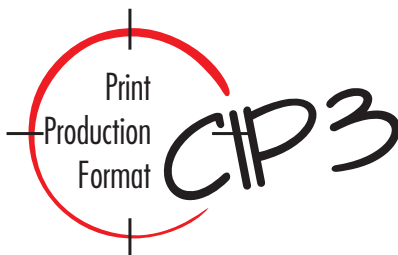


International Cooperation  
for Integration of  
Prepress, Press, and Postpress

# Specification of the **CIP3** Print Production Format

Version 2.1  
May 27, 1997



Stefan Daun  
Georg Lucas  
Jürgen Schönhut

FRAUNHOFER INSTITUTE FOR COMPUTER GRAPHICS (FRAUNHOFER-IGD)  
Document Imaging Department  
Rundeturmstraße 6, D - 64283 Darmstadt  
Phone: +49 - 6151 -155 220  
Fax: +49 - 6151 - 155 299

*Please send any comment on this document to Stefan Daun  
Email: [daun@igd.fhg.de](mailto:daun@igd.fhg.de) — Fax: +49 - 6151 - 155 299*

**97i002-FIGD**

The CIP3 Print Production Format was specified by the Fraunhofer Institute for Computer Graphics in close collaboration with a group of currently 26 well-known manufacturers. This group is called "International Cooperation for Integration of Prepress, Press, and Postpress" (abbreviated as CIP3). The following companies are members of this group:

- Adobe
- Agfa
- Baldwin Technology Company
- Barco Graphics
- Creo
- Fujifilm Electronic Imaging
- Ekotrading-Inkflow
- Eltromat Polygraph
- Ewert Ahrensburg Electronic
- Goebel
- Harlequin
- Heidelberg Druckmaschinen
- Koenig & Bauer - Albert AG
- Kolbus
- Komori
- Linotype-Hell
- MAN Roland
- Mitsubishi Heavy Industries
- Müller Martini
- Polar-Mohr
- RR Donnelley & Sons Company
- Scitex
- Screen
- Ultimate Technographics
- Wohlenberg
- Xerox

This document can be obtained from the Fraunhofer Institute for Computer Graphics in Darmstadt in several ways:

1. As Paper Copy

Send a fax order to Stefan Daun (fax: +49 / 6151 / 155 299) to get a paper copy of the document. You will then be charged a token fee of 100,- DM.

2. Via FTP

Connect to FTP server *ftp.igd.fhg.de*, log in as user *anonymous* (using your mail address as your password). Then change to the *outgoing/igd\_a1/cip3* directory (please note the underline char and the digit one in "igd\_a1"), where you can fetch (in binary mode) the document either as a PostScript (*cip3.ps*) or as a PDF file (*cip3.pdf*). Fetching the electronic document is free.

3. Via WWW

In the World Wide Web you can find more information about the CIP3 Print Production Format (including the specification document) at the following address:

<http://www.igd.fhg.de/www/igd-a1/cip3>

Please note the minus sign and the digit one in "igd-a1". Fetching these documents is also free.

## Content

<b>1.</b>	<b>Introduction .....</b>	<b>4</b>
<b>2.</b>	<b>An Overview of the CIP3 Print Production Format .....</b>	<b>5</b>
	2.1. Logical Structure .....	5
	2.2. Attributes .....	6
	2.3. Content .....	6
	2.4. Embedding in PostScript .....	7
	2.5. Usage of the CIP3 Print Production Format .....	7
<b>3.</b>	<b>Specification .....</b>	<b>9</b>
	3.1. Specification Concepts .....	9
	3.1.1. The CIP3 File Structure .....	9
	3.1.2. Syntax .....	10
	3.1.3. Coordinate System .....	12
	3.1.4. Defining the Logical Structure .....	13
	3.1.5. What is a Valid CIP3 PPF File? .....	16
	3.2. PPF Directory .....	17
	3.3. Administration Data .....	19
	3.4. Continuous Tone Image in Reduced Resolution .....	23
	3.4.1. Composite Preview Image .....	24
	3.4.2. Preview Image with Separations .....	24
	3.4.3. Sequence of Image Data .....	26
	3.4.4. Rules for the Generation of the Preview Image .....	27
	3.4.5. Using Image Encoding and Compression .....	29
	3.5. Characteristic Curves for Transfer .....	30
	3.6. Register Marks .....	32
	3.7. Color and Ink Control .....	33
	3.8. Cutting Data .....	38
	3.9. Folding Data .....	43
	3.10. Comments and Annotations .....	46
	3.11. Private Data .....	47
	3.12. Private Content .....	48
	<b>Appendix A: Changes from CIP3 PPF version 2.0 to version 2.1 .....</b>	<b>49</b>
	<b>Appendix B: Changes from CIP3 PPF version 1.0 to version 2.0 .....</b>	<b>51</b>

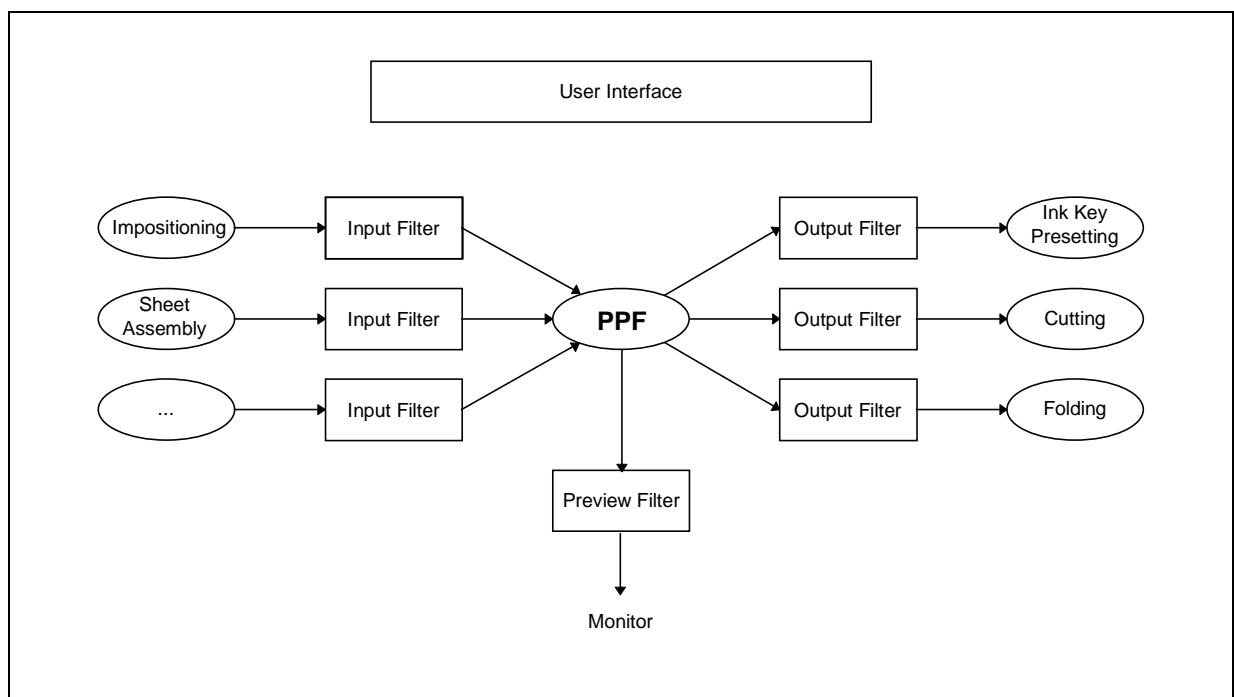
<b>Appendix C: Example of a CIP3 PPF file .....</b>	<b>52</b>
<b>Appendix D: References .....</b>	<b>56</b>
<b>Appendix E: List of Illustrations .....</b>	<b>57</b>
<b>Appendix F: List of Tables .....</b>	<b>58</b>
<b>Appendix G: Index of Key Words and Comments .....</b>	<b>59</b>

## 1. Introduction

The starting point for the development of the CIP3 Print Production Format (CIP3) was the desire to link printing and the post-printing processes closer to the prepress phase. The various data which is generated in prepress should be compiled to enable a uniform administration and further processing. In this way a repeated, time-consuming data acquisition in the subsequent processing steps should be avoided, because this data had already been available in the prepress phase.

In the first two parts of the study various alternatives for the Print Production Format were investigated. Above all existing standards or rather the de-facto standards were taken into account. It turned out that the formats PostScript and TIFF, which play an important part in the connected sectors, are best suited for a realization of the CIP3. Finally PostScript was chosen as the basic format for the CIP3 format, due to the fact that it is more flexible and can be extended more easily.

In the present report this favorable solution on the basis of PostScript will be presented and the Print Production Format will be specified. With this the format is disclosed and can build the interface to other systems.



**Illustration 1-1: CIP3-Scenario**

*This illustration shows the conceptual relationship between the components and should not be seen as a description of an implementation.*

## 2. An Overview of the CIP3 Print Production Format

In the ideal case the CIP3 print production format contains all necessary data for print and for the further processing of a job, except for the data screened in high resolution for the printing.

To allow for easy and flexible access the information of the CIP3 format is stored in three ways:

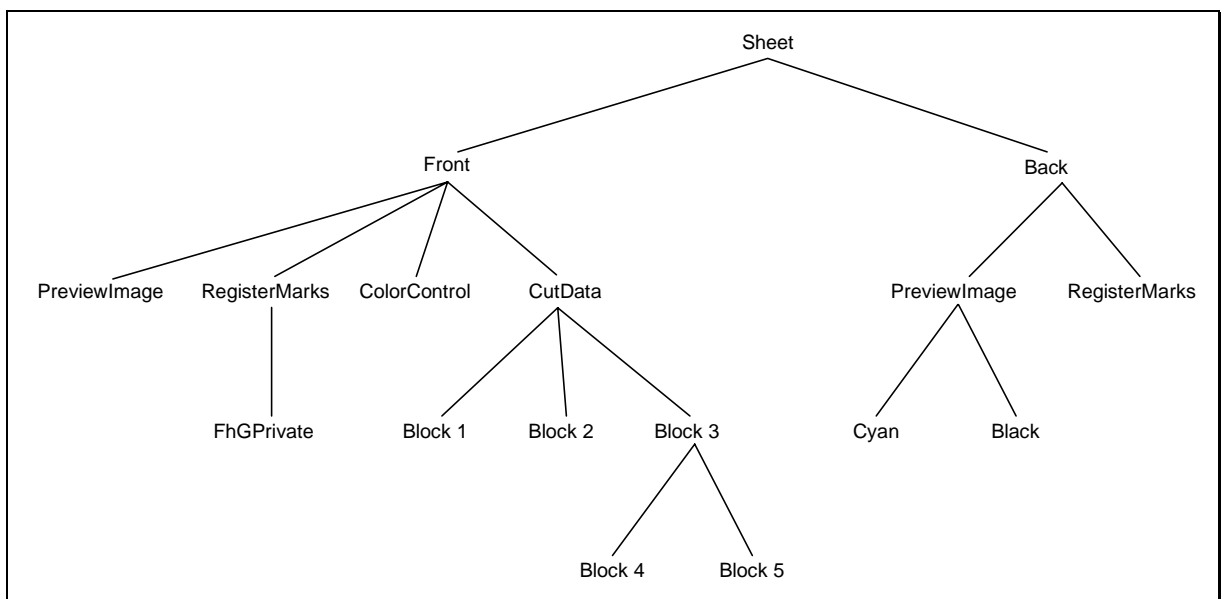
- Structure
- Attributes
- Content

In the following three chapters these elements are described in more detail.

### 2.1. Logical Structure

Instead of storing the content of the CIP3 in a flat file, it was decided to provide a structured way of storing the information. Although this prevents the format from being read by a simple "grep" program, this method allows for much more flexibility.

The structure of the CIP3 builds a tree-like "**part-of-hierarchy**", in which each subelement is a part of its parent element. If, for example, in the real world a sheet consists of a front and a back, then the corresponding CIP3 document contains a sheet structure, which in turn contains a front and a back structure.



**Illustration 2-1: Example of the hierarchical structure of a CIP3 file**

The structure of the CIP3 file is used to establish an **inheritance mechanism**. By this mechanism an element can inherit attributes from its (direct or indirect) parent elements. In

every structure element new attributes can be defined or existing attributes can be overwritten.

It is important that only attributes can be inherited. Content portions (like a register mark) cannot be inherited.

Coordinates specified in a structure element are always relative to the origin of this element.

## 2.2. Attributes

Within the CIP3 format attributes are used to store information that characterizes the job. To define an attribute it is necessary to specify the attribute name and its value. Through the definition the attribute is bound to the structure element in which the definition took place. The attribute value is then valid in the structure element, in which the attribute is defined, and in all (direct or indirect) subelements, unless it is overwritten. Attributes can be optional or necessary.

The following data is stored in attributes:

- **Information for administration**  
The name of the sheet creator, date and time of the creation, software with which the sheet has been created etc., are included here.
- **Data for calculating the ink consumption**  
Here it is possible to store two characteristic curves of the transfer, one for the process of copying the data onto film, and one for copying data onto the printing plate.
- **Folding data**  
Data which describe the position of the sheet and the sequence of the folding procedure. This data can be bound to a cut block.

## 2.3. Content

For the first realization the following content types are planned:

- **Continuous tone image with reduced resolution**  
Display of the sheet to be printed with e.g. 1280 \* 1024 pixels and 8 Bit depth per color separation. Above all this picture is used for display. By this means it should be possible to check the completeness of the print job and (at least roughly) the position of the cutting and folding edges etc. Moreover, it is used for calculating the ink consumption. For the first approach we have in mind the storage of the separation in CMYK. For the future the CIP3 format will also support composite color. However, at a later stage the use of CIE-L\*a\*b\* might be possible. Then the continuous tone image could be available in either only one of the two types or even in both types simultaneously.
- **Register marks**  
Data which describes the position and type of register marks.

- **Color and ink control fields**  
Data for the description of measuring strips with a number of test fields for the quality measurement. Two basic types of measuring fields are supported: color measuring fields and density measuring fields. Control strips can be build up from these two basic fields.
- **Cutting data**  
Data which describe the cutting blocks.
- **Comments**  
Any comments.

## 2.4. Embedding in PostScript

PostScript is used as the basic format for the CIP3 print production format.

As far as sensible all data which should appear on the paper during a test print is directly coded in PostScript with CIP3.

All **structure elements** of the CIP3 are enclosed by a pair of "CIP3Begin..." and "CIP3End...". In order to build the "part-of-hierarchy" it is necessary to nest the structure elements. Each structure element is represented by a PostScript dictionary , which stores all attributes defined in the element.

```
CIP3BeginStructure1
<attribute definitions and content of Structure1>
CIP3BeginStructure2
<attribute definitions and content of Structure2>
CIP3EndStructure2
<more attribute definitions and content of Structure1>
CIP3EndStructure1
```

An **attribute** is defined by use of the PostScript "**def**" operator. Therefore the specification of the name and the value of the attribute is required:

```
/attribute-name value def
```

For more information about the embedding in PostScript see chapter 3.1 "Specification Concepts".

## 2.5. Usage of the CIP3 Print Production Format

The information stored in a CIP3 file can be used in three ways:

1. It can be parsed by a PostScript interpreter to extract specific information. This is the main purpose of the CIP3 print production format. In the first realization it is planned to extract the following information from the CIP3 file:
  - The image data and the transfer curves can be used to calculate the ink consumption and profile and to preset the printing press.



- The cutting data can be used to automatically generate programs for a cutting device.
  - The folding data can be used to automatically generate programs to control or preset a folding device.
2. It can be printed on a normal PostScript printer. This requires an appropriate PostScript prolog. Depending on whether both front and back or only one side of the sheet is specified in the file, one or two pages will be printed. They can be used for documentation purposes or to roughly check the completeness of the print job.
  3. It can be parsed by a PostScript interpreter to produce a description of the job. This can be a page containing administration data etc.

It is important that a PostScript interpreter is necessary to parse the CIP3 file. This is mainly due to the fact that the information is stored in the CIP3 file in a hierarchically structured way. But, in order to reduce the complexity of the parsing process, some restrictions in using the PostScript language have been included in the CIP3 PPF specification (see chapter 3.1.2).

### 3. Specification

In the following the Print Production Format has been specified. The following notation is valid:

<i>Courier normal</i> :	Content of a CIP3 file. The text given in this way must appear in the PPF file in the same way.
<i>Courier italic</i> :	Text given in this way must be replaced by a corresponding figure, name or similar.
< <i>Courier</i> >:	At this place the information described in conical brackets (< >) must be supplied.
...:	Omission sign.

### 3.1. Specification Concepts

#### 3.1.1. The CIP3 File Structure

A CIP3 file is structured in a two line heading, the body (containing structures, attributes and content), and an End-Of-File line. The first two lines serve for a definite detection of a CIP3 file and they must look like this:

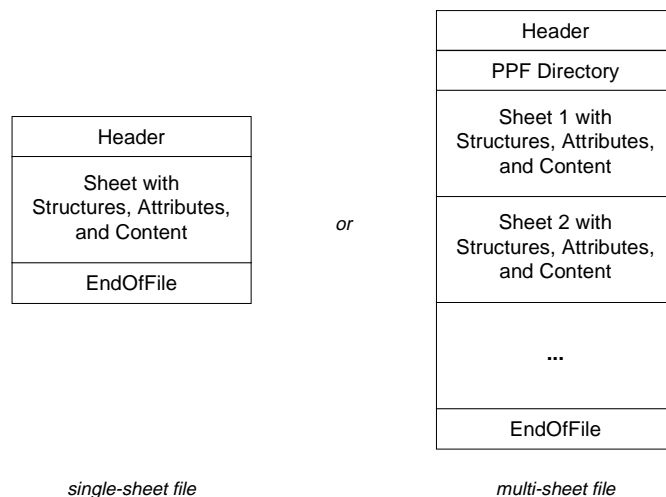
```

%!PS-Adobe-3.0
%%CIP3-File Version 2.1

```

At the beginning of the lines in front of the percent characters no other characters (such as blanks or tabulators) may appear. The last line serves for the control of the complete transmission of a CIP3 file and must always look like this:

```
%%CIP3EndOfFile
```



**Illustration 3-1: Syntactical structure of the CIP3 file**

### 3.1.2. Syntax

For the syntax of the CIP3 format, i.e. especially for the coding of numbers, names, character strings etc. the rules of PostScript are valid. Upper/lower case of names and keywords must be observed. The following table shows the PostScript data types (see chapters 3.2 and 3.3 of the PostScript Language Reference Manual) which may be used within CIP3:

Data Type	Description	Examples
boolean	A boolean value can be "true" or "false".	true false
integer	An integer object represents mathematical integers within an implementation dependent interval centered at zero. A typical range is $-2^{31}$ to $2^{31}-1$ . An integer consists of an optional sign followed by one or more decimal digits.	-12 0 +345 1344
real	A real object represents a mathematical real number within a much larger interval but with limited precision. A real consists of an optional sign followed by one or more decimal digits with an embedded period (decimal point), a trailing exponent, or both. The exponent, if present, consist of "E" or "e" followed by an optional sign and one or more decimal digits.	-345. -3.62 -.01 0.0 4E-2 34.5 155.6e17
name	A name object can consist of any character except delimiters ("(", ")", "<", ">", "[", "]", "{", "}", "/", "%") and white space (null, tab, line-feed, form-feed, carriage return, space) that cannot be interpreted as a number. A "/" introduces a literal name. The slash is not part of the name itself, but is a prefix indicating that the following name is literal. The maximum length of a name is 127 characters.	ANTON printing 5b a-z www.igd /Left /Binary /CutBlock
string	A string consist of up to 65535 characters enclosed in "(" and ")". Any characters may appear in the string other than "(", ")", and "\", which must be treated specially. Balanced pairs of parentheses in the string require no special treatment. Within a string, the backslash character "\" is treated as an escape character: <ul style="list-style-type: none"> <li>\n line-feed or (newline)</li> <li>\r carriage return (CR)</li> <li>\t horizontal tab</li> <li>\b backspace</li> <li>\f form-feed</li> <li>\\ backslash</li> <li>\( left parenthesis</li> <li>\) right parenthesis</li> <li>\ddd character code <i>ddd</i> (octal)</li> </ul>	(This is a string) ( ) (This is (OK)) (This \ (too\)) (a line\n)

array	An array is enclosed in "[" and "]". Each element of the array may be of a different data type (from the list of this table). Arrays may be nested. The maximum number of elements in an array is 65535.	[ 1 2 3 ] [/Left 23.0 true] [[1] [] ] [(str) <</a 5>>]
dict	A dictionary is enclosed in "<<" and ">>" and consists of key-value pairs. Within the CIP3 PPF the key must always be a literal name, while the value can be of one of the data types listed in this table. Dictionaries may be nested. The maximum capacity of a dictionary is 65535 key-value pairs.	<< /5b 5.6 /Date (16.01.97) /arr [1 (str)] << /a 0.2 >> >>

**Table 3-1: List of PostScript data types allowed in CIP3 PPF**

Any occurrence of the "%" character outside a string introduces a comment. The comment consists of all characters between the "%" and the next newline or form-feed character.

The following rule is established in order to reduce the complexity of the CIP3 format:

***A CIP3 file should contain nothing else executable in the CIP3 file body except what is defined in the specification (comments are allowed).***

In particular this means, that it is not allowed to calculate the values of an attribute by means of the PostScript language (e.g. by using operators like mul or add). In addition the implicit version of specifying arrays (using "[" and "]") and dictionaries (using "<<" and ">>") must be used.

The following table lists all PostScript operators that may be used within a CIP3 PPF file. Please note that since version 2.1 of the CIP3 PPF the PostScript operators "begin", "end", and "dict" are no longer supported.

PS Operator	Meaning
[	begin array
]	end array
<<	begin dictionary
>>	end dictionary
def	associate name with value

**Table 3-2: List of PostScript operators allowed in CIP3 PPF**

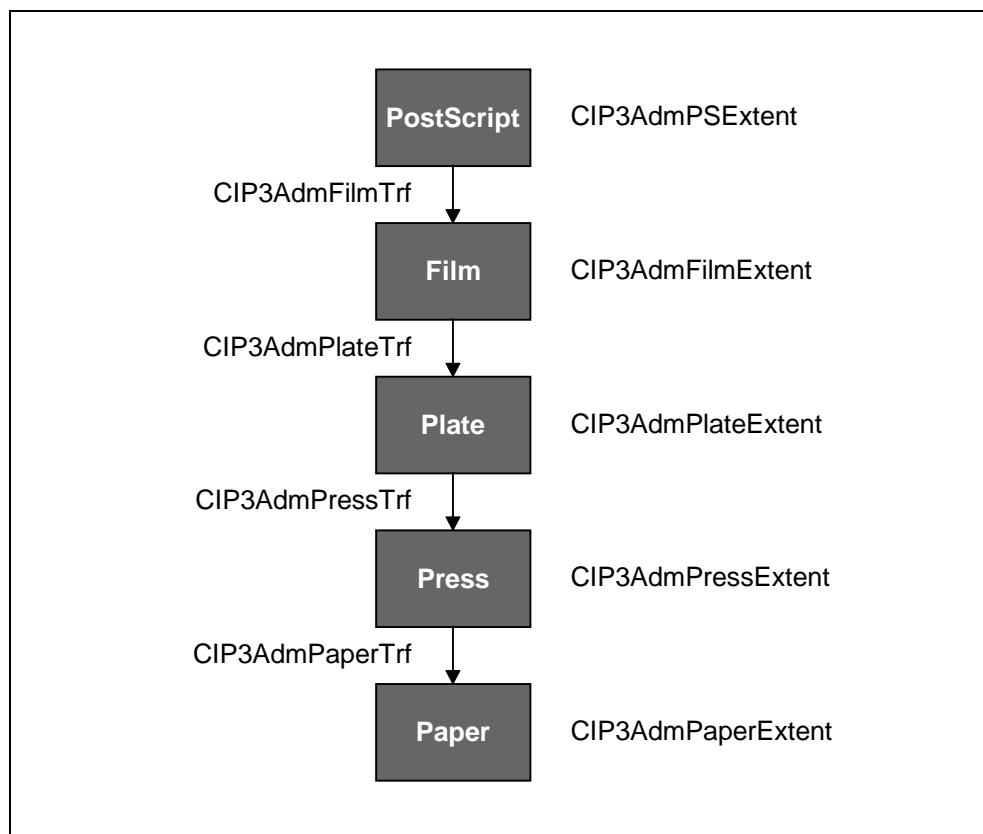
To ensure an unrestricted transmission of CIP3 files, e.g. via any network connections, it is strongly recommended to code all data with the 7-Bit ASCII Code without using the control characters (0x00 to 0x1F). Only the control characters line-feed (0x0A), carriage return (0x0D) and the tabulator (0x09) should be used.

### 3.1.3. Coordinate System

The default coordinate system of PostScript is used as the coordinate system for the CIP3 format. The origin of this system is in the bottom left corner (of the PostScript coordinate system, see below), the unit is 1/72 inches in both directions of axis.

For the different prepress, press, and postpress processes that are encountered during the life cycle of a CIP3 file it is required to map the coordinates of the PostScript coordinate system to other coordinate systems. Therefore transformation matrices can be defined within CIP3, that specify the mapping between two coordinate systems. Furthermore the extent of each coordinate system can be defined.

The default value for an unspecified transformation matrix is the identity transformation. The default value for an unspecified extent is the value inherited from the source coordinate system. Only the specification of the CIP3AdmPSExtent is required.



**Illustration 3-2: Coordinate Systems and Coordinate Mapping**

In addition to the native PostScript units it is also possible to use predefined measuring units. If a number is stated without unit, then 1/72 inch is the accepted unit. The measurements listed in the following can be used, by stating the number and - separated by a blank - the abbreviation of the measuring unit.

Measuring unit	Abbreviation	Conversion factor
Millimeter	mm	72/25.4
Centimeter	cm	72/2.54
Inch	inch	72
Point	point	1

**Table 3-3: Units in the CIP3 format**

The *point* unit mentioned above is the PostScript point, which is defined as the 72th part of an inch.

### 3.1.4. Defining the Logical Structure

The logical structure of the CIP3 file is created by PostScript commands of the form "**CIP3BeginUnit**" and "**CIP3EndUnit**". In general *unit* specifies both type and name of the structure. But since more than one Separation, CutBlock, or PrivateData structure may exist at the same hierarchy level, it is necessary to allow for user defined names for these structures. All substructures of a given structures must have distinct names.

Definition	Structure Type	Structure Name
CIP3BeginSheet CIP3EndSheet	Sheet	Sheet
CIP3BeginFront CIP3EndFront	Front	Front
CIP3BeginBack CIP3EndBack	Back	Back
CIP3BeginPreviewImage CIP3EndPreviewImage	PreviewImage	PreviewImage
CIP3BeginSeparation CIP3EndSeparation	Separation	<i>implicitly inherited from the CIP3AdmSeparationNames attribute defined in the Sheet, Front, or Back structure</i>
CIP3BeginRegisterMarks CIP3EndRegisterMarks	RegisterMarks	RegisterMarks
CIP3BeginColorControl CIP3EndColorControl	ColorControl	ColorControl

CIP3BeginCutData CIP3EndCutData	CutData	CutData
CIP3BeginCutBlock CIP3EndCutBlock	CutBlock	<i>inherited from the CIP3BlockName attribute specified inside the structure</i>
CIP3BeginFoldProcedures CIP3EndFoldProcedures	FoldProcedures	FoldProcedures
<i>name</i> CIP3BeginPrivate CIP3EndPrivate	PrivateData	<i>name</i>

**Table 3-4: Structure types and names**

### Definition of a valid structure tree in a CIP3 file

The rules shown below (in a Backus-Naur like notation) specify how to build a valid logical structure for a CIP3 file.

```

Sheet      ::= Front?, Back?.
Front      ::= PreviewImage?, RegisterMarks?, ColorControl?,
             CutData?, FoldProcedures?.
Back       ::= PreviewImage?, RegisterMarks?, ColorControl?,
             CutData?, FoldProcedures?.
PreviewImage ::= Separation*.
CutData    ::= CutBlock*.
CutBlock   ::= CutBlock*.

```

#### **Notation:**

```

item       : item must occur exactly once
item?      : item is optional
item+      : item can occur once or more
item*      : item can occur an arbitrary number of times or not at all
item-a, item-b : sequence of item-a followed by item-b

```

In addition a private data structure may occur within any other structure.

**Example: Definition of logical structure (see illustration 2-1)**

```
CIP3BeginSheet
  < ... attribute definitions for both front and back ... >
CIP3BeginFront
  < ... attribute definitions for front ... >
CIP3BeginPreviewImage
  < ... composite (CMYK) preview image of front ... >
CIP3EndPreviewImage
CIP3BeginRegisterMarks
  < ... placing of register marks onto front ... >
/FhGPrivate CIP3BeginPrivate
  < ... private data inside RegisterMarks structure ... >
CIP3EndPrivate
CIP3EndRegisterMarks
CIP3BeginColorControl
  < ... placing of color control fields onto front ... >
CIP3EndColorControl
CIP3BeginCutData
CIP3BeginCutBlock
  < ... cut block 1 ... >
CIP3EndCutBlock
CIP3BeginCutBlock
  < ... cut block 2 ... >
CIP3EndCutBlock
CIP3BeginCutBlock
  < ... cut block 3, containing two nested cut blocks ... >
CIP3BeginCutBlock
  < ... cut block 4 ... >
CIP3EndCutBlock
CIP3BeginCutBlock
  < ... cut block 5 ... >
CIP3EndCutBlock
CIP3EndCutBlock
CIP3EndCutData
CIP3EndFront
CIP3BeginBack
CIP3BeginPreviewImage
  < ... preview image of back with two separations ... >
CIP3BeginSeparation
  < ... 1st separation of back preview image... >
CIP3EndSeparation
CIP3BeginSeparation
  < ... 2nd separation of back preview image... >
CIP3EndSeparation
CIP3BeginRegisterMarks
  < ... placing of register marks onto back ... >
CIP3EndRegisterMarks
CIP3EndBack
CIP3EndSheet
```



### 3.1.5. What is a Valid CIP3 PPF File?

A valid CIP3 PPF file must meet the following requirements:

1. It must be syntactically correct, i.e. file structure and syntax must meet the specification.
2. The logical structure, i.e. the nesting of structures, must meet the specification.
3. In each structure, all required attributes must be defined. Due to the inheritance mechanism it is allowed to specify an attribute at a higher level.
4. Private attributes may be placed in any structure or dictionary, if they do not conflict with specified dict entries and if they do not use the CIP3 prefix.
5. The data type for the attribute must meet the specification. If an attribute contains a dictionary, all required entries must be defined.
6. CIP3 content may only occur in the appropriate structures listed below. Only CIP3 comments, CIP3 annotations and CIP3 Private Content may appear within any structure.
  - preview image data in PreviewImage and Separation structures
  - register marks in RegisterMarks structure
  - color or density measuring fields and color control strips in ColorControl structure
  - cut marks in CutData and CutBlock structures

### 3.2. PPF Directory

The **PPFDirectory** structure contains directory information about all single sheet PPF subfiles contained in the whole CIP3 PPF file. For each PPF subfile a **CIP3PPFDirEntry** command is required. It is possible to reserve directory entries for future adding of sheets by setting offset and length to zero. In order to allow for modifications of these entries the PPFDirectory structure uses a fixed length format with 256 bytes for each entry.

The PPFDirectory is an optional structure (see Illustration 3-1).

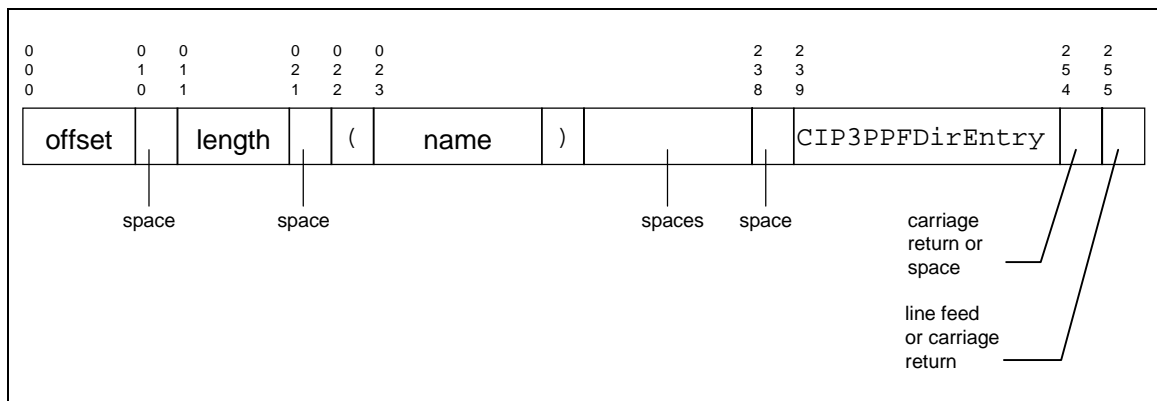


Illustration 3-3: Fixed size directory entry

Parameter	Offset	Length	Type	Necessary	Meaning
<b>offset</b>	0	10	integer	required	Address of the first byte of the single sheet PPF subfile within the complete CIP3 PPF file. A value of zero indicates a missing sheet. <i>The number must be coded in 10 byte fixed format using leading blanks or zeros.</i>
<b>length</b>	11	10	integer	required	Length in bytes of the single sheet PPF subfile. If the subfile is missing, the value zero must be used. <i>The number must be coded in 10 byte fixed format using leading blanks or zeros.</i>
<b>name</b>	22	216	string	required	Sheet name. This name should be copied from the CIP3AdmSheetName attribute as defined in the PPF subfile. <i>The string must be coded in up to 216 bytes (including the parentheses needed to build a PostScript string).</i>

Table 3-5: Parameter for the CIP3PPFDirEntry command

**Example: PPF Directory of a CIP3 PPF describing a brochure with 48 pages**

Please note that in the following example word "CIP3PPFDirEntry" starts in column 239 (assuming the first column is called column 0).

```
CIP3BeginPPFDirectory
0000003116 0000655193 (Cover = pages 1,2,43,44)           CIP3PPFDirEntry
0000658309 0000120080 (Pages 3,4,41,42)           CIP3PPFDirEntry
0000000000 0000000000 (Pages 5,6,39,40)           CIP3PPFDirEntry
0000000000 0000000000 (Pages 7,8,37,38)           CIP3PPFDirEntry
0000000000 0000000000 (Pages 9,10,35,36)          CIP3PPFDirEntry
0000000000 0000000000 (Pages 11,12,33,34)          CIP3PPFDirEntry
0000000000 0000000000 (Pages 13,14,31,32)          CIP3PPFDirEntry
0000000000 0000000000 (Pages 15,16,29,30)          CIP3PPFDirEntry
0000000000 0000000000 (Pages 17,18,27,28)          CIP3PPFDirEntry
0000000000 0000000000 (Pages 19,20,25,26)          CIP3PPFDirEntry
0000000000 0000000000 (Pages 21,22,23,24)          CIP3PPFDirEntry
0000000000 0000000000 (Insert = without page numbers)      CIP3PPFDirEntry
CIP3EndPPFDirectory
```

### 3.3. Administration Data

All administration data is stored in attributes. It should be defined in the *Sheet* structure or in the *Front* and *Back* structure.

Attribute Name	Type	Necessary	Meaning
<b>CIP3AdmJobName</b>	string	required	Job name
<b>CIP3AdmJobCode</b>	string	optional	Code used for job identification
<b>CIP3AdmMake</b>	string	optional	Exposure device make
<b>CIP3AdmModel</b>	string	optional	Exposure device model
<b>CIP3AdmSoftware</b>	string	optional	Sheet creating program
<b>CIP3AdmCreationTime</b>	string	optional	Date and time of sheet creation
<b>CIP3AdmArtist</b>	string	optional	Creator of the sheet
<b>CIP3AdmCopyright</b>	string	optional	Copyright
<b>CIP3AdmCustomer</b>	string	optional	Name of the target customer
<b>CIP3AdmSheetName</b>	string	optional	Name of the sheet
<b>CIP3AdmSheetLay</b>	name	optional	Guiding edge of the paper in the press seen in the direction of paper flow: <b>/Left</b> , <b>/Right</b>
<b>CIP3AdmPrintVolume</b>	integer	optional	Print volume
<b>CIP3AdmFilmType</b>	string	optional	Brand and type of film material
<b>CIP3AdmPlateType</b>	string	optional	Brand and type of plate material
<b>CIP3AdmPaperGrade</b>	string	optional	Type of paper
<b>CIP3AdmPaperGrammage</b>	float	optional	Paper grammage in g/m <sup>2</sup>
<b>CIP3AdmPaperThickness</b>	float	optional	Paper thickness
<b>CIP3AdmPaperColor</b>	array of 3 floats	optional	Paper color in CIE-L*a*b*
<b>CIP3AdmSeparationNames</b>	array of strings	required	Sequence of colors to be printed. A name of a separation may not contain a slash character.
<b>CIP3AdmInkInfo</b>	array of strings	optional	Information about used inks (e.g. exact name or manufacturer)
<b>CIP3AdmInkColors</b>	array of array of 3 floats	optional	Ink color specification in CIE-L*a*b*
<b>CIP3AdmTypeOfScreen</b>	string	optional	(amplitude modulated) or (frequency modulated)

<b>CIP3AdmPSExtent</b>	array of 2 floats	required	Extent of PostScript coordinate system
<b>CIP3AdmFilmTrf</b>	array of 6 floats	optional	Transformation matrix used to map coordinates from the PostScript coordinate system to the film coordinate system.
<b>CIP3AdmFilmExtent</b>	array of 2 floats	optional	Extent of film coordinate system
<b>CIP3AdmPlateTrf</b>	array of 6 floats	optional	Transformation matrix used to map coordinates from the film coordinate system to the plate coordinate system.
<b>CIP3AdmPlateExtent</b>	array of 2 floats	optional	Extent of plate coordinate system
<b>CIP3AdmPressTrf</b>	array of 6 floats	optional	Transformation matrix used to map coordinates from the plate coordinate system to the press coordinate system.
<b>CIP3AdmPressExtent</b>	array of 2 floats	optional	Extent of press coordinate system
<b>CIP3AdmPaperTrf</b>	array of 6 floats	optional	Transformation matrix used to map coordinates from the press coordinate system to the paper coordinate system.
<b>CIP3AdmPaperExtent</b>	array of 2 floats	optional	Extent of paper coordinate system

**Table 3-6: General administration data**

In addition to the general administration data there are some attributes that can be used to specify information used for web presses.

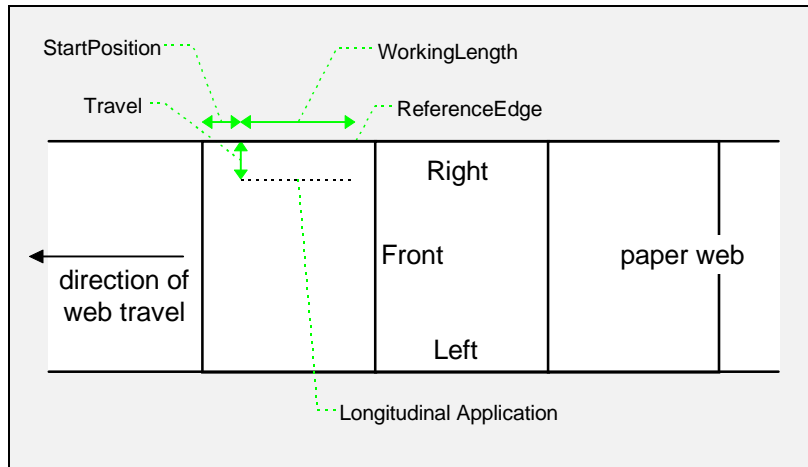
<b>Attribute Name</b>	<b>Type</b>	<b>Necessary</b>	<b>Meaning</b>
<b>CIP3AdmPaperSource</b>	name	optional	Input source of paper: <b>/Reel</b> or <b>/Sheet</b>
<b>CIP3AdmPaperDestination</b>	name	optional	Output destination of paper: <b>/Reel</b> , <b>/Sheet</b> or <b>/Folded</b>
<b>CIP3AdmTintingColorName</b>	string	optional	Name of tinting color
<b>CIP3AdmTintingColor</b>	array of 3 floats	optional	Tinting color specification in CIE-L*a*b*
<b>CIP3AdmReelWidth</b>	float	optional	Reel width (for rotary printing)
<b>CIP3AdmCylinderCircumference</b>	float	optional	Cylinder circumference (for rotary printing)
<b>CIP3AdmCarbonizingColorName</b>	string	optional	Name of color used for carbonizing
<b>CIP3AdmCoating</b>	string	optional	Type of coating

<b>CIP3AdmLongitudinalApp</b>	array of dicts	optional	Specification of longitudinal slitting, perforating, gluing, or stitching (see dictionary description below). This item is only used for reel processing.
<b>CIP3AdmCrossApp</b>	array of dicts	optional	Specification of cross cutting, perforating, or gluing (see dictionary description below). This item is only used for reel processing.

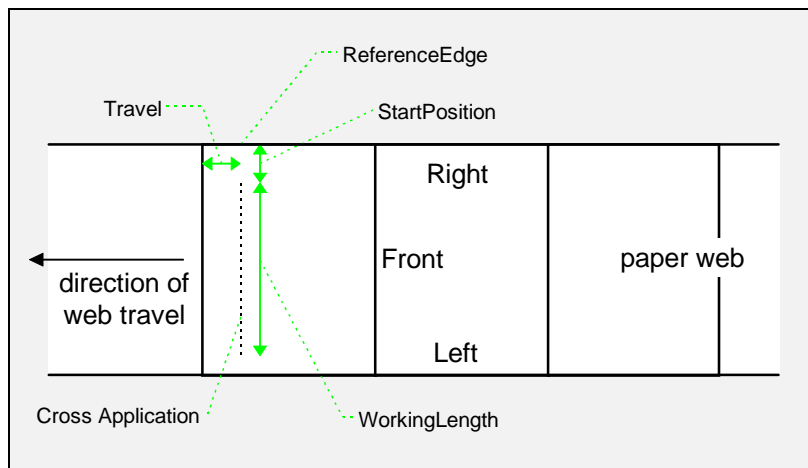
Table 3-7: Web press data

Key	Type	Necessary	Meaning
<b>/ReferenceEdge</b>	name	required	Reference edge: <b>/Left</b> or <b>/Right</b> (seen in direction of web travel)
<b>/Travel</b>	float	required	Position of tool in direction perpendicular to working path. For a longitudinal application the travel is the distance of the tool from the reference edge. For a cross application the travel is the distance of the tool from the front edge.
<b>/StartPosition</b>	float	required	Starting position of tool in direction of working path. For a longitudinal application the travel is the distance of the tool from the front edge. For a cross application the travel is the distance of the tool from the reference edge.
<b>/WorkingLength</b>	float	required	Length of working path. For a longitudinal application the working path is always parallel to the direction of web travel. For a cross application the working path is always parallel to the front edge)
<b>/Function</b>	name	required	Type of function: <b>/Cut</b> = slitting, <b>/Perforate</b> = perforating, <b>/Glue</b> = gluing, <b>/Stitch</b> = stitching

Table 3-8: Data dictionary for longitudinal and cross applications



**Illustration 3-4: Example of longitudinal application (view from top)**



**Illustration 3-5: Example of cross application (view from top)**

**Example:**

```

/CIP3AdmJobName (TestJob) def
/CIP3AdmMake (Prepress Company) def
/CIP3AdmModel (PC210) def
/CIP3AdmSoftware (The Imposition Program) def
/CIP3AdmCreationTime (1994:08:30 12:17:03) def
/CIP3AdmArtist (Stefan Daun) def
/CIP3AdmCopyright (Copyright by Fraunhofer-IGD, 1995) def
/CIP3AdmPSExtent [100 cm 70 cm] def
/CIP3AdmSheetLay Left def
/CIP3AdmPrintVolume 120000 def
/CIP3AdmPaperGrammage 130 def
/CIP3AdmSeparationNames [ (Black) (Cyan) (Magenta) (Yellow) ] def

```

### 3.4. Continuous Tone Image in Reduced Resolution

The **PreviewImage** structure includes the contents of the complete sheet as low resolution continuous tone image. If only the standard printing colors Cyan, Magenta, Yellow, and Black are used, it is possible to store the image as a composite CMYK image. If more or other colors are used, it is required to store separate images for each color separation. In this case it is also allowed to store a bitonal image of a higher resolution instead of a continuous tone image.

In order to reduce the size of the CIP3 PPF file, some of the encoding and compression techniques specified in PostScript can be applied to the image data. For a complete description please refer to the PostScript Language Reference Manual (also known as Red Book, chapter 3.13.3: Details of Individual Filters). The `/Binary` encoding type, which is not specified in the Red Book, is used for raw binary data without any special encoding applied.

It is neither required nor possible to specify the image read procedure. This procedure will be built implicitly during the interpretation of the CIP3 PPF file.

In order to create image data for the CIP3 PPF file, first the compression technique must be applied. The resulting data must then be encoded as specified. When reading a CIP3 PPF, the two mechanisms have to be applied in the opposite order.

Attribute Name	Type	Necessary	Meaning
<b>CIP3PreviewImageWidth</b>	integer	required	Width of preview image in pixel
<b>CIP3PreviewImageHeight</b>	integer	required	Height of preview image in pixel
<b>CIP3PreviewImageBitsPerComp</b>	integer	required	Number of bits per color component (1 or 8)
<b>CIP3PreviewImageComponents</b>	integer	required	Number of image components (1 or 4)
<b>CIP3PreviewImageMatrix</b>	array of 6 floats	required	Image matrix (see chapter 3.4.3)
<b>CIP3PreviewImageResolution</b>	array of 2 floats	required	Resolution of preview image in pixels per inch
<b>CIP3PreviewImageEncoding</b>	name	required	Type of encoding: <code>/Binary</code> , <code>/ASCIIHexDecode</code> , <code>/ASCII85Decode</code>
<b>CIP3PreviewImageCompression</b>	name	required	Type of compression: <code>/None</code> , <code>/RunLengthDecode</code> , <code>/DCTDecode</code> (= JPEG), <code>/CCITTFaxDecode</code> (= Fax Group 3 and 4)
<b>CIP3PreviewImageFilterDict</b>	dict	optional	Dictionary containing special information for compression method (see Red Book [PostScript], page 127)
<b>CIP3PreviewImageByteAlign</b>	integer	optional	Specifies that the image data for one image line is aligned to the number of bytes given (1, 2, or 4; default is 1) This attribute may only be used, if no encoding or compression is used.



<b>CIP3PreviewImageDataSize</b>	integer	optional	Specifies the number of bytes used to store the compressed and encoded image (as specified in the attributes CIP3PreviewImageCompression and CIP3PreviewImageEncoding). See chapter 3.4.5 for detailed description of where the image data starts.
---------------------------------	---------	----------	--

**Table 3-9: Attributes of the preview image**

### 3.4.1. Composite Preview Image

If the standard printing colors Cyan, Magenta, Yellow and Black (in this order) are used, it is possible to store the image as a composite CMYK image. In this case the field CIP3PreviewImageComponents has to be set to 4.

**Example: Composite Preview Image (CMYK)**

```
CIP3BeginPreviewImage
/CIP3PreviewImageWidth 2000 def
/CIP3PreviewImageHeight 1400 def
/CIP3PreviewImageBitsPerComp 8 def
/CIP3PreviewImageComponents 4 def
/CIP3PreviewImageMatrix [2000 0 0 -1400 0 1400] def
/CIP3PreviewImageResolution [ 50.8 50.8 ] def
/CIP3PreviewImageEncoding /ASCIISHexDecode def
/CIP3PreviewImageCompression /DCTDecode def
CIP3PreviewImage
... <image data>
CIP3EndPreviewImage
```

### 3.4.2. Preview Image with Separations

If the preview image is stored in separations, a substructure for each separation must be specified in the CIP3 file. The sequence of the separations must match the sequence specified in the *CIP3AdmSeparationNames* field (see table 3-2: Administration data). In this case the field CIP3PreviewImageComponents must be set to 1.

**Example: Preview Image with two Separations**

```
CIP3BeginPreviewImage
CIP3BeginSeparation
/CIP3PreviewImageWidth 2000 def
/CIP3PreviewImageHeight 1400 def
/CIP3PreviewImageBitsPerComp 8 def
/CIP3PreviewImageComponents 1 def
/CIP3PreviewImageMatrix [2000 0 0 1400 0 0] def
```

```
/CIP3PreviewImageResolution [ 50.8 50.8 ] def
/CIP3PreviewImageEncoding /ASCII85Decode def
/CIP3PreviewImageCompression /RunLengthDecode def
CIP3PreviewImage
  <... runlength compressed and ASCII85 encoded
    image data of first separation ...>
CIP3EndSeparation
CIP3BeginSeparation
/CIP3PreviewImageWidth 2000 def
/CIP3PreviewImageHeight 1400 def
/CIP3PreviewImageBitsPerComp 8 def
/CIP3PreviewImageComponents 1 def
/CIP3PreviewImageMatrix [2000 0 0 1400 0 0] def
/CIP3PreviewImageResolution [ 50.8 50.8 ] def
/CIP3PreviewImageEncoding /Binary def
/CIP3PreviewImageCompression /None def
CIP3PreviewImage
  <... raw image data of second separation ...>
CIP3EndSeparation
CIP3EndPreviewImage
```

### 3.4.3. Sequence of Image Data

In PostScript it is very common to represent the image data line by line, starting with the first pixel in the lower-left corner. Within the CIP3 PPF eight different orientations are allowed. Depending on the orientation used for the generation of the image data, one of the PostScript matrices shown in the table below has to be defined in the CIP3PreviewImageMatrix field. In fact "w" and "h" have to be substituted by the width and the height of the preview image.

Sequence of Data	Fast Running Index	Slow Running Index	PostScript Matrix
	from left to right	from bottom to top	$[w\ 0\ 0\ h\ 0\ 0]$
	from left to right	from top to bottom	$[w\ 0\ 0\ -h\ 0\ h]$
	from right to left	from bottom to top	$[-w\ 0\ 0\ h\ w\ 0]$
	from right to left	from top to bottom	$[-w\ 0\ 0\ -h\ w\ h]$
	from bottom to top	from left to right	$[0\ h\ w\ 0\ 0\ 0]$
	from top to bottom	from left to right	$[0\ h\ -w\ 0\ h\ 0]$
	from bottom to top	from right to left	$[0\ -h\ w\ 0\ 0\ w]$
	from top to bottom	from right to left	$[0\ -h\ -w\ 0\ h\ w]$

**Table 3-10: Orientation of image data**

The first orientation specified in the table shown above is the standard orientation for PostScript images. If one of the other image orientations is used, it is important that only the orientation in which the image is stored in the CIP3 PPF file is changed. The only attribute that reflects this change is the CIP3PreviewImageMatrix attribute. Width and height must be given in the "original" orientation and not in the orientation the image is stored in the file.

### 3.4.4. Rules for the Generation of the Preview Image

In order to be useful for the ink consumption calculation, the preview data must be generated with an appropriate resolution. This does not only mean spatial resolution, but also color or tonal resolution. Spatial resolution is important for thin lines, while tonal resolution becomes important with large areas filled with a certain tonal value.

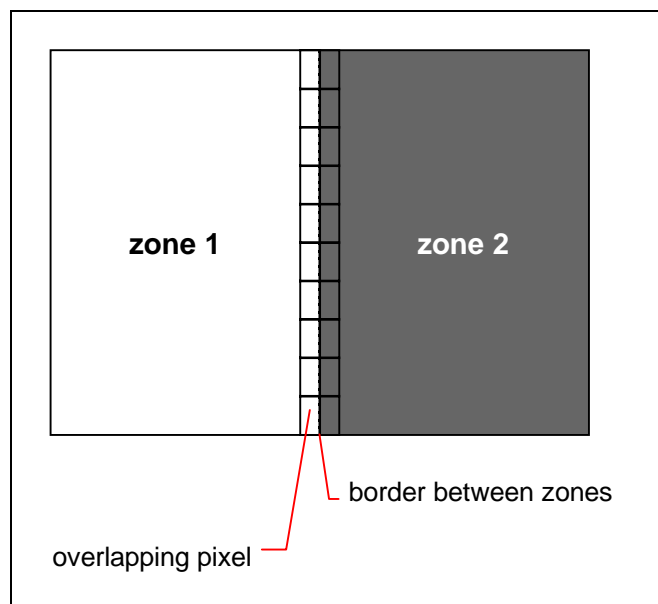
The maximum error caused by limited spatial and tonal resolution should be less than 1 %.

#### Spatial Resolution

Since some pixel of the preview image might fall on the border between two zones, their tonal values have to be split up. In the worst case the pixels are falling just in the middle between a totally white and a totally black zone. In this case the tonal value will be 50 %, but only 25 % contribute to the black zone. Depending on the resolution of the preview image and the zone width the maximum error can be calculated as:

$$\text{error}[\%] = \frac{100}{4 * \text{resolution}[\text{L} / \text{mm}] * \text{zone\_width}[\text{mm}]}$$

For zone width broader than 25 mm a resolution of 2 lines per mm will always result in an error less than 0.5 %. Therefore a resolution of 2 lines per mm (equals 50.8 dpi) is suggested.



**Illustration 3-6: Worst case**

## Tonal Resolution

The error caused by color quantization depends on the number of shades available. If the real tonal value is rounded to the closest (lower or higher) available shade, the error caused by this can be calculated by:

$$\text{error}[\%] = \frac{100}{2 * \text{number\_of\_shades}}$$

Therefore at least 64 shades should be used.

## Line Art Resolution

When rasterizing line art elements, the minimal line width is 1 pixel, i.e. 1/resolution. Therefore the relation between the printing resolution and the (spatial) resolution of the preview image is important for these kind of elements. In addition to that a specific characteristic of PostScript RIPs adds another error: within PostScript each pixel is set, which is touched by a line.

Tests with different PostScript jobs have shown, that a line art resolution of more than 300 dpi normally is sufficient for the ink consumption calculation.

## Conclusion

There are different ways to meet the requirements listed above. One way could be the following:

1. ripping the job with 406.4 dpi monochrome
2. filtering (with anti-aliasing) the image data down by a factor of 8 in both directions. This results in an image of 50.8 dpi with 65 color shades.

Another way might be to filter (with anti-aliasing) the high resolution data (e.g. used for plate making):

1. taking the ripped data (2540 dpi monochrome)
2. filtering (with anti-aliasing) the image data down by a factor of 50 in both directions. This results in an image of 50.8 dpi with 2501 color shades
3. mapping 2501 shades to 256 shades (without affecting the spatial resolution)

**Note:** *Rasterizing a job with 50.8 dpi and 256 shades of gray is not sufficient. The problem in this case is the rendering of thin lines (see: Line Art Resolution).*

## Minimal Requirements

It is strongly recommended to adhere to the requirements described above:

- resolution of ripped line art at least 300 dpi
- spatial resolution of preview image approximately 20 pixel/cm (= 50.8 dpi)
- tonal resolution of preview image at least 64 shades

### 3.4.5. Using Image Encoding and Compression

The details of individual filters as described in chapter 3.13.3 of the PostScript Language Reference Manual must be obeyed when using encoding or compression. In particular the end of data codes must be provided after the image data: ">" for /ASCIIHexDecode, "~>" for /ASCII85Decode, and 128<sub>dec</sub> for /RunLengthDecode.

The image data starts after one white space character following the keyword "CIP3PreviewImage". The combination of carriage return followed by line-feed is regarded as one white space character (see chapter 3.8.1 of the PostScript Language Reference Manual for a more detailed description). If carriage return only is used as EOL convention (as on Macintosh computers) and the first character of the image data might be line-feed, it is recommended to use the space character to separate the keyword "CIP3PreviewImage" from the image data.

### 3.5. Characteristic Curves for Transfer

The characteristic curves for transfer are stored as attributes. They are the basis of the calculation of ink consumption.

Attribute Name	Type	Necessary	Meaning
<b>CIP3TransferFilmCurveData</b>	array of floats	required	Data of copy to film curve of transfer: an even number of values between 0.0 and 1.0; each pair defines one point of the transfer curve.
<b>CIP3TransferPlateCurveData</b>	array of floats	required	Data of copy to printing plate curve of transfer: an even number of values between 0.0 and 1.0; each pair defines one point of the transfer curve.

**Table 3-11: Characteristic curves for transfer**

If the same values of the *CIP3TransferFilmCurveData* and *CIP3TransferPlateCurveData* attributes are valid for both front and back, the definition of the two attributes should be made in the *Sheet* structure of the CIP3 file. If there are different curves of transfer used for front and back, the definitions must be placed in the *Front* and the *Back* structure. If each separation uses different curves of transfer, the definitions can also be placed in the *Separation* structures.

#### Example:

```
/CIP3TransferFilmCurveData [0.0 0.0 0.3 0.2 0.5 0.45 0.7 0.5 1.0 1.0] def
/CIP3TransferfPlateCurveData [0.0 0.0 0.3 0.2 0.5 0.45 0.7 0.5 1.0 1.0] def
```

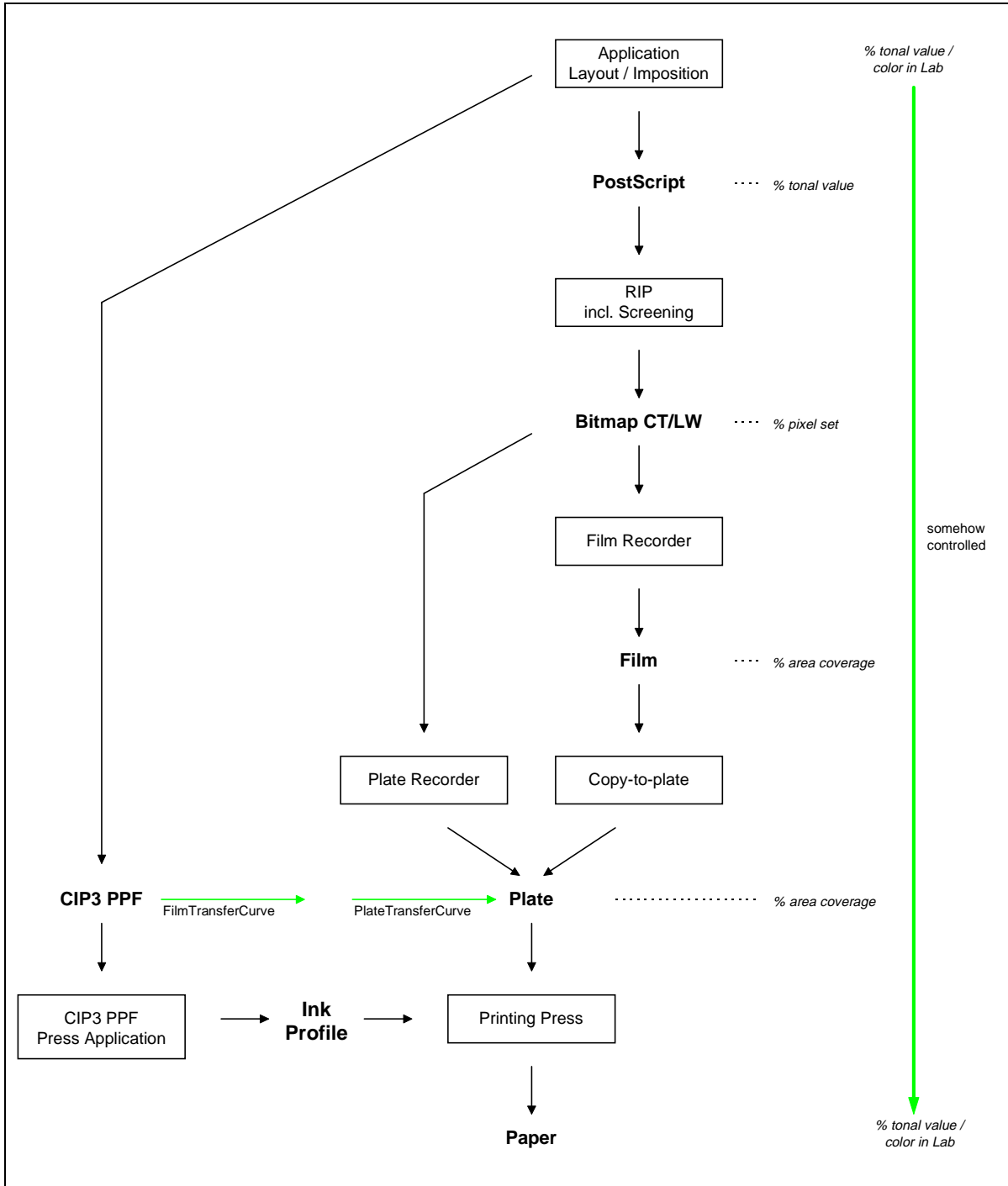


Illustration 3-7: Curves of transfer



### 3.6. Register Marks

All register marks are compiled in the **RegisterMarks** structure. The position and rotation of each register mark can be specified with the help of three parameters. It is important that the register marks are defined in such a way that their centers are on the point of origin of the coordinate system, because otherwise they are not positioned properly.

In addition to their position on the sheet register marks can be associated with a set of separations. By default each register mark is associated with each separation of the job. The **CIP3SetRegisterMarkSeparations** command is used to specify the set of separations to which the subsequently defined marks are bound.

Parameter	Type	Meaning
<i>separations</i>	array of strings	List of separation names. Each name must exactly match one of the separations names specified in the <i>CIP3AdmSeparationNames</i> field (see table 3-2: Administration data).

**Table 3-12: Parameter for the CIP3SetRegisterMarkSeparations**

Parameter	Type	Meaning
<i>translate-x</i>	float	Relocation in X-direction.
<i>translate-y</i>	float	Relocation in Y-direction.
<i>rotation</i>	float	Rotation in degrees: positive graduation figures indicate counter-clockwise rotation; negative figures indicate clockwise rotation.
<i>type</i>	name	Type of register mark

**Table 3-13: Parameter for register marks**

#### Example:

```
CIP3BeginRegisterMarks
30.0 30.0 0 /regm1 CIP3PlaceRegisterMark
625.0 30.0 0 /regm1 CIP3PlaceRegisterMark
625.0 872.0 0 /regm1 CIP3PlaceRegisterMark
30.0 872.0 0 /regm1 CIP3PlaceRegisterMark
[(Cyan) (Magenta) (Yellow)] CIP3SetRegisterMarkSeparations
5.0 30.0 0 /special_mark CIP3PlaceRegisterMark
CIP3EndRegisterMarks
```

### 3.7. Color and Ink Control

All color and ink measuring fields or control strips are compiled in the **ColorControl** structure. There are three different types of data elements defined:

- color measuring fields,
- density measuring fields, and
- color control strips.

#### Color and Density Measuring Fields

Color and density measuring fields can be positioned on the front or back page of the sheet by using the **CIP3PlaceMeasuringField** command.

Parameter	Type	Meaning
<i>position-x</i>	float	Position of the center of the color measuring field in X-direction.
<i>position-y</i>	float	Position of the center of the color measuring field in Y-direction.
<i>data</i>	dict	Dictionary containing detailed information about color measuring field (see following tables).

**Table 3-14: Parameter for the CIP3PlaceMeasuringField command**

To allow for an easy extension of the color and ink control features of the CIP3 format the specific data required by a certain type of measuring field is collected in a PostScript dictionary. Each entry in a dictionary is build by a pair of key and value. Within CIP3 always names are used as keys.

Key	Type	Necessary	Meaning
<i>/CIE-L*</i>	float	required	Color specification: value of L*
<i>/CIE-a*</i>	float	required	Color specification: value of a*
<i>/CIE-b*</i>	float	required	Color specification: value of b*
<i>/Diameter</i>	float	required	Diameter of measuring field
<i>/Light</i>	name	required	Type of light: /D50, /D65, ...
<i>/Observer</i>	int	required	Observer in degree (2 or 10)
<i>/Tolerance</i>	float	required	Tolerance in $\Delta E$
<i>/Setup</i>	string	optional	Description of measurement setup
<i>/Percentages</i>	array of floats	optional	Film percentage values for each separation. The number of array elements must match the number of separations (as defined by the CIP3AdmSeparationNames attribute)

<b>/ScreenRuling</b>	array of floats	optional	Screen ruling values in lines per inch for each separation. The number of array elements must match the number of separations (as defined by the CIP3AdmSeparationNames attribute)
<b>/ScreenShape</b>	string	optional	shape of screening dots
<b>/DensityStandard</b>	string	optional	Density filter norm: <b>/DIN16536</b> , <b>/DIN16536NB</b> , <b>/ANSIA</b> , <b>/ANSIT</b>
<b>/Type</b>	name	required	Type of measuring: <b>/CIELAB</b>

Table 3-15: Data dictionary entries for a color measuring field

Key	Type	Necessary	Meaning
<b>/Screen</b>	string	required	Description of screen
<b>/Separation</b>	name	required	Reference to separation
<b>/DensityBlack</b>	float	required	Density value measured with filter for black
<b>/ToleranceBlack</b>	array of 2 floats	required	Upper and lower tolerance (in density units)
<b>/DensityCyan</b>	float	required	Density value measured with filter for cyan
<b>/ToleranceCyan</b>	array of 2 floats	required	Upper and lower tolerance (in density units)
<b>/DensityMagenta</b>	float	required	Density value measured with filter for magenta
<b>/ToleranceMagenta</b>	array of 2 floats	required	Upper and lower tolerance (in density units)
<b>/DensityYellow</b>	float	required	Density value measured with filter for yellow
<b>/ToleranceYellow</b>	array of 2 floats	required	Upper and lower tolerance (in density units)
<b>/DotGain</b>	float	required	Percentage of dot gain
<b>/ToleranceDotGain</b>	array of 2 floats	required	Upper and lower tolerance (in percentage)
<b>/Percentage</b>	float	required	Film percentage or equivalent
<b>/Diameter</b>	float	required	Diameter of measuring field
<b>/Setup</b>	string	optional	Description of measurement setup
<b>/Type</b>	name	required	Type of measuring field: <b>/Density</b>

Table 3-16: Data dictionary entries for a density measuring field

## Color Control Strips

A color control strip can be specified with the **CIP3PlaceColorControlStrip**. Positioning and Rotation of control strips refer to the point of origin of the coordinate system in which the strip is defined.

Parameter	Type	Meaning
<i>position-x</i>	float	Position of the center of the control strip in X-direction.
<i>position-y</i>	float	Position of the center of the control strip in Y-direction.
<i>rotation</i>	float	Rotation in degrees: positive graduation figures indicate counter-clockwise rotation; negative figures indicate clockwise rotation.
<i>width</i>	float	Width of control strip (before rotation).
<i>height</i>	float	Height of control strip (before rotation).
<i>data</i>	array of arrays	Definition of fields contained in the control strip. This array may be empty, if no definition of the control strip is needed. The lower left corner of the control strip box (defined by position, width, and height) builds the origin of the coordinate system used for the definition of the elements contained in the control strip.
<i>name</i>	name	Name of control strip

**Table 3-17: Parameter for control strips**

### Example:

```

CIP3BeginColorControl
/Black50 16 dict def
Black50 begin
  /Screen (60 / cm) def
  /Separation (Black) def
  /DensityBlack 0.331 def
  /ToleranceBlack [-0.02 0.02] def
  /DensityCyan 0.293 def
  /ToleranceCyan [-0.02 0.02] def
  /DensityMagenta 0.313 def
  /ToleranceMagenta [-0.02 0.02] def
  /DensityYellow 0.054 def
  /ToleranceYellow [-0.02 0.02] def
  /DotGain 15.2 def
  /ToleranceDotGain [2.0 -2.0] def
  /Percentage 50.0 def
  /Diameter 5.0 mm def
  /Type /Density def
end
/Cyan50 16 dict def
Cyan50 begin

```

```
/Screen (60 / cm) def
/Separation (Cyan) def
/DensityBlack 0.338 def
/ToleranceBlack [-0.02 0.02] def
/DensityCyan 0.446 def
/ToleranceCyan [-0.02 0.02] def
/DensityMagenta 0.043 def
/ToleranceMagenta [-0.02 0.02] def
/DensityYellow 0.032 def
/ToleranceYellow [-0.02 0.02] def
/DotGain 13.3 def
/ToleranceDotGain [2.0 -2.0] def
/Percentage 50.0 def
/Diameter 5.0 mm def
/Type /Density def
end
/Magenta50 16 dict def
Magenta50 begin
  /Screen (60 / cm) def
  /Separation (Magenta) def
  /DensityBlack 0.347 def
  /ToleranceBlack [-0.02 0.02] def
  /DensityCyan 0.230 def
  /ToleranceCyan [-0.02 0.02] def
  /DensityMagenta 0.413 def
  /ToleranceMagenta [-0.02 0.02] def
  /DensityYellow 0.069 def
  /ToleranceYellow [-0.02 0.02] def
  /DotGain 12.0 def
  /ToleranceDotGain [2.0 -2.0] def
  /Percentage 50.0 def
  /Diameter 5.0 mm def
  /Type /Density def
end
/Yellow50 16 dict def
Yellow50 begin
  /Screen (60 / cm) def
  /Separation (Yellow) def
  /DensityBlack 0.357 def
  /ToleranceBlack [-0.02 0.02] def
  /DensityCyan 0.157 def
  /ToleranceCyan [-0.02 0.02] def
  /DensityMagenta 0.284 def
  /ToleranceMagenta [-0.02 0.02] def
  /DensityYellow 0.357 def
  /ToleranceYellow [-0.02 0.02] def
  /DotGain 5.7 def
  /ToleranceDotGain [2.0 -2.0] def
  /Percentage 50.0 def
```

```
/Diameter 5.0 mm def
/Type /Density def
end
100.0 mm 15.5 mm Black50 CIP3PlaceMeasuringField
105.0 mm 15.5 mm Cyan50 CIP3PlaceMeasuringField
110.0 mm 15.5 mm Magenta50 CIP3PlaceMeasuringField
115.0 mm 15.5 mm Yellow50 CIP3PlaceMeasuringField
635.0 mm 689 mm 0 20 mm 5 mm
[
  [ 0.0 mm 0 mm Black50 ]
  [ 5.0 mm 0 mm Cyan50 ]
  [ 10.0 mm 0 mm Magenta50 ]
  [ 15.0 mm 0 mm Yellow50 ]
]
/UserControlStrip CIP3PlaceColorControlStrip
CIP3EndColorControl
```

### 3.8. Cutting Data

All necessary data required for cutting are compiled in the **CutData** structure.

Within the CIP3 format cutting data is described by nested blocks, where at the lowest level of hierarchy a folding procedure may be applied on a block. Since cutting is described here in a way that is as much as possible device independent, the cutting blocks specified in the CIP3 format do not directly imply a certain cutting sequence. This has to be determined by a specialized application, that also generates the cutting program, which can be loaded and executed on a cutting device.

Some information about the sheet which is also required for the cutting procedure is stored in attributes of the parent structures (e.g. in the sheet structure or in the front structure). These attributes (e.g. administration data) are inherited from the parent blocks. Therefore this information may not be included in the **CutData** structure. Although it is possible to overwrite these attributes by defining them in a cut block. This can be useful to specify different product identifications for the blocks arranged on a sheet.

#### Defining a cut block

It is possible to define a block which contains a matrix of elements of all the same size. There, the intermediate cut dimension is calculated from the information about element size, block size and number of elements in both directions. A cut block structure must be enclosed in a pair of **CIP3BeginCutBlock** and **CIP3EndCutBlock** commands. Cut block structures may be nested.

Attribute Name	Type	Necessary	Meaning
<b>CIP3CutModel</b>	string	optional	Name of the cutting machine

**Table 3-18: Attributes used in the CutData structure**

Attribute Name	Type	Necessary	Meaning
<b>CIP3BlockTrf</b>	array of 6 floats	required	Block transformation matrix: defines position and orientation of the block relative to the parent block or structure
<b>CIP3BlockSize</b>	array of 2 floats	required	Size of the block
<b>CIP3BlockElementSize</b>	array of 2 floats	optional	Element dimension in X and Y direction
<b>CIP3BlockSubdivision</b>	array of 2 integers	optional	Number of elements in X and Y direction

<b>CIP3BlockType</b>	name	required	Block type: <b>/CutBlock</b> = block to be cut, <b>/SaveBlock</b> = protected block, cut only via outer contour, <b>/TempBlock</b> = auxiliary block, is not taken into account during cutting, <b>/MarkBlock</b> = contains no elements, only marks
<b>CIP3BlockElementType</b>	name	optional	Element type: <b>/CutElement</b> = cutting element, <b>/PunchElement</b> = punching element
<b>CIP3BlockName</b>	string	required	Name of the block. The name must be unique and must not contain a slash character.
<b>CIP3BlockFoldingProcedure</b>	name	optional	Reference to folding procedure specified in the FoldingData section. If a block is subdivided into an array of elements, this folding procedure is applied to each element, otherwise it is applied to the whole block.

**Table 3-19: Attributes used in the CutBlock structure**

### Description of cut marks

In addition to the definition of cut blocks it is possible to position cut marks on the sheet by using the **CIP3PlaceCutMark** command. After printing these marks can be used to adapt the theoretical block positions (as specified in the CIP3) to the real position of the corresponding blocks on the printed sheet.

Parameter	Type	Meaning
<i>position-x</i> <i>position-y</i>	float float	Logical position of the cut mark. This coordinate does not always directly specify the position (e.g. the center) of the cut mark symbol. <i>See illustration 3-8 for the relation between the logical position and the position of the mark symbol.</i>
<i>mark-type</i>	name	Mark type: <b>/CrossCutMark</b> , <b>/TopVerticalCutMark</b> , <b>/BottomVerticalCutMark</b> , <b>/LeftHorizontalCutMark</b> , <b>/RightHorizontalCutMark</b> , <b>/LowerLeftCutMark</b> , <b>/UpperLeftCutMark</b> , <b>/LowerRightCutMark</b> , <b>/UpperRightCutMark</b> . <i>See illustration 3-8 for the geometric shape of each predefined mark type.</i>

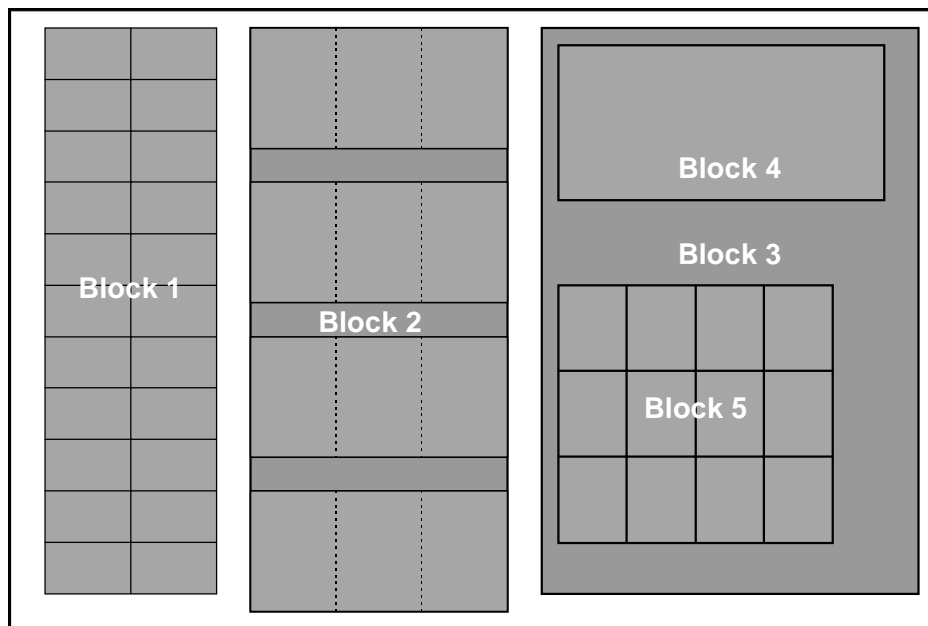
**Table 3-20: Description of cut marks**



Symbol	Name	Position of symbol
	CrossCutMark	Centered at logical position
	TopVerticalCutMark	Slightly above logical position
	BottomVerticalCutMark	Slightly below logical position
	LeftHorizontalCutMark	Slightly to the left of logical position
	RightHorizontalCutMark	Slightly to the right of logical position
	LowerLeftCutMark	Corner at logical position
	UpperLeftCutMark	Corner at logical position
	LowerRightCutMark	Corner at logical position
	UpperRightCutMark	Corner at logical position

**Illustration 3-8: The set of predefined cut marks**

**Example 1: Block Definition**



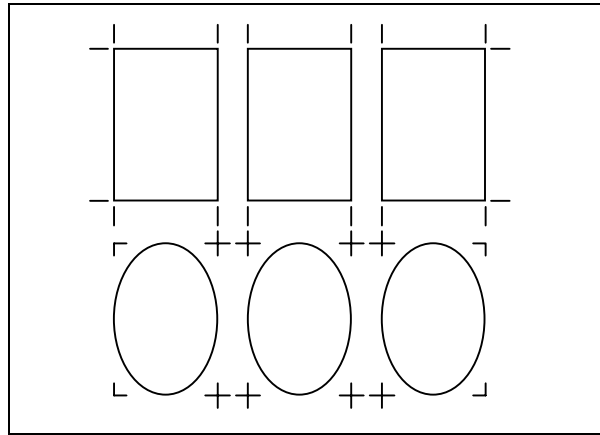
**Illustration 3-9: Example of a cutting arrangement**

```
CIP3BeginCutData
/CIP3CutModel (Model 175) def

CIP3BeginCutBlock
/CIP3BlockTrf      [1 0 0 1 4 cm 4 cm] def
/CIP3BlockSize    [20.2 cm 66.22 cm] def
/CIP3BlockElementSize [10.1 cm 6.02 cm] def
/CIP3BlockSubdivision [2 11] def
/CIP3BlockType     /CutBlock def
/CIP3BlockElementType /Unknown def
/CIP3BlockName     (Block 1) def
CIP3EndCutBlock

CIP3BeginCutBlock
/CIP3BlockTrf      [1 0 0 1 28.25 cm 1.5 cm] def
/CIP3BlockSize    [29.9 cm 68.7 cm] def
/CIP3BlockElementSize [29.9 cm 16.8 cm] def
/CIP3BlockSubdivision [1 4] def
/CIP3BlockType     /CutBlock def
/CIP3BlockElementType /Unknown def
/CIP3BlockName     (Block 2) def
/CIP3FoldingProcedure /DoubleFold def
CIP3EndCutBlock

CIP3BeginCutBlock
/CIP3BlockTrf      [1 0 0 1 63.0 cm 4.0 cm] def
/CIP3BlockSize    [29.9 cm 68.7 cm] def
/CIP3BlockType     /TempBlock def
/CIP3BlockName     (Block 3) def
CIP3BeginCutBlock
/CIP3BlockTrf      [1 0 0 1 2.0 cm 54.1 cm] def
/CIP3BlockSize    [38.8 cm 18.2 cm] def
/CIP3BlockType     /CutBlock def
/CIP3BlockName     (Block 4) def
CIP3EndCutBlock
CIP3BeginCutBlock
/CIP3BlockTrf      [1 0 0 1 4.0 cm 6.5 cm] def
/CIP3BlockSize    [32.4 cm 30.3 cm] def
/CIP3BlockElementSize [8.1 cm 10.1 cm] def
/CIP3BlockSubdivision [4 3] def
/CIP3BlockType     /CutBlock def
/CIP3BlockElementType /PunchElement def
/CIP3BlockName     (Block 5) def
CIP3EndCutBlock
CIP3EndCutBlock
CIP3EndCutData
```

**Example 2: Cut Marks****Illustration 3-10: Example showing the use of cut marks**

15.1 cm	52.0 cm	/TopVerticalCutMark	CIP3PlaceCutMark
29.4 cm	52.0 cm	/TopVerticalCutMark	CIP3PlaceCutMark
33.6 cm	52.0 cm	/TopVerticalCutMark	CIP3PlaceCutMark
47.9 cm	52.0 cm	/TopVerticalCutMark	CIP3PlaceCutMark
52.1 cm	52.0 cm	/TopVerticalCutMark	CIP3PlaceCutMark
66.4 cm	52.0 cm	/TopVerticalCutMark	CIP3PlaceCutMark
15.1 cm	52.0 cm	/LeftHorizontalCutMark	CIP3PlaceCutMark
66.4 cm	52.0 cm	/RightHorizontalCutMark	CIP3PlaceCutMark
15.1 cm	31.0 cm	/LeftHorizontalCutMark	CIP3PlaceCutMark
66.4 cm	31.0 cm	/RightHorizontalCutMark	CIP3PlaceCutMark
15.1 cm	31.0 cm	/BottomVerticalCutMark	CIP3PlaceCutMark
29.4 cm	31.0 cm	/BottomVerticalCutMark	CIP3PlaceCutMark
33.6 cm	31.0 cm	/BottomVerticalCutMark	CIP3PlaceCutMark
47.9 cm	31.0 cm	/BottomVerticalCutMark	CIP3PlaceCutMark
52.1 cm	31.0 cm	/BottomVerticalCutMark	CIP3PlaceCutMark
66.4 cm	31.0 cm	/BottomVerticalCutMark	CIP3PlaceCutMark
15.1 cm	26.0 cm	/UpperLeftCutMark	CIP3PlaceCutMark
29.4 cm	26.0 cm	/CrossCutMark	CIP3PlaceCutMark
33.6 cm	26.0 cm	/CrossCutMark	CIP3PlaceCutMark
47.9 cm	26.0 cm	/CrossCutMark	CIP3PlaceCutMark
52.1 cm	26.0 cm	/CrossCutMark	CIP3PlaceCutMark
66.4 cm	26.0 cm	/UpperRightCutMark	CIP3PlaceCutMark
15.1 cm	5.0 cm	/LowerLeftCutMark	CIP3PlaceCutMark
29.4 cm	5.0 cm	/CrossCutMark	CIP3PlaceCutMark
33.6 cm	5.0 cm	/CrossCutMark	CIP3PlaceCutMark
47.9 cm	5.0 cm	/CrossCutMark	CIP3PlaceCutMark
52.1 cm	5.0 cm	/CrossCutMark	CIP3PlaceCutMark
66.4 cm	5.0 cm	/LowerRightCutMark	CIP3PlaceCutMark

### 3.9. Folding Data

The creation of a folded product is described by a folding procedure. In the **FoldProcedures** section, a folding procedure can be defined. It is possible to specify more than one folding procedure in a CIP3 file. Each procedure is stored under a unique name by which it can be referenced in the CutData and CutBlock structures.

The cutting information contained in this section is only intended for the cutting procedures within the folding equipment.

#### Defining a folding procedure

A folding procedure is defined as a PostScript procedure containing some attribute settings and a sequence of folding steps. At the moment the following five applications of a folding device are possible: folding, cutting, grooving, perforating and liming.

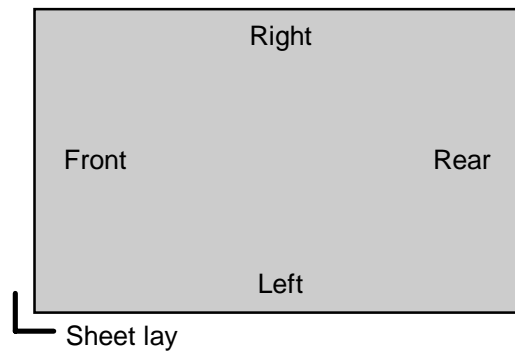
At the beginning of a folding procedure definition it is necessary to specify the size of the input sheet. If this size does not match the size of the corresponding cut block, all coordinates of the folding procedure are scaled respectively. This allows for the specification of a folding procedure that can be used in different sizes.

After each folding or cutting step of a folding procedure the origin of the coordinate system is moved to the lower left corner of the intermediate folding product.

Attribute Name	Type	Necessary	Meaning
<b>CIP3FoldDescription</b>	string	optional	Description of the type of fold according to Bundesverband Druck (Federal Association of Printing) i.e. general description.  <i>Hint: Within a PostScript string parenthesis „( )“ must be marked with a ` in front of it.</i>
<b>CIP3FoldSheetIn</b>	array of 2 floats	required	Input sheet format
<b>CIP3FoldProc</b>	array	required	steps of folding procedure

**Table 3-21: Attributes of a folding procedure**

The specification of reference edges (**/Front**, **/Rear**, **/Left** and **/Right**) for the description of an operation (e.g. for the positioning of a tool) is done by means of determined names. With regard to upper and lower case they have to be written exactly as in the following illustration.



**Illustration 3-11: Names of the reference edges of a sheet**

The individual steps of a folding procedure must be given in the sequence in which they should be carried out. Each step requires the specification of some parameter values and is concluded with the indication of the application name (**Fold**, **Cut**, **Groove**, **Perforate** or **Lime**). The direction of folding is defined by means of a name (**/Up** or **/Down**).

Parameter	Type	Meaning
<i>travel</i>	float	Distance of the reference edge ( <i>from</i> )
<i>from</i>	name	Edge from where it is folded: <b>/Front</b> = from the front, <b>/Left</b> = from the left
<i>to</i>	name	Direction in which it is folded: <b>/Up</b> = upwards, <b>/Down</b> = downwards.
<i>function</i>	name	Function of the operation: <b>Fold</b> = folding

**Table 3-22: Description of a folding operation**

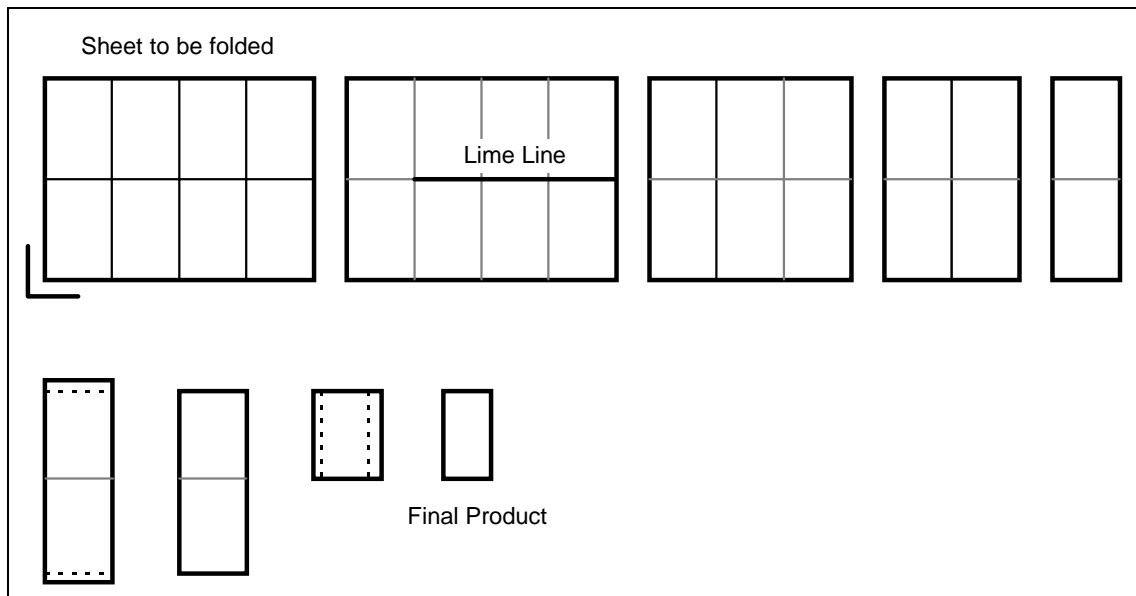
For indicating the direction in which a tool (for cutting, grooving, perforating, or liming) processes the sheet, the following names have been defined: **/Bottom** and **/Top**.

Parameter	Type	Meaning
<i>start-position</i>	2 floats	Starting position of the tool
<i>working-path</i>	2 floats	Relative working path of the tool. Since the tools can only work parallel to the edges, one coordinate must be zero.
<i>working-direction</i>	name	Direction from which the tool is working: <b>/Top</b> = from above, <b>/Bottom</b> = from below
<i>function</i>	name	Function of the operation: <b>Cut</b> = cutting, <b>Groove</b> = grooving, <b>Perforate</b> = perforating, <b>Lime</b> = liming

**Table 3-23: Description of an operation for cutting, grooving, perforating or liming**

**Example:**

*Note: The following example refers to the third example in the description of the PrePress Interface of Stahl (version of 22.8.94, section 3.5.3, pages 15-17).*



**Illustration 3-12: Example of a folded product**

```

CIP3BeginFoldProcedures
/A4-16 <<
  /CIP3FoldDescription (F16 / 3W \((1/4\) + 1 ML)
  /CIP3FoldSheetIn [880 mm 610 mm]
  /CIP3FoldProc
  [
    20 mm 305 mm 660 mm 0 /Top Lime
    0 mm /Front /Up Fold           % origin moves right
    0 mm /Front /Up Fold           % origin moves right
    0 mm /Front /Up Fold           % origin moves right
    0 mm 8 mm 220 mm 0 /Top Cut    % origin moves down
    0 mm 594 mm 220 mm 0 /Top Cut
    297 mm /Left /Up Fold          % origin moves up
    5 mm 0 mm 0 297 mm /Top Cut    % origin moves right
    210 mm 0 mm 0 297 mm /Top Cut
  ]
  >> def
CIP3EndFoldProcedures

```

### 3.10. Comments and Annotations

Comments and annotations are treated as content. A comment is specified by typing the comment text and the **CIP3Comment** command. It is not printed. An annotation is a printed comment. It is specified by 5 parameters (x- and y-coordinate, annotation text, font name, and font size) and the **CIP3Annotation** command.

#### Example

```
(This is a not printed comment) CIP3Comment  
10 cm 12.3 cm (This is a printed comment) /Helvetica 12 CIP3Annotation
```

### 3.11. Private Data

For many applications it is comfortable to store additional, application specific data within the CIP3 PPF file. This can be done by use of a PrivateData structure. This structure can be inserted within any other structure. Even the nesting of PrivateData structures is allowed. PrivateData structures have to be named using the following rule:

- the name of the PrivateData structure must start with a company specific prefix containing at least three characters. It is not allowed to use names starting with the "CIP3" prefix. The company specific prefix should be registered by the CIP3 group.

Within a PrivateData structure arbitrary definitions can be made. Each attribute can be defined by use of the "def" operator.

```
/attribute-name value def
```

The attribute name must be unique within one PrivateData structure. Although it is not required to use a company specific prefix for attribute name, this is regarded as good coding technique. The value has to be provided as a PostScript object (see chapter 3.1.2 for a complete list of supported data types). Arrays and dictionaries may be used to group information.

```
/FhGPrivate CIP3BeginPrivate
/FhGPrivValue 42 def
/FhGPrivDictionary <</Company (Fraunhofer)
                    /Date (07.03.96)
                    /Array [2.4 5.4 (Text)]
                    >> def
CIP3EndPrivate
```



### 3.12. Private Content

Since binary data consisting of more than 65535 bytes cannot be represented as a PostScript string, the concept of private content has been introduced. With this method the amount of data that can be stored in one single attribute is only limited by the available memory. The length of the data in bytes must be provided in order to allow for skipping it by positioning the file pointer.

The private content data may contain arbitrary binary data without any restriction. The data bytes start after one white space character following the keyword "CIP3PrivateContent". The combination of carriage return followed by line-feed is regarded as one white space character (see chapter 3.8.1 of the PostScript Language Reference Manual for a more detailed description). If carriage return only is used as EOL convention (as on Macintosh computers) and the first character of the private content might be line-feed, it is recommended to use the space character to separate the keyword "CIP3PrivateContent" from the private content data.

```
name length CIP3PrivateContent  
< ... length bytes of data ... >
```

#### Example:

```
...  
  
/FhGPrivateDataElem 24 CIP3PrivateContent  
123456789012345678901234  
  
...
```

## Appendix A: Changes from CIP3 PPF version 2.0 to version 2.1

The following changes to the CIP3 PPF specification document have been made:

- **version number changed** (chapter 3.1.1)  
The version number has been changed from 2.0 to 2.1.
- **syntactical structure of CIP3 file** (illustration 3-1)  
The illustration has been adapted to show the PPFDirectory and the possibility of describing more than one sheet within one CIP3 PPF file.
- **explanation of how to code objects** (chapter 3.1.2)  
A description has been added of how to code objects of type boolean, integer, real, string, name, array, and dict. In addition the coding of PostScript comments has been explained.
- **specification of syntax limitations** (chapter 3.1.2)  
In order to further reduce the complexity of a CIP3 PPF file the following PostScript operators have been removed from the list of supported operators:  
`begin, end, dict, {, }`.
- **What is a valid CIP3 PPF file?** (chapter 3.1.5)  
A chapter explaining what a valid CIP3 PPF file must consist of has been added.
- **PPF directory** (chapter 3.2)  
A PPF directory was added to support multi sheet PPF files.
- **administration data** (chapter 3.3)  
The CIP3AdmSheetLay attribute has been changed from required to optional and a short explanation has been added. A new optional attribute CIP3AdmTypeOfScreen has been added. A set of attributes for web presses has been added.
- **structure names** (chapter 3.3 and chapter 3.8)  
No slash characters may be used in structure names. Therefore the separation names (stored in the CIP3AdmSeparationNames array) and the names of a cut blocks (stored in CIP3BlockName) must not contain slash characters.
- **preview image data** (chapter 3.4)  
A new optional attribute specifying the size of the encoded and compressed image data has been added. The fax compression mode has been corrected (changed from CCITTDcode to CCITTFaxDecode).
- **using image encoding and compression** (chapter 3.4.5)  
A new chapter has been added with hints for using PostScript encoding and compression filters to code preview image data.
- **characteristic curves of transfer** (chapter 3.5)  
A diagram explaining the semantic of the curves of transfer has been added. It is now allowed to store separation specific curves of transfer in the PPF file.
- **color and ink control** (chapter 3.7)  
An optional entries specifying the percentage values for each separation, the density filter standard, and the shape of the screening dots has been added to the dictionary for color measuring fields.
- **cutting data** (chapter 3.8)  
The attributes of a CutData structure have been clearly separated from the attributes of a CutBlock structure. The data type used for CIP3BlockTrf has changed from "matrix" to "array of 6 floats".

- **folding data (chapter 3.9)**  
Instead of a PostScript procedure (with "{" and "}" brackets) a dictionary notation (with "<<" and ">>" brackets) is used to code a folding procedure. The sequence of working steps is then coded in an array stored in the CIP3FoldProc attribute
- **private content (chapter 3.12)**  
A means of storing large blocks of private data has been established by adding the concept of private content.
- **dictionary elements**  
All dictionary elements have been explicitly marked as optional or required.
- **list of illustrations (appendix E)**  
A list of illustrations has been added.
- **list of tables (append F)**  
A list of tables has been added.

## Appendix B: Changes from CIP3 PPF version 1.0 to version 2.0

The following changes to the CIP3 PPF specification document have been made:

- **version number changed** (chapter 3.1.1)  
The version number has been changed from 1.0 to 2.0
- **prologue removed** (chapter 3.1.1)  
In order to reduce the complexity of a CIP3 PPF file it is no longer possible to make user specific definition within a prologue.
- **specification of syntax limitations** (chapter 3.1.2.)  
In order to reduce the complexity of a CIP3 PPF file some syntax limitations have been defined. In addition a complete list of all PostScript operators that may be used in a CIP3 PPF file has been added.
- **specification of default value for extent and transformation attributes** (chapter 3.1.3)  
The default values for an unspecified transformation matrix (CIP3AdmFilmTrf, CIP3AdmPlateTrf, CIP3AdmPressTrf, CIP3AdmPaperTrf) and an unspecified extent (CIP3AdmFilmExtent, CIP3AdmPlateExtent, CIP3AdmPressExtent, CIP3AdmPaperExtent) have been defined (see chapter 3.2.: administration data).
- **change of necessary flag of two administration data attributes** (chapter 3.2.)  
The CIP3AdmSeparationNames and the CIP3AdmPSExtent attributes have been changed from optional to required.
- **changes to the preview image** (chapter 3.3)  
In order to reduce to size of a CIP3 PPF file encoding and compression techniques can be applied to preview images. Since the use of LZW compression requires a license fee paid to Unisys, LZW is excluded from the list of supported compression techniques.  
The coding of preview images has been changed from an OPI-like style to the definition of attributes similar to the ones used for administration data. Due to the problems related with links to external files, the OPI-like file references are no longer supported.  
In addition to continuous tone images the new version also allows bitmap images (1 bit per pixel) to be used as preview images. Of course these images must be provided in a higher resolution.
- **specification of register marks associated with a set of separations** (chapter 3.5)  
In addition to their position on the sheet register marks can now be associated with a set of separations. By default a register mark is associated with each separation of the job (as in version 1.0).
- **private data added** (chapter 3.10.)  
To allow for the definition of private data within a CIP3 PPF file, a separate structure has been defined.
- **changes to the list of references** (appendix A)  
The references to the OPI and TIFF specifications have been removed. In addition the reference to the Document Structuring Conventions Specification has been removed, since it is part of the PostScript Language Reference Manual.
- **folding data now in separate structure** (chapter 3.8.)  
The definition of fold procedures has been moved from the beginning of the file (just after the removed prolog) to a separate structure.

## Appendix C: Example of a CIP3 PPF file

*Note: The example shown below is collection of most of the examples shown in previous sections of this specification. Its logical structure is shown in illustration 2-1 on page 5. The sheet described with this CIP3 PPF file comprises a front side with four colors (CMYK) and a back side with two colors (Cyan and Black).*

```
%!PS-Adobe-3.0
%%CIP3-File Version 2.1

CIP3BeginSheet
(Sheet structure of CIP3 example) CIP3Comment
/CIP3AdmJobName (TestJob) def
/CIP3AdmMake (Prepress Company) def
/CIP3AdmModel (PC210) def
/CIP3AdmSoftware (The Imposition Program) def
/CIP3AdmCreationTime (1994:08:30 12:17:03) def
/CIP3AdmArtist (Stefan Daun) def
/CIP3AdmCopyright (Copyright by Fraunhofer-IGD, 1995) def
/CIP3AdmPSExtent [100 cm 70 cm] def
/CIP3AdmSheetLay Left def
/CIP3AdmPrintVolume 120000 def
/CIP3AdmPaperGrammage 130 def

(Transfer data is valid for both front and back) CIP3Comment
/CIP3TransferFilmCurveData [0.0 0.0 0.3 0.2 0.5 0.45 0.7 0.5 1.0 1.0] def
/CIP3TransferfPlateCurveData [0.0 0.0 0.3 0.2 0.5 0.45 0.7 0.5 1.0 1.0] def

CIP3BeginFront
(Front page structure of CIP3 example) CIP3Comment
(Different number of colors for front and back side) CIP3Comment
/CIP3AdmSeparationNames [(Cyan) (Magenta) (Yellow) (Black)] def

CIP3BeginPreviewImage
/CIP3PreviewImageWidth 2000 def
/CIP3PreviewImageHeight 1400 def
/CIP3PreviewImageBitsPerComp 8 def
/CIP3PreviewImageComponents 4 def
/CIP3PreviewImageMatrix [2000 0 0 -1400 0 1400] def
/CIP3PreviewImageResolution [ 50.8 50.8 ] def
/CIP3PreviewImageEncoding /ASCIIHexDecode def
/CIP3PreviewImageCompression /DCTDecode def
CIP3PreviewImage
... <image data>
CIP3EndPreviewImage

CIP3BeginRegisterMarks
30.0 30.0 0 /regm1 CIP3PlaceRegisterMark
625.0 30.0 0 /regm1 CIP3PlaceRegisterMark
```

```
625.0 872.0 0 /regm1 CIP3PlaceRegisterMark
30.0 872.0 0 /regm1 CIP3PlaceRegisterMark

/FhGPrivate CIP3BeginPrivate
/FhGPrivValue 42 def
/FhGPrivDictionary << /Company (Fraunhofer)
                    /Date (07.03.96)
                    /Array [ 2.4 5.4 (Text)]
                    >> def
CIP3EndPrivate

CIP3EndRegisterMarks

CIP3BeginColorControl
/Black50 16 dict def
Black50 begin
  /Screen (60 / cm) def
  /Separation (Black) def
  /DensityBlack 0.331 def
  /ToleranceBlack [-0.02 0.02] def
  /DensityCyan 0.293 def
  /ToleranceCyan [-0.02 0.02] def
  /DensityMagenta 0.313 def
  /ToleranceMagenta [-0.02 0.02] def
  /DensityYellow 0.054 def
  /ToleranceYellow [-0.02 0.02] def
  /DotGain 15.2 def
  /ToleranceDotGain [2.0 -2.0] def
  /Percentage 50.0 def
  /Diameter 5.0 mm def
  /Type /Density def
end
/Cyan50 16 dict def
Cyan50 begin
  /Screen (60 / cm) def
  /Separation (Cyan) def
  /DensityBlack 0.338 def
  /ToleranceBlack [-0.02 0.02] def
  /DensityCyan 0.446 def
  /ToleranceCyan [-0.02 0.02] def
  /DensityMagenta 0.043 def
  /ToleranceMagenta [-0.02 0.02] def
  /DensityYellow 0.032 def
  /ToleranceYellow [-0.02 0.02] def
  /DotGain 13.3 def
  /ToleranceDotGain [2.0 -2.0] def
  /Percentage 50.0 def
  /Diameter 5.0 mm def
  /Type /Density def
```

```
end
/Magenta50 16 dict def
Magenta50 begin
  /Screen (60 / cm) def
  /Separation (Magenta) def
  /DensityBlack 0.347 def
  /ToleranceBlack [-0.02 0.02] def
  /DensityCyan 0.230 def
  /ToleranceCyan [-0.02 0.02] def
  /DensityMagenta 0.413 def
  /ToleranceMagenta [-0.02 0.02] def
  /DensityYellow 0.069 def
  /ToleranceYellow [-0.02 0.02] def
  /DotGain 12.0 def
  /ToleranceDotGain [2.0 -2.0] def
  /Percentage 50.0 def
  /Diameter 5.0 mm def
  /Type /Density def
end
/Yellow50 16 dict def
Yellow50 begin
  /Screen (60 / cm) def
  /Separation (Yellow) def
  /DensityBlack 0.357 def
  /ToleranceBlack [-0.02 0.02] def
  /DensityCyan 0.157 def
  /ToleranceCyan [-0.02 0.02] def
  /DensityMagenta 0.284 def
  /ToleranceMagenta [-0.02 0.02] def
  /DensityYellow 0.357 def
  /ToleranceYellow [-0.02 0.02] def
  /DotGain 5.7 def
  /ToleranceDotGain [2.0 -2.0] def
  /Percentage 50.0 def
  /Diameter 5.0 mm def
  /Type /Density def
end
100.0 mm 15.5 mm Black50 CIP3PlaceMeasuringField
105.0 mm 15.5 mm Cyan50 CIP3PlaceMeasuringField
110.0 mm 15.5 mm Magenta50 CIP3PlaceMeasuringField
115.0 mm 15.5 mm Yellow50 CIP3PlaceMeasuringField
635.0 mm 689 mm 0 20 mm 5 mm
[
  [ 0.0 mm 0 mm Black50 ]
  [ 5.0 mm 0 mm Cyan50 ]
  [ 10.0 mm 0 mm Magenta50 ]
  [ 15.0 mm 0 mm Yellow50 ]
]
/UserControlStrip CIP3PlaceColorControlStrip
```

```
CIP3EndColorControl

CIP3EndFront

CIP3BeginBack
(Back page structure of CIP3 example) CIP3Comment
/CIP3AdmSeparationNames [(Cyan) (Black)] def

CIP3BeginPreviewImage
CIP3BeginSeparation
/CIP3PreviewImageWidth 2000 def
/CIP3PreviewImageHeight 1400 def
/CIP3PreviewImageBitsPerComp 8 def
/CIP3PreviewImageComponents 1 def
/CIP3PreviewImageMatrix [2000 0 0 1400 0 0] def
/CIP3PreviewImageResolution [ 50.8 50.8 ] def
/CIP3PreviewImageEncoding /ASCII85Decode def
/CIP3PreviewImageCompression /RunLengthDecode def
CIP3PreviewImage
  <... runlength compressed and ASCII85 encoded
    image data of Cyan separation ...>
CIP3EndSeparation
CIP3BeginSeparation
/CIP3PreviewImageWidth 2000 def
/CIP3PreviewImageHeight 1400 def
/CIP3PreviewImageBitsPerComp 8 def
/CIP3PreviewImageComponents 1 def
/CIP3PreviewImageMatrix [2000 0 0 1400 0 0] def
/CIP3PreviewImageResolution [ 50.8 50.8 ] def
/CIP3PreviewImageEncoding /Binary def
/CIP3PreviewImageCompression /None def
CIP3PreviewImage
  <... raw image data of Black separation ...>
CIP3EndSeparation
CIP3EndPreviewImage

CIP3BeginRegisterMarks
30.0 30.0 0 /regm1 CIP3PlaceRegisterMark
625.0 30.0 0 /regm1 CIP3PlaceRegisterMark
625.0 872.0 0 /regm1 CIP3PlaceRegisterMark
30.0 872.0 0 /regm1 CIP3PlaceRegisterMark
CIP3EndRegisterMarks

CIP3EndBack

CIP3EndSheet

%%CIP3EndOfFile
```



## Appendix D: References

- [BVD] Bundesverband Druck: Handbuch zur Anwendung einheitlicher Falzbezeichnungen.- Falzartenkatalog, 1990.
- [CIE1] ISO 10526: CIE Standard Colorimetric Illuminance.
- [CIE2] ISO 10527: CIE Standard Colorimetric Observers.
- [DSC] Adobe Systems Inc.: PostScript Language Document Structuring Conventions Specification - Version 3.0 (25.9.92).
- [PostScript] Adobe Systems Inc.: PostScript Language Reference Manual. Reading Mass.: Addison-Wesley, 1990, 2nd Edition.

**Appendix E: List of Illustrations**

Illustration 1-1: CIP3-Scenario .....	4
Illustration 2-1: Example of the hierarchical structure of a CIP3 file .....	5
Illustration 3-1: Syntactical structure of the CIP3 file .....	9
Illustration 3-2: Coordinate Systems and Coordinate Mapping .....	12
Illustration 3-3: Fixed size directory entry .....	17
Illustration 3-4: Example of longitudinal application (view from top) .....	22
Illustration 3-5: Example of cross application (view from top) .....	22
Illustration 3-6: Worst case .....	27
Illustration 3-7: Curves of transfer .....	31
Illustration 3-8: The set of predefined cut marks .....	40
Illustration 3-9: Example of a cutting arrangement .....	40
Illustration 3-10: Example showing the use of cut marks .....	42
Illustration 3-11: Names of the reference edges of a sheet .....	44
Illustration 3-12: Example of a folded product .....	45

## Appendix F: List of Tables

Table 3-1:	List of PostScript data types allowed in CIP3 PPF .....	11
Table 3-2:	List of PostScript operators allowed in CIP3 PPF .....	11
Table 3-3:	Units in the CIP3 format .....	13
Table 3-4:	Structure types and names .....	14
Table 3-5:	Parameter for the CIP3PPFDirEntry command .....	17
Table 3-6:	General administration data .....	20
Table 3-7:	Web press data .....	21
Table 3-8:	Data dictionary for longitudinal and cross applications .....	21
Table 3-9:	Attributes of the preview image .....	24
Table 3-10:	Orientation of image data .....	26
Table 3-11:	Characteristic curves for transfer .....	30
Table 3-12:	Parameter for the CIP3SetRegisterMarkSeparations .....	32
Table 3-13:	Parameter for register marks .....	32
Table 3-14:	Parameter for the CIP3PlaceMeasuringField command .....	33
Table 3-15:	Data dictionary entries for a color measuring field .....	34
Table 3-16:	Data dictionary entries for a density measuring field .....	34
Table 3-17:	Parameter for control strips .....	35
Table 3-18:	Attributes used in the CutData structure .....	38
Table 3-19:	Attributes used in the CutBlock structure .....	39
Table 3-20:	Description of cut marks .....	39
Table 3-21:	Attributes of a folding procedure .....	43
Table 3-22:	Description of a folding operation .....	44
Table 3-23:	Description of an operation for cutting, grooving, perforating or liming .	44

## Appendix G: Index of Key Words and Comments

<b>A</b>	
ASCII85Decode .....	23; 29
ASCIHexDecode .....	23; 29
<b>B</b>	
Binary .....	23
Bottom .....	44
BottomVerticalCutMark .....	39
<b>C</b>	
CCITTFaxDecode .....	23
CIE-a* .....	33
CIE-b* .....	33
CIE-L* .....	33
CIP3AdmArtist .....	19
CIP3AdmCarbonizingColorName .....	20
CIP3AdmCoating .....	20
CIP3AdmCopyright .....	19
CIP3AdmCreationTime .....	19
CIP3AdmCrossApp .....	21
CIP3AdmCustomer .....	19
CIP3AdmCylinderCircumference .....	20
CIP3AdmFilmExtent .....	20
CIP3AdmFilmTrf .....	20
CIP3AdmFilmType .....	19
CIP3AdmInkColors .....	19
CIP3AdmInkInfo .....	19
CIP3AdmJobCode .....	19
CIP3AdmJobName .....	19
CIP3AdmLongitudinalApp .....	21
CIP3AdmMake .....	19
CIP3AdmModel .....	19
CIP3AdmPaperColor .....	19
CIP3AdmPaperDestination .....	20
CIP3AdmPaperExtent .....	20
CIP3AdmPaperGrade .....	19
CIP3AdmPaperGrammage .....	19
CIP3AdmPaperSource .....	20
CIP3AdmPaperThickness .....	19
CIP3AdmPaperTrf .....	20
CIP3AdmPlateExtent .....	20
CIP3AdmPlateTrf .....	20
CIP3AdmPlateType .....	19
CIP3AdmPressExtent .....	20
CIP3AdmPressTrf .....	20
CIP3AdmPrintVolume .....	19
CIP3AdmPSExtent .....	20
CIP3AdmReelWidth .....	20
CIP3AdmSeparationNames .....	13; 19; 24; 33; 34
CIP3AdmSheetLay .....	19
CIP3AdmSheetName .....	19
CIP3AdmSoftware .....	19
CIP3AdmTintingColor .....	20
CIP3AdmTintingColorName .....	20
CIP3AdmTypeOfScreen .....	19
CIP3Annotation .....	46
CIP3BeginBack .....	13; 15
CIP3BeginColorControl .....	13; 35
CIP3BeginCutBlock .....	14; 38
CIP3BeginCutData .....	14; 41
CIP3BeginFoldProcedures .....	14; 45
CIP3BeginFront .....	13; 15
CIP3BeginPPFDirectory .....	18
CIP3BeginPreviewImage .....	13; 24; 52
CIP3BeginPrivate .....	14; 47
CIP3BeginRegisterMarks .....	13; 32; 55
CIP3BeginSeparation .....	13; 24; 25; 55
CIP3BeginSheet .....	13; 15
CIP3BlockElementSize .....	38
CIP3BlockElementType .....	39
CIP3BlockFoldingProcedure .....	39
CIP3BlockName .....	39
CIP3BlockSize .....	38
CIP3BlockSubdivision .....	38
CIP3BlockTrf .....	38
CIP3BlockType .....	39
CIP3Comment .....	46
CIP3CutModel .....	38
CIP3EndBack .....	13; 15
CIP3EndColorControl .....	13; 37
CIP3EndCutBlock .....	14; 38
CIP3EndCutData .....	14; 41
CIP3EndFoldProcedures .....	14; 45
CIP3EndFront .....	13; 15
CIP3EndOfFile .....	9
CIP3EndPPFDirectory .....	18
CIP3EndPreviewImage .....	13; 24
CIP3EndPrivate .....	14; 47
CIP3EndRegisterMarks .....	13; 32; 55
CIP3EndSeparation .....	13; 25; 55
CIP3EndSheet .....	13; 15
CIP3-File .....	9

CIP3FoldDescription .....	43
CIP3FoldProc.....	43
CIP3FoldSheetIn.....	43
CIP3PlaceColorControlStrip.....	35
CIP3PlaceCutMark.....	39
CIP3PlaceMeasuringField.....	33
CIP3PPFDirEntry .....	17
CIP3PreviewImage .....	29
CIP3PreviewImageBitsPerComp .....	23
CIP3PreviewImageByteAlign .....	23; 24
CIP3PreviewImageComponents .....	23; 24
CIP3PreviewImageCompression .....	23
CIP3PreviewImageEncoding .....	23
CIP3PreviewImageFilterDict .....	23
CIP3PreviewImageHeight .....	23
CIP3PreviewImageMatrix.....	23; 26
CIP3PreviewImageResolution .....	23
CIP3PreviewImageWidth .....	23
CIP3PrivateContent .....	48
CIP3SetRegisterMarkSeparations .....	32
CIP3TransferFilmCurveData.....	30
CIP3TransferPlateCurveData .....	30
cm .....	13
CrossCutMark .....	39
Cut.....	44
CutBlock.....	39
CutElement .....	39

**D**

DCTDecode .....	23
DensityBlack .....	34
DensityCyan.....	34
DensityMagenta .....	34
DensityStandard.....	34
DensityYellow.....	34
Diameter.....	33; 34
DotGain.....	34
Down .....	44

**F**

Fold .....	44
Front.....	43
Function .....	21

**G**

Groove .....	44
--------------	----

**I**

inch .....	13
------------	----

**L**

Left.....	19; 43
LeftHorizontalCutMark .....	39
Light .....	33
Lime .....	44
LowerLeftCutMark.....	39
LowerRightCutMark .....	39

**M**

MarkBlock .....	39
mm .....	13

**N**

None .....	23
------------	----

**O**

Observer .....	33
----------------	----

**P**

Percentage.....	34
Percentages.....	33
Perforate .....	44
point .....	13
PunchElement.....	39

**R**

Rear .....	43
ReferenceEdge.....	21
Right.....	19; 43
RightHorizontalCutMark.....	39
RunLengthDecode .....	23; 29

**S**

SaveBlock .....	39
Screen.....	34
ScreenRuling .....	34
ScreenShape .....	34
Separation.....	34
Setup.....	33; 34
StartPosition.....	21

**T**

TempBlock .....	39
Tolerance .....	33

ToleranceBlack .....34  
ToleranceCyan.....34  
ToleranceDotGain .....34  
ToleranceMagenta .....34  
ToleranceYellow.....34  
Top.....44  
TopVerticalCutMark .....39  
Travel .....21  
Type .....34

**U**

Up .....44  
UpperLeftCutMark.....39  
UpperRightCutMark .....39

**W**

WorkingLength.....21