX
defineCIRCLE	defineCOLOR	defineCONTAINER
defineCURVE	defineELLIPSE	defineFINISHINGOP
defineLINE	defineMEDIUMMAP	definePTOCA

2. Specify the location within the file and the data value to be inserted (if any):

insertBCOCA	insertBOX	insertCIRCLE
insertCONTAINER	insertCURVE	insertELLIPSE
insertFINISHINGOP	insertINDEX	insertLINE
insertPTOCA	insertPTOCANOP	insertSHEET

For more flexibility, the defineLEVEL keyword can be used to specify which AFP entity(ies) are going to be enriched and when.

3. Define the kind of data objects to be removed:

removeBCOCA	removeBOX	removeCIRCLE
removeCONTAINER	removeCURVE	removeELLIPSE
removeFINISHINGOP	removeINDEX	removeLINE
removeMEDIUMMAP	removePTOCA	removeSHEET

The data enrichment process in CIS is done as requested by the user. CIS cannot guarantee that the requested function is printable under all circumstances as this is dependent on the combination of keywords, variable data and printer equipment being used.

This section provides information and shows some examples on how to use the CIS data enrichment features.

3.8.1 Defining data enrichment levels to CIS

Before a data enrichment operation can be done, CIS needs to know **what** is going to be enriched and **where**. The DEFINELEVEL keyword is used for this purpose, and allows an installation to define what, where and **which** entity(ies) are to be enriched. Fig. 28 on page 78 shows the combination of data enrichment operation, entity and location which are supported by CIS. In the case of Finishing Operations, the level also determines the scope of the operation, that is, **how long** should it be applied.

CIS level definition works on the base of information units called *entities*. The entities available for data enrichment operations are *document*, *sheet-group*, *sheet*, and *page*. Please refer to *7 Glossary* on page 461 for a definition of these terms. CIS also provides a number of pre-defined levels. Please refer to *4.4.5 Predefined Levels* on page 165 for a list of the pre-defined levels available.

Fig. 29 shows the combination of entity, selector and location parameters allowed in a DEFINELEVEL keyword used in connection with data enrichment.

```
defineLEVEL ( backSheetSide, SHEET, BACK, ALL )
defineLEVEL ( mpBackSide,GROUPLEVEL1, BACK, ALL )
defineLEVEL ( docFront,DOCUMENT, FRONT, ALL )
```

Fig. 28 Example: CIS data enrichment levels

<u>Example</u>:

Fig. 28 shows an example which:

1. Defines level 'backSheetSide' to be the back side of every sheet:
2. Defines level ' mpBackSide ' to be back side of the last sheet of every mailpiece:
3. Defines level 'docFront' to be front side of the first sheet of every document:

Operation	"what"	"where"	"which"
• Presentation objects: insertBCOCA, insertBOX, insertCIRCLE, insertCONTAINER, insertCURVE, insertELLIPSE, insertLINE, insertPTOCA, insertPTOCANOP	✔ **DOCUMENT** ✔ **GROUPLEVELn** ✔ **SHEET**	✔ **BEGIN** ✔ **END** ✔ **FRONT** ✔ **BACK** ✔ **ALL**	✔ **ALL** ✔ ***name*** ✔ ***n*** ✔ ***from-to*** ✔ **ODD** ✔ **EVEN**
	✔ **PAGE**	✔ **ALL**	✔ **ALL** ✔ ***name*** ✔ ***n*** ✔ ***from-to*** ✔ **ODD** ✔ **EVEN** ✔ **FRONT** ✔ **BACK**
• Indexing objects insertINDEX	✔ **DOCUMENT** ✔ **GROUPLEVELn** ✔ **GROUPLEVEL1** ✔ **SHEET**	✔ **BEGIN, END** ✔ **BEGIN** ✔ **END** ✔ **BEGIN, END**	✔ **ALL** ✔ ***name*** ✔ ***n*** ✔ ***from-to*** ✔ **ODD** ✔ **EVEN**
	✔ **PAGE**	✔ **ALL**	✔ **ALL** ✔ ***name*** ✔ ***n*** ✔ ***from-to*** ✔ **ODD** ✔ **EVEN**
• Finishing Operations insertFINISHINGOP	✔ **DOCUMENT** ✔ **GROUPLEVELn** ✔ **SHEET**	✔ **ALL**	✔ **ALL** ✔ ***name*** ✔ ***n*** ✔ ***from-to*** ✔ **ODD**

Operation	"what"	"where"	"which"
			✔ EVEN
• Other insertSHEET	✔ **DOCUMENT**	✔ **BEGIN** ✔ **END**	✔ **ALL** ✔ ***name*** ✔ ***n*** ✔ ***from-to*** ✔ **ODD** ✔ **EVEN**
	✔ **GROUPLEVELn**	✔ **BEGIN** ✔ **END** ✔ **BEFORE** ✔ **AFTER**	✔ **ALL** ✔ ***name*** ✔ ***n*** ✔ ***from-to*** ✔ **ODD** ✔ **EVEN**
	✔ **SHEET**	✔ **BEFORE** ✔ **AFTER**	✔ **ALL** ✔ ***name*** ✔ ***n*** ✔ ***from-to*** ✔ **ODD** ✔ **EVEN**

Fig. 29 CIS data enrichment: levels allowed

3.8.2 Reference point used for positioning data enrichment objects

In the MO:DCA architecture, the position and orientation of a BCOCA, PTOCA and other presentation objects is specified using x_p , y_p co-ordinates that have the Page Origin as reference point (where both co-ordinate values are zero). Similarly, an application (FormDef) specifies the position of the Page Origin using x_m , y_m co-ordinates that have the Media Origin as reference point.

CIS requires that the position and orientation specified in the data enrichment keywords (defineBCOCA, defineCONTAINER and definePTOCA) use the **Media Origin** as reference point (there is one exception in case of NUP, see below). By doing this, an installation can specify a position that is independent of the Page Origin which normally changes from application to application.

Which corner is actually used depends on the medium type (fanfold or cut-sheet), dimensions (fanfold wide or fanfold narrow), presentation (landscape, portrait) and type of partitioning (1UP, X2UP, CSE) being used. The position of a BCOCA, Container or

PTOCA object inserted by CIS is specified either as a position (offset) relative to the Media Origin or as a position relative to the Partition Origin, depending on the NUP settings in the FormDef and on the data enrichment level specified in the insertBCOCA, insertCONTAINER and insertPTOCA keywords.

In the case of non-NUP print files, the Media Origin, as defined by the MO:DCA architecture, is used as reference point for all data enrichment entities (DOC, MP, SH and PAG).

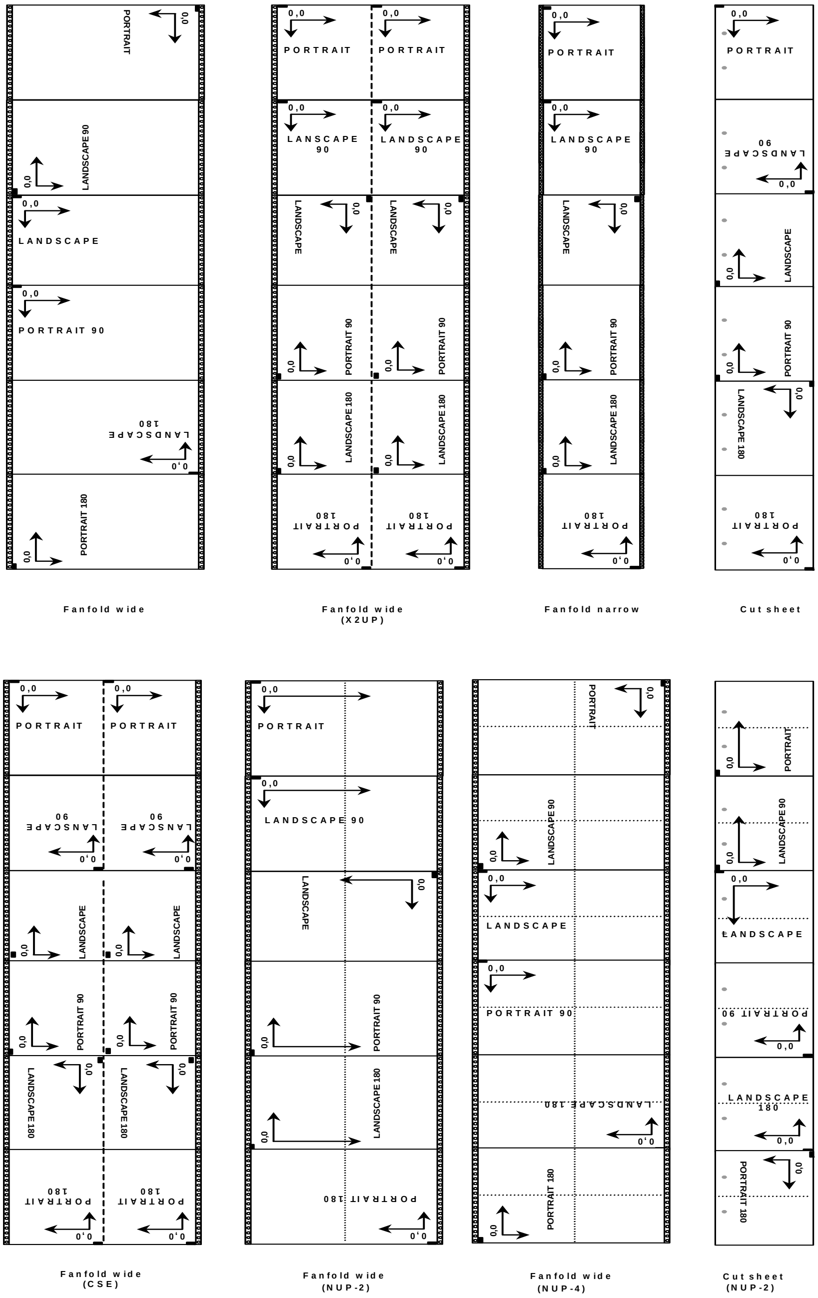


Fig. 30 Media Origin for data enrichment

In the case of NUP print files, the Media Origin is used as reference point when inserting data enrichment objects at the document, mailpiece or sheet levels (DOC, MP and SH). The Partition Origin is used for all data enrichment operations in an NUP file when page level is requested (PAG).

Following the AFP/IPDS rules, CIS considers the two sheetlets in a X2UP or CSE sheet as if they were two separate pieces of cut-sheet media and defines two different Media Origin points. This is not the case of NUP printed sheets where only one Media Origin is defined, regardless of the number of partitions being printed on the sheet.

Fig. 30 *Media Origin for data enrichment* on page 82 shows the Media Origin for the most common paper and partitioning formats. A much more comprehensive list of examples can be found in [18] (see *8 Bibliography* on page 465) under “Cut-Sheet Emulation Print mode” and under “Page Position PGP”.

3.8.3 Inserting and Removing Text Objects (PTOCA)

CIS may be used to insert text strings into an existing print file. For example:

- A print file requires sorting and a footnote added to every sheet displaying the new page sequence.
- Data selection is done and every Mail Piece selected is to display its new number and size in sheets.
- A print shop wants to display the actual printing/archiving date and not the date the application was run.
- A bar code is to be printed using special fonts. This technique is used when the printer (or viewer) does not support BCOCA objects.
- Information only available in the Index Tag Elements (TLE’s) contained in the print file is to be viewed or printed.
- A message, advertisement, or any other kind of text string should be printed.

This strings are inserted as Presentation Text Objects embedded in a Transparent Data (TRN) escape sequence. For this reason, the string should only contain printable data. Any control sequence it may include are handled as such. Any of the formatting options described under *4.4.2 Formatting string* on page 161 can be used.

The INSERTPTOCANOP keyword can be used to request CIS to insert the Presentation Text Object embedded in a No Operation (NOP) escape sequence.

<u>Example:</u>

Using the application shown in *Fig. 18 Legacy application* on page 60 and the indexing keywords shown in Fig. 16, an installation wants to add a footnote to every sheet. The footnote will be used by the printer operators to clearly identify the printed material.

The text should display, for every mail piece, the account number, the sheet number, the total number of sheets in the mail piece, and the actual date. The new text is in German and is printed using a very small font. The position where the text is printed is normally specified using the Media Origin as reference point (see *3.8.2 Reference point used for positioning data enrichment objects* on page 80 for more information).

Fig. 31 shows the control file required for our example. The INITVARIABLES keyword is used to set initial values for the date variables. The information contained in each FIELD is defined next. Field "accountF" is shown in Fig. 16.

The **definePTOCA** keyword is used to define the characteristics of the text to be inserted (e.g. position, rotation, color, font used, etc.). It also assigns a name to the PTOCA definition.

The **removePTOCA** keyword is used to remove previous inserts (e.g. to update the date or change the language).

Finally, the **insertPTOCA** (or **insertPTOCANOP**) keyword specifies the user defined level and the formatting options for the string to be inserted. PTOCA objects may be inserted in Document, Mail Piece, Sheet or Page entities. Where inside the entity and on which entities the data enrichment operation will be applied is defined using the DEFINELEVEL keyword. The USEREXIT parameter allows an installation to dynamically generate the text string to be inserted (see *5.8 PTOCA-insertion exit* on page 305 for more information). Specifying USEREXIT causes the CIS normalization process to store all the input data into the internal work file before the conversion process is started.

```
initVARIABLES ( _dateDD = 20, _dateMM = 01, _dateYYYY = 2001 )

    FIELD ( sheetNumber, _outSheetNumberInMailPiece )
    FIELD ( numberOfSheets, _outNumberOfSheetsInMailPiece )
    FIELD ( day, _dateDD )
    FIELD ( month, _dateMM )
    FIELD ( year, _dateYYYY )
    FIELD ( accountINDEX, 'ACCOUNT', ATT )

 definePTOCA ( 'footNote',      /* Name of this PTOCA definition   */
        MM,         /*  size and position unit     */
        100,        /*  X offset from Media Origin   */
        270,        /*  Y offset from Media Origin   */
        ,          /*  rotation (default = 0°)     */
        ,          /*  color  (default = black)   */
        X0MYFONT,      /*  Coded Font to be used      */
        IBM_0273      /*  Code Page required by this font */
       )

 removePTOCA ( 'footNote' )    /* Remove PTOCA before inserting new */

 insertPTOCA ( 'footNote', SH,
        '%s * Sheet %d of %d * printed at %02.2d.%02.2d.%04.4d',
        accountINDEX,
        sheetNumber,
```

```
          numberOfSheets,
          day,
          month,
          year
         )
```

Fig. 31 Insert and Remove PTOCA keywords

Inserting and/or removing PTOCA objects only applies to the print file and not to the resources it may use. Fig. 32 shows the resulting text string:

```
14034350 * Seite 3 von 5 * Gedruckt am 20.01.2001
```

Fig. 32 Insert PTOCA keywords: resulting string

3.8.4 Inserting and Removing Bar Code Objects (BCOCA)

Bar Code insertion in CIS works in a way similar to the insertion of text strings. The important difference: the objects inserted are embedded in BCOCA structured fields which are built according to the specifications in the CIS Control file.

```
       FIELD ( jobID, 'JOB00123' )
       FIELD ( mpNum, _outMailPieceNumber )
       FIELD ( mpSize, _outNumberOfSheetsInMailPiece )
       FIELD ( shNum, _outSheetNumberInMailPiece )
       FIELD ( pagNum, _outPageNumber )
       FIELD ( accountINDEX, 'ACCOUNT', ATT )

    defineBCOCA ( 'mailPieceBC',   /*  Name of this BCOCA definition  */
           MM,         /*  size and position unit     */
           50,         /*  X offset from Media Origin   */
           50,         /*  Y offset from Media Origin   */
           ,          /*  rotation (default = 0°)     */
           ,          /*  color  (default = black)   */
           ,          /*  type              */
           ,          /*  modifier            */
           ON,         /*  HRI specification        */
           ,          /*  start/stop characters (CODE39) */
           10,         /*  height specification      */
           ,          /*  module width          */
           100,        /*  width ratio           */
```

```
                    /*  code page ID          */
          )

    defineBCOCA ( 'sheetBC',     /* Name of this BCOCA definition   */
           MM,         /*  size and position unit     */
           10,         /*  X offset from Media Origin   */
           100,        /*  Y offset from Media Origin   */
           270,        /*  rotation (default = 0°)     */
           , , , ON, , 10, , 100, )

    defineBCOCA ( 'pageBC',
           MM,
           60,
           100,
           , , DATAMAT, 0, , , , , , )

define2DDataMatrix ( 'pageBC', /* Name corresponding to defineBCOCA      */
           YES,   /* escape-sequence handling           */
           NO,    /* EBCDIC-to-ASCII translation         */
           0,    /* matrix row size              */
           0,    /* number of rows               */
           0,    /* sequence indicator              */
           0,    /* total number of symbols           */
           0x01,   /* high-order byte               */
           0x01,   /* low-order byte                */
                /* function flags (default = no flags specified */
          )

    removeBCOCA ( 'mailPieceBC', 'sheetBC' )   /* Remove BCOCA objects  */

    insertBCOCA ( 'mailPieceBC', MP,
           '%s%s%06.6d%02.2d',
           jobID,
           accountINDEX,
           mpNum,
           mpSize
          )

    insertBCOCA ( 'sheetBC', SH, '%06.6d%02.2d', mpNum, shNum )

    insertBCOCA ( 'pageBC', PAG, '%s%06.6d', jobID, pagNum )
```

Fig. 33 Insert and Remove BCOCA keywords

The **defineBCOCA** keyword allows an installation to specify the characteristics of the Bar Code Object to be inserted, and includes information such as type, position, orientation, color, height, width ratio, etc.

The **insertBCOCA** keyword specifies the level and the contents of the bar code to be inserted. BCOCA objects may be inserted in Document, Mail Piece, Sheet or Page entities. Where inside the entity and on which entities the data enrichment operation will be applied is defined using the DEFINELEVEL keyword. The string parameter is used as template and is merged with the data fields specified. Any of the formatting options described under *4.4.2 Formatting string* on page 161 can be used. The USEREXIT parameter allows an installation to dynamically generate the contents of the BCOCA object be inserted (see *5.7 BCOCA-insertion exit* on page 303 for more information). Specifying USEREXIT causes the CIS normalization process to store all the input data into the internal work file before the conversion process is started.

The **removeBCOCA** keyword can be used to remove existing BCOCA objects. Bar Code removal is done using the name specification. It requires that the existing objects in the print file were clearly labeled by the application program, and that the labels are consistently used in the whole print file.

Inserting and/or removing BCOCA objects only applies to the print file and not to the resources it may use. More information about the BCOCA architecture can be found in [12] (see *8 Bibliography* on page 465).

The Code Page support provided in connection with bar codes remove bar code type dependencies from an application and allows installations to adjust their print output to different Bar Code standards without having to modify the application programs. With this support, an EBCDIC print file may be enhanced with bar codes requiring ASCII data having CIS doing the necessary code page translation. (refer to *3.2 Code page support* on page 33 for more information).

The position where the bar code is printed is normally specified using the Media Origin as reference point (see *3.8.1 Defining data enrichment levels to CIS* on page 78 for more information).

Example:

Using the application shown in *Fig. 18 Legacy application* on page 60 and the indexing parameters shown in Fig. 16, an installation wants to add bar codes to every mail piece and to every sheet. The Mail Piece level bar codes should contain an installation defined Job ID, the account number, the Mail Piece number and the number of sheets in the Mail Piece. The Sheet level bar codes should contain the Mail Piece number and the relative Sheet number within the Mail Piece. The CIS control file specifications are shown in Fig. 33. Fig. 34 shows the resulting bar code data.

sheetBC:

mailPieceBC:

J0B001238544071900000101

00000101

pageBC:

Fig. 34 Insert and Remove BCOCA keywords: resulting strings

3.8.5 Inserting and Removing Graphic Objects (GOCA)

Graphics insertion in CIS works in a way similar to the insertion of bar codes or text strings, with the difference that not one but a whole group of keywords is available for defining, inserting and removing the graphic objects. These keywords are explained below.

Standard (closed) figures

The **defineBOX, defineCIRCLE** and **defineELLIPSE** keywords allow an installation to specify the characteristics of the rectangular, circular and elliptical figures that are to be inserted in form of GOCA objects. They include information such as position, size, line type and width, fill color and pattern , etc. The origin used as reference point for positioning the graphic objects is described in detail under *3.8.1 Defining data enrichment levels to CIS* on page 78.

The **insertBOX**, **insertCIRCLE** and **insertELLIPSE** keywords specify the level for the graphic object to be inserted. GOCA objects may be inserted in Document, Mail Piece, Sheet or Page entities. Where inside the entity and on which entities the data enrichment operation will be applied is defined using the defineLEVEL keyword.

The **removeBOX, removeCIRCLE** and **removeELLIPSE** keywords can be used to remove existing GOCA objects. Only boxes, circles or ellipses previously inserted using the corresponding INSERT keyword can be removed.

Inserting and/or removing GOCA objects only apply to the print file and not to the resources it may use. More information about the GOCA architecture can be found in [14] (see *8 Bibliography* on page 465).

Example:

Fig. 35 lists the keywords required to obtain the results shown in 0.

```
  DefineBOX( 'Box1', cm, 15, 4, 4, 12, , 7, 15, green, 1, , , 0x10, white )
  DefineBOX( 'Box2', cm, 3,13, 5, 15, none, 8, , , 4, 3, 0, 11, red )
  DefineBOX( 'Box3', cm, 17,17,19, 20, none, , , , , , , , )

 DefineCIRCLE( 'Circle1', mm, 40, 40, 25, 3, 15, blue, , , , 8, yellow )
DefineELLIPSE( 'Ellipse1',cm, 5, 20, 2, 1, 2, 0, 0x07, 8, red, 2, 5, 0,0x10, green )
  InsertBOX( 'Box1', PAG )
  InsertBOX( 'Box2', PAG )
  InsertBOX( 'Box3', PAG )
 InsertCIRCLE( 'Circle1', PAG )
InsertELLIPSE( 'Ellipse1', PAG )
```

Fig. 35 Example: defineBOX, defineCIRCLE and defineELLIPSE keywords

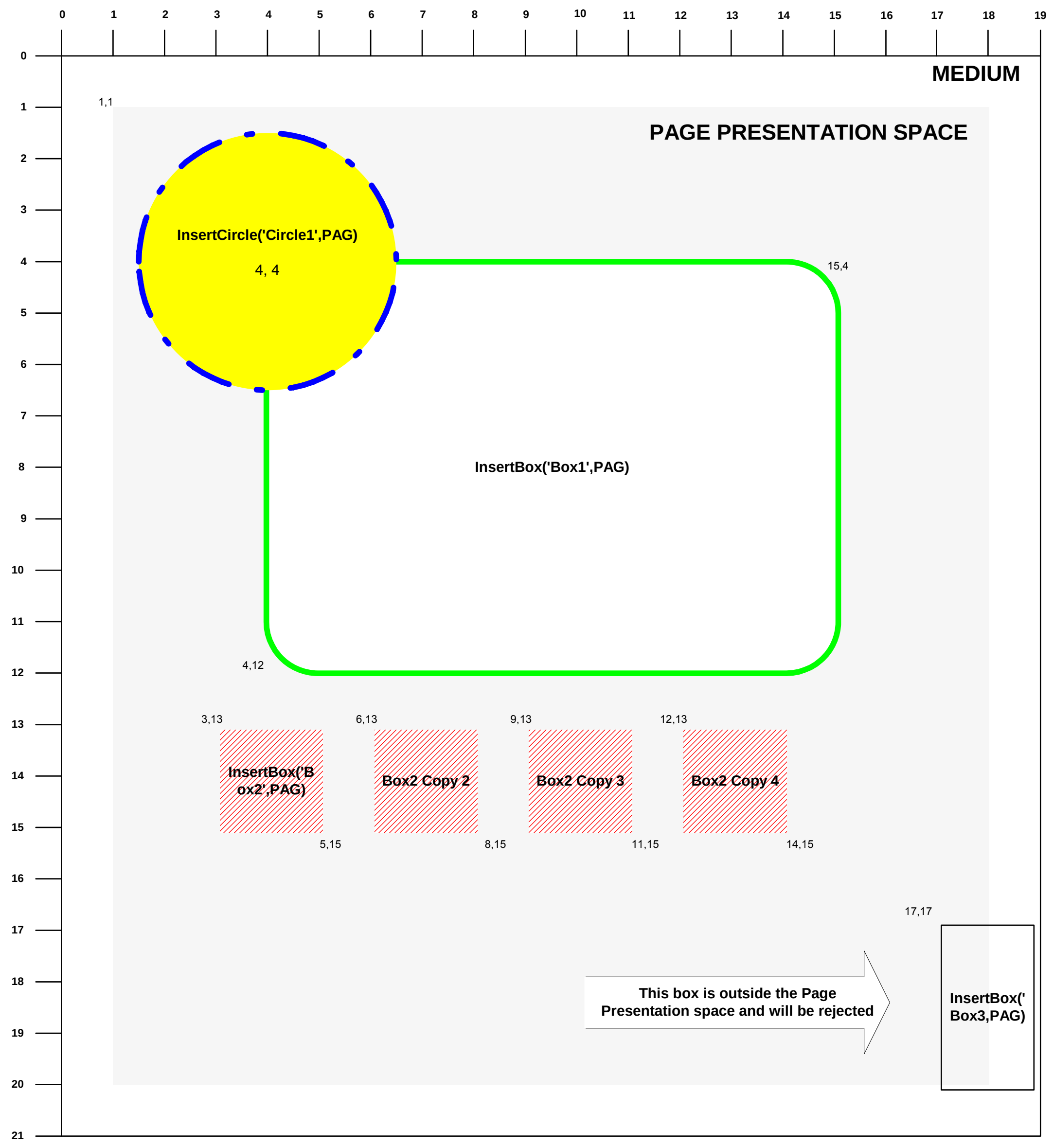


Fig. 36 Example: defineBOX, defineCIRCLE and defineELLIPSE results

Non-standard (open or closed) figures

Open figures can be draw using the **defineCURVE** and **defineLINE** keywords by specifying different start/end points. These keywords may also be used to draw non-standard closed figures by specifying the same start/end points.

Each of these figures may contain up to 62 segments, and are inserted in form of GOCA objects. The origin used as reference point for positioning the graphic objects is described in detail under *3.8.1 Defining data enrichment levels to CIS* on page 78.

The **insertCURVE** and **insertLINE** keywords specify the level for the graphic object to be inserted. GOCA objects may be inserted in Document, Mail Piece, Sheet or Page entities. Where inside the entity and on which entities the data enrichment operation will be applied is defined using the DEFINELEVEL keyword.

The **removeCURVE** and **removeLINE** keywords can be used to remove existing GOCA objects. Only curves or lines previously inserted using the corresponding INSERT keyword can be removed.

Inserting and/or removing GOCA objects only applies to the print file and not to the resources it may use. More information about the GOCA architecture can be found in [14] (see *8 Bibliography* on page 465).

Example:

Fig. 37 lists the keywords required to obtain the results shown in 0.

```
 defineLINE ( 'Linie1', cm, 4,12, 12, 4, 7,15, green, 1, , ,  ,    )
 defineLINE ( 'Linie2', cm, 6, 3, 3, 3,
                  3, 5,
                  6, 7, 3,10,  red, , , , 16, yellow )
 defineLINE ( 'Linie3', cm, 3,13, 3, 18, 1, 8, black, 4, 2, 0,  ,    )
 defineLINE ( 'Linie4', mm,130,60,160, 60, , ,   , 4, 0, 2,  ,    )
 defineLINE ( 'Linie5', cm, 12,15, 12, 16,
                  13, 17,
                  16, 17,
                  17, 16,
                  17, 15, 7,10, blue, , , ,  ,    )
DefineCurve ( 'curv01', , 240, 240, 120, 240, 300, 300, , , , , , , ,    )

 InsertLINE ( 'Linie1', PAG )
 InsertLINE ( 'Linie2', PAG )
 InsertLINE ( 'Linie3', PAG )
 InsertLINE ( 'Linie4', PAG )
 InsertLINE ( 'Linie5', PAG )
 InsertCurve ( 'curv01', PAG )
```

Fig. 37 Example: defineCURVE and defineLINE keywords

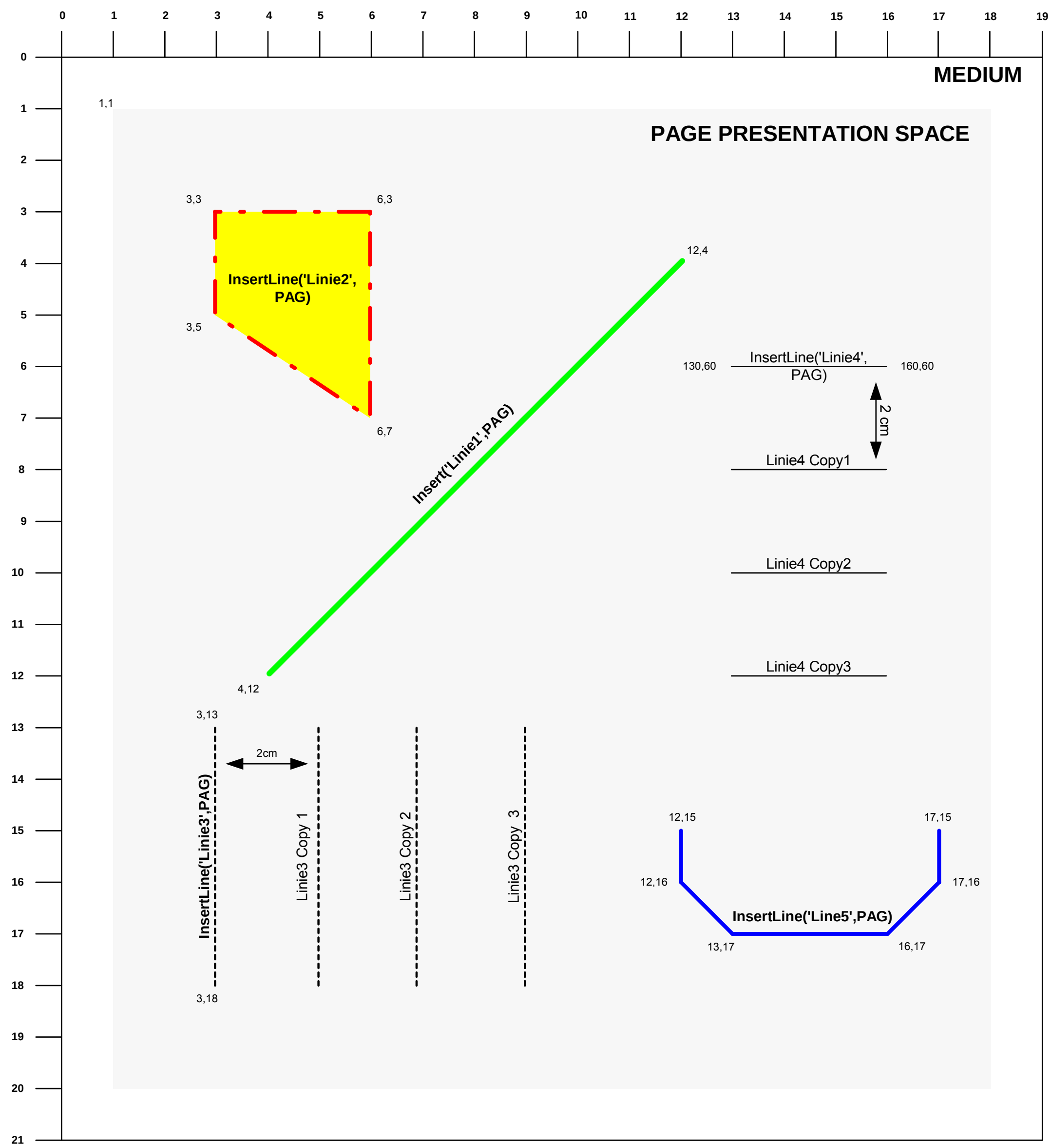


Fig. 38 **Example: defineCURVE, defineLINE results**

3.8.6 Inserting and Removing Index tag Elements (TLE's)

CIS offers two ways of adding Index information to an existing print file:

1. Using the TRIGGER, INDEX and FIELD keywords as explained under *3.6 How to convert, normalize and index data* on page 55.
2. Using the FIELD and insertINDEX keywords as described below.

In the first case, the indexing information is obtained from the print data. CIS uses the TRIGGER keyword to define a reference point, and the FIELD keyword to define one or more data fields used to build the index record.

In the second case, the indexing information is not obtained from the print data. It is obtained from the CIS processing variables and, if requested, from already existing index tags. The **FIELD** and **insertINDEX** keywords are used for this purpose. The **removeINDEX** keyword can be used to remove existing index tags, regardless of the method used to insert them.

The USEREXIT parameter of the **insertINDEX** keyword allows an installation to dynamically generate the index to be inserted (see *5.10 INDEX-insertion exit* on page 309 for more information). Specifying USEREXIT causes the CIS normalization process to store all the input data into the internal work file before the conversion process is started.

The Code Page support provided in connection with Index insertion allows installations to generate Index Tag Elements using different data encoding schemes. With this support, an EBCDIC print file may be enhanced with TLE’s displaying ASCII encoded information, having CIS doing the necessary code page translation. (refer to *3.2 Code page support* on page 33 for more information). The indexCP keyword is used to specify the Code Page to be used to generate the TLE’s inserted in an CIS run. Already existing TLE’s maintain their original encoding.

Example:

An installation wants to extend the example shown in Fig. 33 so that the print file is viewed online. The online user should have fast online access to the Mail Pieces and Sheets. The data reported from Bar Code readers will be used to do the inquiry. In this case, the insertINDEX keyword of CIS can be used to generate the required indexing information so that any AFP viewer is able to handle the task. Fig. 39 shows the keywords required.

Index Tag Elements may be inserted at in Document, Mail Piece, Sheet or Page entities. In the case of Documents, the TLE’s are inserted in the first Mail Piece. In case of Sheet levels, the TLE’s are inserted in the first logical page. The string parameter is used as template and is merged with the data fields specified. Any of the formatting options described under *4.4.2 Formatting string* on page 161 can be used.

As Fig. 4 on page 32 shows, the Index insertion process takes place after the normalization, selection and sorting steps are completed. For this reason, it is not possible to use the Index Tag Elements inserted with the insertINDEX keyword to control the selection or sorting process of the same CIS run. Splitting the index insertion and sorting processes in two CIS runs will bypass this problem.

```
  indexCP ( IBM_0500 )     /* Code Page specification      */

removeINDEX ( 'mailPieceINDEX', 'sheetINDEX' )

insertINDEX ( 'mailPieceINDEX', MP,
       '%s%s%06.6d%02.2d',
       jobID,
       accountF,
       mpNum,
       mpSize
      )

insertINDEX ( 'sheetINDEX', SH, '%06.6d%02.2d', mpNum, shNum )
```

Fig. 39 Insert and Remove Index Tag Elements

3.8.7 Inserting and Removing AFP Object Containers

The data enrichment support in CIS includes support for Object Containers. Object Containers are MO:DCA objects that envelop and carry object data. The Object Containers supported by CIS data enrichment operations must be presentation objects (printable), must be paginated (constrained to a single page) and must be time-invariant (do not change).

Object Container insertion in CIS works in a way similar to the insertion of other MO:DCA objects such as text strings, bar codes (BCOCA) or graphics (GOCA).

The **defineCONTAINER** keyword allows an installation to specify the characteristics of the Object Container to be inserted, including information such as position, size, orientation, color, type, fileName, etc. The origin used as reference point for positioning the container data is described in detail under *3.8.1 Defining data enrichment levels to CIS* on page 78.

The **insertCONTAINER** keyword specifies the level controlling the insert operation. Object Containers may be inserted in Document, Page Group, Sheet or Page entities. Where inside the entity and on which entities the data enrichment operation will be applied is defined using the DEFINELEVEL keyword. The contents of the Object Container is static (printable, single page, time invariant). The types of Object Container supported include OCA objects (such as Page Segments, GOCA graphics, BCOCA bar codes and IOCA images) and non-OCA objects (such as Encapsulated Post Script, TIFF images and JPEG images in file interchange format). Non-OCA objects may be already embedded in a MO:DCA container envelop or may get the envelop automatically created by CIS. The USEREXIT parameter allows an installation to dynamically generate the contents of the Object Container to be inserted (see *5.11 ObjectContainer-insertion exit* on page 311 for more information).

The **removeCONTAINER** keyword can be used to remove existing Objects. Object Container removal is done using the name specification. It requires that the existing objects in the print file were clearly labeled by the application program, and that the labels are consistently used in the whole print file.

Object Container insertion is done in two steps (Fig. 4 on page 32 shows the CIS internal workflow):

1. The normalization step locates the Object file defined using the DEFINECONTAINER keyword and builds the Object Container resource. This step also builds or modifies the MO:DCA envelope with the parameters from DEFINECONTAINER. The resulting resource file is handled as a normal AFP resource and is automatically included in the Resource Object file (see RESOBJDD and RESTYPE keywords).
2. The data enrichment step at the beginning of the conversion phase identifies the entity(ies) where the Object Container is to be inserted, and adds an Include Object (IOB) structured field immediately after the Active Environment Group (AEG) of the target page. The standard IOB inserted by CIS will use the origin, orientation and other parameters from the Object Container resource (which was modified to match the DEFINECONTAINER parameters).

   In case a USEREXIT is active, the parameters modified by the exit are set in the IOB so that they override the object parameters. In case of exits inserting non-OCA data, the CIS conversion phase automatically creates an AFP Object Container envelope for the data which is then inserted directly after the AEG of the target page. No IOB is used in this case. The Object Container is not included in the Resource Object file. Specifying USEREXIT causes the CIS normalization process to store all the input data into the internal work file before the conversion process is started.

In case of a name conflict -when an Object Container with the same name is already present in the print file- CIS replaces the existing container with the new one. Any references to the old container will now use the new one.

```
 removeCONTAINER ( 'photo1' )


defineCONTAINER ( 'photo1', MM, 100, 50, 20, 20, 0x60, 1, 1, 0,
          none, TIFF, SOFT, MYFILE.A.PDS(PHOTO1)
         )
```

```
    insertCONTAINER ( 'photo1', MP )
```

Fig. 40 Insert and Remove Object Containers

More information about Object Containers can be found in [18] (see *8 Bibliography* on page 465)

3.8.8 Inserting and Removing additional Sheets

The Sheet insertion functionality of CIS allows installations to add use extra sheets such as those coming from an interposer or from a set of tab-sheets, without having to change the original application. It consists of the following keywords:

The **defineMEDIUMMAP** keyword allows an installation to specify the characteristics of the Medium Map to be used for the sheet being inserted. A Medium Map includes information such as Input and Output Bin, Simplex/Duplex, Landscape/Portrait, number of sheet copies, Medium Overlays, etc. This keyword may be used to define a new Medium Map or to modify the characteristics of an existing one. The new Medium Map is inserted as CIS processes the first external resource group (if any). In case of resource consolidation (see *3.7 How to Retrieve and Consolidate AFP Resources* on page 69), CIS will automatically rename a Medium Map with the same name starting with the second external resource group to be processed (the rename is done after applying the parameters of the DEFINEMEDIUMMAP).

In case of tab-printing in connection with Océ VarioPrint printers, the INBIN parameter of one or more defineMEDIUMMAP keywords can be used to specify the virtual bin ID of each of the tab sheets to be inserted. The UP3ImediaSetCount and UP3ImediaOrderedSetPiece sub-parameters of the MEDIAATTRIBUTES parameter of the defineMEDIUMMAP keyword can be used to select the tab-sheets based on their media attributes as opposed to using predefined bin numbers. Please check the documentation of your printer for information on which method of tab-printing selection is supported.

The **insertSHEET** keyword specifies the level and the Medium Map controlling the insert operation. Sheets may be inserted in Document, Page Group or Sheet entities. Where inside the entity and on which entities the insert sheet operation will be applied is defined using the DEFINELEVEL keyword. The characteristics of the sheet and any constant data it should contain is determined by the Medium Map used. The sheets are directly inserted in the entity. A page-group envelope, grouping all of the pages for the sheet, is **not** generated.

Inserted sheets may also contain variable data (optional). In this case, the inserted sheet must be defined as a target entity using the DEFINELEVEL keyword, and one or more data enrichment keywords (such as INSERTBCOCA, INSERTPTOCA, INSERTBOX,

INSERTCIRCLE, INSERTCONTAINER, INSERTCURVE, INSERTELLIPSE, or INSERTLINE) may be used to place variable data in the page(s) contained in the new sheet. The INSERTFINISHINGOP and INSERTINDEX keywords may also be applied to a sheet inserted by CIS.

The **removeMEDIUMMAP** keyword can be used to remove existing Medium Maps. This is done using the name specification. It can be used, in combination with DEFINEMEDIUMMAP, to replace existing Medium Maps.

The **removeSHEET** keyword can be used to remove sheets. Only sheets previously inserted using the INSERTSHEET keyword can be removed.

```
  defineLEVEL ( COVER, DOCUMENT, BEGIN, ALL )
defineMEDIUMMAP ( 'MMAP1', DUPLEX(NO) INBIN(20) PRESENTATION(PORTRAIT) )
  insertSHEET ( 'coverLetter', COVER, 'MMAP1' )

  defineLEVEL ( COVERLETTER, SHEET, FRONT, 'coverLetter' )
  definePTOCA ( 'COVERTEXT', , 50, 20, 0, , X0GT10, IBM_0500 )
  insertPTOCA ( 'COVERTEXT', COVERLETTER, 'This is a cover sheet' )
```

Fig. 41 Insert and Remove EXTRA sheets

Sheet insertion is done during the normal data enrichment process (see *Fig. 4 CIS internal workflow* on page 32). The content of all Input-file-related CIS processing variable is undefined for an inserted sheet (see *6.2 Appendix B: CIS processing variables* on page 331 for more information).

3.8.9 Inserting and Removing Finishing Operation triplets (UP³I)

The UP³I interface (Universal Printer-, Pre- and Post-processing Interface) is an "intelligent", open standard defined by the UP³I Core Group in 2000. Today, an operator has to enter job setup information manually on each system component separately – and each of these components (pre- or post- processors) could have a different design of user interface. Via the UP³I Manager, the communications interface between the system components, UP³I will make it possible to use central, unified control facilities for entering the job setup information, enabling remote control of every component in the printing process chain – from the printer through to the trimmer and binding equipment. The unified look & feel and the intuitive handling of all devices will minimize the cost of training operators and reduced make-ready time. More information about the UP³I Interface can be found in http://www.up3i.org and in [18] (see *8 Bibliography* on page 465).

CIS may be used to add finishing operations such as cut, staple, stack, fold or punch to an existing print file, without having to change the original application. The support provided by CIS is based on the UP³I Interface V1.03, and includes:

- Insert a single Finishing Operation for a sheet, mail piece or document.
- Insert nested Finishing Operations for mail pieces and documents.

CIS takes care of all the changes required to the Medium Map(s) associated to the entity being enriched, including the generation and insertion of the UP³I Finishing Operation triplet 0x8e. In case of nested operations, CIS generates the Begin and Continuation Medium Maps as required and modifies the Medium Map invocations (IMM structured fields) in the print data stream to match the new names. In summary, CIS simplifies the implementation of applications using UP³I Finishing Operations.

The **defineFINISHINGOP** keyword is used to define the type (e.g. fold, staple, cut, trim, rotate, punch, bind, etc.) and characteristics (e.g. fold signature, number of stitches, cross/separation cut, etc.) of the UP³I Finishing Operation to be applied. It also assigns a name to the operation.

The **insertFINISHINGOP** keyword identifies the level (e.g. sheet, mail piece or document) at which the operation is to be applied. In combination with the **insertSHEET** keyword, it allows an installation to add cover letters and to include them in the finished output (see *3.8.8 Inserting and Removing additional Sheets* on page 96 for more information). The USEREXIT parameter allows an installation to dynamically modify the parameters (type, position, etc.) of the finishing operation to be inserted (see *5.12 UP³I Finishing Operation-insertion exit* on page 314 for more information). Specifying USEREXIT causes the CIS normalization process to store all the input data into the internal work file before the conversion process is started.

The **removeFINISHINGOP** keyword can be used to remove previous finishing operations present in the input print file. The use of the insertFINISHINGOP keyword causes CIS to automatically remove any finishing triplet (0x85 or 0x8e) which may exist in the input print file.

The Finishing Operations defined in UP³I are based on the Printer Finishing MIB. The following definitions were copied from the Printer Finishing MIB RFC (http://www.ietf.org/rfc/rfc3806.txt):

BANDING:	Band Wrap:	Bundling a finished stack to prepare for shipment. Also known as Strap Wrap.
BINDING:	Adhesive Binding:	A method of attaching sheets together to form a book or booklet using glue or adhesive. Some adhesive binding methods apply the glue to sheets individually, before merging them together for form a book, but most methods involve the application of adhesive to an entire book of sheets.
	Comb Binding:	A method of binding in which a series

		of small rectangular holes is placed along the bind edge of the sheets. The sheets are then held together using a tube shaped plastic binding strip with comb like fingers that fit through the holes in the sheets.
	Spiral Binding:	Sometimes referred to as wire binding, this binding method is a mechanical bind in which the individual leaves are held together by a wire or plastic spiral that is fed through small holes in the paper binding edge.
	Padding:	Applying a non-penetrating adhesive to the edge of a stack of sheets such that the sheets can be easily peeled off one at a time. Frequently used for forms.
	Velo Binding:	A bind formed by punching holes into the edge of the sheets, placing a two piece plastic strip (one side formed with plastic pins that pass through the holes) along the edge and then staking the two pieces together.
	Perfect Binding:	A method of binding in which all pages are cut and roughed up at the back or binding edge and held together by an adhesive.
	Tape Binding:	The act of placing tape over the bind edge of a set. Sometimes contains adhesive to provide a functional bind to the set, and sometimes done for decorative purposes on a set that has been edge stapled.
CUTTING:	Tab Cutting:	The act of cutting the edge of a sheet to form an index tab, thereby allowing quick identification and access. The external tabs are sequentially placed along the book edge for visibility and ease of grasping.
	Perforating:	The act of cutting a series of very small, closely spaced holes or slots into a sheet to allow for ease of separation of a portion of the sheet. Sometimes

		also used to ease bending/hinging of heavy weight papers.
	Scoring:	A means of applying small linear grooves or impressions along a sheet to allow easy folding. Often used on heavy weight sheets and book covers.
FOLDING:	Z Fold:	A fold in which two folds are placed in the sheet in opposite directions. The first fold is located at 25% of the sheet length, and the second is located at 50% of the sheet length (e.g., the center of the sheet). Z Folding is often used on 11x17 inch or A3 size sheets, when they are included in sets containing 8.5x11 inch or A4 size sheets.
	Half Fold:	To fold a sheet in half so that one of the resulting dimensions are exactly half the original sheet. Often used for signatures or booklets.
	Letter Fold:	Folding a sheet roughly in thirds. Usually performed on 8.5x11 inch or A4 size sheets for insertion into an envelope.
	Signature:	The process by which images are placed on a large sheet of paper in correct panel areas and in the proper orientation such that when the sheet is folded it will produce a booklet with each page in the proper order and orientation.
ROTATING:	Sheet Rotator:	A device that rotates each sheet as received from the Media Path to the proper orientation for the finisher processing.
SLITTING:	Slitting:	The action of cutting apart a large sheet to form smaller sheets. Usually done using a sharp circular roll system.
STITCHING/ STAPLING:	Staple:	The process of binding a set of sheets together using a 'U' shaped piece of metal wire that is punched through the

set. The ends of the metal staple are then bent over, or 'clinched' to hold the staple in place. Technically the term 'stapler' refers to devices that use pre-cut metal staples, but the term is also commonly used to refer to devices that use wire spools and then cut/form the staple. (see the definition of Stitch)

	Stitch:	The process of binding a set of sheets together using a 'U' shaped piece of metal wire that is punched through the set. The wire used to form the staple is cut and formed into a 'U' shape in the stitcher head, and the staple 'leg' length is often varied depending on the number of sheets to be bound together. The ends of the metal staple are bent over, or 'clinched' to hold the staple in place. Stitching can also refer to the process of sewing the edges of the signatures of a book together.
	Saddle Stitch:	The process of stapling a set along its center line as part of a booklet making process. Usually 2 or 3 staples are used.
	Dual Stapling:	The process of placing 2 staples along the bind edge of a set. The staples are typically located at 25% and 75% of the length of the bind edge. Although dual stapling is often performed on the long edge of a set, legal documents are frequently dual stapled along the top, or short edge of the set.
	Triple Stapling:	Same as above, but using 3 staples along the bind edge, and usually applies to the long edge only.
TRIMMING:	Trim:	To cut the edges of a sheet or set of sheets.
	Face Trim:	To cut the edges of a set of sheets on a booklet of sheets that have been folded to eliminate the "creep" or edge

		shingling that results from the folding process.
	Gutter Trim:	To cut a larger sheet into smaller sheets eliminating the gutter between adjacent images. This operation requires a minimum of two cuts for each gutter.
WRAPPING	Shrink Wrap:	A wrap of thin plastic which when heated will shrink and wrap tightly around the stack thus preparing it for shipment.

More information about the operations and parameters available using the UP³I (including triplet 0x8E and other AFP/IPDS extensions for UP³I) can be found in [18] (see *8 Bibliography* on page 465).

Example:

Using the application shown in *Fig. 18 Legacy application* on page 60, an installation wants:

- Sort the print file by addressee.
- Add a Corner-Staple finishing operation so that all sheets for each account are stapled together.
- Consolidate all statements belonging to the same addressee in one mail piece.
- Add a cover letter and a flyer to each mail piece in the file. The cover letter includes a coupon which requires a perforation cut.
- The output pile is to be shifted left/right between addressees.
- The complete output pile is to be foil wrapped.

Fig. 42 shows the control file required for this example. Fig. 43 shows the results.

```
 FORMDEF ( F1CIS02 )
 PAGEDEF ( P1CIS02 )
   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 )

       SORT ( ASC, 'PLZ', 'STREET', 'NAME', 'ACCOUNT' )
  MPCONSOLIDATION ( NESTED, 'NAME', 'STREET', 'PLZ' )

/********************* JOB LEVEL OPERATIONS ***************************/

    defineLEVEL ( job_Level, DOCUMENT, BEGIN, ALL )
 defineFINISHINGOP ( 'jobFO', , SHRINK_WRAP, YES, TOP, , , )
 insertFINISHINGOP ( 'jobFO', job_Level )

/********************* CUSTOMER LEVEL OPERATIONS ***************************/

    defineLEVEL ( customer_Level_ODD, GROUPLEVEL1, BEGIN, ODD )
    defineLEVEL ( customer_Level_EVEN, GROUPLEVEL1, BEGIN, EVEN )
 defineFINISHINGOP ( 'customerFOodd' , , OFFSET, LEFT, LEFT, , , )
 defineFINISHINGOP ( 'customerFOeven', , OFFSET, RIGHT, RIGHT, , , )
 insertFINISHINGOP ( 'customerFOodd', customer_Level_ODD )
 insertFINISHINGOP ( 'customerFOeven', customer_Level_EVEN )

  defineMEDIUMMAP ( 'MMAP_001', DUPLEX ( NO)
                 PRESENTATION (PORTRAIT)
                 MEDIANAME ( 'MM1' )
                 MEDIAATTRIBUTES ( ,SATIN;SATIN,
                          ,, ,,,, ,,,, ,,,, )
         )
  defineMEDIUMMAP ( 'MMAP_002', DUPLEX ( NO)
                 CONSTANT ( FRONT )
                 PRESENTATION (PORTRAIT)
                 MEDIANAME ( 'MM2' )
                 MEDIAATTRIBUTES ( 'insert_Sheet_Form_a32454'
                            ,,,, ,,,, ,,,, ,,,, )
         )
    defineLEVEL ( customer_Level_ODD_1, GROUPLEVEL1, BEFORE, ODD )
    defineLEVEL ( customer_Level_EVEN_1, GROUPLEVEL1, BEFORE, EVEN )

  defineCONTAINER ( 'coverLetterContainer', MM, 100, 50, 20, 20, 0x60, 0, 0, 0,
           none, TIFF, SOFT, HLQ.IMAGES(LOGO1) )
  insertCONTAINER ( 'coverLetterContainer', job_Level )

    insertSHEET ( 'coverLetter', customer_Level_ODD_1, 'MMAP_001' )
    insertSHEET ( 'coverLetterSheet', customer_Level_EVEN, 'MMAP_001' )

    defineLEVEL ( coverLetter_Level, SHEET, AFTER, 'coverLetterSheet' )
    insertSHEET ( 'coverLetterAppendix', coverLetter_Level, 'MMAP_002' )

/*********************** ACCOUNT LEVEL OPERATIONS ***************************/

    defineLEVEL ( account_Level, GROUPLEVEL2, ALL, ALL )
 defineFINISHINGOP ( 'accountFO', , STAPLE, CORNER, TOP_LEFT, , , )
 insertFINISHINGOP ( 'accountFO', account_Level )
```

Fig. 42 Insert UP³I Finishing: sample keywords

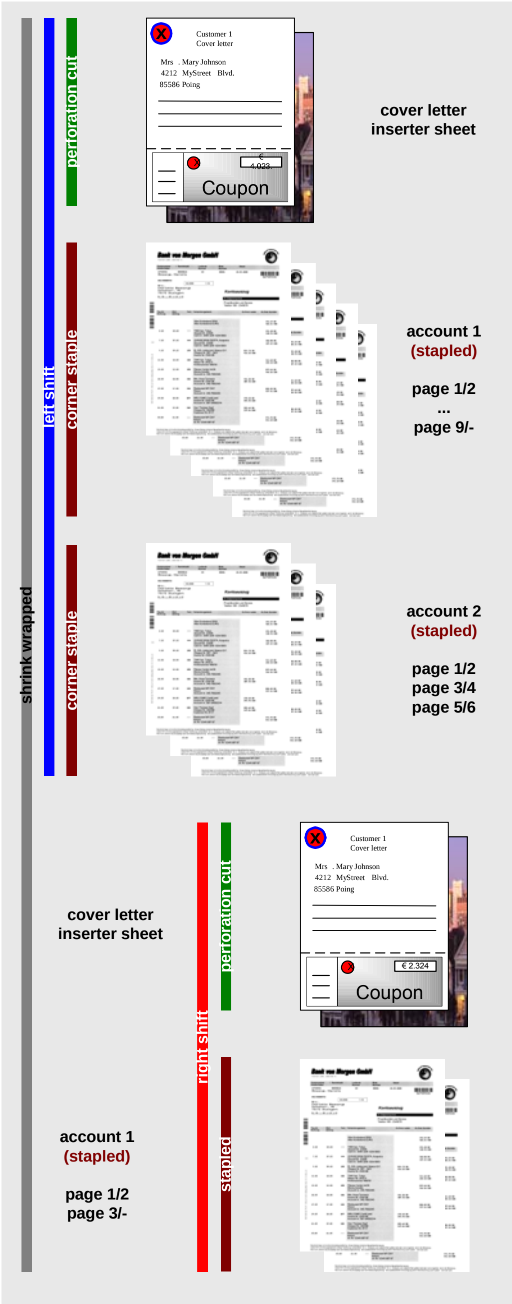


Fig. 43 Insert UP³I Finishing: sample results

3.8.10 Inserting one or more Medium Overlay(s)

CIS also supports the insertion of additional Medium Overlays. This feature is especially useful when used in combination with data selection, when testing applications or in case of reprints. It allows installations to add information that helps to identify the printed material.

Medium Overlay insertion is done in the FormDef resource. The print file itself is not modified. For this reason, inserting a Medium Overlay requires that the resource packaging process of CIS is active, at least for Form definitions.

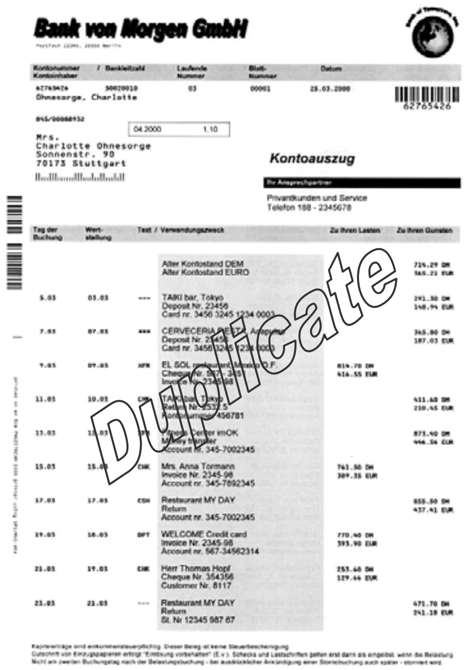


Fig. 44 Medium Overlay: sample output

Example:

An installation wants to extend the example shown in Fig. 39 so that the inquiries and possible reprints always indicate that the printed form is not the original. Fig. 45 shows the keywords required to add a medium overlay. Fig. 44 shows the resulting output.

```
OVERLAYF ( O1COPY )      /* Add overlay for front sheet side */
OVERLAYB ( O1COPY )      /* Add overlay for back sheet side */
```

Fig. 45 Insert Medium Overlays

3.8.11 Overriding other FormDef / PageDef parameters

CIS can be used to override other FormDef / PageDef parameters when specified together with RESTYPE(ALL) or RESTYPE(FDEF). The CHARS keyword is independent of the RESTYPE keyword and, when specified, always overrides the PageDef font list. The same applies to the override of Internal Medium Maps which is always done, independent of the RESTYPE keyword. In case of FORMDEF consolidation, the overrides are applied to the consolidated FORMDEF.

This option is similar to the support provided in SPS where FormDef parameters may be overridden at print time using a set of new values specified in a control file. CIS merges this keywords into the FormDef so that the control file itself does not need to be archived. The FormDef written in the Resource Object file will include all the new options.

The FormDef override keywords (e.g. INBIN, DUPLEX, etc.) do not influence the DEFINEMEDIUMMAP keyword.

The CIS keywords that can be used to override FormDef / PageDef parameters are:

CHARS Overrides any fonts specified in the PAGEDEF or in the input print data stream. It works independent of the RESTYPE keyword and, when specified, always overrides the PageDef font list.

COLORFIDELITY Overrides the Presentation Fidelity Control information available in the Document Environment Group of the FORMDEF.

DEFINEFINISHINGOP Overrides all Media Finishing Control structured fields available in the Document Environment Group and in the Medium Maps of the FORMDEF or as Internal Medium Map(s).

DEFINEMEDIUMMAP Inserts or Overrides selected Medium Maps in the FORMDEF. May be used to define the layout to be applied for printing a sheet inserted during data enrichment (e.g. tab printing).

DUPLEX Overrides the Duplex Control information in the MMC structured field of all the Medium Map(s) available in the FORMDEF or as Internal Medium Map(s).

FINISHINGFIDELITY Overrides the Presentation Fidelity Control information available in the Document Environment Group of the FORMDEF.

FONTFIDELITY Overrides the Presentation Fidelity Control information available in the Document Environment Group of the FORMDEF.

IMAGEOUT May be used to force CIS to convert all IM images in a print file (and resources) into IOCA images.

INBIN Overrides the Input Media selection in the MMC structured field of all the Medium Map(s) available in the FORMDEF or as Internal Medium Map(s).

LUPUB Causes CIS to convert the logical-unit-per-unit-base specifications in the AFP structured fields being generated to a particular value. All positions and dimensions in the print file are converted.

MEDIAFIDELITY Overrides the Presentation Fidelity Control information available in the Document Environment Group of the FORMDEF.

OFFSETX/Y B/F Overrides the XmOset and the YmOset fields in the Page Position (PGP) structured field of all the Medium Map(s) available in the FORMDEF: Old PGP-1 structured fields are converted to PGP-2.

OUTBIN Overrides the Output Media selection in all the Medium Map(s) available in the FORMDEF or as Internal Medium Map(s).

OVERLAYF
OVERLAYB See *3.8.10* Inserting one or more Medium Overlay(s) on page 106.

PRESENTATION Overrides any presentation specification done in the FORMDEF.

REMOVEMEDIUMMAP Removes selected Medium Maps of the FORMDEF.

SETUPVID Overrides the Set-up Verification ID specification in all the Medium Map(s) available in the FORMDEF or as Internal Medium Map(s).

TONERFIDELITY Overrides the Presentation Fidelity Control information available in the Document Environment Group of the FORMDEF.

X2UP Overrides any X2UP specification done in the FORMDEF.

The following CIS keywords may also modify the FormDef by defining additional parameters (Medium maps). They do this in an indirect way, during the processing of the functionality associated to the keyword.

RESCONSOLIDATION May cause Medium Maps available in the print file to be merged into the job's FORMDEF.

SORT May result in Medium Maps being added to the FORMDEF in order to be able to flip the sheets printed.

3.9 How to Generate the AFP Index Object file

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 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 55 under *How to convert, normalize and index* , 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. 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 keyword.

<u>Example 1</u>:

Using the application shown in *Fig. 18 Legacy application* on page 60 and the indexing keywords shown in *Fig. 16 Legacy application: Indexing parameters*, the following CIS Index Object file is created:

```
0000 00 D3A8A7 BDI Begin document index
             coded_graph_charset_global_id GCSGID = <FFFF> , CCSID = <01F4>
             fully_qualified_name      FQNType = 'Replace first GID name' ,
                             FQName = <CISIDX>
             fully_qualified_name      FQNType = 'Begin Resource Group Reference' ,
                             FQName = <CISRES>
0000 00 D3B2A7 IEL   index element
               fully_qualified_name      FQNType = 'Index Element GID' ,
                              FQName = <GROUP00000001>
               fully_qualified_name      FQNType = 'Begin Page Group Reference' ,
                              FQName = <GROUP00000001>
               direct_byte_offsset_triplet  DirByOff = <0000002D>
               object_SF_offset_triplet    SFOff = <00000001>
               object_byte_extent_triplet   ByteExt = <0000B6A6>
               object_SF_extent_triplet    SFExt = <000000BB>
0000 00 D3A090 TLE   tag logical element
               fully_qualified_name      FQNType = 'Attribute GID' ,
                              FQName = <ACCOUNT>
               attribute_value_triplet    AttVal = <F8F5F4F4F0F7F1F9>
0000 00 D3A090 TLE   tag logical element
               fully_qualified_name      FQNType = 'Attribute GID' ,
                              FQName = <NAME>
               attribute_value_triplet    AttVal = <C8859394A4A340D481A2A2>
0000 00 D3A090 TLE   tag logical element
               fully_qualified_name      FQNType = 'Attribute GID' ,
                              FQName = <STREET>
               Attribute_value_triplet    AttVal = <E68985A2859586859384A2A3994B40F1F5>
0000 00 D3A090 TLE   tag logical element
               fully_qualified_name      FQNType = 'Attribute GID' ,
```

```
                    FQName = <PLZ>
           attribute_value_triplet    AttVal = <F3F3F6F1F140C28985938586859384>
0000 00 D3B2A7 IEL   index element
           fully_qualified_name      FQNType = 'Index Element GID' ,
                    FQName = <PAGE00000001>
           fully_qualified_name      FQNType = 'Begin Medium Map Reference' ,
                    FQName = <CIS2F1 >
           direct_byte_offsset_triplet  DirByOff = <00000115>
           object_SF_offset_triplet    SFOff = <00000007>
           object_byte_extent_triplet   ByteExt = <000021FB>
           object_SF_extent_triplet    SFExt = <00000043>
0000 00 D3B2A7 IEL   index element
           fully_qualified_name      FQNType = 'Index Element GID' ,
                    FQName = <PAGE00000002>
           fully_qualified_name      FQNType = 'Begin Medium Map Reference' ,
                    FQName = <CIS2F1 >
           direct_byte_offsset_triplet  DirByOff = <000021FB>
           object_SF_offset_triplet    SFOff = <00000043>
           object_byte_extent_triplet   ByteExt = <00003F2E>
           object_SF_extent_triplet    SFExt = <00000053>
0000 00 D3B2A7 IEL   index element
           fully_qualified_name      FQNType = 'Index Element GID' ,
                    FQName = <PAGE00000003>
           fully_qualified_name      FQNType = 'Begin Medium Map Reference' ,
                    FQName = <CIS2F1 >
           direct_byte_offsset_triplet  DirByOff = <00003F2E>
           object_SF_offset_triplet    SFOff = <00000053>
           object_byte_extent_triplet   ByteExt = <00005E33>
           object_SF_extent_triplet    SFExt = <00000078>
0000 00 D3B2A7 IEL   index element
           fully_qualified_name      FQNType = 'Index Element GID' ,
                    FQName = <PAGE00000004>
           fully_qualified_name      FQNType = 'Begin Medium Map Reference' ,
                    FQName = <CIS2F1 >
           direct_byte_offsset_triplet  DirByOff = <00005E33>
           object_SF_offset_triplet    SFOff = <00000078>
           object_byte_extent_triplet   ByteExt = <00007B66>
           object_SF_extent_triplet    SFExt = <00000088>
0000 00 D3B2A7 IEL   index element
           fully_qualified_name      FQNType = 'Index Element GID' ,
                    FQName = <PAGE00000005>
           fully_qualified_name      FQNType = 'Begin Medium Map Reference' ,
                    FQName = <CIS2F1 >
           direct_byte_offsset_triplet  DirByOff = <00007B66>
           object_SF_offset_triplet    SFOff = <00000088>
           object_byte_extent_triplet   ByteExt = <00009A45>
           object_SF_extent_triplet    SFExt = <000000AC>
0000 00 D3B2A7 IEL   index element
           fully_qualified_name      FQNType = 'Index Element GID' ,
                    FQName = <PAGE00000006>
           fully_qualified_name      FQNType = 'Begin Medium Map Reference' ,
                    FQName = <CIS2F1 >
           direct_byte_offsset_triplet  DirByOff = <00009A45>
           object_SF_offset_triplet    SFOff = <000000AC>
           object_byte_extent_triplet   ByteExt = <0000B6B6>
           object_SF_extent_triplet    SFExt = <000000BB>
0000 00 D3B2A7 IEL   index element
           fully_qualified_name      FQNType = 'Index Element GID' ,
                    FQName = <GROUP00000002>
           fully_qualified_name      FQNType = 'Begin Page Group Reference' ,
                    FQName = <GROUP00000002>
           direct_byte_offsset_triplet  DirByOff = <0000B6D3>
           object_SF_offset_triplet    SFOff = <000000BC>
           object_byte_extent_triplet   ByteExt = <00006BD3>
           object_SF_extent_triplet    SFExt = <00000088>
0000 00 D3A090 TLE   tag logical element
           fully_qualified_name      FQNType = 'Attribute GID' ,
                    FQName = <ACCOUNT>
           attribute_value_triplet    AttVal = <F3F3F1F4F7F1F4F7>
0000 00 D3A090 TLE   tag logical element
           fully_qualified_name      FQNType = 'Attribute GID' , FQName = <NAME>
           attribute_value_triplet    AttVal = <D4969589928140E596878593>
0000 00 D3A090 TLE   tag logical element
           fully_qualified_name      FQNType = 'Attribute GID' ,
                    FQName = <STREET>
           attribute_value_triplet    AttVal = <E28388A68584859583888195A98540F1F7>
0000 00 D3A090 TLE   tag logical element
```

```
              fully_qualified_name      FQNType = 'Attribute GID' ,
                             FQName = <PLZ>
              attribute_value_triplet    AttVal = <F7F8F4F6F240D29695A2A38195A9>
0000 00 D3B2A7 IEL   index element
              fully_qualified_name      FQNType = 'Index Element GID' ,
                             FQName = <PAGE00000007>
              fully_qualified_name      FQNType = 'Begin Medium Map Reference' ,
                             FQName = <CIS2F1 >
              direct_byte_offsset_triplet  DirByOff = <0000B7BB>
              object_SF_offset_triplet    SFOff = <000000C2>
              object_byte_extent_triplet   ByteExt = <0000D8A1>
              object_SF_extent_triplet    SFExt = <000000FE>


  ...


0000 00 D3A9A7 EDI End document index <........>
            fully_qualified_name      FQNType = 'Replace first GID name' ,
                            FQName = <CISIDX>
```

Fig. 46 Index Object file (example 1)

Example 2:

Using the same keywords as in Example 1 but INDEXOBJ, which is set to GROUP instead of ALL causes CIS to generate the following Index Object file:

```
0000 00 D3A8A7 BDI Begin document index
            coded_graph_charset_global_id GCSGID = <FFFF> , CCSID = <01F4>
            fully_qualified_name      FQNType = 'Replace first GID name' ,
                            FQName = <CISIDX>
            fully_qualified_name      FQNType = 'Begin Resource Group Reference' ,
                            FQName = <CISRES>
0000 00 D3B2A7 IEL   index element
              fully_qualified_name      FQNType = 'Index Element GID' ,
                             FQName = <GROUP00000001>
              fully_qualified_name      FQNType = 'Begin Page Group Reference' ,
                             FQName = <GROUP00000001>
              direct_byte_offsset_triplet  DirByOff = <0000002D>
              object_SF_offset_triplet    SFOff = <00000001>
              object_byte_extent_triplet   ByteExt = <0000B6A6>
              object_SF_extent_triplet    SFExt = <000000BB>
0000 00 D3A090 TLE   tag logical element
              fully_qualified_name      FQNType = 'Attribute GID' ,
                             FQName = <ACCOUNT>
              attribute_value_triplet    AttVal = <F8F5F4F4F0F7F1F9>
0000 00 D3A090 TLE   tag logical element
              fully_qualified_name      FQNType = 'Attribute GID' ,
                             FQName = <NAME>
              attribute_value_triplet    AttVal = <C8859394A4A340D481A2A2>
0000 00 D3A090 TLE   tag logical element
              fully_qualified_name      FQNType = 'Attribute GID' ,
                             FQName = <STREET>
              attribute_value_triplet    AttVal = <E68985A2859586859384A2A3994B40F1F5>
0000 00 D3A090 TLE   tag logical element
              fully_qualified_name      FQNType = 'Attribute GID' ,
                             FQName = <PLZ>
              attribute_value_triplet    AttVal = <F3F3F6F1F140C28985938586859384>
0000 00 D3B2A7 IEL   index element
              fully_qualified_name      FQNType = 'Index Element GID' ,
                             FQName = <GROUP00000002>
```

```
               fully_qualified_name      FQNType = 'Begin Page Group Reference' ,
                              FQName = <GROUP00000002>
               direct_byte_offsset_triplet  DirByOff = <0000B6D3>
               object_SF_offset_triplet    SFOff = <000000BC>
               object_byte_extent_triplet   ByteExt = <00006BD3>
               object_SF_extent_triplet    SFExt = <00000088>
0000 00 D3A090 TLE   tag logical element
               fully_qualified_name      FQNType = 'Attribute GID' ,
                              FQName = <ACCOUNT>
               attribute_value_triplet    AttVal = <F3F3F1F4F7F1F4F7>
0000 00 D3A090 TLE   tag logical element
               fully_qualified_name      FQNType = 'Attribute GID' ,
                              FQName = <NAME>
               attribute_value_triplet    AttVal = <D4969589928140E596878593>
0000 00 D3A090 TLE   tag logical element
               fully_qualified_name      FQNType = 'Attribute GID' ,
                              FQName = <STREET>
               attribute_value_triplet    AttVal = <E28388A685848595583888195A98540F1F7>
0000 00 D3A090 TLE   tag logical element
               fully_qualified_name      FQNType = 'Attribute GID' ,
                              FQName = <PLZ>
               attribute_value_triplet    AttVal = <F7F8F4F6F240D29695A2A38195A9>
0000 00 D3B2A7 IEL   index element
               fully_qualified_name      FQNType = 'Index Element GID' ,
                              FQName = <GROUP00000003>
               fully_qualified_name      FQNType = 'Begin Page Group Reference' ,
                              FQName = <GROUP00000003>
               direct_byte_offsset_triplet  DirByOff = <000122A6>
               object_SF_offset_triplet    SFOff = <00000144>
               object_byte_extent_triplet   ByteExt = <00009999>
               object_SF_extent_triplet    SFExt = <000000AC>
0000 00 D3A090 TLE   tag logical element
               fully_qualified_name      FQNType = 'Attribute GID' ,
                              FQName = <ACCOUNT>
               attribute_value_triplet    AttVal = <F6F5F5F6F7F9F8F6>


  ...


0000 00 D3A9A7 EDI End document index <........>
             fully_qualified_name      FQNType = 'Replace first GID name' ,
                             FQName = <CISIDX>
```

Fig. 47 Index Object file (example 2)

3.10 How to Sort and Reorder output pages

- **<u>OVERVIEW</u>**

The CIS sort function reorders the pages or sheets contained in the print file in a new sequence.

Three different kinds of sorting methods are available:

1. **Contents sorting**

   Where documents (mail pieces, sheets) are regrouped based on their data contents (e.g. Postal code, Street name, Customer name, etc.). Suitable for printing and viewing applications.

2. **Layout sorting**

   Where pages are regrouped into a completely new layout (order), replacing the previous document and sheet contents with a new one. Suitable for printing applications.

3. **Collator**

   Where sheets are collated in the sequence required by the output stacking device (face-up or face-down, collated or uncollated stacker) without loosing their document, mail piece and sheet structure. Suitable for printing applications.

- **<u>SORTING PROCESS</u>**

The sorting process implemented in CIS is the same for all sorting types and all sorting schemes (see Fig. 4 on page 32). 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 in an internal work file.

b) The Index tags inserted or collected during the Conversion/Normalization process are kept in storage. Resources, in case RESTYPE other than NONE 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. Any empty pages inserted will not be eligible for data enrichment functions (e.g. insertPTOCA) nor will be counted in the Input File related variables (they are counted in the Output file related variables). Refer to *6.2 Appendix B: CIS processing variables* on page 331 for more information.

e) The Sorting algorithm is applied.

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

The sorting methods available under CIS are explained in this chapter.

3.10.1 Contents sorting

Contents sorting takes advantage of the indexing information obtained during the normalization and conversion process done by CIS. How this process works is described in this manual under *3.6.1 Converting and Indexing print files containing S/370 Line formatted 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 CIS to consolidate (merge) data from different applications into a single output file provided that the same indexing keys are used (see *3.11 How to Consolidate mail pieces* on page 135 for more information).

Sorting input files using their page group level indexes causes 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.

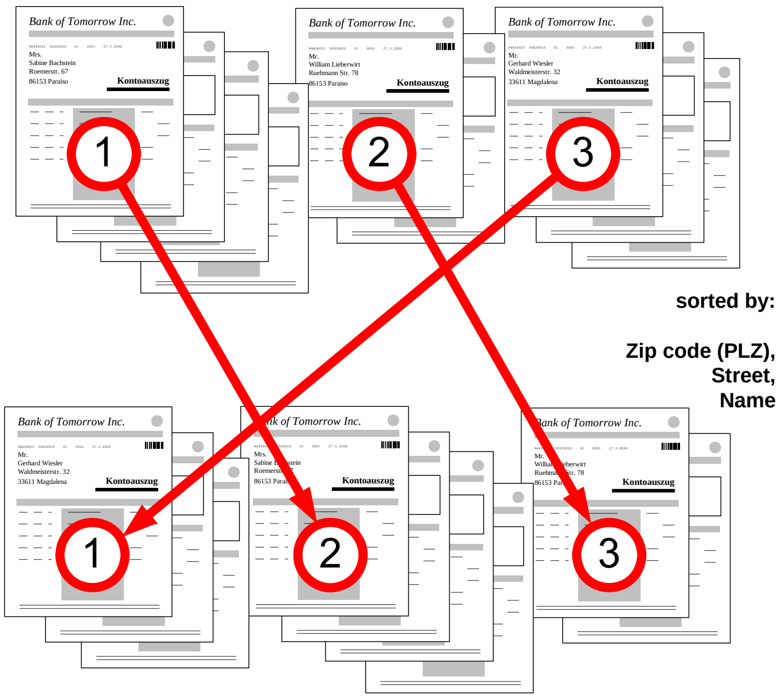
```
SORT ( ASC, ‘PLZ’, ‘STREET’, ‘NAME’, ‘ACCOUNT’ )
```

Fig. 48 Sort keywords

Contents sorting requires a consistent document structure. This structure is either already available in the document or is created by CIS during the Normalization process (see *3.6.1 Converting and Indexing print files containing S/370 Line formatted data* on page 56).

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 CIS for the contents sorting process. An AFP print file can also be indexed using CIS (see Normalizing and indexing print files containing AFP and MO:DCA-P data on page 65). In this case, any Document and Page-Group structured fields (BDT, EDT, BNG, ENG) will be removed and replaced with the structured fields from the indexing process. The structure of the resulting file will be consistent and suitable for the CIS sorting process.

```
1.    Sabine Bachstein      Roemerstr. 67         86153 Paraiso
2.    William Lieberwirt    Ruehmann Str. 78      86153 Paraiso
3.    Gerhard Wiesler       Waldmeisterstr. 32    33611 Magdalena
```




```
1.    Gerhard Wiesler       Waldmeisterstr. 32    33611 Magdalena
2.    Sabine Bachstein      Roemerstr. 67         86153 Paraiso
3.    William Lieberwirt    Ruehmann Str. 78      86153 Paraiso
```

Fig. 49 Contents sorting

Example:

> Using the application shown in *Fig. 18 Legacy application* on page 60, the indexing keywords shown in *Fig. 16 Legacy application: Indexing parameters*, and the sorting parameters described in *Fig. 48 Sort keywords* the following process is done by CIS:

Contents sorting may be done in an ascending or descending sequence. The sorting algorithm implemented in this version of 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, Unicode, etc.), national characters or special symbols. Data alignment within the field (left or right justified), leading and/or trailing characters (e.g. leading zeroes, trailing blanks) and decimal separation characters (e.g. comma or dot) are handled as part of the data and may influence the sort results.

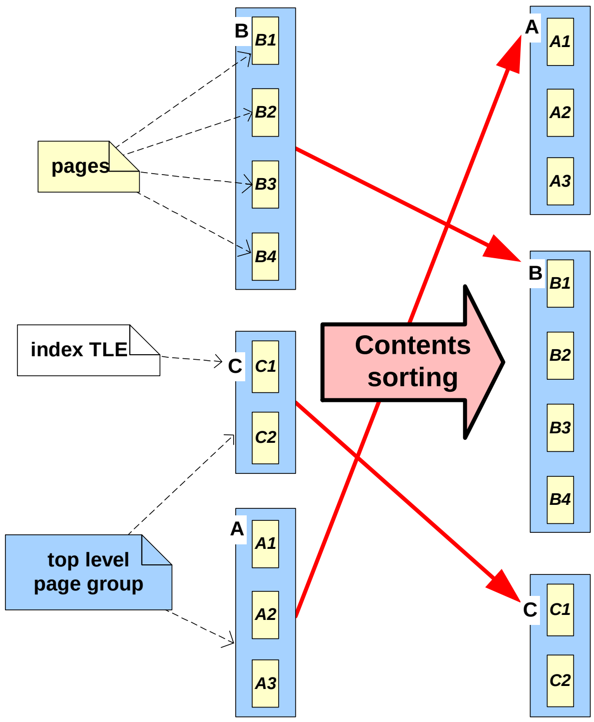


Fig. 50 Data structure with Contents sorting

Example:

Fig. 50 shows the AFP document structure generated by CIS for the example shown in *Fig. 18 Legacy application* and used during the sorting process shown in *Fig. 49 Contents sorting.*

CIS expects all the page groups in the file to include the attribute name(s) specified in the SORT keyword. In case this is not true, CIS issues a warning message for the affected page group(s), assumes an index value of binary zeroes for the missing attribute(s) and continues sorting.

It is not possible to use the Index Tag Elements inserted with the insertINDEX keyword to control the contents sorting of the same CIS run. Splitting the index insertion and sorting processes in two CIS runs will bypass this problem. Fig. 4 on page 32 shows the sequence used for the index insertion, sorting and extraction steps.

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 CIS versions.

1. The sorting algorithms (ASC, DES) implemented in this version of 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 (e.g. leading zeroes, trailing blanks) and decimal separation characters (e.g. comma or dot) are handled as part of the data and may influence the sort results.
2. A mail piece, as defined in the Glossary on page 445, 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[1]) is done before the sorting process. The page numbers inserted reflect the original page sequence.

[1] Océ Software Tool, Smart Layout Editor (SLE)

3.10.2 Layout sorting

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. The layout sorting schemes are:

- <u>FANFOLD LISTING DUPLEX (SCHEME_1)</u>

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 set-up for this report format.

The SCHEME_1 of Layout sorting available in CIS (also implemented by the Page Sort utility for PRISMAproduction/MVS V3.8) offers a solution for these long reports as it allows an installation to immediately realise the benefits of duplex printing, without using any post-processing machinery and without writing specific printing applications especially designed for duplex printers.

Fig. 51 illustrates the resulting output. Black numbers represent the front-side page, grey-numbers are used to represent the back-side pages.

<u>Requirements and considerations</u>:

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 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. An empty 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 CIS

5. Any attempt to control page/sheet grouping within the application (e.g. 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 <u>before</u> the sorting process. The page numbers inserted reflect the original page sequence.

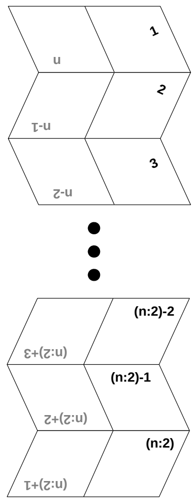


Fig. 51 Layout sorting: SCHEME_1

- <u>**FANFOLD LISTING 2-UP (SCHEME_2)**</u>

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

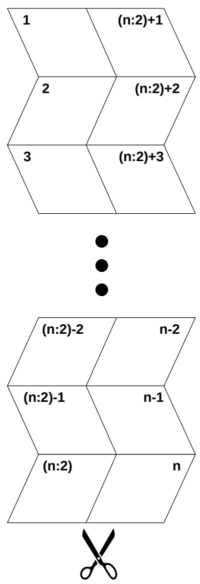


Fig. 52 Layout sorting: SCHEME_2

<u>Requirements and considerations</u>:

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 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. An empty 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 CIS

5. 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.

- **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.

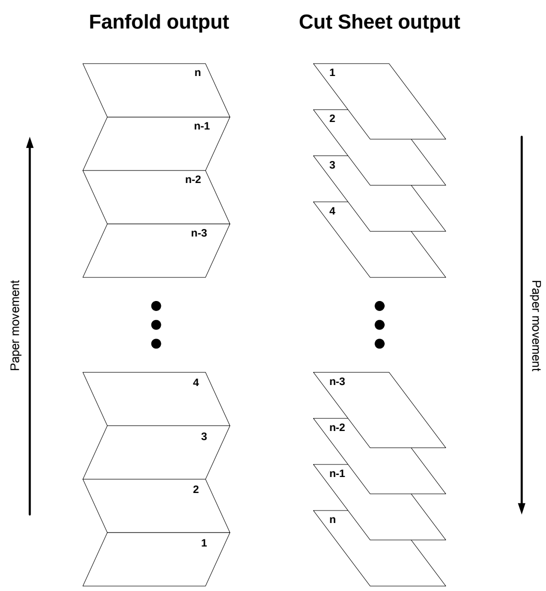


Fig. 53 Layout sorting: SCHEME_3

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 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 (e.g. using IMM to force an 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 page numbers inserted reflect the original page sequence.

- **4-UP BOOKLET SORTING (4UP_BOOKLET)**

4-up booklet sorting can be applied on A3 duplex printers (fanfold or cut-sheet) to get A5 booklets. In case of a fanfold printer, it is expected that the continuous A3 size forms are separated from each other and cut in the middle.

Any number of pages may be sorted. Enough trailing blank pages are generated to obtain a modulo 8 number of pages. The complete print file is sorted and printed as one booklet.

Fig. 54 shows a sample application containing 13 pages. The numbers enclosed in parenthesis represent the pages printed at the back side of the sheets. CIS is used to re-order the output pages in the sequence required during the print process. Pages 14, 15 and 16 are inserted by the 4_UP booklet sorting scheme at the end of the print file before the sorting process is done. The output is stacked face-down, with the cover page at the bottom.

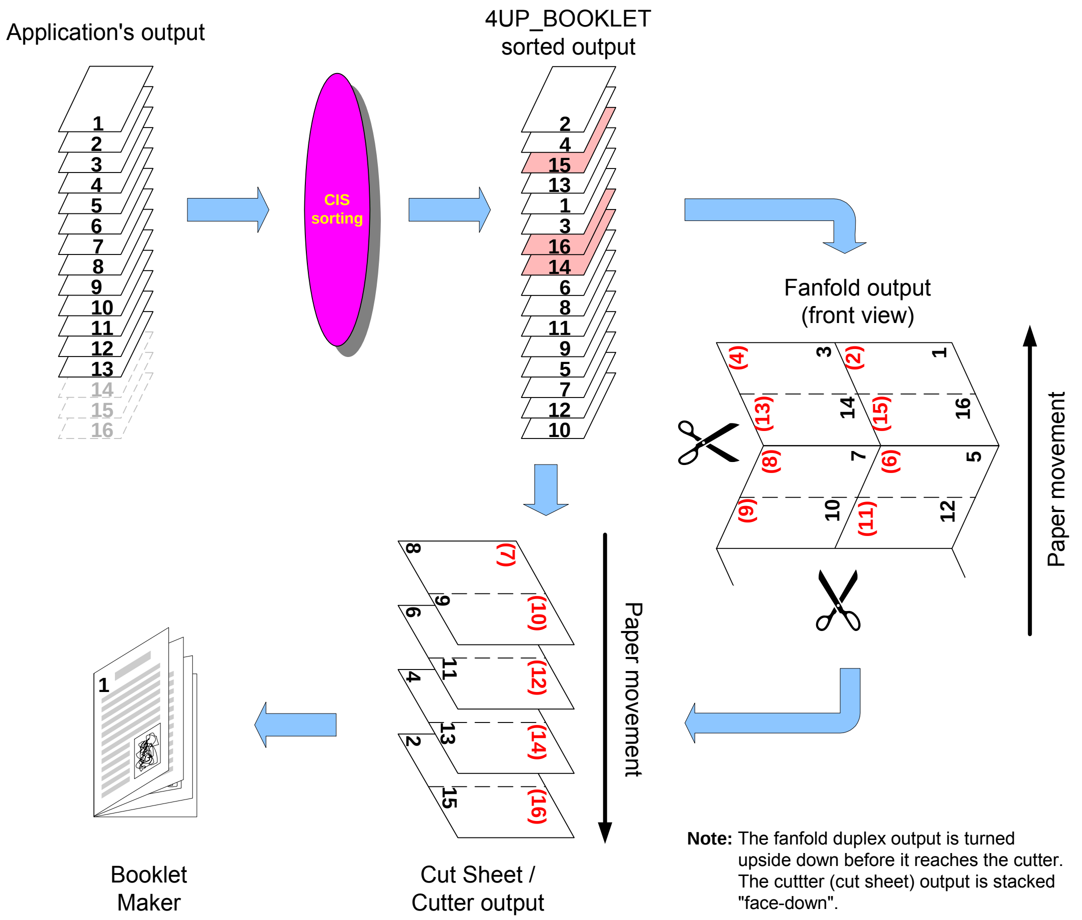


Fig. 54 Layout sorting: 4UP_BOOKLET

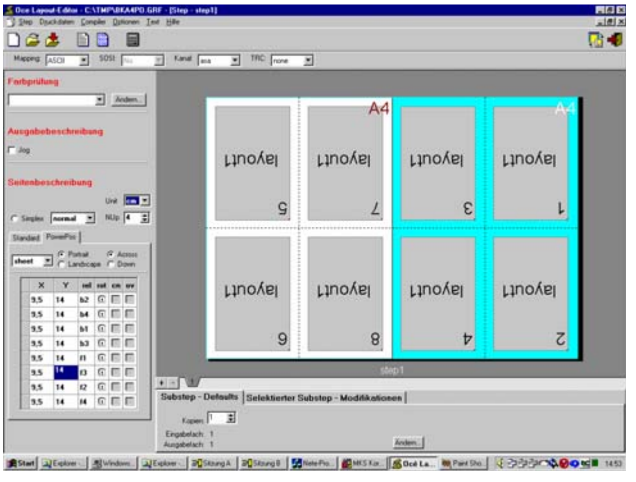
Requirements and considerations:

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 CIS versions.

1. The sorting scheme (and printing sequence) is: 2, 4, n-1, n-3, 1, 3, n, n-2, 6, 8, n-5, n-7, 5, 7, n-4, n-6, ... The number of pages is rounded up to the next value modulo 8 by inserting enough empty pages.

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:

SLE example	PPFA example




```
COPYGROUP 4UPBOOK
 PRESENT portrait DIRECTION across
 DUPLEX normal
 N_UP 4
   PLACE 2 BACK
     OFFSET 9.5 cm 14.0 cm ROTATION 180
   PLACE 4 BACK
     OFFSET 9.5 cm 14.0 cm ROTATION 180
   PLACE 1 BACK
     OFFSET 9.5 cm 14.0 cm ROTATION 180
   PLACE 3 BACK
     OFFSET 9.5 cm 14.0 cm ROTATION 180
   PLACE 1 FRONT
     OFFSET 9.5 cm 14.0 cm ROTATION 180
   PLACE 3 FRONT
     OFFSET 9.5 cm 14.0 cm ROTATION 180
   PLACE 2 FRONT
     OFFSET 9.5 cm 14.0 cm ROTATION 180
   PLACE 4 FRONT
     OFFSET 9.5 cm 14.0 cm ROTATION 180
 INVOKE sheet
```

Fig. 55 Layout sorting: FormDef requirements by 4UP_BOOKLET

5. Any attempt to control page/sheet grouping within the application (e.g. using IMM to force an 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 <u>before</u> the sorting process. The page numbers inserted reflect the original page sequence.

- **<u>TWO-UP BOOKLET SORTING (2UP_BOOKLET AND 2UP_MP_BOOKLET)</u>**

2-UP booklet sorting can be applied on A3 duplex printers (fanfold or cut-sheet) to get A4 booklets. In case of a fanfold printer, it is expected that the continuous A3 size forms are separated from each other and folded in the middle.

Any number of pages may be sorted. Enough trailing blank pages are generated to obtain a modulo 4 number of pages for each booklet to be printed. Two different types of 2UP booklet sorting are available in CIS:

- The complete print file is sorted so that it can be printed as one single booklet. Enough blank pages are inserted at the end of the print file. Fig. 56 shows a sample application containing 13 pages.

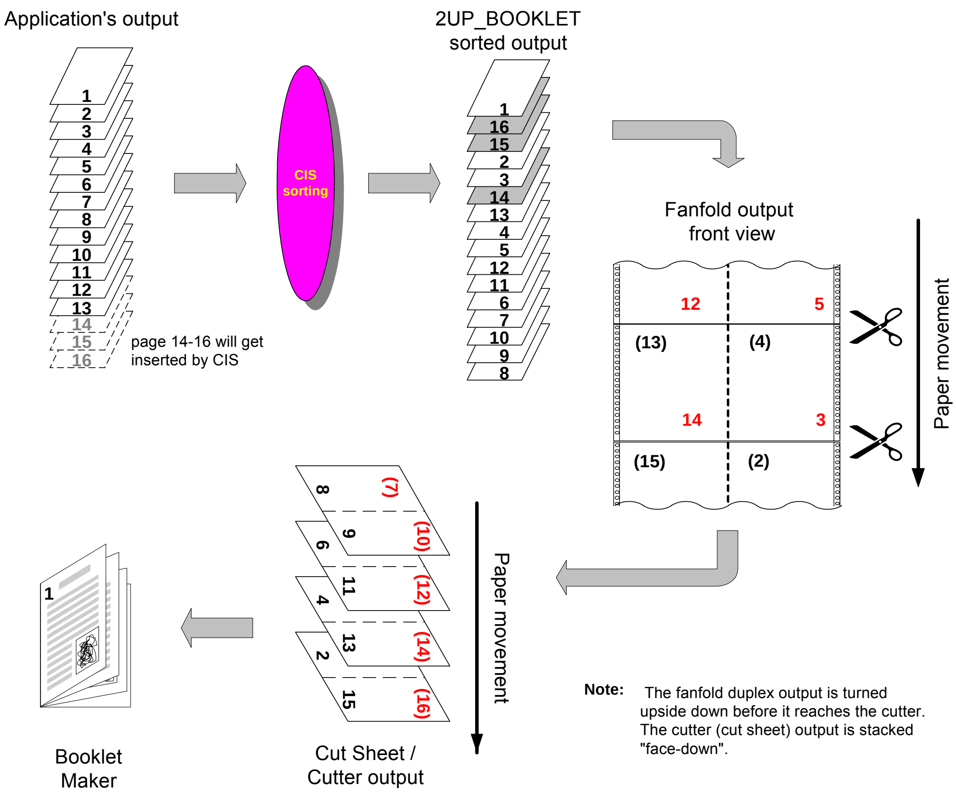


Fig. 56 Layout sorting: 2UP_BOOKLET

The numbers enclosed in parenthesis represent the pages printed at the back side of the sheets. CIS is used to re-order the output pages in the sequence required during the print process. Pages 14, 15 and 16 are inserted by the 2UP_booklet sorting scheme at the end of the print file before the sorting process is done. The output is stacked face-down, with the cover page at the bottom.

- Each mail piece within the print file is sorted so that the print file can be printed as a collection of booklets, where each booklet is a mail piece. Enough blank pages are inserted at the end of every mail piece in order to guarantee that only one mail piece per booklet is printed. As in the case of contents sorting, 2UP_MP_BOOKLET requires a consistent document structure where Document and Page Group structured fields (BDT, EDT, BNG and ENG) are used as delimiters. This document structure may be created by CIS during the Normalization process (see *3.6 How to convert, normalize and index data* on page 55).

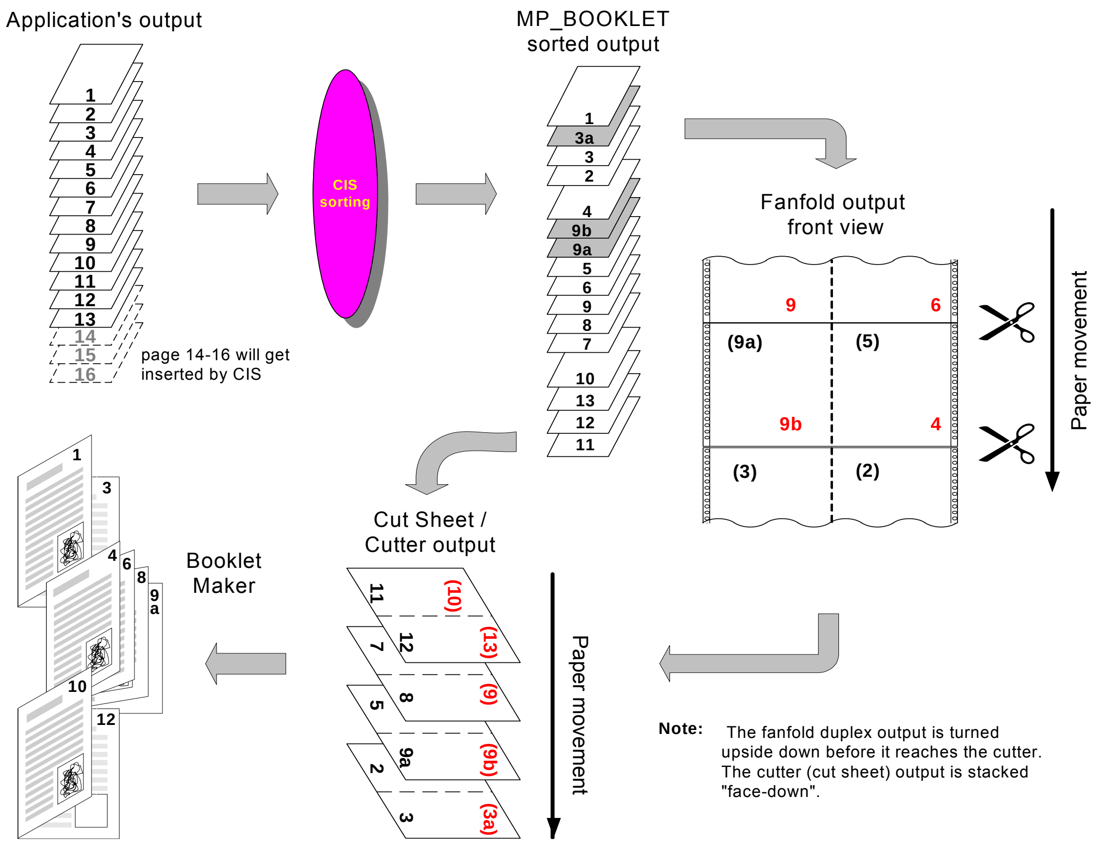


Fig. 57 Layout sorting: 2UP_MP_BOOKLET

0 shows a sample application containing 3 mail pieces of different size each, with a total of 13 pages. CIS is used to re-order the output pages in the sequence required for the print process. Blank pages are inserted by the 2UP_MP_BOOKLET sorting scheme at the end of every mail piece in order to obtain a modulo 4 number of pages per booklet. The pages are inserted as the sorting process is done. The output is stacked face-down, with the cover page at the bottom.

<u>Requirements and considerations:</u>

Following is a list of requirements and considerations that apply to the 2UP_BOOKLET and 2UP_MP_BOOKLET sorting schemes. Deviations from this usage model are possible but not recommended as they may stop working in future CIS versions.

1. The sorting scheme (and printing sequence) is: 1, 2, n, n-1, 3, 4, n-2, n-3, ... The number of pages is rounded up to the next value modulo 4 by inserting enough empty pages.

   In case of 2UP_BOOKLET, the sorting scheme is applied to the complete print file.

   In case of 2UP_MP_BOOKLET, the sorting scheme is applied to each mail piece individually, changing the sequence of the pages within each mail piece but not the sequence of the mail pieces within the file.

2. The 2UP_BOOKLET scheme does a "simplification" of the input print file, removing existing Page Groups. A single Page Group containing all pages is created as output. Index tags are also removed during the simplification process. A warning message is issued in this case.

   The existing Page Group structures are left unchanged in case of 2UP_MP_BOOKLET sorting.

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

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

5. In order to achieve the results expected with this sorting scheme, the Form Definition used should have the following characteristics:

SLE example

PPFA example

```
COPYGROUP 2UPBOOK
 PRESENT portrait DIRECTION across
 DUPLEX normal
 N_UP 2
   PLACE 1 FRONT
     OFFSET 14.0 cm 20.0 cm ROTATION 180
   PLACE 2 BACK
     OFFSET 14.0 cm 20.0 cm ROTATION 180
   PLACE 2 FRONT
     OFFSET 14.0 cm 20.0 cm ROTATION 180
   PLACE 1 BACK
     OFFSET 14.0 cm 20.0 cm ROTATION 180
 INVOKE sheet;
```

Fig. 58

Layout sorting: FormDef for 2UP_BOOKLET and 2UP_MP_BOOKLET

3.10.3 Collator

There are different kinds of stacking devices that can be connected to an AFP/IPDS printer. Depending on the way and sequence in which the output is stacked, stacking devices may be classified in:

- **Face-up or face-down stackers**. Face-up stackers collect the sheets with their front side (e.g. page 1) facing upwards. Face-down stackers collect sheets with their front side facing downwards.
- **Collated or uncollated stackers**. Collated stackers are those where the sheets are stacked in their natural sequence, so that the output stack can be stitched together (e.g. the back side of sheet 1 touches the front side of sheet 2). Uncollated stackers cause the front side of sheet 1 to touch the back side of sheet 2, modifying the natural file sequence.
- **Cut-sheet or fan-fold stackers**.Cut-sheet stackers are prepared to handle sets of cut media. Fan-fold stackers may only accept fan-folded output.

The collating options in CIS can be used to assemble the sheets in a file so that they can be printed on any output stacking device in the sequence intended by the original print file, without having to modify the application.

Collating a print file not only involves re-arranging the sheets sequence, but may require modifications in the FormDef (to flip the sheet sides) and/or the insertion of blank pages. All these steps are automatically done by CIS, allowing installations to print an application on any kind of stacking device with no conversion effort.

- <u>FLIP JOB FOR COLLATED STACKER</u>

The way a print file is stacked in a collated stacker is determined by the stacker device in use: face-down stackers (the most common case) cause output to be stacked face-down. Face-up stackers do it face-up.

The FLIP_FOR_COLLATED sorting scheme available in CIS allows installations to flip print files so that they are stacked with the cover page facing the opposite side as it would normally do: face-up in case of a face-down stacker, face-down in case of a face-up stacker.

0 illustrates how this sorting scheme works for a one-up, mixplex application. Black numbers represent the front-side page, grey-numbers (in parenthesis) are used to represent the back-side pages.

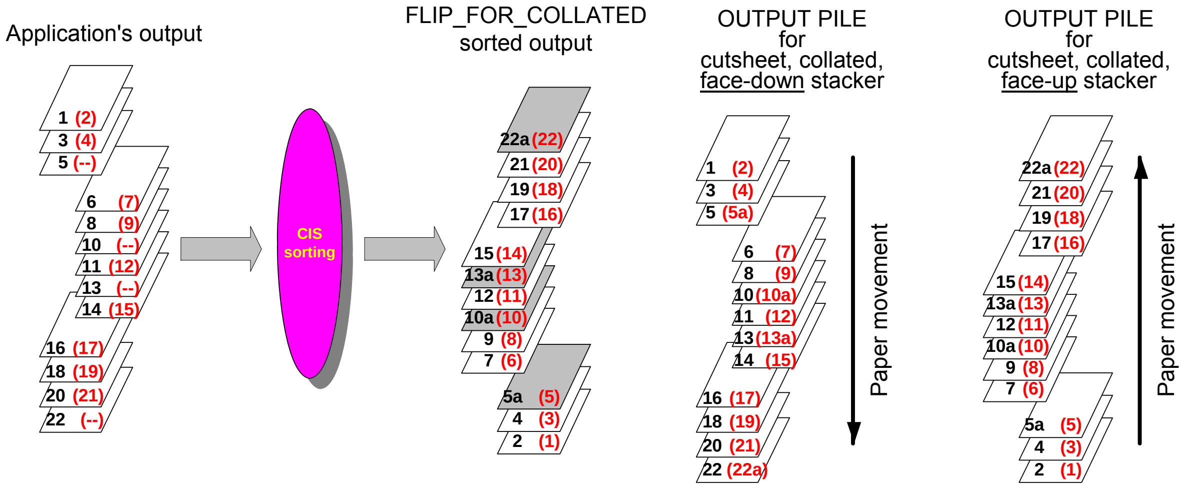


Fig. 59 Collator: FLIP_FOR_COLLATED

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 CIS versions.

1. The sorting scheme (and printing sequence) is: [n, n-1], [n-2, n-3], ... , [4, 3], [2, 1]. Simplex sheets are converted to duplex, NUP-1 sheets.
2. The sorting process is done on a sheet basis, so that the content of each sheet remains the same. The sheet positions within the output stack are changed. The contents of the front and back sides of every sheet are swapped.
3. Page groups (if used) must start at a sheet boundary.
4. A duplex and N-up capable printer must be used. It is not possible to flip jobs on a simplex printer.
5. A single, collated output stacker must be used. Using an uncollated output stacker will not produce the expected results.
6. The document, mail piece and sheet structure of the input print file are not affected. Index tags (if any) are moved together with their corresponding pages or page groups.
7. CIS automatically modifies the Medium Map(s) used by the print file in order to obtain the desired results. All Medium Map options such as N-up, multiple copies, constant

data and Finishing Operations are supported. The layout of the printed file is not changed. The Medium Maps created by CIS use N-up functionality.

8. 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.

- **FOR UNCOLLATED STACKER**

The FOR_UNCOLLATED sorting scheme available in CIS allows installations to use an uncollated stacking device (such as a roll device which will not be rewinded afterwards) and to obtain an output pile as if the print job was printed on a collated stacker. To do this, CIS re-arranges the sequence of all sheets in the print file.

0 illustrates how this sorting scheme works for a one-up, mixplex application. Black numbers represent the front-side page, grey-numbers (in parenthesis) are used to represent the back-side pages.

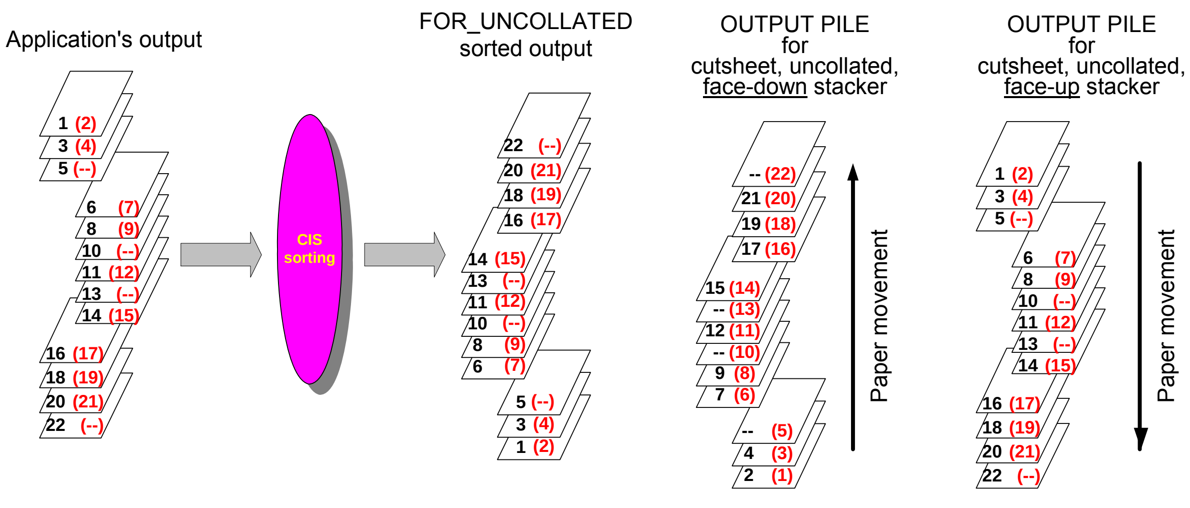


Fig. 60 Collator: FOR_UNCOLLATED

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 CIS versions.

1. The sorting scheme (and printing sequence) is: [n-1, n], [n-3, n-2], ..., [3, 4], [1, 2]. Simplex sheets are left as they are.
2. The sorting process is done on a sheet basis, so that the content of each sheet remains the same. The sheet positions within the output stack are changed. The contents of the front and back sides of every sheet are left unchanged.
3. Page groups (if used) must start at a sheet boundary.
4. A simplex printer may be used in case the whole input file is simplex.
5. A single, uncollated output stacker must be used. Using a collated output stacker will not produce the expected results.
6. The document, mail piece and sheet structure of the input print file are not affected. Index tags (if any) are moved together with their corresponding pages or page groups.
7. CIS does not need to modify the medium map(s). All Medium Map options such as N-up, multiple copies, constant data and Finishing Operations are supported. The format of the printed file is the same as if no sorting would have been done.
8. 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.

- **FLIP JOB FOR UNCOLLATED STACKER**

The FLIP_FOR_UNCOLLATED sorting scheme available in CIS combines the characteristics of the FLIP_FOR COLLATED and FOR_UNCOLLATED schemes, allowing installations to use an uncollated stacking device and to flip the jobs so that they are stacked with the cover page facing the opposite side as it would normally do: face-up in case of a face-down stacker, face-down in case of a face-up stacker.

0 illustrates how this sorting scheme works for a one-up, mixplex application. Black numbers represent the front-side page, grey-numbers (in parenthesis) 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 CIS versions.

1. The sorting scheme (and printing sequence) is: [2, 1], [4, 3], ..., [n-2, n-3], [n, n-1]. Simplex sheets are converted to duplex, NUP-1 sheets.
2. The sorting process is done on a sheet basis, so that the content of each sheet remains the same. The sheet positions within the output stack are changed. The contents of the front and back sides of every sheet are swapped.

3. Page groups (if used) must start at a sheet boundary.
4. A duplex, N-up capable printer must be used. It is not possible to flip jobs on a simplex printer.
5. A single, uncollated output stacker must be used. Using a collated output stacker will not produce the expected results.
6. The document, mail piece and sheet structure of the input print file are not affected. Index tags (if any) are moved together with their corresponding pages or page groups.
7. CIS automatically modifies the Medium Map(s) used by the print file in order to obtain the desired results. All Medium Map options such as N-up, multiple copies, constant data and Finishing Operations are supported. The layout of the printed file is not changed. The Medium Maps created by CIS use N-up functionality.
8. The Page Numbering feature in the Page Definition (generated using SLE) is done <u>before</u> the sorting process. The page numbers inserted reflect the original page sequence.

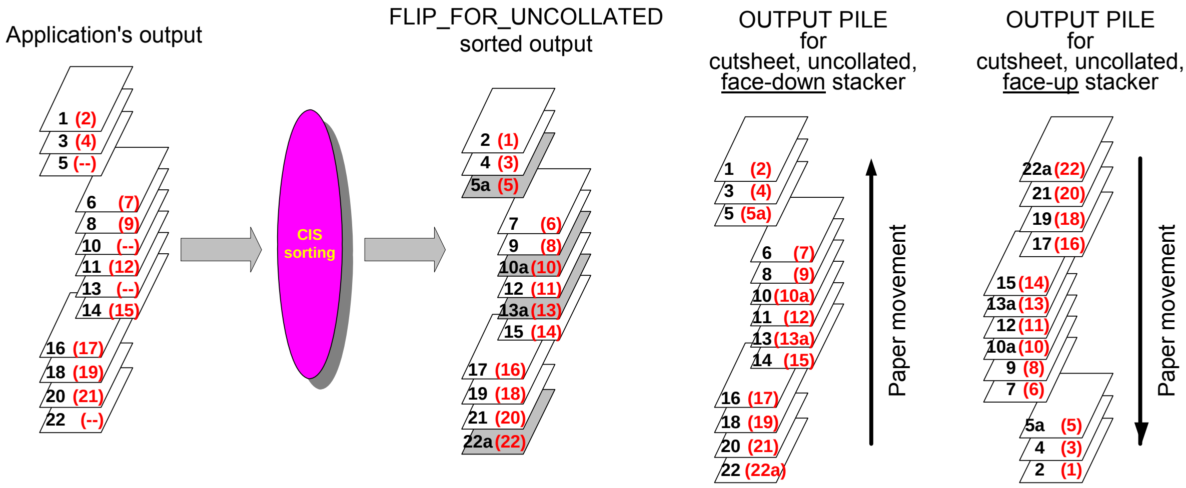


Fig. 61 Collator: FLIP_FOR_UNCOLLATED

3.11 How 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.

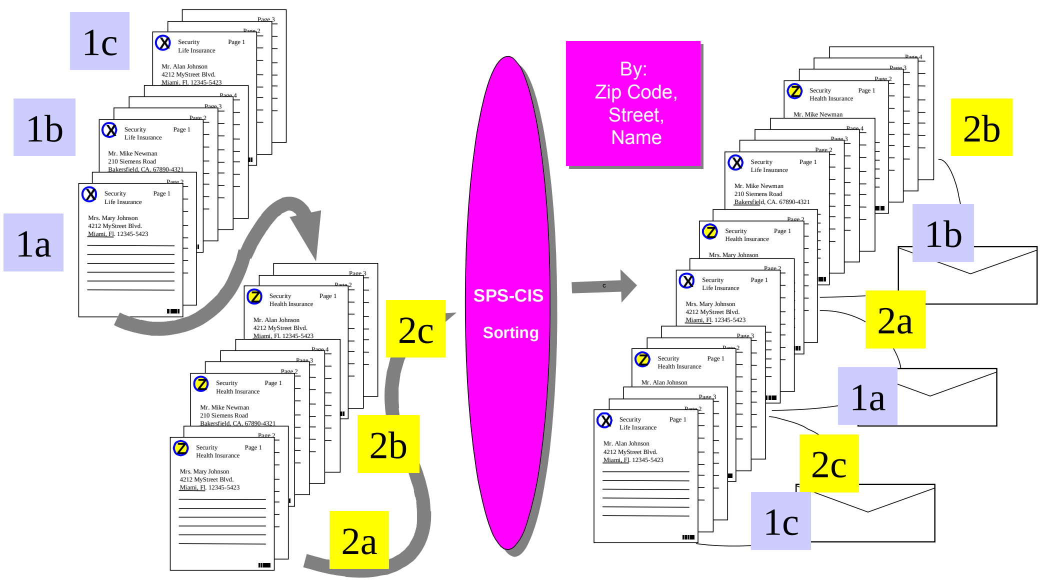


Fig. 62 Mail Piece Consolidation

It is not always possible and many times not even recommended to modify the applications for this kind of post optimization. With CIS it is also not necessary. Provided that some requirements (see below) are fulfilled, the Contents sorting facility of 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 CIS to sort the mail pieces in the correct order. Fig. 4 on page 32 shows the internal CIS workflow. Fig. 62 shows how Mail Piece Consolidation works.

The MPCONSOLIDATION keyword may be used to control the way CIS consolidates two or more consecutive Mail Pieces when they have one or more identical index values.

<u>Requirements and considerations</u>:

Any application to be consolidated using the Contents sorting facility of CIS must fulfil 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 CIS in a previous run (see *3.6 How to convert, normalize and index* on page 55).
2. Resource requirements

   The printing layout of the applications to be consolidated may be completely different (e.g. one application prints Portrait-Simplex, the next one Landscape-duplex and a third one Landscape-2up). 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 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 CIS Consolidation process. It is not possible to use the same name to refer to two or more different resources of the same type (e.g. 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 <u>before</u> the sorting/consolidation process. The page 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 *3.10.1 Contents sorting* on page 115 for more information.

3.12 How to Extract information

Automating quality assurance in a print shop very often requires knowledge about the contents of the Documents and Mail Pieces that are being printed. Without this information, it is almost impossible to know if really ALL documents and mail pieces were correctly sorted, printed, passed though the post-processing, shipped and finally delivered to their addressee. No item should be missing, no one should be duplicate. The information required is normally included in the print file itself, mixed with other information which is not relevant to the quality assurance process. What is required is a simple flat file, a “shipping list”.

Such lists are sometimes created by the application programs as the print files are generated. This is a valid and very efficient procedure. The majority of applications however does not generate this kind of lists and cannot be modified to do so (e.g. third party software). CIS allows installations to select parts of a print file, re-order the printing sequence and to segment the data being printed. These modifications should also be reflected in the “shipping list” in order to simplify the quality assurance process. In summary, lists created by the application programs are not flexible enough. A better alternative is to dynamically create the list using the print file's data.

3.12.1 Defining data extraction levels to CIS

Data extraction in CIS works in a way similar to the data enrichment process: before a data extraction operation can be done, CIS needs to know **what** is going to be extracted, from **where** in the AFP file, and **which** entities should be selected. The DEFINELEVEL keyword is used for this purpose.

CIS level definition works on the base of information units called *entities*. The entities available for data extraction operations are *document*, *sheet-group*, *sheet*, and *page*. Please refer to *7 Glossary* on page 461 for a definition of these terms. CIS also provides a number of pre-defined levels. Please refer to *4.4.5 Predefined Levels* on page 165 for a list of the pre-defined levels available.

Fig. 63 shows the combination of entity, selector and location parameters allowed in a DEFINELEVEL keyword used in connection with data extraction.

Operation	"what"	"where"	"which"
• Indexing objects:			

Operation	"what"	"where"	"which"
ExtractINDEX	✔ **DOCUMENT**	✔ **BEGIN**	✔ **ALL** ✔ ***name*** ✔ ***n*** ✔ ***from-to*** ✔ **ODD** ✔ **EVEN**
	✔ **GROUPLEVELn** ✔ **GROUPLEVEL1**	✔ **BEGIN** ✔ **END**	✔ **ALL** ✔ ***name*** ✔ ***n*** ✔ ***from-to*** ✔ **ODD** ✔ **EVEN**
	✔ **SHEET**	✔ **BEGIN**	✔ **ALL** ✔ ***name*** ✔ ***n*** ✔ ***from-to*** ✔ **ODD** ✔ **EVEN**
	✔ **PAGE**	✔ **ALL**	✔ **ALL** ✔ ***name*** ✔ ***n*** ✔ ***from-to*** ✔ **ODD** ✔ **EVEN**

Fig. 63 CIS data extraction: levels allowed

3.12.2 Extracting information from Index TLE's

Using CIS, an installation may extract AFP indexing information from a print application and write it into a separate sequential file. This is done using the **extractINDEX** keyword. The format of the output file can be easily controlled using the string formatting available as parameter. By requesting the appropriate level (see *3.12.1 Defining data extraction levels to CIS* on page 137) an installation can control the size and level of detail to be included in the extraction file. By specifying more than one Index Tag, an installation can control the degree of information to be included in the file, for each of the items selected. In addition, the **extractCP** keyword can be used to request CIS to use a specific (better yet, standard) Code Page, removing all data encoding dependencies from the extraction file, simplifying its further processing. Such a list is created out of the print file's data and reflects the information to be printed, nothing more, and nothing less.

Considering Fig. 4 on page 32, the data extraction process is done after the normalization, selection, sort and data enrichment steps. For this reason, the data extracted corresponds to the Documents, Mail Pieces and Pages selected, and is written in the same sequence as in the output Document file. The attribute names specified in the extractINDEX keyword may correspond any of the TLE's in the print file, regardless on whether they were built using the TRIGGER-INDEX-FIELD keywords, they were created by the application itself, or they were inserted by CIS using the insertINDEX keyword.

Fig. 64 shows how to use CIS' data extraction facility. Using the application shown in *Fig. 18 Legacy application* on page 60 as basis, this Control file shown will cause CIS to generate a Data Extraction file. The content of the resulting extraction file is shown in Fig. 65.

```
     CC ( ASA )
   TRCTYPE ( IBM )
    INPUTCP( IBM_0500 )

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

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

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

 FIELD ( accountAtt, 'ACCOUNT', ATT )  /* to extract   */
 FIELD ( nameAtt,  'NAME'  , ATT )  /* index the FIELD */
 FIELD ( plzAtt,   'PLZ'  , ATT )  /* must be defined */
                     /* as ATTRIBUT   */
  EXTRACTCP ( IBM_0500 )
 EXTRACTINDEX ( EXTRDD, MP,
        '%10.10s;%32.32s;%5.5s;\r\n',
        accountAtt,
        nameAtt,
        plzAtt
       )
```

Fig. 64 Example: Data Extraction parameters

```
         1         2         3         4         5
1...|....0....|....0....|....0....|....0....|....0

14034350  ;William Lieberwirt             ;99091;
62765426  ;Charlotte Ohnesorge            ;70173;
87421358  ;Sabine Bachstein               ;86153;
12387642  ;Gerhard Wiesler                ;33611;
...
```

Fig. 65 Example: Data Extraction file contents

3.13 How to Analyze a print file

Which operator involved in the electronic printing business does not know this problem: an application is ready for print, but nobody can say for sure how big it is (number of sheets, number of Mail Pieces), what printer features are required (high-light color, duplex, input bins), which resources it uses (fonts, page segments), etc.

Whether this happens because of the increment in distributed printing or because of files being printed after been archived for years, not knowing what exactly is going to be printed has caused serious headaches to the people involved in scheduling workload in large print shops.

Using the CIS' Print Stream Analyzer, an installation may request CIS to do a complete verification of the print application, creating a Summary file with all the information required to know exactly what the file is going to consume. In connection with archiving systems, CIS cannot only be used to extract the indexing information required, but also to create the Summary file which can then be archived together with the print file, increasing the usability of information being archived.

- **THE CIS ANALYST FILE**

As it is to be expected, the Print Stream Analyzer runs as last process in the CIS internal workflow (please refer to Fig. 4 on page 32). The Summary file it creates contains the information gained in all previous steps. The name of the Analyst file is specified using the ANALYSTDD keyword. The ANALYSTCP keyword can be used to specify the encoding of this file.

The Analyst output file is based on the XML (eXtensible Mark-up Language) format. It uses three different kind of tags:

- Root tag
- Branch tag
- Leaf tag

A **Root** tag encloses the entire document and is the only element that has no parent structure. The elements that are contained in a Root tag are called sub-elements. They may also contain sub-elements (**Branch** tags) or not (**Leaf** tags). Leaf tags only contain character data and have no sub-elements.

The CIS Analyst file also contains an internal DTD (Document Type Definition) that describes the document structure. The DTD is a powerful feature of XML that provides a format set of rules that define the document's structure.

Example:

An installation wants to archive a print application. The information archived is intended to be used for reprints. The reprints may need to be done on different locations. In order to determine which location is appropriate and which not, the characteristics of the print file should be archived too. Fig. 66 shows the CIS Control file that can be used to create such a Summary File.

```
ANALYSTDD ( MYFILE )
ANALYSTCP ( ISO_8859-15 )
 ANALYST ( XML )
```

Fig. 66 Example: Analyst parameters

Fig. 67 shows the resulting file contents. The abstract DTD tree (not shown in the picture) is automatically generated by CIS. The description of each of the tags is also included.

```
<?xml version="1.0"?>
<ANALYST>
 <VERSION>CIS version 4.01.15</VERSION>
 <OUTPUTDATA>
  <DOCUMENT>
   <FILE>
    <FILENAME>CISOUT</FILENAME>
    <FILESIZE unit="byte">97529159</FILESIZE>
    <FILECONTENTS>
     <DOCUMENTS unit="number">1</DOCUMENTS>
     <MAILPIECES unit="number">5280</MAILPIECES>
     <SHEETS unit="number">6720</SHEETS>
     <PAGES unit="number">13440</PAGES>
    </FILECONTENTS>
   </FILE>
   <SUMMARY>
    <DOCUMENTS unit="number">1</DOCUMENTS>
    <MAILPIECES unit="number">5280</MAILPIECES>
    <SHEETS unit="number">6720</SHEETS>
    <PAGES unit="number">13440</PAGES>
    <MINPAGESIZE unit="byte">1486</MINPAGESIZE>
    <MAXPAGESIZE unit="byte">13070</MAXPAGESIZE>
    <AVERAGEPAGESIZE unit="byte">7168</AVERAGEPAGESIZE>
    <JOBSIZE unit="byte">97529159</JOBSIZE>
   </SUMMARY>
  </DOCUMENT>
```

```
   <RESOURCE>
    <FILE>
     <FILENAME>CISRES</FILENAME>
     <FILESIZE unit="byte">1431006</FILESIZE>
     <FILECONTENTS>
      <RESOURCENAME>F1010111</RESOURCENAME>
      <RESOURCETYPE identifier="0x1">form definition</RESOURCETYPE>
      <RESOURCENETSIZE unit="byte">290</RESOURCENETSIZE>
      <RESOURCENAME>X0357E </RESOURCENAME>
      <RESOURCETYPE identifier="0x5">coded font</RESOURCETYPE>
      <RESOURCENETSIZE unit="byte">63</RESOURCENETSIZE>
      <RESOURCENAME>T1FG999P</RESOURCENAME>
      <RESOURCETYPE identifier="0x7">code page type</RESOURCETYPE>
      <RESOURCENETSIZE unit="byte">2230</RESOURCENETSIZE>
      <RESOURCENAME>C0357E </RESOURCENAME>
      <RESOURCETYPE identifier="0x6">font character set (raster)</RESOURCETYPE>
      <RESOURCENETSIZE unit="byte">24145</RESOURCENETSIZE>
      <RESOURCENAME>O1MMO001</RESOURCENAME>
      <RESOURCETYPE identifier="0x4">overlay</RESOURCETYPE>
      <RESOURCENETSIZE unit="byte">1911</RESOURCENETSIZE>
     </FILECONTENTS>
    </FILE>
   </RESOURCE>
   <INDEX>
    <FILE>
     <FILENAME>CISIDX</FILENAME>
     <FILESIZE unit="byte">1230488</FILESIZE>
    </FILE>
   </INDEX>
   <EXTRACT>
    <FILE>
     <FILENAME>MPLIST.TXT</FILENAME>
     <FILESIZE unit="byte">269280</FILESIZE>
     <FILECONTENTS>
      <ATTRIBUTENAME>MP Barcode-Nummer</ATTRIBUTENAME>
     </FILECONTENTS>
    </FILE>
   </EXTRACT>
  </OUTPUTDATA>
</ANALYST>
```

Fig. 67 Example: Analyst Summary file

A detailed description of each of the XML tags contained in the Summary file can be found under *6.3 Appendix C: Description of the Analyst tags* on page 346.

3.14 AFP Resource Packaging

The *AFP Resource Packaging* option of CIS will be released as a modular software component for integration in other Océ products under Linux and z/OS. It cannot be ordered separately as a standalone product or for use with non-Océ software products. Its functionality is restricted to a defined subset of parameters listed under *6.1.1 Licensing under OS/390, z/OS* on page 323 and *6.1.2 Licensing under LINUX* on page 327.

AFP resource packaging consists of identifying, locating and copying all the AFP resources used in a print job into a single Resource Object file. During this process, the AFP data is validated, parsed and eventually converted to MO:DCA-P format (e.g. in case of line-format applications). Summarized:

- Normalize S/370 Line format, Record-format line, XML or MO:DCA-P format data (see *3.1 Internal Workflow* on page 29) with full Code page support (see *3.2 Code page support* on page 33) and
- Retrieve and Consolidate AFP Resources with the parameter RESCONSOLIDATION set to its default BASIC (see 3.7 How to Retrieve and Consolidate AFP Resources on page 69) into a single Resource Object file (parameter RESPDS set to its default SEQ).

The following diagram shows the internal CIS workflow, presenting the functions which are enabled in case of the CIS *AFP Resource Packaging* option (compare with *CIS internal workflow* on page 32):

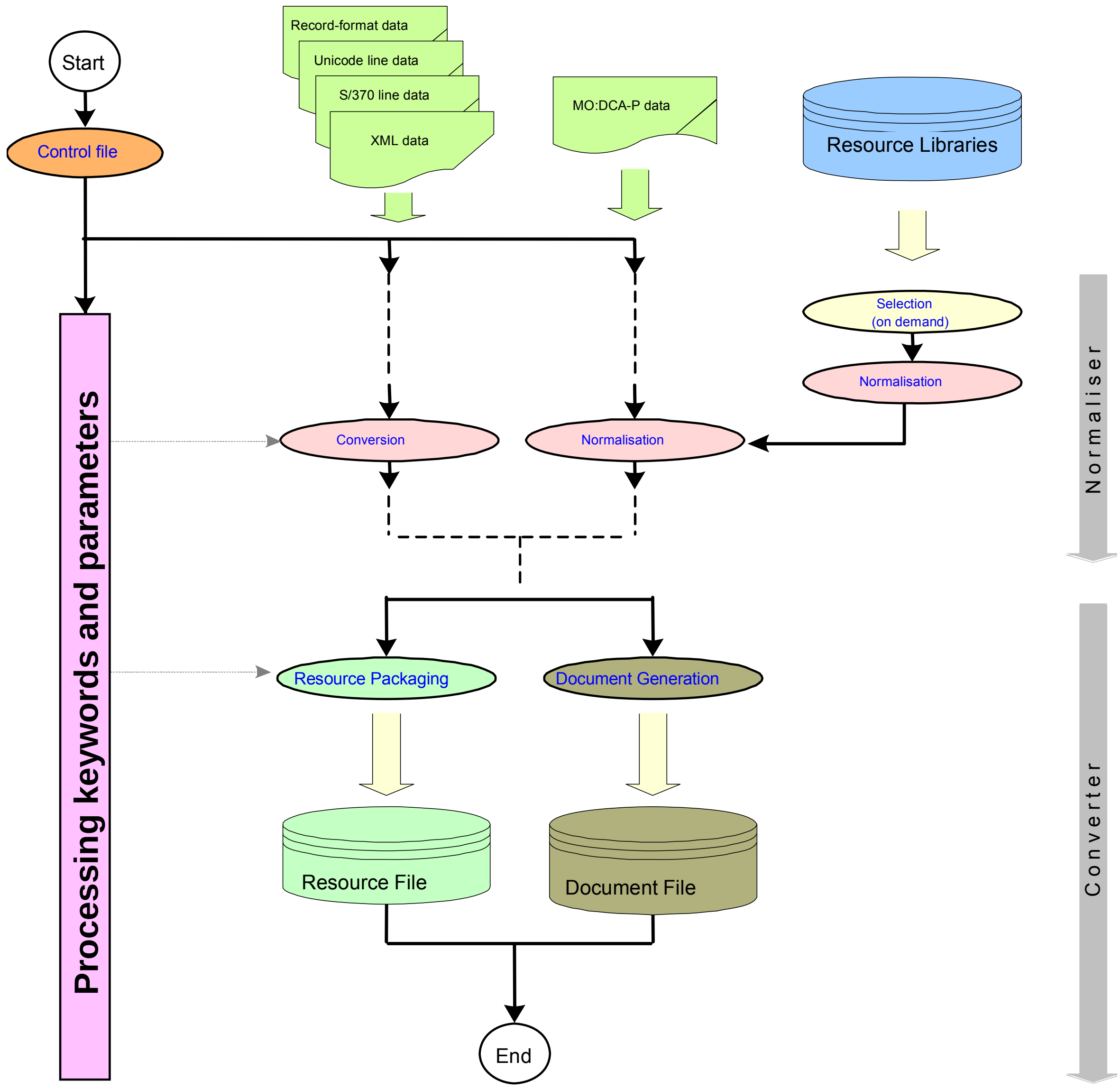


Fig. 68 CIS AFP Resource Packaging workflow

4 Using CIS

Executing CIS under Linux

4.2.1 Execution requirements

CIS execution is started by entering the following command under Linux:

```
cis control-file [ codePageID ]
```

Fig. 69 Linux: sample CIS invocation

Where:

cis Specifies the CIS program name.

control-file Specifies the name of the file containing the keywords and parameters to be used for this CIS invocation. The syntax supported is described on page 160 under *CIS Control File reference*. This is a required parameter.

codePageID Specifies the Code Page ID to be used for interpreting the keywords and parameters stored in the control file. See also *3.2.1 Supported Code Page IDs* on page 38. By default, when this parameter is not specified, CIS uses the currently active internal Code Page (ISO_8859-15 in case of Linux).

During execution, CIS uses `stdout` as default destination for all runtime messages it generates. The messages include information such as product's version number and control file contents. Error and warning messages describing an invalid or unexpected condition are also issued. All messages issued are described in the CIS Message's Guide.

Return codes

On termination, CIS passes a return code which summarizes the errors detected during execution. The return codes (decimal) set by CIS are:

-4 Licence error detected.
PRISMAproduction server only !
On PRISMAproduction host return code 16 is set.

0 Successful execution. No errors were found.

4 Successful execution with warning messages.

8 Data error detected during execution.
One or more error messages were written to `stdout` describing the problem found.

12 Keyword or File related error detected during execution.
One or more error messages were written to `stdout` describing the problem found.

16 Program related error detected during execution.
One or more error messages were written to `stdout` describing the problem found.

4.2.2 License key processing

For details please see *PRISMAproduction User Manual – Administration Settings and Tasks V4.02*, chapter *Configuration Licenses*.

4.2.3 Estimating System Resource requirements

- <u>**MEMORY REQUIREMENTS**</u>

  - When converting line data
  - When normalizing AFP data
  - When indexing

- When retrieving resources
- When sorting
- When tracing

- <u>**DISK REQUIREMENTS**</u>
    - For index file
    - For resource file
    - For document file
    - For trace files

- <u>**PROCESSOR REQUIREMENTS**</u>
    - When converting line data
    - When normalizing AFP data
    - When indexing
    - When retrieving resources
    - When sorting
    - When tracing

Executing CIS under MVS

4.3.1 JCL requirements

CIS runs as a normal batch job and requires Job Control Language (JCL) definitions. Fig. 70 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 CIS program name 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 *Memory requirements* on page 159.

The following load modules are part of CIS:

- CIS — This is the standard module. Should be used to run all production applications.

- CIS1 — This is the trace generation module and is used during error diagnosis. It gets automatically loaded when tracing is requested (see TRACELEVEL keyword).

Renaming the CIS modules is not supported and will cause unpredictable results.

PARM The following run time keywords are available:

CP= Specifies the code page ID to be used for interpreting the keywords and parameters stored in the CIS Control file. See also *3.2.1 Supported Code Page IDs* on page 38. By default, when this parameter is not specified, CIS uses the currently active internal Code Page. For more information please refer to *3.2 Code page support* on page 33.

PARMDD= Specifies the DDname of the data set containing the CIS Control file. The default, when this keyword is not specified, is SYSIN.

MSGDD= Specifies the DDname of the data set used by CIS to write all execution messages. The default, when this parameter is not specified, is SYSPRINT.

STEPLIB Specifies the DDname for the library where CIS is installed

CISLSK Specifies the DDname of the data set containing the license key to be used for this CIS run. The format of the license key file under MVS and the processing done in CIS is shown in *4.3.2 License key requirements* on page 155.

SYSPRINT Indicates the destination where CIS will generate the information and error messages it issues. May be overridden using the MSGDD keyword in the EXEC JCL statement.

The information messages contain the CIS module name, version number and the input control file read. This information is displayed twice in case tracing is specified.

SYSTERM Indicates the destination where CIS and the C runtime library will generate additional messages in case a severe runtime error occurs. May be allocated to DUMMY.

```
[ //*************************************************************** ]
[ //*  THIS STEP EXECUTES CIS                     ]
[ //*************************************************************** ]
 //RUNCIS  EXEC PGM=CIS[,REGION=5M,]
[ //      PARM='CP=IBM_0500,PARMDD=SYSIN,MSGDD=SYSPRINT'      ]
 //STEPLIB DD  DSN=PPH4.CIS.LOAD,DISP=SHR
 //CISLSK  DD  DSN=PPH4.CISLSK.FILE,DISP=SHR
 //SYSPRINT DD  SYSOUT=*
[ //SYSUDUMP DD  SYSOUT=*                     ]
[ //SYSTERM DD  SYSOUT=*                     ]
[ //CISTRA1 DD  DSN=PPH4.CIS.TRACE1,DISP=(NEW,CATLG,CATLG),     ]
[ //      DCB=(RECFM=VBS,BLKSIZE=32760),              ]
[ //      SPACE=(CYL,(100)),UNIT=SYSDA               ]
[ //CISTRA2 DD  DSN=PPH4.CIS.TRACE2,DISP=(NEW,CATLG,CATLG),     ]
[ //      DCB=(RECFM=VBS,BLKSIZE=32760),              ]
[ //      SPACE=(CYL,(100)),UNIT=SYSDA               ]
[ //CISIDX  DD  DSN=PPH4.CISOUT.INDEX,DISP=(NEW,CATLG,CATLG),    ]
[ //      DCB=(RECFM=VBA,BLKSIZE=32760,LRECL=32756),        ]
[ //      SPACE=(CYL,(5,5)),UNIT=SYSDA               ]
[ //CISRES  DD  DSN=PPH4.CISOUT.RES,DISP=(NEW,CATLG,CATLG),     ]
[ //      DCB=(RECFM=VBA,BLKSIZE=32760,LRECL=32756),        ]
[ //      SPACE=(CYL,(5,5)),UNIT=SYSDA               ]
 //CISOUT  DD  DSN=PPH4.CISOUT.DOC,DISP=(NEW,CATLG,CATLG),
 //      DCB=(RECFM=VBA,BLKSIZE=32760,LRECL=32756),
 //      SPACE=(CYL,(5,5)),UNIT=SYSDA
 //CISIN  DD  DSN=CIS.OPEN2000.DATA,DISP=SHR
 //SYSIN  DD  DATA,DLM='%%'

     /* CIS keywords and parameters are included here */

 %%
[ //*************************************************************** ]
[ //*  THIS STEP CONCATENATES THE CIS OUTPUT FILES          ]
[ //*************************************************************** ]
[ //  IF ( RUNCIS.RC <= 4 ) THEN                   ]
[ //CONCACT EXEC PGM=IEBGENER                    ]
[ //SYSPRINT DD  SYSOUT=*                     ]
[ //SYSIN  DD  DUMMY                       ]
[ //SYSUT1  DD  DSN=PPH4.CISOUT.INDEX,DISP=OLD            ]
[ //      DD  DSN=PPH4.CISOUT.RES,DISP=OLD             ]
[ //      DD  DSN=PPH4.CISOUT.DOC,DISP=OLD             ]
[ //SYSUT2  DD  DSN=PPH4.CISOUT.ALL,DISP=(NEW,CATLG,CATLG),     ]
[ //      DCB=(RECFM=VA,BLKSIZE=32760,LRECL=32756),        ]
[ //      SPACE=(CYL,(15,15)),UNIT=SYSDA              ]
[ //  ENDIF                          ]
```

Fig. 70 Sample JCL to invoke CIS

SYSIN Specifies the DDname for the data set containing the control file for CIS. The syntax of the keywords and parameters supported is described on page 160 under *CIS Control File* reference . A data delimiter other than /* should be specified as this string is used in the CIS control file syntax. . May be overridden using the PARMDD keyword in the EXEC JCL statement.

CISIN Specifies the DDname for the print file to be processed by CIS. It is possible to specify a different DDname using the INPUTDD keyword. Refer to *4.4 CIS Control File reference* for more information. The data set used as input to CIS may include an Index and a Resource Object (inline resources).

CISIDX Specifies the default DDname for the Index Object file to be generated by CIS. It is possible to specify a different DDname using the INDEXDD keyword. Refer to *CIS Control File* reference for more information.

The Index Object file may have the following characteristics:

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

Do not specify carriage control (e.g. VB instead of VBA) if you want 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.

CISOUT Specifies the default DDname for the Output document file to be generated by CIS. It is possible to specify a different DDname using the OUTPUTDD keyword. Refer to *CIS Control File* reference for more information.

The output file may have the following characteristics:

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

Do not specify carriage control (e.g. VB instead of VBA) if you want 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.

CISRES Specifies the default DDname for the Resource Object file to be generated by CIS. It is possible to specify a different DDname using the RESOBJDD keyword. Refer to *CIS Control File* reference for more information.

The Resource Object file may have the following characteristics:

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

Do not specify carriage control (e.g. VB instead of VBA) if you want 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 The current CIS version uses MVS hiperspaces as internal work file. The specification of a CISSWAP DDname and the allocation of a VSAM data set is not required.

CISTRA1
CISTRA2 Specify the DDname for the trace data sets where CIS will write diagnostic information in case the TRACE function is activated. The two data sets specified are processed in a wrap-around manner.

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

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

CIS trace data sets must be transferred in binary format (e.g. via FTP). The trace data sets may contain secondary extents. In this case, the data set 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 CIS are:

0 Successful execution. No errors were found. Control file contents and one or more information messages were written to SYSPRINT.

4 Successful execution with warning messages. Control file contents and one or more warning messages were written to SYSPRINT.

8	Data error detected during execution. Control file contents and one or more error messages were written to SYSPRINT.
12	File or Parameter related error detected during execution. Control file contents and one or more error messages were written to SYSPRINT.
16	Program related error detected during execution. One or more error messages were written to SYSPRINT describing the problem found.

IEBGENER This standard MVS utility may be optionally used after an CIS run in order to concatenate the Index, Resource and/or Document output files into a self-contained AFP file which includes in-line resources.

The resource object file used in the example on page 152 (DDname is CISRES) must be generated using the parameter RESFILE(SEQ).

4.3.2 License key requirements

In order to obtain a license key for your product your Océ representative requires your CPUID or CPUID's in the case of a loosely coupled system. **Only one license key is required for a single or multi-processor system.**

The MVS command D M=CPU should be used and the output from this command emailed or faxed to your Océ representative. Examples of the output from this command are shown below.

- **SINGLE PROCESSOR SYSTEM:**

```
RESPONSE=OCE
 IEE174I 15.31.24 DISPLAY M 250
 PROCESSOR STATUS
 ID CPU         SERIAL
 0  +          01D5DA2066
```

- **MULTI-PROCESSOR SYSTEM:**

```
RESPONSE=OCE
 IEE174I 15.31.24 DISPLAY M 250
 PROCESSOR STATUS
 ID CPU         SERIAL
 0  +          01D5DA2066
 1  +          11D5DA2066
 2  +          21D5DA2066
 3  +          31D5DA2066
```

4.3.3 License key handling

The license key information that is sent per email or post on floppies consists of 2 files:

- `Lcxxxxxx.eti` is an information file that should be presented if there are problems or questions pertaining to your license.
- `License` contains your license key for your product.

The license file sent has the following format:

```
CIS          5.0     zbghjT+23ypXPxMMNPW:ltgg
SNR_CC99009990     0.0     9uXXXX4-+P-YqpVyzyz4PQAv
```

Fig. 71 License file used for distribution

Note: If you prefer copy and pasting your license key to a dataset, copy the 24 byte license string in that way, that it starts at the first column of a record. See chapter License key file syntax below.

In order to avoid problems caused by differing code pages a stand-alone tool (program), License Import Module (PPHLIM), reads data file(s) that have been transmitted in binary format to the mainframe, transforms them into a valid license control file data format and appends the transformed data to the PPHost-license control file.

```
//ADDKEY   EXEC PGM=PPHLIM
//SYSPRINT DD  SYSOUT=*
//SYSTERM DD  SYSOUT=*
//STEPLIB DD  DSN=&PREFIX..PPHLOAD,DISP=SHR
//INFILE  DD  DSN=binaryFileName_1,DISP=SHR
//      DD  DSN=binaryFileName_2,DISP=SHR
//OUTFILE DD  DSN=&PREFIX..licfile,DISP=OLD
```

Fig. 72 JCL step required to run PPHLIM

One or more license key files may be concatenated on the INFILE DD statement so that only a single PPHLIM run is required to create or update the PRISMAproduction Host license control file.

Input license files may be binary license key files generated by Océ or license control files generated by PPHLIM.

NOTE: When files are concatenated on the INFILE DD statement the files must have the same attributes for RECFM, LRECL and BLKSIZE.

DD	RECFM	LRECL	BLKSIZE	BLOCKS	DIR	DISP	ORG
INFILE	VB FB	80-32756	84-32760 Multiple of LRECL	?,?	?	SHR	PS
OUTFILE	FB	80	Multiple of LRECL	?.?	?	OLD	PS

When the INFILE DD card is omitted, PPHLIM decodes all the information already contained in the PPHOST license control file (OUTFILE) and displays it in SYSPRINT. The summary includes information such as product ID, expiration date, CPU ID, etc. Processing terminates with return code 4.

In all other cases, PPHLIM validates the contents of the input file(s) specified in INFILE and creates or updates the PRISMAproduction Host license control file (OUTFILE). At the end, a summary report for all the keys found in the PPHOST license control file is displayed in SYSPRINT. The summary includes information such as product ID, expiration date, CPU ID, etc. Processing terminates with return code 0.

In case of errors (e.g. invalid input key, invalid PPHOST file, etc.) a descriptive message is issued to SYSTERM or to the system console if SYSTERM is not assigned and the processing terminates with a return code of 12.

All warnings and error messages are self-explanatory.

- **LICENSE KEY FILE SYNTAX**

LSKEY control file

```
/*** DO NOT EDIT THIS FILE BY HAND, USE PPHLIM INSTEAD ***/
j--4Met2EY3Z7BZAzAAxRCIZ
 /* CPU:01d5da206600 Rank:0003 Expiration:2003-12-31 */
 /* Product:????????(CIS 5.00) */
qY2AXTc69iHCKUFUPh3cCG%J
 /* CPU:d5da20660000 Rank:0000 Expiration:2003-07-27 */
 /* Product:a6072138(ROUTER 2.0) */
```

Fig. 73 License Code control file sample

The file containing the CIS license code is specified using the LSKEYDD keyword. The license file consists of comments and license codes (see Fig. 73). A valid license code starts at the first column and ends with the first blank character, a comment string or with the end of line.

The license key file may contain one or more license codes for CIS and other PRISMAproduction Host products, for the same or for different CPU's. CIS will process all the license codes in the file and will select the one with the highest license type for the current CPU (for more information please refer to *6.1.1 Licensing under* OS/390, z/OS

License Types on page 323).

A request may be made from an Océ representative for your MAC ADDRESS. This is equivalent to your mainframes CPUID and is stored also in the file Lcxxxxxx.eti.

4.3.5 In case of error

In case of an internal error CIS brings a message to console possibly followed by an ABEND. Read the job log carefully. If the error situation cannot be corrected contact your Océ representative providing as many information as possible:

- Complete job log
- Input data and Resources used for this job or in case of sensitive data at least a
- Trace (see 4.3.1 JCL requirements on page 149 for further information about using CIS1 Module for tracing)

4.4 CIS Control File reference

4.4.1 Syntax rules

1. An CIS Control File is a text file that can consist of:

   - Keywords, their parameters and sub-parameters as described below
   - Comments
   - Blank characters

   The control file may be coded in any of the standard code pages supported by CIS (see *3.2 Code page support* on page 33) and is portable across the different system platforms as long as the file transfer process does not affect the encoding standard and line breaks used in the control file and the file naming conventions are not in conflict.

2. In general, keywords and their parameters are not case sensitive. Exception are those parameters that are enclosed using single quotation marks (`'Case Sensitive'`). File names (including absolute or relative path information) and pagedef and formdef should be specified using double quotes (`"user/myFile.CIS"`) when using a case sensitive file system such as Linux.

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. Parameters within a keyword are positional and must be specified in the documented sequence.

5. The parameters and sub-parameters are positional and must be specified in the sequence indicated in this manual. A Comma (**,**) is used to separate parameters. A Semi-colon (**;**) is used to separate sub-parameters.

6. Numeric values may be entered in decimal or hexadecimal notation. The following formats (which are not case sensitive) may be used for hexadecimal input:

   - X'*nn*'
   - 0x*nn*

   A decimal point is not allowed in parameters entered using the hexadecimal notation.

7. Blank characters, tab characters, carriage return and line feed may be used to improve the readability of the control file. They can appear anywhere between keywords and parameters, and are ignored unless they appear as part of a parameter enclosed using two quotation marks (e.g. `'parameter with blanks'`). Blank lines are also allowed.

8. 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.

9. Square brackets (**`[ ]`**) are used to indicate optional parameters.

10. Curly brackets (**`{ }`**) are used to group parameters which belong and must be entered together.

11. A vertical bar (**|**) is used in the documentation to separate two or more options where only one may be specified.

12. Underlined text (default) is used to show the default values when a keyword is not specified.

4.4.2 Formatting string

A special formatting string is available in some of the CIS keywords. The string is used as a template and is normally followed by a series of data fields (*'attributeName', fileName,* etc.) which are then merged with the template, according to the formatting options specified.

This string formatting facility is based on the **vsprintf** functionality available in the C programming language, and provides a high degree of flexibility when it comes down to specifying fixed field lengths, right/left alignment, leading zeroes and others. Each formatting string consists of one mandatory “template” and one or more optional insertion fields.

There must be enough arguments for the format. Excess arguments (more than specified in the formatting template) are ignored.

The support provided is restricted to data encoded in any of the code pages supported by CIS (see *3.2 Code page support* on page 33) as long as the characters to be displayed can be represented in the CIS Internal code page. Double-byte encoding schemes (e.g. UCS-2) are not supported.

- **<u>TEMPLATE CONTENTS</u>**

The formatting template is a character string which may contain two types of information:

- Normal text which is copied, unchanged, into the output string.
- Conversion and formatting escape sequences which are applied to the series of data fields (arguments) for formatting.

The formatting template may be 1 to 250 characters long. It may contain any of the code points (characters) defined in the Control File's code page. The string may combine lower case, upper case and national characters as long as they are defined in the code page. The string is always processed from left to right.

Two characters have a special meaning when specified in the formatting template: the percent sign (**%**) and the back slash (****) character. They indicate the beginning of a format specification which is processed according. The escape sequences supported by CIS are:

`\r\n`	Force a new line before processing the rest of the string.
`\t`	Generate one horizontal tab character.
`\'`	Generate one single quotation mark.
`\"`	Generate one double quotation mark.
`\\`	Generate one back slash character.
`%%`	Generate one percent sign.

The escape sequences starting with a percent sign indicate that a field parameter is to be formatted. One field parameter should be specified for each of these escape sequences. Escape sequences may be specified one after the other with or without separating text. The complete escape sequence will be removed from the resulting string and replaced with the formatted field data. The resulting string may be up to 250 characters long depending on the keyword in which it is being used.

The options available are:

`% [ `*`flags`*` ] [ `*`width`*` ] [ `*`.precision`*` ] `*`type`*

% Indicates the beginning of a escape sequence.

flags Optional parameter. May be:

-	Left align the result within the given field width. Default is right.
+	Prefix the output value with a sign (+ or −). Default is no sign.
blank	Prefix the output with a blank character. Default is no blank.

Prefix the output with a `0x` string. Default is no prefix.
0 Prefix the output with zeroes until width is reached.

width Optional parameter. It is a decimal integer controlling the minimum number of characters to be printed. It is used in combination with the *flags* parameter when the number of characters in the output value is less than the specified width. In this case blanks (or zeroes) are added to the left (or right) until the minimum width is reached.

precision Optional parameter. It specifies a decimal integer preceded by a period. Controls the number of characters or significant digits to be printed. Unlike the width specification, precision may cause truncation of the output value.

type The types supported by CIS are:

s Text string.
x Unsigned hexadecimal integer, using “abcdef”.
X Unsigned hexadecimal integer, using “ABCDEF”.
d Signed decimal integer.

Note that all parameters are case sensitive.

4.4.3 Portability

1. Directory name

   Under MVS, the *directory* parameter represents a Partitioned Data Set (PDS) that is specified either as a DDNAME or as a DSNAME depending on the keyword in which it is specified. A DDNAME may be 1 to 8 alphanumeric or national ($, #, @) characters long and may not start with a numeric character (e.g. "`MYDD$$1`"). A DSNAME may consist of one or more period-separated strings of 1 to 8 alphanumeric or national ($, #, @) characters each, where the first character is not a number and the total DSNAME does not exceed 44 characters (e.g. "`PRISMA.V400.MYDSNAME`"). All names are automatically converted to uppercase.

   Under LINUX, the *directory* parameter specifies either a relative directory (e.g. "`./myDir`") in which case the name is appended to the current working directory, or it contains an absolute directory name (e.g. "`/home/myDir`") in which case the current working directory is ignored. All paths are case sensitive.

2. File name

Under MVS, the *fileName* parameter represents a sequential data set that is specified either as a DDNAME or as a DSNAME depending on the keyword in which it is specified. A DDNAME may be 1 to 8 alphanumeric or national ($, #, @) characters long and may not start with a numeric character (e.g. "`MYDD$$1`"). A DSNAME may consist of one or more period-separated strings of 1 to 8 alphanumeric or national ($, #, @) characters each, where the first character is not a number and the total DSNAME does not exceed 44 characters (e.g. "`PRISMA.V400.MYDSNAME`"). All names are automatically converted to uppercase.

Under LINUX, the *fileName* parameter specifies either a relative directory name together with a file name (e.g. "`./myDir`") in which case the name is appended to the current working directory, or it contains an absolute directory name together with a file name (e.g. "`/home/myDir/myFileName`") in which case the current working directory is ignored. All names are case sensitive.

4.4.4 Predefined color names

Fig. 74 lists all the predefined color names available in CIS and shows the keywords in which they are allowed. Not all color names and/or models are supported by all MO:DCA objects.

Keyword	OCA																					non-OCA			other			
	BLACK	BLUE	BROWN	CYAN	DARKBLUE	DARKCYAN	DARKGREEN	DARKTURQ	DEFAULT	GREEN	GREY	MAGENTA	MUSTARD	NONE	ORANGE	PINK	PURPLE	RED	TURQ	WHITE	YELLOW	BUFF	IVORY	VIOLET	GOLD	GOLDENROD	SILVER	Self Defined
DEFINEBCOCA	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X							
DEFINECONTAINER	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				X
DEFINEPTOCA	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				X
DEFINEBOX	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				X
DEFINECIRCLE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				X
DEFINECURVE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				X
DEFINEELLIPSE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				X
DEFINELINE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				X
DEFINEMEDIUMMAP	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

Fig. 74 DEFINECOLOR names vs. CIS keyword

4.4.5 Predefined Levels

The following table shows the predefined levels available in CIS describing the entity, location and selector used. Predefined levels cannot be changed by the user. Please refer to refer to *3.8.1 Defining data enrichment levels to CIS* on page 78 for more information about the DEFINELEVEL keyword

Level name	Description
DOC	defineLEVEL('DOC', DOCUMENT, BEGIN, ALL)
MP	defineLEVEL('MP', GROUPLEVEL1, BEGIN, ALL)
SH	defineLEVEL('SH', SHEET, BEGIN, ALL)
PAG	defineLEVEL('PAG', PAGE, ALL, ALL)

Fig. 75 Predefined Levels

4.4.6 Units of measurement

The standard way of specifying units of measurement for almost all CIS keywords allows any of the following units to be used:

IN specifies a unit of inches
MM specifies a unit of millimeters
CM specifies a unit of centimeters
POINTS specifies a unit of points (1/72 inch)
PELS specifies a unit of picture elements (1/240, 1/300, 1/600, ... inch)
MILLIPOINTS specifies a unit of millipoints (1/72000 inch)

The following table shows the way the different measurement units are converted.

	IN	CM	MM	POINTS	PELS	1/1440 inch	MILLIPOINTS
10 x IN	**10**	25,4	254	720	2.400	14.400	720.000
10 x CM	3,937008	**10**	100	283,464567	944,881890	5.669,291339	283.464,566929
10 x MM	0,393701	1	**10**	28,346457	94,488189	566,929134	28.346,456693

10 x POINTS	0,138889	0,352778	3,527778	**10**	33,333333	200	10.000
10 x PELS	0,041667	0,105833	1,058333	3	**10**	60	3.000
10 x 1/1440 INCH	0,006944	0,017639	0,176389	0,5	1,666667	**10**	500
10 x MILLIPOINTS	0,000139	0,000353	0,003528	0,01	0,033333	0,2	**10**

Fig. 76 Units of measurement: equivalence table

The following table shows the values that correspond to 2^{32}-1 millipoints (typical upper range):

IN	CM	MM	POINTS	PELS	1/1440 inch	MILLIPOINTS
59652,32354	151516,9018	1515169,018	4294967,295	14316557,65	85899345,897	**4294967295**

Fig. 77 Units of measurement: maximum values

4.4.7 Keyword and Parameter syntax

<table>
<tr><th>CIS
Keyword and Parameter syntax</th><th>CONVERSION</th><th>INDEXING</th><th>SORTING</th><th>RESOURCES</th><th>SELECTION</th><th>SEGMENTATION</th><th>ENRICHMENT</th><th>EXTRACTION</th></tr>
<tr><td>ANALYST (XML | NONE)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>ANALYSTCP (codePageId)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>ANALYSTDD (CISANA | fileName | "fileName")</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CASECTRL (0, 1, 2, 3)</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CC (NO | SNI | ASA | IBM | ASAA | SS | DS | TS)</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CHARS (fontName [<mxm>] [, fontName [<mxm>]] [,...])</td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td></tr>
<tr><td>CODEDFONTEXT ("" | "extension" [,"extension"] [,...])</td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td></tr>
<tr><td>CODEPAGEEXT ("" | "extension" [,"extension"] [,...])</td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td></tr>
<tr><td>COLORFIDELITY ({ASIS | NONE | STOP | CONTINUE}, { YES | NO })</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>COLORMAP (colorMapName | DUMMY)</td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td></tr>
<tr><td>COMPRESS (NO | CHARS | { STRING, ‘string’ [, ‘string’] [, ...] } |
{ ALL , ‘string’ [, ‘string’] [, ...] })</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>DEFINE2DDATAMATRIX (‘name’, [escape], [convert], [rowSize],
[rowNumber], [sequenceIndicator], [totalSymbols], [fileId1], [fileId2],
[functionFlags] [,...])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>DEFINE2DMAXICODE (‘name’, [escape], [convert], mode,
[sequenceIndicator], [totalSymbols], [zipper])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>DEFINE2DPDF417 (‘name’, [escape], [convert], dataSymbolsNumber,
[rows], [securityLevel], [controlBlockData])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>DEFINE2DQRCODE (‘name’, [escape], [convert], [codePageId],
[symbolVersion], [sequenceIndicator], [totalSymbols], [parityData],
[errorCorrectionLevel], [functionFlags], [applicationIndicator])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>DEFINEBCOCA ('name', [unit] , posX , posY , [orientation] , [color] ,
[type] , [modifier] , [hri] , [asterisk] , [height] , [modWidth] ,
[ratio] , [codePageId])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>DEFINEBOX (‘name’ , [unit] , p1X , p1Y , p2X , p2Y , [rounded] ,</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
</table>

CIS **Keyword and Parameter syntax**	C O N V E R S I O N	I N D E X I N G	S O R T I N G	R E S O U R C E S	S E L E C T I O N	S E G M E N T A T I O N	E N R I C H M E N T	E X T R A C T I O N
[*lineType*] , [*lineWidth*] , [*lineColor*] , [*copyNumber*] , [[-]*copyOffsetX*] , [[-]*copyOffsetY*] , [*gocaPattern*] , [*fillColor*])								
DEFINECIRCLE ('name' , [*unit*] , *posX* , *posY* , *radius* , [*lineType*] , [*lineWidth*] , [*lineColor*] , [*copyNumber*] , [[-]*copyOffsetX*] , [[-]*copyOffsetY*], [*gocaPattern*] , [*fillColor*])							X	
DEFINECOLOR (*name* , *colorModel* , *colParm1* , *[-]colParm2* , *[-]colParm3* , *colParm4*)							X	
DEFINECONTAINER ('name' , [*unit*] , [[-]*posX*] , [[-]*posY*], [*areaWidth*], [*areaHeight*], [*map*] , [[-]*posXInObjArea*] , [[-]*posYInObjArea*] , [*orientation*] , [*backgroundColor*], *type* , [*mode*] , *fileName* | *"fileName"*)							X	
DEFINECURVE ('name' , [*unit*] , *startPosX* , *startPosY* , *endPosX* , *endPosY* [, *endPosX* , *endPosY* [,...,...]] , [*lineType*] , [*lineWidth*] , [*lineColor*] , [*copyNumber*] , [[-]*copyOffsetX*] , [[-]*copyOffsetY*] , [*gocaPattern*] , [*fillColor*])							X	
DEFINEELLIPSE ('name' , [*unit*] , *posX* , *posY* , *[-]ax1PosX* , *[-]ax1PosY,* *[-]ax2PosX* , *[-]ax2PosY* , [*lineType*] , [*lineWidth*] , [*lineColor*] , [*copyNumber*] ,[[-]*copyOffsetX*] , [[-]*copyOffsetY*], [*gocaPattern*] , [*fillColor*])							X	
DEFINEFINISHINGOP ('name' , [*unit*] , *type* , *parameter* , *reference* , [*count*] , [*offset*] , [*position* [, ...]])							X	
DEFINELEVEL (*name*, *entity* [, *location*, *entitySelector*])							X	
DEFINELINE ('name' , [*unit*] , *startPosX* , *startPosY* , *endPosX* , *endPosY* [, *endPosX* , *endPosY* [,...,...]] , [*lineType*] , [*lineWidth*] , [*lineColor*] , [*copyNumber*] ,[[-]*copyOffsetX*] , [[-]*copyOffsetY*] , [*gocaPattern*] , [*fillColor*])							X	
DEFINEMEDIUMMAP ('*mMapName*' , [CONSTANT (NO | FRONT | BACK | BOTH)] [COPIES (1 | *n*)] [DUPLEX (NO | SIMPLEX | NORMAL | TUMBLE)] [EXTENTX (*n* [.*m*] [*unit*])] [EXTENTY (*n* [.*m*] [*unit*])] [INBIN (*n*)] [JOG (YES | NO)]							X	

CIS
Keyword and Parameter syntax

Keyword and Parameter syntax	CONVERSION	INDEXING	SORTING	RESOURCES	SELECTION	SEGMENTATION	ENRICHMENT	EXTRACTION
[MEDIAATTRIBUTES (['*UP3ImediaName*'] , [*UP3ImediaFrontCoating* **;** *UP3ImediaBackCoating*], [*UP3ImediaBrightness*] , [*UP3ImediaColorResourceID*] , [*UP3ImediaImagableSide*] , [*UP3ImediaColorName*] , [*UP3ImediaSetCount*] , [*UP3ImediaOpacity*] , [*UP3ImediaPreprinted*] , [*UP3ImediaRecycled*] , [*UP3ImediaRollDiameter*] , [*UP3ImediaThickness*] , [*UP3ImediaType*] , [*UP3ImediaWeight*] , [*UP3IpinHole*] , [*UP3ImediaWidth* **;** *UP3ImediaLength*] , [*UP3ImediaOrderedSetPiece*])] [MEDIATYPE (*mediaType*)] [MEDIANAME ('*mediaName*')] [SHEETLETS (1 | 2 [, LEFT | RIGHT])] [OFFSETXB ([-] *n* [.*m*] [*unit*])] [OFFSETXF ([-] *n* [.*m*] [*unit*])] [OFFSETYB ([-] *n* [.*m*] [*unit*])] [OFFSETYF ([-] *n* [.*m*] [*unit*])] [OUTBIN (*n*)] [OVERLAYB (*overlayName* [, ...])] [OVERLAYF (*overlayName* [, ...])] [PRESENTATION (PORTRAIT | LANDSCAPE | PORTRAIT90 | LANDSCAPE90 | PORTRAIT180 | LANDSCAPE180)] [SUPPRESSIONID (*n* [, ...])])								
DEFINEPTOCA (*'name'* , [*unit*] , *posX* , *posY* , [*orientation*] , [*color*] , [*font*] , [*codePageId*])							X	
DUPLEX (ASIS | NO | SIMPLEX | NORMAL | TUMBLE)	X							

CIS
Keyword and Parameter syntax

Keyword and Parameter syntax	CONVERSION	INDEXING	SORTING	RESOURCES	SELECTION	SEGMENTATION	ENRICHMENT	EXTRACTION
EXTRACTCP (*codePageId*)								X
EXTRACTINDEX (*fileName* **|** *"fileName", level, '_sprintfString', fieldName* [*, fieldName*] [, ...])								X
FDEFLIB (*directory* **|** "*directory*" [,*directory* **|** "*directory*"] [,...])				X				
FIELD (*fieldName*, { *record*, *column*, *length* } **|** { '*literal value*' [, <u>LIT</u>] } **|** { x'*literal value*' [, <u>LIT</u>] } **|** { _*variableName* [, <u>VAR</u>] } **|** { '*attributeName*', ATT } **|** { x'*attributeName*', ATT } **|** { 'searchString' | x'searchString', offset, length })		X					X	
FINISHINGFIDELITY (<u>ASIS</u> | NONE | STOP | CONTINUE, <u>YES</u> | NO)							X	
FONTCHARSETEXT (<u>""</u> | *"extension" [,"extension"] [,...]*)				X				
FONTFIDELITY (<u>ASIS</u> | NONE | STOP | CONTINUE)							X	
FONTLIB (*directory* **|** "*directory*" [,*directory* **|** "*directory*"] [,...])				X				
FONTMAP (*fontMapName* **|** <u>DUMMY</u> [, <u>RESOLVE</u> **|** ASIS])				X				
FORMDEF (*formDefName* **|** <u>DUMMY</u>)				X				
FORMDEFEXT (<u>""</u> | *"extension" [,"extension"] [,...]*)				X				
GROUPNAME (*indexName*)		X						
IMAGEOUT (ASIS **|** <u>IOCA</u> | IOCANOR)	X							
INBIN (*value*)	X							
INDEX (*indexName , triggerName* , '*attributeName*' **|** x'*attributeName*' , *fieldName* [, *fieldName*] [,...])		X						
INDEXCP (*codePageId* **|** UNKNOWN)		X						
INDEXDD (<u>CISIDX</u> **|** *fileName* **|** "*fileName*")		X						
INDEXEXIT ('*exitName*')		I						
INDEXOBJ (GROUP | ALL | <u>NONE</u>)		X						
INDEXSTARTBY (<u>1</u> | *value*)		X						
INITVARIABLES (_*variableName* = *initialValue* [, [-] *increment*]							X	

CIS
Keyword and Parameter syntax

<table>
<tr><th></th><th>CONVERSION</th><th>INDEXING</th><th>SORTING</th><th>RESOURCES</th><th>SELECTION</th><th>SEGMENTATION</th><th>ENRICHMENT</th><th>EXTRACTION</th></tr>
<tr><td>[, _variableName = initialValue [, [-] increment]] [, ...])</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>INLINERESOURCECACHING (ON | <u>OFF</u>)</td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td></tr>
<tr><td>INPEXIT ('exitName') Linux
('exitName' | exitName) OS/390, z/OS</td><td></td><td>I</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>INPUTCP (codePageId | UTF16LE | UTF16BE | <u>UNKNOWN</u>)</td><td></td><td>X</td><td>X</td><td></td><td></td><td></td><td></td><td>X</td></tr>
<tr><td>INPUTDD (<u>CISIN</u> | fileName | "fileName" [, fileName | "fileName"] [, ...])</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>INPUTFORMAT (MODCA | RDW | AFPSTREAM, x'delimiter' | FIXED, size |
REC_PREFIX, prefixLength, lengthPosition, [INCL, <u>EXCL</u>]))</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>INSERTBCOCA ('name', level, '_sprintfString' [, fieldName] [, ...])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>INSERTBOX ('name' , level)</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>INSERTCIRCLE ('name' , level)</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>INSERTCONTAINER ('name', level [, USEREXIT, 'exitName'])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>INSERTCURVE ('name' , level)</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>INSERTELLIPSE ('name' , level)</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>INSERTFINISHINGOP ('name' [, 'name' [, ...]] , level
[, USEREXIT, 'exitName'])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>INSERTINDEX ('attributeName' | x'attributeName' , level, '_sprintfString' [,
fieldName] [, ...])</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>INSERTLINE ('name' , level)</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>INSERTPTOCA ('name', level, '_sprintfString' [, fieldName] [, ...])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>INSERTPTOCANOP ('name', level, '_sprintfString' [, fieldName] [, ...])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>INSERTSHEET ('sheetName' , level , 'mMapName')</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>JOBID (<u>0</u> | value) PRISMAproduction Server internal only !!!</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>LINEMERGE (<u>NO</u> | 3800)</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>LSKEYDD (<u>CISLSK</u>)</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>LUPUB (<u>0</u> | value)</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>MEDIAFIDELITY (<u>ASIS</u> | NONE | STOP | CONTINUE)</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr>
</table>

<table>
<tr><th>CIS
Keyword and Parameter syntax</th><th>CONVERSION</th><th>INDEXING</th><th>SORTING</th><th>RESOURCES</th><th>SELECTION</th><th>SEGMENTATION</th><th>ENRICHMENT</th><th>EXTRACTION</th></tr>
<tr><td>MESSAGECP (codePageId)</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>MPCONCEPT (ON | OFF)</td><td></td><td></td><td>X</td><td></td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>MPCONSOLIDATION (NO |
{ NESTED , 'attributeName' | x'attributeName' [, 'attributeName' |
x'attributeName' [, ...]] } |
{ INONEGROUP , 'attributeName' | x'attributeName' [,
'attributeName' | x'attributeName' [, ...]] })</td><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>OBJCONEXT ("" | "extension" [,"extension"] [,...])</td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td></tr>
<tr><td>OBJCONLIB (directory | "directory" [,directory | "directory"] [,...])</td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td></tr>
<tr><td>OFFSETXB ([-] n [.m] [unit])</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>OFFSETXF ([-] n [.m] [unit])</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>OFFSETYB ([-] n [.m] [unit])</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>OFFSETYF ([-] n [.m] [unit])</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>OUTBIN (value)</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>OUTLINEFONTEXT ("" | "extension" [,"extension"] [,...])</td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td></tr>
<tr><td>OUTPUTDD (CISOUT | fileName)</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>OUTPUTFORMAT (MODCA | RDW | AFPSTREAM, x'delimiter')</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>OUTPUTSEG ({ entity [, count [, boundary]] })</td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td></tr>
<tr><td>OUTPUTSEGPSDEF (spaceType, primary [, secondary [, volSer]])</td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td></tr>
<tr><td>OVERLAYB (overlayName [,overlayName] [,...])</td><td></td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>OVERLAYEXT ("" | "extension" [,"extension"] [,...])</td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td></tr>
<tr><td>OVERLAYF (overlayName [,overlayName] [,...])</td><td></td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td></tr>
<tr><td>OVLYLIB (directory | "directory" [,directory | "directory"] [,...])</td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td></tr>
<tr><td>PAGEDEF (pageDefName | DUMMY)</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>PAGEDEFEXT ("" | "extension" [,"extension"] [,...])</td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td></tr>
<tr><td>PAGESEGEXT ("" | "extension" [,"extension"] [,...])</td><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td></tr>
<tr><td>PDEFLIB (directory | "directory" [,directory | "directory"] [,...])</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>PRESENTATION (ASIS | PORTRAIT | PORTRAIT90 | PORTRAIT180 |</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

CIS
Keyword and Parameter syntax

<table>
<tr><th>Keyword and Parameter syntax</th><th>CONVERSION</th><th>INDEXING</th><th>SORTING</th><th>RESOURCES</th><th>SELECTION</th><th>SEGMENTATION</th><th>ENRICHMENT</th><th>EXTRACTION</th></tr>
<tr><td>LANDSCAPE | LANDSCAPE90 | LANDSCAPE180)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>PRINTMODE (SOSI1 | SOSI2 | SOSI3 | string)</td><td>x</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>PSEGLIB (directory | "directory" [,directory | "directory"] [,...])</td><td></td><td></td><td></td><td>x</td><td></td><td></td><td></td><td></td></tr>
<tr><td>REMOVEBCOCA ('name' [, 'name'] [, ...])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>x</td><td></td></tr>
<tr><td>REMOVEBOX ('name' [, 'name' [, ...]])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>x</td><td></td></tr>
<tr><td>REMOVECIRCLE ('name' [, 'name' [, ...]])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>x</td><td></td></tr>
<tr><td>REMOVECONTAINER ('name' [, 'name' [, ...]])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>x</td><td></td></tr>
<tr><td>REMOVECURVE ('name' [, 'name' [, ...]])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>x</td><td></td></tr>
<tr><td>REMOVEELLIPSE ('name' [, 'name' [, ...]])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>x</td><td></td></tr>
<tr><td>REMOVEFINISHINGOP ('name' [, 'name' [, ...]])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>x</td><td></td></tr>
<tr><td>REMOVEINDEX ('attributeName' | x'attributeName' [, 'attributeName' | x'attributeName'] [, ...])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>x</td><td></td></tr>
<tr><td>REMOVELINE ('name' [, 'name' [, ...]])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>x</td><td></td></tr>
<tr><td>REMOVEMEDIUMMAP ('mMapName' [, 'mMapName' [, ...]])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>x</td><td></td></tr>
<tr><td>REMOVEPTOCA ('name' [, 'name'] [, ...])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>x</td><td></td></tr>
<tr><td>REMOVESHEET ('name' [, 'name' [, ...]])</td><td></td><td></td><td></td><td></td><td></td><td></td><td>x</td><td></td></tr>
<tr><td>RESCONSOLIDATION (<u>BASIC</u> | { [EXTERNALRESGROUP]
[, INTERNALRESGROUP]
[, INTERNALMEDIUMMAP] [, INTERNALMMAPOPTIMIZE]})</td><td></td><td></td><td></td><td>x</td><td></td><td></td><td></td><td></td></tr>
<tr><td>RESEXIT ('exitName')</td><td></td><td></td><td></td><td>x</td><td></td><td></td><td></td><td></td></tr>
<tr><td>RESFIDELITY (<u>STOP</u> | CONTINUE)</td><td></td><td></td><td></td><td>x</td><td></td><td></td><td></td><td></td></tr>
<tr><td>RESFILE (<u>SEQ</u> | PDS)</td><td></td><td></td><td></td><td>x</td><td></td><td></td><td></td><td></td></tr>
<tr><td>RESOBJDD (<u>CISRES</u> | fileName | "fileName" | directory | "directory")</td><td></td><td></td><td></td><td>x</td><td></td><td></td><td></td><td></td></tr>
<tr><td>RESPRUNING (<u>ALL</u> | NONE | OVLY)</td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td></td><td></td></tr>
<tr><td>RESTYPE (<u>NONE</u> | { ALL [, PDEF] } | { [FDEF] [, PDEF]
[, PSEG] [, OVLY] [, FONT][, OBJCON] [, BCOCA] [, GOCA] [, IOCA] })</td><td></td><td></td><td></td><td>x</td><td></td><td></td><td></td><td></td></tr>
<tr><td>SELINDEXRANGE (entity, 'attributeName' | x'attributeName',</td><td></td><td></td><td></td><td></td><td>x</td><td></td><td></td><td></td></tr>
</table>

CIS
Keyword and Parameter syntax

<table>
<tr><th></th><th>CONVERSION</th><th>INDEXING</th><th>SORTING</th><th>RESOURCES</th><th>SELECTION</th><th>SEGMENTATION</th><th>ENRICHMENT</th><th>EXTRACTION</th></tr>
<tr><td>[!]'from'-'to' | [!]x'from'-x'to' [, [!]'from'-'to' | [!]x'from'-x'to'] [, ...])</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>SELRANGE (entity, [!]from-to [, [!]from-to] [, ...])</td><td></td><td></td><td></td><td></td><td>x</td><td></td><td></td><td></td></tr>
<tr><td>SELRANGERANDOM (entity, [!]from-to, amount [, seed])</td><td></td><td></td><td></td><td></td><td>x</td><td></td><td></td><td></td></tr>
<tr><td>SELSIZE (entity, operator, size, unit [, operator, size])</td><td></td><td></td><td></td><td></td><td>x</td><td></td><td></td><td></td></tr>
<tr><td>SETUPCHARS (<i>fontName</i> [<<i>mxm</i>>] [, <i>fontName</i> [<<i>mxm</i>>]] [,...])
Restricted for usage in combinaion with MVS ROUTER</td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td></td><td></td></tr>
<tr><td>SETUPFORMDEF (formDefName | <u>DUMMY</u>)
Restricted for usage in combinaion with MVS ROUTER</td><td></td><td></td><td></td><td>x</td><td></td><td></td><td></td><td></td></tr>
<tr><td>SETUPPAGEDEF (pageDefName | <u>DUMMY</u>)
Restricted for usage in combinaion with MVS ROUTER</td><td>x</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>SETUPVID (<u>ASIS</u> | NONE | { n [, n [, ...]] })</td><td></td><td></td><td></td><td></td><td></td><td></td><td>x</td><td></td></tr>
<tr><td>SORT (<u>NONE</u> | SCHEME_1 | SCHEME_2 | SCHEME_3 |
4UP_BOOKLET | 2UP_BOOKLET | 2UP_MP_BOOKLET |
FLIP_FOR_COLLATED | FLIP_FOR_UNCOLLATED |
FOR_UNCOLLATED | USEREXIT, 'program name' |
ASC, 'attributeName' | x'attributeName' [,'attributeName' |
x'attributeName'] [,...] |
DES, 'attributeName' | x'attributeName' [,'attributeName' |
x'attributeName'] [,...])</td><td></td><td></td><td>x</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>TMPDIR (path | "path")</td><td></td><td></td><td>x</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>TONERFIDELITY (<u>ASIS</u> | NONE | n)</td><td></td><td></td><td></td><td></td><td></td><td></td><td>x</td><td></td></tr>
<tr><td>TRACEDD (<u>CISTRA1</u> | fileName | "fileName"
[, <u>CISTRA2</u> | fileName | "fileName"])</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>TRACELEVEL (<u>0</u> | value [, value] [, ...])</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td></tr>
<tr><td>TRACESIZE (<u>100</u> | size)</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td></tr>
<tr><td>TRACETYPE (<u>ALL</u> | { [,...])</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td></tr>
<tr><td>TRCTYPE (<u>NO</u> | IBM | SNI)</td><td>x</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>TRIGGER (triggerName , *, column | * , ‘value’ | x‘value’
[, record, column | * , ‘value’ | x‘value’] [,...])</td><td></td><td>x</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

CIS
Keyword and Parameter syntax

UNIQUEBNGS (YES | NO)

USEPAGENAMES (YES | NO)

USERLIB (*directory* **|** "*directory*" [,*directory* **|** "*directory*"] [,...])

WORKDD (CISSWAP)

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

Keyword	CONVERSION	INDEXING	SORTING	RESOURCES	SELECTION	SEGMENTATION	ENRICHMENT	EXTRACTION
UNIQUEBNGS		X						
USEPAGENAMES		X						
USERLIB				X				
WORKDD			X		X			
X2UP	X							

CIS
Keyword and Parameter syntax
for params resticted to use in combination with ROUTER

SETUPCHARS (*fontName* [<mxm>] [, *fontName* [<mxm>]] [,...])

SETUPFORMDEF (*formDefName* **|** DUMMY)

SETUPPAGEDEF (*pageDefName* **|** DUMMY)

Keyword	CONVERSION	INDEXING	SORTING	RESOURCES	SELECTION	SEGMENTATION	ENRICHMENT	EXTRACTION
SETUPCHARS								
SETUPFORMDEF								
SETUPPAGEDEF								

ANALYST **(XML | <u>NONE</u>)**

Activates the Print Data Stream Analyzer function and controls the summary report format.

XML	Indicates that the summary report should be created in XML format.
<u>NONE</u>	Deactivates the Print Stream Analyser function.

The Print file Analyst function is described under *3.13 How to Analyze a print file* on page 141. A description of the summary report created with this keyword is available under *6.3 Appendix C: Description of the Analyst tags* on page 346.

ANALYSTCP **(*codePageID*)**

Specifies the Code Page to be used during the generation of the output file of the Print Data Stream Analyzer. A description of this function is available under *3.13 How to Analyze a print file* on page 141.

codePageID	Specifies the ID of the code Page to be used for translating . See also *3.2.1 Supported Code Page IDs* on page 38. By default, when this keyword is not specified, CIS uses the currently active internal Code Page. For more information please refer to *3.2 Code page support* on page 33.

ANALYSTDD **(<u>CISANA</u> | *fileName* | "*fileName*")**

Specifies the file name to be used for storing the output from the Print Data Stream Analyzer. A description of this function is available under.*3.13 How to Analyze a print file* on page 141.

fileName	is the file name where the information collected by the CIS Analyzer will be written.
"fileName"	This notation form must be used when the file name contains special characters such as blanks.

The Print file Analyst function is described under *3.13 How to Analyze a print file* on page 141. A description of the summary report created with this keyword is available under *6.3 Appendix C: Description of the Analyst tags* on page 346.

CASECTRL **(0, <u>1</u>, <u>2</u>, 3)**

Controls the spelling of resource names and extensions before searching the file in the attached resource libraries.

0 read the resource name and the extensions in lower case

1 read the resource name and the extensions in upper case
default value on host

2 do not convert the spelling of the resource name and the extensions
default value on server

3 at first the resource name and the extensions are read in upper case and afterwards in lower case (combination of "0" and "1")

Please note that directory names (specified with the library parameters USERLIB, FDEFLIB, FONTLIB, etc.) are always handled without converting the spelling.

CC **(<u>NO</u> | SNI | ASA | IBM | ASAA | SS | DS | TS)**

Specifies the type of printer control characters to be used.

<u>NO</u>	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

CHARS **(*fontName* [<mxm>] [, *fontName* [<mxm>]] [,...])**

Specifies the member name of the coded font (s) to be used to process an S/370 Line format or AFP Mixed data file. It is ignored for MO:DCA-P files. The matrix memory position <mxm> may be specified too. This list overrides any fonts specified in the Pagedef.

fontName A 1 to 8 character coded font name(s) including the font prefix (e.g. 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 be allowed to 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.

CODEDFONTEXT **("" | "*extension*" [, "*extension*"] [,...])**

Specifies one or more suffixes to be appended to an AFP Coded Font resource name while searching for the corresponding file.

extension A variable length character string starting with a dot. The default is no suffix.

CIS searches in the same order as the extensions are given. The first match is used.

Example: CODEDFONTEXT (".600", ".300", ".240", ".afp", "")

The above example causes CIS to search for the following resource names:

*name***.600**
*name***.300**
*name***.240**
*name***.afp**
name

CODEPAGEEXT **(<u>""</u> | "*extension*" [, "*extension*"] [,...])**

Specifies one or more suffixes to be appended to an AFP Code Page resource name while searching for the corresponding file.

extension A variable length character string starting with a dot. The default is no suffix.

CIS searches in the same order as the extensions are given. The first match is used.

Example: CODEPAGEEXT (“.600”, “.300”, “.240”, “.afp”, “”)

The above example causes CIS to search for the following resource names:

*name***.600**
*name***.300**
*name***.240**
*name***.afp**
name

COLORFIDELITY **({<u>ASIS</u> | NONE | STOP | CONTINUE} , {<u>YES</u> | NO})**

Controls the way the presentation software (e.g. SPS) will handle Color exceptions reported by the printer. Not all printers support Color Fidelity. It causes CIS to insert a PFC structured field containing a Color Fidelity triplet x'75' in all Medium Maps in the form definition being used.

This keyword must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise a warning message is issued.

<u>ASIS</u> Indicates that the current Color fidelity options in the FORMDEF are to be left unchanged. This is the default.

NONE Causes CIS to remove all Color fidelity (triplet x'75') options from the FORMDEF.

STOP Causes SPS to stop printing the job and to place it in hold in case the printer reports a media exception.

CONTINUE Causes SPS to issue a message and to continue printing the job using a substitute media.

<u>YES</u> Report color exceptions that do not stop presentation.

NO — Do not report color exceptions that do not stop presentation.

Example: COLORFIDELITY (ASIS, NO)

COLORMAP **(*colorMapName* | <u>DUMMY</u>)**

Specifies the name of a Color Mapping table.

colorMapName

On host only: the name is restricted to one to eight alphanumeric characters, including the two-character prefix, if there is one.

If DUMMY (the default) is specified, CIS uses the first Color Mapping table found and ignores all others.

COMPRESS **(<u>NO</u> | CHARS |**
{ STRING, *'string'* [, *'string'*] [, ...] } |
{ ALL , *'string'* [, *'string'*] [, ...] })

Activates PTOCA character and string compression. String compression works as an optimisation feature that reduces the size of the output document file by combining sequence of characters in the input data into a single Repeat String (RPS) control sequence. A description of this function is available under *3.6.8 Compressing AFP and MO:DCA-P data*. In case of AFP data, CIS propagates all No-Operation (NOP, 0xD3EEEE) structured fields which are part of the page data (between BPG and EPG structured fields) and writes them immediately after the MO:DCA-P Active Environment Group in the output file. NOP's which are not part of the page data (outside BPG/EPG) are not propagated.

<u>NO</u> — No compression is done.

CHARS — Any character which is repeated more than 6 times consecutively is replaced with an RPS control sequence. All characters in a Presentation Text Data (PTX) structured field are checked.

STRING — All characters in a Presentation Text Data (PTX) structured field are compared against the specified string(s). Matching characters are replaced with an RPS control sequence.

ALL — Is the combination of the CHARS and STRING parameters, together. Provides the highest level of AFP compression.

Example: COMPRESS (STRING, ' ', ' *', X'4243')

DEFINE2DDATAMATRIX **('name', [escape] , [convert] , [rowSize] , [rowNumber] , [sequenceIndicator] , [totalSymbols] , [fileId1] , [fileId2] , [functionFlags] [,...])**

Defines the characteristics of a 2D Data Matrix Bar Code object to be inserted by CIS in form of a BCOCA AFP structure. More information about the data enrichment functionality in CIS can be found under *3.8 How to Enrich Print Data streams* on page 77.

'name' Corresponds to the name of a BCOCA object previously defined using the DEFINEBCOCA keyword.

escape Describes the escape-sequence handling
- NO X'5C' (Backslash) will be handled as normal character.
- <u>YES</u> X'5C' (Backslash) will be handled as an escape character.

convert Describes, if a translation from EBCDIC-to-ASCII will be done
- <u>NO</u> No conversion will be done
- YES A conversion from the given EBCDIC CodePage to ASCII will be done.

rowSize Defines the matrix row size, as allowed by symbology, the range is from 1 – 65535, default is 0.

rowNumber Defines the number of rows, as allowed by symbology, the range is from 1 – 65535, default is 0.

sequenceIndicator Structured append sequence indicator, a value from 0 – 16 is allowed, default is 0.

totalSymbols Total number of structured append symbols, value 0 or 2 –16 is allowed, default is 0.

fileId1 High-order byte of a 2-byte unique file identification for a set of structured append symbols, a value from 1 – 254 is allowed. Default is 1.

fileId2 Low-order byte of a 2-byte unique file identification for a set of structured append symbols, a value from 1 – 254 is allowed. Default is 1.

functionFlags FNC1UCC | FN1IND | RDRPROG | MAC5 | MAC6

Describes different possibilities, for alternate data type identifier reader programming symbol header and trailer instructions to the bar code reader. Per default no function flag is set.

FNC1UCC	Symbol confirms to UCC/EAN standards
FN1IND	Symbol confirm to industry standards
RDRPROG	If given, this symbol encodes a message used to program the reader system.
MAC5	Use the 05 Macro header/trailer
MAC6	Use the 06 Macro header/trailer

For further details please refer to BCOCA Reference, Data Matrix Special-Function Parameters.

DEFINE2DMAXICODE **('name', [escape] , [convert] , mode, [sequenceIndicator] , [totalSymbols] , [zipper])**

Defines the characteristics of a 2D MaxiCode Bar Code object to be inserted by CIS in form of a BCOCA AFP structure. More information about the data enrichment functionality in CIS can be found under *3.8 How to Enrich Print Data streams* on page 77.

'name' Corresponds to the name of a BCOCA object previously defined using the DEFINEBCOCA keyword.

escape Describes the escape-sequence handling

NO	X'5C' (Backslash) will be handled as normal character.
<u>YES</u>	X'5C' (Backslash) will be handled as an escape character.

convert Describes, if a translation from EBCDIC-to-ASCII will be done

<u>NO</u>	No conversion will be done

YES A conversion from the given EBCDIC CodePage to ASCII will be done.

mode Defines the symbol mode for the default character encoding
MODE2 | MODE3 | MODE4 | MODE5 | MODE6
MODE2 numeric postal code
MODE3 alphanumeric postal code
MODE4 standard symbol
MODE5 full ECC symbol
MODE6 reader program SEC
The *mode* parameter must be specified. No default is applied automatically.

sequenceIndicator Defines structured append sequence indicator, a value from 0 – 8 is allowed, default is 0.

totalSymbols Defines the total number of structured-append symbols, value 0 or 2 – 8 is allowed, default is 0.

zipper Defines special functions which can be used with a MaxiCode symbol.

ZIPPER A vertical zipper-like test pattern and a contrast block is printed to the right of the symbol.

<u>NOZIPPER</u> Zipper is off

For further details please refer to BCOCA Reference, MaxiCode Special-Function Parameters.

DEFINE2DPDF417 **(‘name’, [escape] , [convert] , dataSymbolsNumber, [rows] , [securityLevel] ,[controlBlockData])**

Defines the characteristics of a 2D PDF417 Bar Code object to be inserted by CIS in form of a BCOCA AFP structure. More information about the data enrichment functionality in CIS can be found under *3.8 How to Enrich Print Data streams* on page 77.

‘name’ Corresponds to the name of a BCOCA object previously defined using the DEFINEBCOCA keyword.

escape Describes the escape-sequence handling
NO X’5C’ (Backslash) will be handled as normal character.

YES X'5C' (Backslash) will be handled as an escape character.

convert Describes, if a translation from EBCDIC-to-ASCII will be done
NO No conversion will be done
YES A conversion from the given EBCDIC CodePage to ASCII will be done.

dataSymbolsNumber Defines the number of data symbol characters per row, a value from 1 – 30 is allowed.

rows Defines the desired number of rows in the bar code symbol, a value from 3 – 90 is allowed, default is 255.

securityLevel Defines the security level for the symbol, a value from 0 – 8 is allowed, default is 0.

controlBlockData Macro PDF417 control block data.

For further details please refer to BCOCA Reference, PDF417 Special-Function Parameters.

DEFINE2DQRCODE **('name', [escape] , [convert] , [codePageId] , [symbolVersion] , [sequenceIndicator] , [totalSymbols] , [parityData] , [ErrorCorrectionLevel] , [functionFlags] , ['applicationIndicator'])**

'name' Corresponds to the name of a BCOCA object previously defined using the DEFINEBCOCA keyword.

escape Describes the escape-sequence handling
NO X'5C' (Backslash) will be handled as normal character.
YES X'5C' (Backslash) will be handled as an escape character.

convert Describes, if a translation from EBCDIC-to-ASCII will be done
NO No conversion will be done
YES A conversion from the given EBCDIC CodePage to ASCII will be done.

codePageId CodePage which will be used for conversion, possible values:
IBM_0290 | IBM_0500 | IBM_1027
The *codePageId* parameter must be specified. No default

is applied automatically.

symbolVersion	Defines the desired size of the symbol, a value from 0 – 40 is allowed, default is 0.
sequenceIndicator	Defines the structured append sequence indicator, a value from 0 – 16 is allowed, default is 0.
totalSymbols	Defines the total number of structured-append symbols, a value from 0, 2 – 16 is allowed, default is 0.
parityData	Defines the structured append parity data, a value from 0 – 255 is allowed, default is 0.
errorCorrectionLevel	Defines the level of error correction, a value from 0 – 3 is allowed, default is 0.
functionFlags	Defines the alternate type identifier. Per default no function flag is set. FNC1UCC Symbol conforms to UCC/EAN standards. FNC1IND Symbol conforms to industry standards.
'applicationIndicator'	Defines the application indicator for industry FNC1, the value is either an alphabetic value (a-z, A-Z) plus 100, or a two digit decimal number (00 – 99). If FNC1IND is not given, the default 0 is set.

For further details please refer to BCOCA Reference, 2D QRCode Special-Function Parameters.

DEFINEBCOCA **(*'name',* [*unit*] , *posX* , *posY* , [*orientation*] , [*color*] , [*type*] , [*modifier*] , [*hri*] , [*asterisk*] , [*height*] , [*modWidth*] , [*ratio*] , [*codePageID*])**

Defines the characteristics of a Bar Code object to be inserted by CIS in form of a BCOCA AFP structure. More information about the data enrichment functionality in CIS can be found under *3.8 How to Enrich Print Data streams* on page 77.

'name' Is a 1 to 250 characters name which is used to uniquely identify the Bar Code object being defined. It is allowed to use a name which is already being used to identify

BCOCA objects within a print file however, it will not be possible to differentiate between the "old" and the "new" bar codes after that.

Bar Code objects inserted by CIS include a Begin Bar Code Object (BBC) structured field with a “Fully Qualified Name” triplet (0x02) specifying the *name* parameter as fully qualified name.

unit Any of the units of measurement supported by CIS (see *4.4.6 Units of measurement* on page 165 for more information). PELS is the default.

posX Specifies the offset in the X direction, relative to the Medium Origin, in the units specified in *unit*, where the Bar Code is to be printed. Printing a Bar Code outside the currently defined page is not allowed. See *3.9* on page 109 for more information.

posY Specifies the offset in the Y direction, relative to the Medium Origin, in the units specified in *unit*, where the Bar Code is to be printed. Printing a Bar Code outside the currently defined page is not allowed. See *3.9* on page 109 for more information.

orientation Specifies the orientation of the bar code, relative to the Medium Origin. Possible values are 0°, 90°, 180° and 270°. The default is 0°.

color Specifies the color of the bars. Please refer to the DEFINECOLOR keyword for a description of the allowed values. The default is BLACK.

type Specifies the type of Bar code to be inserted. It may be specified as a number (between 0 and 255) or as one of the names listed below. The corresponding type and the default values for the *modifier* and *codePageID* are listed too.

Name	**type**	**modifier**	**codePageID**
CODE39	1	1	IBM_0500
MSI	2	1	IBM_0500
UPCA	3	0	IBM_0893
UPCE	5	0	IBM_0893
UPC2SUPP	6	0	IBM_0893
UPC5SUPP	7	0	IBM_0893
EAN8	8	0	IBM_0893
EAN13	9	0	IBM_0893

IND2OF5	10	1	IBM_0500
MAT2OF5	11	1	IBM_0500
ITL2OF5	12	1	IBM_0500
CDB2OF7	13	1	IBM_0500
CODE128	17	2	IBM_1303
EAN2SUP	22	0	IBM_0893
EAN5SUP	23	0	IBM_0893
POSTNET	24	0	IBM_0500
RM4SCC	26	0	IBM_0500
JPOSTAL	27	0	IBM_0500
DATAMAT	28	0	IBM_0500
MAXICODE	29	0	IBM_0500
PDF417	30	0	IBM_0500
APOSTAL	31	1	IBM_0500
QRCODE	32	2	IBM_0500
CODE93	33	0	IBM_0500
OCE_QRCODE	0xEC	0	IBM_0500
OCE_MAXICOD	0xED	0	IBM_1303
OCE_DATAMAT	0xEE	0	IBM_1303
OCE_PDF417	0xEF	0	IBM_1303

modifier

Specifies additional processing information about the Bar Code symbol to be generated (e.g. whether a check-digit is to be generated). The allowed range for this value depends on the Bar Code Type. Please refer to *Data Stream and Object Architectures: Bar Code Object Content Architecture Reference,* for more information about this parameter. The default value depends on the type parameter. The *modifier* parameter must be specified when an undefined numeric *type* is specified.

hri

Controls the placement of Human Readable Interpretation for the Bar code inserted. The allowed options are:

<u>ON</u>	Place HRI on its default location. The default location may differ between Bar code types.
OFF	Suppress HRI printing.
ABOVE	Place HRI above the Bar code symbol.
BELOW	Place HRI below the Bar code symbol.

asterisk

Specifies whether an asterisk is to be generated as start and stop characters in the HRI for CODE39 Bar Codes. It is ignored for all other Bar code types.

<u>ON</u>	Indicates that start/stop asterisks should be generated in the HRI.
OFF	No asterisks are to be generated.

height Specifies the height of the Bar code symbol. For UPC and EAN Bar codes, the total height also includes the HRI characters. This field is ignored by the printer for Bar code types which explicitly define their height (e.g. POSTNET).
A value between 1 and 32767 may be specified and indicates the number of *units* to be used as height. The default value used is x'FFFF' which causes the printer to use its internal default.

modWidth Specifies the width of the smallest defined Bar code element. This field is ignored by the printer for Bar code types which explicitly define their module width (e.g. POSTNET).
A value between 1 and 254 may be specified and indicates the number of thousands of an inch (1/1000) to be used as module width. The default value used is x'FF' which causes the printer to use its internal default.

ratio Specifies the ratio between the width and the narrow Bar Code elements. This field is ignored by the printer for Bar Code types which explicitly define their module width (e.g. POSTNET).
A value between 100 and 500 may be specified and indicates a percentage value (e.g. a value of 250 represents a ratio of 2.5 to 1. The default value used is x'FFFF' which causes the printer to use its internal default.

codePageID Specifies the ID of the Code Page to be used for translating. See also *3.2.1 Supported Code Page IDs* on page 38. The default is to use the Code Page ID associated to the Bar Code type specified in *type*.

For more information about Bar Codes please refer to publication [12] under *8 Bibliography* on page 465.

DEFINEBOX **('name' , [*unit*] , *p1X* , *p1Y* , *p2X* , *p2Y* , [*rounded*] , [*lineType*] , [*lineWidth*] , [*lineColor*] , [*copyNumber*] ,[[-]*copyOffsetX*] , [[-]*copyOffsetY*] , [*gocaPattern*] , [*fillColor*])**

Defines the characteristics of a rectangular figure (box) to be inserted in form of a GOCA object. More information about this data enrichment functionality in CIS can be found under *3.8.5 Inserting and Removing Graphic Objects (GOCA)* on page 88.

'name' Is a 1 to 250 characters name which is used to uniquely identify the Box object being defined. It is allowed to use a name which is already being used to identify GOCA objects within a print file, but it will not be possible to differentiate between the "old" and the "new" box after that.

unit Any of the units of measurement supported by CIS (see *4.4.6 Units of measurement* on page 165 for more information). PELS is the default.

p1X Specifies the offset in the X direction, relative to the Medium Origin, in the units specified in *unit*, where the first corner is to be placed. A value between 0 and 2^{32}-1 is allowed. The position must be inside the current page presentation space.

p1Y Specifies the offset in the Y direction, relative to the Medium Origin, in the units specified in *unit*, where the first corner is to be placed. A value between 0 and 2^{32}-1 is allowed. The position must be inside the current page presentation space.

p2X Specifies the offset in the X direction, relative to the Medium Origin, in the units specified in *unit*, where the opposite corner is to be placed. A value between 0 and 2^{32}-1 is allowed. The position must be inside the current page presentation space.

p2Y Specifies the offset in the Y direction, relative to the Medium Origin, in the units specified in *unit*, where the opposite corner is to be placed. A value between 0 and 2^{32}-1 is allowed. The position must be inside the current page presentation space.

rounded Specifies the radius to be used for drawing the box corners. One of the following values is possible:

- NONE no rounded corners are to be drawn
- *n* a radius of *n* units is to be used. May not be longer than the length of the shortest box side divided by 2.
- *a/b* is multiplied to the length of the shortest box side and the result is used as radius. May not be greater than 1/2. The default is <u>1/8</u>.

lineType Specifies the type of GOCA line to be used for the figure's border line. The allowed values are:

0x01	dotted
0x02	short dashed
0x03	dash-dot
0x04	double dotted
0x05	long dashed
0x06	dash double-dot
0x07	solid
0x08	invisible

Please refer to "Line type" in *S544-5498-01 Graphics Object Content Architecture for Advanced Function Presentation Reference,* page 32 for more information.

lineWidth Specifies the width (thickness) of the figure's border line in units of 1/100 inches. A value between 1 and 255 is allowed. The default is 5.

lineColor Specifies the color of the figure's border line. Please refer to the DEFINECOLOR keyword for a description of the allowed values. The default is BLACK.

copyNumber Specifies the number of times the figure should be inserted. A value between 1 and 255 may be specified. The default value is 1.

[-]copyOffsetX

Specifies the offset in the X direction, relative to the previous figure's copy, where the next copy of the figure is to be placed. A value between $-(2^{31}-1)$ and $2^{31}-1$ is allowed. The resulting position must be inside the current page presentation space. This parameter is mandatory when *copyNumber* > 1.

[-]copyOffsetY

Specifies the offset in the Y direction, relative to the previous figure's copy, where the next copy of the figure is to be placed. A value between $-(2^{31}-1)$ and $2^{31}-1$ is allowed. The resulting position must be inside the current page presentation space. This parameter is mandatory when *copyNumber* > 1.

gocaPattern Specifies the pattern to be used to fill the figure. The allowed values are:

0x01-0x0E	Please refer to "patterns" in [14] (see *8 Bibliography* on page 465)
0x0F	no fill
0x10	solid pattern

fillColor Specifies the color which should be used to fill the closed figure. Please refer to the DEFINECOLOR keyword for a description of the allowed values. The default is BLACK.

DEFINECIRCLE ('name' , [*unit*] , *posX* , *posY* , *radius* , [*lineType*] , [*lineWidth*] , [*lineColor*] , [*copyNumber*] , [[-]*copyOffsetX*] , [[-]*copyOffsetY*] , [*gocaPattern*] , [*fillColor*])

Defines the characteristics of a circular figure to be inserted in form of a GOCA object. More information about this data enrichment functionality in CIS can be found under *3.8.5 Inserting and Removing Graphic Objects (GOCA)* on page 88

'name' Is a 1 to 250 characters name which is used to uniquely identify the Circle object being defined. It is allowed to use a name which is already being used to identify GOCA objects within a print file, but it will not be possible to differentiate between the "old" and the "new" circle after that.

unit Any of the units of measurement supported by CIS (see *4.4.6 Units of measurement* on page 165 for more information). PELS is the default.

posX Specifies the offset in the X direction, relative to the Medium Origin, in the units specified in *unit*, where the circle's center point is placed. A value between 0 and $2^{32}-1$ is allowed. The position must be inside the current page presentation space.

posY Specifies the offset in the Y direction, relative to the Medium Origin, in the units specified in *unit*, where the circle's center point is placed. A value between 0 and $2^{32}-1$ is allowed. The position must be inside the current page presentation space.

radius Specifies the radius of the circle, in the units specified in *unit*.

lineType Specifies the type of GOCA line to be used for the figure's border line. The allowed values are:

0x01	dotted
0x02	short dashed
0x03	dash-dot
0x04	double dotted
0x05	long dashed

0x06	dash double-dot
0x07	solid
0x08	invisible

Please refer to "Line type" in *S544-5498-01 Graphics Object Content Architecture for Advanced Function Presentation Reference,* page 32 for more information.

lineWidth Specifies the width (thickness) of the figure's border line in units of 1/100 inches. A value between 1 and 255 is allowed. The default is 5.

lineColor Specifies the color of the figure's border line. Please refer to the DEFINECOLOR keyword for a description of the allowed values. The default is BLACK.

copyNumber Specifies the number of times the figure should be inserted. A value between 1 and 255 may be specified. The default value is 1.

[-]copyOffsetX

Specifies the offset in the X direction, relative to the previous figure's copy, where the next copy of the figure is to be placed. A value between $-(2^{31}-1)$ and $2^{31}-1$ is allowed. The resulting position must be inside the current page presentation space. This parameter is mandatory when *copyNumber* > 1.

[-]copyOffsetY

Specifies the offset in the Y direction, relative to the previous figure's copy, where the next copy of the figure is to be placed. A value between $-(2^{31}-1)$ and $2^{31}-1$ is allowed. The resulting position must be inside the current page presentation space. This parameter is mandatory when *copyNumber* > 1.

gocaPattern Specifies the pattern to be used to fill the figure. The allowed values are:

0x01-0x0E	Please refer to "patterns" in [14] (see *8 Bibliography* on page 465)
0x0F	no fill
0x10	solid pattern

fillColor Specifies the color which should be used to fill the closed figure. Please refer to the DEFINECOLOR keyword for a description of the allowed values. The default is BLACK.

DEFINECOLOR (*name* , *colorModel* , *colParm1* , *[-]colParm2* , *[-]colParm3* , *[colParm4]*)

Assigns a user defined name to a MO:DCA color. Supports the RGB, CMYK , CIELAB and HIGHLIGHT color models. Color names defined with the DEFINECOLOR keyword may be used in the other CIS keywords where color names must be specified. Not all color models are supported by all MO:DCA objects. Refer to *4.4.4 Predefined color names* on page 164 for a description of the supported colors.

name Is a 1 to 250 characters name which is used to uniquely identify the color being defined for the duration of the current CIS run. The name is not saved in the data stream.

The name of the following OCA colors is already defined within CIS (refer to the MO:DCA reference for the corresponding RGB values of each of these colors) and **cannot** be redefined by the user:

BLUE	RED
MAGENTA or PINK	GREEN
CYAN or TURQ	YELLOW
BLACK	BROWN
MUSTARD	DARKBLUE
DARKGREEN	PURPLE
DARKTURQ or DARKCYAN	ORANGE
GRAY	GOLDENROD
WHITE	VIOLET
GOLD	SILVER
BUFF	IVORY
NONE	DEFAULT

colorModel one of the following values:

RGB RGB color model. The color value is specified with three components.
colParm1 represents a value for red,
colParm2 represents a value for green *colParm3* represents a value for blue.
colParm4 is ignored.

Each of the three integer values is specified as a positive percentage from 0 to 100.
An RGB specification of 0/0/0 is black.
An RGB specification of 100/100/100 is white.
Any other value is a color somewhere between black and white, depending on the output device.

CMYK Defines the cyan/magenta/yellow/black color model.

colParm1 specifies the cyan value.
colParm2 specifies the magenta value.
colParm3 specifies the yellow value
colParm4 specifies the black value.
The values are specified as an integer percentage within the range of 0 to 100.

CIELAB Defines the CIELAB color model.
colParm1 specifies the luminance value as an integer percentage within the range of 0 to 100.
colParm2 and
colParm3 specify the chrominance differences as signed integers from -127 to 127.
colParm4 is ignored.

HLC Defines the HIGHLIGHT color model.
colParm1 specifies the highlight color number to be used. This value can be a decimal value (range 1 – 65535) or a hex value (range 0x0001 – 0xFFFF). Highlight color numbers >= 256 are "Indexed Colors" defined by an Indexed CMR. colParm2 and colParm3 are ignored by the printer in this case.
colParm2 specifies the coverage value as an integer percentage within the range of 0 to 100.
colParm3 specifies the shading value as an integer percentage within the range of 0 to 100.
colParm4 is ignored.

Color model	RGB	CMYK	CIELAB	HLC
colParm1	red	cyan	luminance	name
colParm2	green	magenta	chrominance	coverage
colParm3	blue	yellow	chrominance	shading
colParm4	-	black	-	-

DEFINECONTAINER **('*name*' *,* [*unit*] *,* [[-]*posX*] , [[-]*posY*], [*areaWidth*], [*areaHeight*], [*map*] , [[-]*posXInObjArea*] , [[-]*posYInObjArea*] , [*orientation*] , *backgroundColor* , *type* , *mode* , *fileName* | *"fileName"*)**

Defines the characteristics of an object to be inserted in form of an AFP Object Container. The object's data is read from an external file and is saved in RESOBJDD.

In case of OCA objects, all parameters, except *name* and *filename*, are optional and no defaults are used.
For non-OCA-Objects all parameters, except unit are mandatory.

This keyword must be specified together with RESTYPE(ALL) or RESTYPE(OBJCON) otherwise a warning message is issued.

More information about the data enrichment functionality in CIS can be found under *3.8 How to Enrich Print Data streams* on page 77.

'name'	Is a 1 to 8 characters name which is used to uniquely identify the Object being defined. It is allowed to use a name which is already being used to identify an Object Container within a print file, but it will not be possible to differentiate between the "old" and the "new" object after that.
unit	Any of the units of measurement supported by CIS (see *4.4.6 Units of measurement* on page 165 for more information). PELS is the default.
[-]posX	Specifies the offset in the X direction, relative to the Medium Origin, in the units specified in *unit*, where the Objects Area presentation space starts. A value between -(2^{31}-1) and 2^{31}-1 is allowed. The position must be inside the page presentation space. This value corresponds to the value specified in field XoaOset (applied to the Medium Origin) of the IOB structured field.
[-]posY	Specifies the offset in the Y direction, relative to the Medium Origin, in the units specified in *unit*, where the Object Area presentation space starts. A value between -(2^{31}-1) and 2^{31}-1 is allowed. The position must be inside the page presentation space. This value corresponds to the value specified in field YoaOset (applied to the Medium Origin) of the IOB structured field.
areaWidth	Specifies the width, in the units specified in *unit*, of the Object Area presentation space. This parameter is mandatory for non-OCA objects. For OCA objects the parameter is optional and may be used to override the area width specification defined in the object container. A value between 1 and 2^{32}-1 is allowed. This value corresponds to the value specified in field XoaSize of triplet 0x4C in structured field OBD in an Object Container.
areaHeight	Specifies the height, in the units specified in *unit*, of the Object Area presentation space. This parameter is

mandatory for non-OCA objects. For OCA objects the parameter is optional and may be used to override the area height specification defined in the object container. A value between 1 and 2^{32}-1 is allowed. This value corresponds to the value specified in field YoaSize of triplet 0x4C in structured field OBD in an Object Container.

map
Specifies the mapping of the object in the object placement area. The allowed values are:

0x00	Position
0x10	Position and trim
0x20	Scale to fit
0x30	Center and trim
0x50	Replicate and trim
0x60	Scale to fill

This value corresponds to the value specified in field MapValue of triplet 0x04 of structured field IOB. Please refer to "MapValue" under "Mapping Option Triplet X'04' " in [18] (see *8 Bibliography* on page 465) for more information.

[-]posXInObjArea
Specifies the X offset, in units specified in *unit*, of the object contents within the Object Area presentation space. A value between -(2^{31}-1) and 2^{31}-1 is allowed. This value corresponds to the value specified in field XocaOset of structured field OBP in an Object Container. This parameter is only used when the *map* parameter specifies "Position" or "Position and trim".

[-]posYInObjArea
Specifies the Y offset, in units specified in *unit*, of the object contents within the object placement area. A value between -(2^{31}-1) and 2^{31}-1 is allowed. This value corresponds to the value specified in field YocaOset of structured field OBP in an Object Container. This parameter is only used when the *map* parameter specifies "Position" or "Position and trim".

orientation
Specifies the orientation of the object, relative to the Medium Origin. Possible values are 0°, 90°, 180° and 270°. The default is 0°. This value corresponds to the value specified in fields XoaOrent / YoaOrent of the IOB structured field.

backgroundColor

Specifies the color of the background to be used for filling the object's area. Please refer to the DEFINECOLOR keyword for a description of the allowed values. The default is NONE. This value corresponds to the values specified in triplet 0x4E of structured field OBD in an Object Container.

type Specifies the type of the Object to be inserted. OCA and non-OCA types are supported. Following types are possible:

OCA types:

PSEG	Page segment object
GOCA	Graphic object
BCOCA	Bar Code object
IOCA	Image object
WRAPPED	object file which includes MO:DCA container structures describing the object's type. CIS expects a WRAPPED object to contain all mandatory structured fields and definition triplets defined by the MO:DCA architecture (e.g. BOC structured field with Object Classification X'10' triplet and valid OEG and OCD structure fields).

non-OCA types:

EPS	Encapsulated PostScript
EPST	EPS with transparency
GIF	Graphics Interchange Format
JFIF	JPEG File Interchange Format
PCL	PCL 5 page object (non-OCA)
PCX	Paintbrush Picture File Format
PDF	Portable Document Format single-page object
PDFT	PDF with transparency (non-OCA)
TIFF	Tag Image File Format single, paginated image
DIBOS2	Device Independent Bit Map for OS/2
DIBWIN	Device Independent Bit Map for Windows

This value corresponds to the value specified in field ObjType of structured field IOB in case of OCA objects, or in field ObjClass of triplet 0x10 in structured field BOC in case of an Object Container.

The object data to be included is a paginated presentation object whose format may (OCA) or may not (non-OCA) be defined by an IBM presentation architecture. For non-OCA un-WRAPPED objects, CIS uses the information contained in the DEFINECONTAINER keyword in order to generate the necessary MO:DCA container structures describing the object's type (BOC structure field with Object Classification X'10' triplet and valid OEG and OCD structure fields). Please refer to " Include Object (IOB)" and "Appendix D: Object Type identifiers" in *SC31-6802-05 Mixed Object Document Content Architecture Reference* for more information.

mode Specifies whether the Object Container is to be mapped in a Map Data Resource (MDR) structured field, or not. Following modes are possible:

<u>SOFT</u> Specifies that the object is inserted without an MDR in the Active Environment Group (AEG).

HARD Specifies that the object's name is to be added to the MDR of the AEG of the page where the IOB for the container is being inserted.

fileName Specifies the name of the file (Data Set Name under MVS) containing the Object's data. The OBJCONLIB and/or USERLIB directories are not searched. See also *<u>File name</u>* description under *4.4.3 Portability* on page 163. This parameter cannot be specified in case the INSERTCONTAINER keyword associated to this DEFINECONTAINER specifies the USEREXIT parameter.

"fileName" This notation form must be used when the file name contains special characters such as blanks.

DEFINECURVE **(‘name’ , [*unit*] , *startPosX* , *startPosY* , *endPosX* , *endPosY* [, *endPosX* , *endPosY* [,...,...]] , [*lineType*] , [*lineWidth*] , [*lineColor*] , [*copyNumber*] ,[[-]*copyOffsetX*] , [[-]*copyOffsetY*] , [*gocaPattern*] , [*fillColor*])**

Defines the characteristics of a Curve object to be inserted in form of a GOCA object. Multiple positions ("curve segments") may be specified so that the DEFINECURVE keyword can also be used to draw closed figures containing up to 62 curve segments. More information about this data enrichment functionality in CIS can be found under *3.8.5 Inserting and Removing Graphic Objects (GOCA)* on page 88.

'name' Is a 1 to 250 characters name which is used to uniquely

identify the Curve object being defined. It is allowed to use a name which is already being used to identify GOCA objects within a print file, butit will not be possible to differentiate between the "old" and the "new" curve after that.

unit Any of the units of measurement supported by CIS (see *4.4.6 Units of measurement* on page 165 for more information). PELS is the default.

startPosX Specifies the offset in the X direction, relative to the Medium Origin, in the units specified in *unit*, where the curve segment starts. A value between 0 and 2^{32}-1 is allowed. The position must be inside the current page presentation space.

startPosY Specifies the offset in the Y direction, relative to the Medium Origin, in the units specified in *unit*, where the curve segment starts. A value between 0 and 2^{32}-1 is allowed. The position must be inside the current page presentation space.

endPosX Specifies the offset in the X direction, relative to the Medium Origin, in the units specified in *unit*, where the curve segment ends. A value between 0 and 2^{32}-1 is allowed. The position must be inside the current page presentation space.

endPosY Specifies the offset in the Y direction, relative to the Medium Origin, in the units specified in *unit*, where the curve segment ends. A value between 0 and 2^{32}-1 is allowed. The position must be inside the current page presentation space.

A maximum of 62 end positions may be specified in one DEFINECURVE keyword by entering additional *endPosX* / *endPosY* pairs of parameters.

lineType Specifies the type of GOCA line to be used for the figure's border line. The allowed values are:

0x01	dotted
0x02	short dashed
0x03	dash-dot
0x04	double dotted
0x05	long dashed
0x06	dash double-dot

0x07	solid
0x08	invisible

Please refer to "Line type" in *S544-5498-01 Graphics Object Content Architecture for Advanced Function Presentation Reference,* page 32 for more information.

lineWidth Specifies the width (thickness) of the figure's border line in units of 1/100 inches. A value between 1 and 255 is allowed. The default is 5.

lineColor Specifies the color of the figure's border line. Please refer to the DEFINECOLOR keyword for a description of the allowed values. The default is BLACK.

copyNumber Specifies the number of times the figure should be inserted. A value between 1 and 255 may be specified. The default value is 1.

[-]copyOffsetX
Specifies the offset in the X direction, relative to the previous figure's copy, where the next copy of the figure is to be placed. A value between $-(2^{31}-1)$ and $2^{31}-1$ is allowed. The resulting position must be inside the current page presentation space. This parameter is mandatory when *copyNumber* > 1.

[-]copyOffsetY
Specifies the offset in the Y direction, relative to the previous figure's copy, where the next copy of the figure is to be placed. A value between $-(2^{31}-1)$ and $2^{31}-1$ is allowed. The resulting position must be inside the current page presentation space. This parameter is mandatory when *copyNumber* > 1.

gocaPattern Specifies the pattern to be used to fill a closed figure. It forces an open figure to be automatically closed by drawing a last line segment between the first and last points entered. Specify NONE (the default) for figures which should stay open. The allowed values are:

NONE	open figure
0x01-0x0E	Please refer to "patterns" in [14] (see *8 Bibliography* on page 465)
0x0F	no fill
0x10	solid pattern

fillColor Specifies the color which should be used to fill the closed figure. Please refer to the DEFINECOLOR keyword for a description of the allowed values. The default is BLACK.

DEFINEELLIPSE **(‘name’ , [*unit*] , *posX* , *posY* , *[-]ax1PosX* , *[-]ax1PosY, [-]ax2PosX* , *[-]ax2PosY* , [*lineType*] , [*lineWidth*] , [*lineColor*] , [*copyNumber*] ,[[-]*copyOffsetX*] , [[-]*copyOffsetY*], [*gocaPattern*] , [*fillColor*])**

Defines the characteristics of a Ellipse object to be inserted in form of a GOCA object. More information about this data enrichment functionality in CIS can be found under *3.8.5 Inserting and Removing Graphic Objects (GOCA)* on page 88.

'name' Is a 1 to 250 characters name which is used to uniquely identify the Ellipse object being defined. It is allowed to use a name which is already being used to identify GOCA objects within a print file, but it will not be possible to differentiate between the "old" and the "new" ellipse after that.

unit Any of the units of measurement supported by CIS (see *4.4.6 Units of measurement* on page 165 for more information). PELS is the default.

posX Specifies the offset in the X direction, relative to the Medium Origin, in the units specified in *unit*, where the horizontal X-position of the Ellipse is placed. A value between 0 and 2^{32}-1 is allowed. The position must be inside the current page presentation space.

posY Specifies the offset in the Y direction, relative to the Medium Origin, in the units specified in *unit*, where the horizontal Y-position of the Ellipse is placed. A value between 0 and 2^{32}-1 is allowed. The position must be inside the current page presentation space.

[-]ax1PosX Specifies the offset in the X direction, relative to the *posX*, in the units specified in *unit*. A value between -(2^{31}-1) and 2^{31}-1 is allowed. The position must be inside the current page presentation space.

[-]ax1PosY Specifies the offset in the Y direction, relative to the *posY*, in the units specified in *unit*. A value between -(2^{31}-1) and 2^{31}-1 is allowed. The position must be inside the current page presentation space.

[-]ax2PosX Specifies the offset in the X direction, relative to the *posX*, in the units specified in *unit*. A value between -(2^{31}-1) and 2^{31}-1 is allowed. The position must be inside the current page presentation space.

[-]ax2PosY Specifies the offset in the Y direction, relative to the *posY*, in the units specified in *unit*. A value between $-(2^{31}-1)$ and $2^{31}-1$ is allowed. The position must be inside the current page presentation space.

lineType Specifies the type of GOCA line to be used for the figure's border line. The allowed values are:

0x01	dotted
0x02	short dashed
0x03	dash-dot
0x04	double dotted
0x05	long dashed
0x06	dash double-dot
0x07	solid
0x08	invisible

Please refer to "Line type" in *S544-5498-01 Graphics Object Content Architecture for Advanced Function Presentation Reference,* page 32 for more information.

lineWidth Specifies the width (thickness) of the figure's border line in units of 1/100 inches. A value between 1 and 255 is allowed. The default is 5.

lineColor Specifies the color of the figure's border line. Please refer to the DEFINECOLOR keyword for a description of the allowed values. The default is BLACK.

copyNumber Specifies the number of times the figure should be inserted. A value between 1 and 255 may be specified. The default value is 1.

[-]copyOffsetX

Specifies the offset in the X direction, relative to the previous figure's copy, where the next copy of the figure is to be placed. A value between $-(2^{31}-1)$ and $2^{31}-1$ is allowed. The resulting position must be inside the current page presentation space. This parameter is mandatory when *copyNumber* > 1.

[-]copyOffsetY

Specifies the offset in the Y direction, relative to the previous figure's copy, where the next copy of the figure is to be placed. A value between $-(2^{31}-1)$ and $2^{31}-1$ is allowed. The resulting position must be inside the current page presentation space. This parameter is mandatory when *copyNumber* > 1.

gocaPattern Specifies the pattern to be used to fill the figure. The allowed values are:

0x01-0x0E	Please refer to "patterns" in [14] (see *8 Bibliography* on page 465)
0x0F	no fill
0x10	solid pattern

fillColor Specifies the color which should be used to fill the closed figure. Please refer to the DEFINECOLOR keyword for a description of the allowed values. The default is BLACK.

DEFINEFINISHINGOP **('name' , [*unit*] , *type* , *parameter* , *reference* , [*count*] , [*offset*] , [*position* [, ...]])**

Defines the characteristics of a Finishing Operation to be inserted by SPS-CIS in form of an AFP 0x8E triplet structure. Any finishing triplets already contained in the input file are removed. More information about the data enrichment functionality in CIS can be found under *3.8 How to Enrich Print Data streams* on page 77.

This keyword must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise a warning message is issued.

'name' Is a 1 to 250 characters name which is used to uniquely identify the Finishing Operation being defined. It is allowed to use a name which is already being used to identify Finishing Operations within a print file however, it will not be possible to differentiate between the "old" and the "new" operations after that.

unit Any of the units of measurement supported by CIS (see *4.4.6 Units of measurement* on page 165 for more information). PELS is the default.

type, parameter

one of the following Finishing Operation type/parameter combinations:

type	*parameter*	
FOLD	F4-1	
	F6-*n*	(1 - *n* - 5)
	F8-*n*	(1 - *n* - 7)
	F10-*n*	(1 - *n* - 3)
	F12-*n*	(1 - *n* - 14)
	F14-1	

	F16-*n*	(1 - *n* - 13)
	F18-*n*	(1 - *n* - 8)
	F20-*n*	(1 - *n* - 2)
	F24-*n*	(1 - *n* - 10)
	F28-1	
	F32-*n*	(1 - *n* - 9)
	F36-*n*	(1 - *n* - 2)
	F40-1	
	F48-*n*	(1 - *n* - 2)
	F64-*n*	(1 - *n* - 2)
STAPLE	CORNER EDGE_STITCH SADDLE_STITCH_IN SADDLE_STITCH_OUT	
CUT	SEPARATION PERFORATION CROSS	
TRIM	FRONT 1_EDGE 3_EDGE 5_EDGE	
OFFSET	LEFT RIGHT	
STACK	ALTERNATE	
ROTATE	90 180 270 180XDIR 180YDIR	
PUNCH	ROUND RECTANGULAR	
BIND	YES	
REWIND	WIND	
PRINT	PRINT	

TABLE	YES
BOOKLETMAKER	BOOKLET BOOK
MERGE	LEFTMOST_FIRST RIGHTMOST_FIRST
BANDING	SINGLE DOUBLE CROSS
SHRINK_WRAP	YES
n	*m* For testing purposes. Can be used to specify any value allowed by the UP³I Form Finishing Operating triplet. *n* may specify a value between 0 and 255 (byte 4 in the triplet). *m* may specify a value between 0 and 65535 (bytes 5 and 6 in the triplet).

Please refer to the UP³I V1.03 standard for a definition of the operations done for each of the values specified in *type* and *parameter*.

reference Specifies the sheet's corner or edge to be used as reference when specifying the position of the Finishing Operation. The type of Finishing Operation being defined and the capabilities of the finishing device will determine whether this parameter is used or ignored. One of the following values is possible:

TOP	TOP_LEFT
BOTTOM	TOP_RIGHT
LEFT	BOTTOM_LEFT
RIGHT	BOTTOM_RIGHT

Please note that the reference position is not affected by the print orientation (portrait vs. landscape) and always refers to the same physical, device dependent, corner or edge.

count Specifies the number of times that the operation being defined is to be applied on each sheet. The type of

Finishing Operation being defined and the capabilities of the finishing device will determine whether this parameter is used or ignored. A value between 1 and 255 may be specified. The default is 1. The number of *position* parameters required is determined by the *count* parameter.

offset — Specifies the Axis offset (in *units)* for this Finishing Operation. This parameter specifies the offset of the positioning axis relative to the *reference* edge when the sheet is viewed from the front side. The type of Finishing Operation being defined and the capabilities of the finishing device will determine whether this parameter is used or ignored. A value between 0 and 2^{32}-1 may be specified, with the resulting value not exceeding 2^{32}-1 millipoints. The default is 0.

position — Specifies the position (in *units)* within the Finishing Axis for this Operation. Position 0 is either at the bottom edge or the left edge of the sheet when viewed from the front side, depending on the edge specified in *reference*. The type of Finishing Operation being defined and the capabilities of the finishing device may cause some or all *position* parameters to be ignored. The number of *position* parameters required is determined by the *count* parameter. A value between 0 and 2^{32}-1 may be specified, with the resulting value not exceeding 2^{32}-1 millipoints. The default is 0.

DEFINELEVEL **(*name* , *entity* [, *location* , *entitySelector*])**

Defines a "level" which may be used in other CIS keywords for operations such as data enrichment. The definition of a data enrichment level requires the specification of the entity to be enriched ("what"), the location at which the enrichment operation will be applied ("where") and a selector ("which"). Refer to *4.4.5 Predefined Levels* on page 165 for a list of the predefined levels available in CIS. Predefined levels cannot be changed by the user.

name — Is a 1 to 250 characters name which is used to uniquely identify the Level being defined.

entity — Specifies the entity or scope to be used for this level. The allowed values are listed below. Please refer to *3.8.1 Defining data enrichment levels to CIS* on page

78 or to *3.12.1 Defining data extraction levels to CIS* on page 137 for more information about the entities available in CIS.

DOCUMENT	each MO:DCA document
GROUPLEVEL*n*	each MO:DCA Page Group at level *n (*a 1 digit number between 1 and 9).
SHEET	each MO:DCA sheet
PAGE	each MO:DCA page. Causes *location* to be ignored (ALL is assumed).

location Specifies the location within the entity where the data enrichment operation will be done. It is ignored for PAGE entities. May be one of the following values:

<u>BEGIN</u>	in the first sequential page of the entity.
END	in the last sequential page of the entity.
FRONT	in the first sequential page on the FRONT side of the first sheet of the entity.
BACK	in the last sequential page on the BACK side of the last sheet of the entity.
BEFORE	immediately before the first sequential page of the entity.
AFTER	immediately after the last sequential page of the entity.
ALL	in all pages of the entity.

entitySelector Identifies the entities at which the enrichment will take place . May be one of the following values:

<u>ALL</u>	indicates that all entities are part of this define.
'*name*'	specifies the name of the entity that is part of this define.
n	a number between 1 and 2^{32}-1. Specifies the sequential number within the print file of the entity that is part of this define.
from - to	two numbers between 1 and 2^{32}-1.

	Specify a range of sequential numbers within the print file of the entities that are part of this define. *from* cannot be greater than *to*.
ODD	indicates that all entities with an ODD sequential number are part of this define.
EVEN	indicates that all entities with an EVEN sequential number are part of this define.

More information about the data enrichment functionality in CIS can be found under *3.8 How to Enrich Print Data streams* on page 77. Refer to *3.8.1 Defining data enrichment levels to CIS* on page 78 for a description of the supported level combinations in each applicable CIS keyword.

DEFINELINE **('name' , [*unit*] , *startPosX* , *startPosY* , *endPosX* , *endPosY* [, *endPosX* , *endPosY* [, ... , ...]] , [*lineType*] , [*lineWidth*] , [*lineColor*] , [*copyNumber*] , [[-]*copyOffsetX*] , [[-]*copyOffsetY*] , [*gocaPattern*] , [*fillColor*])**

Defines the characteristics of a Line to be inserted in form of a GOCA object. Multiple positions ("line segments") may be specified so that the DEFINELINE keyword can also be used to draw polygons of up to 62 sides. More information about this data enrichment functionality in CIS can be found under *3.8.5 Inserting and Removing Graphic Objects (GOCA)* on page 88.

'name'	Is a 1 to 250 characters name which is used to uniquely identify the Line object being defined. It is allowed to use a name which is already being used to identify GOCA objects within a print file however, it will not be possible to differentiate between the "old" and the "new" line after that.
unit	Any of the units of measurement supported by CIS (see *4.4.6 Units of measurement* on page 165 for more information). PELS is the default.
startPosX	Specifies the offset in the X direction, relative to the Medium Origin, in the units specified in *unit*, where the line segment starts. A value between 0 and 2^{32}-1 is allowed. The position must be inside the current page presentation space.
startPosY	Specifies the offset in the Y direction, relative to the

Medium Origin, in the units specified in *unit*, where the line segment starts. A value between 0 and 2^{32}-1 is allowed. The position must be inside the current page presentation space.

endPosX Specifies the offset in the X direction, relative to the Medium Origin, in the units specified in *unit*, where the line segment ends. A value between 0 and 2^{32}-1 is allowed. The position must be inside the current page presentation space.

endPosY Specifies the offset in the Y direction, relative to the Medium Origin, in the units specified in *unit*, where the line segment ends. A value between 0 and 2^{32}-1 is allowed. The position must be inside the current page presentation space.

A maximum of 62 end positions may be specified in one DEFINELINE keyword by entering additional *endPosX* / *endPosY* pair of parameters.

lineType Specifies the type of GOCA line to be draw. The allowed values are:

0x01	dotted
0x02	short dashed
0x03	dash-dot
0x04	double dotted
0x05	long dashed
0x06	dash double-dot
<u>0x07</u>	solid
0x08	invisible

Please refer to "Line type" in *S544-5498-01 Graphics Object Content Architecture for Advanced Function Presentation Reference,* page 32 for more information.

lineWidth Specifies the width (thickness) of the line in units of 1/100 inches. A value between 1 and 255 is allowed. The default is <u>5</u>.

lineColor Specifies the color to be used for printing the line. Please refer to the DEFINECOLOR keyword for a description of the allowed values. The default is <u>BLACK</u>.

copyNumber Specifies the number of times the line should be inserted. A value between 1 and 255 may be specified. The default value is <u>1</u>.

[-]copyOffsetX

Specifies the offset in the X direction, relative to the previous figure's copy, where the next copy of the figure is to be placed. A value between $-(2^{31}-1)$ and $2^{31}-1$ is allowed. The resulting position must be inside the current page presentation space. This parameter is mandatory when *copyNumber* > 1.

[-]copyOffsetY
Specifies the offset in the Y direction, relative to the previous figure's copy, where the next copy of the figure is to be placed. A value between $-(2^{31}-1)$ and $2^{31}-1$ is allowed. The resulting position must be inside the current page presentation space. This parameter is mandatory when *copyNumber* > 1.

gocaPattern Specifies the pattern to be used to fill a closed figure. It forces an open figure to be automatically closed by drawing a last line segment between the first and last points entered. Specify NONE (the default) for figures which should stay open. The allowed values are:

<u>NONE</u>	open figure
0x01-0x0E	Please refer to "patterns" in [14] (see *8 Bibliography* on page 465)
<u>0x0F</u>	no fill
0x10	solid pattern

fillColor Specifies the color which should be used to fill the closed figure. Please refer to the DEFINECOLOR keyword for a description of the allowed values. The default is <u>BLACK</u>.

DEFINEMEDIUMMAP **('*mMapName*' ,**
[CONSTANT (NO | FRONT | BACK | BOTH)]
[COPIES (1 | *n*)]
[DUPLEX (NO | SIMPLEX | NORMAL | TUMBLE)]
[EXTENTX (*n* [.*m*] [*unit*])]
[EXTENTY (*n* [.*m*] [*unit*])]
[INBIN (*n*)]
[JOG (YES | NO)]
[MEDIAATTRIBUTES (
['*UP3ImediaName*'] ,
[*UP3ImediaFrontCoating* ;
***UP3ImediaBackCoating*],**
[*UP3ImediaBrightness*] ,
[*UP3ImediaColorResourceID*] ,
[*UP3ImediaImagableSide*] ,

```
                                  [ UP3ImediaColorName ] ,
                                  [ UP3ImediaSetCount ] ,
                                  [ UP3ImediaOpacity ] ,
                                  [ UP3ImediaPreprinted ] ,
                                  [ UP3ImediaRecycled ] ,
                                  [ UP3ImediaRollDiameter ] ,
                                  [ UP3ImediaThickness ] ,
                                  [ UP3ImediaType ] ,
                                  [ UP3ImediaWeight ] ,
                                  [ UP3IpinHole ] ,
                                  [ UP3ImediaWidth ; UP3ImediaLength ] ,
                                  [ UP3ImediaOrderedSetPiece ]
                                 )                                  ]
   [ MEDIATYPE         ( mediaType )                                ]
   [ MEDIANAME         ( 'mediaName' )                              ]
   [ SHEETLETS         ( 1 | 2 [ , LEFT | RIGHT ] )                 ]
   [ OFFSETXB          ( [-] n [.m ] [ unit ] )                     ]
   [ OFFSETXF          ( [-] n [.m ] [ unit ] )                     ]
   [ OFFSETYB          ( [-] n [.m ] [ unit ] )                     ]
   [ OFFSETYF          ( [-] n [.m ] [ unit ] )                     ]
   [ OUTBIN            ( n )                                        ]
   [ OVERLAYB          ( overlayName [ , ... ] )                    ]
   [ OVERLAYF          ( overlayName [ , ... ] )                    ]
   [ PRESENTATION      ( PORTRAIT | LANDSCAPE
                       | PORTRAIT90 | LANDSCAPE90
                       | PORTRAIT180 | LANDSCAPE180 )               ]
   [ SUPPRESSIONID     ( n [ , ... ] )                              ]
 )
```

Provides the characteristics required to add or modify a Medium Map stored in the Form Definition specified by the FORMDEF keyword. In combination with REMOVEMEDIUMMAP, the DEFINEMEDIUMMAP keyword can be used to replace existing Medium Maps. Specifying a DEFINEMEDIUMMAP parameter causes the corresponding fields in the FORMDEF to be replaced.

The Medium Maps created or modified with this keyword will only contain one subgroup and may not use the n-up functionality. The number of sheet copies to be printed by the subgroup is specified using the COPIES sub-keyword.

This keyword must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise a warning message is issued. In case of resource consolidation, the DEFINEMEDIUMMAP keyword is applied to all relevant resource groups. The FormDef override keywords (e.g. INBIN, DUPLEX, etc.) do not influence the DEFINEMEDIUMMAP keyword.

The DEFINEMEDIUMMAP keyword is divided in sub-keywords and parameters. They are:

'*mMapName*'	Is a 1 to 8 characters name which is used to uniquely identify the Medium Map being defined. It is allowed to use a name which is already defined in the Form Definition, in which case the existing medium map is modified to include all the parameters specified in the DEFINEMEDIUMMAP keyword. The Medium Map is inserted or modified regardless on whether it is being used in another CIS keyword or not.
CONSTANT	one of the following values enclosed in parenthesis:

NO	constant forms function is disabled
FRONT	front side includes constant data only
BACK	back side includes constant data only
BOTH	front and back side include constant data only. No variable data is printed on the sheets using this medium map.

COPIES The following value enclosed in parenthesis:

n	Specifies the number of sheet copies to be printed. A value between 1 and 255 may be specified.

DUPLEX one of the following values enclosed in parenthesis:

NO	single (front) sided printing
SIMPLEX	same as NO
NORMAL	two sided printing with top of the front page being the top of the back page
TUMBLE	two sided printing with bottom of the front page being the top of the back page

EXTENTX
EXTENTY Specify the extent in the **X** (**Y**) direction for each sheet. For the maximum values allowed please see *Fig. 77 Units of measurement: maximum values* on page 166.

n	A 1 to 8 digits number.
m	A 1 to 3 digits number.
unit	Any of the units of measurement supported by CIS. PELS is the default.

INBIN The following value enclosed in parenthesis:

n	A decimal number from 0 to 254. See

Keyword INBIN for further details.

JOG one of the following values enclosed in parenthesis:

YES	change stack offset / Carriage Edge Marks (CEM) for the first sheet
NO	no offset / no CEM change

MEDIAATTRIBUTES

Specifies one or more media attributes enclosed in parenthesis. Only the attribute which are specified are inserted in the Medium Map. For more information please refer to *UP³I Universal printer pre- and post-processing interface, Version 1.03* under *"Paper Input Media triplet"*.

'*UP3ImediaName*'	A 1 to 250 characters name used to identify the Media. The name may only contains characters available in the ISO_646 IRV:1991 (US ASCII) code page.
UP3ImediaFrontCoating *UP3ImediaBackCoating*	one of the following values: NONE GLOSSY HIGHGLOSS MATTE SATIN SEMIGLOSS n $(0 \le n \le 255)$
UP3ImediaBrightness	reflectance percentage. A value between 1 and 100.
UP3ImediaColorResourceID	A value between 0 and 255. Specifies a color resource that provides the color of the chosen media.
UP3ImediaImagableSide	one of the following values: FRONT BACK BOTH NONE n $(0 \le n \le 65535)$

UP3ImediaColorName	one of the following values: BLACK BLUE BROWN BUFF GOLD GOLDENROD GRAY GREEN IVORY MUSTARD NO_COLOR ORANGE PINK RED SILVER TURQUOISE VIOLET WHITE YELLOW *n* ($0 \le n \le 255$) CIS always sets the Color-Prefix fields of the "Paper Input Media Color Name Sub Triplet" to the default values.
UP3ImediaSetCount	a value between 1 and 65535. Specifies the number of pieces of media in the set.
UP3ImediaOpacity	one of the following values: OPAQUE TRANSPARENT
UP3ImediaPreprinted	one of the following values: BLANK PREPRINTED
UP3ImediaRecycled	one of the following values: NEW RECYCLED
UP3ImediaRollDiameter	a value between 1 and 2^{32}-1. Specifies the Roll diameter in millipoints.

UP3ImediaThickness	a value between 1 and 2^{32}-1. Specifies the thickness of the chosen media in microns (µm).
UP3ImediaType	one of the following values: CONTINOUSCONTINUOUS CONTINOUSCONTINUOUS_LONG CONTINOUSCONTINUOUS_SHORT ENVELOPE ENVELOPE_PLAIN ENVELOPE_WINDOW FULL_CUT_TABS LABELS LETTERHEAD MULTI_LAYER MULTI_PART_FORM PHOTOGRAPHIC PRE_CUT_TABS STATIONERY TAB_STOCK TRANSPARENCY *n* $(0 \le n \le 255)$
UP3ImediaWeight	a value between 1 and 2^{32}-1. Specifies the weight of the chosen media in grams per square meter (g/m²).
UP3IpinHole	one of the following values: PINHOLES PINLESS
UP3ImediaWidth *UP3ImediaLength*	two values between 1 and 2^{32}-1. Specify the Media dimensions (width and length) in millipoints. A length value of 0 is used for continuous-form media.
UP3ImediaOrderedSetPiece	a value between 1 and the

UP3ImediaSetCount. Identifies the piece (sheet) within the media set which is to be used for this Medium Map.

MEDIATYPE The following value enclosed in parenthesis:

mediaType Specifies the component ID of the media to be selected. One of the following values may be specified:

10x13_ENV	ISO_B4_ENV
12x18_MED	ISO_B5
14x17_MED	ISO_B5_CO
14x18_MED	ISO_B5_CO
8.5x10_MED	ISO_B5_ENV
8x10.5_MED	ISO_C4_ENV
8x10_MED	ISO_C5_ENV
9x12_ENV	ISO_LNG_ENV
9x12_MAN	JIS_B4
9x14_MED	JIS_B5
BSNS_ENV	JP_PC
C5_ENV	JP_PC_ENV
COM_10_ENV	LEDGER
DL_ENV	LEGAL
EXEC	LEGAL_13
INDEX_CD	LEGAL_CO
ISO_A3	LEGAL_TAB
ISO_A3_CO	LETTER
ISO_A4	LETTER_CO
ISO_A4_CO	LETTER_TAB
ISO_A4_TAB	LETTER_TR
ISO_A4_THD	MON_ENV
ISO_A4_TR	RA3
ISO_A5	RA4
ISO_A5_CO	STATEMNT
ISO_A6_PC	US_PC
ISO_B4	x'nn'
ISO_B4_CO	

10x13_ENV	14x17_MED	
	14x18_MED	8.5x10_MED
	8x10.5_MED	8x10_MED
	9x12_ENV	9x12_MAN
	9x14_MED	BSNS_ENV
	C5_ENV	COM_10_ENV

DL_ENV	EXEC
INDEX_CD	ISO_A3
ISO_A3_CO	ISO_A4
ISO_A4_CO	ISO_A4_TAB
ISO_A4_THD	ISO_A4_TR
ISO_A5	ISO_A5_CO
ISO_A6_PC	ISO_B4
ISO_B4_CO	ISO_B4_ENV
ISO_B5	ISO_B5_CO
ISO_B5_CO	ISO_B5_ENV
ISO_C4_ENV	ISO_C5_ENV
ISO_LNG_ENV	JIS_B4
JP_PC	JP_PC_ENV
LEDGER	LEGAL
LEGAL_13	LEGAL_CO
LEGAL_TAB	LETTER
LETTER_CO	LETTER_TAB
LETTER_TR	MON_ENV
RA3	RA4
STATEMNT	US_PC
x'nn'	

where *nn* is a Media Type Identifier. Please refer to Media Type Identifiers registry (OIDs) in *SC31-6802-05 MO:DCA Reference* for more information.
If MEDIATYPE not defined, the MEDIANAME must be defined.

MEDIANAME The following value enclosed in parenthesis:

'mediaName' A 1 to 12 characters name used to identify the Media. The name may only contain characters available in the IBM 500 code page.
If MEDIANAME not defined, the MEDITYPE must be defined.

SHEETLETS

two of the following values enclosed in parenthesis:

n May be 1, or 2. Specifies the number of sheetlets per sheet to be printed.

<u>LEFT</u> Specifies that the sheetlets are to be printed starting with the leftmost one, or

RIGHT Specifies that the sheetlets are to be printed starting with the rightmost one.

OFFSETXB OFFSETXF OFFSETYB OFFSETYF	Specify the offset in the **X** (**Y**) direction of the logical page origin from the media origin for the **F**ront (**B**ack) side of each sheet. For the maximum values allowed please see *Fig. 77 Units of measurement: maximum values* on page 166.

n	A 1 to 8 digits number.
m	A 1 to 3 digits number.
unit	Any of the units of measurement supported by CIS. PELS is the default.

OUTBIN The following value enclosed in parenthesis:

n	Specifies the output bin number. A decimal number from 0 to 254.

OVERLAYB
OVERLAYF The following value enclosed in parenthesis:

mediaName	One or more 1 to 8 character names specifying the name of the medium overlay to be placed on the **F**ront or **B**ack side of each sheet.

PRESENTATION
one of the following values enclosed in parenthesis:

PORTRAIT	force presentation PORTRAIT
LANDSCAPE	force presentation LANDSCAPE
PORTRAIT90	
LANDSCAPE90	
PORTRAIT180	
LANDSCAPE180	

SUPPRESSIONID
one to eight of the following values enclosed in parenthesis:

n	Specifies the suppression ID. A decimal number from 1 to 127.

DEFINEPTOCA **(*'name'*, [*unit*] , *posX* , *posY* , [*orientation*] , [*color*] , [*font*] , [*codePageID*])**

Defines the characteristics of a Presentation Text object to be inserted by CIS in form of a PTOCA AFP structure using the INSERTPTOCA or the INSERTPTOCANOP keywords. More information about the data enrichment functionality in CIS can be found under *3.8 How to Enrich Print Data streams* on page 77.

'name' Is a 1 to 250 characters name which is used to uniquely identify the Presentation Text object being defined. It is allowed to use a name which is already being used to identify PTOCA objects within a print file, but it will not be possible to differentiate between the "old" and the "new" objects after that.

PTOCA objects inserted by CIS include a Begin Presentation Text Object (BPT) structured field with a “Fully Qualified Name” triplet (0x02) specifying the *name* parameter as fully qualified name.

unit Any of the units of measurement supported by CIS (see *4.4.6 Units of measurement* on page 165 for more information). PELS is the default.

posX Specifies the offset in the X direction, relative to the Medium Origin, in the units specified in *unit*, where the Presentation Text is to be printed. Printing a PTOCA object outside the currently defined page is not allowed. See *3.9* on page 109 for more information.

posY Specifies the offset in the Y direction, relative to the Medium Origin, in the units specified in *unit*, where the Presentation Text is to be printed. Printing a PTOCA object outside the currently defined page is not allowed. See *3.9* on page 109 for more information.

orientation Specifies the orientation of the Presentation Text, relative to the Medium Origin. Possible values are 0°, 90°, 180° and 270°. The default is 0°.

color Specifies the color of the figure's border line. Please refer to the DEFINECOLOR keyword for a description of the allowed values. The default is BLACK.

font Specifies the name of a Coded Font resource to be loaded and used to print the Presentation Text. The font must be available in the first External Resource Group in the print file (if any) otherwise it is loaded from an external resource library (see USERLIB and FONTLIB keywords). The printer default font is used when this parameter is not specified.

codePageID	Specifies the ID of the code Page to be used for translating. See also *3.2.1 Supported Code Page IDs* on page 38. The default is to use the same code page ID value as specified (or used by default) in the INPUTCP keyword.

For more information about Presentation Text objects please refer to *Data Stream and Object Architectures: Presentation Text Object Content Architecture Reference*.

DUPLEX **(<u>ASIS</u> | NO | SIMPLEX | NORMAL | TUMBLE)**

Determines whether printing will be on the front side of the sheet (simplex) or on both sides. This keyword overrides the MMC Duplex Control specifications in all Medium Map(s) included in the FORMDEF.

This keyword must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise a warning message is issued.

<u>ASIS</u>	The Duplex control specified in the FormDef should be left unchanged.
NO	Simplex printing requested. Simplex printing is single (front) sided printing.
SIMPLEX	Same as NO.
NORMAL	Normal duplex printing requested. Normal duplex printing is two sided printing with top of the front page being the top of the back page. This option permits binding a two sided document on the left edge.
TUMBLE	Tumble duplex printing requested. Tumble duplex printing is duplex printing with the bottom of the front page being the top of the back page. This option permits binding a two sided portrait document on the top edge.

Example: DUPLEX (NORMAL)

EXTRACTCP **(*codePageID*)**

Specifies the Code Page to be used when generating a data extraction file. A description of the data extraction function is available under *3.12 How to Extract information* on page 137. An overview on code page support can be found under *3.2 Code page support* on page 33.

codePageID Specifies the ID of the code Page to be used for translating. See also *3.2.1 Supported Code Page IDs* on page 38. By default, when this parameter is not specified, CIS uses the currently active internal Code Page. For more information please refer to *3.2 Code page support* on page 33.

Example: EXTRACTCP (ISO_8859-1)

EXTRACTINDEX **(*fileName* | "*fileName*" , *level*, '*_sprintfString*', '*fieldName*' [, '*fieldName*'] [, ...])**

Requests the generation of an Index data extraction file and defines the format and source of the data to be extracted.

fileName Is the file name where the data extracted by CIS will be written.

"fileName" This notation form must be used when the filename contains special characters such as blanks.

level Describes the data level from which the Index data is to be extracted. Please refer to the DEFINELEVEL keyword for a description of the allowed values.

'_sprintfString'

Specifies a template followed by a series of data fields (*'attributeName'*) and applies to each data field a format identifier contained in the *'_sprintfString'* template.

Refer to *4.4.2 Formatting string* on page 161 for a detailed description of the options available.

'fieldName' Specifies a valid field name as defined in the control file. Only FIELDs of type LIT, VAR or ATT are valid.

FDEFLIB **(*directory* | "*directory*" [, *directory* | "*directory*"] [, ...])**

Specifies the directories where CIS searches for the form definitions. CIS searches in the same order as the directories are given. Any number of directories can be specified.

directory Specifies the directory to be searched.

"directory" This notation must be used when the directory name contains special characters such as blanks.

The order CIS searches for Form definitions is:

Inline resource group
Userlibs if given
Defined directories

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

FIELD **(*fieldName*, { *record*, *column*, *length* } |**
{ '*literal value*' [, LIT] } |
{ x'*literal value*' [, LIT] } |
{ _*variableName* [, VAR] } |
{ '*attributeName*', ATT } |
{ x'*attributeName*', ATT } |
{ '*searchString*' | x'*searchString*', offset, length })

Specifies the data fields to be used to construct the indexing information, when processing S/370 line formatted or MO:DCA-P data. May be specified more than once.

fieldName A 1 to 250 byte character name which is used as reference in the INDEX keyword.

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

column Will be used as character 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 to compose this field. Supported values are 1 to 250.

'literal value' [, LIT]
x*'literal value'* [, LIT]
Specifies a constant value. This value can consist of alphanumeric (case sensitive) or hexadecimal data. The length has a range from 1 to 250 for alphanumeric and 1 – 500 for hexadecimal data (primarily intended for UCS-2 and other double-byte encoding schemes). The keyword LIT indicates that this is a literal field.

_variableName [, VAR]

Specifies the name of one of the internal CIS variables. The variable names available in CIS are explained under *3.4 Processing variables* on page 46. The initial value of each of these variables can be set using the INITVARIABLES keyword.
The keyword VAR indicates that this is a field containing a variable name.

'attributeName' [, ATT]
x*'attributeName'* [, ATT]

Specifies a user-defined attribute name to be associated with the index tag. This value can consist of alphanumeric (case sensitive) or hexadecimal data. The length has a range from 1 to 250 for alphanumeric and 1 – 500 for hexadecimal data (primarily intended for UCS-2 and other double-byte encoding schemes).
The keyword ATT indicates this and must be specified.

'*searchString*' | x'*searchString*'

Specifies a search string (use the x'...' variant in case of hex data).
Note: CIS only searches for searchString on the page where the trigger is found.

offset — number of characters relative to the first character of search string to use as value.

length — number of characters read as index value.

FINISHINGFIDELITY **({ASIS | NONE | STOP | CONTINUE} , {YES | NO})**

Controls the way the presentation software (e.g. SPS) will handle finishing exceptions reported by the printer. Not all printers support finishing fidelity. Causes CIS to insert a PFC structured field containing the Finishing Fidelity triplet x'88' in all Medium Maps in the form definition being used.

This keyword must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise a warning message is issued.

ASIS — Indicates that the current Finishing fidelity options in the

FORMDEF are to be left unchanged. This is the default.

NONE	Causes to remove all Finishing fidelity (triplet x'88') options from the FORMDEF.
STOP	Causes the presentation software (SPS) to stop printing the job and to place it in hold in case the printer reports a finishing exception.
CONTINUE	Causes the presentation software (SPS) to issue a message and to continue printing the job ignoring the requested finishing operation(s).

<u>YES</u>	Report finishing exceptions that do not stop presentation.
NO	Do not report finishing exceptions that do not stop presentation.

FONTCHARSETEXT **(<u>""</u> | "*extension*" [, "*extension*"] [,...])**

Specifies one or more suffixes to be appended to an AFP Font Character Set resource name while searching for the corresponding file.

extension A variable length character string starting with a dot. The default is no suffix.

CIS searches in the same order as the extensions are given. The first match is used.

Example: FONTCHARSETEXT (“.600”, “.300”, “.240”, “.afp”, “”)

The above example causes CIS to search for the following resource names:

*name***.600**
*name***.300**
*name***.240**
*name***.afp**
name

FONTFIDELITY **(<u>ASIS</u> | NONE | STOP | CONTINUE)**

Controls the way the CIS and the presentation software (e.g. SPS) will handle "font unavailable" situations. It causes CIS to insert a PFC structured field containing a Font Fidelity triplet x'78' in all Medium Maps in the form definition being used. The default (in CIS and SPS) when this keyword is not specified is to use the font fidelity options specified in the FormDef.

This keyword must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise a warning message is issued.

<u>ASIS</u>	Indicates that the current Font Fidelity options in the FORMDEF are to be left unchanged. This is the default.
NONE	Causes to remove all Font Fidelity (triplet x'78') options from the FORMDEF.
STOP	Causes the presentation software (SPS) to stop printing the job and to place it in hold in case a font requested by the print file is not available. It causes CIS to stop processing the print file in case RESTYPE(ALL) or RESTYPE(FONT) was also specified.
CONTINUE	Causes the presentation software (SPS) to issue a message and to continue printing the job using a substitute font. It causes CIS to issue a message and to ignore the unavailable font; no substitution is done by CIS.

FONTLIB **(*directory* | *"directory"* [, *directory* | *"directory"*] [, ...])**

Specifies the directories where CIS searches for the font definitions. CIS searches in the same order as the directories are given. Any number of directories can be specified.

directory	Specifies the directory to be searched.
"directory"	This notation must be used when the directory name contains special characters such as blanks.

The order CIS searches for Font definitions is:

Inline resource group
Userlibs if given
Defined directories

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

FONTMAP **(*fontMapName* | <u>DUMMY</u> [, <u>RESOLVE</u> | ASIS])**

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

fontMapName The name can be one to eight alphanumeric characters, including the two-character prefix, if there is one. Specifying DUMMY (the default) requires the print file to contain at least one inline Font Mapping table. CIS uses the first Font Mapping table found and ignores all others.

fontMapProcessing
ASIS|RESOLVE

ASIS Causes CIS to leave all fonts requests (including those made via GRID) unchanged. The Font Mapping table is also left unchanged.

RESOLVE Causes CIS to process the Font Mapping table and to collect the substitution fonts. The font invocations are changed to use the substitution fonts.

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 (the default) requires the print file to contain at least one inline FormDef. CIS uses the first FormDef found. All others are ignored.

Example: FORMDEF (F1MYFORM)

FORMDEFEXT **("" | "*extension*" [, "*extension*"] [,...])**

Specifies one or more suffixes to be appended to an AFP FormDef resource name while searching for the corresponding file.

extension A variable length character string starting with a dot. The default is no suffix.

CIS searches in the same order as the extensions are given. The first match is used.

Example: FORMDEFEXT (“.600”, “.300”, “.240”, “.afp” , “”)

The above example causes CIS to search for the following resource names:

*name***.600**
*name***.300**
*name***.240**
*name***.afp**
name

GROUPNAME **(*indexName*)**

Specifies which of the index values should be used as the name 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 keyword but does not include the attribute name. The maximum name length allowed is 250 characters (AFP restriction).

Example: GROUPNAME(accountI)

This keyword is rejected when specified together with the sorting schemes SCHEME_1, SCHEME_2, SCHEME_3, 2UP_BOOKLET, 2UP_MP_BOOKLET or 4UP_BOOKLET.

IMAGEOUT **(ASIS | <u>IOCA</u> | IOCANOR)**

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

ASIS	CIS uses same image format as in input file.
<u>IOCA</u>	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 **(*value*)**

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 keyword must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise a warning message is issued.

value	A decimal number from 0 to 254. CIS will increment this value by one. Thus INBIN(0) selects the Inbin 1 at the printer.

Example: INBIN (1)

INDEX **(*indexName , triggerName* , '*attributeName*' | x'*attributeName*' , *fieldName* [, *fieldName*] [,..])**

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

indexName	A 1 to 250 character name that specifies the index.
triggerName	Specifies a trigger.
'*attributeName*' x'*attributeName*'	Specifies a user-defined attribute name to be associated with the index tag. This value can consist of alphanumeric (case sensitive) or hexadecimal data. The length has a range from 1 to 250 for alphanumeric and 1 – 500 for hexadecimal data (primarily intended for UCS-2 and other double-byte encoding schemes).
fieldName	Specifies one or more FIELD keywords that compose the index value. The total length of all fields used in an INDEX keyword may not exceed 250.

INDEXCP **(*codePageID* | UNKNOWN)**

Specifies the Code Page to be used for storing the Index Object file . A description of the indexing process is available under *3.6 How to convert, normalize and index* on page 55. An overview on code page support can be found under *3.2 Code page support* on page 33. By default CIS uses the same Code Page as specified in the INPUTCP keyword.

codePageID	Specifies the ID of the code Page to be used for translating. See also *3.2.1 Supported Code Page IDs* on page 38. For more information please refer to *3.2 Code*

page support on page 33.

UNKNOWN Causes CIS to write all Index entries as they are, without doing any code page translation. This is required in case of double-byte data such as Unicode.

Example: INDEXCP (ISO_8859-1)

INDEXDD **(<u>CISIDX</u> | *fileName* | "*fileName*")**

Specifies the name of the Index Object file. A description of this function is available under *3.6 How to convert, normalize and index* on page 55.

fileName Specifies the file name of the Index Object file.

"fileName" This notation form must be used when the file name contains special characters such as blanks.

Example: INDEXDD ("..\..\myFile")

INDEXEXIT **(exitName | '*exitName*')**

exitName | *'exitName':*
Specifies the name of a user exit program. The program will be called every time this INDEXEXIT operation is to be done. The exit program will be called for every index tag found or inserted in the document file and in the index object file, and will be allowed to modify the index information before it is written.

Note: Under OS/390 and z/OS CIS:
- the exitName is restricted to 1 to 8 characters.
- exitName without apostrophes are accepted (done for compatibility with version 4.00.)

Example: INDEXEXIT ('myExit')

More information can be found under *5.2 Index exit* on page 289.

INDEXOBJ **(GROUP | ALL | <u>NONE</u>)**

Controls the amount of information to be written in the Index Object file.

Selecting ALL may result in a very large Index Object file. A value other than NONE causes 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 OUTPUTSEG or 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 keyword to tell CIS to continue looking for an Indexing Trigger on a page other than the first one in the file.

Example: INDEXSTARTBY (3)

INITVARIABLES **(*_variableName* = *initialValue* [, [-] *increment*] [, *_variableName* = *initialValue* [, [-] *increment*]] [*, ...*])**

Specifies the name of one or more of CIS defined variables and assigns each of them an initial value. The variables names available in CIS, their meaning, their defaults, and their “change / “reset” conditions are explained under *3.4 Processing variables* on page 46.

_variableName Specifies the variable name to be initialized.

initialValue	Specifies the value that will be assigned to the variable anytime a "reset" condition occurs. Different reset conditions apply depending on the variable.
increment	Specifies the increment (positive or negative) to be applied to the variable every time a "change" condition occurs. The default is 1. Different reset conditions apply depending on the variable.

Example: INITVARIABLES(_outDocumentNumber = 10)

INLINERESOURCECACHING **(ON | <u>OFF</u>)**

Processing data with one or more inline resources might increase cpu consumption because of the need to reposition within the print data file. Activating InlineResourceCaching allows CIS to save the content of inline resources in memory. This might lead to more memory consumption however the cpu consumption will be drastically reduced, especially under z/OS.

INPEXIT **(exitName | '*exitName*')**

exitName | 'exitName'
Specifies the name of input record exit program. The program will be called for every input record read from the input file and will be allowed to add, modify or delete records.

Note: Under OS/390 and z/OS CIS:
- the exitName is restricted to 1 to 8 characters.
- exitName without apostrophes are accepted (done for compatibility with version 4.00.)

Example: INPEXIT ('myExit')

More information can be found under 5.3 Input Record exit on page 292. Please contact your Océ representative for more information concerning the CIS exit interface.

INPUTCP **(*codePageID* | UTF16LE | UTF16BE | <u>UNKNOWN</u>)**

Specifies the Code Page to be used for interpreting the Input print file when it is not in MO:DCA format. A description of this function is available under *3.6 How to convert, normalize and index* on page 55. An overview on code page support can be found under *3.2 Code page support* on page 33.

codePageID Specifies the ID of the code Page to be used for translating. See also *3.2.1 Supported Code Page IDs* on page 38. For more information please refer to *3.2 Code page support* on page 33.

UTF16LE Indicates that the input print file contains Unicode (UCS-2, Little Endian) Line format data. Use the INPUTFORMAT keyword to specify the end-of-line delimiter. For more information please refer to *3.6.4 Converting and Indexing print files containing Unicode (UCS-2) Line data* on page 61. The output file generated by CIS will be in UTF16BE format.

UTF16BE — Indicates that the input print file contains Unicode (UCS-2, Big Endian) Line format data. Use the INPUTFORMAT keyword to specify the end-of-line delimiter. For more information please refer to *3.6.4 Converting and Indexing print files containing Unicode (UCS-2) Line data* on page 61.

<u>UNKNOWN</u> — Indicates that the encoding used in the input print file is unknown or unsupported. Use this parameter in case of double-byte applications other than UTF16. This is the default.

A *codePageID* other than UNKNOWN, UTF16LE or UTF16BE is required when the name of a FIELD keyword of type other than LIT, VAR or ATT is specified in an INSERTBCOCA(), INSERTINDEX(), INSERTPTOCA(), INSERTPTOCANOP() or EXTRACTINDEX keyword.

Example: INPUTCP (ISO_8859-1)

INPUTDD **(<u>CISIN</u> | *fileName* | "*fileName*" [, *fileName* | "*fileName*"] [, ...])**

Specifies the file name(s) to be used for reading the input print data. CIS reads the file in the same order as they are specified.

fileName — Specifies the name of the file where the input print data is stored.

"fileName" — This notation form must be used when the filename contains special characters such as blanks.

Example: INPUTDD ("myInFile_1", "myInFile_2")

INPUTFORMAT **(MODCA | RDW | AFPSTREAM , x'*delimiter*' | FIXED , *size* | REC_PREFIX**, prefixLength, lengthPosition, [**INCL, <u>EXCL</u>**]) **)**

Specifies the data format used in the input print file(s) specified by the INPUTDD keyword. It applies to the complete input file(s), including any inline resources it(they) may contain.

MODCA Cannot be specified together with the CC or TRCTYPE keywords which must be set to NO (their default). Indicates that the input print file(s) contain the following data format:

- AFP structured fields beginning with a 2-bytes length field (big endian), followed by at least one data byte containing 0xD3. The length field must contain a value between 3 and 32767.
- One optional 0x5A byte preceding the MODCA structured fields.

RDW Used for input print file(s) in S/390 record format where a 4-bytes "record descriptor word" (RDW) specifies the record length. CIS uses the RDW to determine the record size, regardless on whether the data contained in the file is S/370 Line format data, AFP Mixed data, Unformatted ASCII data, Record-format Line data, Unicode Line data, XML data or AFP structured fields.

Under MVS, the RDW is stored outside the data portion of the record ("record length"). In this case, CIS automatically uses the specifications stored in the file's DCB at the time the input print file was created.

Under Linux the first 4 bytes of each record specify the RDW. The first 2 bytes contain the record length, including the RDW length itself, and are followed by 2 additional bytes which must be present and are ignored by CIS (reserved). Files containing RDW's are usually MVS files which are NFS mounted for processing under LINUX. They may also be files which were transferred using FTP or the AFP Download protocol.

Optional CC and/or TRC bytes may follow the 4-bytes RDW. Empty lines (where the RDW is set to 4) are valid as far as the INPUTFORMAT processing is concerned. They may be invalid in the AFP context and may be rejected by CIS.

AFPSTREAM , x'*delimiter*'

Used for input print file(s) containing S/370 Line format data, AFP Mixed data, Record-format Line data, Unicode Line data, Unformatted ASCII data, XML data or AFP structured fields separated by delimiters.

x'*delimiter*' specifies a hexadecimal value to be used as line-end delimiter. The length of the delimiter may vary between 1 and 4 bytes. The delimiter is searched, recognized and removed from the input data with the following exceptions:

- The CC and TRC bytes are not checked for line-end delimiters.
- The content of AFP structured fields is not checked for line-end delimiters.

  The INPUTFORMAT processing in CIS recognizes AFP structured fields as follows: the contents of the CC byte **must** be 0x5A (CC keyword must specify to SNI, ASA, IBM or ASAA); the data after the CC byte (ignoring any TRC specification) begins with a 2-bytes length field with a value between 3 and 32767 (big endian); and is followed by at least one data byte containing 0xD3. If a structured field is recognized, CIS checks for subsequent AFP structured fields within the same line (2-bytes length field followed by 0xD3) before it checks for the line-end delimiter.

Empty lines (two or more consecutive line-end delimiters) are valid as far as the INPUTFORMAT processing is concerned. They may be invalid in the AFP context and may be rejected by CIS.

FIXED , *size* Used for input print file(s) containing fixed length records. The total number of bytes in each input file must be an integer multiple of *size*.

size specifies a value between 1 and 32767 which indicates the length of each line in the input file(s).

CIS uses the size specified, regardless on whether the data contained in the file is S/370 Line format data, AFP Mixed data, Unformatted ASCII data, XML data or AFP structured fields. Applications generating fixed size AFP structured fields may use the AFP padding facility.

REC_PREFIX, prefixLength, lengthPosition, [INCL, <u>EXCL</u>])

Used for input print file(s) containing length prefixes, which can contain more information than only the length.

prefixLength
Is the length of the prefix of each record. The range for the prefixLength is 2 – 32767.

lengthPosition
Specifies the positions inside the prefix, where the length information will start and end.

INCL | EXCL
Specifies if the length is inclusive or exclusive of the record prefixLength.

Example: INPUTFORMAT(REC_PREFIX,6,1-2,INLC)

With this parameter, the user specifies that the input records have a prefix of length 6, where the first two bytes contain the record length, which is inclusive in the record length. CIS takes these records and removes from each record byte 3-6.

Example: INPUTFORMAT (AFPSTREAM, x'25')

The default under MVS is: INPUTFORMAT (RDW)

The default under Linux is
INPUTFORMAT (AFPSTREAM, X'0D0A')

INSERTBCOCA **(*'name'* , *level*, *'_sprintfString'* [, *fieldName*] [, ...])**

Requests the insertion of a Bar Code object and defines the level, format and contents it should have.

'name' Corresponds to the name of a BCOCA object previously defined using the DEFINEBCOCA keyword.

level Describes the data level at which the object is to be inserted. Please refer to the DEFINELEVEL keyword and to *3.8.1 Defining data enrichment levels to CIS* for a description of the allowed values.

'_sprintfString'
Specifies a template followed by a series of data fields (*fieldName*) and applies to each data field a format identifier contained in the *'_sprintfString'* template. Refer to *4.4.2 Formatting string* on page 161 for a detailed description of the options available.

The string 'USEREXIT' may be used to indicate that *fieldName* contains the name of a user exit program to be invoked every time this INSERTBCOCA operation is to be done. The exit program determines the contents of the string to be inserted. Specifying USEREXIT causes the CIS normalization process to store all the input data into the internal work file before the conversion process is started.

fieldName When *_sprintfString* is not 'USEREXIT', this field specifies one or more FIELD keywords that compose the Bar Code value. The total length of all fields used in an INSERTBCOCA keyword may not exceed 250 otherwise it is truncated. The INPUTCP keyword must specify a *codePageID* other than UNKNOWN, UTF16LE or UTF16BE in case *fieldName* corresponds to a FIELD keyword of type other than LIT, VAR or ATT.

When *_sprintfString* specifies 'USEREXIT', this field contains the name of the user exit program to be invoked. Only one occurrence of *fieldName* is allowed in this case. More information can be found under *5.7 BCOCA-insertion exit* on page 303.

INSERTBOX **('*name', level*)**

Requests the insertion of a Box object and defines the level. More information about this data enrichment functionality in CIS can be found under *3.8.5 Inserting and Removing Graphic Objects (GOCA)* on page 88.

'name' Corresponds to the name of a Box object defined using the DEFINEBOX keyword.

level Describes the data level at which the object is to be inserted. Please refer to the DEFINELEVEL keyword and to *3.8.1 Defining data enrichment levels to CIS* for a description of the allowed values.

INSERTCIRCLE **('*name', level*)**

Requests the insertion of a Circle object and defines the level. More information about this data enrichment functionality in CIS can be found under *3.8.5 Inserting and Removing Graphic Objects (GOCA)* on page 88.

'name' Corresponds to the name of a Circle object defined using

the DEFINECIRCLE keyword.

level Describes the data level at which the object is to be inserted. Please refer to the DEFINELEVEL keyword and to *3.8.1 Defining data enrichment levels to CIS* for a description of the allowed values.

INSERTCONTAINER **('*name', level* [, USEREXIT, '*exitName*'])**

Requests the insertion of an AFP Object Container at the requested level. More information about this data enrichment functionality in CIS can be found under *3.8.7 Inserting and Removing AFP Object Containers* on page 94.

'name' Corresponds to the name of an Object Container defined using the DEFINECONTAINER keyword.

level Describes the data level at which the object is to be inserted. Please refer to the DEFINELEVEL keyword and to *3.8.1 Defining data enrichment levels to CIS* for a description of the allowed values.

USEREXIT Indicates that *exitName* contains the name of a user exit program to be invoked every time this INSERTCONTAINER operation is to be done. Specifying USEREXIT causes the CIS normalization process to store all the input data into the internal work file before the conversion process is started.

exitName | *'exitName'*
Specifies the name of a user exit program to be invoked every time this INSERTCONTAINER operation is to be done. The exit program determines the parameters and contents of the container to be inserted. More information about User exits can be found under *5.11 ObjectContainer-insertion exit* on page 311.

Note: Under OS/390 and z/OS CIS:
- the exitName is restricted to 1 to 8 characters.
- exitName without apostrophes are accepted
(done for compatibility with version 4.00.)

Example: INPEXIT ('myExit')

INSERTCURVE **('*name', level*)**

Requests the insertion of a Curve object and defines the level. More information about this data enrichment functionality in CIS can be found under *3.8.5 Inserting and Removing Graphic Objects (GOCA)* on page 88.

‘name’	Corresponds to the name of a Curve object defined using the DEFINECURVE keyword.
level	Describes the data level at which the object is to be inserted. Please refer to the DEFINELEVEL keyword and to *3.8.1 Defining data enrichment levels to CIS* for a description of the allowed values.

INSERTELLIPSE **('*name', level*)**

Requests the insertion of an Ellipse object and defines the level. More information about this data enrichment functionality in CIS can be found under *3.8.5 Inserting and Removing Graphic Objects (GOCA)* on page 88.

‘name’	Corresponds to the name of a Ellipse object defined using the DEFINEELLIPSE keyword.
Level	Describes the data level at which the object is to be inserted. Please refer to the DEFINELEVEL keyword and to *3.8.1 Defining data enrichment levels to CIS* for a description of the allowed values.

INSERTFINISHINGOP **('*name'* [, *'name'* [*, ...*]] *, level* [, USEREXIT, '*exitName*'])**

Requests the insertion of a Finishing Operation and defines the level for it. Specifying this keyword causes all finishing operations (MFC structured fields) that the input file may contain to be removed before the new finishing operations are inserted. More information about this data enrichment functionality in CIS can be found under *3.8.9 Inserting and Removing Finishing Operation triplets (UP³I)* on page 97.

‘name’	Corresponds to the name of one or more finishing operations defined using the DEFINEFINISHINGOP keyword.
level	Describes the data level at which the object is to be inserted. Please refer to the DEFINELEVEL keyword and to *3.8.1 Defining data enrichment levels to CIS* for a description of the allowed values.
USEREXIT	Indicates that *exitName* contains the name of a user exit program to be invoked every time this

INSERTFINISHINGOP operation is to be done. Specifying USEREXIT causes the CIS normalization process to store all the input data into the internal work file before the conversion process is started.

exitName | *'exitName'*

Specifies the name of a user exit program to be invoked every time this INSERTFINISHINGOP operation is to be done. The exit program determines the parameters of the operation to be inserted. More information about User exits can be found under *5.12 UP³I Finishing Operation-insertion exit* on page 314.

Note: Under OS/390 and z/OS CIS:
- the exitName is restricted to 1 to 8 characters.
- exitName without apostrophes are accepted (done for compatibility with version 4.00.)

Depending on the level and the number of finishing operations specified, CIS will identify the AFP structure (e.g. DEG or Medium Map) and will add one MFC structured field per INSERTFINISHINGOP keyword. The number of finishing operations specified (the here listed *name* parameters) determine the number of UP³I finishing triplets (0x8E) that the MFC will contain. For a given level the MFC structured fields and the UP³I finishing triplets are inserted in the same order as the INSERTFINISHING keywords and the names within an INSERTFINISHING keyword are specified in the CIS control file.

INSERTINDEX **('*attributeName*' | x'*attributeName*' , *level, '_sprintfString'* [, *fieldName*] [, ...])**

Requests the insertion of an Index tag and defines the level, format and contents of the index to be inserted.

x'*attributeName*'
'*attributeName*' Specifies a user-defined attribute name to be associated with the index tag. This value can consist of alphanumeric (case sensitive) or hexadecimal data. The length has a range from 1 to 250 for alphanumeric and 1 – 500 for hexadecimal data (primarily intended for UCS-2 and other double-byte encoding schemes).

level Describes the data level at which the object is to be inserted. Please refer to the DEFINELEVEL keyword and to *3.8.1 Defining data enrichment levels to CIS* for a description of the allowed values.

'_sprintfString'

Specifies a template followed by a series of data fields (*fieldName*) and applies to each data field a format identifier contained in the *'_sprintfString'* template. Refer to *4.4.2 Formatting string* on page 161 for a detailed description of the options available.

The string 'USEREXIT' may be used to indicate that *fieldName* contains the name of a user exit program to be invoked every time this INSERTINDEX operation is to be done. The exit program determines the contents of the string to be inserted. Specifying USEREXIT causes the CIS normalization process to store all the input data into the internal work file before the conversion process is started.

fieldName When *_sprintfString* is not 'USEREXIT', this field specifies one or more FIELD keywords that compose the index value. The total length of all fields used in an INSERTINDEX keyword may not exceed 250 otherwise it is truncated. The INPUTCP keyword must specify a *codePageID* other than UNKNOWN, UTF16LE or UTF16BE in case *fieldName* corresponds to a FIELD keyword of type other than LIT, VAR or ATT.

When *_sprintfString* specifies 'USEREXIT', this field contains the name of the user exit program to be invoked. Only one occurrence of *fieldName* is allowed in this case. More information can be found under *5.10 INDEX-insertion exit* on page 309.

INSERTLINE **('*name*', *level*)**

Requests the insertion of a Line object at the requested level. More information about this data enrichment functionality in CIS can be found under *3.8.5 Inserting and Removing Graphic Objects (GOCA)* on page 88.

'name' Corresponds to the name of a Line object defined using the DEFINELINE keyword.

level Describes the data level at which the object is to be inserted. Please refer to the DEFINELEVEL keyword and to *3.8.1 Defining data enrichment levels to CIS* for a description of the allowed values.

INSERTPTOCA **(*'name'* , *level*, *'_sprintfString'* [, *fieldName*] [, ...])**

Requests the insertion of a Presentation Text object and defines the level, format and contents it should have. The inserted string is enclosed in a PTOCA Transparent data (TRN) control sequence.

'name' Corresponds to the name of a PTOCA object previously defined using the DEFINEPTOCA keyword.

level Describes the data level at which the object is to be inserted. Please refer to the DEFINELEVEL keyword and to *3.8.1 Defining data enrichment levels to CIS* for a description of the allowed values.

'_sprintfString' Specifies a template followed by a series of data fields (*fieldName*) and applies to each data field a format identifier contained in the *'_sprintfString'* template. Refer to *4.4.2 Formatting string* on page 161 for a detailed description of the options available.

The string 'USEREXIT' may be used to indicate that *fieldName* contains the name of a user exit program to be invoked every time this INSERTPTOCA operation is to be done. The exit program determines the contents of the string to be inserted. Specifying USEREXIT causes the CIS normalization process to store all the input data into the internal work file before the conversion process is started.

fieldName When *_sprintfString* is not 'USEREXIT', this field specifies one or more FIELD keywords that compose the Presentation Text value. The total length of all fields used in an INSERTPTOCA keyword may not exceed 250 otherwise it is truncated. The INPUTCP keyword must specify a *codePageID* other than UNKNOWN, UTF16LE or UTF16BE in case *fieldName* corresponds to a FIELD keyword of type other than LIT, VAR or ATT.

When *_sprintfString* specifies 'USEREXIT', this field contains the name of the user exit program to be invoked. Only one occurrence of *fieldName* is allowed in this case. More information can be found under *5.8 PTOCA-insertion exit* on page 305.

INSERTPTOCANOP **(*'name'* , *level*, *'_sprintfString'* [, *fieldName*] [, ...])**

Requests the insertion of a non-printable Presentation Text object and defines the level, format and contents it should have. The inserted string is enclosed in a PTOCA No-Operation (NOP) control sequence instead of Transparent data (TRN). Other than that, it works exactly the same as the INSERTPTOCA keyword.

This keyword is intended to be used with the Océ ImageStream archiving software. Please refer to the INSERTPTOCA keyword for information about the INSERTPTOCANOP parameters. Please refer to *5.9 PTOCA-NOP-insertion exit* on page 307 for information concerning the user exit facility.

INSERTSHEET **('*sheetName*' , *level* , '*mMapName*')**

Requests the insertion of a Sheet and specifies the level, Medium Map and optional variable data to be printed on the sheet. Once defined, a *sheetName* may be used in a DEFINELEVEL keyword as entity selector and may be the target of other data enrichment operations such as define/insertBCOCA, define/insertPTOCA, define/insertCONTAINER, define/insertFINISHINGOP, etc.. More information about this data enrichment functionality in CIS can be found under *3.8.8 Inserting and Removing additional Sheets* on page 96.

'sheetName' Is a 1 to 250 characters name which is used to uniquely identify the sheet being defined.

level Describes the data level at which the object is to be inserted. Please refer to the DEFINELEVEL keyword and to *3.8.1 Defining data enrichment levels to CIS* for a description of the allowed values.

'mMapName' Specifies the name of a valid Medium Map which must exist in the FormDef used for the current print file (in first External Resource Group or in an external library) or which is being added using the DEFINEMEDIUMMAP keyword.

JOBID **(<u>0</u> | *value*) *PRISMAproduction Server internal usage only !!!***

LINEMERGE **(<u>NO</u> | 3800)**

Determines which kind of line merging is to be used to convert the input data.

<u>NO</u> Specifies that the 3800 compatible line merging is disabled. The standard AFP line merging (which overprints characters) is used.

3800 Specifies that the 3800 compatible line merging is to be used. In this case, the characters contained in two or more input lines are merged in order to produce one single output line. The merge process is controlled by the carriage control.

LSKEYDD **(<u>CISLSK</u>)**

Specifies the name of the License key file. A description of the Licensing requirements in order to run CIS can be found under *6.1 Appendix A: Licensing Requirements* on page 323.

LUPUB **(<u>0</u> | *value*)**

Not all printers support the same l_units_per_unit_base values. It is sometimes necessary to convert the value(s) used by an application to a value which is known as supported by the target presentation device (e.g. 2400).

value Number of l_units_per_unit_base (LUPUB) supported by the printer for a unit base of 10 inches. A value of 1 to 32767 may be specified. Specifying 0 causes all l_units_per_unit_base values in the input data to be left unchanged.

Example: LUPUB (2400)

MEDIAFIDELITY **(<u>ASIS</u> | NONE | STOP | CONTINUE)**

Controls the way the presentation software (e.g. SPS) will handle input Media exceptions. Causes CIS to insert a PFC structured field containing a Media Fidelity triplet x'87' in all Medium Maps in the form definition being used.

This keyword must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise a warning message is issued.

<u>ASIS</u> Indicates that the current Media fidelity options in the FORMDEF are to be left unchanged. This is the default.

NONE Causes to remove all Media fidelity (triplet x'87') options from the FORMDEF.

STOP Causes the presentation software (SPS) to stop printing the job and to place it in hold in case the printer reports a media exception.

CONTINUE — Causes the presentation software (SPS) to issue a message and to continue printing the job using a substitute media.

MESSAGECP **(*codePageID*)**

Specifies the Code Page to be used when generating CIS messages. An overview on code page support can be found under *3.2 Code page support* on page 33.

codePageId — Specifies the ID of the code Page to be used for translating. See also *3.2.1 Supported Code Page IDs* on page 38. UNKNOWN, UTF16BE, UTF16LE are **not** supported. By default CIS uses the currently active internal Code Page. For more information please refer to *3.2 Code page support* on page 33.

Example: MESSAGECP (ISO_8859-1)

MPCONCEPT **(<u>ON</u> | OFF)**

Specifies whether the Mail Piece concept available in CIS should be used in the current run or not.

<u>ON</u> — Indicates that the Mail Piece concept should be used, and that the top Page Group level must start at a sheet boundary. CIS will issue an error in case it finds a Mail Piece which does not start at a new sheet. Some CIS functions such as SORT and MPCONSOLIDATION require that the Mail concept is enabled.

OFF — Indicates that the Mail Piece concept should be disabled.

MPCONSOLIDATION **(<u>NO</u> | { NESTED , '*attributeName*' | x'*attributeName*' [, '*attributeName*' | x'*attributeName*' [, ...]] } | { INONEGROUP , '*attributeName*' | x'*attributeName*' [, '*attributeName*' | x'*attributeName*' [, ...]] })**

Controls the Mail Piece consolidation processes in CIS. A description is available under *3.6 How to convert, normalize and index* on page 55 and *3.11 How to Consolidate mail pieces* on page 135.

NO	Two or more consecutive Mail Pieces may have the same Index values and are not consolidated into one bigger Mail Piece.
NESTED	Two or more consecutive Mail Pieces with the same Index values are consolidated into one bigger Mail Piece. Existing Page Group structures are kept unchanged, nested inside the newly inserted Mail Piece level. The common Index tags are moved to the new Mail Piece level.
INONEGROUP	Two or more consecutive Mail Pieces with the same Index values and are consolidated into one bigger Mail Piece. Existing Page Group structures are removed. The common Index Tags are moved to the consolidated mail piece
x'*attributeName*' '*attributeName*'	Specifies at least one attribute name to be checked for NESTED and INONEGROUP operations. This value can consist of alphanumeric (case sensitive) or hexadecimal data. The length has a range from 1 to 250 for alphanumeric and 1 – 500 for hexadecimal data (primarily intended for UCS-2 and other double-byte encoding schemes).

The Mail Piece consolidation process in CIS is done after the normalization, and sorting steps, before the data enrichment process (see Fig. 4 on page 320 on page 1). Index tags inserted with the INSERTINDEX keyword are not included in the consolidation process. A second CIS run may be done in this case.

OBJCONEXT **("" | "*extension*" [, "*extension*"] [,...])**

Specifies one or more suffixes to be appended to an AFP Object Container resource name while searching for the corresponding file.

extension A variable length character string starting with a dot. The default is no suffix.

CIS searches in the same order as the extensions are given. The first match is used.

Example: OBJCONEXT (".600", ".300", ".240", ".afp", “”)

The above example causes CIS to search for the following resource names:

name.**600**
name.**300**
name.**240**
name.**afp**
name

OBJCONLIB **(*directory* | *"directory"* [, *directory* | *"directory"*] [, ...])**

Specifies the directories where CIS searches for the Object Container definitions such as Color Mapping table. CIS searches in the same order as the directories are given. Any number of directories can be specified.

directory	Specifies the directory to be searched.
"directory"	This notation must be used when the directory name contains special characters such as blanks.

The order CIS searches for Object Containers is:

Inline resource group
Userlibs if given
Defined directories

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

OFFSETXF **([-] *n* [.*m*] [*unit*])**
OFFSETYF **([-] *n* [.*m*] [*unit*])**
OFFSETXB **([-] *n* [.*m*] [*unit*])**
OFFSETYB **([-] *n* [.*m*] [*unit*])**

Specify the offset in the **X** (**Y**) direction of the logical page origin from the media origin for the **F**ront (**B**ack) side of each sheet. The value specified in the FormDef is used when no OFFSETxx keyword is specified.

The OFFSETxx keywords must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise a warning message is issued.

n	A 1 to 8 digits number.
m	A 1 to 3 digits number.
unit	Any of the units of measurement supported by CIS (see

4.4.6 Units of measurement on page 165 for more information). PELS is the default.

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 keyword must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise a warning message is issued.

Value Specifies a decimal number between 0 and 254

Example: OUTBIN (2)

OUTLINEFONTEXT **(<u>""</u> | "*extension*" [, "*extension*"] [,...])**

Specifies one or more suffixes to be appended to an AFP Outline Font resource name while searching for the corresponding file.

extension A variable length character string starting with a dot. The default is no suffix.

CIS searches in the same order as the extensions are given. The first match is used.

Example: OUTLINEFONTEXT (".600", ".300", ".240", ".afp", "")

The above example causes CIS to search for the following resource names:

*name***.600**
*name***.300**
*name***.240**
*name***.afp**
name

OUTPUTDD **(<u>CISOUT</u> | *fileName*)**

Specifies the name of the file where CIS will write the output document print file(s).

fileName Specifies the Output document print file. In case of executing CIS on the OS/390 or z/OS environment, *fileName* is a JCL dataset definition. If it defines:

- a SYSOUT dataset, make sure that at least the DCB subparameters RECFM, LRECL and BLKSIZE are defined. CIS will not accept any JES defaults. In case of OUTPUTSEG, please pay attention to define DCB subparameters for every defined SYSOUT DD even if they are not used by CIS for writing its output.
- a Sequential Data set (PS) and Output Segmentation (see OUTPUTSEG) is activated, the additional OUTPUTSEGPSDEF keyword can be used to specify the size of the allocated Data sets.
- a Data set mask (see OUTPUTSEG) make sure all Data sets are of the same type and format.

Examples: OUTPUTDD (MYFILEDD)
OUTPUTDD(OUT*****) will use Data sets defined in the step's JCL in incremental sequence order OUT00001, OUT00002, ...

OUTPUTFORMAT **(MODCA | RDW | AFPSTREAM , x'*delimiter*')**

Specifies the data format used in the output and resource files created by CIS as specified by the INDEXDD, OUTPUTDD and RESOBJDD keywords. It applies to all output file(s), including any segmentation that may take place.

MODCA Requests CIS to generate all output and resource files as continuous byte streams containing MO:DCA-P structured fields. The specifications for a MODCA file are the same as described for the INPUTFORMAT keyword.

RDW Under MVS, requests CIS to generate one sequential record for each MO:DCA-P structured field being written. The DCB specifications of the file (see INDEXDD, OUTPUTDD and RESOBJDD keywords) are used to control whether a 0x5A carriage control byte is inserted or not (e.g. RECFM=VBA vs. RECFM=VB). The specifications for an RDW file are the same as described for the INPUTFORMAT keyword.

Under Linux, requests CIS to precede every AFP record written to the Index, Resource and Output file with a 4-bytes

Record Descriptor Word (RDW). The specifications for an RDW file are the same as described for the INPUTFORMAT keyword.

AFPSTREAM , x'*delimiter*'

Similar to MODCA, except that every MO:DCA-P structured field is preceded by a 0x5A carriage control byte and is followed by a delimiter. The specifications for an AFPSTREAM file are the same as described for the INPUTFORMAT keyword.

Example: OUTPUTFORMAT (AFPSTREAM, x'0D0A')

OUTPUTFORMAT (RDW) is the default under MVS.

OUTPUTFORMAT (AFPSTREAM, X'') is the default under Linux indicating that each structured field is preceded by a 0x5a carriage control byte and no delimiter follows.

OUTPUTSEG **({ *entity* {, *count* [, *boundary*] })**

Controls the output segmentation function in CIS. More information about how this function works can be found under *3.5.1 Segmenting Output data* on page *48*

entity Specifies the entity or scope at which the output segmentation is to be applied. The following values may be specified:

<u>DOC</u>	AFP document object
MP	Mail Piece (top level page group)
SH	Sheet
PAG	Page
BYTE	Bytes

count Specifies the number of *entities* that should be included in each output segment or the number of *megabytes* (x'100000' bytes) in case of BYTE. * is the default and indicates that all items are stored in one output segment (in other words, a single output file containing all items is generated). Specifying a value of 0 causes the suppression of output file generation for all levels.

boundary Specifies the entity level that should not be spanned at the end of the single segment files. It has to be at least one level “higher” than the entity parameter. Thus valid values are:

DOC	(entity being MP, SH, PAG or BYTE)
MP	(entity being SH, PAG or BYTE)
SH	(entity being PAG or BYTE)
PAG	(entity being BYTE)

Example: OUTPUTSEG (SH, 2000)

This keyword is rejected when specified together with INDEXOBJ.

See the description of OUTPUTDD for information where the single segments are written.

OUTPUTSEGPSDEF **(*spaceType*, *primary* [, *secondary* [, *volSer*]])**

is used when Output Segmentation (OUTPUTSEG) is activated and the Output Data set (OUTPUTDD) refers to a sequential Data set (PS). Ignored in all other situations. It specifies the allocation size and volume for sequential Data sets. More information about how this function works can be found under *3.5.1 Segmenting Output data* on page 48.

spaceType Specifies one of the following values:

- CYL space type is Cylinder
- TRK space type is Track

primary Specifies the expected primary allocation size. 30 is the default. Your installation might change this value dynamically to it’s needs.

secondary Specifies the expected secondary allocation size. 0 is the default. Your installation might change this value dynamically to it’s needs.

volSer Specifies the Volume where the data set is to be allocated. This parameter must be a 6 characters long string.

Example: OUTPUTSEGPSDEF (TRK, 20, 40, SMS924)

OVERLAYF (*overlayName* [, *overlayName*] [,...])
OVERLAYB (*overlayName* [, *overlayName*] [,...])

These keywords must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise a warning message is issued.

OverlayName Up to 8 names (1 to 8 character) specifying the medium overlays to be placed on the **F**ront or **B**ack side of each sheet, in addition to overlays from other sources.

Example: OVERLAYF (MYOVLY1, MYOVLY2)

OVERLAYEXT ("" | "*extension*" [, "*extension*"] [,...])

Specifies one or more suffixes to be appended to an AFP Overlay resource name while searching for the corresponding file.

extension A variable length character string starting with a dot. The default is no suffix.

CIS searches in the same order as the extensions are given. The first match is used.

Example: OVERLAYEXT (".600", ".300", ".240", ".afp", “”)

The above example causes CIS to search for the following resource names:

*name***.600**
*name***.300**
*name***.240**
*name***.afp**
name

OVLYLIB (*directory* | *"directory"* [, *directory* | *"directory"*] [,...])

Specifies the directories where CIS searches for the Overlay definitions. CIS searches in the same order as the directories are given. Any number of directories can be specified.

directory Specifies the directory to be searched.

"directory" This notation must be used when the directory name contains special characters such as blanks.

The order CIS searches for Overlay definitions is:

Inline resource group
Userlibs if given
Defined directories in OVLYLIB param

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

PAGEDEF **(*pageDefName* | DUMMY)**

A 1 to 8 character name of the page definition to be used while converting S/370 Line format or XML data. The complete name must be specified. No prefix is added by CIS. More information about the CIS data conversion process can be found under *3.6.1 Converting and Indexing print files containing S/370 Line formatted data* on page 56.

Specifying DUMMY (the default) requires the print file to contain at least one inline Pagedef. CIS uses the first Pagedef found. All others are ignored.

Example: PAGEDEF (P1TEST)

PAGEDEFEXT **("" | "*extension*" [, "*extension*"] [,...])**

Specifies one or more suffixes to be appended to an AFP PageDef resource name while searching for the corresponding file.

extension A variable length character string starting with a dot. The default is no suffix.

CIS searches in the same order as the extensions are given. The first match is used.

Example: PAGEDEFEXT (".600", ".300", ".240", ".afp", “”)

The above example causes CIS to search for the following resource names:

name.**600**
name.**300**
name.**240**
name.**afp**
name

PAGESEGEXT **("" | "*extension*" [, "*extension*"] [,...])**

Specifies one or more suffixes to be appended to an AFP Page Segment resource name while searching for the corresponding file.

extension A variable length character string starting with a dot. The default is no suffix.

CIS searches in the same order as the extensions are given. The first match is used.

Example: PAGESEGEXT (".600", ".300", ".240", ".afp", “”)

The above example causes CIS to search for the following resource names:

name.**600**
name.**300**
name.**240**
name.**afp**
name

PDEFLIB **(*directory* | *"directory"* [, *directory* | *"directory"*] [,...])**

Specifies the directories where CIS searches for the Page definitions. CIS searches in the same order as the directories are given. Any number of directories can be specified.

directory Specifies the directory to be searched.

"directory" This notation must be used when the directory name contains special characters such as blanks.

The order CIS searches for Page definitions is:

Inline resource group
Userlibs if given
Defined directories

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

PRESENTATION **(ASIS | PORTRAIT | PORTRAIT90 | PORTRAIT180 | LANDSCAPE | LANDSCAPE90 | LANDSCAPE180)**

Overrides any presentation specification specified in the Form definition. This keyword must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise a warning message is issued.

ASIS	Presentation specified via resource file resources should be left unchanged.
PORTRAIT	Force presentation PORTRAIT.
PORTRAIT90	Force presentation PORTRAIT90.
PORTRAIT180	Force presentation PORTRAIT180.
LANDSCAPE	Force presentation LANDSCAPE.
LANDSCAPE90	Force presentation LANDSCAPE90.
LANDSCAPE180	Force presentation LANDSCAPE180.

Example: PRESENTATION (PORTRAIT)

PRINTMODE **(SOSI1 | SOSI2 | SOSI3 | *string*)**

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.
string	A 1 to 8 character string. May be used as user defined parameter in connection with any of the User exits available.

Example: PRINTMODE (SOSI1)

PSEGLIB **(*directory* | *"directory"* [, *directory* | *"directory"*] [,...])**

Specifies the directories where CIS searches for the Page Segment definitions. CIS searches in the same order as the directories are given. Any number of directories can be specified.

directory	Specifies the directory to be searched.
"directory"	This notation must be used when the directory name contains special characters such as blanks.

The order CIS searches for Page Segment definitions is:

Inline resource group
Userlibs if given
Defined directories

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

REMOVEBCOCA **(*'name'* [, *'name'*] [,...])**

Requests the removal of Bar Code objects already contained in the input print file(s).

'name' Corresponds to the name of a BCOCA object previously inserted using the INSERTBCOCA keyword or any other BCOCA generation method. All Bar Codes with this name will be removed from the data and will not appear in the output file.

CIS checks all Begin Bar Code Object (BBC) structured fields for the presence of a "Fully Qualified Name" triplet (0x02) and compares the *name* parameter with the fully qualified name. The BCdoName field is only used when no triplet 0x02 is found. Trailing blanks (if any) must be explicitly specified.

REMOVEBOX **('*name*' [, '*name*' [, ...]])**

Requests the removal of Box objects already contained in the input print file(s).

'name' Corresponds to the name of a GOCA box object previously inserted using the INSERTBOX keyword. All box objects with this name will be removed from the data and will not appear in the output file.

GOCA Box objects inserted using the INSERTBOX keyword are the only ones that have a name field and that may be removed using this keyword.

REMOVECIRCLE **('*name*' [, '*name*' [, ...]])**

Requests the removal of Circle objects already contained in the input print file(s).

'name' Corresponds to the name of a GOCA circle object previously inserted using the INSERTCIRCLE keyword. All circle objects with this name will be removed from the data and will not appear in the output file.

GOCA Circle objects inserted using the INSERTCIRCLE keyword are the only ones that have a name field and that may be removed using this keyword.

REMOVECONTAINER **('*name*' [, '*name*' [, ...]])**

Requests the removal of Object Containers already contained in the input print file(s).

This keyword must be used together with RESTYPE(ALL) or RESTYPE(OBJCON) otherwise a warning message is issued.

'name' Corresponds to the name of an Object Container previously inserted using the INSERTCONTAINER keyword or any other Object Container generation method. All Object Containers with this name will be removed from the data and will not appear neither in the output file nor in the resource object file. All references to the Container (IOB and MDR structured fields) are also removed. Object Containers directly included in a Page or Overlay are also removed in case of a name match.

REMOVECURVE **('*name*' [, '*name*' [, ...]])**

Requests the removal of Curve objects already contained in the input print file(s).

'name' Corresponds to the name of a GOCA curve object previously inserted using the INSERTCURVE keyword. All curve objects with this name will be removed from the data and will not appear in the output file.

GOCA Curve objects inserted using the INSERTCURVE keyword are the only ones that have a name field and that may be removed using this keyword.

REMOVEELLIPSE **('*name*' [, '*name*' [, ...]])**

Requests the removal of Ellipse objects already contained in the input print file(s).

'name' Corresponds to the name of a GOCA ellipse object previously inserted using the INSERTELLIPSE keyword. All ellipse objects with this name will be removed from the data and will not appear in the output file.

GOCA ellipse objects inserted using the INSERTELLIPSE keyword are the only ones that have a name field and that may be removed using this keyword.

REMOVEFINISHINGOP **('*name*' [, '*name*' [, ...]])**

Causes CIS to search for and to remove the MFC structured field(s) containing the name(s) specified as parameter from the input print file(s).

'name' Corresponds to the name of a Finishing Operation previously inserted using the INSERTFINISHINGOP keyword. May be specified more than once.

MFC structured fields inserted using the INSERTFINISHINGOP keyword are the only ones that have a name field and that may be removed using this keyword.

REMOVEINDEX **('*attributeName*' | x'*attributeName*' [, '*attributeName*' | x'*attributeName*'] [,...])**

Requests the removal of Index Tags already contained in the input print file(s).

x'attributeName'
'attributeName' Corresponds to the name of an Index Tag Element (TLE) previously inserted using the INSERTINDEX keyword, the TRIGGER/INDEX keywords or any other Index Tag generation method. All TLE's with this attribute name will be removed from the data and will not appear in the output file. This value can consist of alphanumeric (case sensitive) or hexadecimal data. The length has a

range from 1 to 250 for alphanumeric and 1 – 500 for hexadecimal data (primarily intended for UCS-2 and other double-byte encoding schemes).

REMOVELINE **('*name*' [, '*name*' [, ...]])**

Requests the removal of line objects already contained in the input print file(s).

'name' Corresponds to the name of a GOCA line object previously inserted using the INSERTLINE keyword. All line objects with this name will be removed from the data and will not appear in the output file.

GOCA Line objects inserted using the INSERTLINE keyword are the only ones that have a name field and that may be removed using this keyword.

REMOVEMEDIUMMAP **('*mMapName*' [, '*mMapName*' [*, ...*]])**

Causes CIS to remove selected Medium Map(s) from the Form Definition specified in the FORMDEF keyword. REMOVEMEDIUMMAP is used in combination with the DEFINEMEDIUMMAP keyword to replace existing Medium Maps. Removing an existing Medium Map without replacing it may cause CIS to stop processing in case the Medium Map is still referenced within the print file.

This keyword must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise a warning message is issued. In case of resource consolidation, the REMOVEMEDIUMMAP keyword is applied to all relevant resource groups.

'mMapName' Specifies the name of the Medium Map to be removed. The Form Definition specified with the FORMDEF keyword is searched and the Medium Map(s) is(are) removed. A 1 to 8 characters name.

REMOVEPTOCA **(*'name'* [, *'name'*] [,...])**

Requests the removal of Presentation Text objects already contained in the input print file(s).

'name' Corresponds to the name of a Presentation Text object (PTOCA) previously inserted using the INSERTPTOCA or INSERTPTOCANOP keywords, or any other PTOCA generation method. All PTOCA elements with this name

will be removed from the data and will not appear in the output file.

CIS checks all Begin Presentation Text Object (BPT) structured fields for the presence of a “Fully Qualified Name” triplet (0x02) and compares the *name* parameter with the fully qualified name. The PTdoName field is only used when no triplet 0x02 is found. Trailing blanks (if any) must be explicitly specified.

REMOVESHEET **('*name*' [, '*name*' [*, ...*]])**

Causes CIS to search for and to remove the sheet(s) containing the name(s) specified as parameter

‘name’ Corresponds to the name of a sheet to be removed. May be specified one or more times.

Sheets inserted using the INSERTSHEET keyword are the only ones that have a name field and that may be removed.

RESCONSOLIDATION **(<u>BASIC</u> | { [EXTERNALRESGROUP] [, INTERNALRESGROUP] [, INTERNALMEDIUMMAP] [, INTERNALMMAPOPTIMIZE]})**

Controls the Resource consolidation processes in CIS. A description is available under *3.7 How to Retrieve and Consolidate AFP Resource* on page 69.

<u>BASIC</u> Basic resource consolidation (compatible to CIS versions prior V4.04) is done.

EXTERNALRESGROUP
Causes CIS to consolidate all the resources used in the input print file and which are read from external resource group(s) part of the same print file, into the single external resource group. This parameter is rejected when specified together with RESTYPE other than ALL. Page Definition resources cannot be consolidated.

INTERNALRESGROUP
Causes CIS to consolidate all the resources used in the input print file and which are read from internal resource group(s) into the single external resource group. This parameter is rejected when specified together with RESTYPE other than ALL. Page Definition resources cannot be consolidated.

INTERNALMEDIUMMAP
Causes CIS to consolidate all the internal Medium Maps used in the input print file into the FORMDEF which is written to RESOBJDD. Naming conflicts are avoided by renaming the affected Medium Maps. The presentation (layout) of the print file is not affected. This parameter is rejected when specified together with RESTYPE other than ALL or FDEF.

INTERNALMMAPOPTIMIZE
Causes CIS to consolidate all the internal Medium Maps used in the input print file into the FORMDEF which is written to RESOBJDD. Before the Medium Maps are saved to FORMDEF they will be compared. If the contens of MediumMaps is identical they will only be stored once in FORMDEF. Naming conflicts are avoided by renaming the affected Medium Maps. The presentation (layout) of the print file is not affected. This parameter is rejected when specified together with RESTYPE other than ALL or FDEF

Example: RESCONSOLIDATION (INTERNALRESGROUP)

The Resource consolidation process in CIS is done parallel to the normalization, sorting and data enrichment steps (see Fig. 4 on page 320 on page 1). Resources inserted as a result of the data enrichment process (e.g. INSERTOBJECT) are also included in the external Resource group written to RESOBJDD.

The presentation (layout) of the print file is not affected. Refer to [18] in *8 Bibliography* on page 465, chapter 4 under "Resource Objects" for more information about the different resource grouping options available in MO:DCA.

RESEXIT **(' *exitName*')**

exitName | *'exitName'*
Specifies the name of the resource exit program. The program will be called for every resource to be written in the output resource file and will be allowed to decide whether the resource should be written into the file or not.

Note: Under OS/390 and z/OS CIS:
- the exitName is restricted to 1 to 8 characters.

- exitName without apostrophes are accepted
  (done for compatibility with version 4.00.)

Example: RESEXIT ('myExit')

More information can be found under *5.4 Resource-selection exit* on page 299.

RESFIDELITY **(<u>STOP</u> | CONTINUE)**

Controls the way CIS handles situations where resource are not available.

This keyword must be used together with RESTYPE(ALL) otherwise a warning message is issued.

<u>STOP</u>	Causes CIS to stop processing the print file in case a resource is not found in any of the libraries and resource groups available.
CONTINUE	Causes CIS to issue a message and to ignore the missing resource; no substitution is done by CIS.

RESFILE **(<u>SEQ</u> | PDS)**

Specifies in which format the used resources should be saved.

<u>SEQ</u>	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 **(<u>CISRES</u> | *fileName* | "*fileName*" | *directory* | "*directory*")**

Specifies the name of the file or directory where CIS will write the output resource file(s). A *directory* must be specified when RESFILE(PDS) is also specified otherwise a *fileName* is expected.

directory *fileName*	Specifies the Resource Object file.

"*directory*"
"*fileName*" This notation must be used when the name contains special characters such as blanks.

Example: OUTPUTDD ("..\..\myResFile")

RESPRUNING **(<u>ALL</u> | NONE | OVLY)**

Specifies if Resource Pruning is enabled. The default is ALL. CIS prunes data and resources. This keyword cannot be used in combination with OUTPUTSEG(COUNT = 0).

<u>ALL</u> Specifies Resource Pruning is enabled for all kind of resources. All unused resource-mapping operations and also the related resources are eliminated by CIS.

NONE Resource Pruning is disabled.

OVLY Only overlays referenced by a MPO Structured Field with a corresponding IPO Structured Field are written to the CISRES file. Also MPO Structured Fields without a corresponding IPO Structured Field will be discarded from the output.

Attention:

After a CIS-run with Resource Pruning other than NONE, the following applications, like printing or file-transfers, have to use the generated CISOUT- and CISRES-File(s).

RESTYPE **(<u>NONE</u> | { ALL [,PDEF] } | { [FDEF] [,PDEF] [,PSEG] [,OVLY] [,FONT] [,OBJCON][,BCOCA] [,GOCA] [,IOCA] })**

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

<u>NONE</u> No resource file will be created.
ALL All resources are saved in RESOBJDD. The Page

	definition is not included and must be specified extra.
FDEF	The Form definition is saved in RESOBJDD.
PDEF	The Page definition is saved in RESOBJDD.
PSEG	All Page segments are saved in RESOBJDD.
OVLY	All Overlays are saved in RESOBJDD.
FONT	All font character sets, code pages and coded fonts are saved in RESOBJDD.
OBJCON	All Object containers are saved in RESOBJDD.
BCOCA	All BCOCA objects included via IOB are saved in RESOBJDD.
GOCA	All GOCA objects included via IOB are saved in RESOBJDD.
IOCA	All IOCA objects included via IOB are saved in RESOBJDD.

CIS issues a warning message in case a RESTYPE other than ALL or FDEF is specified together with any of the following keywords:

COLORFIDELITY	DEFINEFINISHINGOP
DEFINEMEDIUMMAP	DUPLEX
FINISHINGFIDELITY	FONTFIDELITY
INBIN	MEDIAFIDELITY
OFFSETxx	OUTBIN
OVERLAYx	PRESENTATION
REMOVEMEDIUMMAP	SETUPVID
SORT (FLIP_FOR_COLLATED)	SORT (FLIP_FOR_UNCOLLATED)
TONERFIDELITY	X2UP
RESCONSOLIDATION (INTERNALMEDIUMMAP)	

CIS issues a warning message in case a RESTYPE other than ALL or OBJCON is specified together with any of the following keywords:

DEFINECONTAINER	REMOVECONTAINER

CIS issues an error message in case a RESTYPE other than ALL is specified together with any of the following keywords:

RESCONSOLIDATION (INTERNALRESGROUP)
RESCONSOLIDATION (EXTERNALRESGROUP)

CIS issues a warning message in case a RESTYPE other than ALL is specified together with any of the following keywords:

RESFIDELITY

Example:

RESTYPE (ALL, PDEF)

Note: The PageDef resource is not required in order to print the output document file generated by CIS. The PDEF parameter is primarily intended for applications requiring a complete resource package. The Page definition resource is copied into the output resource file and will not reflect any of the data enrichment and/or sorting operations that may have been requested.

exit interface.

SELINDEXRANGE **(*entity*, '*attributeName*' | x'*attributeName*' , [!]'*from*'-'*to*' | [!]x'*from*'-x'*to*' [, [!]'*from*'-'*to*' | [!]x'*from*'-x'*to*'] [, ...])**

Requests a partial processing of the input data, based on index ranges that are to be selected and/or excluded. Can only be specified once. For more information please refer to *Index-range based selection* on page 53.

entity Specifies the entity or scope at which the Input selection is to be applied. The following values may be specified:

DOC	AFP document object
MP	Mail Piece (top level page group)
SH	Sheet
PAG	Page

x'*attributeName*'

'*attributeName*' Specifies the name of an Index Tag Element (TLE) previously inserted using the INSERTINDEX keyword, the TRIGGER/INDEX keywords or any other Index Tag generation method. The *from-to* selection range will be applied to the data in this index entry. This value can consist of alphanumeric (case sensitive) or hexadecimal data. The length has a range from 1 to 250 for alphanumeric and 1 – 500 for hexadecimal data (primarily intended for UCS-2 and other double-byte encoding schemes).

! Indicates that the *from-to* range specification shall be excluded.

x'*from*'-x'*to*'
from-to Specify the start and end item in a range. Any number of selection/exclusion *from-to* parameters may be specified. Matching at least one of the ranges causes a particular item to be selected. A matching condition occurs when the binary comparison of this parameter with the contents of the TLE is equal. The length specified in the *from-to* parameters is used for this purpose.

Please be aware that the CIS normalization step converts all UTF16LE data to UTF16BE. Any string entered using the hexadecimal notation must specify the data in UTF16BE encoding.

Example: SELINDEXRANGE(MP, 'POST OFFICE', '75431'-'85435')

SELRANGE **(*entity,* [!] *from-to* [, [!] *from-to*] [, ...])**

Requests a partial processing of the input data, based on ranges that are to be selected and/or excluded. Can only be specified once. For more information on range based selection with CIS please refer to *Entity-range based selection* on page 52.

The SELRANGE and SELRANGERANDOM keywords are mutually exclusive and may not be specified together.

entity Specifies the entity or scope at which the Input selection is to be applied. The following values may be specified:

DOC	AFP document object
MP	Mail Piece (top level page group)
SH	Sheet
PAG	Page

! Indicates that the *from-to* range specification shall be excluded.

from-to Specify the start and end item in a range. Any number of selection/exclusion *from-to* parameters may be specified. Matching at least one of the ranges causes a particular item to be selected.

Example: SELRANGE(SH, 2001-4000, 6001-8000)

SELRANGERANDOM **(*entity,* [!]*from-to*, *amount* [, *seed*])**

Similar to SELRANGE except that the number of the *from-to* ranges specified by *amount* is randomly generated. Can only be specified once. The *from-to* parameter may be used to restrict the generation to a particular portion of the input file. The *seed* parameter may be used to obtain reproducible results. For more information on range based selection with CIS please refer to *Entity-range based selection* on page 52.

The SELRANGE and SELRANGERANDOM keywords are mutually exclusive and may not be specified together.

entity Specifies the entity or scope at which the Input selection is to be applied. The following values may be specified:

DOC	AFP document object
MP	Mail Piece (top level page group)
SH	Sheet
PAG	Page

! Indicates that the following *from-to* range specification is for Exclusion.

from-to Specifies the start and end item of a range where the random selection will be applied.

amount Specifies the number of selection ranges to be generated.

seed Specifies the seed to be used in the random generation process.

SELSIZE **(*entity, operator, size, unit* [*, operator, size*])**

Requests a partial generation of the output data, based on the size of the items. Refer to *Entity-size based selection* on page 54 for more information.

entity Specifies the entity or scope at which the Input selection is to be applied. The following values may be specified:

DOC	AFP document object
MP	Mail Piece (top level page group)

operator Specifies the compare operator to be used. The values allowed are:

EQ	equal
NE	not equal
LT	less than
LE	less or equal than
GE	greater or equal than
GT	greater than

size Specifies the size (number of items specified in *unit*) criteria to be applied to the comparison in order to select an *entity*.

unit Specifies the size unit.

MP	Mail Piece (top level page group)
SH	Sheet
PAG	Page

Specifying a second operator and size is handled as an AND condition. Any number of SELSIZE parameters may be specified.

Example: SELSIZE(MP, GE, 11, SH, LE, 20)

SETUPCHARS Restricted for usage in combination with MVS ROUTER

(*fontName* [<mxm>] [, *fontName* [<mxm>]] [,...])

Specifies the member name of the coded font (s) to be used to process an S/370 Line format or AFP Mixed data file. It is ignored for MO:DCA-P files. The matrix memory position <mxm> may be specified too. This list overrides any fonts specified in the Pagedef.

fontName A 1 to 8 character coded font name(s) including the font prefix (e.g. 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 be allowed to 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: SETUPCHARS (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.

SETUPFORMDEF Restricted for usage in combination with MVS ROUTER

(*formDefName* | <u>DUMMY</u>)

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 (the default) requires the print file to contain at least one inline FormDef. CIS uses the first FormDef found. All others are ignored.

Example: SETUPFORMDEF (F1MYFORM)

SETUPPAGEDEF Restricted for usage in combination with MVS ROUTER

(*pageDefName* | <u>DUMMY</u>)

A 1 to 8 character name of the page definition to be used while converting S/370 Line format or XML data. The complete name must be specified. No prefix is added by CIS. More information about the CIS data conversion process can be found under *3.6.1 Converting and Indexing print files containing S/370 Line formatted data* on page 56.

Specifying DUMMY (the default) requires the print file to contain at least one inline Pagedef. CIS uses the first Pagedef found. All others are ignored.

Example: SETUPPAGEDEF (P1TEST)

SETUPVID

(<u>ASIS</u> | NONE | { *n* [, *n* [, ...]] })

Specifies one or more Setup Verification ID(s) to be used for printing the job. Causes CIS to replace the Presentation Subsystem Setup ID (keywords X'B4' and X'B5') in the MMC structured field of all Medium Maps in the form definition being used.

This keyword must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise a warning message is issued.

ASIS	Indicates that the Setup Verification Ids in the FORMDEF are to be left unchanged. This is the default.
NONE	Indicates that the Setup Verification Ids in the FORMDEF (if any) are to be removed.
n	Specifies the Setup Verification ID. One or more values between 1 and 65535 may be specified. Not all printers support this function. The presentation software (SPS) compares this value(s) with the value(s) specified in the printer's panel before printing the job.

SORT **(NONE | SCHEME_1 | SCHEME_2 | SCHEME_3 | 4UP_BOOKLET | 2UP_BOOKLET | 2UP_MP_BOOKLET | FLIP_FOR_COLLATED | FLIP_FOR_UNCOLLATED | FOR_UNCOLLATED | USEREXIT, '*program name*' | ASC, '*attributeName*' | x'*attributeName*' [,'*attributeName*' | x'*attributeName*'] [,...] | DES ,'*attributeName*' | x'*attributeName*' [,'*attributeName*' | x'*attributeName*'] [,...])**

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. They are specified in the INDEX keyword (in connection with Line Format data) or to index values already present in the input file (AFP data).

A user written program name may also be specified as parameter. In this case, CIS invokes the exit program a lets it reorder the page, sheet, mail piece and/or document sequence under the control of a user defined sorting criteria.

NONE	no sorting is to be done
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	For fan-fold printers. Print file is prepared for 4up booklet printing. See also *4-UP booklet sorting (4UP_BOOKLET)* on page 123.

2UP_BOOKLET	For fan-fold and cut-sheet printers. The print file is prepared for 2up booklet printing. See also *Two-up Booklet sorting (2UP_BOOKLET and 2UP_MP_BOOKLET)* on page 126
2UP_MP_BOOKLET	For fan-fold and cut-sheet printers. Each mail piece in the print file is prepared for 2up booklet printing. See also *Two-up Booklet sorting (2UP_BOOKLET and 2UP_MP_BOOKLET)* on page 126
FLIP_FOR_COLLATED	
	The job pages (including the medium maps used) are reordered so that -when using a collated stacker- the front page of the job is facing out in the last printed sheet, and the complete job is collated. This parameter must be specified together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise a warning message is issued. See also *Flip Job for collated stacker* on page 130.
FLIP_FOR_UNCOLLATED	
	The job pages (including the medium maps used) are reordered so that -when using an uncollated stacker- the front page of the job is facing out in the first printed sheet, and the complete job is collated. This parameter must be specified together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise a warning message is issued. See also *Flip Job for uncollated stacker* on page 133.
FOR_UNCOLLATED	
	The job pages are reordered so that -when using an uncollated stacker- the front page of the job is facing out in the last printed sheet, and the complete job is collated. See also *For uncollated stacker* on page 132.

ASC	Output will be sorted in ascending index sequence.
DES	Output will be sorted in descending index sequence.
'*attributeName*'	
x'*attributeName*'	Specifies a user-defined attribute name used to identify indexing information in the file. This value can consist of alphanumeric (case sensitive) or hexadecimal data. The length has a range from 1 to 250 for alphanumeric and 1 – 500 for

hexadecimal data (primarily intended for UCS-2 and other double-byte encoding schemes). When sorting S/370 line format data, *attributeName* must match with an ´attribute name´ specified in an INDEX keyword. In case of more than one *attributeName*, the highest sorting level is done with the first name, the lowest level with the last name specified.
Please be aware that the CIS normalization step converts all UTF16LE data to UTF16BE. Any string entered using the hexadecimal notation must specify the data in UTF16BE encoding.

USEREXIT	An exit program will be invoked.
‘*program name*’	Specifies the name of a user-written program that is invoked instead during CIS sorting process. More information about User exits can be found under *5.13 Sort exit* on page 315.

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

CIS uses a temporary work file for sorting. More information about the CISSWAP work file can be found under *“TMPDIR” below*. More information about sorting can be found under *3.10 How to Sort and Reorder output pages* on page 114.

TMPDIR **(*path* | "*path*")**

Specifies the path for temporary working files. This parameter is only processed under Linux.

path Specifies the path to be used for writing temporary working files.

"path" This notation form must be used when the path name contains special characters such as blanks.

Example: TMPDIR(/u/tmp)

Consider to have at least 2 times space left compared to size of input data in the file system which contains your TMPDIR (use the Linux system command df to get information about your local file systems). If CIS is writing its output data to the same file system it has to be even bigger.
Consider CIS to have full access to the directory specified by TMPDIR.

In all cases CIS having problems accessing, creating or writing to TMPDIR it provides a message. Check for enough free space and access rights.

TONERFIDELITY **(ASIS | NONE | *n*)**

Specifies the toner saver function to be used for printing the job. Causes CIS to insert a PFC structured field containing a Toner Saver triplet x'74' in all Medium Maps in the form definition being used.

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

ASIS	Indicates that the current toner saver function options in the FORMDEF are to be left unchanged. This is the default.
NONE	Causes CIS to remove all Toner Saver (triplet x'74') options from the FORMDEF.
n	Specifies the level of toner saver function. One of the following values may be specified: 0x00 Deactivate toner saver function 0x01 Activate toner saver function 0xFF Use device default setting Not all printers support more than one toner saver function. Refer to the printer's documentation for more information about the supported saver function values.

TRACEDD **(CISTRA1 | *fileName* | "*fileName*" [, CISTRA2 | *fileName* | "*fileName*"])**

Specifies the name of the trace files.

fileName	Specifies the name to be used for writing the trace files generated by CIS.
"fileName"	This notation form must be used when the directory name contains special characters such as blanks.

Example: TRACEDD ("..\..\myDir\T1.BIN", "..\..\myDir\T2.BIN")

TRACELEVEL **(<u>0</u> | *value* [, *value*] [,...])**

Specifies the level of trace information to be generated.

Value	Can be 1 to 6 numbers, where number has a range from 0 to 5.
0	All TraceFlags are set to 0x00 (OFF). NO trace is written, no trace files are allocated even if they are specified with `TRACEDD(..)`.

Example: TRACELEVEL (1, 3, 5)

Meanings of the trace levels

Type	Layers	Size	Frequency	Level
Automatic flow trace (TR_MEMBER_...)	**1-4**	**Small**	**Very high**	**5**
External data trace	**3**	**Large**	**Moderate**	**4**
Input (AFPDS)				
Output (AFPDS/IPDS)				
Resources				
Internal data trace	**1-4**	**Medium**	**High**	**3**
Structures				
Private/Public attributes				
States				
Internal execution flow	**3-4**	**Small**	**Moderate/ High**	**2**
Component relevant “state changes”				
Component relevant “actions”				
Component relevant “events”				
Main execution flow	**3-4**	**Small**	**Low/ /Moderate**	**1**
CIS relevant “state changes”				
CIS relevant “actions”				
CIS relevant “events”				

Level 1 (Program error)

Level 1 traces all information which lead to possible program terminations. The trace entries accord to the messages. E.g.: "Invalid Triplet 'X02'!". Very little information is provided inadequately for error analysis.

Level 2 (Program error)

Level 2 traces the program stream in a rough manner – “Which functions/methods are running”, “Which if branch is running”, ”Which event/value leads to a possible program termination".

Developer-Info:

Level 2 traces all trace messages and trace dumps generated with `TR_MSG2()` & `TR_DUMP2()`.

Level 3 (internal data error – Interface errors)

Level 3 traces all data being sent between different modules. It is useful for localization where data got corrupted.

Developer-Info:

Level 3 traces all trace messages and trace dumps generated with `TR_MSG3()` & `TR_DUMP3()`.

Level 4 (external data error)

Level 4 should be used when there are suspected input data errors. All input buffers are traced (TRACETYPE must be set to 'ALL', 'IM', or 'LAYER3'). For keeping the trace small following settings are useful:

```
TRACELEVEL( 4 )
```

```
TRACETYPE( IM )
```

Developer-Info:

Level 4 traces all trace dumps generated with `TR_DUMP( 4, ...)`. `TR_MSG4()` does not exist. As all input data are traces completely customer data may be regained.

Level 5 (flow trace)

Level 5 traces the program flow. Every function/method entry and return (with its return value) is provided independent of the specified TRACETYPE.

Trace information is written in the directory specified in the TRACEDD keyword. More information about these files can be found under *3.14* ***Fehler! Kein gültiges Resultat für Tabelle.*** on page 144.

The trace files generated by CIS are independent from the record/block structure available under MVS. They may be transferred between systems using protocols such as NJE, FTP or 3270 file transfer facilities. They must be transferred in Binary mode.

Warning: performance degradation will occur when this keyword is used in production mode.

TRACESIZE **(*size* | 100)**

Specifies the size (in megabytes) that each trace file is allowed to have. This keyword is ignored under MVS.

size Size of the trace file in MB. Only one trace file is written. **0** indicates that the size is unlimited.

Example: TRACESIZE (1024)

TRACETYPE **(ALL | { [,EX] [,BQ] [,LM] [,LK] [,SM] [,PB] [,AF] [,EV] [,QM] [,TG] [,TM] [,TS] [,ME] [,SV] [,SC] [,IM] [,OM] [,RM] [,PM] [,MO] [,RP] [,JP] [,SF] [,DN] [,CV] [,IT] [,LAYER1] [,LAYER2] [,LAYER3] [,LAYER4])**

Specifies the type(s) of trace entries to be included in the trace file generated by CIS. The supported types are:

EXException Handling (Layer 1)

BQBasic Queuing (Layer 1)

LMLI Interface Management (Layer 1)

LKLock facility (Layer 2)

SMStorage Management (Layer 2)

PBPS basic (Layer 2)

AFArray facility (Layer 2)

EVEvent Handling (Layer 2)

QMQueue ManagementLayer 2)

TGThread Global StorageLayer 2)

TMThread ManagementLayer 2)

TSTimer facilityLayer 2)

MEMessage facilityLayer 2)

SVServer facilityLayer 2)

SCStorage CleanerLayer 3)

IMInput ManagementLayer 3)

OMOutput ManagementLayer 3)

RMResource ManagementLayer 3)

PMPage ManagementLayer 3)

MOModifiers (Sort)Layer 3)

RPResource ProcessingLayer 4)

JPJob ProcessingLayer 4)

SFStructured Fields GenerationIT, Layer 4)

DNData NormalizerIT, Layer 4)

CVAFP ConverterIT, Layer 4)

ITAll Input Transformers’ Traceflags will be setIT, Layer 4)

LAYER1

LAYER2

LAYER3

LAYER4

ALLAll Traceflags will be set inclusive program flow information

Example: TRACETYPE (IT, RM)

TRCTYPE **(NO | IBM | SNI)**

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

NO	No table reference characters
IBM	IBM table reference characters
SNI	SNI table reference characters

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. May be specified more than once.

Specifying the TRIGGER keyword causes CIS to remove and/or replace any existing Document, Page Group and TLE structured fields (BDT, EDT, BNG, ENG) in the input file. Page level TLE's, if any, are left unchanged. The INDEX and MPCONSOLIDATION keywords control the way CIS generates its new Page Group structures.

triggerName	A 1 to 250 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 cannot be *. Any number of record references may be specified.
column | *	Will be used as offset (in characters) 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.
'*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 2 – 500 for hexadecimal data. This is a case-sensitive field. In case of hexadecimal notation and 16-bit (double-byte) characters, the byte order used in *value* must match with the byte order used in the input file (little-endian or big-endian). No endian conversion is done by CIS in this case.

Refer to *3.6 How to convert, normalize and index* on page 55 for more information about the indexing process.

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

UNIQUEBNGS **(<u>YES</u> | NO)**

Specifies whether CIS creates a unique group name or not.

<u>YES</u>	CIS generates an 8-digit numeric string and appends it to the group name.
NO	No string is appended.

Example: UNIQUEBNGS (NO)

This keyword 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 BPG's with FQN's, 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 **(*directory* | *"directory"* [, *directory* | *"directory"*] [, ...])**

Specifies the directories where CIS searches for the resource definitions. CIS searches in the same order as the directories are given. Any number of directories can be specified.

directory Specifies the directory to be searched.

"directory" This notation must be used when the directory name contains special characters such as blanks.

The order CIS searches for resource definitions is:

Inline resource group
Userlibs if given
Defined directories

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

WORKDD **(CISSWAP)**

In case of MVS, the current CIS version uses MVS hiperspaces as internal work file. The specification of a CISSWAP DDname and the allocation of a VSAM data set is not required.

The WORKDD keyword is ignored.

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

Select whether the two up feature should be used or not. 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 keyword must be used together with RESTYPE(ALL) or RESTYPE(FDEF) otherwise a warning message is issued.

ASIS	X2UP specified via resource files should be left unchanged.
OFF	Set to one up.
ON	The same as LEFT
LEFT	Use the two up feature with a left-right sequence.
RIGHT	Use the two up feature with a right-left sequence.
ICOPIES	Use the two up feature to print two copies of each input page on each physical page.

Example: X2UP (ON)

5 User exit facility

The User exit (also called plug-in) facility available in CIS allows user-written programs to obtain control at specific points within the CIS workflow, and allows them to participate in the data conversion (INPEXIT), resource packaging (RESEXIT), indexing (INDEXEXIT), sorting (SORT), data enrichment (INSERTBCOCA, INSERTCONTAINER, INSERTINDEX, INSERTPTOCA) process.

This chapter describes the functionality and the Application Programming Interface used in each of the exit points available in CIS under the following environments:

- OS/390 and z/OS
  - with SAS-C/C++ compiler Version 7.00
  - with OS/390 High Level Assembler (t.b.d.!)
  - other OS/390 C/C++ compiler (IBM) (t.b.d.!)
  - with other ($MAINO) entry (t.b.d.!)

5.1 General technical guidelines

Before getting platform dependent some general principles:

- C++ exceptions and signals thrown by the Plug-in must be caught by the Plug-in. Be aware that CIS deactivates signal handling before calling a Plug-in API function or returning to a Plug-in API function. Thus a signal thrown by the Plug-in – and not caught by it - will lead to an ABEND (OS/390).
- To avoid recursions, the CIS API functions do not call Plug-in API functions.
- Exit programs may be written in assembler or in any other language where the compiled module supports the above mentioned linkage conventions (e.g. C or C++ programs compiled with SAS/C V6.00 or higher and using entry point $MAINO).
- In case of OS/390 or z/OS, the exit program must support AMODE(31) or must switch back to AMODE(31) before returning to CIS. It must also be re-entrant.
- **Unpredictable results may occur if an exit program does not fulfill all these requirements.**

5.1.1 IBM-OS/390 with SAS-C compiler

Some notes for writing a CIS Plug-in with the SAS-C compiler. For further details refer to the SAS documentation.

- <u>**SAMPLE CODE**</u>

The Plug-in has to provide an entry function called `_dynamn()`. This function simply has to return the address of the PluginInit function. See the code below.

```
// Plugin.c
#include <dynam.h>
extern "C"
{
 int _dynamn( void** _pCisAPIFunctions,
        int   _nCisAPIFunctions,
        int   _nInterfaceVersionReq,
        void*  _pCisAPIContext,
        void*** _pPlugInFunctions,
        int*  _nPlugInFunctions,
        int*  _nInterfaceVersionConf,
        void** _pPlugInContext )
 {
  return PluginInit( _pCisAPIFunctions,
            _nCisAPIFunctions,
            _nInterfaceVersionReq,
            _pCisAPIContext,
            _pPlugInFunctions,
            _nPlugInFunctions,
            _nInterfaceVersionConf,
            _pPlugInContext );
 }

int PluginInit( void** IN _pCisAPIFunctions,
        int   IN _nCisAPIFunctions,
        int   IN _nInterfaceVersionReq,
        void*  IN _pCisAPIContext,
        void*** OUT _pPlugInFunctions,
        int*  OUT _nPlugInFunctions,
        int*  OUT _nInterfaceVersionConf,
        void** OUT _pPlugInContext );
```

Fig. 78 Sample _dynamn() code

- **COMPILER SETTINGS**

We recommend using the SAS-C provided procedures LCXXC to compile and LCXXL to link the Plug-in. Following JCL snippets are for compiling and linking.

```
//COMPLINK JOB MSGCLASS=...,CLASS=...
// JCLLIB ORDER=(SASC.C700.PROCLIB)
//*-------------------------------------------------------------*/
//* SASC COMPILE
//*-------------------------------------------------------------*/
//COMPILE EXEC PROC=LCXXC,
// PARM.X=('SN(PLUGIN) AU ENX EXT HX IX RENT',
// 'OMD SOU DO EXCE REFDEF OPTIM SNAME(DYNAM)'),
// REGION=0K
//X.SYSPRINT DD DSN=&&LIST1,DISP=(MOD,PASS)
//X.SYSTERM  DD DSN=&&LIST2,DISP=(MOD,PASS)
//X.SYSTRIN  DD DSN=<Source Library>(PLUGIN),DISP=SHR
//X.SYSLIB  DD DSN=SASC.C700.MACLIBC,DISP=SHR
//      DD DSN=SASC.C700.RW.MACLIBC,DISP=SHR
//X.H     DD DSN=<Include Library>,DISP=SHR
//X.SYSLIN  DD DSN=<Object Library>(PLUGIN),DISP=SHR
//*
//*-------------------------------------------------------------*/
//* SASC LINK
//*-------------------------------------------------------------*/
//LINK EXEC PROC=LCXXL,REGION=0K,ALLRES=YES,
// PARM.LKED=('AMODE=31,RMODE=24,RENT,LIST,MAP,NOTERM,PREM,PRMAP',
// 'XREF')
//LKED.SYSDBLIB DD DSN=&&DBGLIB,SPACE=(4080,(2000,2000,100)),
//    DCB=(RECFM=U,BLKSIZE=4080),DISP=(,PASS),UNIT=VIO
//LKED.SYSARLIB DD DSN=SASC.C700.RW.LIBSTD.A,DISP=SHR
//       DD DSN=SASC.C700.LIBCXX.A,DISP=SHR
//LKED.SYSLMOD DD DSN=<Load library for the Plugin>,DISP=SHR
//LKED.SYSLIN  DD UNIT=DISK,DSN=&&LKEDIN,SPACE=(3200,(1000,500)),
//    DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),VOL=SER=SPSCI1
//LKED.SYSPRINT DD SYSOUT=*
//LKED.SYSTERM DD SYSOUT=*
//LKED.OBJECT  DD DSN=<Object Library>,DISP=SHR
//LKED.SYSIN  DD *
//*   INLCUDE OBJECT(PLUGIN)
//*   ENTRY DYN
//       DD *
 NAME PLUGIN(R)
/*
//
```

Fig. 79 Sample JCL for compiling and linking an SAS-C exit

5.1.4 Plug-in Start-up

This section describes the Plug-in start-up/initialization phase and routines which should be the same for all CIS provided Plug-in APIs. After CIS has loaded a Plug-in module it interchanges following information with the Plug-in:

- The API functions CIS provides for this Plug-in type
- The number of API functions CIS provides for this Plug-in type
- The interface version number for this specific Plug-in
- The transparent context parameter which has to be passed by the Plug-in to CIS in every subsequent API function call.
- The API functions the Plug-in provides for this Plug-in type
- The number of API functions the Plug-in provides for this Plug-in type
- The interface version number for this specific Plug-in
- The transparent context parameter which has to be passed by CIS to the Plug-in in every subsequent API function call.

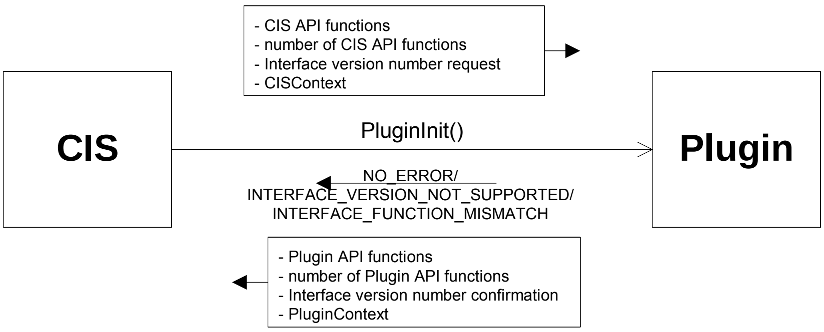


Fig. 83 Plug-in Start-up

This is done by CIS by calling a function called `PluginInit()` which has to be provided by every plug-in. Here is its formal declaration:

```
int PluginInit( void** IN _pCisAPIFunctions,
        int   IN _nCisAPIFunctions,
        int   IN _nInterfaceVersionReq,
        void*  IN _pCisAPIContext,
        void*** OUT _pPlugInFunctions,
        int*  OUT _nPlugInFunctions,
```

```
            int*  OUT _nInterfaceVersionConf,
            void** OUT _pPlugInContext );
```

Fig. 84 Sample PluginInit() code

The purpose of the IN and OUT – empty – defines are to clarify, from the plug-in's perspective, which parameters are passed in which direction. Additional the Plug-in has a return value to notify its internal status:

```
// Return codes
#define NO_ERROR             0
#define INTERFACE_VERSION_NOT_SUPPORTED 1
#define INTERFACE_FUNCTION_MISMATCH   2
```

Fig. 85 PluginInit() return codes

5.1.5 Interface version number handshake

CIS and the Plug-in agree upon an interface version number by using the following handshake/protocol:

- With the `PluginInit()` function call CIS requests an interface version number by passing its highest provided interface with the `_nInterfaceVersionReq` parameter.
- If the Plug-in confirms with this interface version it returns the same value with the `nInterfaceVersionConf` parameter and sets the return value to `NO_ERROR`. The handshake is completed at this moment. This scenario is depicted in the following graphic.

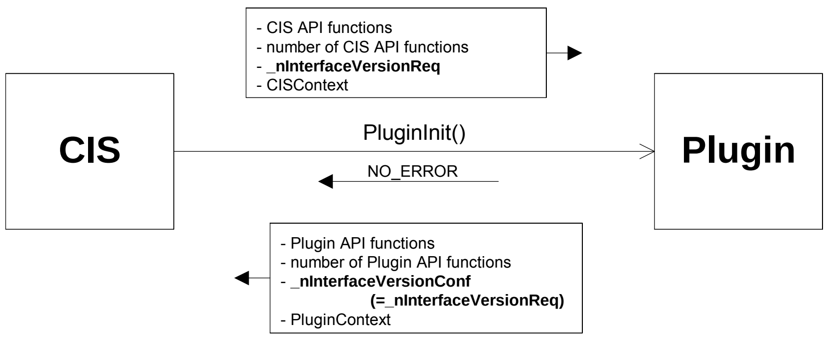


Fig. 86 Interface handshake: normal case

- If the Plug-in does not confirm with the CIS requested interface version number it returns by itself requesting an interface version in the `nInterfaceVersionConf` parameter and by setting the return value to `INTERFACE_VERSION_NOT_SUPPORTED`. If CIS supports this interface version it again does the `PluginInit()` function call with the `_nInterfaceVersionReq` parameter set to the value requested by the Plug-in (this scenario is depicted by following graphic). If CIS does not support this interface version it passes an appropriate message to console and terminates.

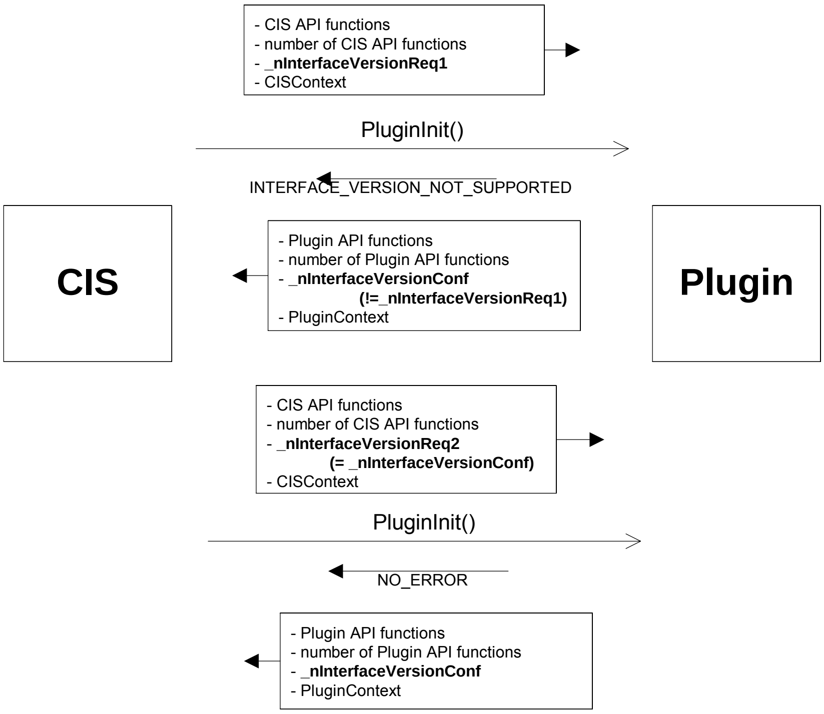


Fig. 87 Interface handshake: not supported case

- If the Plug-in recognizes a mismatch between the CIS passed interface version number `_nCisAPIFunctions` and the number of CIS API functions `_nInterfaceVersionReq` parameter it will return with `INTERFACE_FUNCTION_MISMATCH`. CIS then passes an appropriate message to the console and terminates.

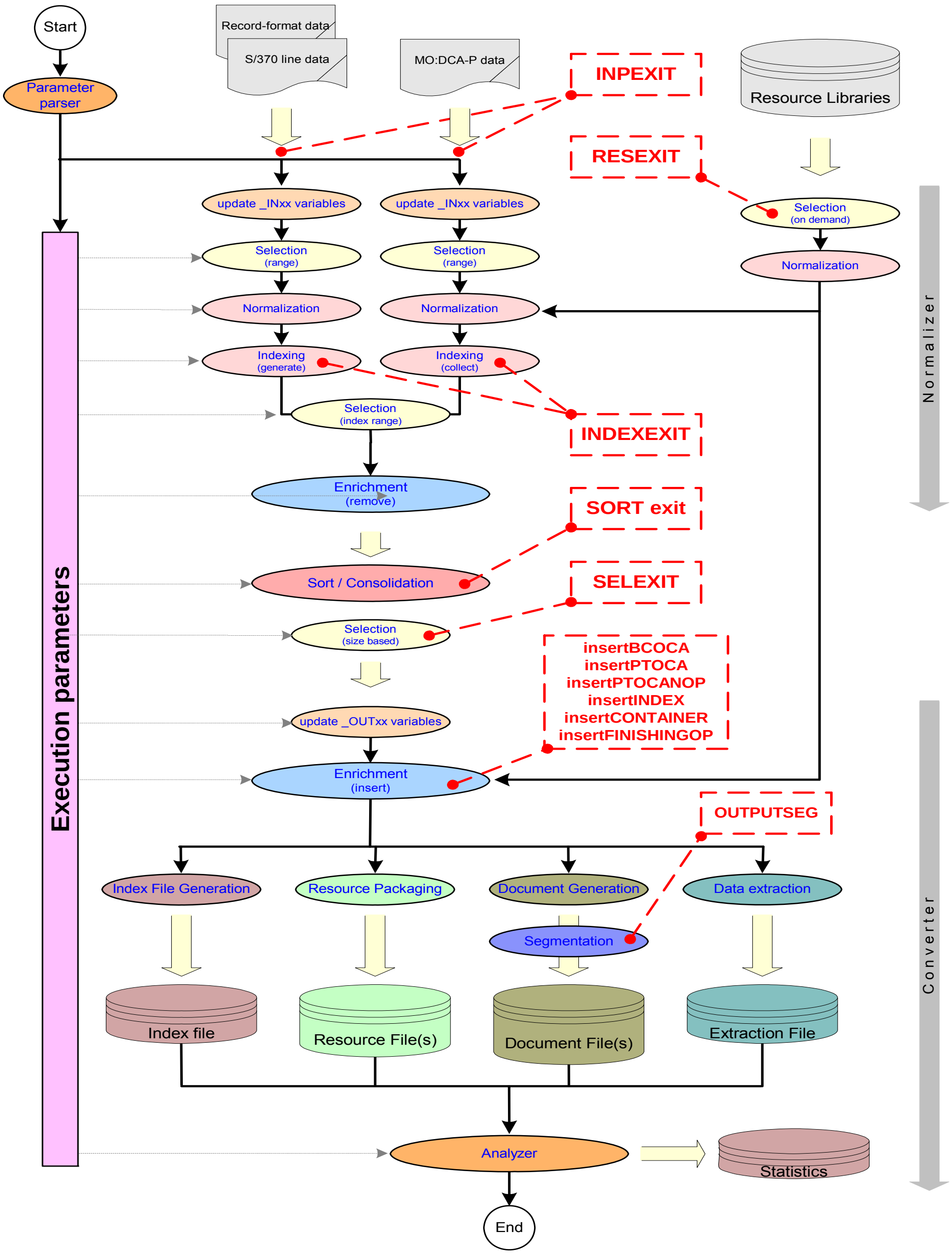


Fig. 88 CIS workflow and EXIT points

5.2 Index exit

The Index exit facility available in CIS allows installations to manipulate the index tag elements (TLE's) generated or collected during the normalization step using a user-written exit program. The name of the program invoked is specified in the **INDEXEXIT** keyword of the CIS control file (Refer to *4.4 CIS Control File* reference on page 160 for more information).

```
INDEXEXIT ( MYPROG )
```

Fig. 89 Example: INDEXEXIT keyword

The AFP architecture defines two different indexing levels: page group level and page level. In case of Page Group index tags, the exit program is invoked during the index generation or collection process, when all the index tags for a group were generated (using the TRIGGER keyword) and/or collected (already present in the input data stream), before the first page in the group or the first nested page group is processed. In case of Page index tags, the exit program is invoked during the index generation or collection step, at the end of the normalization process of a page. The Index exit program is **not** invoked for index tags inserted using the INSERTINDEX keyword as this occurs later within the CIS internal workflow (see Fig. 88 on page 287). The exit facility available in the INSERTINDEX keyword may be used instead.

The exit program receives a list with all the index tags elements CIS generated or collected for that page or page group during the normalization process. The following functionality is supported:

1. Insert additional index tag(s) in a page or page group
2. Discard index tag(s) from a page or page group
3. Modify existing index tag(s) in a page or page group

As Fig. 88 on page 287 shows, the Index generation and collection step (where the index exit program is invoked) is done before Index-Range based selection (see *Index-range based selection* on page 53) and before Sorting. Any modifications introduced by the Index exit program may be used during the index-based selection and/or sorting process. Some of the possible uses of the input exit facility include:

- Reformatting of index text including removal of leading and trailing blanks, case conversion, word separation, left/right alignment, standardized abbreviations (Mr., Mrs., Str., ...), insertion of leading zeroes, etc.

- Translation of index text to a different language or code page.
- Insertion of binary "work" index tags to be used for international alphanumeric sorting (e.g. German sorting sequence a, ä, o, ö, u, ü, s, ß; Spanish sorting sequence c, ch, l, ll, n, ñ; etc.)
- Insertion of binary "work" index tags to be used when sorting special fields such as dates (e.g. 01.Mar.2002), left aligned numeric fields, numeric fields with a variable number of decimal digits (e.g. 4.23, 2.1, 5.123), etc.
- Postal code clearance (in combination with appropriate data bases).

- **EXIT REQUIREMENTS**

    - See also *5.1 General technical guidelines* on page 281 for additional guidelines.

- **STORAGE OWNERSHIP**

- **LINKAGE CONVENTIONS**

- **INTERFACE DESCRIPTION**

    - Input:
        - Input file attributes
        - Entity attributes
        - List of Index tags generated/collected for current entity
        - CIS variables with current values
    - Output:
        - List of Index tags for current entity (replaces input list)

5.2.1 API provided by CIS

Please contact your Océ representative for more information concerning the CIS exit interface.

5.2.2 API expected by Index Plug-in

Please contact your Océ representative for more information concerning the CIS exit interface.

5.3 Input Record exit

CIS provides a User Exit facility that enables installations to modify the input record stream. The name of the program invoked is specified using the **INPEXIT** keyword in the CIS control file (Refer to *4.4 CIS Control File* reference on page 160 for more information).

```
INPEXIT ( MYPROG )
```

Fig. 90 Example: INPEXIT keyword

The exit is called after each record is read from the input file and before any further processing is performed with the data. The exit can request that the record is discarded, processed, or processed and control returned to the exit for the next exit-generated input record. The following functionality is supported:

1. Insert new records into input stream
2. Remove records from input stream
3. Modify records from input stream

The largest record that can be processed is 32756 bytes long.

- <u>EXIT REQUIREMENTS</u>

    - The input data processing done in CIS-Module is not necessarily sequential. While sequential processing is the normal case, the presence (and use) of inline resources within the input data stream will cause SPS/MVS-CIS to "jump" positions within the input file in order to read (or re-read) the data belonging to such a resource. An input record exit should check whether data is being read sequentially or not using the *sequence_hi* and *sequence_lo* (INPRECN) parameters and must be prepared to handle this situation.
    - The exit program has to ensure an absolute reproducibility of each input stream manipulation done. This means that if the exit adds a record at position *x* of the input data stream the exit must insert the same record at the same position a second time the input stream is read. Similarly, if the exit wants to remove a record it must remove that record every time it comes in. Same behavior is required in case of modified records.

- The exit is only allowed to insert one record at a time. In order to insert two or more records consecutively, the exit must use the *request* (INPREQ) parameter to indicate that control is to be returned immediately after processing the current record.
- See also *5.1 General technical guidelines* on page 281 for additional guidelines.

- **STORAGE OWNERSHIP**

The following rules define the memory allocation/release responsibilities of the user exit program:

- None of the control blocks allocated by CIS may be released by the exit program. This includes: the *work* memory, the *file* structure, the *charList* and the *record* buffer regardless on whether it is left unchanged, modified or deleted.
- None of the control blocks allocated by the exit program will be released by CIS.
- In order to avoid memory leak problems, the exit program must use the "terminate" call to release all the storage it may have allocated for a CIS run.

- **LINKAGE CONVENTIONS**

CIS uses the following linkage conventions when calling an exit program:

- General register 1 — Parameter list address. The first word contains the INPPARM pointer.
- General register 13 — 18-word save area
- General register 14 — Return address
- General register 15 — Exit's entry point address

The exit program is expected to restore the general registers 2 through 13 before returning to CIS. Restoring other general registers is optional. Floating-point registers do not need to be restored.

- **INTERFACE DESCRIPTION**

Fig. 91 shows a sample C header file that describes the structures passed to the exit program. Fig. 92 shows a sample DSECTs that describe the control blocks for exit programs written in assembler. The following lines describe these fields:

work
INPWORK

A pointer to a static, 1024-byte memory block. The exit program can use this memory block to save information across calls. The 1024-byte work area is aligned on a full word boundary and is initialized to binary zero before the first call is done. The exit program may not change this field.

mode
INPMODE

CIS initializes this field before every exit call. The exit program may not change it. It may be one of the following:

- X'01' Initialize processing. The exit is being invoked after the file was opened and before any data is read. This type of exit call is done once per input file. On entry the *record*, *length*, sequence number (*sequence_hi*, *sequence_lo*), *cc* and *trc* parameters are invalid. The *file* parameter is valid. On return, the *request* parameter is ignored.

- X'02' Record processing. The exit is being invoked after a record was read or inserted. This type of exit call is done many times per input file, depending on the number of records (and record inserts) found. On entry the *file*, *record*, *length*, sequence number (*sequence_hi*, *sequence_lo*), *cc* and *trc* parameters are valid. On return, the *request*, *record*, *length*, *cc* and *trc* parameters are used.

- X'03' End-of-file reached. The exit is being invoked after attempting to read past the EOF. This type of exit call is done once per input file. On entry the *record*, *length*, *cc* and *trc* parameters are not set. The *file* parameter is valid. On return, the *request, record, length, cc* and *trc* parameters may be used to insert additional records. Please note that the exit may still be invoked to process records even after an EOF call.

- X'04' Terminate processing. The exit is being invoked after the file has been closed. This type of exit call is done once per input file. The *record, length,* sequence number (*sequence_hi*, *sequence_lo*), *cc* and *trc* parameters are not used. The *file* parameter is valid. On return, the *request* parameter is ignored.

```
enum INPEXIT_MODE       // Exit calling mode
{
      INPEXIT_INIT      = 0x01;        // Initialization call
      INPEXIT_RECORD    = 0x02;        // Record processing call
      INPEXIT_EOF       = 0x03;        // End-of-file call
      INPEXIT_TERM      = 0x04;        // Termination call
};

enum INPEXIT_REQUEST       // Exit return request
{
      INPEXIT_REQ_PROCESS  = 0x00;     // Process this record
      INPEXIT_REQ_REMOVE   = 0x01;     // Do not process (remove) this record
      INPEXIT_REQ_PROCESS_AND_RETURN   = 0x02;    // Process this record
};
enum INPEXIT_CC         // Carriage control definition
{
      INPEXIT_CC_NONE   = 0x00;        // No CC present
      INPEXIT_CC_ASA    = 0x01;        // ASA control character (EBCDIC)
      INPEXIT_CC_IBM    = 0x02;        // Machine control character (IBM)
      INPEXIT_CC_ASAA   = 0x03;        // ASA control character (ASCII)
      INPEXIT_CC_OPS    = 0x04;        // OPS control character
      INPEXIT_CC_SIN    = 0x10;        // Force single spacing
      INPEXIT_CC_DOU    = 0x20;        // Force double spacing
      INPEXIT_CC_TRI    = 0x40;        // Force triple spacing
};

enum INPEXIT_TRC        // TRC definition
{
      INPEXIT_TRC_NONE  = 0x00;        // No TRC present
      INPEXIT_TRC_IBM   = 0x01;        // IBM TRC present
      INPEXIT_TRC_OPS   = 0x02;        // OPS TRC present
};

struct INPEXIT_CHARS    // Code font definition
{
      char              name[8];       // Coded font name
      short             mxm;           // MXM number
};

struct INPEXIT_FILE     // Input file attributes
{
      char              formDef[8];    // Form definition (see FORMDEF keyword)
      char              pageDef[8];    // Page definition (see PAGEDEF keyword)
      INPEXIT_CHARS*    charsList;     // Coded font list (see CHARS keyword)
      size_t            charslen;      // Number font list elements
      char              colorMap[8];   // Color mapping table (see COLORMAP)
      char              printMode[8];  // Print mode (see PRINTMODE keyword)
};

struct INPEXIT_PARMS    // Exit parameter list
{
```

```
       char*            work;          // Exit's work area
       INPEXIT_MODE     mode;          // Exit's calling mode
       INPEXIT_REQUEST  request;       // Exit's return request
       INPEXIT_FILE*    file;          // Points to file attributes
       char*            record;        // Points to record's data
       size_t           sequence_hi;   // Record's sequence number (high word)
       size_t           sequence_lo;   // Record's sequence number (low word)
       size_t           length;        // Record's length
       INPEXIT_CC       cc;            // Record's Carriage Control
       INPEXIT_TRC      trc;           // Record's Translate Character
};
```

Fig. 91 Input exit: C interface

request
INPREQ

Specifies, how the record is to be processed by CIS. On entry to the exit program, this parameter contains X'00'. When the exit program returns control, this parameter may be one of the following

- X'00' Specifies that the record is to be processed. The *request, record*, *length*, *cc* and *trc* parameters identify the record.
- X'01' Specifies that the record is not to be processed
- X'02' Specifies that the record is to be processed and control returned to the exit program to let it insert the next record. The exit program can set this value to save the current record, insert a record, and then supply the saved record at the next call. After the exit inserts the last record, the exit program must reset the *request*-byte to 0x00.

```
INPPARM    DSECT          * Exit parameter list
INPWORK    DS   AL4       * Pointer to exit's work area
INPMODE    DS   XL4       * Exit call mode
*                                F'01'    Initialization call
*                                F'02'    Record processing call
*                                F'03'    End-of-file call
*                                F'04'    Termination call
INPREQ     DS   XL4       * Exit return order
*                                F'00'    Process the record
*                                F'01'    Do not process (remove) the record
*                                F'02'    Process the record and return
```

```
INPFILE    DS   AL4       * Pointer to FILEPARM control block
INPREC     DS   AL4       * Pointer to record's data
INPRECN    DS   XL8       * Record's sequence number
INPRECL    DS   XL4       * Record's length
INPRECC    DS   XL4       * Record's Carriage Control (CC)
*                                F'00'    No CC present
*                                F'01'    ASA control character (EBCDIC)
*                                F'02'    Machine control character (IBM)
*                                F'03'    ASA control character (ASCII)
*                                F'04'    OPS control character
*                                F'16'    Force single spacing
*                                F'32'    Force double spacing
*                                F'64'    Force triple spacing
INPRECT    DS   XL4       * Record's Translate Character (TRC)
*                                F'00'    No TRC present
*                                F'01'    IBM TRC present
*                                F'02'    OPS TRC present


FILEPARM   DSECT          * Input file attributes
FILEFDEF   DS   CL8       * Form definition (see FORMDEF keyword)
FILEPDEF   DS   CL8       * Page definition (see PAGEDEF keyword)
FILECHAR   DS   AL4       * Coded font list (see CHARS keyword)
*                                CL8,XL2         Coded font name, MXM number
*                                ...
FILECHAN   DS   XL4       * Number of entries in Code font list
FILECOLM   DS   CL8       * Color mapping table (see COLORMAP keyword)
FILEPRM    DS   CL8       * Print mode (see PRINTMODE keyword)
```

Fig. 92 Input exit: assembler interface

file
INPFILE
Pointer to control block containing the file attributes. The exit program may not change this field.

record
INPREC
A pointer to the first byte of the input record including carriage control character, if any. The record resides in a buffer (32768 byte overall size) that resides in storage allocated by CIS. The exit program is allowed to modify the record's data but may not change the address stored in the *record* field.

sequence_hi
sequence_lo
INPRECN
Ordinal sequence number for incoming records (high and low 4-byte words in case of C). For unique identification each record sent to the exit gets a sequence number. The sequence number is incremented after each exit call done by CIS. The sequence is not incremented

when a record is not processed due to a 0x01 request. The first sequence number is 0. The exit program may not change this field.

length
INPRECL
Specifies the number of bytes of the input record. If the input record is modified, this parameter must also be updated to reflect the actual length of record.

cc
INPRECC
Specifies the Carriage Control attribute of this record. May be changed by the exit program. The allowed values are defined in INPEXIT_CC and INPRECC

trc
INPRECT
Specifies the Translate Reference Character attribute of this record. May be changed by the exit program. The allowed values are defined in INPEXIT_TRC and INPRECT.

5.4 Resource-selection exit

The Resource-selection exit available in CIS provides installations with a resource filter that can be used on a resource by resource basis. The name of the program invoked is specified in the **RESEXIT** keyword of the CIS control file (Refer to *4.4 CIS Control File* reference on page 160 for more information). The exit cannot be used to modify the resource data.

```
RESEXIT ( MYPROG )
```

Fig. 93 Example: RESEXIT keyword

Two levels of resource selection are available (refer to *3.7 How to Retrieve and Consolidate AFP Resource* on page 69 for more information): on a resource type basis using the RESTYPE keyword, or on a resource by resource basis using the Resource-selection exit. Both selection levels may be active at the same time. In this case the exit program is only invoked for the resources which fulfilled the RESTYPE criteria.

When resource consolidation is requested (see RESCONSOLIDATION keyword), the Resource-selection exit is invoked for all kind of resources used in the print file including external resources (from a resource library), external resource groups (inside the print file) or internal resource groups (inside a document). When resource consolidation is not active, the Resource-selection exit is invoked for all kind of resources used in the print file including external resources (from a resource library) and external resource groups (inside the print file) but **not** for internal resource groups (inside a document).

The Resource-selection exit is invoked during the resource normalization and resource packaging steps in the CIS workflow (see Fig. 88 on page 287), before the resource is read from a library or resource group. Any resources specified during a data enrichment operation will also be passed to the Resource-selection exit. The exit program decides whether a resource should be processed by CIS or not. Skipping a resource causes CIS to automatically skip any nested resources. The FormDef (for all print files) and PageDef (in case of S/370 Line format, XML or AFP Mixed format data) cannot be bypassed. The exit is only invoked for the resource files **used** in the print data. Resources which are not used in the print data are automatically always skipped by CIS.

Some of the possible uses of the Resource-selection exit facility include:

- To suppress resources which are part of a standard (never changed) resource package.

- To suppress resources due to security reasons.
- To bypass missing resources so that a print file can still be processed by CIS.
- To generate statistics about resource utilization.
- To integrate CIS resource packaging functionality in a more comprehensive resource management system.

- <u>EXIT REQUIREMENTS</u>
    - See also *5.1 General technical guidelines* on page 281 for additional guidelines.
- <u>STORAGE OWNERSHIP</u>
- <u>LINKAGE CONVENTIONS</u>
- <u>INTERFACE DESCRIPTION</u>
    - Input:
        - Resource name and type
        - File attributes including library name, path, etc.
    - Output:
        - YES or NO

5.4.1 API provided by CIS

Please contact your Océ representative for more information concerning the CIS exit interface.

5.4.2 API expected by Resource Plug-in

Please contact your Océ representative for more information concerning the CIS exit interface.

5.7 BCOCA-insertion exit

The BCOCA-insertion exit can be used by an installation to generate the information string to be inserted during a data enrichment operation. The name of the program(s) invoked is specified in one or more INSERTBCOCA keywords (refer to *4.4 CIS Control File* reference on page 160 for more information). For more information concerning data enrichment please refer to *3.8 How to Enrich Print Data streams* on page 77.

```
INSERTBCOCA  ( 'name', myLevel, 'USEREXIT', myProg1 )
```

Fig. 96 Example: INSERTBCOCA, USEREXIT parameter

The BCOCA-insertion exits are invoked during the data enrichment step in the conversion phase of the CIS workflow (see Fig. 88 on page 287). The exit program(s) are invoked every time an INSERTBCOCA keyword is to be processed. The exit program decides whether the data enrichment operation should be done or not, and returns the information string to be used. The exit program may modify one or more of the parameters to be used for the insertion which are based on the corresponding DEFINEBCOCA keyword in the CIS control file.

Some of the possible uses of the BCOCA-insertion exit facility include:

- To generate Bar Codes types requiring check-sum digits.
- To generate Bar Code symbols containing post-processing device controls.
- To generate a BCOCA string based on an encoding scheme (code page) not supported by CIS.
- See also *5.1 General technical guidelines* on page 281 for additional guidelines.
- Input:
  - Input file attributes
  - Entity and Level attributes including Index tags generated/collected for current entity and list of PTOCA strings for current entity
  - CIS variables with current values
  - DEFINExxx parameters
- Output:

- YES or NO
- If YES: string to be inserted and DEFINExxx parameters

The Insert Bcoca plug-in allows an user to generate the information string and to redefine the bar code symbol defines by an exit program. The name of the program invoked is specified in the *insertBcoca* keyword.

```
DEFINELEVEL( myLevel, GROUPLEVEL1, FRONT, EVEN )
DEFINEBCOCA ( 'myBcoca', CM,10 ,10 ,90 ,RED ,CODE39,
        1, OFF, OFF, 21, 15, 100, IBM_0500 )

FIELD( myBcocaPlugIn, 'BCOCEXIT' )

INSERTBCOCA( 'myBcoca', myLevel, 'USEREXIT', myBcocaPlugIn )
```

The exit program is invoked every time an *insertBcoca* keyword is to be processed. The exit program decides whether the bar code symbol enrichment should be done or not. This technique offer the user much more flexibility to generate bar code symbols.

5.7.1 API provided by CIS

Please contact your Océ representative for more information concerning the CIS exit interface.

5.7.2 API provided by the Plug-in

Please contact your Océ representative for more information concerning the CIS exit interface.

5.8 PTOCA-insertion exit

The PTOCA-insertion exit can be used by an installation to generate the information string to be inserted during a data enrichment operation. The name of the program(s) invoked is specified in one or more INSERTPTOCA keywords (refer to *4.4 CIS Control File* reference on page 160 for more information). For more information concerning data enrichment please refer to *3.8 How to Enrich Print Data streams* on page 77.

```
INSERTPTOCA  ( 'name', myLevel, 'USEREXIT', myProg1 )
```

Fig. 97 Example: INSERTPTOCA, USEREXIT parameter

The PTOCA-insertion exits are invoked during the data enrichment step in the conversion phase of the CIS workflow (see Fig. 88 on page 287). The exit program(s) are invoked every time an INSERTPTOCA keyword is to be processed. The exit program decides whether the data enrichment operation should be done or not, and returns the information string to be used. The exit program may modify one or more of the parameters to be used for the insertion which are based on the corresponding DEFINEPTOCA keyword in the CIS control file.

Some of the possible uses of the PTOCA-insertion exit facility include:

Some of the possible uses of the OCA-insertion exit facility include:

- To generate Optical marks for controlling post-processing devices.
- To generate font-based bar code symbols including 2D bar codes.
- To generate a PTOCA string based on an encoding scheme (code page) not supported by CIS.
- See also *5.1 General technical guidelines* on page 281 for additional guidelines.
- Input:
    - Input file attributes
    - Entity and Level attributes including Index tags generated/collected for current entity and list of PTOCA strings for current entity
    - CIS variables with current values
    - DEFINExxx parameters

- Output:
  - YES or NO
  - If YES: string to be inserted and DEFINExxx parameters

The Insert Ptoca plug-in allows an user to generate the information string and to redefine the presentation text defines by an exit program. The name of the program invoked is specified in the *insertPtoca* keyword.

```
DEFINELEVEL( myLevel, GROUPLEVEL1, FRONT, EVEN )
DEFINEPTOCA( 'myPtoca', MM, 100, 220, , , X0GT12, IBM_0500 )

FIELD( myPtocaPlugIn, 'PTOCEXIT' )

INSERTPTOCA( 'myPtoca', myLevel, 'USEREXIT', myPtocaPlugIn )
```

The exit program is invoked every time an *insertPtoca* keyword is to be processed. The exit program decides whether the presentation text enrichment should be done or not. This technique offer the user much more flexibility to generate presentation texts.

5.8.1 API provided by CIS

Please contact your Océ representative for more information concerning the CIS exit interface.

5.8.2 API provided by the Plug-in

Please contact your Océ representative for more information concerning the CIS exit interface.

5.9 PTOCA-NOP-insertion exit

The PTOCA-NOP-insertion exit can be used by an installation to generate the information string to be inserted during a data enrichment operation. The name of the program(s) invoked is specified in one or more INSERTPTOCANOP keywords (refer to *4.4 CIS Control File reference* on page 160 for more information). For more information concerning data enrichment please refer to *3.8 How to Enrich Print Data streams* on page 77.

```
INSERTPTOCANOP ( 'name', myLevel, 'USEREXIT', myProg1 )
```

Fig. 98 Example: INSERTPTOCA, USEREXIT parameter

The PTOCA-NOP-insertion exits are invoked during the data enrichment step in the conversion phase of the CIS workflow (see Fig. 88 on page 287). The exit program(s) are invoked every time an INSERTPTOCANOP keyword is to be processed. The exit program decides whether the data enrichment operation should be done or not, and returns the information string to be used. The exit program may modify one or more of the parameters to be used for the insertion which are based on the corresponding DEFINEPTOCA keyword in the CIS control file.

Some of the possible uses of the PTOCA-NOP-insertion exit facility include:

Some of the possible uses of the OCA-insertion exit facility include:

- To generate PTOCA NOP control sequences to be used in connection with the Océ ImageStream archiving solution.
- To generate a bar code or PTOCA string based on an encoding scheme not supported by CIS.
- See also *5.1 General technical guidelines* on page 281 for additional guidelines.
- Input:
  - Input file attributes
  - Entity and Level attributes including Index tags generated/collected for current entity and list of PTOCA strings for current entity
  - CIS variables with current values
  - DEFINExxx parameters
- Output:

- YES or NO
- If YES: string to be inserted and DEFINExxx parameters

The Insert PtocaNop plug-in allows an user to generate the information string and by an exit program. The name of the program invoked is specified in the *insertPtocaNop* keyword.

```
DEFINELEVEL( myLevel, GROUPLEVEL1, FRONT, EVEN )
DEFINEPTOCA( 'myPtoca', MM, 100, 220, , , X0GT12, IBM_0500 )

FIELD( myPtocaNopPlugIn, 'PTOCEXIT' )

INSERTPTOCANOP( 'myPtoca', myLevel, 'USEREXIT', myPtocaNopPlugIn )
```

The exit program is invoked every time an *insertPtocaNop* keyword is to be processed. The exit program decides whether the none operated presentation text enrichment should be done or not. This technique offer the user much more flexibility to generate none operated presentation texts.

5.9.1 API provided by CIS

Please contact your Océ representative for more information concerning the CIS exit interface.

5.9.2 API provided by the Plug-in

Please contact your Océ representative for more information concerning the CIS exit interface.

5.10 INDEX-insertion exit

The INDEX-insertion exit can be used by an installation to generate the information string to be inserted during a data enrichment operation. The name of the program(s) invoked is specified in one or more INSERTINDEX keywords (refer to *4.4 CIS Control File* reference on page 160 for more information). For more information concerning data enrichment please refer to *3.8 How to Enrich Print Data streams* on page 77.

```
INSERTINDEX ( 'name', myLevel, 'USEREXIT', myProg1 )
```

Fig. 99 Example: INSERTINDEX, USEREXIT parameter

The INDEX-insertion exits are invoked during the data enrichment step in the conversion phase of the CIS workflow (see Fig. 88 on page 287). The exit program(s) are invoked every time an INSERTINDEX keyword is to be processed. The exit program decides whether the data enrichment operation should be done or not, and returns the information string to be used. The exit program may modify one or more of the parameters to be used for the insertion which are based on the corresponding DEFINEINDEX keyword in the CIS control file.

Some of the possible uses of the INDEX-insertion exit facility include:

- To generate an Index string based on an encoding scheme not supported by CIS.
- To generate Index entries using strategies not possible with the TRIGGER and FIELD keywords.
- See also *5.1 General technical guidelines* on page 281 for additional guidelines.
- Input:
  - Input file attributes
  - Entity and Level attributes including Index tags generated/collected for current entity and list of PTOCA strings for current entity
  - CIS variables with current values
  - DEFINExxx parameters
- Output:
  - YES or NO

- If YES: string to be inserted and DEFINExxx parameters

The Insert Index plug-in allows an user to generate an whole index entry by an exit program. The name of the program invoked is specified in the *insertIndex* keyword.

```
DEFINELEVEL( myLevel, GROUPLEVEL1, FRONT, EVEN )

FIELD( myIndexPlugIn, 'INDXEXIT' )

INSERTINDEX( 'myIndex', myLevel, 'USEREXIT', myIndexPlugIn )
```

The exit program is invoked every time an *insertIndex* keyword is to be processed. The exit program decides whether the index entry enrichment should be done or not. This technique offer the user much more flexibility to generate index entries.

5.10.1 API provided by CIS

Please contact your Océ representative for more information concerning the CIS exit interface.

5.10.2 API provided by the Plug-in

Please contact your Océ representative for more information concerning the CIS exit interface.

5.11 ObjectContainer-insertion exit

The INSERTCONTAINER exit can be used by an installation to generate an AFP Object Container which is inserted by CIS in the print data stream. The name of the program invoked is specified as parameter in the INSERTCONTAINER keyword (refer to *4.4 CIS Control File* reference on page 160 for more information). For more information concerning data enrichment please refer to *3.8 How to Enrich Print Data streams* on page 77.

```
INSERTCONTAINER ( 'name', myLevel, 'USEREXIT', myProg )
```

Fig. 100 Example: ObjectContainer-insertion exit

The ObjectContainer-insertion exits are invoked during the data enrichment step in the conversion phase of the CIS workflow (see Fig. 88 on page 287). The exit program is invoked every time the INSERTCONTAINER keyword where the exit name was specified is to be processed.

The exit program:

- Decides whether the data enrichment operation should be done or not.
- Is allowed to modify the parameters of the corresponding DEFINECONTAINER keyword
- May return an optional buffer containing non-OCA data. CIS uses this data, together with the parameters from DEFINECONTAINER, to build an Object Container which is directly inserted in the target page. An IOB structured field and an OC resource are **not** generated when a data buffer is returned by the ObjectContainer-insertion exit. The format of the data included must correspond to the object class and type specified in the DEFINECONTAINER keyword. The mode parameter is ignored.

Some of the possible uses of the ObjectContainer-insertion exit facility include:

- To decide whether the Container is to be inserted or not.
- To modify the parameters from the DEFINECONTAINER keyword. fileName cannot be modified. Mode can be modified only if the exit returns a buffer containing non-Oca data.
- To dynamically generated / retrieve non-OCA objects which are then inserted to the print file.
- See also *5.1 General technical guidelines* on page 281 for additional guidelines.

- Input:
  - Input file attributes
  - Entity and Level attributes including Index tags generated/collected for current entity and list of PTOCA strings for current entity
  - CIS variables with current values
  - DEFINECONTAINER parameters
- Output:
  - YES or NO
  - If YES: object data to be inserted (optional) and DEFINECONTAINER parameters

The Object Container plug-in allows an user to generate object container data and to redefine the object container defines by an exit program. The name of the program invoked is specified in the *insertContainer* keyword.

```
DEFINELEVEL( myLevel, GROUPLEVEL1, FRONT, EVEN )
DEFINECONTAINER( 'myContainer', pels, 200, 200, 800, 800,
         x'00', 1, 1, 0 ,GREEN, EPS, SOFT,
         HLQ.DIAGRAMS(CONT01) )

INSERTCONTAINER( 'myContainer', 'myLevel',
         USEREXIT, 'CONTEXIT' )
```

The exit program is invoked every time a new entity is to be processed. The exit program decides whether the object container enrichment should be done or not. This technique offer the user much more flexibility to generate object containers.

5.11.1 API provided by CIS

Please contact your Océ representative for more information concerning the CIS exit interface.

5.11.2 API provided by the Plug-in

Please contact your Océ representative for more information concerning the CIS exit interface.

5.12 UP³I Finishing Operation-insertion exit

The UP³I Finishing Operation-insertion exit can be used by an installation to modify or suppress the UP³I Finishing operation which is to be inserted by CIS in the print data stream. The name of the program invoked is specified as parameter in the INSERTFINISHINGOP keyword (refer to *4.4 CIS Control File* reference on page 160 for more information). For more information concerning data enrichment please refer to *3.8 How to Enrich Print Data streams* on page 77.

```
INSERTFINISHINGOP ( 'name', myLevel, 'USEREXIT', myProg )
```

Fig. 101 Example: INSERTFINISHINGOP keyword, USEREXIT parameter

The INSERTFINISHINGOP exit is invoked during the data enrichment step in the conversion phase of the CIS workflow (see Fig. 88 on page 287). The exit program is invoked every time the INSERTFINISHINGOP keyword where the exit name was specified is to be processed. The exit program decides whether the data enrichment operation should be done or not, and is allowed to modify the parameters of the corresponding DEFINEFINISHINGOP keyword.

Some of the possible uses of the UP³I Finishing Operation-insertion exit facility include:

- To define variable parameters to be used for a UP³I Finishing Operation based on the page contents or a similar criteria.
- To suppress the insertion of a UP³I Finishing Operation based on the page contents or a similar criteria.
- See also *5.1 General technical guidelines* on page 281 for additional guidelines.
- Input:
    - Input file attributes
    - Entity and Level attributes including Index tags generated/collected for current entity and list of PTOCA strings for current entity
    - CIS variables with current values
    - DEFINEFINISHINGOP parameters
- Output:
    - YES or NO

- If YES: parameters to be used for INSERTFINISHINGOP parameters

The Insert FinishingOp plug-in allows an user to modify the finishing operation parameters by an exit program. The name of the program invoked is specified in the *insertFinishingOp* keyword.

```
DEFINELEVEL( myLevel, GROUPLEVEL1, ALL, ALL )

DEFINEFINISHINGOP( 'myFinishingOp', PELS, OFFSET, LEFT, TOP,1 ,0, )

INSERTFINISHINGOP( 'myFinishingOp', myLevel, USEREXIT,
  'FINIEXIT')
```

The exit program is invoked every time an *insertFinishingOp* keyword is to be processed. The exit program decides whether the finishing operation should be done or not. This technique offer the user much more flexibility to generate finishing operations.

5.12.1 API provided by CIS

Please contact your Océ representative for more information concerning the CIS exit interface.

5.12.2 API provided by the Plug-in

Please contact your Océ representative for more information concerning the CIS exit interface.

Sort exit

The Sort exit can be used by an installation to write their own sorting algorithm. The name of the program invoked is specified in the SORT keyword (refer to *4.4 CIS Control File* reference on page 160 for more information). Please refer to *3.10 How to Sort and Reorder output pages* on page 114 for more information concerning the sorting algorithms available in CIS.

```
SORT ( USEREXIT, 'myProg' )
```

Fig. 102 Example: SORT keyword, USEREXIT parameter

The Sort exit is invoked during the sort step, at the end of the normalization phase of the CIS workflow (see Fig. 4 on page 32). The exit program is invoked once.

- The Sort Plug-in has to take care that every handle in the handle tree is linked at only one position and in semantic correct order, that is page handles are only allowed to be linked into page group handles, which themselves must be linked into document handles which must be linked into the PDS handle.
- Please refer to *5.1 General technical guidelines* on page 281 for additional information.

a) The handle tree

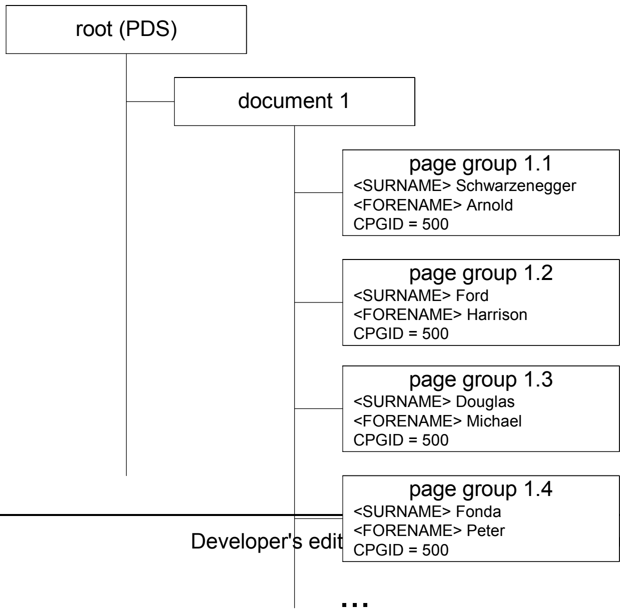
There is one root entity (we call it the PDS handle or root handle) which contains one or more **document handles**, which contain themselves one or more **page group handles**. Page group handles can contain **nested page group handles** or **page handles**.

Fig. 103 on page 317 depicts such an hierarchical internal representation. This abstract representation can be browsed and modified by the Sort Plug-in.

b) Considerations about sorting handles by their index

Page group and/or page handles can have one or more indices associated to them. CIS provides access to these indices. An index consists of an attribute name (e.g. "SURNAME"), an value (e.g. "Schwarzenegger") and the IBM codepage global ID it is encoded with (a number between 0x0001 and 0xFFFF).

Following graphic depicts an handle tree with indexed page groups. For reasons of clarity



pages or nested page groups linked to the first level page groups (also called mailpiece's) are left out. The attributes are written in squared brackets, the associated values right after them.

Fig. 103 Sample handle tree

An hypothetical Sort Plug-in might sort these page group handles, with all their child handles linked to them, in a total arbitrary order.

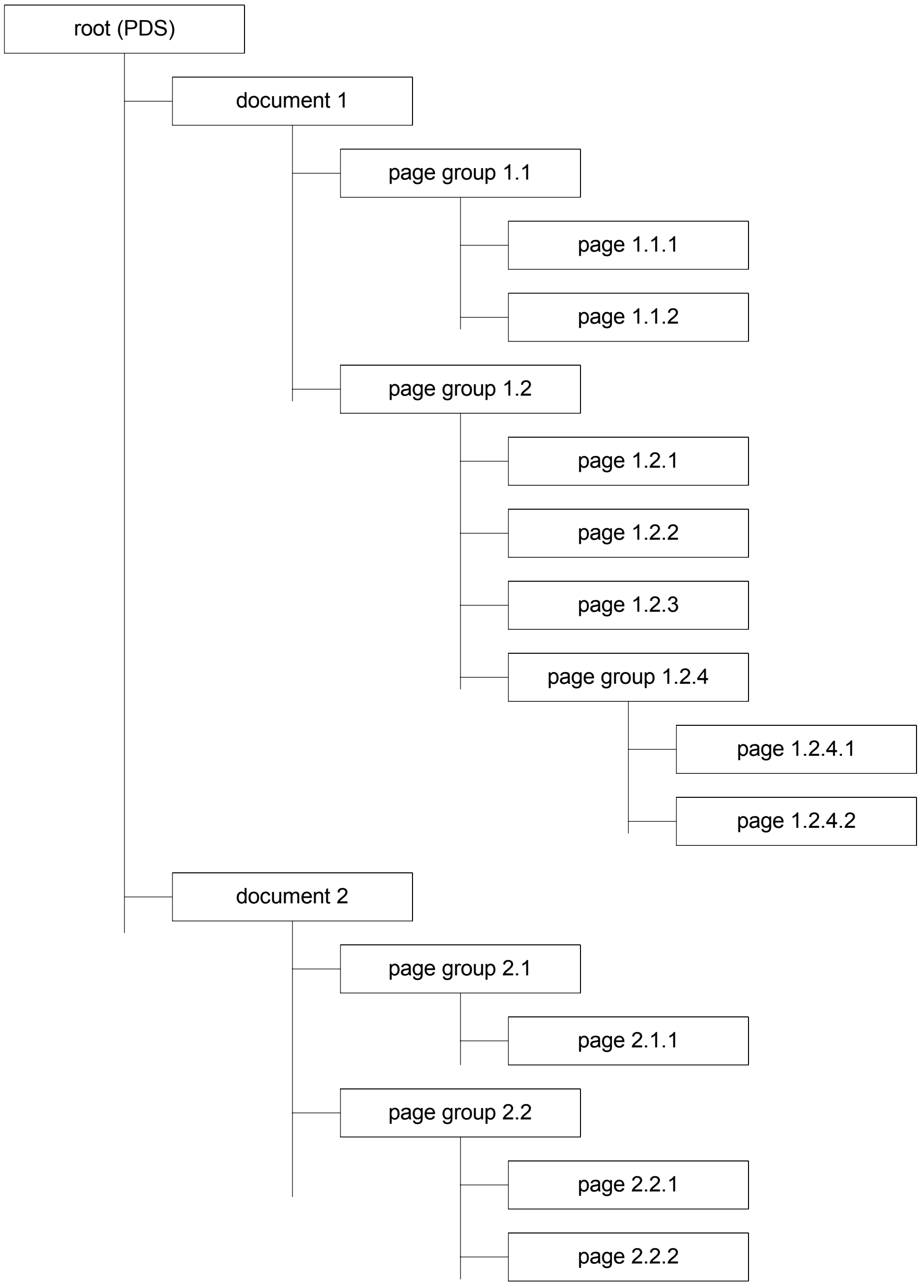


Fig. 104 The handle tree

c) Sort Plug-in Interface version 1

This section describes the Sort Plug-in Interface version 1. It contains the specifications of functions CIS provides to the loaded Plug-in and the functions the Plug-in has to provide to CIS.

5.13.1 API provided by CIS

Please contact your Océ representative for more information concerning the CIS exit interface.

5.13.2 API provided by the Plug-in

Please contact your Océ representative for more information concerning the CIS exit interface.

6 Appendix

6.1 Appendix A: Licensing Requirements

CIS is a licensed product. Proper licensing is required in order to use the product. Product features can be licensed separately as dictated by the installation needs. The license key will contain all of the elements necessary to validate the authorization level of an installation.

The license key processing in CIS is different depending on the platform where the product is running. Please refer to *4.2 Executing CIS under Linux* or *4.3 Executing CIS under MVS* for more information.

6.1.1 Licensing under OS/390, z/OS

- **<u>LICENSE TYPES</u>**

Type	Product Feature	Related Keywords
AFP Resource Packaging		[Input] INPUTDD INPUTCP INPUTCP INPUTFORMAT LINEMERGE [Formatting] CC FORMDEF PAGEDEF TRCTYPE [Output] OUTPUTDD OUTPUTFORMAT RESOBJDD RESTYPE [Resources] CHARS FDEFLIB FONTLIB OBJCONLIB OVLYLIB PDEFLIB PSEGLIB USERLIB CODEDFONTEXT CODEPAGEEXT FONTCHARSETEXT FORMDEFEXT OBJCONEXT OUTLINEFONTEXT OVERLAYEXT PAGEDEFEXT PAGESEGEXT

Type	Product Feature	Related Keywords
		[Miscellaneous] COLORMAP LSKEYDD MESSAGECP TRACEDD TRACELEVEL TRACESIZE TRACETYPE
Basic	Conversion and Normalization	CC CHARS COMPRESS GROUPNAME IMAGEOUT INPEXIT INPUTCP INPUTDD INPUTFORMAT LINEMERGE LSKEYDD LUPUB MESSAGECP PRESENTATION PRINTMODE OUTPUTFORMAT OUTPUTDD TRACEDD TRACELEVEL TRACESIZE TRACETYPE TRCTYPE UNIQUEBNGS USEPAGENAMES WORKDD
	Indexing	FIELD INDEX INDEXCP INDEXDD INDEXOBJ INDEXSTARTBY INSERTINDEX REMOVEINDEX TRIGGER
	Sort and Consolidation	MPCONSOLIDATION SORT

	Resource Packaging	CODEDFONTEXT CODEPAGEEXT COLORMAP FDEFLIB FONTCHARSETEXT FONTLIB FONTMAP FORMDEF FORMDEFEXT OBJCONEXT OBJCONLIB OVERLAYEXT OVLYLIB PAGEDEF PAGEDEFEXT PAGESEGEXT PDEFLIB PSEGLIB RESCONSOLIDATION RESEXIT RESFIDELITY RESFILE RESOBJDD RESTYPE USERLIB
	Data Stream Analyst	ANALYST ANALYSTCP ANALYSTDD
	Basic data enrichment (including UP³I)	DEFINEFINISHINGOP COLORFIDELITY DEFINEMEDIUMMAP DEFINELEVEL DUPLEX FINISHINGFIDELITY FONTFIDELITY INSERTFINISHINGOP INBIN INSERTSHEET MEDIAFIDELITY OFFSETXB OFFSETXF OFFSETYB OFFSETYF OUTBIN OVERLAYB OVERLAYF REMOVEFINISHIGOP REMOVEMEDIUMMAP REMOVESHEET SETUPVID TONERFIDELITY X2UP
Data splitting	Data selection	SELINDEXRANGE SELRANGE SELRANGERANDOM SELSIZE
	Data segmentation	OUTPUTSEG OUTPUTSEGPSDEF
Data enrichment		DEFINECOLOR DEFINELEVEL INITVARIABLES
	Insert and Remove BCOCA objects	DEFINEBCOCA INSERTBCOCA REMOVEBCOCA
	Insert and Remove GOCA objects	DEFINEBOX DEFINECIRCLE DEFINECURVE DEFINEELLIPSE DEFINELINE INSERTBOX INSERTCIRCLE INSERTCURVE INSERTELLIPSE INSERTLINE REMOVEBOX REMOVECIRCLE REMOVECURVE REMOVEELLIPSE REMOVELINE

	Insert and Remove Object Containers	DEFINECONTAINER INSERTCONTAINER REMOVECONTAINER
	Insert and Remove PTOCA objects	DEFINEPTOCA INSERTPTOCA INSERTPTOCANOP REMOVEPTOCA
Professional Services	Data extraction	EXTRACTCP EXTRACTINDEX
	User Exits	INDEXEXIT USEREXIT parameter in: OUTPUTSEG INSERTBCOCA INSERTINDEX INSERTPTOCA INSERTPTOCANOP SORT INSERTCONTAINER

- **TIER LEVELS**

The CIS tier levels and their correspondent license type is shown in the following table:

License type	Tier level
Basic	0
Data splitting	1
Data enrichment	2
Professional Services	3

Tier levels are accumulative so that Data Splitting includes the Basic functions, Data enrichment includes the Basic and the Data Splitting functions, etc.

6.1.2 Licensing under LINUX

6.1.3 Licensing under LINUX: Overview

From CIS Version 5.00 the following license levels are defined:

				Technique										
				former CIS Service										
				1				2	3	4	5	6	7	8
				PRISMAproduction basic										
				Resource Packaging	LineData Normalization	PageCount	Preview prep./selection	DataManipulation CIS	Indexing LineData	AFP Preflight&MR	DataManipulation Dpconnect	DataManipulatio n LCDS	Navigation ManualReprint	Indexing "OCR"
Sale	PRISMAproduction	Server4.02	BaseTier **mandatory**	X	X	X	X							
			Data Manipulation **optional**					X	X					
			AFP Preflight&MR **optional**							X				
			DPconnect **optional**								X			
			LCDS **optional**									X		
			Navigation ManualReprint **optional**										X	
			Indexing"OCR" **optional**											X
		Host	CIS	X	X	X		X	X					

Product Feature	**Related Keywords**
CIS "Basic Features"	Basic CIS functions, require PRISMAproduction basic

Product Feature	Related Keywords
	license
	1. ANALYST 2. ANALYSTCP 3. ANALYSTDD 4. CASECTRL 5. CC 6. CHARS 7. CODEDFONTEXT 8. CODEPAGEEXT 9. COLORFIDELITY 10. COLORMAP 11. FDEFLIB 12. FONTCHARSETEXT 13. FONTLIB 14. FONTMAP 15. FORMDEF 16. FORMDEFEXT 17. IMAGEOUT 18. INLINERESOURCECACHING 19. INPEXIT 20. INPUTCP 21. INPUTDD 22. INPUTFORMAT 23. LINEMERGE 24. LSKEYDD 25. MESSAGECP 26. MPCONCEPT 27. OBJCONEXT 28. OBJCONLIB 29. OUTLINEFONTEXT 30. OUTPUTDD 31. OUTPUTFORMAT 32. OUTPUTSEG 33. OUTPUTSEGEXIT 34. OUTPUTSEGPSDEF 35. OVERLAYB 36. OVERLAYEXT 37. OVERLAYF 38. OVLYLIB 39. PAGEDEF 40. PAGEDEFEXT 41. PAGESEGEXT 42. PDEFLIB 43. PRESENTATION 44. PRINTMODE 45. PSEGLIB 46. RESCONSOLIDATION 47. RESEXIT 48. RESFIDELITY 49. RESFILE 50. RESOBJDD 51. RESPRUNING 52. RESTYPE 53. SELEXIT 54. SELINDEXRANGE 55. SELRANGE 56. SELRANGERANDOM 57. SELSIZE 58. SELTEXT 59. SETUPCHARS 60. SETUPFORMDEF 61. SETUPPAGEDEF 62. TMPDIR 63. TRACEDD 64. TRACELEVEL 65. TRACESIZE 66. TRACETYPE 67. TRCTYPE 68. USERLIB 69. WORKDD 70. JOBID ~~71. TRACEOPTIMIZE~~
CIS "Data manipulation"	1. COMPRESS 2. DEFINE2DDATAMATRIX 3. DEFINE2DMAXICODE 4. DEFINE2DPDF417 5. DEFINE2DQRCODE 6. DEFINEBCOCA 7. DEFINEBOX 8. DEFINECIRCLE 9. DEFINECOLOR 10. DEFINECONTAINER 11. DEFINECURVE 12. DEFINEELLIPSE 13. DEFINEFINISHINGOP 14. DEFINELEVEL 15. DEFINELINE 16. DEFINEMEDIUMMAP 17. DEFINEPTOCA 18. DUPLEX 19. EXTRACTCP 20. EXTRACTINDEX 21. FIELD 22. FINISHINGFIDELITY 23. FONTFIDELITY 24. INBIN 36. INSERTINDEX 37. INSERTINDEXEXIT 38. INSERTLINE 39. INSERTPTOCA 40. INSERTPTOCAEXIT 41. INSERTPTOCANOP 42. INSERTPTOCANOPEXIT 43. INSERTSHEET 44. LUPUB 45. MEDIAFIDELITY 46. MPCONSOLIDATION 47. OFFSETXB 48. OFFSETXF 49. OFFSETYB 50. OFFSETYF 51. OUTBIN 52. REMOVEBCOCA 53. REMOVEBOX 54. REMOVECIRCLE 55. REMOVECMR 56. REMOVECONTAINER 57. REMOVECURVE 58. REMOVEELLIPSE 59. REMOVEFINISHINGOP

Product Feature	Related Keywords
	25. INITVARIABLES 26. INSERTBCOCA 27. INSERTBCOCAEXIT 28. INSERTBOX 29. INSERTCIRCLE 30. INSERTCONTAINER 31. INSERTCONTAINEREXIT 32. INSERTCURVE 33. INSERTELLIPSE 34. INSERTFINISHINGOP 35. INSERTIMM 60. REMOVEINDEX 61. REMOVELINE 62. REMOVEMEDIUMMAP 63. REMOVEPTOCA 64. REMOVESHEET 65. SETUPVID 66. SORT 67. SORTEXIT 68. TONERFIDELITY 69. USEPAGENAMES 70. X2UP
CIS "Indexing LineData"	In this case CIS will check for a separate license key and activate the relevant functions. 1. GROUPNAME 2. INDEX 3. INDEXCP 4. INDEXDD 5. INDEXEXIT 6. INDEXOBJ 7. INDEXSTARTBY 8. TRIGGER 9. UNIQUEBNGS
"AFP Preflight and MakeReady"	In this case CIS will check for a separate license key and activate the relevant functions. 1. ANALYST 2. ANALYSTDD 3. BLACKWHITE 4. CASECTRL 5. CODEDFONTEXT 6. CODEPAGEEXT 7. COLORFILEDD 8. COLORINFO 9. COLORINFOCP 10. COLORMAP 11. COLORREPLACE 12. COLORREPLACE_PTOCAGRAY2BLACK 13. COLORREPLACE_RICHBLACK 14. COLORRESDD 15. DUPLEX 16. FDEFLIB 17. FONTMAP 18. FORMDEF 19. FORMDEFEXT 20. IMAGEOUT 21. INBIN 22. INPEXIT 23. INPUTDD 24. INPUTFORMAT 25. INSERTCMR 26. INSERTPPO 27. JOBID 28. MESSAGECP 29. OBJECTCONEXT 30. OBJECTCONLIB 31. OBJEXTRACT 32. OBJEXTRACT_DIRECTORY 33. OUTBIN 34. OUTPUTDD 35. OUTPUTSEG 36. OVERLAYB 72. OVERLAYB 73. OVERLAYEXT 74. OVERLAYF 75. OVLYLIB 76. PAGEDEF 77. PAGEDEFEXT 78. PAGESEGEXT 79. PDEFLIB 80. PRESENTATION 37. REPLACECMR 81. RESFIDELITY 82. RESFILE 83. RESOBJDD 84. RESPRUNING 85. RESTYPE 86. SETUPCHARS 87. TMPDIR 88. TRCTYPE 89. USERLIB
DPconnect "Data manipulation"	1. COLORMAP 2. DEFINECOLOR 3. DEFINECONTAINER 4. DEFINEFINISHINGOP

Product Feature	Related Keywords
	5. DEFINEMEDIUMMAP 6. DUPLEX 7. FINISHINGFIDELITY 8. IMAGEOUT 9. INPUTDD 10. INSERTCONTAINER 11. INSERTSHEET 12. JOBID 13. MEDIAFIDELITY 14. OUTPUTDD 15. OVERLAYB 16. OVERLAYF 17. REMOVESHEET 18. RESTYPE 19. SORT 20. SORTEXIT
LCDS "Data manipulation"	1. ANALYST 2. ANALYSTDD 3. FONTLIB 4. IMAGEOUT 5. INPUTDD 6. JOBID 7. OVLYLIB 8. PSEGLIB 9. RESCONSOLIDATION 10. RESFILE 11. RESOBJDD 12. RESTYPE 13. SORT 14. SORTEXIT 15. TMPDIR 16. TRACEDD 17. TRACELEVEL
"Navigation and Manual Reprint"	In this case CIS will check for a separate license key and activate the relevant functions. No additional param keywords defined so far
"Indexing OCR"	In this case CIS will check for a separate license key and activate the relevant functions. No additional param keywords defined so far

6.2 Appendix B: CIS processing variables

The following table describes the processing variable names available in CIS. These variables can be used in the FIELD and INITVARIABLES keywords of the control file, and are available to some User Exits as additional information. Their initial value and change conditions are described too.

- **INPUT FILE RELATED VARIABLES**

  - **_inDocumentNumber**

Description:	Input Document number. Contains the relative number of an AFP document within the input data stream. CIS uses the Begin Document (BDT) and End Document (EDT) structured fields in order to determine which pages and page groups belong to a document. S/370 Line format data files are considered to contain a single AFP document. The complete input file is used to determine the relative number, including items which are excluded in the selection steps.
Reset condition(s):	At the beginning of the print file The initial value set every time a reset condition occurs is 0. The initVARIABLES keyword may be used to specify a different initial value.
Change condition(s):	Every time a Begin Document AFP structured field is found in the input data stream. The increment value applied every time a change condition occurs is 1. The initVARIABLES keyword may be used to specify a different increment value.

  - **_inMailPieceNumber**

Description:	Input Mail Piece Number Contains the relative number of a Mail Piece within the input data stream. In CIS in connection with AFP data, a mail piece is defined as a "top" level page group (see Glossary on page 445). The complete input file is used to determine the relative number, including items which are excluded in the selection steps.
Reset condition(s):	At the beginning of the print file The initial value set every time a reset condition occurs is 0. The initVARIABLES keyword may be used to specify a different initial value.
Change condition(s):	Every time a non-nested Begin Named Page Group (BNG) structured field is found in the input data stream. Nested BNG's do not represent the beginning of a new mail piece and do not cause a change condition. The increment value applied every time a change condition occurs is 1. The initVARIABLES keyword may be used to specify a different increment value.

- **_inSheetNumber**

Description:	Input Sheet Number Contains the relative number of a sheet within the input data stream. CIS uses the formatting information stored in the FORMDEF in order to determine the way in which one or more pages are grouped into a sheet. The complete input file is used to determine the relative number, including items which are excluded in the selection steps.
Reset condition(s):	At the beginning of the print file The initial value set every time a reset condition occurs is 0. The initVARIABLES keyword may be used to specify a different initial value.
Change condition(s):	Every time a Begin Page (BPG) structured field is found in the input data stream **and** the page being started corresponds to the beginning of a new physical sheet. BPG's which do not correspond to the beginning of a new

sheet do not cause a change condition.

The increment value applied every time a change condition occurs is 1. The initVARIABLES keyword may be used to specify a different increment value.

- **_inPageNumber**

Description:	Input Page Number.
	Contains the relative number of a page within the input data stream.
	The complete input file is used to determine the relative number, including items which are excluded in the selection steps.
Reset condition(s):	At the beginning of the print file
	The initial value set every time a reset condition occurs is 0. The initVARIABLES keyword may be used to specify a different initial value.
Change condition(s):	Every time a Begin Page (BPG) structured field is found in the input data stream.
	The increment value applied every time a change condition occurs is 1. The initVARIABLES keyword may be used to specify a different increment value.

- **_inSheetNumberInMailPiece**

Description:	Sheet Number within Mail Piece
	Contains the relative number of a Sheet sheet within the Mail Piece in the input data stream. CIS uses the formatting information stored in the FORMDEF in order to determine the way in which one or more pages are grouped into a sheet.
	The complete input file is used to determine the relative number, including items which are excluded in the selection steps. The sheet number reflects the original file sequence. It does not reflect any sorting or reformatting operations that may be done in later steps during the same CIS run (use _outSheetNumberInMailPiece in that case).

Reset condition(s):

1. At the beginning of the print file.
2. Every time a Begin Document AFP structured field is found in the input data stream or is assumed to be there (e.g. when processing mixed data).
3. Every time a non-nested Begin Named Page Group (BNG) structured field is found in the input data stream or is inserted as a result of a successful TRIGGER search. Nested BNG's do not represent the beginning of a new mail piece and do not cause a change condition.

The initial value set every time a reset condition occurs is 0. The initVARIABLES keyword may be used to specify a different initial value.

Change condition(s): Same as _inSheetNumber

The increment value applied every time a change condition occurs is 1. The initVARIABLES keyword may be used to specify a different increment value.

- **_inPageNumberInMailPiece**

Description: Page Number within Mail Piece.

Contains the relative number of a Page page within the Mail Piece in the input data stream. The user should be aware that AFP pages may be positioned in a sheet in a different sequence as they appear in the data stream.

The complete input file is used to determine the relative number, including items which are excluded in the selection steps. The page number reflects the original file sequence. It does not reflect any sorting or reformatting operations that may be done in later steps during the same CIS run (use _outPageNumberInMailPiece in that case).

Reset condition(s):

1. At the beginning of the print file.
2. Every time a Begin Document AFP structured field is found in the input data stream or is assumed to be there (e.g. when processing mixed data).
3. Every time a non-nested Begin Named Page Group (BNG) structured field is found in the input data

stream or is inserted as a result of a successful TRIGGER search. Nested BNG's do not represent the beginning of a new mail piece and do not cause a change condition.

The initial value set every time a reset condition occurs is 0. The initVARIABLES keyword may be used to specify a different initial value.

Change condition(s): Same as _inPageNumber

The increment value applied every time a change condition occurs is 1. The initVARIABLES keyword may be used to specify a different increment value.

- <u>**OUPUTOUTPUT FILE RELATED VARIABLES**</u>

  - **_outSegmentNumber**

Description: Output segment number.

Contains the relative number of a segment within the output data stream datasets (see OUTPUTSEG keyword).

Reset condition(s): At the beginning of the output file processing.

The initial value set every time a reset condition occurs is 0. The INITVARIABLES keyword may be used to specify a different initial value.

Change condition(s): Every time an output segment file is opened for processing.

The increment value applied every time a change condition occurs is 1. The INITVARIABLES keyword may be used to specify a different increment value.

  - **_outDocumentNumber**

Description: Output Document number.

Contains the relative number of a document within the output data stream. The items excluded in the selection steps are not considered.

Reset condition(s): At the beginning of the output file processing.

The initial value set every time a reset condition occurs is 0. The initVARIABLES keyword may be used to specify a different initial value.

Change condition(s): Every time a Begin Document (BDT) structured field is written to the output document file.

The increment value applied every time a change condition occurs is 1. The initVARIABLES keyword may be used to specify a different increment value.

- **_outMailPieceNumber**

Description: Output Mail Piece Number.

Contains the relative number of a Mail Piece within the output data stream. The items Excluded in the selection steps are not considered.

Reset condition(s): At the beginning of the output file processing.

The initial value set every time a reset condition occurs is 0. The initVARIABLES keyword may be used to specify a different initial value.

Change condition(s): Every time a "top" level Begin Named Page Group (BNG) structured field is written to the output document file. Nested BNG's do not represent the beginning of a new mail piece and do not cause a change condition.

The increment value applied every time a change condition occurs is 1. The initVARIABLES keyword may be used to specify a different increment value.

- **_outSheetNumber**

Description: Output Sheet Number.

Contains the relative number of a sSheet within the output data stream. The items Excluded in the selection steps are not considered.

Reset condition(s): 1. At the beginning of the output file processing.

The initial value set every time a reset condition occurs is

0. The initVARIABLES keyword may be used to specify a different initial value.

Change condition(s): 1. Every time a Begin Page (BPG) structured field corresponding to a new sheet is written to the output document file. BPG's which do not correspond to the beginning of a sheet do not cause a change condition.

The increment value applied every time a change condition occurs is 1. The initVARIABLES keyword may be used to specify a different increment value.

- **_outPageNumber**

Description: Output Page Number.

Contains the relative number of a Page page within the output data stream. The items Excluded in the selection steps are not considered.

Reset condition(s): At the beginning of the output file processing.

The initial value set every time a reset condition occurs is 0. The initVARIABLES keyword may be used to specify a different initial value.

Change condition(s): Every time a Begin Page (BPG) structured field is written to the output document file.

The increment value applied every time a change condition occurs is 1. The initVARIABLES keyword may be used to specify a different increment value.

- **_outMailPieceNumberInDocument**

Description: Mail Piece Number within Document

Contains the relative number of a Mail Piece within the document in which it is contained. The items Excluded in the selection steps are not considered.

Reset condition(s): Every time a Begin Document (BDT) structured field is written to the output document file.

The initial value set every time a reset condition occurs is 0. The initVARIABLES keyword may be used to specify a different initial value.

Change condition(s): Same as _outMailPieceNumber

The increment value applied every time a change condition occurs is 1. The initVARIABLES keyword may be used to specify a different increment value.

- **_outSheetNumberInMailPiece**

Description: Sheet Number within Mail Piece.

Contains the relative number of a Sheet sheet within the Mail Piece in which it is contained. The items Excluded in the selection steps are not considered.

Reset condition(s): Every time a non-nested Begin named Group (BNG) structured field is written to the output document file.

The initial value set every time a reset condition occurs is 0. The initVARIABLES keyword may be used to specify a different initial value.

Change condition(s): Same as _outSheetNumber

The increment value applied every time a change condition occurs is 1. The initVARIABLES keyword may be used to specify a different increment value.

- **_outPageNumberInMailPiece**

Description: Page Number within Mail Piece.

Contains the relative number of a Page within the Mail Piece in which it is contained. The items Excluded in the selection steps are not considered.

Reset condition(s): 1. Every time a non-nested Begin named Group (BNG) structured field is written to the output document file.

The initial value set every time a reset condition occurs is 0. The initVARIABLES keyword may be used to specify a different initial value.

Change condition(s): Same as _outPageNumber

The increment value applied every time a change condition occurs is 1. The initVARIABLES keyword may be used to specify a different increment value.

- **_outPageNumberInSheet**

Description:	Page Number within Sheetsheet.
	Contains the relative number of a Page page within the sSheet in which it is contained. The items Excluded in the selection steps are not considered.
Reset condition(s):	1. Every time a Begin Page (BPG) structured field is written to the output document file.
	The initial value set every time a reset condition occurs is 0. The initVARIABLES keyword may be used to specify a different initial value.
Change condition(s):	Same as _outPageNumber
	The increment value applied every time a change condition occurs is 1. The initVARIABLES keyword may be used to specify a different increment value.

- **_outNumberOfMailPiecesInDocument**

Description:	Number of mMail Pieceail pieces in Documentdocument
	Contains the number of Mail Pieces within the Document document being written to the output document file.
	Using this variable name in a FIELD keyword causes CIS to store all the selected print data into the internal work file before the output processing is started.
Reset condition(s):	None. This processing variable name cannot be initialized using the INITVARIABLES keyword.
Change condition(s):	None.

- **_outNumberOfSheetsInMailPiece**

Description:	Number of Sheets sheets in Mail Piece
	Contains the number of Sheets sheets within the Mail Piece being written to the output document file.
	Using this variable name in a FIELD keyword causes CIS to store all the selected print data into the internal work

file before the output processing is started.

Reset condition(s): None. This processing variable name cannot be initialized using the INITVARIABLES keyword.

Change condition(s): None.

- **_outNumberOfPagesInMailPiece**

Description: Number of pPages in mMail Pieceail piece.

Contains the number of Pages pages within the Mail Piece being written to the output document file.

Using this variable name in a FIELD keyword causes CIS to store all the selected print data into the internal work file before the output processing is started.

Reset condition(s): None. This processing variable name cannot be initialized using the INITVARIABLES keyword.

Change condition(s): None.

- **_outNumberOfPagesInSheet**

Description: Number of Pages pages in Sheetsheet.

Contains the number of Pages pages within the Sheet sheet being written to the output document file.

Using this variable name in a FIELD keyword causes CIS to store all the selected print data into the internal work file before the output processing is started.

Reset condition(s): None. This processing variable name cannot be initialized using the INITVARIABLES keyword.

Change condition(s): None.

- **_outPageSfOffset**

Description: Page offset (in number of structured fields) within the current output document file segment.

Contains the offset of a page within the current output data stream segment. This is the same 8-bytes value stored in the Object Structured Field Offset Triplet X'58' in the index object file for this page. The items excluded in the selection steps are not considered.

Reset condition(s): Every time a new Output document segment file is created.

The initial value set every time a reset condition occurs is 0. This processing variable name cannot be initialized using the INITVARIABLES keyword.

Change condition(s): Same as _outPageNumber

The increment value applied every time a change condition occurs corresponds to the number of structured fields written before the BPG. The INITVARIABLES keyword may not be used to specify a different increment value.

- **_outCurrentByteOffset**

Description: Current offset (in number of bytes) within the current output document file segment.

Contains the number of bytes written in the current output data stream segment. The items excluded in the selection steps are not considered.

Reset condition(s): Every time a new Output document segment file is created.

The initial value set every time a reset condition occurs is 0. This processing variable name cannot be initialized using the INITVARIABLES keyword.

Change condition(s): Every time information is written to the output document file segment.

The increment applied every time a change condition occurs is equal to the length of the data added to the Output segment file. This INITVARIABLES keyword cannot be used to specify an increment.

- **_outPageByteOffset**

Description: Page offset (in number of bytes) within the current output document file segment.

Contains the offset of a page within the current output data stream segment. This is the same 8-bytes value stored in the Object Byte Offset Triplet X'2D' in the index object file for this page. The items excluded in the selection steps are not considered.

Reset condition(s): Every time a new Output document segment file is created.

The initial value set every time a reset condition occurs is 0. This processing variable name cannot be initialized using the INITVARIABLES keyword.

Change condition(s): Same as _outPageNumber

The increment value applied every time a change condition occurs corresponds to the number of bytes written before the BPG. The INITVARIABLES keyword may not be used to specify a different increment value.

- **_outSheetSfOffset**

Description: Sheet offset (in number of structured fields) within the current output document file segment.

Contains the offset of a sheet within the current output data stream segment. This is the same 8-bytes value stored in the Object Structured Field Offset Triplet X'58' in the index object file for the first page of the Sheet. The items excluded in the selection steps are not considered.

Reset condition(s): Every time a new Output document segment file is created.

The initial value set every time a reset condition occurs is 0. This processing variable name cannot be initialized using the INITVARIABLES keyword.

Change condition(s): Same as _outSheetNumber.

The increment value applied every time a change condition occurs corresponds to the number of structured fields written before the BPG. The INITVARIABLES keyword may not be used to specify a different increment value.

- **_outSheetByteOffset**

Description: Sheet offset (in number of bytes) within the current output document file segment.

Contains the offset of a Sheet within the current output data stream segment. This is the same 8-bytes value stored in the Object Byte Offset Triplet X'2D' in the index object file for the first page of the Sheet. The items excluded in the selection steps are not considered.

Reset condition(s): Every time a new Output document segment file is created.

The initial value set every time a reset condition occurs is 0. This processing variable name cannot be initialized using the INITVARIABLES keyword.

Change condition(s): Same as _outSheetNumber.

The increment value applied every time a change condition occurs corresponds to the number of bytes written before the BPG. The INITVARIABLES keyword may not be used to specify a different increment value.

- **_outMailPieceSfOffset**

Description: Mail piece offset (in number of structured fields) within the current output document file segment.

Contains the offset of a mail piece within the current output data stream segment. This is the same 8-bytes value stored in the Object Structured Field Offset Triplet X'58' in the index object file for the first page of the mail piece. The items excluded in the selection steps are not considered.

Reset condition(s): Every time a new Output document segment file is created.

The initial value set every time a reset condition occurs is 0. This processing variable name cannot be initialized using the INITVARIABLES keyword.

Change condition(s): Same as _outMailPieceNumber.

The increment value applied every time a change condition occurs corresponds to the number of structured fields written before the BPG. The INITVARIABLES keyword may not be used to specify a different increment value.

- **_outMailPieceByteOffset**

Description:	Mail piece offset (in number of bytes) within the current output document file segment.
	Contains the offset of a mail piece within the current output data stream segment. This is the same 8-bytes value stored in the Object Byte Offset Triplet X'2D' in the index object file for the first page of the mail piece. The items excluded in the selection steps are not considered.
Reset condition(s):	Every time a new Output document segment file is created.
	The initial value set every time a reset condition occurs is 0. This processing variable name cannot be initialized using the INITVARIABLES keyword.
Change condition(s):	Same as _outMailPieceNumber.
	The increment value applied every time a change condition occurs corresponds to the number of bytes written before the BPG. The INITVARIABLES keyword may not be used to specify a different increment value.

- <u>OTHER PROCESSING VARIABLES</u>

- **_dateYYYY**

Description:	Current year in long format (e.g. 2001).
Reset condition(s):	The initial value is the current year. The initVARIABLES keyword may be used to specify a different value, from 0 to 9999. The value remains constant regardless of date and time changes within the CIS run.
Change condition(s):	None.

- **_dateYY**

Description:	Current year in short format (e.g. 99, 0, 1, etc.).
Reset condition(s):	None. This processing variable cannot be initialized using the initVARIABLES keyword. Variable _dateYYYY may be used instead.
Change condition(s):	None.

- **_dateMM**

Description:	Current month.
Reset condition(s):	The initial value is the current month number. The initVARIABLES keyword may be used to specify a different value, from 1 to 12. The value remains constant regardless of date and time changes within the CIS run.
Change condition(s):	None.

- **_dateDD**

Description:	Current day.
Reset condition(s):	The initial value is the current day number. The initVARIABLES keyword may be used to specify a different value, from 1 to 31. No checks are made to determine is the combination year/month/day is valid (e.g. the combination 2001, 02, 30 is accepted). The value remains constant regardless of date and time changes within the CIS run.
Change condition(s):	None.

6.3 Appendix C: Description of the Analyst tags

Fig. 4 on page 32 shows the internal CIS workflow. As it is to be expected, the Print Stream Analyzer runs as last process in the chain. The Summary file it creates contains the information gained in all previous steps. The name of the Analyst file is specified using the ANALYSTDD keyword. The ANALYSTCP keyword can be used to specify the encoding of this file.

ANALYST Root-tag

ATTRIBUTENAME Leaf-tag

Value: String

The extract file contains attribute values which corresponds to this attribute name.

AVERAGEPAGESIZE Leaf-tag

Value: Numeric

Unit: Byte

The average page size is the arithmetical mean value. This value is calculated in the following way: All bytes go through the converter divided by the amount of data pages. A data page starts with a Begin Page (BPG) structured field and terminates with an End Page (EPG) structured field.

BCOCA Branch-tag

BCOCATYPE Leaf-tag

Value: Constant String 'CODE39' | 'MSI' | 'UPCA' |

'UPCE' | 'UPC2SUPP' |

'UPC5SUPP' | 'EAN8' |

'EAN13' | 'IND20F5' |

'MAT20F5' | 'ITL20F5' |

'CDB20F7' | 'CODE128' |

'EAN2SUPP' | 'EAN5SUPP' |

'POSTNET' | 'RM4SCC' |

'JPOSTAL' | 'APOSTAL' |

'undefined'

Bar code symbology:

CODE39	Code 39 (3-of-9 Code), AIM USS-39
MSI	MSI (modified Plessey code)
UPCA	UPC/CGPC – Version A
UPCE	UPC/CGPC – Version E
UPC2SUPP	UPC – Two-digit Supplemental (Periodicals)
UPC5SUPP	UPC – Five-digit Supplemental (Paperbacks)
EAN8	EAN-8 (includes JAN-short)
EAN13	EAN-13 (includes JAN-standard)
IND2OF5	Industrial 2-of-5
MAT2OF5	Matrix 2-of-5
ITL20F5	Interleaved 2-of-5, AIM USS-I 2/5
CDB2OF7	Codabar, 2-of-7, AIM USS-Codabar
CODE128	Code 128, AIM USS-128
EAN2SUPP	EAN Two-digit Supplemental
EAN5SUPP	EAN Five-digit Supplemental
POSTNET	POSTNET
RMSCC	RM4SCC
JPOSTAL	Japan Postal Bar Code
APOSTAL	Australia Post Bar Code

For further information about valid code pages and type styles see *IBM Bar Code Object Content Architecture Reference*.

BINNAME Leaf-tag

Value: String

The media type name is defined in the Map Media Type (MMT) structured field. It is either a character-encoded name or a binary identifier.

Example for a character-encoded name:

'ISO A4 white (210 x 297 mm)'

See also COMPONENTID. For further information about media type see *IBM Mixed Object Document Content Architecture Reference.*

BINNUMBER

Leaf-tag

Value: Numeric

BINNUMBER means the Media source selector defined in the Medium Modification Control (MMC) structured field, keyword X'E1'.

There are two formats available:

Format 1

X'01'	Media source ID X'00'
X'02'	Media source ID X'01'
X'03'	Media source ID X'02'
X'04'	Media source ID X'03'
X'41'	Envelope media source
X'64'	Manual feed media source

Format 2

X‘01'	Media source ID X'00'
X'02'	Media source ID X'01'
X'FE'	Media source ID X'FD'
X'FF'	Media source ID X'FE'

COLOR

Leaf-tag

Value: Boolean yes | no

If a color is specified, this tag will set to 'yes'. The color specification is described in the Object Area Descriptor (OBD) structured field, Triplet X'4E'.

For further information about this structured field see *IBM Mixed Object Document Content Architecture Reference*.

COMPONENTID

Leaf-tag

Value: String

This encoded media-type identifier is defined in the Map Media Type (MMT) structured field.

This ID is an ASN.1 Object Identifier (OID), defined in ISO/IEC 8824:1990(E)

Example for a encoded media-type OID:

X'06 072B120004030114'

There is a close coherency between COMPONENTID and BINNAME. These two tags describe the same media type in two different ways.

For further information about media type see *IBM Mixed Object Document Content Architecture Reference*.

CPGID Leaf-tag

Value: Numeric

X'0000'	No CPGID is Assigned
X'0001' – X'FFFE'	CPGID
X'FFFF'	No CPGID is assigned

The code page global identifier is described in the Code Page Descriptor (CPD) structured field.

For further information about this structured field see *IBM Font Object Content Architecture (FOCA) Reference*.

DATA Branch-tag

DOCUMENT Branch-tag

DOCUMENTS Leaf-tag

Value: Numeric

Unit: Number

There are two possibilities of interpretation which depends on the contents of occurrence.

Inside the Branch-tag SUMMARY
Amount of documents which passed through the converter.

Inside the Branch-tag FILECONTENTS
Amount of documents storing in the output document file

DOUBLEBYTECODEPAGE

Leaf-tag

Value:	Boolean	yes | no

The double-byte code page is selected ('yes'), when the CPIRGLen in the Code Page Control (CPC) structured field is set to X'0B'

For further information about this structured field see *IBM Font Object Content Architecture (FOCA) Reference*.

DOUBLEBYTERASTER

Leaf-tag

Value:	Boolean	yes | no

The double-byte raster is selected ('yes'), when the section number inside the Coded Font Index(CFI) structured field is set to a value in the range X'41' up to X'FE'.

For further information about this structured field see *IBM Font Object Content Architecture (FOCA) Reference*.

ENCODING

Leaf-tag

Value: String

No encoding scheme specified |
Single-byte, encoding not specified |
Double-byte, encoding not specified |
Single-byte, IBM-PC Data |
Single-byte, EBCDIC Presentation |
Double-byte, EBCDIC Presentation |
Double-byte, UCS Presentation

This tag specifies the encoding scheme defined in the Code Page Descriptor (CPD) structured field.

For further information about this structured field see *IBM Font Object Content Architecture (FOCA) Reference* .

EXTRACT

Branch-tag

FGID

Leaf-tag

Value: Numeric

X'0000' No information is given
X'0001' – X'FFFE' Registered FGID
X'FFFF' No information is given

The Font Typeface GID is defined in the Font Descriptor (FND) structured field.

For further information about this structured field see *IBM Font Object Content Architecture (FOCA) Reference*.

FILE

Branch-tag

FILECONTENTS

Branch-tag

FILENAME

Leaf-tag

Value: String

The output file name can be an absolute directory name and drive included (e.g. "c:\myDir\myFile.txt"), a relative directory name (e.g. ".\myFile.txt") or only the name (myFile.txt).

Example:

Input parameter OUTPUTDD(.\OutCIS42\Document) will result in a output file name '.\OutCIS42\Document'.

FILESIZE

Leaf-tag

Value: Numeric

Unit: Byte

Contains the file size of the output file in bytes.

Do not compare a output file size with its corresponding input file size. For this comparison use the RESOURCENETSIZE. It is nearly so as large like as the input file size.

See also RESOURCENETSIZE.

FONT Branch-tag

FORM Branch-tag

GCSGID Leaf-tag

Value:	Numeric	X'0000'	No information is Given
		X'0001' – X'FFFE'	Registered GCSGID
		X'FFFF'	No information is Given

The Graphic Character Set GID is defined in the Code Page Descriptor (CPD) structured field and/or in the Font Descriptor structured field (FND).

For further information about these structured fields see *IBM Font Object Content Architecture (FOCA) Reference*.

GOCA Leaf-tag

Value:	Boolean	yes | no

If a Graphic Object Contents Architecture (GOCA) was found in the document this flag is set to 'yes'

IMAGE Branch-tag

IMAGETYPE Leaf-tag

Value: Constant String	'IM image' |
	'IOCA' |
	'IM to IOCA'

This tag describes which kind of Image is located in the document,

See also Begin Image Object (BIM) structured field and Begin IM Image Object (BII) structured field.

For further information about these structured fields see *IBM Mixed Object Document Content Architecture Reference*.

INDEX Branch-tag

INPUTBIN Branch-tag

INPUTDATA Branch-tag

JOBSIZE Leaf-tag

Value: Numeric

Unit: Byte

The job size is the total sum of all document data which passed through the Converter. Resource data are not considered.

MAILPIECES Leaf-tag

Value: Numeric

Unit: Number

There are two possibilities of interpretation which depends on the contents of occurrence.

Inside the Branch-tag SUMMARY
Amount of mail pieces which passed through the converter.

Inside the Branch-tag FILECONTENTS
Amount of mail pieces storing in the output document file

MAXPAGESIZE Leaf-tag

Value: Numeric

Unit: Byte

Size of the largest data page passed through the Converter. A data page starts with a Begin Page (BPG) structured field and terminates with an End Page (EPG) structured field.

MEDIUM Branch-tag

MICR Leaf-tag

Value: Boolean yes | no

If Magnetic Ink Character Recognition (MICR) is active this tag is set to 'yes'. The MICR bit is specified in the Font Character Set (FNC) structured field.

For further information about this structured field see *IBM Font Object Content Architecture (FOCA) Reference*.

MINPAGESIZE

Leaf-tag

Value: Numeric

Unit: Byte

Size of the shortest data page passed through the Converter. A data page starts with a Begin Page (BPG) structured field and terminates with an End Page (EPG) structured field.

NUP

Leaf-tag

Value: Boolean yes | no

If N-up format control (MMC, keyword X'FC') is available this tag is set to 'yes'.

For further information about the MMC structured field see *IBM Mixed Object Document Content Architecture Reference*.

OBJECTCONTAINER

Leaf-tag

Value: Boolean (yes | no)

If an object container was found in the document this flag is set to 'yes'

ORIENTATION

Leaf-tag

Value: Constant String 'portrait' | 'landscape'

'portrait 90' |

'landscape 90' |

'portrait 180' |

'landscape 180' |

'portrait 270' |
'landscape 270'

OUTLINE

Leaf-tag

Value:	Boolean	yes | no

If a outline font was found, this tag is set to 'yes'.

For further information about outline fonts see Coded Font Index (CFI) structured field described in the *IBM Font Object Content Architecture (FOCA) Reference*.

OUTPUTBIN

Branch-tag

OUTPUTDATA

Branch-tag

PAGES

Leaf-tag

Value:	Numeric
Unit:	Number

There are two possibilities of interpretation which depends on the contents of occurrence.

- Inside the Branch-tag SUMMARY
  Amount of data pages passed through the converter.
- Inside the Branch-tag FILECONTENTS
  Amount of data pages storing in the output document file

A data page starts with a Begin Page (BPG) structured field and terminates with an End Page (EPG) structured field.

PAGEUNITBASE

Leaf-tag

Value:	Constant String	'ten inches' | 'ten centimeters'

Specifies the unit base of the medium co-ordinate system, defined in the Medium Descriptor (MDD) structured field.

For further information about this structured field see *IBM Mixed Object Document Content Architecture Reference*.

PAGEUNITSPERUNITBASE

Leaf-tag

Value: Numeric

Specifies the number of units per unit base of the medium coordinate system, defined in the Medium Descriptor (MDD) structured field.

For further information about this structured field see *IBM Mixed Object Document Content Architecture Reference*.

PATTERN

Leaf-tag

Value: String Laser Matrix N-bit Wide |
CID keyed font (type 0) |
PFB (type 1)

The Pattern Technology Identifier parameter specified the technologies for the font graphic patterns for this font. This identifier is specified in the Font Character Set (FNC) structured field.

For further information about this structured field see *IBM Font Object Content Architecture (FOCA) Reference*.

PRINTCONTROL

Leaf-tag

Value: Constant String 'simplex' |
'duplex' |
'mixplex'

Specifies whether data is generated on the front side of the sheet or on both sides. The print control is specified in the Medium Modification Control (MMC) structured field, keyword X'F4'.

For further information about this structured fields see *IBM Mixed Object Document Content Architecture Reference*.

RELATIVEMETRICS

Leaf-tag

Value: Boolean yes | no

The relative metrics is defined by the X/Y-unit-base inside the Font Control (FNC) structured field. If a relative metrics is defined this tag is set to 'yes'.

Further information about this structured field see *IBM Font Object Content Architecture (FOCA) Reference*.

RESOLUTION

Leaf-tag

Value: Numeric

Unit: Pels per 10 Inches

The resolution is specified in the Font Control (FNC) structured field.

For further information about this structured field see *IBM Font Object Content Architecture (FOCA) Reference*.

RESOURCE

Branch-tag

RESOURCEDATA

Branch-tag

RESOURCENAME

Leaf-tag

Value: String

Name of the resource. This name corresponds to its Begin structured field.

For example 'Bar code object':

The corresponding Begin structured field is the 'Begin Bar Code Object' (BBC). The name is specified in Byte 0-7 or in the Triplet X'02'

Further information about Begin structured fields see IBM Mixed Object Document Content Architecture Reference.

RESOURCENETSIZE

Leaf-tag

Value: Numeric

Unit: Byte

Specifies the net size of one resource type. This size is nearly equal to the size of the input resource.

There are two RESOURCENETSIZE values available:

derived from input data, and calculated from the written resource output data.

These two values must not have the absolute same size they can be different.

RESOURCETYPE

Leaf-tag

Value:	Constant String	'form definition' | 'page definition' | 'page segment' | 'overlay' | 'coded font' | 'font character set (raster) | 'code page type' | 'font character set (outline)' 'color mapping table' | 'font character set (GRID)' | code page (GRID)'

The resource type describes which kind of resource is in use.

For further information about resource object types see Resource Object Type Triplet X'21' in the *IBM Mixed Object Document Content Architecture Reference* specs.

ROTATION00

Leaf-tag

Value:	Boolean	yes | no

Set to 'yes' If when the character rotation 0° is available for this font this tag is set to 'yes'.

Font Orientation (FNO) structured field is responsible for character rotation.

For further information about this structured field see *IBM Font Object Content Architecture (FOCA) Reference*.

ROTATION180

Leaf-tag

Value:	Boolean	yes | no

Set to 'yes' when the character rotation 180° is available for this font. If the character rotation 180° is available for this font this tag is set to 'yes'.

Font Orientation (FNO) structured field is responsible for character rotation.

Further information about this structured field see IBM Font Object Content Architecture (FOCA) Reference.

ROTATION270

Leaf-tag

Value:	Boolean	yes | no

Set to 'yes' when the character rotation 270° is available for this font. If the character rotation 270° is available for this font this tag is set to 'yes'.

Font Orientation (FNO) structured field is responsible for character rotation.

For further information about this structured field see *IBM Font Object Content Architecture (FOCA) Reference*.

ROTATION90

Leaf-tag

Value:	Boolean	yes | no

Set to 'yes' when the character rotation 90° is available for this font. If the character rotation 90° is available for this font this tag is set to 'yes'.

Font Orientation (FNO) structured field is responsible for character rotation.

For further information about this structured field see *IBM Font Object Content Architecture (FOCA) Reference*.

SHEETS

Leaf-tag

Value:	Numeric
Unit:	Number

There are two possibilities of interpretation which depends on the contents of occurrence.

- Inside the Branch-tag SUMMARY
  Amount of documents passed through the converter.
- Inside the Branch-tag FILECONTENTS
  Amount of documents storing in the output document file

The sheet counter result is based on data pages. A data page starts with a Begin Page (BPG) structured field and terminates with an End Page (EPG) structured field.

SUMMARY

Branch-tag

TYPEFACE

Leaf-tag

Value: String

Typeface Description contains descriptive information about the font family name (e.g. 'Times New Roman'). It's a 32 characters long string. Unused characters are filled up with blanks.

The Font Descriptor (FND) structured field specifies the typeface description.

For further information about this structured field see *IBM Font Object Content Architecture (FOCA) Reference*.

VERSION

Leaf-tag

Value: String

This tag specifies the CIS version which generates this analyst file.

X2UP

Leaf-tag

Value: Boolean yes | no

If X2-up enable this tag is set to 'yes'

Further information about X2-up see Siemens Nixdorf Printer Data Stream (SPDS) Reference Manual and the PdsInfo PDSFLG3 Default processing flag.

XSIZE

Leaf-tag

Value: Numeric

Specifies the extent of the medium presentation space along the X axis, defined in the Medium Descriptor (MDD) structured field. A value of 0x00 indicates that the extent is not specified.

For further information about this structured field see *IBM Mixed Object Document Content Architecture Reference*.

YSIZE Leaf-tag

Value: Numeric

Specifies the extent of the medium presentation space along the Y axis, defined in the Medium Descriptor (MDD) structured field. A value of 0x00 indicates the extent is not specified.

For further information about this structured field see *IBM Mixed Object Document Content Architecture Reference*.

6.4 Appendix D: Code Page Description

The following tables document the code points supported in each standard code page available in CIS. Each table cell contains the textual representation of the code point and the corresponding Unicode code as defined by the Unicode Consortium.

IBM_0037

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	à 00e0	á 00e1	ã 00e3	å 00e5	ç 00e7	ñ 00f1	¢ 00a2	. 002e	< 003c	(0028	+ 002b	| 007c
5	& 0026	é 00e9	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	! 0021	$ 0024	* 002a	) 0029	; 003b	¬ 00ac
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	Ã 00c3	Å 00c5	Ç 00c7	Ñ 00d1	¦ 00a6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	æ 00e6	¸ 00b8	Æ 00c6	¤ 00a4
a	µ 00b5	~ 007e	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	^ 005e	£ 00a3	¥ 00a5	· 00b7	© 00a9	§ 00a7	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	[005b	] 005d	¯ 00af	¨ 00a8	´ 00b4	× 00d7
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	ö 00f6	ò 00f2	ó 00f3	õ 00f5
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	ü 00fc	ù 00f9	ú 00fa	ÿ 00ff
e	\ 005c	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	Ö 00d6	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 105 IBM_0037: EBCDIC (USA and Canada)

IBM_0256

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	à 00e0	á 00e1	ã 00e3	å 00e5	ç 00e7	ñ 00f1	[005b	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	é 00e9	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	] 005d	$ 0024	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	Ã 00c3	Å 00c5	Ç 00c7	Ñ 00d1	¦ 00a6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	æ 00e6	¸ 00b8	Æ 00c6	¤ 00a4
a	µ 00b5	~ 007e	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	¢ 00a2	£ 00a3	¥ 00a5	₧ 20a7	ƒ 0192	§ 00a7	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	¬ 00ac	| 007c	¯ 00af	¨ 00a8	´ 00b4	‗ 2017
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	ö 00f6	ò 00f2	ó 00f3	õ 00f5
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	ı 0131	û 00fb	ü 00fc	ù 00f9	ú 00fa	ÿ 00ff
e	\ 005c	2007	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	Ö 00d6	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 106 IBM_0256: EBCDIC Netherlands

IBM_0273

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	{ 007b	à 00e0	á 00e1	ã 00e3	å 00e5	ç 00e7	ñ 00f1	Ä 00c4	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	é 00e9	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	~ 007e	Ü 00dc	$ 0024	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Â 00c2	[005b	À 00c0	Á 00c1	Ã 00c3	Å 00c5	Ç 00c7	Ñ 00d1	ö 00f6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	` 0060	: 003a	# 0023	§ 00a7	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	æ 00e6	¸ 00b8	Æ 00c6	¤ 00a4
a	µ 00b5	ß 00df	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	¢ 00a2	£ 00a3	¥ 00a5	· 00b7	© 00a9	@ 0040	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	¬ 00ac	| 007c	¯ 00af	¨ 00a8	´ 00b4	× 00d7
c	ä 00e4	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	¦ 00a6	ò 00f2	ó 00f3	õ 00f5
d	ü 00fc	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	} 007d	ù 00f9	ú 00fa	ÿ 00ff
e	Ö 00d6	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	\ 005c	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	] 005d	Ù 00d9	Ú 00da	· 009f

Fig. 107 IBM_0273: EBCDIC Austria, Germany

IBM_0277

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	à 00e0	á 00e1	ã 00e3	} 007d	ç 00e7	ñ 00f1	# 0023	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	é 00e9	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	¤ 00a4	Å 00c5	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	Ã 00c3	$ 0024	Ç 00c7	Ñ 00d1	ø 00f8	, 002c	% 0025	_ 005f	> 003e	? 003f
7	¦ 00a6	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	` 0060	: 003a	Æ 00c6	Ø 00d8	' 0027	= 003d	" 0022
8	@ 0040	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	{ 007b	¸ 00b8	[005b	] 005d
a	µ 00b5	ü 00fc	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	¢ 00a2	£ 00a3	¥ 00a5	· 00b7	© 00a9	§ 00a7	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	¬ 00ac	| 007c	¯ 00af	¨ 00a8	´ 00b4	× 00d7
c	æ 00e6	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	ö 00f6	ò 00f2	ó 00f3	õ 00f5
d	å 00e5	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	~ 007e	ù 00f9	ú 00fa	ÿ 00ff
e	\ 005c	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	Ö 00d6	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 108 IBM_0277: EBCDIC Denmark, Norway

IBM_0278

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	{ 007b	à 00e0	á 00e1	ã 00e3	} 007d	ç 00e7	ñ 00f1	§ 00a7	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	` 0060	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	¤ 00a4	Å 00c5	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Â 00c2	# 0023	À 00c0	Á 00c1	Ã 00c3	$ 0024	Ç 00c7	Ñ 00d1	ö 00f6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	\ 005c	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	é 00e9	: 003a	Ä 00c4	Ö 00d6	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	æ 00e6	¸ 00b8	Æ 00c6	] 005d
a	µ 00b5	ü 00fc	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	¢ 00a2	£ 00a3	¥ 00a5	· 00b7	© 00a9	[005b	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	¬ 00ac	| 007c	¯ 00af	¨ 00a8	´ 00b4	× 00d7
c	ä 00e4	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	¦ 00a6	ò 00f2	ó 00f3	õ 00f5
d	å 00e5	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	~ 007e	ù 00f9	ú 00fa	ÿ 00ff
e	É 00c9	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	@ 0040	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 109 IBM_0278: EBCDIC Finland, Sweden

IBM_0280

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	{ 007b	á 00e1	ã 00e3	å 00e5	\ 005c	ñ 00f1	° 00b0	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	] 005d	ê 00ea	ë 00eb	} 007d	í 00ed	î 00ee	ï 00ef	~ 007e	ß 00df	é 00e9	$ 0024	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	Ã 00c3	Å 00c5	Ç 00c7	Ñ 00d1	ò 00f2	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	ù 00f9	: 003a	£ 00a3	§ 00a7	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	[005b	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	æ 00e6	¸ 00b8	Æ 00c6	¤ 00a4
a	µ 00b5	ì 00ec	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	¢ 00a2	# 0023	¥ 00a5	· 00b7	© 00a9	@ 0040	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	¬ 00ac	| 007c	¯ 00af	¨ 00a8	´ 00b4	× 00d7
c	à 00e0	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	ö 00f6	¦ 00a6	ó 00f3	õ 00f5
d	è 00e8	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	ü 00fc	` 0060	ú 00fa	ÿ 00ff
e	ç 00e7	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	Ö 00d6	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 110 IBM_0280: EBCDIC Italy

IBM_0284

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	à 00e0	á 00e1	ã 00e3	å 00e5	ç 00e7	¦ 00a6	[005b	. 002e	< 003c	(0028	+ 002b	| 007c
5	& 0026	é 00e9	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	] 005d	$ 0024	* 002a	) 0029	; 003b	¬ 00ac
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	Ã 00c3	Å 00c5	Ç 00c7	# 0023	ñ 00f1	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	` 0060	: 003a	Ñ 00d1	@ 0040	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	æ 00e6	¸ 00b8	Æ 00c6	¤ 00a4
a	µ 00b5	¨ 00a8	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	¢ 00a2	£ 00a3	¥ 00a5	· 00b7	© 00a9	§ 00a7	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	^ 005e	! 0021	¯ 00af	~ 007e	´ 00b4	× 00d7
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	ö 00f6	ò 00f2	ó 00f3	õ 00f5
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	ü 00fc	ù 00f9	ú 00fa	ÿ 00ff
e	\ 005c	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	Ö 00d6	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 111 IBM_0284: EBCDIC Spain, Latin America (Spanish)

IBM_0285

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	à 00e0	á 00e1	ã 00e3	å 00e5	ç 00e7	ñ 00f1	$ 0024	. 002e	< 003c	(0028	+ 002b	| 007c
5	& 0026	é 00e9	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	! 0021	£ 00a3	* 002a	) 0029	; 003b	¬ 00ac
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	Ã 00c3	Å 00c5	Ç 00c7	Ñ 00d1	¦ 00a6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	æ 00e6	¸ 00b8	Æ 00c6	¤ 00a4
a	µ 00b5	¯ 00af	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	¢ 00a2	[005b	¥ 00a5	· 00b7	© 00a9	§ 00a7	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	^ 005e	] 005d	~ 007e	¨ 00a8	´ 00b4	× 00d7
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	ö 00f6	ò 00f2	ó 00f3	õ 00f5
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	ü 00fc	ù 00f9	ú 00fa	ÿ 00ff
e	\ 005c	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	Ö 00d6	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 112 IBM_0285: EBCDIC United Kingdom

IBM_0290

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	｡ ff61	｢ ff62	｣ ff63	､ ff64	･ ff65	ｦ ff66	ｧ ff67	ｨ ff68	ｩ ff69	£ 00a3	. 002e	< 003c	(0028	+ 002b	| 007c
5	& 0026	ｪ ff6a	ｫ ff6b	ｬ ff6c	ｭ ff6d	ｮ ff6e	ｯ ff6f	SUB 001a	ｰ ff70	SUB 001a	! 0021	¥ 00a5	* 002a	) 0029	; 003b	¬ 00ac
6	- 002d	/ 002f	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	SUB 001a	, 002c	% 0025	_ 005f	> 003e	? 003f
7	[005b	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	] 005d	ｱ ff71	ｲ ff72	ｳ ff73	ｴ ff74	ｵ ff75	ｶ ff76	ｷ ff77	ｸ ff78	ｹ ff79	ｺ ff7a	q 0071	ｻ ff7b	ｼ ff7c	ｽ ff7d	ｾ ff7e
9	ｿ ff7f	ﾀ ff80	ﾁ ff81	ﾂ ff82	ﾃ ff83	ﾄ ff84	ﾅ ff85	ﾆ ff86	ﾇ ff87	ﾈ ff88	ﾉ ff89	r 0072	SUB 001a	ﾊ ff8a	ﾋ ff8b	ﾌ ff8c
a	~ 007e	‾ 203e	ﾍ ff8d	ﾎ ff8e	ﾏ ff8f	ﾐ ff90	ﾑ ff91	ﾒ ff92	ﾓ ff93	ﾔ ff94	ﾕ ff95	s 0073	ﾖ ff96	ﾗ ff97	ﾘ ff98	ﾙ ff99
b	^ 005e	¢ 00a2	\ 005c	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	ﾚ ff9a	ﾛ ff9b	ﾜ ff9c	ﾝ ff9d	ﾞ ff9e	ﾟ ff9f
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	$ 0024	SUB 001a	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	· 009f

Fig. 113 IBM_0290: EBCDIC Japanese Katakana

IBM_0297

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	@ 0040	á 00e1	ã 00e3	å 00e5	\ 005c	ñ 00f1	° 00b0	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	{ 007b	ê 00ea	ë 00eb	} 007d	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	§ 00a7	$ 0024	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	Ã 00c3	Å 00c5	Ç 00c7	Ñ 00d1	ù 00f9	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	µ 00b5	: 003a	£ 00a3	à 00e0	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	[005b	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	æ 00e6	¸ 00b8	Æ 00c6	¤ 00a4
a	` 0060	¨ 00a8	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	¢ 00a2	# 0023	¥ 00a5	· 00b7	© 00a9	] 005d	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	¬ 00ac	| 007c	¯ 00af	~ 007e	´ 00b4	× 00d7
c	é 00e9	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	ö 00f6	ò 00f2	ó 00f3	õ 00f5
d	è 00e8	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	ü 00fc	¦ 00a6	ú 00fa	ÿ 00ff
e	ç 00e7	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	Ö 00d6	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 114 IBM_0297: EBCDIC France

IBM_0367

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	DEL 007f
8	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
9	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
b	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
c	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
f	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a

Fig. 115 IBM_0367: ANSI X3.4 ASCII Standard USA

IBM_0420

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	ﹼ fe7c	ﹽ fe7d	ـ 0640	· f8fc	ﺀ fe80	ﺁ fe81	ﺂ fe82	ﺃ fe83	¢ 00a2	. 002e	< 003c	(0028	+ 002b	| 007c
5	& 0026	ﺄ fe84	ﺅ fe85	SUB 001a	SUB 001a	ﺋ fe8b	ﺍ fe8d	ﺎ fe8e	ﺏ fe8f	ﺑ fe91	! 0021	$ 0024	* 002a	) 0029	; 003b	¬ 00ac
6	- 002d	/ 002f	ﺓ fe93	ﺕ fe95	ﺗ fe97	ﺙ fe99	ﺛ fe9b	ﺝ fe9d	ﺟ fe9f	ﺡ fea1	¦ 00a6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ﺣ fea3	ﺥ fea5	ﺧ fea7	ﺩ fea9	ﺫ feab	ﺭ fead	ﺯ feaf	· f8f6	ﺳ feb3	، 060c	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	· f8f5	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	ﺷ feb7	· f8f4	ﺻ febb	· f8f7	ﺿ febf	ﻃ fec3
9	ﻇ fec7	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ﻉ fec9	ﻊ feca	ﻋ fecb	ﻌ fecc	ﻍ fecd	ﻎ fece
a	ﻏ fecf	÷ 00f7	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	ﻑ fed0	ﻑ fed1	ﻓ fed3	ﻕ fed5	ﻗ fed7	ﻙ fed9
b	ﻛ fedb	ﻝ fedd	ﻵ fef5	ﻶ fef6	ﻷ fef7	ﻸ fef8	SUB 001a	SUB 001a	ﻻ fefb	ﻼ fefc	ﻟ fedf	ﻡ fee1	ﻣ fee3	ﻥ fee5	ﻧ fee7	ﻩ fee9
c	؛ 061b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ﻫ feeb	SUB 001a	ﻬ feec	SUB 001a	ﻭ feed
d	؟ 061f	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	ﻯ feef	ﻰ fef0	ﻱ fef1	ﻲ fef2	ﻳ fef3	٠ 0660
e	× 00d7	2007	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	١ 0661	٢ 0662	SUB 001a	٣ 0663	٤ 0664	٥ 0665
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	SUB 001a	٦ 0666	٧ 0667	٨ 0668	٩ 0669	· 009f

Fig. 116 IBM_0420: EBCDIC Arabic

IBM_0423

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	Α 0391	Β 0392	Γ 0393	Δ 0394	Ε 0395	Ζ 0396	Η 0397	Θ 0398	Ι 0399	[005b	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	Κ 039a	Λ 039b	Μ 039c	Ν 039d	Ξ 039e	Ο 039f	Π 03a0	Ρ 03a1	Σ 03a3	] 005d	$ 0024	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Τ 03a4	Υ 03a5	Φ 03a6	Χ 03a7	Ψ 03a8	Ω 03a9	SUB 001a	SUB 001a	| 007c	, 002c	% 0025	_ 005f	> 003e	? 003f
7	SUB 001a	Ά 0386	Έ 0388	Ή 0389	SP 00a0	Ί 038a	Ό 038c	Ύ 038e	Ώ 038f	` 0060	: 003a	£ 00a3	§ 00a7	' 0027	= 003d	" 0022
8	Ä 00c4	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	α 03b1	β 03b2	γ 03b3	δ 03b4	ε 03b5	ζ 03b6
9	Ö 00d6	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	η 03b7	θ 03b8	ι 03b9	κ 03ba	λ 03bb	μ 03bc
a	Ü 00dc	¨ 00a8	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	ν 03bd	ξ 03be	ο 03bf	π 03c0	ρ 03c1	σ 03c3
b	SUB 001a	ά 03ac	έ 03ad	ή 03ae	ϊ 03ca	ί 03af	ό 03cc	ύ 03cd	ϋ 03cb	ώ 03ce	ς 03c2	τ 03c4	υ 03c5	φ 03c6	χ 03c7	ψ 03c8
c	¸ 00b8	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ω 03c9	â 00e2	à 00e0	ä 00e4	ê 00ea
d	´ 00b4	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	± 00b1	é 00e9	è 00e8	ë 00eb	î 00ee	ï 00ef
e	° 00b0	SUB 001a	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	½ 00bd	ö 00f6	ô 00f4	û 00fb	ù 00f9	ü 00fc
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	ÿ 00ff	ç 00e7	Ç 00c7	SUB 001a	SUB 001a	· 009f

Fig. 117 IBM_0423: EBCDIC Greek

IBM_0424

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	א 05d0	ב 05d1	ג 05d2	ד 05d3	ה 05d4	ו 05d5	ז 05d6	ח 05d7	ט 05d8	¢ 00a2	. 002e	< 003c	(0028	+ 002b	| 007c
5	& 0026	י 05d9	ך 05da	כ 05db	ל 05dc	ם 05dd	מ 05de	ן 05df	נ 05e0	ס 05e1	! 0021	$ 0024	* 002a	) 0029	; 003b	¬ 00ac
6	- 002d	/ 002f	ע 05e2	ף 05e3	פ 05e4	ץ 05e5	צ 05e6	ק 05e7	ר 05e8	ש 05e9	¦ 00a6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	SUB 001a	ת 05ea	SUB 001a	SUB 001a	SP 00a0	SUB 001a	SUB 001a	SUB 001a	‗ 2017	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	SUB 001a	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	SUB 001a	SUB 001a	SUB 001a	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	SUB 001a	SUB 001a	SUB 001a	¸ 00b8	SUB 001a	¤ 00a4
a	µ 00b5	~ 007e	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	® 00ae
b	^ 005e	£ 00a3	¥ 00a5	• 2022	© 00a9	§ 00a7	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	[005b	] 005d	‾ 203e	¨ 00a8	´ 00b4	× 00d7
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	\ 005c	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	SUB 001a	SUB 001a	SUB 001a	SUB 001a	· 009f

Fig. 118 IBM_0424: EBCDIC Hebrew

IBM_0500

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	à 00e0	á 00e1	ã 00e3	å 00e5	ç 00e7	ñ 00f1	[005b	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	é 00e9	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	] 005d	$ 0024	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	Ã 00c3	Å 00c5	Ç 00c7	Ñ 00d1	¦ 00a6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	æ 00e6	¸ 00b8	Æ 00c6	¤ 00a4
a	µ 00b5	~ 007e	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	¢ 00a2	£ 00a3	¥ 00a5	· 00b7	© 00a9	§ 00a7	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	¬ 00ac	| 007c	¯ 00af	¨ 00a8	´ 00b4	× 00d7
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	ö 00f6	ò 00f2	ó 00f3	õ 00f5
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	ü 00fc	ù 00f9	ú 00fa	ÿ 00ff
e	\ 005c	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	Ö 00d6	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 119 IBM_0500: EBCDIC International Latin-1

IBM_0813

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	DEL 007f
8	· 0080	· 0081	· 0082	· 0083	· 0084	· 0085	· 0086	· 0087	· 0088	· 0089	· 008a	· 008b	· 008c	· 008d	· 008e	· 008f
9	· 0090	· 0091	· 0092	· 0093	· 0094	· 0095	· 0096	· 0097	· 0098	· 0099	· 009a	· 009b	· 009c	· 009d	· 009e	· 009f
a	SP 00a0	‘ 2018	’ 2019	£ 00a3	SUB 001a	SUB 001a	¦ 00a6	§ 00a7	¨ 00a8	© 00a9	SUB 001a	« 00ab	¬ 00ac	- 00ad	SUB 001a	― 2015
b	° 00b0	± 00b1	² 00b2	³ 00b3	΄ 00b4	΅ 0385	Ά 0386	· 0387	Έ 0388	Ή 0389	Ί 038a	» 00bb	Ό 038c	½ 00bd	Ύ 038e	Ώ 038f
c	ΐ 0390	Α 0391	Β 0392	Γ 0393	Δ 0394	Ε 0395	Ζ 0396	Η 0397	Θ 0398	Ι 0399	Κ 039a	Λ 039b	Μ 039c	Ν 039d	Ξ 039e	Ο 039f
d	Π 03a0	Ρ 03a1	SUB 001a	Σ 03a3	Τ 03a4	Υ 03a5	Φ 03a6	Χ 03a7	Ψ 03a8	Ω 03a9	Ϊ 03aa	Ϋ 03ab	ά 03ac	έ 03ad	ή 03ae	ί 03af
e	ΰ 03b0	α 03b1	β 03b2	γ 03b3	δ 03b4	ε 03b5	ζ 03b6	η 03b7	θ 03b8	ι 03b9	κ 03ba	λ 03bb	μ 03bc	ν 03bd	ξ 03be	ο 03bf
f	π 03c0	ρ 03c1	ς 03c2	σ 03c3	τ 03c4	υ 03c5	φ 03c6	χ 03c7	ψ 03c8	ω 03c9	ϊ 03ca	ϋ 03cb	ό 03cc	ύ 03cd	ώ 03ce	SUB 001a

Fig. 120 IBM_0813: ISO 8859-7 Greek/Latin

IBM_0819

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	DEL 007f
8	· 0080	· 0081	· 0082	· 0083	· 0084	· 0085	· 0086	· 0087	· 0088	· 0089	· 008a	· 008b	· 008c	· 008d	· 008e	· 008f
9	· 0090	· 0091	· 0092	· 0093	· 0094	· 0095	· 0096	· 0097	· 0098	· 0099	· 009a	· 009b	· 009c	· 009d	· 009e	· 009f
a	SP 00a0	¡ 00a1	¢ 00a2	£ 00a3	¤ 00a4	¥ 00a5	¦ 00a6	§ 00a7	¨ 00a8	© 00a9	ª 00aa	« 00ab	¬ 00ac	- 00ad	® 00ae	¯ 00af
b	° 00b0	± 00b1	² 00b2	³ 00b3	´ 00b4	µ 00b5	¶ 00b6	· 00b7	¸ 00b8	¹ 00b9	º 00ba	» 00bb	¼ 00bc	½ 00bd	¾ 00be	¿ 00bf
c	À 00c0	Á 00c1	Â 00c2	Ã 00c3	Ä 00c4	Å 00c5	Æ 00c6	Ç 00c7	È 00c8	É 00c9	Ê 00ca	Ë 00cb	Ì 00cc	Í 00cd	Î 00ce	Ï 00cf
d	Ð 00d0	Ñ 00d1	Ò 00d2	Ó 00d3	Ô 00d4	Õ 00d5	Ö 00d6	× 00d7	Ø 00d8	Ù 00d9	Ú 00da	Û 00db	Ü 00dc	Ý 00dd	Þ 00de	ß 00df
e	à 00e0	á 00e1	â 00e2	ã 00e3	ä 00e4	å 00e5	æ 00e6	ç 00e7	è 00e8	é 00e9	ê 00ea	ë 00eb	ì 00ec	í 00ed	î 00ee	ï 00ef
f	ð 00f0	ñ 00f1	ò 00f2	ó 00f3	ô 00f4	õ 00f5	ö 00f6	÷ 00f7	ø 00f8	ù 00f9	ú 00fa	û 00fb	ü 00fc	ý 00fd	þ 00fe	ÿ 00ff

Fig. 121 IBM_0819: Same as ISO_8859-1

IBM_0833

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SUB 001a	ffa0	ㄱ ffa1	ㄲ ffa2	ㄳ ffa3	ㄴ ffa4	ㄵ ffa5	ㄶ ffa6	ㄷ ffa7	¢ 00a2	. 002e	< 003c	(0028	+ 002b	| 007c
5	& 0026	SUB 001a	ㄸ ffa8	ㄹ ffa9	ㄺ ffaa	ㄻ ffab	ㄼ ffac	ㄽ ffad	ㄾ ffae	ㄿ ffaf	! 0021	$ 0024	* 002a	) 0029	; 003b	¬ 00ac
6	- 002d	/ 002f	ㅀ ffb0	ㅁ ffb1	ㅂ ffb2	ㅃ ffb3	ㅄ ffb4	ㅅ ffb5	ㅆ ffb6	ㅇ ffb7	¦ 00a6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	[005b	SUB 001a	ㅈ ffb8	ㅉ ffb9	ㅊ ffba	ㅋ ffbb	ㅌ ffbc	ㅍ ffbd	ㅎ ffbe	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	] 005d	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	ㅏ ffc2	ㅐ ffc3	ㅑ ffc4	ㅒ ffc5	ㅓ ffc6	ㅔ ffc7
9	SUB 001a	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ㅕ ffca	ㅖ ffcb	ㅗ ffcc	ㅘ ffcd	ㅙ ffce	ㅚ ffcf
a	‾ 203e	~ 007e	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	ㅛ ffd2	ㅜ ffd3	ㅝ ffd4	ㅞ ffd5	ㅟ ffd6	ㅠ ffd7
b	^ 005e	SUB 001a	\ 005c	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	ㅡ ffda	ㅢ ffdb	ㅣ ffdc	SUB 001a	SUB 001a	SUB 001a
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	₩ 20a9	SUB 001a	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	· 009f

Fig. 122 IBM_0833: EBCDIC Korean

IBM_0836

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	£ 00a3	. 002e	< 003c	(0028	+ 002b	| 007c
5	& 0026	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	! 0021	¥ 00a5	* 002a	) 0029	; 003b	¬ 00ac
6	- 002d	/ 002f	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	¦ 00a6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	SUB 001a	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
9	SUB 001a	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
a	~ 007e	‾ 203e	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
b	^ 005e	SUB 001a	\ 005c	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	[005b	] 005d	SUB 001a	SUB 001a	SUB 001a	SUB 001a
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	$ 0024	SUB 001a	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	· 009f

Fig. 123 IBM_0836: EBCDIC South-China

IBM_0838

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	ก 0e01	ข 0e02	ฃ 0e03	ค 0e04	ฅ 0e05	ฆ 0e06	ง 0e07	[005b	¢ 00a2	. 002e	< 003c	(0028	+ 002b	| 007c
5	& 0026	่ 0e48	จ 0e08	ฉ 0e09	ช 0e0a	ซ 0e0b	ฌ 0e0c	ญ 0e0d	ฎ 0e0e	] 005d	! 0021	$ 0024	* 002a	) 0029	; 003b	¬ 00ac
6	- 002d	/ 002f	ฏ 0e0f	ฐ 0e10	ฑ 0e11	ฒ 0e12	ณ 0e13	ด 0e14	ต 0e15	^ 005e	¦ 00a6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	฿ 0e3f	๎ 0e4e	ถ 0e16	ท 0e17	ธ 0e18	น 0e19	บ 0e1a	ป 0e1b	ผ 0e1c	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	๏ 0e4f	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	ฝ 0e1d	พ 0e1e	ฟ 0e1f	ภ 0e20	ม 0e21	ย 0e22
9	๚ 0e5a	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ร 0e23	ฤ 0e24	ล 0e25	ฦ 0e26	ว 0e27	ศ 0e28
a	๛ 0e5b	~ 007e	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	ษ 0e29	ส 0e2a	ห 0e2b	ฬ 0e2c	อ 0e2d	ฮ 0e2e
b	๐ 0e50	๑ 0e51	๒ 0e52	๓ 0e53	๔ 0e54	๕ 0e55	๖ 0e56	๗ 0e57	๘ 0e58	๙ 0e59	ฯ 0e2f	ะ 0e30	ั 0e31	า 0e32	ำ 0e33	ิ 0e34
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	้ 0e49	ี 0e35	ึ 0e36	ื 0e37	ุ 0e38	ู 0e39
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	ฺ 0e3a	เ 0e40	แ 0e41	โ 0e42	ใ 0e43	ไ 0e44
e	\ 005c	๊ 0e4a	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	ๅ 0e45	ๆ 0e46	็ 0e47	่ 0e48	้ 0e49	๊ 0e4a
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	๋ 0e4b	์ 0e4c	ํ 0e4d	๋ 0e4b	์ 0e4c	· 009f

Fig. 124 IBM_0838: EBCDIC Thai

IBM_0850

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	FS 001c	ESC 001b	DEL 007f	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	SUB 001a
8	Ç 00c7	ü 00fc	é 00e9	â 00e2	ä 00e4	à 00e0	å 00e5	ç 00e7	ê 00ea	ë 00eb	è 00e8	ï 00ef	î 00ee	ì 00ec	Ä 00c4	Å 00c5
9	É 00c9	æ 00e6	Æ 00c6	ô 00f4	ö 00f6	ò 00f2	û 00fb	ù 00f9	ÿ 00ff	Ö 00d6	Ü 00dc	ø 00f8	£ 00a3	Ø 00d8	× 00d7	ƒ 0192
a	á 00e1	í 00ed	ó 00f3	ú 00fa	ñ 00f1	Ñ 00d1	ª 00aa	º 00ba	¿ 00bf	® 00ae	¬ 00ac	½ 00bd	¼ 00bc	¡ 00a1	« 00ab	» 00bb
b	░ 2591	▒ 2592	▓ 2593	│ 2502	┤ 2524	Á 00c1	Â 00c2	À 00c0	© 00a9	╣ 2563	║ 2551	╗ 2557	╝ 255d	¢ 00a2	¥ 00a5	┐ 2510
c	└ 2514	┴ 2534	┬ 252c	├ 251c	─ 2500	┼ 253c	ã 00e3	Ã 00c3	╚ 255a	╔ 2554	╩ 2569	╦ 2566	╠ 2560	═ 2550	╬ 256c	¤ 00a4
d	ð 00f0	Ð 00d0	Ê 00ca	Ë 00cb	È 00c8	ı 0131	Í 00cd	Î 00ce	Ï 00cf	┘ 2518	┌ 250c	█ 2588	▄ 2584	¦ 00a6	Ì 00cc	▀ 2580
e	Ó 00d3	ß 00df	Ô 00d4	Ò 00d2	õ 00f5	Õ 00d5	µ 00b5	þ 00fe	Þ 00de	Ú 00da	Û 00db	Ù 00d9	ý 00fd	Ý 00dd	¯ 00af	´ 00b4
f	- 00ad	± 00b1	‗ 2017	¾ 00be	¶ 00b6	§ 00a7	÷ 00f7	¸ 00b8	° 00b0	¨ 00a8	· 00b7	¹ 00b9	³ 00b3	² 00b2	■ 25a0	SP 00a0

Fig. 125 IBM_0850: ASCII PC-Data-190: Latin Alphabet Number 1

IBM_0851

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	FS 001c	ESC 001b	DEL 007f	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	SUB 001a
8	Ç 00c7	ü 00fc	é 00e9	â 00e2	ä 00e4	à 00e0	Ά 0386	ç 00e7	ê 00ea	ë 00eb	è 00e8	ï 00ef	î 00ee	Έ 0388	Ä 00c4	Ή 0389
9	Ί 038a	SUB 001a	Ό 038c	ô 00f4	ö 00f6	Ύ 038e	û 00fb	ù 00f9	Ώ 038f	Ö 00d6	Ü 00dc	ά 03ac	£ 00a3	έ 03ad	ή 03ae	ί 03af
a	ϊ 03ca	ΐ 0390	ό 03cc	ύ 03cd	Α 0391	Β 0392	Γ 0393	Δ 0394	Ε 0395	Ζ 0396	Η 0397	½ 00bd	Θ 0398	Ι 0399	« 00ab	» 00bb
b	░ 2591	▒ 2592	▓ 2593	│ 2502	┤ 2524	Κ 039a	Λ 039b	Μ 039c	Ν 039d	╣ 2563	║ 2551	╗ 2557	╝ 255d	Ξ 039e	Ο 039f	┐ 2510
c	└ 2514	┴ 2534	┬ 252c	├ 251c	─ 2500	┼ 253c	Π 03a0	Ρ 03a1	╚ 255a	╔ 2554	╩ 2569	╦ 2566	╠ 2560	═ 2550	╬ 256c	Σ 03a3
d	Τ 03a4	Υ 03a5	Φ 03a6	Χ 03a7	Ψ 03a8	Ω 03a9	α 03b1	β 03b2	γ 03b3	┘ 2518	┌ 250c	█ 2588	▄ 2584	δ 03b4	ε 03b5	▀ 2580
e	ζ 03b6	η 03b7	θ 03b8	ι 03b9	κ 03ba	λ 03bb	μ 03bc	ν 03bd	ξ 03be	ο 03bf	π 03c0	ρ 03c1	σ 03c3	ς 03c2	τ 03c4	´ 00b4
f	- 00ad	± 00b1	υ 03c5	φ 03c6	χ 03c7	§ 00a7	ψ 03c8	¸ 00b8	° 00b0	¨ 00a8	ω 03c9	ϋ 03cb	ΰ 03b0	ώ 03ce	■ 25a0	SP 00a0

Fig. 126 IBM_0851: ASCII PC-Data Greek

IBM_0852

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	FS 001c	ESC 001b	DEL 007f	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	SUB 001a
8	Ç 00c7	ü 00fc	é 00e9	â 00e2	ä 00e4	ů 016f	ć 0107	ç 00e7	ł 0142	ë 00eb	Ő 0150	ő 0151	î 00ee	Ź 0179	Ä 00c4	Ć 0106
9	É 00c9	Ĺ 0139	ĺ 013a	ô 00f4	ö 00f6	Ľ 013d	ľ 013e	Ś 015a	ś 015b	Ö 00d6	Ü 00dc	Ť 0164	ť 0165	Ł 0141	× 00d7	č 010d
a	á 00e1	í 00ed	ó 00f3	ú 00fa	Ą 0104	ą 0105	Ž 017d	ž 017e	Ę 0118	ę 0119	SUB 001a	ź 017a	Č 010c	ş 015f	« 00ab	» 00bb
b	░ 2591	▒ 2592	▓ 2593	│ 2502	┤ 2524	Á 00c1	Â 00c2	Ě 011a	Ş 015e	╣ 2563	║ 2551	╗ 2557	╝ 255d	Ż 017b	ż 017c	┐ 2510
c	└ 2514	┴ 2534	┬ 252c	├ 251c	─ 2500	┼ 253c	Ă 0102	ă 0103	╚ 255a	╔ 2554	╩ 2569	╦ 2566	╠ 2560	═ 2550	╬ 256c	¤ 00a4
d	đ 0111	Đ 0110	Ď 010e	Ë 00cb	ď 010f	Ň 0147	Í 00cd	Î 00ce	ě 011b	┘ 2518	┌ 250c	█ 2588	▄ 2584	Ţ 0162	Ů 016e	▀ 2580
e	Ó 00d3	ß 00df	Ô 00d4	Ń 0143	ń 0144	ň 0148	Š 0160	š 0161	Ŕ 0154	Ú 00da	ŕ 0155	Ű 0170	ý 00fd	Ý 00dd	ţ 0163	´ 00b4
f	- 00ad	˝ 02dd	˛ 02db	ˇ 02c7	˘ 02d8	§ 00a7	÷ 00f7	¸ 00b8	° 00b0	¨ 00a8	˙ 02d9	ű 0171	Ř 0158	ř 0159	■ 25a0	SP 00a0

Fig. 127 IBM_0852: ASCII PC-Data Latin-2 Multilingual

IBM_0855

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	FS 001c	ESC 001b	DEL 007f	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	SUB 001a
8	ђ 0452	Ђ 0402	ѓ 0453	Ѓ 0403	ё 0451	Ё 0401	є 0454	Є 0404	ѕ 0455	Ѕ 0405	і 0456	І 0406	ї 0457	Ї 0407	ј 0458	Ј 0408
9	љ 0459	Љ 0409	њ 045a	Њ 040a	ћ 045b	Ћ 040b	ќ 045c	Ќ 040c	ў 045e	Ў 040e	џ 045f	Џ 040f	ю 044e	Ю 042e	ъ 044a	Ъ 042a
a	а 0430	А 0410	б 0431	Б 0411	ц 0446	Ц 0426	д 0434	Д 0414	е 0435	Е 0415	ф 0444	Ф 0424	г 0433	Г 0413	« 00ab	» 00bb
b	░ 2591	▒ 2592	▓ 2593	│ 2502	┤ 2524	х 0445	Х 0425	и 0438	И 0418	╣ 2563	║ 2551	╗ 2557	╝ 255d	й 0439	Й 0419	┐ 2510
c	└ 2514	┴ 2534	┬ 252c	├ 251c	─ 2500	┼ 253c	к 043a	К 041a	╚ 255a	╔ 2554	╩ 2569	╦ 2566	╠ 2560	═ 2550	╬ 256c	¤ 00a4
d	л 043b	Л 041b	м 043c	М 041c	н 043d	Н 041d	о 043e	О 041e	п 043f	┘ 2518	┌ 250c	█ 2588	▄ 2584	П 041f	я 044f	▀ 2580
e	Я 042f	р 0440	Р 0420	с 0441	С 0421	т 0442	Т 0422	у 0443	У 0423	ж 0436	Ж 0416	в 0432	В 0412	ь 044c	Ь 042c	№ 2116
f	- 00ad	ы 044b	Ы 042b	з 0437	З 0417	ш 0448	Ш 0428	э 044d	Э 042d	щ 0449	Щ 0429	ч 0447	Ч 0427	§ 00a7	■ 25a0	SP 00a0

Fig. 128 IBM_0855: ASCII PC-Data Cyrillic

IBM_0856

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	FS 001c	ESC 001b	DEL 007f	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	SUB 001a
8	א 05d0	ב 05d1	ג 05d2	ד 05d3	ה 05d4	ו 05d5	ז 05d6	ח 05d7	ט 05d8	י 05d9	ך 05da	כ 05db	ל 05dc	ם 05dd	מ 05de	ן 05df
9	נ 05e0	ס 05e1	ע 05e2	ף 05e3	פ 05e4	ץ 05e5	צ 05e6	ק 05e7	ר 05e8	ש 05e9	ת 05ea	SUB 001a	£ 00a3	SUB 001a	× 00d7	SUB 001a
a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	® 00ae	¬ 00ac	½ 00bd	¼ 00bc	SUB 001a	« 00ab	» 00bb
b	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	© 00a9	SUB 001a	SUB 001a	SUB 001a	SUB 001a	¢ 00a2	¥ 00a5	SUB 001a
c	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	¤ 00a4
d	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	¦ 00a6	SUB 001a	SUB 001a
e	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	µ 00b5	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	‾ 203e	´ 00b4
f	- 00ad	± 00b1	‗ 2017	¾ 00be	¶ 00b6	§ 00a7	÷ 00f7	¸ 00b8	° 00b0	¨ 00a8	• 2022	¹ 00b9	³ 00b3	² 00b2	SUB 001a	SP 00a0

Fig. 129 IBM_0856: ASCII PC-Data Hebrew

IBM_0857

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	FS 001c	ESC 001b	DEL 007f	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	SUB 001a
8	Ç 00c7	ü 00fc	é 00e9	â 00e2	ä 00e4	à 00e0	å 00e5	ç 00e7	ê 00ea	ë 00eb	è 00e8	ï 00ef	î 00ee	ı 0131	Ä 00c4	Å 00c5
9	É 00c9	æ 00e6	Æ 00c6	ô 00f4	ö 00f6	ò 00f2	û 00fb	ù 00f9	İ 0130	Ö 00d6	Ü 00dc	ø 00f8	£ 00a3	Ø 00d8	Ş 015e	ş 015f
a	á 00e1	í 00ed	ó 00f3	ú 00fa	ñ 00f1	Ñ 00d1	Ğ 011e	ğ 011f	¿ 00bf	® 00ae	¬ 00ac	½ 00bd	¼ 00bc	¡ 00a1	« 00ab	» 00bb
b	░ 2591	▒ 2592	▓ 2593	│ 2502	┤ 2524	Á 00c1	Â 00c2	À 00c0	© 00a9	╣ 2563	║ 2551	╗ 2557	╝ 255d	¢ 00a2	¥ 00a5	┐ 2510
c	└ 2514	┴ 2534	┬ 252c	├ 251c	─ 2500	┼ 253c	ã 00e3	Ã 00c3	╚ 255a	╔ 2554	╩ 2569	╦ 2566	╠ 2560	═ 2550	╬ 256c	¤ 00a4
d	º 00ba	ª 00aa	Ê 00ca	Ë 00cb	È 00c8	SUB 001a	Í 00cd	Î 00ce	Ï 00cf	┘ 2518	┌ 250c	█ 2588	▄ 2584	¦ 00a6	Ì 00cc	▀ 2580
e	Ó 00d3	ß 00df	Ô 00d4	Ò 00d2	õ 00f5	Õ 00d5	µ 00b5	SUB 001a	× 00d7	Ú 00da	Û 00db	Ù 00d9	ì 00ec	ÿ 00ff	¯ 00af	´ 00b4
f	- 00ad	± 00b1	SUB 001a	¾ 00be	¶ 00b6	§ 00a7	÷ 00f7	¸ 00b8	° 00b0	¨ 00a8	· 00b7	¹ 00b9	³ 00b3	² 00b2	■ 25a0	SP 00a0

Fig. 130 IBM_0857: ASCII PC-Data Turkey Latin-5

IBM_0860

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	FS 001c	ESC 001b	DEL 007f	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	SUB 001a
8	Ç 00c7	ü 00fc	é 00e9	â 00e2	ã 00e3	à 00e0	Á 00c1	ç 00e7	ê 00ea	Ê 00ca	è 00e8	Í 00cd	Ô 00d4	ì 00ec	Ã 00c3	Â 00c2
9	É 00c9	À 00c0	È 00c8	ô 00f4	õ 00f5	ò 00f2	Ú 00da	ù 00f9	Ì 00cc	Õ 00d5	Ü 00dc	¢ 00a2	£ 00a3	Ù 00d9	₧ 20a7	Ó 00d3
a	á 00e1	í 00ed	ó 00f3	ú 00fa	ñ 00f1	Ñ 00d1	ª 00aa	º 00ba	¿ 00bf	Ò 00d2	¬ 00ac	½ 00bd	¼ 00bc	¡ 00a1	« 00ab	» 00bb
b	░ 2591	▒ 2592	▓ 2593	│ 2502	┤ 2524	╡ 2561	╢ 2562	╖ 2556	╕ 2555	╣ 2563	║ 2551	╗ 2557	╝ 255d	╜ 255c	╛ 255b	┐ 2510
c	└ 2514	┴ 2534	┬ 252c	├ 251c	─ 2500	┼ 253c	╞ 255e	╟ 255f	╚ 255a	╔ 2554	╩ 2569	╦ 2566	╠ 2560	═ 2550	╬ 256c	╧ 2567
d	╨ 2568	╤ 2564	╥ 2565	╙ 2559	╘ 2558	╒ 2552	╓ 2553	╫ 256b	╪ 256a	┘ 2518	┌ 250c	█ 2588	▄ 2584	▌ 258c	▐ 2590	▀ 2580
e	α 03b1	ß 00df	Γ 0393	π 03c0	Σ 03a3	σ 03c3	μ 03bc	τ 03c4	Φ 03a6	Θ 0398	Ω 03a9	δ 03b4	∞ 221e	φ 03c6	ε 03b5	∩ 2229
f	≡ 2261	± 00b1	≥ 2265	≤ 2264	⌠ 2320	⌡ 2321	÷ 00f7	≈ 2248	° 00b0	∙ 2219	· 00b7	√ 221a	ⁿ 207f	² 00b2	■ 25a0	SP 00a0

Fig. 131 IBM_0860: ASCII PC-Data Portugal

IBM_0861

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	FS 001c	ESC 001b	DEL 007f	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	SUB 001a
8	Ç 00c7	ü 00fc	é 00e9	â 00e2	ä 00e4	à 00e0	å 00e5	ç 00e7	ê 00ea	ë 00eb	è 00e8	Ð 00d0	ð 00f0	Þ 00de	Ä 00c4	Å 00c5
9	É 00c9	æ 00e6	Æ 00c6	ô 00f4	ö 00f6	þ 00fe	û 00fb	Ý 00dd	ý 00fd	Ö 00d6	Ü 00dc	ø 00f8	£ 00a3	Ø 00d8	₧ 20a7	ƒ 0192
a	á 00e1	í 00ed	ó 00f3	ú 00fa	Á 00c1	Í 00cd	Ó 00d3	Ú 00da	¿ 00bf	⌐ 2310	¬ 00ac	½ 00bd	¼ 00bc	¡ 00a1	« 00ab	» 00bb
b	░ 2591	▒ 2592	▓ 2593	│ 2502	┤ 2524	╡ 2561	╢ 2562	╖ 2556	╕ 2555	╣ 2563	║ 2551	╗ 2557	╝ 255d	╜ 255c	╛ 255b	┐ 2510
c	└ 2514	┴ 2534	┬ 252c	├ 251c	─ 2500	┼ 253c	╞ 255e	╟ 255f	╚ 255a	╔ 2554	╩ 2569	╦ 2566	╠ 2560	═ 2550	╬ 256c	╧ 2567
d	╨ 2568	╤ 2564	╥ 2565	╙ 2559	╘ 2558	╒ 2552	╓ 2553	╫ 256b	╪ 256a	┘ 2518	┌ 250c	█ 2588	▄ 2584	▌ 258c	▐ 2590	▀ 2580
e	α 03b1	ß 00df	Γ 0393	π 03c0	Σ 03a3	σ 03c3	µ 03bc	τ 03c4	Φ 03a6	Θ 0398	Ω 03a9	δ 03b4	∞ 221e	φ 03c6	ε 03b5	∩ 2229
f	≡ 2261	± 00b1	≥ 2265	≤ 2264	⌠ 2320	⌡ 2321	÷ 00f7	≈ 2248	° 00b0	∙ 2219	· 00b7	√ 221a	ⁿ 207f	² 00b2	■ 25a0	SP 00a0

Fig. 132 IBM_0861: ASCII PC-Data Iceland

IBM_0862

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	FS 001c	ESC 001b	DEL 007f	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	SUB 001a
8	א 05d0	ב 05d1	ג 05d2	ד 05d3	ה 05d4	ו 05d5	ז 05d6	ח 05d7	ט 05d8	י 05d9	ך 05da	כ 05db	ל 05dc	ם 05dd	מ 05de	ן 05df
9	נ 05e0	ס 05e1	ע 05e2	ף 05e3	פ 05e4	ץ 05e5	צ 05e6	ק 05e7	ר 05e8	ש 05e9	ת 05ea	¢ 00a2	£ 00a3	¥ 00a5	₧ 20a7	ƒ 0192
a	á 00e1	í 00ed	ó 00f3	ú 00fa	ñ 00f1	Ñ 00d1	ª 00aa	º 00ba	¿ 00bf	⌐ 2310	¬ 00ac	½ 00bd	¼ 00bc	¡ 00a1	« 00ab	» 00bb
b	░ 2591	▒ 2592	▓ 2593	│ 2502	┤ 2524	╡ 2561	╢ 2562	╖ 2556	╕ 2555	╣ 2563	║ 2551	╗ 2557	╝ 255d	╜ 255c	╛ 255b	┐ 2510
c	└ 2514	┴ 2534	┬ 252c	├ 251c	─ 2500	┼ 253c	╞ 255e	╟ 255f	╚ 255a	╔ 2554	╩ 2569	╦ 2566	╠ 2560	═ 2550	╬ 256c	╧ 2567
d	╨ 2568	╤ 2564	╥ 2565	╙ 2559	╘ 2558	╒ 2552	╓ 2553	╫ 256b	╪ 256a	┘ 2518	┌ 250c	█ 2588	▄ 2584	▌ 258c	▐ 2590	▀ 2580
e	α 03b1	ß 00df	Γ 0393	π 03c0	Σ 03a3	σ 03c3	µ 03bc	τ 03c4	Φ 03a6	Θ 0398	Ω 03a9	δ 03b4	∞ 221e	φ 03c6	ε 03b5	∩ 2229
f	≡ 2261	± 00b1	≥ 2265	≤ 2264	⌠ 2320	⌡ 2321	÷ 00f7	≈ 2248	° 00b0	∙ 2219	· 00b7	√ 221a	^{n} 207f	^{2} 00b2	■ 25a0	SP 00a0

Fig. 133 IBM_0862: ASCII PC-Data Hebrew (Migration)

IBM_0863

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	FS 001c	ESC 001b	DEL 007f	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	SUB 001a
8	Ç 00c7	ü 00fc	é 00e9	â 00e2	Â 00c2	à 00e0	¶ 00b6	ç 00e7	ê 00ea	ë 00eb	è 00e8	ï 00ef	î 00ee	‗ 2017	À 00c0	§ 00a7
9	É 00c9	È 00c8	Ê 00ca	ô 00f4	Ë 00cb	Ï 00cf	û 00fb	ù 00f9	¤ 00a4	Ô 00d4	Ü 00dc	¢ 00a2	£ 00a3	Ù 00d9	Û 00db	ƒ 0192
a	¦ 00a6	´ 00b4	ó 00f3	ú 00fa	¨ 00a8	¸ 00b8	³ 00b3	¯ 00af	Î 00ce	⌐ 2310	¬ 00ac	½ 00bd	¼ 00bc	¾ 00be	« 00ab	» 00bb
b	░ 2591	▒ 2592	▓ 2593	│ 2502	┤ 2524	╡ 2561	╢ 2562	╖ 2556	╕ 2555	╣ 2563	║ 2551	╗ 2557	╝ 255d	╜ 255c	╛ 255b	┐ 2510
c	└ 2514	┴ 2534	┬ 252c	├ 251c	─ 2500	┼ 253c	╞ 255e	╟ 255f	╚ 255a	╔ 2554	╩ 2569	╦ 2566	╠ 2560	═ 2550	╬ 256c	╧ 2567
d	╨ 2568	╤ 2564	╥ 2565	╙ 2559	╘ 2558	╒ 2552	╓ 2553	╫ 256b	╪ 256a	┘ 2518	┌ 250c	█ 2588	▄ 2584	▌ 258c	▐ 2590	▀ 2580
e	α 03b1	ß 00df	Γ 0393	π 03c0	Σ 03a3	σ 03c3	µ 03bc	τ 03c4	Φ 03a6	Θ 0398	Ω 03a9	δ 03b4	∞ 221e	φ 03c6	ε 03b5	∩ 2229
f	≡ 2261	± 00b1	≥ 2265	≤ 2264	⌠ 2320	⌡ 2321	÷ 00f7	≈ 2248	° 00b0	∙ 2219	· 00b7	√ 221a	ⁿ 207f	² 00b2	■ 25a0	SP 00a0

Fig. 134 IBM_0863: ASCII PC-Data Canada

IBM_0864

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	FS 001c	ESC 001b	DEL 007f	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	SUB 001a
8	° 00b0	· 00b7	• 2219	√ 221a	▓ 2592	─ 2500	│ 2502	┼ 253c	┤ 2524	┬ 252c	├ 251c	┴ 2534	┐ 2510	┌ 250c	└ 2514	┘ 2518
9	β 03b2	∞ 221e	φ 03c6	± 00b1	½ 00bd	¼ 00bc	≈ 2248	« 00ab	» 00bb	ﻷ fef7	ﻸ fef8	SUB 001a	SUB 001a	ﻻ fefb	ﻼ fefc	· f8fc
a	SP 00a0	- 00ad	ﺂ fe82	£ 00a3	¤ 00a4	ﺄ fe84	SUB 001a	SUB 001a	ﺎ fe8e	ﺏ fe8f	ﺕ fe95	ﺙ fe99	، 060c	ﺝ fe9d	ﺡ fea1	ﺥ fea5
b	٠ 0660	١ 0661	٢ 0662	٣ 0663	٤ 0664	٥ 0665	٦ 0666	٧ 0667	٨ 0668	٩ 0669	ﻑ fed1	؛ 061b	· f8f6	· f8f5	· f8f4	؟ 061f
c	¢ 00a2	ﺀ fe80	ﺁ fe81	ﺃ fe83	ﺅ fe85	ﻊ feca	ﺋ fe8b	ﺍ fe8d	ﺑ fe91	ﺓ fe93	ﺗ fe97	ﺛ fe9b	ﺟ fe9f	ﺣ fea3	ﺧ fea7	ﺩ fea9
d	ﺫ feab	ﺭ fead	ﺯ feaf	ﺳ feb3	ﺷ feb7	ﺻ febb	ﺿ febf	ﻃ fec3	ﻇ fec7	ﻋ fecb	ﻏ fecf	¦ 00a6	¬ 00ac	÷ 00f7	× 00d7	ﻉ fec9
e	ـ 0640	ﻓ fed3	ﻗ fed7	ﻛ fedb	ﻟ fedf	ﻣ fee3	ﻧ fee7	ﻫ feeb	ﻭ feed	ﻯ feef	ﻳ fef3	· f8f7	ﻌ fecc	ﻎ fece	ﻍ fecd	ﻡ fee1
f	ﹽ fe7d	ﹼ fe7c	ﻥ fee5	ﻩ fee9	ﻬ feec	ﻰ fef0	ﻲ fef2	ﻐ fed0	ﻕ fed5	ﻵ fef5	ﻶ fef6	ﻝ fedd	ﻙ fed9	ﻱ fef1	■ 25a0	SUB 001a

Fig. 135 IBM_0864: ASCII PC-Data Arabic

IBM_0865

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	FS 001c	ESC 001b	DEL 007f	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	SUB 001a
8	Ç 00c7	ü 00fc	é 00e9	â 00e2	ä 00e4	à 00e0	å 00e5	ç 00e7	ê 00ea	ë 00eb	è 00e8	ï 00ef	î 00ee	ì 00ec	Ä 00c4	Å 00c5
9	É 00c9	æ 00e6	Æ 00c6	ô 00f4	ö 00f6	ò 00f2	û 00fb	ù 00f9	ÿ 00ff	Ö 00d6	Ü 00dc	ø 00f8	£ 00a3	Ø 00d8	₧ 20a7	ƒ 0192
a	á 00e1	í 00ed	ó 00f3	ú 00fa	ñ 00f1	Ñ 00d1	ª 00aa	º 00ba	¿ 00bf	⌐ 2310	¬ 00ac	½ 00bd	¼ 00bc	¡ 00a1	« 00ab	¤ 00a4
b	░ 2591	▒ 2592	▓ 2593	│ 2502	┤ 2524	╡ 2561	╢ 2562	╖ 2556	╕ 2555	╣ 2563	║ 2551	╗ 2557	╝ 255d	╜ 255c	╛ 255b	┐ 2510
c	└ 2514	┴ 2534	┬ 252c	├ 251c	─ 2500	┼ 253c	╞ 255e	╟ 255f	╚ 255a	╔ 2554	╩ 2569	╦ 2566	╠ 2560	═ 2550	╬ 256c	╧ 2567
d	╨ 2568	╤ 2564	╥ 2565	╙ 2559	╘ 2558	╒ 2552	╓ 2553	╫ 256b	╪ 256a	┘ 2518	┌ 250c	█ 2588	▄ 2584	▌ 258c	▐ 2590	▀ 2580
e	α 03b1	ß 00df	Γ 0393	π 03c0	Σ 03a3	σ 03c3	µ 03bc	τ 03c4	Φ 03a6	Θ 0398	Ω 03a9	δ 03b4	∞ 221e	φ 03c6	ε 03b5	∩ 2229
f	≡ 2261	± 00b1	≥ 2265	≤ 2264	⌠ 2320	⌡ 2321	÷ 00f7	≈ 2248	° 00b0	∙ 2219	· 00b7	√ 221a	^{n} 207f	^{2} 00b2	■ 25a0	SP 00a0

Fig. 136 IBM_0865: ASCII PC-Data Denmark, Norway

IBM_0866

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	FS 001c	ESC 001b	DEL 007f	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	SUB 001a
8	А 0410	Б 0411	В 0412	Г 0413	Д 0414	Е 0415	Ж 0416	З 0417	И 0418	Й 0419	К 041a	Л 041b	М 041c	Н 041d	О 041e	П 041f
9	Р 0420	С 0421	Т 0422	У 0423	Ф 0424	Х 0425	Ц 0426	Ч 0427	Ш 0428	Щ 0429	Ъ 042a	Ы 042b	Ь 042c	Э 042d	Ю 042e	Я 042f
a	а 0430	б 0431	в 0432	г 0433	д 0434	е 0435	ж 0436	з 0437	и 0438	й 0439	к 043a	л 043b	м 043c	н 043d	о 043e	п 043f
b	░ 2591	▒ 2592	▓ 2593	│ 2502	┤ 2524	╡ 2561	╢ 2562	╖ 2556	╕ 2555	╣ 2563	║ 2551	╗ 2557	╝ 255d	╜ 255c	╛ 255b	┐ 2510
c	└ 2514	┴ 2534	┬ 252c	├ 251c	─ 2500	┼ 253c	╞ 255e	╟ 255f	╚ 255a	╔ 2554	╩ 2569	╦ 2566	╠ 2560	═ 2550	╬ 256c	╧ 2567
d	╨ 2568	╤ 2564	╥ 2565	╙ 2559	╘ 2558	╒ 2552	╓ 2553	╫ 256b	╪ 256a	┘ 2518	┌ 250c	█ 2588	▄ 2584	▌ 258c	▐ 2590	▀ 2580
e	р 0440	с 0441	т 0442	у 0443	ф 0444	х 0445	ц 0446	ч 0447	ш 0448	щ 0449	ъ 044a	ы 044b	ь 044c	э 044d	ю 044e	я 044f
f	Ё 0401	ё 0451	Є 0404	є 0454	Ї 0407	ї 0457	Ў 040e	ў 045e	° 00b0	∙ 2219	· 00b7	√ 221a	№ 2116	¤ 00a4	■ 25a0	SP 00a0

Fig. 137 IBM_0866: ASCII PC-Data Cyrillic, Russian

IBM_0869

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	FS 001c	ESC 001b	DEL 007f	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	SUB 001a
8	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	Ά 0386	SUB 001a	· 0387	¬ 00ac	¦ 00a6	‘ 2018	’ 2019	Έ 0388	― 2015	Ή 0389
9	Ί 038a	Ϊ 03aa	Ό 038c	SUB 001a	SUB 001a	Ύ 038e	Ϋ 03ab	© 00a9	Ώ 038f	² 00b2	³ 00b3	ά 03ac	£ 00a3	έ 03ad	ή 03ae	ί 03af
a	ϊ 03ca	ΐ 0390	ό 03cc	ύ 03cd	Α 0391	Β 0392	Γ 0393	Δ 0394	Ε 0395	Ζ 0396	Η 0397	½ 00bd	Θ 0398	Ι 0399	« 00ab	» 00bb
b	░ 2591	▒ 2592	▓ 2593	│ 2502	┤ 2524	Κ 039a	Λ 039b	Μ 039c	Ν 039d	╣ 2563	║ 2551	╗ 2557	╝ 255d	Ξ 039e	Ο 039f	┐ 2510
c	└ 2514	┴ 2534	┬ 252c	├ 251c	─ 2500	┼ 253c	Π 03a0	Ρ 03a1	╚ 255a	╔ 2554	╩ 2569	╦ 2566	╠ 2560	═ 2550	╬ 256c	Σ 03a3
d	Τ 03a4	Υ 03a5	Φ 03a6	Χ 03a7	Ψ 03a8	Ω 03a9	α 03b1	β 03b2	γ 03b3	┘ 2518	┌ 250c	█ 2588	▄ 2584	δ 03b4	ε 03b5	▀ 2580
e	ζ 03b6	η 03b7	θ 03b8	ι 03b9	κ 03ba	λ 03bb	μ 03bc	ν 03bd	ξ 03be	ο 03bf	π 03c0	ρ 03c1	σ 03c3	ς 03c2	τ 03c4	΄ 00b4
f	- 00ad	± 00b1	υ 03c5	φ 03c6	χ 03c7	§ 00a7	ψ 03c8	΅ 0385	° 00b0	¨ 00a8	ω 03c9	ϋ 03cb	ΰ 03b0	ώ 03ce	■ 25a0	SP 00a0

Fig. 138 IBM_0869: ASCII PC-Data Greek

IBM_0870

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	ţ 0163	á 00e1	ă 0103	č 010d	ç 00e7	ć 0107	[005b	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	é 00e9	ę 0119	ë 00eb	ů 016f	í 00ed	î 00ee	ľ 013e	ĺ 013a	ß 00df	] 005d	$ 0024	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Â 00c2	Ä 00c4	˝ 02dd	Á 00c1	Ă 0102	Č 010c	Ç 00c7	Ć 0106	| 007c	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ˇ 02c7	É 00c9	Ę 0118	Ë 00cb	Ů 016e	Í 00cd	Î 00ce	Ľ 013d	Ĺ 0139	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	˘ 02d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	ś 015b	ň 0148	đ 0111	ý 00fd	ř 0159	ş 015f
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ł 0142	ń 0144	š 0161	¸ 00b8	˛ 02db	¤ 00a4
a	ą 0105	~ 007e	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	Ś 015a	Ň 0147	Đ 0110	Ý 00dd	Ř 0158	Ş 015e
b	˙ 02d9	Ą 0104	ż 017c	Ţ 0162	Ż 017b	§ 00a7	ž 017e	ź 017a	Ž 017d	Ź 0179	Ł 0141	Ń 0143	Š 0160	¨ 00a8	´ 00b4	× 00d7
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	ö 00f6	ŕ 0155	ó 00f3	ő 0151
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	Ě 011a	ű 0171	ü 00fc	ť 0165	ú 00fa	ě 011b
e	\ 005c	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	ď 010f	Ô 00d4	Ö 00d6	Ŕ 0154	Ó 00d3	Ő 0150
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	Ď 010e	Ű 0170	Ü 00dc	Ť 0164	Ú 00da	· 009f

Fig. 139 IBM_0870: EBCDIC Latin-2 Multilingual

IBM_0871

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	à 00e0	á 00e1	ã 00e3	å 00e5	ç 00e7	ñ 00f1	Þ 00de	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	é 00e9	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	Æ 00c6	$ 0024	* 002a	) 0029	; 003b	Ö 00d6
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	Ã 00c3	Å 00c5	Ç 00c7	Ñ 00d1	¦ 00a6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	ð 00f0	: 003a	# 0023	Đ 00d0	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	` 0060	ý 00fd	{ 007b	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	} 007d	¸ 00b8	] 005d	¤ 00a4
a	µ 00b5	ö 00f6	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	@ 0040	Ý 00dd	[005b	® 00ae
b	¢ 00a2	£ 00a3	¥ 00a5	· 00b7	© 00a9	§ 00a7	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	¬ 00ac	| 007c	¯ 00af	¨ 00a8	\ 005c	× 00d7
c	þ 00fe	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	~ 007e	ò 00f2	ó 00f3	õ 00f5
d	æ 00e6	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	ü 00fc	ù 00f9	ú 00fa	ÿ 00ff
e	´ 00b4	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	^ 005e	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 140 IBM_0871: EBCDIC Iceland

IBM_0874

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	FS 001c	ESC 001b	DEL 007f	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	SUB 001a
8	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
9	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
a	่ 0e48	ก 0e01	ข 0e02	ฃ 0e03	ค 0e04	ฅ 0e05	ฆ 0e06	ง 0e07	จ 0e08	ฉ 0e09	ช 0e0a	ซ 0e0b	ฌ 0e0c	ญ 0e0d	ฎ 0e0e	ฏ 0e0f
b	ฐ 0e10	ฑ 0e11	ฒ 0e12	ณ 0e13	ด 0e14	ต 0e15	ถ 0e16	ท 0e17	ธ 0e18	น 0e19	บ 0e1a	ป 0e1b	ผ 0e1c	ฝ 0e1d	พ 0e1e	ฟ 0e1f
c	ภ 0e20	ม 0e21	ย 0e22	ร 0e23	ฤ 0e24	ล 0e25	ฦ 0e26	ว 0e27	ศ 0e28	ษ 0e29	ส 0e2a	ห 0e2b	ฬ 0e2c	อ 0e2d	ฮ 0e2e	ฯ 0e2f
d	ะ 0e30	ั 0e31	า 0e32	ำ 0e33	ิ 0e34	ี 0e35	ึ 0e36	ื 0e37	ุ 0e38	ู 0e39	ฺ 0e3a	้ 0e49	๊ 0e4a	๋ 0e4b	์ 0e4c	฿ 0e3f
e	เ 0e40	แ 0e41	โ 0e42	ใ 0e43	ไ 0e44	ๅ 0e45	ๆ 0e46	็ 0e47	่ 0e48	้ 0e49	๊ 0e4a	๋ 0e4b	์ 0e4c	ํ 0e4d	๎ 0e4e	๏ 0e4f
f	๐ 0e50	๑ 0e51	๒ 0e52	๓ 0e53	๔ 0e54	๕ 0e55	๖ 0e56	๗ 0e57	๘ 0e58	๙ 0e59	๚ 0e5a	๛ 0e5b	¢ 00a2	¬ 00ac	¦ 00a6	SP 00a0

Fig. 141 IBM_0874: ASCII PC-Data Thai

IBM_0875

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	Α 0391	Β 0392	Γ 0393	Δ 0394	Ε 0395	Ζ 0396	Η 0397	Θ 0398	Ι 0399	[005b	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	Κ 039a	Λ 039b	Μ 039c	Ν 039d	Ξ 039e	Ο 039f	Π 03a0	Ρ 03a1	Σ 03a3	] 005d	$ 0024	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Τ 03a4	Υ 03a5	Φ 03a6	Χ 03a7	Ψ 03a8	Ω 03a9	Ϊ 03aa	Ϋ 03ab	| 007c	, 002c	% 0025	_ 005f	> 003e	? 003f
7	¨ 00a8	Ά 0386	Έ 0388	Ή 0389	SP 00a0	Ί 038a	Ό 038c	Ύ 038e	Ώ 038f	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	΅ 0385	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	α 03b1	β 03b2	γ 03b3	δ 03b4	ε 03b5	ζ 03b6
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	η 03b7	θ 03b8	ι 03b9	κ 03ba	λ 03bb	μ 03bc
a	´ 00b4	~ 007e	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	ν 03bd	ξ 03be	ο 03bf	π 03c0	ρ 03c1	σ 03c3
b	£ 00a3	ά 03ac	έ 03ad	ή 03ae	ϊ 03ca	ί 03af	ό 03cc	ύ 03cd	ϋ 03cb	ώ 03ce	ς 03c2	τ 03c4	υ 03c5	φ 03c6	χ 03c7	ψ 03c8
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ω 03c9	ΐ 0390	ΰ 03b0	‘ 2018	― 2015
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	± 00b1	½ 00bd	SUB 001a	· 0387	’ 2019	¦ 00a6
e	\ 005c	SUB 001a	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	§ 00a7	SUB 001a	SUB 001a	« 00ab	¬ 00ac
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	© 00a9	SUB 001a	SUB 001a	» 00bb	· 009f

Fig. 142 IBM_0875: EBCDIC Greek

IBM_0880

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	ђ 0452	ѓ 0453	ё 0451	є 0454	ѕ 0455	і 0456	ї 0457	ј 0458	[005b	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	љ 0459	њ 045a	ћ 045b	ќ 045c	ў 045e	џ 045f	Ъ 042a	№ 2116	Ђ 0402	] 005d	$ 0024	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Ѓ 0403	Ё 0401	Є 0404	Ѕ 0405	І 0406	Ї 0407	Ј 0408	Љ 0409	| 007c	, 002c	% 0025	_ 005f	> 003e	? 003f
7	Њ 040a	Ћ 040b	Ќ 040c	- 00ad	Ў 040e	Џ 040f	ю 044e	а 0430	б 0431	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	ц 0446	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	д 0434	е 0435	ф 0444	г 0433	х 0445	и 0438
9	й 0439	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	к 043a	л 043b	м 043c	н 043d	о 043e	п 043f
a	я 044f	~ 007e	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	р 0440	с 0441	т 0442	у 0443	ж 0436	в 0432
b	ь 044c	ы 044b	з 0437	ш 0448	э 044d	щ 0449	ч 0447	ъ 044a	Ю 042e	А 0410	Б 0411	Ц 0426	Д 0414	Е 0415	Ф 0424	Г 0413
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	Х 0425	И 0418	Й 0419	К 041a	Л 041b	М 041c
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	Н 041d	О 041e	П 041f	Я 042f	Р 0420	С 0421
e	\ 005c	¤ 00a4	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	Т 0422	У 0423	Ж 0416	В 0412	Ь 042c	Ы 042b
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	З 0417	Ш 0428	Э 042d	Щ 0429	Ч 0427	· 009f

Fig. 143 IBM_0880: EBCDIC Cyrillic Multilingual

	x0	x1	x2	X3	x4	x5	x6	x7	x8	x9	XA	xB	xC	xD	xE	xF
0x																
1x																
2x																
3x																
4x																
5x																
6x																
7x																
8x																
9x																
Ax																
Bx																
Cx																
Dx																
Ex																
Fx																

Fig. 144 IBM_0893: Optical Character Recognition OCR B

IBM_0905

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	à 00e0	á 00e1	SUB 001a	ċ 010b	{ 007b	ñ 00f1	Ç 00c7	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	é 00e9	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	Ğ 011e	İ 0130	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	SUB 001a	Ċ 010a	[005b	Ñ 00d1	ş 015f	, 002c	% 0025	_ 005f	> 003e	? 003f
7	SUB 001a	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	ı 0131	: 003a	Ö 00d6	Ş 015e	' 0027	= 003d	Ü 00dc
8	˘ 02d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	ħ 0127	ĉ 0109	ŝ 015d	ŭ 016d	SUB 001a	| 007c
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ĥ 0125	ĝ 011d	ĵ 0135	¸ 00b8	SUB 001a	¤ 00a4
a	µ 00b5	ö 00f6	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	Ħ 0126	Ĉ 0108	Ŝ 015c	Ŭ 016c	SUB 001a	@ 0040
b	˙ 02d9	£ 00a3	ż 017c	} 007d	Ż 017b	§ 00a7	] 005d	· 00b7	½ 00bd	$ 0024	Ĥ 0124	Ĝ 011c	Ĵ 0134	¨ 00a8	´ 00b4	× 00d7
c	ç 00e7	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	~ 007e	ò 00f2	ó 00f3	ġ 0121
d	ğ 011f	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	` 0060	û 00fb	\ 005c	ù 00f9	ú 00fa	SUB 001a
e	ü 00fc	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	# 0023	Ò 00d2	Ó 00d3	Ġ 0120
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	" 0022	Ù 00d9	Ú 00da	· 009f

Fig. 145 IBM_0905: EBCDIC Turkey Latin 3 Multilingual

IBM_0912

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	DEL 007f
8	. 0080	. 0081	. 0082	. 0083	. 0084	. 0085	. 0086	. 0087	. 0088	. 0089	. 008a	. 008b	. 008c	. 008d	. 008e	. 008f
9	. 0090	. 0091	. 0092	. 0093	. 0094	. 0095	. 0096	. 0097	. 0098	. 0099	. 009a	. 009b	. 009c	. 009d	. 009e	. 009f
a	SP 00a0	Ą 0104	˘ 02d8	Ł 0141	¤ 00a4	Ľ 013d	Ś 015a	§ 00a7	¨ 00a8	Š 0160	Ş 015e	Ť 0164	Ź 0179	- 00ad	Ž 017d	Ż 017b
b	° 00b0	ą 0105	˛ 02db	ł 0142	´ 00b4	ľ 013e	ś 015b	ˇ 02c7	¸ 00b8	š 0161	ş 015f	ť 0165	ź 017a	˝ 02dd	ž 017e	ż 017c
c	Ŕ 0154	Á 00c1	Â 00c2	Ă 0102	Ä 00c4	Ĺ 0139	Ć 0106	Ç 00c7	Č 010c	É 00c9	Ę 0118	Ë 00cb	Ě 011a	Í 00cd	Î 00ce	Ď 010e
d	Đ 0110	Ń 0143	Ň 0147	Ó 00d3	Ô 00d4	Ő 0150	Ö 00d6	× 00d7	Ř 0158	Ů 016e	Ú 00da	Ű 0170	Ü 00dc	Ý 00dd	Ţ 0162	ß 00df
e	ŕ 0155	á 00e1	â 00e2	ă 0103	ä 00e4	ĺ 013a	ć 0107	ç 00e7	č 010d	é 00e9	ę 0119	ë 00eb	ě 011b	í 00ed	î 00ee	ď 010f
f	đ 0111	ń 0144	ň 0148	ó 00f3	ô 00f4	ő 0151	ö 00f6	÷ 00f7	ř 0159	ů 016f	ú 00fa	ű 0171	ü 00fc	ý 00fd	ţ 0163	˙ 02d9

Fig. 146 IBM_0912: ASCII Latin 2 ISO 8859-2

IBM_0915

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	DEL 007f
8	· 0080	· 0081	· 0082	· 0083	· 0084	· 0085	· 0086	· 0087	· 0088	· 0089	· 008a	· 008b	· 008c	· 008d	· 008e	· 008f
9	· 0090	· 0091	· 0092	· 0093	· 0094	· 0095	· 0096	· 0097	· 0098	· 0099	· 009a	· 009b	· 009c	· 009d	· 009e	· 009f
a	SP 00a0	Ё 0401	Ђ 0402	Ѓ 0403	Є 0404	Ѕ 0405	І 0406	Ї 0407	Ј 0408	Љ 0409	Њ 040a	Ћ 040b	Ќ 040c	- 00ad	Ў 040e	Џ 040f
b	А 0410	Б 0411	В 0412	Г 0413	Д 0414	Е 0415	Ж 0416	З 0417	И 0418	Й 0419	К 041a	Л 041b	М 041c	Н 041d	О 041e	П 041f
c	Р 0420	С 0421	Т 0422	У 0423	Ф 0424	Х 0425	Ц 0426	Ч 0427	Ш 0428	Щ 0429	Ъ 042a	Ы 042b	Ь 042c	Э 042d	Ю 042e	Я 042f
d	а 0430	б 0431	в 0432	г 0433	д 0434	е 0435	ж 0436	з 0437	и 0438	й 0439	к 043a	л 043b	м 043c	н 043d	о 043e	п 043f
e	р 0440	с 0441	т 0442	у 0443	ф 0444	х 0445	ц 0446	ч 0447	ш 0448	щ 0449	ъ 044a	ы 044b	ь 044c	э 044d	ю 044e	я 044f
f	№ 2116	ё 0451	ђ 0452	ѓ 0453	є 0454	ѕ 0455	і 0456	ї 0457	ј 0458	љ 0459	њ 045a	ћ 045b	ќ 045c	§ 00a7	ў 045e	џ 045f

Fig. 147 IBM_0915: ASCII Cyrillic ISO 8859-5

IBM_0916

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	DEL 007f
8	· 0080	· 0081	· 0082	· 0083	· 0084	· 0085	· 0086	· 0087	· 0088	· 0089	· 008a	· 008b	· 008c	· 008d	· 008e	· 008f
9	· 0090	· 0091	· 0092	· 0093	· 0094	· 0095	· 0096	· 0097	· 0098	· 0099	· 009a	· 009b	· 009c	· 009d	· 009e	· 009f
a	SP 00a0	SUB 001a	¢ 00a2	£ 00a3	¤ 00a4	¥ 00a5	¦ 00a6	§ 00a7	¨ 00a8	© 00a9	× 00d7	« 00ab	¬ 00ac	- 00ad	® 00ae	‾ 203e
b	° 00b0	± 00b1	² 00b2	³ 00b3	´ 00b4	µ 00b5	¶ 00b6	• 2022	¸ 00b8	¹ 00b9	÷ 00f7	» 00bb	¼ 00bc	½ 00bd	¾ 00be	SUB 001a
c	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	‗ 2017
e	א 05d0	ב 05d1	ג 05d2	ד 05d3	ה 05d4	ו 05d5	ז 05d6	ח 05d7	ט 05d8	י 05d9	ך 05da	כ 05db	ל 05dc	ם 05dd	מ 05de	ן 05df
f	נ 05e0	ס 05e1	ע 05e2	ף 05e3	פ 05e4	ץ 05e5	צ 05e6	ק 05e7	ר 05e8	ש 05e9	ת 05ea	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a

Fig. 148 IBM_0916: ASCII Hebrew ISO 8859-8

IBM_0920

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	DEL 007f
8	· 0080	· 0081	· 0082	· 0083	· 0084	· 0085	· 0086	· 0087	· 0088	· 0089	· 008a	· 008b	· 008c	· 008d	· 008e	· 008f
9	· 0090	· 0091	· 0092	· 0093	· 0094	· 0095	· 0096	· 0097	· 0098	· 0099	· 009a	· 009b	· 009c	· 009d	· 009e	· 009f
a	SP 00a0	¡ 00a1	¢ 00a2	£ 00a3	¤ 00a4	¥ 00a5	¦ 00a6	§ 00a7	¨ 00a8	© 00a9	ª 00aa	« 00ab	¬ 00ac	- 00ad	® 00ae	¯ 00af
b	° 00b0	± 00b1	² 00b2	³ 00b3	´ 00b4	µ 00b5	¶ 00b6	· 00b7	¸ 00b8	¹ 00b9	º 00ba	» 00bb	¼ 00bc	½ 00bd	¾ 00be	¿ 00bf
c	À 00c0	Á 00c1	Â 00c2	Ã 00c3	Ä 00c4	Å 00c5	Æ 00c6	Ç 00c7	È 00c8	É 00c9	Ê 00ca	Ë 00cb	Ì 00cc	Í 00cd	Î 00ce	Ï 00cf
d	Ğ 011e	Ñ 00d1	Ò 00d2	Ó 00d3	Ô 00d4	Õ 00d5	Ö 00d6	× 00d7	Ø 00d8	Ù 00d9	Ú 00da	Û 00db	Ü 00dc	İ 0130	Ş 015e	ß 00df
e	à 00e0	á 00e1	â 00e2	ã 00e3	ä 00e4	å 00e5	æ 00e6	ç 00e7	è 00e8	é 00e9	ê 00ea	ë 00eb	ì 00ec	í 00ed	î 00ee	ï 00ef
f	ğ 011f	ñ 00f1	ò 00f2	ó 00f3	ô 00f4	õ 00f5	ö 00f6	÷ 00f7	ø 00f8	ù 00f9	ú 00fa	û 00fb	ü 00fc	ı 0131	ş 015f	ÿ 00ff

Fig. 149 IBM_0920: ASCII Turkey ISO 8859-9 Latin 5

IBM_0924

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	DEL 007f
8	· 0080	· 0081	· 0082	· 0083	· 0084	· 0085	· 0086	· 0087	· 0088	· 0089	· 008a	· 008b	· 008c	· 008d	· 008e	· 008f
9	· 0090	· 0091	· 0092	· 0093	· 0094	· 0095	· 0096	· 0097	· 0098	· 0099	· 009a	· 009b	· 009c	· 009d	· 009e	· 009f
a	SP 00a0	¡ 00a1	¢ 00a2	£ 00a3	€ 20ac	¥ 00a5	Š 0160	§ 00a7	š 0161	© 00a9	ª 00aa	« 00ab	¬ 00ac	- 00ad	® 00ae	¯ 00af
b	° 00b0	± 00b1	² 00b2	³ 00b3	Ž 017d	µ 00b5	¶ 00b6	· 00b7	ž 017e	¹ 00b9	º 00ba	» 00bb	Œ 0152	œ 0153	Ÿ 0178	¿ 00bf
c	À 00c0	Á 00c1	Â 00c2	Ã 00c3	Ä 00c4	Å 00c5	Æ 00c6	Ç 00c7	È 00c8	É 00c9	Ê 00ca	Ë 00cb	Ì 00cc	Í 00cd	Î 00ce	Ï 00cf
d	Đ 00d0	Ñ 00d1	Ò 00d2	Ó 00d3	Ô 00d4	Õ 00d5	Ö 00d6	× 00d7	Ø 00d8	Ù 00d9	Ú 00da	Û 00db	Ü 00dc	Ý 00dd	Þ 00de	ß 00df
e	à 00e0	á 00e1	â 00e2	ã 00e3	ä 00e4	å 00e5	æ 00e6	ç 00e7	è 00e8	é 00e9	ê 00ea	ë 00eb	ì 00ec	í 00ed	î 00ee	ï 00ef
f	ð 00f0	ñ 00f1	ò 00f2	ó 00f3	ô 00f4	õ 00f5	ö 00f6	÷ 00f7	ø 00f8	ù 00f9	ú 00fa	û 00fb	ü 00fc	ý 00fd	þ 00fe	ÿ 00ff

Fig. 150 IBM_0924: same as ISO 8859-15

IBM_1009

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	¤ 00a4	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	DEL 007f
8	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
9	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
b	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
c	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
f	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a

Fig. 151 IBM_1009: ASCII ISO-7 IRV (prior 1992)

IBM_1010

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	£ 00a3	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	à 00e0	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	° 00b0	ç 00e7	§ 00a7	^ 005e	_ 005f
6	µ 00b5	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	é 00e9	ù 00f9	è 00e8	¨ 00a8	DEL 007f
8	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
9	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
b	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
c	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
f	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a

Fig. 152 IBM_1010: ASCII ISO-7 France

IBM_1011

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	§ 00a7	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	Ä 00c4	Ö 00d6	Ü 00dc	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	ä 00e4	ö 00f6	ü 00fc	ß 00df	DEL 007f
8	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
9	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
b	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
c	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
f	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a

Fig. 153 IBM_1011: ASCII ISO-7 Germany

IBM_1012

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	£ 00a3	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	§ 00a7	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	° 00b0	ç 00e7	é 00e9	^ 005e	_ 005f
6	ù 00f9	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	à 00e0	ò 00f2	è 00e8	ì 00ec	DEL 007f
8	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
9	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
b	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
c	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
f	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a

Fig. 154 IBM_1012: ASCII ISO-7 Italy

IBM_1013

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	£ 00a3	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	‾ 00af	DEL 007f
8	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
9	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
b	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
c	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
f	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a

Fig. 155 IBM_1013: ASCII ISO-7 United Kingdom

IBM_1014

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	• 00b7	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	¡ 00a1	Ñ 00d1	Ç 00c7	¿ 00bf	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	´ 00b4	ñ 00f1	ç 00e7	¨ 00a8	DEL 007f
8	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
9	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
b	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
c	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
f	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a

Fig. 156 IBM_1014: ASCII ISO-7 Spain

IBM_1015

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	´ 00b4	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	Ã 00c3	Ç 00c7	Õ 00d5	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	ã 00e3	ç 00e7	õ 00f5	~ 007e	DEL 007f
8	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
9	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
b	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
c	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
f	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a

Fig. 157 IBM_1015: ASCII ISO-7 Portugal

IBM_1016

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	Æ 00c6	Ø 00d8	Å 00c5	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	æ 00e6	ø 00f8	å 00e5	¯ 00af	DEL 007f
8	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
9	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
b	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
c	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
f	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a

Fig. 158 IBM_1016: ASCII ISO-7 Norway

IBM_1017

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	¤ 00a4	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	Æ 00c6	Ø 00d8	Å 00c5	Ü 00dc	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	æ 00e6	ø 00f8	å 00e5	ü 00fc	DEL 007f
8	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
9	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
b	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
c	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
f	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a

Fig. 159 IBM_1017: ASCII ISO-7 Denmark

IBM_1018

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	¤ 00a4	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	Ä 00c4	Ö 00d6	Å 00c5	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	ä 00e4	ö 00f6	å 00e5	¯ 00af	DEL 007f
8	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
9	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
b	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
c	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
f	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a

Fig. 160 IBM_1018: ASCII ISO-7 Finland and Sweden

IBM_1019

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	¯ 00af	DEL 007f
8	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
9	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
b	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
c	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
f	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a

Fig. 161 IBM_1019: ASCII ISO-7 Belgium and Netherlands

IBM_1025

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	ђ 0452	ѓ 0453	ё 0451	є 0454	ѕ 0455	і 0456	ї 0457	ј 0458	[005b	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	љ 0459	њ 045a	ћ 045b	ќ 045c	ў 045e	џ 045f	Ъ 042a	№ 2116	Ђ 0402	] 005d	$ 0024	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Ѓ 0403	Ё 0401	Є 0404	Ѕ 0405	І 0406	Ї 0407	Ј 0408	Љ 0409	| 007c	, 002c	% 0025	_ 005f	> 003e	? 003f
7	Њ 040a	Ћ 040b	Ќ 040c	- 00ad	Ў 040e	Џ 040f	ю 044e	а 0430	б 0431	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	ц 0446	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	д 0434	е 0435	ф 0444	г 0433	х 0445	и 0438
9	й 0439	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	к 043a	л 043b	м 043c	н 043d	о 043e	п 043f
a	я 044f	~ 007e	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	р 0440	с 0441	т 0442	у 0443	ж 0436	в 0432
b	ь 044c	ы 044b	з 0437	ш 0448	э 044d	щ 0449	ч 0447	ъ 044a	Ю 042e	А 0410	Б 0411	Ц 0426	Д 0414	Е 0415	Ф 0424	Г 0413
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	Х 0425	И 0418	Й 0419	К 041a	Л 041b	М 041c
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	Н 041d	О 041e	П 041f	Я 042f	Р 0420	С 0421
e	\ 005c	§ 00a7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	Т 0422	У 0423	Ж 0416	В 0412	Ь 042c	Ы 042b
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	З 0417	Ш 0428	Э 042d	Щ 0429	Ч 0427	· 009f

Fig. 162 IBM_1025: EBCDIC Cyrillic Multilingual

	x0	x1	x2	X3	x4	x5	x6	x7	x8	x9	XA	xB	xC	xD	xE	xF
0x																
1x																
2x																
3x																
4x																
5x																
6x																
7x																
8x																
9x																
Ax																
Bx																
Cx																
Dx																
Ex																
Fx																

Fig. 163 IBM_1026: EBCDIC Turkey Latin 5

IBM_1027

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SUB 001a	｡ ff61	｢ ff62	｣ ff63	､ ff64	･ ff65	ｦ ff66	ｧ ff67	ｨ ff68	¢ 00a2	. 002e	< 003c	(0028	+ 002b	| 007c
5	& 0026	ｩ ff69	ｪ ff6a	ｫ ff6b	ｬ ff6c	ｭ ff6d	ｮ ff6e	ｯ ff6f	ｰ ff70	ｱ ff71	! 0021	$ 0024	* 002a	) 0029	; 003b	¬ 00ac
6	- 002d	/ 002f	ｲ ff72	ｳ ff73	ｴ ff74	ｵ ff75	ｶ ff76	ｷ ff77	ｸ ff78	ｹ ff79	SUB 001a	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ｺ ff7a	ｻ ff7b	ｼ ff7c	ｽ ff7d	ｾ ff7e	ｿ ff7f	ﾀ ff80	ﾁ ff81	ﾂ ff82	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	SUB 001a	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	ﾃ ff83	ﾄ ff84	ﾅ ff85	ﾆ ff86	ﾇ ff87	ﾈ ff88
9	SUB 001a	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ﾉ ff89	ﾊ ff8a	ﾋ ff8b	ﾌ ff8c	ﾍ ff8d	ﾎ ff8e
a	‾ 203e	~ 007e	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	ﾏ ff8f	ﾐ ff90	ﾑ ff91	[005b	ﾒ ff92	ﾓ ff93
b	^ 005e	£ 00a3	¥ 00a5	ﾔ ff94	ﾕ ff95	ﾖ ff96	ﾗ ff97	ﾘ ff98	ﾙ ff99	ﾚ ff9a	ﾛ ff9b	ﾜ ff9c	ﾝ ff9d	] 005d	ﾞ ff9e	ﾟ ff9f
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	\ 005c	SUB 001a	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	· 009f

Fig. 164 IBM_1027: EBCDIC Japanese Latin

IBM_1046

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	DEL 007f
8	ﺈ fe88	× 00d7	÷ 00f7	· f8f6	· f8f5	· f8f4	· f8f7	ﹱ fe71	· 0088	■ 25a0	│ 2502	─ 2500	┐ 2510	┌ 250c	└ 2514	┘ 2518
9	ﹹ fe79	ﹻ fe7b	ﹽ fe7d	ﹿ fe7f	ﹷ fe77	ﺊ fe8a	ﻰ fef0	ﻳ fef3	ﻲ fef2	ﻎ fece	ﻏ fecf	ﻐ fed0	ﻶ fef6	ﻸ fef8	ﻺ fefa	ﻼ fefc
a	SP 00a0	· f8fa	· f8f9	· f8f8	¤ 00a4	· f8fb	ﺋ fe8b	ﺑ fe91	ﺗ fe97	ﺛ fe9b	ﺟ fe9f	ﺣ fea3	، 060c	- 00ad	ﺧ fea7	ﺳ feb3
b	٠ 0660	١ 0661	٢ 0662	٣ 0663	٤ 0664	٥ 0665	٦ 0666	٧ 0667	٨ 0668	٩ 0669	ﺷ feb7	؛ 061b	ﺻ febb	ﺿ febf	ﻊ feca	؟ 061f
c	ﻋ fecb	ﺀ fe80	ﺁ fe81	ﺃ fe83	ﺅ fe85	ﺇ fe87	ﺉ fe89	ﺍ fe8d	ﺏ fe8f	ﺓ fe93	ﺕ fe95	ﺙ fe99	ﺝ fe9d	ﺡ fea1	ﺥ fea5	ﺩ fea9
d	ﺫ feab	ﺭ fead	ﺯ feaf	ﺱ feb1	ﺵ feb5	ﺹ feb9	ﺽ febd	ﻃ fec3	ﻇ fec7	ﻉ fec9	ﻍ fecd	ﻌ fecc	ﺂ fe82	ﺄ fe84	ﺎ fe8e	ﻓ fed3
e	ـ 0640	ﻑ fed1	ﻕ fed5	ﻙ fed9	ﻝ fedd	ﻡ fee1	ﻥ fee5	ﻫ feeb	ﻭ feed	ﻯ feef	ﻱ fef1	ﹰ fe70	ﹲ fe72	ﹴ fe74	ﹶ fe76	ﹸ fe78
f	ﹺ fe7a	ﹼ fe7c	ﹾ fe7e	ﻗ fed7	ﻛ fedb	ﻟ fedf	· f8fc	ﻵ fef5	ﻷ fef7	ﻹ fef9	ﻻ fefb	ﻣ fee3	ﻧ fee7	ﻬ feec	ﻩ fee9	SUB 001a

Fig. 165 IBM_1046: ASCII Arabic Windows

IBM_1089

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	DEL 007f
8	· 0080	· 0081	· 0082	· 0083	· 0084	· 0085	· 0086	· 0087	· 0088	· 0089	· 008a	· 008b	· 008c	· 008d	· 008e	· 008f
9	· 0090	· 0091	· 0092	· 0093	· 0094	· 0095	· 0096	· 0097	· 0098	· 0099	· 009a	· 009b	· 009c	· 009d	· 009e	· 009f
a	SP 00a0	SUB 001a	SUB 001a	SUB 001a	¤ 00a4	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	، 060c	- 00ad	SUB 001a	SUB 001a
b	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	؛ 061b	SUB 001a	SUB 001a	SUB 001a	؟ 061f
c	SUB 001a	ء 0621	آ 0622	أ 0623	ؤ 0624	إ 0625	ئ 0626	ا 0627	ب 0628	ة 0629	ت 062a	ث 062b	ج 062c	ح 062d	خ 062e	د 062f
d	ذ 0630	ر 0631	ز 0632	س 0633	ش 0634	ص 0635	ض 0636	ط 0637	ظ 0638	ع 0639	غ 063a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a
e	ـ 0640	ف 0641	ق 0642	ك 0643	ل 0644	م 0645	ن 0646	ه 0647	و 0648	ى 0649	ي 064a	ً 064b	ٌ 064c	ٍ 064d	َ 064e	ُ 064f
f	ِ 0650	ّ 0651	ْ 0652	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a	SUB 001a

Fig. 166 IBM_1089: ASCII ISO-8859 Arabic

IBM_1097

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	، 060c	ً 064b	ﺁ fe81	ﺂ fe82	· f8fa	ﺍ fe8d	ﺎ fe8e	· f8fb	¤ 00a4	. 002e	< 003c	(0028	+ 002b	| 007c
5	& 0026	ﺀ fe80	ﺃ fe83	ﺄ fe84	· f8f9	ﺅ fe85	ﺋ fe8b	ﺏ fe8f	ﺑ fe91	ﭖ fb56	! 0021	$ 0024	* 002a	) 0029	; 003b	¬ 00ac
6	- 002d	/ 002f	ﭘ fb58	ﺕ fe95	ﺗ fe97	ﺙ fe99	ﺛ fe9b	ﺝ fe9d	ﺟ fe9f	ﭺ fb7a	؛ 061b	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ﭼ fb7c	ﺡ fea1	ﺣ fea3	ﺥ fea5	ﺧ fea7	ﺩ fea9	ﺫ feab	ﺭ fead	ﺯ feaf	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	ﮊ fb8a	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ﺱ feb1	ﺳ feb3	ﺵ feb5	ﺷ feb7
9	ﺹ feb9	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ﺻ febb	ﺽ febd	ﺿ febf	ﻁ fec1	ﻃ fec3	ﻅ fec5
a	ﻇ fec7	~ 007e	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	ﻉ fec9	ﻊ feca	ﻋ fecb	ﻌ fecc	ﻍ fecd	ﻎ fece
b	ﻏ fecf	ﻐ fed0	ﻑ fed1	ﻓ fed3	ﻕ fed5	ﻗ fed7	ﮎ fb8e	ﻛ fedb	ﮒ fb92	ﮔ fb94	[005b	] 005d	ﻝ fedd	ﻟ fedf	ﻡ fee1	× 00d7
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ﻣ fee3	ﻥ fee5	ﻧ fee7	ﻭ feed	ﻩ fee9
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	ﻫ feeb	ﻬ feec	ﮤ fba4	ﯼ fbfc	ﯽ fbfd	ﯾ fbfe
e	\ 005c	؟ 061f	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	ـ 0640	۰ 06f0	۱ 06f1	۲ 06f2	۳ 06f3	۴ 06f4
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	۵ 06f5	۶ 06f6	۷ 06f7	۸ 06f8	۹ 06f9	· 009f

Fig. 167 IBM_1097: EBCDIC Farsi

IBM_1098

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	FS 001c	ESC 001b	DEL 007f	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	SUB 001a
8	SUB 001a	SUB 001a	، 060c	؛ 061b	؟ 061f	ً 064b	ﺁ fe81	ﺂ fe82	· f8fa	ﺍ fe8d	ﺎ fe8e	· f8fb	ﺀ fe80	ﺃ fe83	ﺄ fe84	· f8f9
9	ﺅ fe85	ﺋ fe8b	ﺏ fe8f	ﺑ fe91	ﭖ fb56	ﭘ fb58	ﺕ fe95	ﺗ fe97	ﺙ fe99	ﺛ fe9b	ﺝ fe9d	ﺟ fe9f	ﭺ fb7a	ﭼ fb7c	× 00d7	ﺡ fea1
a	ﺣ fea3	ﺥ fea5	ﺧ fea7	ﺩ fea9	ﺫ feab	ﺭ fead	ﺯ feaf	ﮊ fb8a	ﺱ feb1	ﺳ feb3	ﺵ feb5	ﺷ feb7	ﺹ feb9	ﺻ febb	« 00ab	» 00bb
b	░ 2591	▒ 2592	▓ 2593	│ 2502	┤ 2524	ﺽ febd	ﺿ febf	ﻁ fec1	ﻃ fec3	╣ 2563	║ 2551	╗ 2557	╝ 255d	¤ 00a4	ﻅ fec5	┐ 2510
c	└ 2514	┴ 2534	┬ 252c	├ 251c	─ 2500	┼ 253c	ﻇ fec7	ﻉ fec9	╚ 255a	╔ 2554	╩ 2569	╦ 2566	╠ 2560	═ 2550	╬ 256c	SUB 001a
d	ﻊ feca	ﻋ fecb	ﻌ fecc	ﻍ fecd	ﻎ fece	ﻏ fecf	ﻐ fed0	ﻑ fed1	ﻓ fed3	┘ 2518	┌ 250c	█ 2588	▄ 2584	ﻕ fed5	ﻗ fed7	▀ 2580
e	ﮎ fb8e	ﻛ fedb	ﮒ fb92	ﮔ fb94	ﻝ fedd	ﻟ fedf	ﻡ fee1	ﻣ fee3	ﻥ fee5	ﻧ fee7	ﻭ feed	ﻩ fee9	ﻫ feeb	ﻬ feec	ﮤ fba4	ﯼ fbfc
f	- 00ad	ﯽ fbfd	ﯾ fbfe	ـ 0640	۰ 06f0	۱ 06f1	۲ 06f2	۳ 06f3	۴ 06f4	۵ 06f5	۶ 06f6	۷ 06f7	۸ 06f8	۹ 06f9	■ 25a0	SP 00a0

Fig. 168 IBM_1098: ASCII Farsi Personal Computer

IBM_1140

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	à 00e0	á 00e1	ã 00e3	å 00e5	ç 00e7	ñ 00f1	¢ 00a2	. 002e	< 003c	(0028	+ 002b	| 007c
5	& 0026	é 00e9	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	! 0021	$ 0024	* 002a	) 0029	; 003b	¬ 00ac
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	Ã 00c3	Å 00c5	Ç 00c7	Ñ 00d1	¦ 00a6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	æ 00e6	¸ 00b8	Æ 00c6	€ 20ac
a	µ 00b5	~ 007e	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	^ 005e	£ 00a3	¥ 00a5	· 00b7	© 00a9	§ 00a7	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	[005b	] 005d	¯ 00af	¨ 00a8	´ 00b4	× 00d7
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	ö 00f6	ò 00f2	ó 00f3	õ 00f5
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	ü 00fc	ù 00f9	ú 00fa	ÿ 00ff
e	\ 005c	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	Ö 00d6	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 169 IBM_1140: same as IBM_0037 (Euro-Sign replaces Code Pos 9F)

IBM_1141

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	{ 007b	à 00e0	á 00e1	ã 00e3	å 00e5	ç 00e7	ñ 00f1	Ä 00c4	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	é 00e9	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	~ 007e	Ü 00dc	$ 0024	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Â 00c2	[005b	À 00c0	Á 00c1	Ã 00c3	Å 00c5	Ç 00c7	Ñ 00d1	ö 00f6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	` 0060	: 003a	# 0023	§ 00a7	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	æ 00e6	¸ 00b8	Æ 00c6	€ 20ac
a	µ 00b5	ß 00df	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	¢ 00a2	£ 00a3	¥ 00a5	· 00b7	© 00a9	@ 0040	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	¬ 00ac	| 007c	¯ 00af	¨ 00a8	´ 00b4	× 00d7
c	ä 00e4	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	¦ 00a6	ò 00f2	ó 00f3	õ 00f5
d	ü 00fc	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	} 007d	ù 00f9	ú 00fa	ÿ 00ff
e	Ö 00d6	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	\ 005c	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	] 005d	Ù 00d9	Ú 00da	· 009f

Fig. 170 IBM_1141: same as IBM_0237 (Euro-Sign replaces Code Pos 9F)

IBM_1142

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	à 00e0	á 00e1	ã 00e3	} 007d	ç 00e7	ñ 00f1	# 0023	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	é 00e9	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	€ 20ac	Å 00c5	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	Ã 00c3	$ 0024	Ç 00c7	Ñ 00d1	ø 00f8	, 002c	% 0025	_ 005f	> 003e	? 003f
7	¦ 00a6	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	` 0060	: 003a	Æ 00c6	Ø 00d8	' 0027	= 003d	" 0022
8	@ 0040	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	{ 007b	¸ 00b8	[005b	] 005d
a	µ 00b5	ü 00fc	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	¢ 00a2	£ 00a3	¥ 00a5	· 00b7	© 00a9	§ 00a7	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	¬ 00ac	| 007c	¯ 00af	¨ 00a8	´ 00b4	× 00d7
c	æ 00e6	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	ö 00f6	ò 00f2	ó 00f3	õ 00f5
d	å 00e5	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	~ 007e	ù 00f9	ú 00fa	ÿ 00ff
e	\ 005c	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	Ö 00d6	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 171 IBM_1142: same as IBM_0277 (Euro-Sign replaces Code Pos 5A)

IBM_1143

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	{ 007b	à 00e0	á 00e1	ã 00e3	} 007d	ç 00e7	ñ 00f1	§ 00a7	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	` 0060	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	€ 20ac	Å 00c5	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Â 00c2	# 0023	À 00c0	Á 00c1	Ã 00c3	$ 0024	Ç 00c7	Ñ 00d1	ö 00f6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	\ 005c	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	é 00e9	: 003a	Ä 00c4	Ö 00d6	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	æ 00e6	¸ 00b8	Æ 00c6	] 005d
a	µ 00b5	ü 00fc	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	¢ 00a2	£ 00a3	¥ 00a5	· 00b7	© 00a9	[005b	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	¬ 00ac	| 007c	¯ 00af	¨ 00a8	´ 00b4	× 00d7
c	ä 00e4	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	¦ 00a6	ò 00f2	ó 00f3	õ 00f5
d	å 00e5	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	~ 007e	ù 00f9	ú 00fa	ÿ 00ff
e	É 00c9	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	@ 0040	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 172 IBM_1143: same as IBM_0278 (Euro-Sign replaces Code Pos 5A)

IBM_1144

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	{ 007b	á 00e1	ã 00e3	å 00e5	\ 005c	ñ 00f1	° 00b0	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	] 005d	ê 00ea	ë 00eb	} 007d	í 00ed	î 00ee	ï 00ef	~ 007e	ß 00df	é 00e9	$ 0024	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	Ã 00c3	Å 00c5	Ç 00c7	Ñ 00d1	ò 00f2	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	ù 00f9	: 003a	£ 00a3	§ 00a7	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	[005b	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	æ 00e6	¸ 00b8	Æ 00c6	€ 20ac
a	µ 00b5	ì 00ec	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	¢ 00a2	# 0023	¥ 00a5	· 00b7	© 00a9	@ 0040	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	¬ 00ac	| 007c	¯ 00af	¨ 00a8	´ 00b4	× 00d7
c	à 00e0	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	ö 00f6	¦ 00a6	ó 00f3	õ 00f5
d	è 00e8	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	ü 00fc	` 0060	ú 00fa	ÿ 00ff
e	ç 00e7	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	Ö 00d6	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 173 IBM_1144: same as IBM_0280 (Euro-Sign replaces Code Pos 9F)

IBM_1145

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	à 00e0	á 00e1	ã 00e3	å 00e5	ç 00e7	¦ 00a6	[005b	. 002e	< 003c	(0028	+ 002b	| 007c
5	& 0026	é 00e9	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	] 005d	$ 0024	* 002a	) 0029	; 003b	¬ 00ac
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	Ã 00c3	Å 00c5	Ç 00c7	# 0023	ñ 00f1	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	` 0060	: 003a	Ñ 00d1	@ 0040	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	æ 00e6	¸ 00b8	Æ 00c6	€ 20ac
a	µ 00b5	¨ 00a8	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	¢ 00a2	£ 00a3	¥ 00a5	· 00b7	© 00a9	§ 00a7	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	^ 005e	! 0021	¯ 00af	~ 007e	´ 00b4	× 00d7
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	ö 00f6	ò 00f2	ó 00f3	õ 00f5
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	ü 00fc	ù 00f9	ú 00fa	ÿ 00ff
e	\ 005c	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	Ö 00d6	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 174 IBM_1145: same as IBM_0284 (Euro-Sign replaces Code Pos 9F)

IBM_1146

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	à 00e0	á 00e1	ã 00e3	å 00e5	ç 00e7	ñ 00f1	$ 0024	. 002e	< 003c	(0028	+ 002b	| 007c
5	& 0026	é 00e9	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	! 0021	£ 00a3	* 002a	) 0029	; 003b	¬ 00ac
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	Ã 00c3	Å 00c5	Ç 00c7	Ñ 00d1	¦ 00a6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	æ 00e6	¸ 00b8	Æ 00c6	€ 20ac
a	µ 00b5	¯ 00af	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	¢ 00a2	[005b	¥ 00a5	· 00b7	© 00a9	§ 00a7	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	^ 005e	] 005d	~ 007e	¨ 00a8	´ 00b4	× 00d7
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	ö 00f6	ò 00f2	ó 00f3	õ 00f5
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	ü 00fc	ù 00f9	ú 00fa	ÿ 00ff
e	\ 005c	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	Ö 00d6	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 175 IBM_1146: same as IBM_0285 (Euro-Sign replaces Code Pos 9F)

IBM_1147

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	@ 0040	á 00e1	ã 00e3	å 00e5	\ 005c	ñ 00f1	° 00b0	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	{ 007b	ê 00ea	ë 00eb	} 007d	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	§ 00a7	$ 0024	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	Ã 00c3	Å 00c5	Ç 00c7	Ñ 00d1	ù 00f9	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	µ 00b5	: 003a	£ 00a3	à 00e0	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	[005b	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	æ 00e6	¸ 00b8	Æ 00c6	€ 20ac
a	` 0060	¨ 00a8	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	¢ 00a2	# 0023	¥ 00a5	· 00b7	© 00a9	] 005d	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	¬ 00ac	| 007c	¯ 00af	~ 007e	´ 00b4	× 00d7
c	é 00e9	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	ö 00f6	ò 00f2	ó 00f3	õ 00f5
d	è 00e8	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	ü 00fc	¦ 00a6	ú 00fa	ÿ 00ff
e	ç 00e7	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	Ö 00d6	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 176 IBM_1147: same as IBM_0297 (Euro-Sign replaces Code Pos 9F)

IBM_1148

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	à 00e0	á 00e1	ã 00e3	å 00e5	ç 00e7	ñ 00f1	[005b	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	é 00e9	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	] 005d	$ 0024	* 002a	) 0029	; 003b	^ 005e
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	Ã 00c3	Å 00c5	Ç 00c7	Ñ 00d1	¦ 00a6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	` 0060	: 003a	# 0023	@ 0040	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	ð 00f0	ý 00fd	þ 00fe	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	æ 00e6	¸ 00b8	Æ 00c6	€ 20ac
a	µ 00b5	~ 007e	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	Ð 00d0	Ý 00dd	Þ 00de	® 00ae
b	¢ 00a2	£ 00a3	¥ 00a5	· 00b7	© 00a9	§ 00a7	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	¬ 00ac	| 007c	¯ 00af	¨ 00a8	´ 00b4	× 00d7
c	{ 007b	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	ö 00f6	ò 00f2	ó 00f3	õ 00f5
d	} 007d	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	ü 00fc	ù 00f9	ú 00fa	ÿ 00ff
e	\ 005c	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	Ö 00d6	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 177 IBM_1148: same as IBM_0500 (Euro-Sign replaces Code Pos 9F)

IBM_1149

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	· 009c	HT 0009	· 0086	DEL 007f	· 0097	· 008d	· 008e	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	· 009d	· 0085	BS 0008	· 0087	CAN 0018	EM 0019	· 0092	· 008f	FS 001c	GS 001d	RS 001e	US 001f
2	· 0080	· 0081	· 0082	· 0083	· 0084	LF 000a	ETB 0017	ESC 001b	· 0088	· 0089	· 008a	· 008b	· 008c	ENQ 0005	ACK 0006	BEL 0007
3	· 0090	· 0091	SYN 0016	· 0093	· 0094	· 0095	· 0096	EOT 0004	· 0098	· 0099	· 009a	· 009b	DC4 0014	NAK 0015	· 009e	SUB 001a
4	SP 0020	SP 00a0	â 00e2	ä 00e4	à 00e0	á 00e1	ã 00e3	å 00e5	ç 00e7	ñ 00f1	Þ 00de	. 002e	< 003c	(0028	+ 002b	! 0021
5	& 0026	é 00e9	ê 00ea	ë 00eb	è 00e8	í 00ed	î 00ee	ï 00ef	ì 00ec	ß 00df	Æ 00c6	$ 0024	* 002a	) 0029	; 003b	Ö 00d6
6	- 002d	/ 002f	Â 00c2	Ä 00c4	À 00c0	Á 00c1	Ã 00c3	Å 00c5	Ç 00c7	Ñ 00d1	¦ 00a6	, 002c	% 0025	_ 005f	> 003e	? 003f
7	ø 00f8	É 00c9	Ê 00ca	Ë 00cb	È 00c8	Í 00cd	Î 00ce	Ï 00cf	Ì 00cc	ð 00f0	: 003a	# 0023	Ð 00d0	' 0027	= 003d	" 0022
8	Ø 00d8	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	« 00ab	» 00bb	` 0060	ý 00fd	{ 007b	± 00b1
9	° 00b0	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f	p 0070	q 0071	r 0072	ª 00aa	º 00ba	} 007d	¸ 00b8	] 005d	€ 20ac
a	µ 00b5	ö 00f6	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	¡ 00a1	¿ 00bf	@ 0040	Ý 00dd	[005b	® 00ae
b	¢ 00a2	£ 00a3	¥ 00a5	· 00b7	© 00a9	§ 00a7	¶ 00b6	¼ 00bc	½ 00bd	¾ 00be	¬ 00ac	| 007c	¯ 00af	¨ 00a8	\ 005c	× 00d7
c	þ 00fe	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	- 00ad	ô 00f4	~ 007e	ò 00f2	ó 00f3	õ 00f5
d	æ 00e6	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f	P 0050	Q 0051	R 0052	¹ 00b9	û 00fb	ü 00fc	ù 00f9	ú 00fa	ÿ 00ff
e	´ 00b4	÷ 00f7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	² 00b2	Ô 00d4	^ 005e	Ò 00d2	Ó 00d3	Õ 00d5
f	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	³ 00b3	Û 00db	Ü 00dc	Ù 00d9	Ú 00da	· 009f

Fig. 178 IBM_1149: same as IBM_0871 (Euro-Sign replaces Code Pos 9F)

	x0	x1	x2	X3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL 0000		STX 0001	ETX 0003		HT 0009						VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
1x	DLE 0010	DC1 0011	DC2 0012	DC3 0013					CAN 0018	EM 0019				GS 001D	RS 001E	US 001F
2x			FS 001C			LF 000A	ETB 0017	ESC 001B						ENQ 0005	ACK 0006	BEL 0007
3x			SYN 0016					EOT 0004					DC4 0014	NAK 0015		SUB 001A
4x	BLK 0020											. 002E	< 003C	(0028	+ 002B	¦ 00A6
5x	& 0026										! 0021	$ 0024	* 002A	) 0029	; 003B	
6x	- 002D	/ 002F										, 002C	% 0025	_ 005F	> 003E	? 003F
7x										` 0060	: 003A	# 0023	@ 0040	' 0027	= 003D	" 0022
8x		a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069						
9x		j 006A	k 006B	l 006C	m 006D	n 006E	o 006F	p 0070	q 0071	r 0072						
Ax		~ 007E	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A						
Bx	^ 005E										[005B	] 005D				
Cx	{ 007B	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049						
Dx	} 007D	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F	P 0050	Q 0051	R 0052						
Ex	\ 005C	÷ 00F7	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A						
Fx	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039						DEL 007F

Fig. 179 IBM_1303: Code 128

ISO_646

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	DEL 007f
8	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000
9	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000
a	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000
b	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000
c	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000
d	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000
e	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000
f	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000	NUL 0000

Fig. 180 ISO_646: ISO 646 IRV: 1991 (US ASCII, 0x00-0x7F)

ISO_8859-1

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	DEL 007f
8	· 0080	· 0081	· 0082	· 0083	· 0084	· 0085	· 0086	· 0087	· 0088	· 0089	· 008a	· 008b	· 008c	· 008d	· 008e	· 008f
9	· 0090	· 0091	· 0092	· 0093	· 0094	· 0095	· 0096	· 0097	· 0098	· 0099	· 009a	· 009b	· 009c	· 009d	· 009e	· 009f
a	SP 00a0	¡ 00a1	¢ 00a2	£ 00a3	¤ 00a4	¥ 00a5	¦ 00a6	§ 00a7	¨ 00a8	© 00a9	ª 00aa	« 00ab	¬ 00ac	- 00ad	® 00ae	¯ 00af
b	° 00b0	± 00b1	² 00b2	³ 00b3	´ 00b4	µ 00b5	¶ 00b6	· 00b7	¸ 00b8	¹ 00b9	º 00ba	» 00bb	¼ 00bc	½ 00bd	¾ 00be	¿ 00bf
c	À 00c0	Á 00c1	Â 00c2	Ã 00c3	Ä 00c4	Å 00c5	Æ 00c6	Ç 00c7	È 00c8	É 00c9	Ê 00ca	Ë 00cb	Ì 00cc	Í 00cd	Î 00ce	Ï 00cf
d	Ð 00d0	Ñ 00d1	Ò 00d2	Ó 00d3	Ô 00d4	Õ 00d5	Ö 00d6	× 00d7	Ø 00d8	Ù 00d9	Ú 00da	Û 00db	Ü 00dc	Ý 00dd	Þ 00de	ß 00df
e	à 00e0	á 00e1	â 00e2	ã 00e3	ä 00e4	å 00e5	æ 00e6	ç 00e7	è 00e8	é 00e9	ê 00ea	ë 00eb	ì 00ec	í 00ed	î 00ee	ï 00ef
f	ð 00f0	ñ 00f1	ò 00f2	ó 00f3	ô 00f4	õ 00f5	ö 00f6	÷ 00f7	ø 00f8	ù 00f9	ú 00fa	û 00fb	ü 00fc	ý 00fd	þ 00fe	ÿ 00ff

Fig. 181 ISO_8859-1: West European - Latin1 (Unicode 3.0, U+0000 - U+00FF)

ISO_8859-15

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	DEL 007f
8	· 0080	· 0081	· 0082	· 0083	· 0084	· 0085	· 0086	· 0087	· 0088	· 0089	· 008a	· 008b	· 008c	· 008d	· 008e	· 008f
9	· 0090	· 0091	· 0092	· 0093	· 0094	· 0095	· 0096	· 0097	· 0098	· 0099	· 009a	· 009b	· 009c	· 009d	· 009e	· 009f
a	SP 00a0	¡ 00a1	¢ 00a2	£ 00a3	€ 20ac	¥ 00a5	Š 0160	§ 00a7	š 0161	© 00a9	ª 00aa	« 00ab	¬ 00ac	- 00ad	® 00ae	¯ 00af
b	° 00b0	± 00b1	² 00b2	³ 00b3	Ž 017d	µ 00b5	¶ 00b6	· 00b7	ž 017e	¹ 00b9	º 00ba	» 00bb	Œ 0152	œ 0153	Ÿ 0178	¿ 00bf
c	À 00c0	Á 00c1	Â 00c2	Ã 00c3	Ä 00c4	Å 00c5	Æ 00c6	Ç 00c7	È 00c8	É 00c9	Ê 00ca	Ë 00cb	Ì 00cc	Í 00cd	Î 00ce	Ï 00cf
d	Ð 00d0	Ñ 00d1	Ò 00d2	Ó 00d3	Ô 00d4	Õ 00d5	Ö 00d6	× 00d7	Ø 00d8	Ù 00d9	Ú 00da	Û 00db	Ü 00dc	Ý 00dd	Þ 00de	ß 00df
e	à 00e0	á 00e1	â 00e2	ã 00e3	ä 00e4	å 00e5	æ 00e6	ç 00e7	è 00e8	é 00e9	ê 00ea	ë 00eb	ì 00ec	í 00ed	î 00ee	ï 00ef
f	ð 00f0	ñ 00f1	ò 00f2	ó 00f3	ô 00f4	õ 00f5	ö 00f6	÷ 00f7	ø 00f8	ù 00f9	ú 00fa	û 00fb	ü 00fc	ý 00fd	þ 00fe	ÿ 00ff

Fig. 182 ISO_8859-15: West European - Latin 15 (includes Euro sign)

PC_437

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	☺ 263a	☻ 263b	♥ 2665	♦ 2666	♣ 2663	♠ 2660	• 2022	◘ 25d8	○ 25cb	◙ 25d9	♂ 2642	♀ 2640	♪ 266a	♫ 266b	☼ 263c
1	► 25ba	◄ 25c4	↕ 2195	‼ 203c	¶ 00b6	§ 00a7	▬ 25ac	↨ 21a8	↑ 2191	↓ 2193	→ 2192	← 2190	∟ 221f	↔ 2194	▲ 25b2	▼ 25bc
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	DEL 007f
8	Ç 00c7	ü 00fc	é 00e9	â 00e2	ä 00e4	à 00e0	å 00e5	ç 00e7	ê 00ea	ë 00eb	è 00e8	ï 00ef	î 00ee	ì 00ec	Ä 00c4	Å 00c5
9	É 00c9	æ 00e6	Æ 00c6	ô 00f4	ö 00f6	ò 00f2	û 00fb	ù 00f9	ÿ 00ff	Ö 00d6	Ü 00dc	¢ 00a2	£ 00a3	¥ 00a5	₧ 20a7	ƒ 0192
a	á 00e1	í 00ed	ó 00f3	ú 00fa	ñ 00f1	Ñ 00d1	ª 00aa	º 00ba	¿ 00bf	® 00ae	¬ 00ac	½ 00bd	¼ 00bc	¡ 00a1	« 00ab	» 00bb
b	░ 2591	▒ 2592	▓ 2593	│ 2502	┤ 2524	╡ 2561	╢ 2562	╖ 2556	╕ 2555	╣ 2563	║ 2551	╗ 2557	╝ 255d	╜ 255c	╛ 255b	┐ 2510
c	└ 2514	┴ 2534	┬ 252c	├ 251c	─ 2500	┼ 253c	╞ 255e	╟ 255f	╚ 255a	╔ 2554	╩ 2569	╦ 2566	╠ 2560	═ 2550	╬ 256c	╧ 2567
d	╨ 2568	╤ 2564	╥ 2565	╙ 2559	╘ 2558	╒ 2552	╓ 2553	╫ 256b	╪ 256a	┘ 2518	┌ 250c	█ 2588	▄ 2584	▌ 258c	▐ 2590	▀ 2580
e	α 03b1	ß 00df	Γ 0393	π 03c0	Σ 03a3	σ 03c3	µ 00b5	τ 03c4	Φ 03a6	Θ 0398	Ω 03a9	δ 03b4	∞ 221e	φ 03d5	ε 03b5	∩ 2229
f	≡ 2261	± 00b1	≥ 2265	≤ 2264	⌠ 2320	⌡ 2321	÷ 00f7	÷ 00f7	° 00b0	• 2022	· 00b7	√ 221a	^{n} 207f	^{2} 00b2	■ 25a0	NUL 0000

Fig. 183 PC_437: IBM PC (DOS Latin US)

PC_850

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	☺ 263a	☻ 263b	♥ 2665	♦ 2666	♣ 2663	♠ 2660	• 2022	◘ 25d8	○ 25cb	◙ 25d9	♂ 2642	♀ 2640	♪ 266a	♫ 266b	☼ 263c
1	► 25ba	◄ 25c4	↕ 2195	‼ 203c	¶ 00b6	§ 00a7	▬ 25ac	↨ 21a8	↑ 2191	↓ 2193	→ 2192	← 2190	∟ 221f	↔ 2194	▲ 25b2	▼ 25bc
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	⌂ 2302
8	Ç 00c7	ü 00fc	é 00e9	â 00e2	ä 00e4	à 00e0	å 00e5	ç 00e7	ê 00ea	ë 00eb	è 00e8	ï 00ef	î 00ee	ì 00ec	Ä 00c4	Å 00c5
9	É 00c9	æ 00e6	Æ 00c6	ô 00f4	ö 00f6	ò 00f2	û 00fb	ù 00f9	ÿ 00ff	Ö 00d6	Ü 00dc	ø 00f8	£ 00a3	Ø 00d8	× 00d7	ƒ 0192
a	á 00e1	í 00ed	ó 00f3	ú 00fa	ñ 00f1	Ñ 00d1	ª 00aa	º 00ba	¿ 00bf	® 00ae	¬ 00ac	½ 00bd	¼ 00bc	¡ 00a1	« 00ab	» 00bb
b	░ 2591	▒ 2592	▓ 2593	│ 2502	┤ 2524	Á 00c1	Â 00c2	À 00c0	© 00a9	╣ 2563	║ 2551	╗ 2557	╝ 255d	¢ 00a2	¥ 00a5	┐ 2510
c	└ 2514	┴ 2534	┬ 252c	├ 251c	─ 2500	┼ 253c	ã 00e3	Ã 00c3	╚ 255a	╔ 2554	╩ 2569	╦ 2566	╠ 2560	═ 2550	╬ 256c	¤ 00a4
d	ð 00f0	Ð 00d0	Ê 00ca	Ë 00cb	È 00c8	ı 0131	Í 00cd	Î 00ce	Ï 00cf	┘ 2518	┌ 250c	█ 2588	▄ 2584	¦ 00a6	Ì 00cc	▀ 2580
e	Ó 00d3	ß 00df	Ô 00d4	Ò 00d2	õ 00f5	Õ 00d5	µ 00b5	þ 00fe	Þ 00de	Ú 00da	Û 00db	Ù 00d9	ý 00fd	Ý 00dd	‾ 203e	´ 00b4
f	- 002d	± 00b1	‗ 2017	¾ 00be	¶ 00b6	§ 00a7	÷ 00f7	¸ 00b8	° 00b0	¨ 00a8	· 00b7	^{1} 00b9	^{3} 00b3	^{2} 00b2	■ 25a0	SP 00a0

Fig. 184 PC_850: IBM PC (DOS Latin 1 Western Europe)

PC_1250

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	⌂ 2302
8	€ 20ac	· 0081	‚ 201a	· 0083	„ 201e	… 2026	† 2020	‡ 2021	· 0088	‰ 2030	Š 0160	‹ 2039	Ś 015a	Ť 0164	Ž 017d	Ź 0179
9	· 0090	‘ 2018	’ 2019	“ 201c	” 201d	• 2022	– 2013	— 2014	· 0098	™ 2122	š 0161	› 203a	ś 015b	ť 0165	ž 017e	ź 017a
a	SP 00a0	ˇ 02c7	˘ 02d8	Ł 0141	¤ 00a4	Ą 0104	¦ 00a6	§ 00a7	¨ 00a8	© 00a9	Ş 015e	« 00ab	¬ 00ac	- 00ad	® 00ae	Ż 017b
b	° 00b0	± 00b1	˛ 02db	ł 0142	´ 00b4	µ 00b5	¶ 00b6	· 00b7	¸ 00b8	ą 0105	ş 015f	» 00bb	Ľ 013d	˝ 02dd	Ǐ 013e	ż 017c
c	Ŕ 0154	Á 00c1	Â 00c2	Ă 0102	Ä 00c4	Ĺ 0139	Ć 0106	Ç 00c7	Č 010c	É 00c9	Ę 0118	Ë 00cb	Ě 011a	Í 00cd	Î 00ce	Ď 010e
d	Đ 0110	Ń 0143	Ň 0147	Ó 00d3	Ô 00d4	Ő 0150	Ö 00d6	× 00d7	Ř 0158	Ů 016e	Ú 00da	Ű 0170	Ü 00dc	Ý 00dd	Ţ 0162	ß 00df
e	ŕ 0155	á 00e1	â 00e2	ă 0103	ä 00e4	ĺ 013a	ć 0107	ç 00e7	č 010d	é 00e9	ę 0119	ë 00eb	ě 011b	í 00ed	î 00ee	ǰ 01f0
f	đ 0111	ń 0144	ň 0148	ó 00f3	ô 00f4	ő 0151	ö 00f6	÷ 00f7	ř 0159	ů 016f	ú 00fa	ű 0171	ü 00fc	ý 00fd	ţ 0163	˙ 02d9

Fig. 185 PC_1250: MS-Windows Latin-2

PC_1252

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000a	VT 000b	FF 000c	CR 000d	SO 000e	SI 000f
1	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001a	ESC 001b	FS 001c	GS 001d	RS 001e	US 001f
2	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002a	+ 002b	, 002c	- 002d	. 002e	/ 002f
3	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003a	; 003b	< 003c	= 003d	> 003e	? 003f
4	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004a	K 004b	L 004c	M 004d	N 004e	O 004f
5	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005a	[005b	\ 005c	] 005d	^ 005e	_ 005f
6	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006a	k 006b	l 006c	m 006d	n 006e	o 006f
7	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007a	{ 007b	| 007c	} 007d	~ 007e	DEL 007f
8	€ 20ac	· 0081	‚ 201a	ƒ 0192	„ 201e	… 2026	† 2020	‡ 2021	ˆ 02c6	‰ 2030	Š 0160	‹ 2039	Œ 0152	· 008d	Ž 017d	· 008f
9	· 0090	‘ 2018	’ 2019	“ 201c	” 201d	• 2022	– 2013	— 2014	˜ 02dc	™ 2122	š 0161	› 203a	œ 0153	· 009d	ž 017e	Ÿ 0178
a	SP 00a0	¡ 00a1	¢ 00a2	£ 00a3	¤ 00a4	¥ 00a5	¦ 00a6	§ 00a7	¨ 00a8	© 00a9	ª 00aa	« 00ab	¬ 00ac	- 00ad	® 00ae	¯ 00af
b	° 00b0	± 00b1	² 00b2	³ 00b3	´ 00b4	µ 00b5	¶ 00b6	· 00b7	¸ 00b8	¹ 00b9	º 00ba	» 00bb	¼ 00bc	½ 00bd	¾ 00be	¿ 00bf
c	À 00c0	Á 00c1	Â 00c2	Ã 00c3	Ä 00c4	Å 00c5	Æ 00c6	Ç 00c7	È 00c8	É 00c9	Ê 00ca	Ë 00cb	Ì 00cc	Í 00cd	Î 00ce	Ï 00cf
d	Ð 00d0	Ñ 00d1	Ò 00d2	Ó 00d3	Ô 00d4	Õ 00d5	Ö 00d6	× 00d7	Ø 00d8	Ù 00d9	Ú 00da	Û 00db	Ü 00dc	Ý 00dd	Þ 00de	ß 00df
e	à 00e0	á 00e1	â 00e2	ã 00e3	ä 00e4	å 00e5	æ 00e6	ç 00e7	è 00e8	é 00e9	ê 00ea	ë 00eb	ì 00ec	í 00ed	î 00ee	ï 00ef
f	ð 00f0	ñ 00f1	ò 00f2	ó 00f3	ô 00f4	õ 00f5	ö 00f6	÷ 00f7	ø 00f8	ù 00f9	ú 00fa	û 00fb	ü 00fc	ý 00fd	þ 00fe	ÿ 00ff

Fig. 186 PC_1252: MS-Windows Latin-1

UTF16LE and UTF16BE

What is UTF-16 ?

UTF-16 uses a single 16-bitcode unit to encode the most common 63K characters, and a pair of 16-bit code unites, called surrogates, to encode the 1M less commonly used characters in Unicode.

Originally, Unicode was designed as a pure 16-bit encoding, aimed at representing all modern scripts. (Ancient scripts were to be represented with private-use characters.) Over time, and especially after the addition of over 14,500 composite characters for compatibility with legacy sets, it became clear that 16-bits were not sufficient for the user community. Out of this arose UTF-16.

What are surrogates?

Surrogates are code points from two special ranges of Unicode values, reserved for use as the leading, and trailing values of paired code units in UTF-16. Leading, also called high, surrogates are from $D800_{16}$ to $DBFF_{16}$, and trailing, or low, surrogates are from $DC00_{16}$ to $DFFF_{16}$. They are called surrogates, since they do not represent characters directly, but only as a pair.

6.5 Appendix D: Trace Formatter

6.5.1 Introduction

This chapter contains the user's guide of trace formatting tool ***Trace Formatter, Release V2.13*** and higher, called Formatter from now. The Formatter enables the user to convert traces to a comfortable, readable, adaptable shape.

Note that parts of this chapter refer to CIS internal structures and are for Océ service and development experts only.

6.5.2 System requirements

The Formatter will be delivered in two different releases referring to operating system. At present Formatter supports Windows XP and MVS (z/OS, OS/390).

- **UNDER WINDOWS XP**

With regard to a reasonable performance a Pentium4 system
with a clock rate of at least 3 GHz is recommended.

- **UNDER MVS**

A C-Language Runtime Environment is required.

6.5.3 User's guide

- **USER INTERFACE**

a) Under NT/2000

The Formatter supports a command line user interface. The user has to specify command line parameters to influence its behavior.

Usage: TrcForm <source file 1> <source file 2> <target file> [<command file>]

<source file 1>	Name of first Trace file.
<source file 2>	Name of second Trace file.
<target file>	Name of output file. Contains filtered data.
<command file>	Name of command file. Used to set filter and format options.

Source File Conventions:

If only one source file should be handled the other file name has to be substituted through quotation marks working as a place holder for absent file name (e.g. file1 "" OR "" file2).

If both file names are specified formatter tries to handle both files in order of its record order. If record dates don't match the trace of older records keeps unhandled.

Instead of <source file> the user can give two special commands:

TrcForm –h	Displays the user’s manual.
TrcForm –c	Displays a set of commands helpful to create and customize a meaningful command file.

To specify a command file is optional. In case of not declaring a third parameter Formatter will use internal defined defaults.

b) *Under MVS*

In an MVS environment the communication to the Formatter works via DDnames.

The following DDnames will be used:

TRCIN1	DDname of first Trace file.
TRCIN2	DDname of second Trace file.
TRCOUT	DDname of output file. Contains filtered data.
SYSIN	DDname of command file. Used to set filter and format options.

- <u>COMMAND SPECIFICATION</u>

To control the behavior of formatted output the user can specify formatting and filter commands collected in a file called command file.

a) General conventions

Manipulating the command file the user has to pay attention to some general conventions listed as follows:

1. Parameter file contains commands to specify the behavior of Formatter. It will be distinguish between filter and formatter commands.

2. Any C-like comments are allowed everywhere: /* comment */. Nested comments are allowed.

3. Commands and keywords are case sensitive.

4. Whitespace like <SPACE>,<CR>,<LF>,<TAB> are allowed everywhere and will be ignored except inside character strings.

5. Character strings have to be covered through apostrophes: 'Hallo'.

6. There are no negative numerical values allowed.

7. The range operator is '-': '23 - 45' means from 23 to including 45.

8. The parameter file contains multiple commands of general syntax:

   <command>(<argument>[-<argument][,<argument>[-<argument]] ...).

9. Commands must be keywords.

10. Arguments can be (k)eywords, (n)umerical values or (s)tring values depending on type of command.

11. If more then one filter command with the same keyword is specified the filter works like a single command including all arguments.

12. To turn off the related filter command has to be removed from parameter file or to comment out.

13. If more then one format command with the same keyword is specified the Formatter takes the LAST command of this keyword.

14. Format commands that not appear will be represented through default values internal (in this documentation covered in brackets).

15. Some filter commands support arguments containing a single wildcard character as trailing character. The wildcard ('*') replaces any or no character. (e.g. 'Hallo*' matches to 'HalloWorld' and 'Hallo').

    Wildcards in the middle of an argument (e.g. 'Ha*llo') cause an error.

16. All filter commands are also available as negative filters. With the prefix 'Not' all records matching the arguments are printed out.

b) Filter commands

Filter commands force the Formatter to include specified records only and exclude non specified records referring to corresponding fields. Negative filter commands (Command prefix = Not) cause the formatter to exclude all specified records.

Supported filter commands are:

CompId **NotCompId**	Specifies a list or range of component identifications to be used for selecting trace records using the PSTCMPID field [Not]CompId(n1[-n2][,n3[-n4][,n5[-n6]]] ...)
CpuTime **NotCpuTime**	Specifies a list or range of time entries to be used for selecting trace records using the PSTCPUT field [Not]CpuTime(s1[-s2][,s3[-s4][,s5[-s6]]] ...) s:hhmmssiii
Date **NotDate**	Specifies a list or range of date entries to be used for selecting trace records using the PSTDATE field [Not]Date(s1[-s2][,s3[-s4][,s5[-s6]]] ...) s:yyyymmdd
DayTime **NotDayTime**	Specifies a list or range of time entries to be used for selecting trace records using the PSTTIME field [Not]DayTime(s1[-s2][,s3[-s4][,s5[-s6]]] ...) s:hhmmssiii
EntryLevel **NotEntryLevel**	Specifies a list or range of entry levels to be used for selecting trace records using the PSTENTLV field [Not]EntryLevel(n1[-n2][,n3[-n4][,n5[-n6]]] ...)

OsId **NotOsId**	Specifies a list of operating system identifications to be used for selecting trace records using the PSTOSID field [Not]OsId(s1,s2,s3 ...) s:{ MV | DO | BS | SI | UN | NT } MV MVS SI Simulator DO VSE UN SCO Unix BS BS2000 NT Win NT
OwnerId **NotOwnerId**	Specifies a list of owner identifications to be used for selecting trace records using the PSTOWNID field [Not]OwnerId(s1,s2,s3 ...) s:{PS|LI}
PrinterId **NotPrinterId**	Specifies a list of printer identifications to be used for selecting trace records using the PSTPRTID field [Not]PrinterId(s1,s2,s3 ...) s consist of maximum 8 characters
RecordId **NotRecordId**	Specifies a list or range of record identifications to be used for selecting trace records using the PSTENTID field [Not]RecordId(n1[-n2][,n3[-n4][,n5[-n6]]] ...)
Sequence **NotSequence**	Specifies a list or range of record sequence numbers to be used for selecting trace records using the PSTSEQ field [Not]Sequence(n1[-n2][,n3[-n4][,n5[-n6]]] ...)
SegLevel **NotSegLevel**	Specifies a list or range of segment levels to be used for selecting segments using level field [Not]SegLevel(n1[-n2][,n3[-n4][,n5[-n6]]] ...)
TimeDiff **NotTimeDiff**	Specifies a list or range of time entries to be used for selecting trace records using the differential PSTCPU time from previous to current header [Not]TimeDiff(s1[-s2][,s3[-s4][,s5[-s6]]] ...) s:hhmmssiii

ThreadName **NotThreadName**	Specifies a list of thread names to be used for selecting trace records using thread name field (PSTTHNAM) Wildcards are allowed (see General Conventions). [Not]ThreadName(s1,s2,s3 ...) s consist of exact 8 characters
SegCompId **NotSegCompId**	Specifies a list of segment component identifiers used selecting records using segment identifier component id field (PS entries only). Wildcards are allowed (see General Conventions). [Not]SegCompId(s1,s2,s3 ...)
SegModNam **NotSegModNam**	Specifies a list of segment module name identifiers used for selecting records using segment identifier module name field (PS entries only). Wildcards are allowed (see General Conventions). [Not]SegModNam(s1,s2,s3 ...)
SegThis **NotSegThis**	Specifies a list of segment component identifiers used for selecting records using segment identifier this pointer field (PS entries only). [Not]SegThis(n1[-n2][,n3[-n4][,n5[-n6]]] ...)
SegFctDepth **NotSegFctDepth**	Specifies a list of segment component identifiers used for selecting records using segment identifier function depth field (PS entries only). [Not]SegFctDepth(n1[-n2][,n3[-n4][,n5[-n6]]] ...)
SegText **NotSegText**	Specifies a list of segment module name identifiers used for selecting records using segment identifier text field (PS entries only). Wildcards are allowed (see General Conventions). [Not]SegText(s1,s2,s3 ...)

c) Format commands

Format commands influence the outer shape of printed records.

To all size specification command applies: If a specified field size is less than required space the field size increases automatic and lasting. This behavior can be switched off by turning Truncate mode on.

Supported format commands are:

AutoIndent

This command is valid for PS segment identifiers only and determines a character string which will used to make the function depth clear. The autoindent string will insert a depths time left of text of segment identification.

The occurrence of command AutoIndent switches on the automatic indent mode. On default the autoindent mode is off and no characters will inserted.

If command SegColumns is set AutoIndent will be ignored.

On default the AutoIndent mode is off.

AutoIndent(s) s: string representing function depth ['..']

Gap

Specifies the gap among the ruler fields.

Gap(s) s: character string representing gap [' | ']

HexOnError

Specifies whether an hex listing of the unhandled rest of trace file should be printed out in case of inconsistent data. If HexOnError mode has been turned on the bytes will be interpreted as ASCII or EBCDIC data dependent on command DataCode.

HexOnError(s) s:{ON|OFF} [OFF]

FileDump

Forces a printout of source file as hex listing without any smart formattings. If FileDump mode is ON all other commands will be ignored except for Width and DataCode. Width specifies the maximum line width. Line width expands automatically if it is to narrow to show at least 16 Bytes in a line.

Command DataCode specifies the character encoding standard.

FileDump(s) s:{ON|OFF} [OFF]

Data Specifies whether the output listing should include a printout of the data portion of each trace record segment

Data(s) s:{ON|OFF} [ON]

DataCode Specifies the encoding standard used in the data contained in the trace file

DataCode(s) s:{EBCDIC|ASCII} [ASCII]

Ruler Specifies the fields of records used as rulers.

The occurrence of a keyword determines the occurrence of the corresponding ruler in printout.

The order of keywords determines the order corresponding

rulers in printout.

Rulers left of keyword SegId are left aligned. Rulers right of keyword SegId are right aligned.

All keywords can occur twice or more but not SegId.

If keyword SegId not occurs no segment identifications will be printed.

Ruler(k1, k2, k3 ...) k:{ThreadName | SegLevel | SegId | Sequence | DayTime | CpuTime | TimeDiff | CompId | ThreadId | OwnerId | EntryLevel | PrinterId }

[as same as k]

SegId

This command is valid for PS segment identifiers only. It specifies the fields of segment identification used as rulers inside the segment id area of printout.

There are the same conventions valid like above but each keyword must not occur twice ore more.

SegId(k1, k2, k3 ...) k:{SegCompId | SegModName |
SegFctName | SegThis |
SegFctDepth | SegText
|SegFctAndText }

[as same as k]

SegCompIdWd

SegModNamWd
SegFctNamWd

SegThisWd

SegFctDepthWd
SegTextWd

SegFctAndTextWd

This commands specifies the width of corresponding ruler fields of segment identification (above). They will be ignored if no SegId command is set.

SegCompIdWd(n)	n: width in number of characters [2]
SegModNamWd(n)	n: width in number of characters [10]
SegFctNamWd(n)	n: width in number of characters [10]
SegThisWd(n)	n: width in number of characters [8]
SegFctDepthWd(n)	n: width in number of characters [2]
SegTextWd(n)	n: width in number of characters [10]
SegFctAndTextWd(n)	n: width in number of characters [10]

SegColumns

Specifies the width of a single column in column mode. In column mode the segment identifiers will be printed as a column on a thread dependent position. For LI segment identifications the whole identifier will be handled. For PS the segment identification text will be handled only. In case of PS the SegColumns command will be ignored if

no SegText keyword appears in command SegId. In any case the rulers left and right of segment identification area will go on.

The occurrence of command SegColumns switches on the column

mode. On default the column mode is off..

SegColumns(n) n: numerical value determining column width

ThreadDiv

Turns on the thread diversification function. Each thread will be printed out to its own output file. Output file names will be derived from PSTTHNAM field of header portion. The main output file contains a mapping table showing the occurred threads and the name of files the thread was mapped to.

ThreadDiv doesn't work in common with SegColumns mode!

ThreadDiv(s) s:{ON|OFF} [OFF]

Truncate

Limits the field size of segment id fields strict to default size or the user specified size (see SegCompIdWd,SegModNamWd, SegFctNamWD etc.). If the required length exceeds the field size the content of field will be truncated without any comments. Think twice before using this command!

Truncate doesn't work in common with SegColumns mode!

Truncate(s) s:{ON|OFF} [OFF]

Width

Specifies the starting line width (in number of characters)

Width(n) [200]

d) Sample parameter file

```
/*
PST Trace Formatter, Debug V2.13
(c) Copyright Oce Printing Systems GmbH 2000

Default command set patterns. Feel free to edit for customizing!
*/


/******** Filter commands **********
 Don't forget to replace the comments inside the braces
 through appropriate arguments! */
```

```
/*
CompId   ( /* arguments */ )
Date     ( /* arguments */ )
CpuTime  ( /* arguments */ )
DayTime  ( /* arguments */ )
TimeDiff  ( /* arguments */ )
EntryLevel ( /* arguments */ )
OsId     ( /* arguments */ )
OwnerId  ( /* arguments */ )
PrinterId ( /* arguments */ )
ThreadName ( /* arguments */ )
RecordId  ( /* arguments */ )
Sequence  ( /* arguments */ )
SegLevel  ( /* arguments */ )
SegCompId ( /* arguments */ )
SegModName ( /* arguments */ )
SegFctName ( /* arguments */ )
SegThis  ( /* arguments */ )
SegFctDepth( /* arguments */ )
SegText  ( /* arguments */ )
*/
```

```
/******** Format commands **********/

/* Force printout as hex listing without formatting */
/*
FileDump( ON )
*/

/* Length of line in printout */
Width( 200 )

/* Ruler using fields of header portion */
Ruler( ThreadName,
```

```
SegLevel,
SegId,
Sequence,
DayTime,
CpuTime,
TimeDiff,
CompId,
ThreadId,
OwnerId,
EntryLevel,
PrinterId )

/* Gap among rulers */
Gap(' | ')

/* Include segment data */
Data( ON )

/* Specify encoding standard */
DataCode( ASCII )

/* Segment identification area fields; valid for PS */
SegId( SegCompId,
SegModName,
SegFctName,
SegThis,
SegFctDepth,
SegText,
SegFctAndText )

/* Length definition of segment id fields; valid for PS */
SegCompId  ( 2 )
SegModName  ( 10 )
SegFctName  ( 10 )
```

```
SegThis    ( 8 )
SegFctDepth ( 2 )
SegText    ( 10 )
SegFctAndText( 10 )

/* Truncate field sizes to default or user specified values (see
above).*/
/*
Truncate( ON )
*/

/* Turn on automatic function depth indent due to defining a character
string */
AutoIndent( '..' )

/* Turn printout the segment id text as columns on due to defining a
column width */
/*
SegColumns( 20 )
*/

/* Print out unformatted data in case of error due to inconsistent data
*/
/*
HexOnError( ON )
*/

/* Print out each thread in its own file */
/*
ThreadDiv( ON )
*/
```

- **MESSAGES**

Message	Comment	return

		code (dec)
No error occurred	No error but status message	0
Unable to open file <filename>	Formatter is unable to open file <filename>. Check filename and path.	2
Unable to read file <filename>	There has an read error occurred on file <filename>.	3
Unable to write file <filename>	There has an write error occurred on file <filename>.	4
Unexpected end of file <filename>	The Formatter has tried to read an input record from <filename> but the end of file was reached before record was complete read. The input file seems to be corrupt.	5
Inconsistent data <filename>	The Formatter has tried to read an input record from <filename> but the end of record didn't match to the expected end of block. The input file seems to be corrupt.	6
Not enough memory <procedure>	On processing <procedure> trying to allocate memory from heap has failed.	7
Invalid number of command line parameters <command>	There has an syntax error occurred parsing the Formatter starting command line.	8
Invalid command >><command> >><argument><<	Within the command <command> an error was found. Possibly <argument> was invalid. Check case sensitivity!	9
No parameter file specified	Warning message. If no command file was specified as third command line parameter on starting Formatter this message will be issued.	no abort
Unsupported format of trace file, possibly V8 trace	Formatter has found an input file not containing PST format specification.	11
Invalid file name	Formatter has detected a file name that doesn't meet the file name conventions of the according operating environment. (Not under MVS)	12
Duplicate file name	There was a duplicate filename at starting command line detected.	13

	(Not under MVS)	
Unexpected sequence number	Warning Message. Each record header block owns a ascending sequence number (field PSTSEQ) with an increment of one. This warning message shows an irregularity of number series.	no abort
Record dates of input files seem to be different, using newer file	Warning Message. Formatter tries to concatenate both input files. If it is obvious that both files don’t belong together the older file will not be handled.	no abort
Empty file has been found, trying to skip over <source file>	Warning Message Though Formatter was able to open the source file but it was empty. In this case the Formatter tries to continue with other source file.	no abort
Unable to locate expected ddname, trying to skip over <ddname>	Warning Message Formatter couldn’t find the expected ddname of source data set. In this case the Formatter tries to continue with other ddname. (Under MVS only)	no abort
No valid source file was found	Formatter was not able to find or to process a source file.	18

7 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*.

consolidate. To combine two or more entities into one whole. The Mail Piece Consolidation process in CIS combines two or more mail pieces with the same addressee into a larger one.

control file. In PRISMAproduction Host, a text file which contains one or more keywords that specify the kind of processing being requested and provide all the necessary parameters for this processing.

Conversion, Indexing and Sorting facility (CIS). A data manipulation tool part of the PRISMAproduction product family.

D

document. (1) In AFP, a data stream object delimited by a Begin Document structured field and End Document structured field. (2) A collection of one or more sheet groups, sheets, page groups and pages.

E

entity. That which is perceived or known or inferred to have its own physical existence. In CIS, an item which can be used for operations such as data enrichment, input selection and output segmentation. May be a file, document, page group, sheet, page or byte.

F

file. See *print file*.

I

Intelligent Printer Data Stream (IPDS). An architected host-to-printer data stream that contains both data and controls defining how the data is to be presented, how the printer informs the host about the progress of the printing process and that gives the host control over the error recovery actions.

IPDS. See *Intelligent Printer Data Stream*.

K

keyword. In CIS, a reserved word in the input Control File which is used as a reference point for finding other information, and which is followed by text enclosed in parenthesis. Such a text is made of sub-keywords and parameters.

M

mail piece. (1) In CIS, a generic term used to refer to a sheet group that has the same addressee for all its sheets. (2) The "top level" page group(s) in a print file. It is delimited by a Begin Named Page Group structured field and End Named Page Group structured field. Must coincide with the begin of a new sheet.

media collection. In AFP, the collection of media or collection of sheets generated by a Medium Map under the control of the *medium-map-level sheet collection* boundary condition of the Medium Finishing Control structured field.

Mixed Object Document Contents Architecture (MO:DCA). An architected, device-independent data stream for interchanging documents.

MO:DCA. See *Mixed Object Document Contents Architecture*.

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

N

normalize. To return to a normal, usual or generally accepted condition. The AFP normalizer in PRISMAproduction 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.

P

page. (1) A data stream object delimited by a Begin Page structured field and an End Page structured field. A page can contain presentation data such as text, image, graphics, and bar code data. (2) The final representation of such an object in a physical medium. One or more pages may be presented in a sheet-side.

page group. A named group of sequential pages. A page group is delimited by a Begin Named Page Group structured field and an End Named Page Group structured field. A page group may contain nested page groups. All pages in the page group inherit the attributes and processing characteristics that are assigned to the page group.

parameter. In CIS, an item contained in keywords and sub-keywords that specifies an operation to be performed or type of data is used during processing.

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.

print file. A disk file containing information that was created with the primarily purpose of having it printed as a unit composed of a collection of documents, sheet groups and sheets.

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 (e.g. overhead foils).

sheet group. (1) In CIS, a collection of sheets, normally more than one, with one or more common attributes such as layout, size, index, addressee, finishing operations, etc., delimited by a Begin Named Page Group structured field and an End Named Page Group Structured field. A sheet group may contain nested sheet groups. (2) See also *media collection*.

sheetlet. One of the two halves in which an IPDS printer logically divides a continuous form sheet, parallel to the carrier strips, when operating in X2UP or CSE mode. The two portions of the physical media are treated as if they were two separate pieces of cut-sheet media.

side. A physical surface of a sheet. May be *front* or *back*.

simplify. Make less complicated or less difficult. The optional AFP simplifier in CIS 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.

Smart Print Subsystem (SPS). A high performance printing subsystem that drives Océ and other IPDS compatible printers under OS/390 and z/OS. Runs as a Functional Subsystem under the control of JES.

SPS. See Smart Print Subsystem.

sub-keyword. In CIS, similar to keyword, except that it is part of the text in parenthesis that follows a keyword. A sub-keyword is also followed by text which is enclosed in parenthesis. Such a text is made of parameters.

8 Bibliography

[1]

[2]

[3]

[4]

[5]

[6]

[7] Data Stream and Object Architectures:
Font Object Content Architecture Reference
IBM publication number **S544-3285-04** (FOCA)

[8] **Using OpenType Fonts in an AFP System**
IBM publication number **G544-5876-02**

[9] **IPDS Extension: TrueType/OpenType Fonts**
IBM publication number **S544-3417-06**

[10] **TrueType Font Files Technical Specification**
Microsoft Corporation(Microsoft web site)

[11] **TrueType Reference Manual**
Apple Computer, Inc. (Apple web site)

[12] Data Stream and Object Architectures:
Bar Code Object Content Architecture Reference (BCOCA)
IBM publication number **S544-3766-05**

[13] **Océ** Electronic Printing Systems **Barcode Printing**
U2495-J-Z249-6-7600

[14] Data Stream and Object Architectures:
Graphic Objects Content Architecture Reference (GOCA)
IBM publication number **S544-5498-01**

[15] Data Stream and Object Architectures:
Image Object Content Architecture Reference (IOCA)
IBM publication number **SC31-6805-05**

[16] Data Stream and Object Architectures:
Presentation Text Object Content Architecture Reference (PTOCA)
IBM publication number **SC31-6803-02**

[17] Advanced Function Presentation
Programming Guide and Line Data Reference
IBM publication number **S544-3884-03**
including the architecture addendum's:
san0092 AFP Line Data Extension: **2D Bar Code** Support, Mar. 2003
san0020 AFP Line Data Extension: **XML Data** processing, Mar. 2003

[18] Data Stream and Object Architectures:
Mixed Object Document Content Architecture Reference (MO::DCA)
IBM publication number SC31-6802-06,
including the architecture addendum's:
MO:DCA Extension: **Media and Finishing Fidelity**, Oct. 2002
MO:DCA Extension: **UP3i Finishing Conduit**, Oct. 2002

Universal Printer-, Pre- and Post-processing Interface V1.20,
UP³I Core Group
www.up3i.org

[19] The **Unicode Standard** Version 3.0,
The Unicode Consortium
www.unicode.org

[20] SPDS Printer Data Stream Model 4
Reference Manual
U9737-J-Z247-X-7600

Index

—E—

—F—

—G—

—H—

—I—

—J—

—L—

—M—

—N—

—O—

—P—

—R—

—S—

—T—

—U—

—V—

—W—

—X—

—Y—