Data Stream and Object Architectures

MO:DCA-L: The OS/2 Presentation Manager Metafile (.met) Format

S550-1135-00

RICOH | IBM

InfoPrint Solutions Company

Data Stream and Object Architectures

MO:DCA-L: The OS/2 Presentation Manager Metafile (.met) Format

S550-1135-00

RICOH | IBM
InfoPrint Solutions Company

Note:
Before using this information and the product it supports, read the information in "Notices" on page 33.

First edition (August 2008)

Internet
Visit our home page: `http://www.infoprint.com`

You can send comments by e-mail to **printpub@us.ibm.com** or by mail to:

InfoPrint Solutions Company
6300 Diagonal Hwy 002J
Boulder, CO 80301-9270
U.S.A.

Contents

Figures

Tables

Chapter 1. Introduction to the MO:DCA-L format

Two forms of the Mixed Object Document Content Architecture™ (MO:DCA™) data stream format existed in the early 1990's and have been documented in the *Mixed Object Document Content Architecture Reference* since that time:

- The presentation form, called MO:DCA-P, describes final-form documents in terms of a document structure and the mixtures of presentation objects that define page content within that structure.
- The library form, called MO:DCA-L, describes a mixture of presentation objects without providing document structure.

The MO:DCA-P form defines what is probably better known as the Advanced Function Presentation™ (AFP™) format. In the past, this format has also been called the AFP Data Stream (APFDS). It is now usually simply referred to as AFP, and should be considered synonymous with MO:DCA.

The MO:DCA-L form was used in the IBM® Operating System/2® (OS/2®) product to define the Presentation Manager (PM) metafile format. These files were known as .met files. Unfortunately the fact that the .met file format is formally defined by the MO:DCA architecture as the MO:DCA-L format was not publicized very well and is not widely known.

IBM no longer markets the OS/2 product, and the OS/2 PM file format has been stable for a long time. There are no plans to extend this format in any manner, therefore the MO:DCA-L format should be considered functionally capped. There is no need to maintain the definition of this format inside an active architecture like the MO:DCA architecture, which is now maintained and developed by the AFP Consortium™ (AFPC™). Nor is any further development of this format within the scope of the work of the AFPC. The definition of this format has therefore been removed from the MO:DCA reference (SC31-6802) and is now published as a "retired" IBM data stream format in the MO:DCA-L document.

Chapter 2. MO:DCA-L overview

The MO:DCA-L format uses the syntax defined by the MO:DCA architecture. For a complete definition, see the *Mixed Object Document Content Architecture Reference*, available at www.afpcolor.org. The primary syntax elements are self-defining structures called *structured fields* and *triplets*. Most of the structured fields and triplets that are used in the MO:DCA-L format are fully defined in the *Mixed Object Document Content Architecture Reference*. These definitions are not repeated in this document. Only syntactic and semantic information that is unique to the MO:DCA-L format and that is not covered in the MO:DCA reference is included in this document.

Structured field ID

MO:DCA structured fields are identified by a three-byte identifier:

SFTypeID A three-byte field that uniquely identifies the structured field. It has the form *D3TTCC*, where:

Code	Description
D3	The structured field *class* code that has been assigned to the MO:DCA architecture.
TT	The structured field *type* code. The type code identifies the function of the structured field, such as begin, end, descriptor, or data.
CC	The structured field *category* code. It identifies the lowest-level component that can be constructed using the structured field, such as document, resource group, or object. The same category code point assigned to a component's begin structured field also is assigned to that component's end structured field. These code points identify and delimit an entire component within a data stream or an encompassing component.

MO:DCA-L structured field type codes

Table 1 lists the type codes that are used in the MO:DCA-L format.

Table 1. Structured field type codes

Type Code	Function	Description
X'A6'	Descriptor	A *descriptor* structured field defines the initial characteristics and, optionally, the formatting directives for all objects. Depending on the specific descriptor structured field type, it may contain some set of parameters that identify: • The size of the object • Measurement units • Initial presentation conditions
X'A8'	Begin	A *begin* structured field introduces and identifies a document component. In general, a begin structured field may contain a parameter that identifies the name of the component.

Table 1. Structured field type codes (continued)

Type Code	Function	Description
X'A9'	End	An *end* structured field identifies the end of a document component. In general, an end structured field may contain a parameter that identifies the name of the component.
X'AB'	Map	A *map* structured field provides the following functions in the MO:DCA architecture: • All occurrences of a variable embedded in structured field parameter data can be given a new value by changing only one reference in the mapping, rather than having to physically change each occurrence. Thus all references to font X may cause a Times Roman font to be used in one instance and a Helvetica font in another instance merely by specifying the proper *map coded font* structured field. • The presence of the map structured field in a MO:DCA-L environment group indicates use of the named resource within the scope of the environment group.
X'B0'	Table	A *table* structured field contains a list of items of the same or similar type that are related to one another.
X'EE'	Data	A *data* structured field consists of data whose meaning and interpretation is governed by the object architecture for the particular data object type.

MO:DCA-L structured field category codes

The following category codes are used in the MO:DCA-L format. All other category codes are reserved.

Category Code	Description
X'77'	Color attribute table
X'8A'	Coded font
X'A8'	Document
X'BB'	Graphics
X'C3'	Data resource
X'C6'	Resource group
X'C7'	Object environment group (OEG)
X'EE'	No operation
X'FB'	Image

Data stream states

The MO:DCA architecture defines a *state* as a domain within the data stream, bounded by a begin-end structured field pair, within which certain structured fields are permitted. The processor of a MO:DCA data stream is required to check the validity of the structured field sequence received. A valid structured field sequence is one in which each structured field that is processed belongs to the set of permissible structured fields for the begin-end envelope in which it is found. If a structured field other than one belonging to the set of permissible structured fields is detected, a violation of the state has occurred, and the processor is required to raise an exception condition.

The following states are used in the MO:DCA-L format:

State Description

Document
Initiated by a Begin Document structured field and terminated by an End Document structured field. The Begin Document structured field defines the beginning of the MO:DCA-L data stream, within which all other MO:DCA-L document-level structured fields are contained.

Resource Group
Initiated by a Begin Resource Group structured field and terminated by an End Resource Group structured field. Structured fields that define resources such as color tables may be encountered in the resource group state.

Resource Object
Initiated by a begin resource object structured field for resources such as color attribute tables, graphics objects, and image objects, and terminated by a corresponding end resource object structured field. Structured fields that define the contents of resource objects may be encountered in the resource object state.

Object Environment Group
Initiated by a Begin Object Environment Group structured field and terminated by an End Object Environment Group structured field. Structured fields that provide environment specifications affecting objects may be encountered in the object environment group state.

State hierarchies

States are grouped and organized hierarchically. The following state hierarchies are used in the MO:DCA-L format:

- States permitted within the Document state:
  - Resource Group
- States permitted within the Resource Group state:
  - Resource Object
- States permitted within the Resource Object state:
  - Resource Group if the object is an image object
  - Object Environment Group if the object is a graphics objects or an image object
- States permitted within the Object Environment Group state:
  - None

Exception conditions

For a complete discussion of MO:DCA exception conditions, see the *Mixed Object Document Content Architecture Reference*. All MO:DCA-L architecture violations are handled as exception conditions. It is possible for the processor of a MO:DCA-L data stream to continue processing when it encounters exception conditions. This permits a process that cannot faithfully present a document to continue with its best approximation.

MO:DCA and MO:DCA-L syntax tables identify the categories of exception conditions that can occur for each data element through the use of a code listed in the *Exc* column. Each of the exception conditions is related to a bit position, as shown in Table 2 on page 6. The value assigned to *Exc* is based on the positions of the bits that represent the exception condition categories that can apply to the data element. If no exception condition is possible, the *Exc* column contains X'00'.

For example, if it is possible for the data element to contain a value outside of the prescribed range, or if it is possible for its value to conflict with that of another parameter, then both the unacceptable parameter value and the inconsistent parameter value exception conditions can apply. The unacceptable parameter value is represented by bit position six or B'00000010', and the inconsistent parameter value is represented by bit position seven or B'00000001'. The code that is entered into the *Exc* column is formed by ORing the bit representations of the exception condition categories that are possible, in this example resulting in B'00000011' or X'03'.

Table 2. MO:DCA-L exception condition categories

Bit position	Exception condition category	Code	
		Binary	**Hexadecimal**
Bit 0	Invalid structured field identifier	B'10000000'	X'80'
Bit 1	Unrecognized identifier code	B'01000000'	X'40'
Bit 2	Data stream state violation	B'00100000'	X'20'
Bit 3	Unrecognized structured field or triplet	B'00010000'	X'10'
Bit 4	Required structured field missing	B'00001000'	X'08'
Bit 5	Required parameter missing	B'00000100'	X'04'
Bit 6	Unacceptable parameter value	B'00000010'	X'02'
Bit 7	Inconsistent parameter values	B'00000001'	X'01'
None	None	B'00000000'	X'00'

The action to be performed by a product that detects an exception condition depends on the product.

Chapter 3. MO:DCA-L structured fields

The MO:DCA-L format uses a subset of the structured fields and triplets that are defined in the *Mixed Object Document Content Architecture Reference*. The syntax is as defined in the reference. Wherever there are MO:DCA-L specific exceptions, these are normally noted with an architecture note in that reference.

The following structured fields are unique to the MO:DCA-L format and are therefore not defined in the MO:DCA reference. See the MO:DCA reference for the definition of the triplets used on these structured fields.

Begin Color Attribute Table (BCA)

The Begin Color Attribute Table structured field begins a Color Attribute Table resource object, which becomes the current resource object. A color attribute table contains color attribute data.

BCA (X'D3A877') syntax

Structured field introducer				
SF length (2B)	ID = **X'D3A877'**	Flags (1B)	Reserved; X'0000'	Structured field data

Offset	Type	Name	Range	Meaning	M/O	Exc
0–7	CHAR	CATName		Name of the color attribute table	M	X'06'
8–*n*		Triplets		See “BCA semantics” for triplet applicability.	O	X'10'

BCA semantics

CATName Is the name of the color attribute table. This name may not appear on more than one Begin Color Attribute Table in the same resource group or a X’01’ exception condition exists .

The resource group containing the Begin Color Attribute Table structured field must also contain a subsequent matching End Color Attribute Table structured field, or a X’08’ exception condition exists.

Color attribute tables may reside in external libraries, in one or more resource groups within a MO:DCA document, or in a combination of the two. See the *Mixed Object Document Content Architecture Reference* for details on locating resource objects within libraries and resource groups.

Triplets Appear as follows:

Triplet	Type	Usage
X'01'	Coded Graphic Character Set Global Identifier	Optional. May occur more than once. Specifies encoding for structured field parameters defined with a CHAR data type. See the *Mixed Object Document Content Architecture Reference*.
X'02'	Fully Qualified Name	Optional. May occur once. See the *Mixed Object Document Content Architecture Reference*. The Fully Qualified Name type that may appear is **X'01'**—*Replace First GID Name*. This GID overrides the Begin Color Attribute Table structured field name and is used as the name of the color attribute table.
X'65'	Comment	Optional. May occur more than once. Carries unarchitected data. See the *Mixed Object Document Content Architecture Reference*.

BCA exception condition summary

- A X'08' exception condition exists when a subsequent matching End Color Attribute Table structured field is not present in the same resource group.
- A X'01' exception condition exists when multiple Begin Color Attribute Table structured fields with the same name exist within the same resource group.

Color Attribute Table (CAT)

The Color Attribute Table structured field contains the data for a color attribute table resource object.

CAT (X'D3B077') syntax

Structured field introducer				
SF length (2B)	ID = **X'D3B077'**	Flags (1B)	Reserved; X'0000'	Structured field data

Offset	Type	Name	Range	Meaning	M/O	Exc
0–*n*	UNDF	CATData		Up to 32,759 bytes of color table data	O	X'00'

CAT semantics

CATData Contains the color attribute table data. See Chapter 4, “MO:DCA-L color attribute table,” on page 15 for a detailed description.

End Color Attribute Table (ECA)

The End Color Attribute Table structured field terminates the Color Attribute Table resource object initiated by a Begin Color Attribute Table structured field.

ECA (X'D3A977') syntax

Structured field introducer				
SF length (2B)	ID = **X'D3A977'**	Flags (1B)	Reserved; X'0000'	Structured field data

Offset	Type	Name	Range	Meaning	M/O	Exc
0–7	CHAR	CATName		Name of the color attribute table	O	X'02'
8–*n*		Triplets		See “ECA semantics” for triplet applicability.	O	X'10'

ECA semantics

CATName Is the name of the color attribute table being terminated. If a name is specified, it must match the name in the most recent Begin Color Attribute Table structured field in the resource group or a X’01’ exception condition exists . If the first two bytes of CATName contain the value X'FFFF', the name matches any name specified on the Begin Color Attribute Table structured field that initiated the current definition.

A matching Begin Color Attribute Table structured field must appear within the resource group at some location preceding the End Color Attribute Table structured field, or a X’20’ exception condition exists.

Triplets Appear as follows:

Triplet	Type	Usage
X'02'	Fully Qualified Name	Optional. May occur once. See the *Mixed Object Document Content Architecture Reference*. The Fully Qualified Name type that may appear is **X'01'**—*Replace First GID Name*. This GID overrides the End Color Attribute Table structured field name and is used as the name of the color attribute table being terminated.

Note: If a triplet is included on this structured field, the optional positional parameter becomes mandatory.

ECA exception condition summary

- A X’01’ exception condition exists when a name is specified that does not match the name on the most recent Begin Color Attribute Table structured field.
- A X’20’ exception condition exists when the End Color Attribute Table structured field is not preceded by a matching Begin Color Attribute Table structured field.

Map Color Attribute Table (MCA)

The Map Color Attribute Table structured field maps a unique Resource Local ID to the name of a Begin Color Attribute Table structured field. A local ID may be embedded one or more times within an object's data.

MCA (X'D3AB77') syntax

Structured field introducer				
SF length (2B)	ID = **X'D3AB77'**	Flags (1B)	Reserved; X'0000'	Structured field data

Offset	Type	Name	Range	Meaning	M/O	Exc
One to 254 repeating groups in the following format:						
0–1	UBIN	RGLength	7–260	Total length of this repeating group	M	X'06'
2–*n*		Triplets		See "MCA semantics" for triplet applicability.	M	X'14'

MCA semantics

RGLength Specifies the total length of the repeating group, including the length of the RGLength parameter itself.

Triplets Appear within each repeating group as follows:

Triplet	Type	Usage
X'02'	Fully Qualified Name	Mandatory. Must occur once in each repeating group. See the *Mixed Object Document Content Architecture Reference*. The Fully Qualified Name type that may appear is **X'84'**—*Begin Resource Object Reference*, which must match the name on a Begin Color Attribute Table structured field or a X'01' exception condition exists .
X'24'	Resource Local Identifier	Mandatory for image, not present for graphics. For image, this triplet must occur once in each repeating group. See the *Mixed Object Document Content Architecture Reference*. The only resource type that may appear is **X'07'**—*Color Attribute Table*.
X'01'	Coded Graphic Character Set Global Identifier	Optional. May occur more than once. Specifies encoding for structured field parameters defined with a CHAR data type. See the *Mixed Object Document Content Architecture Reference*.

Within the same Map Color Attribute Table structured field, it is not permissible to map the same Resource Local ID to more than one color attribute table or a X'01' exception condition exists . However, two or more repeating groups within the same Map Color Attribute Table structured field may be used to map different LIDs to the same color attribute table.

Note: If this structured field is not present in the data stream, the architected default LID is X'00'.

MCA exception condition summary

- A X'02' exception condition exists when:
  - A Fully Qualified Name (X'02') triplet other than a type X'84' (Begin Resource Object Reference) appears within any repeating group
  - A Resource Local Identifier (X'24) triplet type other than X'07' appears within any repeating group
- A X'01' exception condition exists when:
  - A Begin Color Attribute Table structured field with the same name as that specified on the type X'84' (Begin Resource Object Reference) Fully Qualified Name triplet could not be located
  - Multiple type X'84' (Begin Resource Object Reference) Fully Qualified Name triplets appear within the same repeating group
  - Multiple type X'07' Resource Local Identifier triplets appear within the same repeating group
  - The same LID is mapped to more than one color attribute table within the same structured field

Chapter 4. MO:DCA-L color attribute table

The color table is preceded by the Begin Color Attribute Table structured field and is terminated by the End Color Attribute Table structured field. Within this bracket, the color table definition is carried in a set of Color Attribute Table structured fields.

Color representation in MO:DCA-L data streams

Colors are represented by one basic color model, RGB, and one subsidiary model, grayscale, which has an architected representation and an architected conversion to RGB.

- RGB is the only form allowed for interchange.
- Grayscale is provided for compatibility with existing products.
- Entries can be loaded as explicit values or can be generated by a formula based on a breakdown of each index.
- A variable number of bits per component is supported, for instance, 4 red bits, 5 green bits and 3 blue bits.
- Interchange forms are limited to RGB only, and to a maximum of 8 bits per component.

RGB representation

The RGB representation of colors is based on the way in which display terminals create color. It defines each color to be composed of various proportions of three primary colors: red, green and blue.

The Color Table assumes the x,y chromaticity coordinates defined by the Society of Motion Picture and Television Engineers (SMPTE) recommended practices. Specifically, the x,y chromaticity coordinates for the three primary colors and the reference white point are defined in SMPTE RP 145-1987 entitled *Color Monitor Colorimetry* and RP 37-1969 entitled *Color Temperature for Color Television Studio Monitors*, respectively. The reference white point is commonly known as *Illuminant* D_{6500} or simply *D65*. The recommended gamma is 2.2.

The SMPTE-defined x,y chromaticity coordinates are:

Red	x = 0.630, y = 0.340
Green	x = 0.310, y = 0.595
Blue	x = 0.155, y = 0.070
White point	x = 0.313, y = 0.329

where x and y are the coordinates within the Commission Internationale de l'Eclairage (CIE) chromaticity diagram.

This does not mean that any particular device is required to produce these exact values. It defines the intent of the user. A system's software and hardware would be expected to produce a reasonable match with these values.

Grayscale representation

The Y component represents the intensity, and ranges from 0, which corresponds to black, through 1. The basic formula to convert from RGB to grayscale is:

```
Y = 0.299R + 0.587G + 0.114B
```

Color table definition in MO:DCA-L data streams

The definition consists of a base part, followed by one or more self-defining parameters (SDP). Each SDP defines a set of entries to be loaded into the color table.

There are two types of SDP:
- Element List
- Bit Generator

Each SDP is processed in turn. The entries defined by each SDP replace any entries defined by a previous SDP.

Base part

Offset	Type	Name	Range	Meaning
0	BITS	FLAGS		
Bit 0			B'0'	Only valid value
Bit 1		RESET	B'0', B'1'	**B'0'** Do not reset LCT **B'1'** Reset LCT
Bits 2–7			B'000000'	Only valid value
1			X'00'	Only valid value
2	CODE	LCTID	X'00'–X'FF'	Local identifier of the color table

The base part defines the initialization conditions for the color table.

RESET Has the following values:

Value	Description
B'1'	The color table is reset prior to setting according to the following SDPs.
B'0'	No reset is performed.

LCTID Is a local identifier of the color table.

Element List self-defining parameter

Offset	Type	Name	Range	Meaning
0	UBIN	LEN	X'0B'–X'FF'	Length of this parameter
1	CODE		X'01'	Type: **X'01'** Element List
2			X'00'	Only valid value
3	CODE	FORMAT	X'01', X'02'	**X'01'** RGB **X'02'** Grayscale **All others** Reserved
4–6	UBIN	INDEX_1		Starting index for load
7	UBIN	SIZE_1	X'00'–X'FF'	Number of bits in component 1
8	UBIN	SIZE_2	X'00'–X'FF'	Number of bits in component 2
9	UBIN	SIZE_3	X'00'–X'FF'	Number of bits in component 3

Offset	Type	Name	Range	Meaning
10	UBIN	TRILEN	X'00'–X'F4'	Number of bytes in each element
11–*n*		ELEMENTS		Color elements

An Element List SDP defines a contiguous block of entries in the color table by defining the explicit content of each entry as a set of values.

FORMAT Specifies the format of each element in the element list:

RGB Each element consists of a set of red, green, blue intensity values. The set is in the order red, green, blue.

Grayscale Each element consists of a Y-component.

INDEX_1 Specifies the position in the Color Table where the first element is to be loaded.

SIZE_1—SIZE_3

Specify the number of bits in each part of the element. For example, if FORMAT is RGB, then SIZE_1 specifies the number of red bits, SIZE_2 specifies the number of green bits and SIZE_3 specifies the number of blue bits.

The maximum integer value of each component of the color is determined by the corresponding SIZE parameter. Thus, M1, M2 and M3 are computed using the following formulas:

$$M1 = (2^{SIZE\_1} - 1)$$
$$M2 = (2^{SIZE\_2} - 1)$$
$$M3 = (2^{SIZE\_3} - 1)$$

TRILEN Specifies the length of each element in bytes.

ELEMENTS Is a variable sized list of elements. Each element contains three components. The size of each component is an integral number of bytes, given by the formula:

```
COMP_i = 1 + INT((SIZE_i - 1) / 8)
```

where SIZE_i is the number of bits in that component. The value of each component is right-aligned in the bytes and padded with zeros, giving a set of integers (I1,I2,I3). I1, I2 and I3 are used to generate values for each component as defined in "Calculation of color value" on page 19.

- If TRILEN is greater than the sum of COMP_i, then each element is padded on the left with X'00' to make its length equal to TRILEN.
- If TRILEN is less than the sum of COMP_i, then each element is truncated on the left to make its length equal to TRILEN.

Successive values are loaded into successive positions in the table until the element list is exhausted.

Bit Generator self-defining parameter

Offset	Type	Name	Range	Meaning
0	UBIN	LEN	X'0A'	Length of this parameter

Offset	Type	Name	Range	Meaning
1	CODE		X'02'	Type: **X'02'** Bit Generator
2	BITS	FLAGS		
Bit 0		ASFLAG	B'0', B'1'	**B'0'** Additive **B'1'** Subtractive
Bits 1–7			B'0000000'	Only valid value
3	CODE	FORMAT	X'01', X'02'	**X'01'** RGB **X'02'** Grayscale **All others** Reserved
4–6	UBIN	INDEX_1		Starting index for load
7	UBIN	SIZE_1	X'00'–X'FF'	Number of bits in component 1
8	UBIN	SIZE_2	X'00'–X'FF'	Number of bits in component 2
9	UBIN	SIZE_3	X'00'–X'FF'	Number of bits in component 3

A Bit Generator SDP defines a contiguous block of entries in the color table, by defining how each entry is to be generated from its index value.

ASFLAG Specifies the meaning of the color values:

Additive The maximum color value represents full intensity of that color and the minimum color value represents zero intensity of that color. For example, in a black-and-white system, the minimum color value (usually zero) means black, and the maximum value means white.

Subtractive The minimum color value represents full intensity of that color and the maximum color value represents zero intensity of that color. For example, in a black-and-white system, the minimum color value (usually zero) means white, and the maximum value means black.

FORMAT Specifies the breakdown format for each value:

RGB Each value is to be treated as a set of red, green, blue intensity values. The set is in the order red, green, blue.

Grayscale Each value is to be treated as a Y-component.

INDEX_1 Specifies the position in the color table where the first element is to be loaded.

SIZE_1–SIZE_3

Specify the number of bits in each part of the value. The sum of these sizes, N = SIZE_1 + SIZE_2 + SIZE_3, defines how many color values are to be loaded, namely 2^N. The maximum integer value of each component of the color is determined by the corresponding SIZE parameter, giving M1, M2 and M3 respectively, thus:

$M1 = (2^{SIZE\_1} - 1)$
$M2 = (2^{SIZE\_2} - 1)$
$M3 = (2^{SIZE\_3} - 1)$

For each index, from INDEX_1 through (INDEX_1 + 2^N – 1):

1. INDEX_1 is subtracted from the index, giving a value to be broken down.
2. This value is converted to a binary integer of N bits.
3. This integer is then treated as a bit string and broken down, from left to right, into three substrings, with lengths SIZE_1, SIZE_2 and SIZE_3 respectively.
4. Each of these substrings is then converted back to a binary integer, treating the leftmost bit as most significant. This process produces a set of integers (I1,I2,I3). The process is illustrated in "Example of index breakdown" on page 20.
5. If the ASFLAG is set, then each integer is reversed by subtracting it from the corresponding maximum; M1, M2, or M3.
6. I1, I2 and I3 are then used to generate values for each component as defined in "Calculation of color value."

Successive values are loaded into successive positions in the table until 2^N colors have been loaded.

Calculation of color value

Each color value is a set of values (V1,V2,V3) where each value is in the range 0 through 1. The meaning of these values depends on the FORMAT parameter:

FORMAT X'01'
The values represent red, green, blue proportions.

FORMAT X'02'
The values represent Y values.

Each of these values is generated from the corresponding integers, I1, I2 and I3, and the corresponding maximum value, M1, M2 and M3, respectively.

Let I be an integer and M be the corresponding maximum value. Then the formulas defining this conversion are as follows:

```
If  I < M / 2  then  V =  I / (M + 1)
If  I ≥ M / 2  then  V =  (I + 1) / (M + 1)
```

These formulas produce values that *migrate* when the number of bits representing the color is increased. The values also map naturally using a *best* fit to the nearest fraction on an *equal step* device. This is illustrated in Table 3, which assumes a 3-bit representation (integers 0 through 7).

Table 3. Calculating color values

	Value		3-bit device fit	
Integer	**Fraction**	**Decimal**	**Decimal**	**Fraction**
0	0 / 8	0.000	0.0000	0 / 7
1	1 / 8	0.125	0.1429	1 / 7
2	2 / 8	0.250	0.2857	2 / 7
3	3 / 8	0.375	0.4286	3 / 7
4	5 / 8	0.625	0.5714	4 / 7
5	6 / 8	0.750	0.7143	5 / 7

Table 3. Calculating color values (continued)

Integer	Value		3-bit device fit	
	Fraction	**Decimal**	**Decimal**	**Fraction**
6	7 / 8	0.875	0.8571	6 / 7
7	8 / 8	1.000	1.0000	7 / 7

Example of index breakdown

The process of breaking down an index into three parts, as described under "Bit Generator self-defining parameter" on page 17, is illustrated in the following example.

Suppose that:

```
SIZE_1 = 2
SIZE_2 = 1
SIZE_3 = 3
```

Thus the maximum integer values are:

```
M1 = 3
M2 = 1
M3 = 7
```

Each index value is converted to a 6-bit number, and broken down into substrings of 2, 1, and 3 bits.

Table 4 shows some sample index values.

Table 4. Sample index values

Index	String	Substrings	I1	I2	I3
3	000011	00 0 011	0	0	3
9	001001	00 1 001	0	1	1
27	011011	01 1 011	1	1	3
45	101101	10 1 101	2	1	5

Interchange

In interchange, there are a number of limitations on the format of the color table:

- The only forms permitted are RGB LIST and RGB GENERATOR.
- If an RGB GENERATOR SDP is used, then that must be the only SDP.
- The maximum value permitted in each of SIZE_1, SIZE_2 and SIZE_3 is 8. Thus, each RGB component is limited to values 0 through 255.

Carrying color tables in MO:DCA-L data streams

Color tables are carried in MO:DCA-L as a required resource within the main resource group, or as a resource within the optional image objects.

The following tables show the required formats for the two cases:

Main resource group

Either a set of Element List self-defining parameters, one for each table entry, or a single Bit Generator self-defining Parameter is required. They are mutually exclusive.

Base part

Offset	Type	Name	Range	Meaning
0	BITS	FLAGS		
Bit 0			B'0'	Only valid value
Bit 1		RESET	B'1'	Reset LCD
Bits 2–7			B'000000'	Only valid value
1			X'00'	Only valid value
2	CODE	LCTID	X'00'	Local identifier of the color table

Element List self-defining parameter

Offset	Type	Name	Range	Meaning
0	UBIN	LEN	X'0F'	Length of this parameter
1	CODE		X'01'	Type: **01** Element List
2			X'00'	Only valid value
3	CODE	FORMAT	X'01'	RGB
4–6	UBIN	INDEX_1		Starting index for load
7	UBIN	SIZE_1	X'08'	Number of bits in component 1
8	UBIN	SIZE_2	X'08'	Number of bits in component 2
9	UBIN	SIZE_3	X'08'	Number of bits in component 3
10	UBIN	TRILEN	X'04'	Number of bytes in each element
11–14		ELEMENTS		Color elements

Bit Generator self-defining parameter

Offset	Type	Name	Range	Meaning
0	UBIN	LEN	X'0A'	Length of this parameter
1	CODE		X'02'	Type: **02** Bit Generator
2	BITS	FLAGS		
Bit 0		ASFLAGS	B'0'	Additive
Bit 1–7			B'0000000'	Only valid value
3	CODE	FORMAT	X'01'	RGB
4–6	UBIN	INDEX_1	X'000000'	Starting index for load
7	UBIN	SIZE_1	X'08'	Number of bits in component 1
8	UBIN	SIZE_2	X'08'	Number of bits in component 2
9	UBIN	SIZE_3	X'08'	Number of bits in component 3

Image resource group

A set of Element List self-defining parameters is allowed. These parameters must be ordered in ascending index order starting at index zero.

Base part

Offset	Type	Name	Range	Meaning
0	BITS	FLAGS		
Bit 0			B'0'	Only valid value
Bit 1		RESET	B'0'	Do not reset LCD
Bits 2–7			B'000000'	Only valid value
1			X'00'	Only valid value
2	CODE	LCTID	X'01'	Local identifier of the color table

Element List self-defining parameter

Offset	Type	Name	Range	Meaning
0	UBIN	LEN	X'0E'–X'FE'	Length of this parameter
1	CODE		X'01'	Type: **01** Element List
2			X'00'	Only valid value
3	CODE	FORMAT	X'01'	RGB
4–6	UBIN	INDEX_1		Starting index for load
7	UBIN	SIZE_1	X'08'	Number of bits in component 1
8	UBIN	SIZE_2	X'08'	Number of bits in component 2
9	UBIN	SIZE_3	X'08'	Number of bits in component 3
10	UBIN	TRILEN	X'03'	Number of bytes in each element
11–n		ELEMENTS		Color elements

Chapter 5. MO:DCA-L format definition

This chapter defines the MO:DCA-L data stream format.

For a complete definition of the MO:DCA-L structured fields and triplets that are not defined in Chapter 3, "MO:DCA-L structured fields," on page 7, see the *Mixed Object Document Content Architecture Reference*.

For information on the level of function required for the object content architectures as included in this interchange set, refer to the MO:DCA environment appendix in the following documents:

GOCA *Graphics Object Content Architecture Reference*, SC31-6804
IOCA *Image Object Content Architecture Reference*, SC31-6805

Data stream syntax structure

The groupings of MO:DCA and MO:DCA-L structured fields that follow identify those structured fields which appear within each begin-end structured field pair or state. This section specifies the structured fields allowed within a MO:DCA-L data stream and shows both the MO:DCA-L state hierarchy and the validity of structured fields within each state.

If a structured field that is not identified as being part of this interchange set appears anywhere within the data stream, a X'40' exception condition exists If a structured field appears within any state where it is not permitted, or if it appears out of the stated order or more than the permitted number of times, a X'20' exception condition exists. If a structured field that is identified as required does not appear within a specific state, a X'08' exception condition exists.

The conventions used in these structured field groupings are:

( ) The structured field acronym and identifier are shown in parentheses. The presence of dots or periods in the identifier indicates that the item is not a structured field, but instead is a structure, for example a page. The structure is composed of an assortment of structured fields, and is defined separately.

[ ] Brackets indicate optional structured fields. When a structured field is shown without brackets, it *must* appear between the begin and end structured fields.

+ Plus signs indicate structured fields may appear in any order relative to those that precede or succeed it except when the preceding or succeeding structured field does not have a plus (+) sign. Then the order is as listed.

(S) The enclosed (S) indicates that the structured field may be repeated. When present on a required structured field, at least one occurrence of the structured field is required, but multiple instances of it may occur.

F2 An F2 indicates that the structured field is a format two structured field. See *Mixed Object Document Content Architecture Reference* for further details.

Notes:

1. The Begin Document and End Document structured fields are required in a MO:DCA-L data stream.

2. The No Operation structured field may appear within any begin-end domain and thus is not listed in the structured field groupings.
3. The architecture that owns and controls the content of each of the data and resource objects carried in a MO:DCA-L data stream is identified in the following structured field groupings. Please refer to the referenced documentation for further details.
4. The Flag byte (byte 5) in the structured field introducer (SFI) must be set to X'00'. MO:DCA-L does not support SFI extension, structured field segmentation, or structured field padding.

Document

```
Begin Document  (BDT, D3A8A8)
        (      D3..C6)    Resource Group
End Document  (EDT, D3A9A8)
```

Figure 1. Document structure

Document Resource Group

```
Begin Resource Group  (BRG, D3A8C6)
  +     (      D3..77)    Color Attribute Table
  +  [  (      D3..FB)    Image Object                              (S)  ]
        (      D3..BB)    Graphics Object
End Document  (ERG, D3A9C6)
```

Figure 2. Document Resource Group structure

Color Attribute Table

```
Begin Color Attribute Table  (BCA, D3A877)
       (CAT,  D3B077)    Color Attribute Table
End Color Attribute Table  (ECA, D3A977)
```

Figure 3. Color Attribute Table structure

Image Object (IOCA FS20)

```
Begin Image Object  (BIM, D3A8FB)
     [  (      D3..C6)    Resource Group                                ]
     [  (      D3..C7)    Object Environment Group                      ]
        (IDD,  D3A6FB)    Image Data Descriptor
        (IPD,  D3EEFB)    Image Picture Data                              1
        (IPD,  D3EEFB)    Image Picture Data                       (S)    1
End Image Object  (EIM, D3A9FB)
```

Figure 4. Image Object structure

Note: Refer to the *Image Object Content Architecture Reference* for a full description of the IOCA FS20 content, syntax, and semantics for MO:DCA-L.

1. At least two IPD structured fields are *mandatory*. The first contains only the IPD parameters, while the second and any subsequent IPD structured fields contain the image data.

Image Resource Group

```
Begin Resource Group  (BRG, D3A8C6)
       (      D3..77)    Color Attribute Table
End Resource Group  (ERG, D3A9C6)
```

Figure 5. Image Resource Group structure

Object Environment Group (OEG) for Image Object

```
Begin Object Environment Group  (BOG, D3A8C7)
       (MCA,  D3AB77)    Map Color Attribute Table
End Object Environment Group  (EOG, D3A9C7)
```

Figure 6. Object Environment Group for Image Object structure

Graphics Object (GOCA DR/3V1)

```
Begin Graphics Object  (BGR, D3A8BB)
       (      D3..C7)    Object Environment Group
       (GDD,  D3A6BB)    Graphics Data Descriptor
       (GAD,  D3EEBB)    Graphics Data                                  (S)
End Graphics Object  (EGR, D3A9BB)
```

Figure 7. Graphics Object structure

Note: Refer to the *Graphics Object Content Architecture Reference*, SC31-6804, for a full description of the GOCA DR/3V1 content, syntax, and semantics for MO:DCA-L.

Object Environment Group (OEG) for Graphics Object

```
Begin Object Environment Group  (BOG, D3A8C7)
 +      (MCA,  D3AB77)    Map Color Attribute Table
 +      (MCF,  D3AB8A)    Map Coded Font                          F2  (S)    2
 +   [  (MDR,  D3ABC3)    Map Data Resource                           (S)  ]
End Object Environment Group  (EOG, D3A9C7)
```

Figure 8. Object Environment Group for Graphics Object structure

Permitted structured fields

This section describes the parameters and ranges of values supported for each of the structured fields contained in the MO:DCA-L format.

The structured fields are listed alphabetically and described using tables. The table heading for each structured field contains the structured field's acronym, its three-byte hexadecimal identifier, and its full name. If the structured field is defined in the MO:DCA-L document, the description also includes the page number in the document where a detailed description of the structured field can be

2. At least one MCF structured field is *mandatory* for the default font. A separate MCF structured field is required for each specific coded font desired.

found. If the structured field is not defined in the MO:DCA-L document, see the *Mixed Object Document Content Architecture Reference*.

Structured field parameters

In general, the structured field tables contain the following information for each parameter:

1. The offset from the beginning of the data portion of the structured field or from the beginning of the triplet.
2. Values and description:
   - When a specific parameter value is required, the specific value or the range of acceptable values is specified, followed by → and an explanation or description of the parameter.
   - When no specific value is required, or when a choice of values is required, the parameter name or a description of the parameter is given. If a choice of values is required, the choices are identified in the table.
3. Parameter occurrence is specified either as a lowercase *n* indicating that the occurrence is unlimited by the MO:DCA-L format, or as a number representing the maximum number of times the parameter may appear within the containing structured field, repeating group, or triplet.
4. Parameter optionality is specified as:
   O Optional. The parameter may or may not appear.
   M Mandatory. The parameter must always appear.
   C Conditional. The parameter is mandatory under certain conditions, but is optional or not allowed under other conditions.

Unless a specific order is required, triplets are listed in alphanumeric sequence by identifier; see the *Mixed Object Document Content Architecture Reference* for a definition of all triplets used in the MO:DCA-L format.

In general, no exception conditions are identified within the MO:DCA-L definition for the structured fields or their parameters. However, the following general rules apply:

- For those structured fields where a parameter order is stated, if a parameter appears outside that stated order, a X'01' exception condition exists .
- If a parameter value appears that is outside the range specified for that parameter, a X'02' exception condition exists.
- If a parameter that is identified as mandatory does not appear on a specific structured field, a X'04' exception condition exists.
- Unless otherwise stated, if any unrecognized parameter or triplet appears on any structured field, a X'10' exception condition exists.

Notes:

1. Any triplet encountered on any of the *Begin* structured fields listed below that is not explicitly defined as being valid for that structured field should be ignored and should not cause an exception condition.
2. If specified, the name contained in the name parameter on an *End* structured field must match that specified in the name parameter on its matching *Begin* structured field, or a X'01' exception condition exists.

Begin Color Attribute Table

BCA X'D3A877' Begin Color Attribute Table (See "Begin Color Attribute Table (BCA)" on page 8)			
0–7	Color Attribute Table name (8 characters)	1	M

Begin Document

BDT X'D3A8A8' Begin Document (See the *Mixed Object Document Content Architecture Reference***)**			
0–7	Document name (8 characters)	1	M
8–9	**X'0000'** → Reserved, must be binary zero	1	M
10–*n*	The following triplets, in any order:		
	Coded Graphic Character Set Global Identifier Triplet (See the *Mixed Object Document Content Architecture Reference*)	1	M
	0–1 **X'0601'** → Triplet length and identifier	1	M
	2–5 **X'03AA0352'** → Character set and code page identification (character set 938, code page 850)	1	M
	MO:DCA Interchange Set Triplet (See the *Mixed Object Document Content Architecture Reference*)	1	M
	0–1 **X'0518'** → Triplet length and identifier	1	M
	2 **X'03'** → Interchange set type, resource	1	M
	3–4 **X'0C00'** → Interchange set identifier (MO:DCA-L)	1	M
	Comment Triplet (See the *Mixed Object Document Content Architecture Reference*)	1	O
	0–1 **X'***nn***65'** → Triplet length and identifier	1	M
	2–*n* Comment used for metafile description of up to 252 bytes	1	M

Begin Graphics Object

BGR X'D3A8BB' Begin Graphics Object (See the *Mixed Object Document Content Architecture Reference***)**			
0–7	Graphics Object name (8 characters)	1	M

Begin Image Object

BIM X'D3A8FB' Begin Image Object (See the *Mixed Object Document Content Architecture Reference***)**			
0–7	Image Object name (8 characters)	1	M

Begin Object Environment Group

BOG X'D3A8C7' Begin Object Environment Group (See the *Mixed Object Document Content Architecture Reference***)**			
0–7	Object Environment Group name (8 characters)	1	M

Begin Resource Group

BRG X'D3A8C6' Begin Resource Group (See the *Mixed Object Document Content Architecture Reference***)**			
0–7	Resource Group name (8 characters)	1	M

Color Attribute Table

CAT X'D3B077' Color Attribute Table (See "Color Attribute Table (CAT)" on page 10)			
0–*n*	Color Attribute Table data as defined in Chapter 4, "MO:DCA-L color attribute table," on page 15		

End Color Attribute Table

ECA X'D3A977' End Color Attribute Table (See "End Color Attribute Table (ECA)" on page 11)			
0–7	Color Attribute Table name (8 characters)	1	M

End Document

EDT X'D3A9A8' End Document (See the *Mixed Object Document Content Architecture Reference*)			
0–7	Document name (8 characters)	1	M

End Graphics Object

EGR X'D3A9BB' End Graphics Object (See the *Mixed Object Document Content Architecture Reference*)			
0–7	Graphics Object name (8 characters)	1	M

End Image Object

EIM X'D3A9FB' End Image Object (See the *Mixed Object Document Content Architecture Reference*)			
0–7	Image Object name (8 characters)	1	M

End Object Environment Group

EOG X'D3A9C7' End Object Environment Group (See the *Mixed Object Document Content Architecture Reference*)			
0–7	Object Environment Group name (8 characters)	1	M

End Resource Group

ERG X'D3A9C6' End Resource Group (See the *Mixed Object Document Content Architecture Reference*)			
0–7	Resource Group name (8 characters)	1	M

Graphics Data

GAD X'D3EEBB' Graphics Data (See the *Mixed Object Document Content Architecture Reference*)	
0–*n*	Up to 32759 bytes of graphics data as defined by GOCA DR/3V1

Graphics Data Descriptor

GDD X'D3A6BB' Graphics Data Descriptor (See the *Mixed Object Document Content Architecture Reference*)	
0–*n*	Graphics descriptor data as defined by GOCA DR/3V1

Image Data Descriptor

IDD X'D3A6FB' Image Data Descriptor (See the *Mixed Object Document Content Architecture Reference***)**	
0–*n*	Image descriptor data as defined by IOCA FS20

Image Picture Data

IPD X'D3EEFB' Image Picture Data (See the *Mixed Object Document Content Architecture Reference***)**	
0–*n*	Up to 32759 bytes of image segment data as defined by IOCA FS20

Note: At least two IPD structured fields are *mandatory*. The first contains only the IPD parameters while the second (and any subsequent ones) contain the image data.

Map Coded Font, format 2

MCF X'D3AB8A' Map Coded Font (See the *Mixed Object Document Content Architecture Reference***)**				
0–1		**X'00***nn*' → Length of this repeating group	1	M
Note: Only one repeating group is permitted on this structured field.				
2–*n*		The following triplets, *in the order specified*:		
		Fully Qualified Name Triplet (See the *Mixed Object Document Content Architecture Reference*)	1	M
	0–1	**X'0C02'** → Triplet length and identifier	1	M
	2–3	**X'8400'** → FQN type and format, reference to coded font	1	M
	4–11	External name of the coded font	1	M
Note: The coded font name is the eight-character name supplied by the CPI call. The default name is indicated by the use of a X'FF' as the first character of the name. The default name is always mapped to LID X'00'.				
		Resource Local Identifier Triplet (See the *Mixed Object Document Content Architecture Reference*)	1	M
	0–1	**X'0424'** → Triplet length and identifier	1	M
	2	**X'05'** → Resource type, coded font	1	M
	3	**X'01'–X'FE'** → Resource Local Identifier. It must be in the range of 1 to 254.	1	M
		Font Descriptor Specification Triplet (See the *Mixed Object Document Content Architecture Reference*)	1	C
Note: This triplet is *mandatory* on all Map Coded Font structured fields other than the one for the default font. It is *not permitted* on the Map Coded Font structured field that specifies the default font.				
	0–1	**X'141F'** → Triplet length and identifier	1	M
	2	Font Weight Class. It must be one of the following: **X'05'** → Medium (normal) **X'07'** → Bold	1	M
	3	**X'05'** → Font Width Class of medium (normal)	1	M
	4–5	**X'0001'–X'7FFF'** → Font Height. It must be in the range of 1 to 32767 in world coordinate units.	1	M

MCF X'D3AB8A' Map Coded Font (See the *Mixed Object Document Content Architecture Reference*)				
	6–7	**X'0001'–X'7FFF'** → Font Width. It must be in the range of 1 to 32767 in world coordinate units.	1	M
	8	Font Descriptor Flags, as follows: **Bits** **Description** **0** Italics **1** Underscored **2** Reserved, must be zero **3** Hollow **4** Overstruck **5** Proportional **6** Kerned characters (pairwise) **7** Reserved, must be zero	1	M
	9–18	Reserved	1	M
	19	Font Flags, as follows: **Bits** **Description** **0** Reserved, must be set to zero **1** Font type **0** Bitmapped font **1** Outline (vector) font **2** Transform font **0** Font will not be transformed **1** Font may be transformed (scaled, rotated, sheared) **3–7** Reserved, must be zero	1	M
	Font Coded Graphic Character Set Global Identifier Triplet (See the *Mixed Object Document Content Architecture Reference*)		1	M
	0–1	**X'0620'** → Triplet length and identifier	1	M
	2–5	The GCSGID and CPGID for the font.	1	M
	Fully Qualified Name Triplet (See the *Mixed Object Document Content Architecture Reference*)		1	C
Note: This triplet is *mandatory* on all Map Coded Font structured fields other than the one for the default font. It is *not permitted* on the Map Coded Font structured field that specifies the default font.				
	0–1	**X'2402'** → Triplet length and identifier	1	M
	2–3	**X'0800'** → FQN type and format, typeface identifier	1	M
	4–35	External name of the font typeface. It must be 32 bytes in length.	1	M

Note: At least one Map Coded Font structured field is *mandatory* for the default font. The default font is indicated by the use of a X'FF' as the first byte of the coded font name in the Begin Resource Object Reference Fully Qualified Name triplet. The default font is always mapped to a X'00' local identifier. The Font Descriptor Specification triplet and the Fully Qualified Name triplet for the font typeface are not permitted on the Map Coded Font structured field for the default font.

Map Color Attribute Table

MCA X'D3AB77' Map Color Attribute Table (See "Map Color Attribute Table (MCA)" on page 12)				
0–1	**X'00***nn*'** → Length of this repeating group is either 14 or 18 bytes		254	M
2–*n*	The following triplets, in any order:			

MCA X'D3AB77' Map Color Attribute Table (See "Map Color Attribute Table (MCA)" on page 12)				
	Fully Qualified Name Triplet (See the *Mixed Object Document Content Architecture Reference*)		1	M
	0–1	**X'0C02'** → Triplet length and identifier	1	M
	2–3	**X'8400'** → FQN type and format, reference to color table	1	M
	4–11	External name of the color table	1	M
	Resource Local Identifier Triplet (See the *Mixed Object Document Content Architecture Reference*)		1	C
Note: This triplet is *mandatory* for image, *not permitted* for graphics.				
	0–1	**X'0424'** → Triplet length and identifier	1	M
	2	**X'07'** → Color table resource type	1	M
	3	**X'01'–X'FE'** → Resource Local Identifier. It must be in the range of 1 to 254.	1	M

Map Data Resource

MDR X'D3ABC3' Map Data Resource (See the *Mixed Object Document Content Architecture Reference*)				
0–1	**X'0015'** → Length of this repeating group is 21 bytes		*n*	M
2–*n*	The following triplets, in any order:			
	Fully Qualified Name Triplet (See the *Mixed Object Document Content Architecture Reference*)		1	M
	0–1	**X'0C02'** → Triplet length and identifier	1	M
	2–3	**X'8400'** → FQN type and format, reference to image object	1	M
	4–11	External name of the image object	1	M
	Extended Resource Local Identifier Triplet (See the *Mixed Object Document Content Architecture Reference*)		1	M
	0–1	**X'0722'** → Triplet length and identifier	1	M
	2	**X'10'** → Resource type, image	1	M
	3–6	**X'00000000'–X'FFFFFFFF'** → Resource Local Identifier (used as bitmap handle)	1	M

No Operation

NOP X'D3EEEE' No Operation (See the *Mixed Object Document Content Architecture Reference*)	
0–*n*	Up to 32,759 bytes of data

Notices

This information was developed for products and services offered in the U.S.A.

InfoPrint Solutions Company may not offer the products, services, or features discussed in this document in other countries. Consult your local InfoPrint Solutions Company representative for information on the products and services currently available in your area. Any reference to an InfoPrint Solutions Company product, program, or service is not intended to state or imply that *only* that InfoPrint Solutions Company product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any InfoPrint Solutions Company intellectual property rights may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-InfoPrint Solutions Company product, program, or service.

InfoPrint Solutions Company may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

InfoPrint Solutions Company, LLC
6300 Diagonal Hwy 002J
Boulder, CO 80301-9270
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the InfoPrint Solutions Company Intellectual Property Department in your country or send inquiries, in writing, to:

InfoPrint Solutions Company, LLC
6300 Diagonal Hwy 002J
Boulder, CO 80301-9270
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INFOPRINT SOLUTIONS COMPANY PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. InfoPrint Solutions Company may make improvements and/or changes in the product(s) described in this publication at any time without notice.

Any references in this information to non-InfoPrint Solutions Company Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this InfoPrint Solutions Company product and use of those Web sites is at your own risk.

InfoPrint Solutions Company may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

InfoPrint Solutions Company, LLC
6300 Diagonal Hwy 002J
Boulder, CO 80301-9270
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by InfoPrint Solutions Company under terms of the InfoPrint Solutions Company Customer Agreement, InfoPrint Solutions Company International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-InfoPrint Solutions Company products was obtained from the suppliers of those products, their published announcements or other publicly available sources. InfoPrint Solutions Company has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-InfoPrint Solutions Company products. Questions on the capabilities of non-InfoPrint Solutions Company products should be addressed to the suppliers of those products.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to InfoPrint Solutions Company, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. InfoPrint Solutions Company, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

References in this document to InfoPrint Solutions Company products, product features, programs or services do not imply that InfoPrint Solutions Company intends to make such products, product features, programs or services available in all countries in which InfoPrint Solutions Company operates or does business.

Trademarks

These terms are trademarks or registered trademarks of Ricoh Co., Ltd., in the United States, other countries, or both:

- Advanced Function Presentation
- AFP
- AFP Consortium
- AFPC
- InfoPrint
- Infoprint
- Mixed Object Document Content Architecture
- MO:DCA
- Ricoh

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Other company, product, or service names may be trademarks or service marks of others.

RICOH | IBM

InfoPrint Solutions Company™

Printed in USA

S550-1135-00