

Chapter 18 Adobe Acrobat Support

An Introduction to Adobe Acrobat™

Adobe Acrobat provides significant capability for publishing documents in electronic form. Fully formatted documents may be sent to other computers regardless of hardware platform, operating system or application software used to create the original.

The key element in this capability is the Portable Document Format (PDF) which is based on PostScript. A PDF file describes a document of any length, containing any combination of text and graphics in a device- and resolution-independent format. The recipient of a PDF file may view the document on his computer monitor and print a single page or a range of pages from Acrobat. Depending on the PostScript printer used, Acrobat may convert the PDF file to PostScript on the fly for printing.

Additional features of Acrobat are bookmarks, links, form XObjects and “sticky” notes. Bookmarks provide a visible “outline” of the document at the side of the document being displayed on the computer monitor. Bookmarks provide access to any page that contains a bookmark. The form XObject facility provides for incorporating a “page template”, or overlay, once or multiple times on one page or a number of pages. Sticky notes provide a means of attaching notes that are visible to the viewer on his monitor but which are discarded when the document is printed.

Acrobat is comprised of three basic programs that provide the following functions:

- *Distiller*: translates PostScript files to PDF format. Distiller is available in a Personal, single user, version or a Network, multiple-user, version for Windows and Macintosh.
- *Exchange*: provides facilities to view, collate, navigate and print PDF files. The user may create bookmarks, links and “sticky notes” as well. It is available for Windows and Macintosh.
- *Reader*: provides facilities to view, navigate and print PDF files. It is available for Windows, Macintosh and DOS. Reader is a sub-set of Exchange.

Bookmarks and Links

Bookmarks provide ready access to “bookmarked” points in a document rendered in PDF format. Each bookmark accesses a different page or a different view of a page. Bookmarks may be placed (by your Versacomp application program) at the top of each page, or they may be placed in some logical manner based on the logical structure of the document being composed. For example, bookmarks may be placed on side heads. Their placement may be guided by the structure of an SGML-coded document, for example.

The LINK statement provides a means to generate links within a Portable Document Format file being generated or between a Portable Document Format file and another document. The external document to be linked may be another Portable Document Format file, or it may be a Universal Resource Locator.

The Versacomp PDF and Dual drivers make generation of the PDF file possible from Versacomp application programs. Versacomp now supports the placement of bookmarks and links in a PDF file. Bookmarks may be nested; ten levels of bookmarks are provided. Comments, called *pdf marks*, that relate to bookmarks and links are placed in PostScript files generated by the PostScript and Dual drivers. If these PostScript files are submitted to Adobe Distiller, the generated PDF file will translate the pdf marks to bookmarks and links appropriately.

Adobe Acrobat Support

Form XObjects

A form XObject is a self-contained description of any text or graphics that is to be rendered multiple times on several pages or at different locations on a single page.

Form XObjects offer the very large advantage that a single form XObject, no matter how many times it appears in a composed document, appears only once in the PDF file. This is advantageous from the viewpoint of file size and related transmission time to a printer or elsewhere. This Acrobat feature is supported by *Versacomp Overlays*.

Versacomp Support for Acrobat

Three instructions support Adobe Acrobat bookmarks:

- **BOOKMARK** – an instruction that may be used to place “bookmarks” in a PDF file generated by the PDF driver or the Dual driver, such that Adobe Acrobat may display an “outline” which will provide instant access to selected pages no matter how long the document being displayed on a computer monitor may be. The instruction also places information lines in the PostScript file generated by the PostScript driver and the Dual driver that will be converted to Acrobat bookmarks if the PostScript file is processed by the Acrobat Distiller. These information lines are ignored by PostScript printers and interpreters that may process the PostScript file.
- **SAVPOS** – provides the means for Versacomp to adjust saved horizontal and vertical positions to account for horizontal movement made by **COMPX** and vertical movement caused by **VERJUS** at the close of lines and blocks, respectively. Depending upon the nature of the bookmarks being generated, accounting for such movement is important in the generation of bookmarks.
- **\$SWITCH** – provides the means to control certain aspects of the composer during execution of an application program. One of the options provided controls the display of bookmarks when a PDF file is opened under Adobe Acrobat.

Two instructions support Adobe Acrobat links:

- **LINK** – an instruction that provides a means to generate links within a Portable Document Format file being generated or between a Portable Document Format file and another document. The external document to be linked may be another Portable Document Format file, or it may be a Universal Resource Locator on the World Wide Web.
- **SAVPOS** – provides the means for Versacomp to adjust saved horizontal and vertical positions to account for horizontal movement made by **COMPX** and vertical movement caused by **VERJUS** at the close of lines and blocks, respectively. Depending upon the nature of the links being generated, accounting for such movement is important in the generation of link annotations.

One instruction supports the Acrobat open action feature:

- **OPENACT** – provides user control over the initial display when the PDF document generated by the PDF driver or the Dual driver is opened under Acrobat. It provides control over bookmark display, page displayed at open and the display mode and zoom factor of the page opened. The instruction also places information lines in the PostScript file generated by the PostScript driver and the Dual driver that will be converted appropriately if the PostScript file is pro-

cessed by Acrobat Distiller to generate a PDF file. These information lines are ignored by PostScript printers and interpreters that may process the PostScript file.

One instruction supports the Acrobat document information feature:

- **DOCINFO** – provides the facility to place descriptive information about a document being composed into a PDF file generated by the PDF driver or the Dual Driver. Such information may then be viewed by the user when the document is open under Acrobat. The instruction also places information lines in the PostScript file generated by the PostScript driver and the Dual driver that will be converted appropriately if the PostScript file is processed by Acrobat Distiller to generate a PDF file. These information lines are ignored by PostScript printers and interpreters that may process the PostScript file.

One descriptor supports the Acrobat crop box feature:

- **\$VIEW** – provides the facility to suppress everything outside the type page. A display document may have wide white margins which occupy space on the monitor when the document is displayed under Acrobat. This results in a much smaller magnification factor and a display that is difficult to read. Using the crop box feature together with the zoom feature of the **OPENACT** statement results in a display that is more easily read when the document is open under Acrobat. The entire page, including the wide margins, is printed if the file is printed under Acrobat.

Four instructions support Adobe Acrobat form XObjects:

- **OVLVBEG** – the instruction that declares the start of a new mode for Versacomp: that of generating Versacomp overlays that may subsequently be incorporated one or multiple times in one or many PDF documents.
- **OVLVEND** – the instruction that signals the end of overlay generation and causes the Versacomp overlay to be written to a disk file.
- **OVLVCALL** – the instruction that provides for incorporating an existing Versacomp overlay into a document being composed.
- **OVLVINFO** – an instruction that may be used to retrieve and store information about a Versacomp overlay that will subsequently be merged using the **OVLVCALL** instruction.

Overlay Limitations

The size and number of Versacomp overlays that may be included in a PDF file are limited by the amount of memory available on the target printer.

Multiple tiff files may be used in overlays. A maximum sum of 1000 tiff files may be called in all of the overlays in a single composition job. One thousand tiff files may be called in a single overlay.

A maximum number of 1000 fonts may be used in all of the overlays in a single composition job.

A maximum of 1000 overlays may be used in a single composition job.

BOOKMARK STATEMENT

The BOOKMARK instruction is used to place “bookmarks” in a PDF file generated by the PDF driver or the Dual driver, such that Adobe Acrobat may display an “outline” which will provide instant access to selected pages no matter how long the document being displayed on a computer monitor may be. The instruction also places information lines in the PostScript file generated by the PostScript driver and the Dual driver that will be converted to Acrobat bookmarks if the PostScript file is processed by the Acrobat Distiller. These information lines are ignored by PostScript printers and interpreters that may process the PostScript file.

The BOOKMARK statement may be used only with the PostScript, PDF and Dual drivers.

In some circumstances and with several display-related options provided by the BOOKMARK statement, the vertical, and horizontal, positions on a page at the time the statement is issued must be considered. For example, the “left-top-zoom” display option requires the X- and Y-coordinates of the point on the page that is to occupy the upper-left corner of the monitor when the page is displayed under Acrobat. The “width fits the window” display option requires the Y-coordinate on the page that is to be at the top of the monitor when the page is displayed. CY figures into specification of the vertical position desired; whether the line to appear at the top of the display was started by a LINE or a LEAD statement makes a difference.

Bookmarks may be placed “on-the-fly” as pages are composed or appropriate data may be saved such that bookmarks for the entire composition job may be issued at the end of job.

Note: the eight optional keywords and their related arguments below are mutually exclusive. One of the eight must be specified. No two may be specified together.

BOOKMARK a,[,F=b][,C=(c,d,e)]L=f,P=g,O=h[,T=i][,N=j][,Z=(k,l,m)]
[,F]
[,H=k]
[,V=k]
[,R=(k,l,m,n)]
[,BF]
[,BH=k]
[,BV=k]

a - Title String

Specifies the name of the storage area or field to be used to cite the title expression to be associated with a bookmark. Title may be up to 240 characters in length and any character present in the PDF Document Encoding scheme. See Appendix A. A pointer variable may be used. Subscripting is permitted. Trailing blanks are not suppressed. Application programs should delimit this argument to save storage and, more importantly, output file size.

When a PDF document is opened under Acrobat and bookmarks are displayed on the monitor, the screen width is divided into two areas: a narrow band at the left for bookmarks and a wider band at the right for the page displayed.

Under these circumstances the maximum number of bookmark caption characters viewable varies from 80 to 160 depending on the computer monitor used

and the characters that appear in a particular caption. The viewer may modify screen allocation such that much more horizontal space is allocated to bookmarks and much longer captions become visible.

b - Title Font Style

Specifies the font style to be used for the Bookmark title string. May have one of the following values:

0 - Regular

The Bookmark title string will be composed in regular type. This is the default if the keyword “F” is omitted.

1 - Italic

The Bookmark title string will be composed in italic type.

2 - Bold

The Bookmark title string will be composed in bold type.

3 - Bold Italic

The Bookmark title string will be composed in bold italic type.

- Title Font Color

The color to be used for the Bookmark title string is specified as the individual RGB components as follows:

c - Red Component

A 32-bit binary number in the range zero through 1000, both inclusive. Zero specifies no red contribution to the font color and 1000 specifies maximum intensity of that color.

d - Green Component

A 32-bit binary number in the range zero through 1000, both inclusive. Zero specifies no green contribution to the font color and 1000 specifies maximum intensity of that color.

e - Blue Component

A 32-bit binary number in the range zero through 1000, both inclusive. Zero specifies no blue contribution to the font color and 1000 specifies maximum intensity of that color.

f - Level Number

Specifies the level number of the bookmark. May be a constant or a variable in the range 1 through 10, inclusive.

g - Page Number

Specifies the sequential page number (not the page number that may be typeset on the page); the first page of the document has the number one. May be a variable.

h - Open Flag

Specifies whether this level is to be opened at display time. May be either of the Reserved words “Y” or “N” or a variable that contains the related value.

Adobe Acrobat Support BOOKMARK

i - Bookmark Type

A constant or a variable that specifies the type of bookmark to be generated. May have one of the following values:

- 0 - Normal Bookmark
This is the default if the keyword "T" is omitted.
- 1 - Reserved
This value will be treated as zero.
- 2 - External File
Specifies that the bookmark will link to an external file (PDF or non-PDF).
- 3 - Web Target
Specifies that the bookmark will link to a Universal Resource Locator (URL).
- 4 - Uniform Resource Identifier
Specifies that the bookmark statement is to generate a base document-level Uniform Resource Identifier (URI). See File Specifications in the LINK statement description (Table 18-1).

j - Pathname

Specifies the path name of the PDF file or URL cited in argument "e" above. This keyword and pathname are required only for bookmark types 2, 3 and 4. Subscripting is permitted. See Table 18-1 for file specification examples, and below for pathname requirements for bookmark types 2 through 4:

- 2 - Link to Another PDF File.
The full pathname of the link object is required unless the target PDF file is in the same directory with PDF file being generated. In the latter case the target filename alone is sufficient.
- 3 - Target a URL.
May be a full pathname to a Universal Resource Locator, or it may be the continuing and unique part of individual pathnames when a Base Uniform Resource Identifier has been issued. See type 4 below.
- 4 - Specification of a Base URI.
A Base Uniform Resource Identifier is the common leading part of many longer pathnames. If a Base URI has been issued, unique parts of individual pathnames may be issued. See Type 3, above. In cases of this sort, Acrobat concatenates the Base URI and link pathnames to generate full pathnames for individual links.

If a URI has been established, the related argument to N= in all Type 3 BOOKMARK statements, that are NOT complete pathnames in and of themselves, will be concatenated with the current URI by Acrobat.

Note: that only one URI may be in force at any given time. The last URI specified, whether using a BOOKMARK statement or a LINK statement, will be in force.

Trailing blanks are not suppressed. Application programs should delimit this argument to save storage and, more importantly, output file size.

The eight display-related view specifications provided are described below. One and only one may be specified when the BOOKMARK instruction is issued. If this keyword is omitted, the default value is "F"

1 - Z=(k,l,m)

The "left top zoom" option specifies the coordinates of the point on the page that is to occupy the upper-left corner of the window and the zoom factor to be in effect when the page is displayed. **Note:** The value zero for any of these three arguments specifies that no change from the preceding bookmark is to be made.

k - Left

Specifies the X-component of the point on the page that is to be at the top-left corner of the display window when the page is displayed under Acrobat. X-position must be stated in tenth point units.

l - Top

Specifies the Y-component of the point on the page that is to be at the top-left corner of the display window when the page is displayed under Acrobat. Vertical position must be stated in tenth point units.

m - Zoom

Specifies the zoom factor to be in force when the page is displayed in the window. The zoom value specified is a percentage number and must be in the range 8 to 3200, inclusive. These limits are not fixed; they vary with size of the page being displayed, as well as of pages previously viewed in the current file.

The value zero is allowed, as well, to specify that zoom is to be the same as the current page. When Zoom is set to zero, Acrobat ignores the left and top values supplied in this statement. The values used on the target page will be the same as those in effect for the current page.

A discussion of Acrobat General Preferences and Document Information, as they relate to document magnification, appears in the OPENACT description.

2 - F

Specifies that the page displayed is to be scaled to fit the window.

3 - H=k

Specifies that the page displayed is to be scaled such that the width fits the window. Full page depth may or may not be visible.

k- Top

Specifies the Y-position on the page that is to be at the top edge of the display window. Y-position must be stated in tenth point units.

4 - V=k

Specifies that the page displayed is to be scaled such that the height fits the window.

Adobe Acrobat Support BOOKMARK

k - Left

Specifies the X-position on the page that is to be at the left edge of the display window. X-position must be stated in tenth point units.

5 - R=(k,l,m,n)

Specifies a rectangular area of the page that is to be displayed on the monitor when the page is opened. The area displayed is defined as the horizontal and vertical coordinates of its lower-left and upper-right corners.

k - Left

Specifies X-component of the point on the page that is to be at the lower-left corner of the display window when the page is displayed under Acrobat. X-position must be stated in tenth point units.

l - Bottom

Specifies Y-component of the point on the page that is to be at the lower-left corner of the display window when the page is displayed under Acrobat. Y-position must be stated in tenth point units.

m - Right

Specifies X-component of the point on the page that is to be at the upper-right corner of the display window when the page is displayed under Acrobat. X-position must be stated in tenth point units.

n - Top

Specifies Y-component of the point on the page that is to be at the upper-right corner of the display window when the page is displayed under Acrobat. Y-position must be stated in tenth point units.

6 - BF

Specifies that the page displayed is to be scaled such that its bounding box fits entirely within the window both horizontally and vertically.

7 - BH=k

Specifies that the page is to be displayed such that the width of its bounding box fits within the window. Full page depth may or may not be visible.

k - Top

Specifies the Y-position on the page that is to be at the top edge of the display window. Y-position must be stated in tenth point units.

8 - BV=k

Specifies that the page is to be displayed such that the depth of its bounding box fits within the window. Full page width may or may not be visible.

k - Left

Specifies the X-position on the page that is to be at the left edge of the display window. X-position must be stated in tenth point units.

Examples:

1) Normal Bookmark

Perhaps the simplest bookmark that can be issued is “scale the page to fit the window” when the file is opened under Acrobat. The five arguments to this instruction are described below.

```
BOOKMARK    TTLSTG , L=GV51 , P=PNUM , O=GV52 , F
```

- a - The storage area TTLSTG is the data area that contains the title string to be associated with the bookmark generated.
- b - Bookmark level number is specified by the integer stored in GV51. It must be in the range 1 through 10, inclusive and it must be no more than one larger than the bookmark it follows.
- c - Sequential page number is maintained in the user variable PNUM.
- d - Whether the level cited is to be opened at the time the PDF document is opened under the display operating system depends on the reserved word, “Y” or “N”, stored in GV52.
- e - The display is specified to be scaled such that the entire image fits the screen.

The keywords “F” and “C” are both absent. The font used for bookmark title strings will be the regular font; color will be black.

2) Bookmark Targets a URL

TTLSTG, GV51, PNUM and GV52 are as defined in Example 1. The other arguments are described below.

MAKE	FNT=2	Font Style bold.
MAKE	RED=1000	Maximum red component.
MAKE	GREEN=0	Zero green component.
MAKE	BLUE=0	Zero blue component.
MAKE	KIND=3	Bookmark is a URL.

```
BOOKMARK    TTLSTG ( , GV61 ) , F=FNT , C = ( RED , GREEN , BLUE ) , @  
L=GV51 , P=PNUM , O=GV52 , T=KIND , N=PATH ( , GV62 )
```

- a - The user variable FNT specifies that the bookmark text will appear in bold text.
- b - The values assigned to the user variables RED, GREEN and BLUE specify that the bookmark text will be red.
- c - The value assigned to the user variable KIND specifies that the target of the bookmark is a URL.
- d - The storage area PATH is the data area that contains the full path to the URL that is the target of the bookmark.

Note that both TTLSTG and PATH are delimited by the general variables GV61 and GV62, respectively. Versacomp does not suppress trailing blanks in either.

Adobe Acrobat Support BOOKMARK

Notes:

- 1) The OPENACT statement may be used to specify whether bookmarks are displayed when a PDF file generated by the PDF driver or the Dual driver is opened under Acrobat.

The OPENACT statement also causes information lines to be placed in the PostScript file generated by the PostScript driver or the Dual driver. These lines will be converted appropriately if the PostScript file is processed by Acrobat Distiller to generate a PDF file.

These information lines are ignored by PostScript printers and interpreters that may process the PostScript file.

- 2) The \$SWITCH descriptor provides the facility to specify that bookmarks are displayed, or not displayed, when the PDF file is opened under Adobe Acrobat.

LINK STATEMENT

The LINK statement provides a means to generate links within a Portable Document Format file being generated or between a Portable Document Format file and another document. The external document to be linked may be another Portable Document Format file, or it may be a Universal Resource Locator.

If links are to a Universal Resource Locator, the link path may be separated into two parts. The first part is called the Base Uniform Resource Identifier (URI) which is the common leading part of many longer pathnames. The second part is the continuing and unique part of individual pathnames. When links are expressed in this way, Acrobat concatenates the Base URI and the link pathname to generate full pathnames for individual links.

Caveat: The full extent of the Acrobat facilities provided by this statement are available only using the latest version of Adobe Acrobat™.

A link statement that defines a link annotation must be issued while the source page is open even though the destination page, or view information relating to the destination page, may be unknown. Therefore:

- 1) Issue the LINK on the page, and
- 2) update the LINK after EPAGE/SPAGE or end of job.

A link in this class may be subsequently updated.

A link annotation is a rectangular area on the source page that is defined by a border which may be a rectangle, may have round corners, may be colored or may be invisible. The annotation is only useful within Adobe Acrobat. The annotations are not printed when a page that carries annotations is printed.

LINK TYPE=a[,U=b][,N=c][,D=d][,R=(e,f,g,h)][,B=(i,j,k,l)][,C=(m,n,o)][,V=(Z,p,q,r)]
[,V=(F)]
[,V=(H,p)]
[,V=(V,p)]
[,V=(R,p,q,r,s)]
[,V=(BF)]
[,V=(BH,p)]
[,V=(BV,p)]

a - Link Type

A constant or a variable that specifies the type of link to be made. May have one of the following values:

- 0 - No link is to be made. The statement is treated as a No operation. The System sets the System Variable CRC to zero.
- 1 - The link is to a destination in the same document.
- 2 - The link is external; to another PDF file.
- 3 - The link is external; to a Universal Resource Locator (URL).
- 4 - Specifies a base document-level Uniform Resource Identifier annotation.

Adobe Acrobat Support LINK

b - Update Key

When a LINK statement is issued, the System returns a unique “link key” in the System Variable CRC. This key provides the facility to subsequently update a link issued earlier.

For example, a link statement must be issued on the page that contains a link annotation. But all of the information required for the final link may not yet be known. The destination page may be unknown; the view parameters on the destination page may be unknown.

When the destination page that relates to a link issued earlier is known, that link may be updated by using that link key as an argument to this keyword in the update LINK statement. The link type in the update LINK statement must be the same as the type in the initial LINK statement.

The keywords: N=, D=, R=, B=, C= and V= may be changed when the LINK is updated. If N= is to be updated, the initial LINK statement must contain a dummy argument that is at least as long as the path name that is to be supplied in the updated LINK statement.

c - Pathname of Object of Link

Specifies the complete path name of the PDF file or URL cited in argument “a” above. This keyword and pathname are required only for link types 2, 3 and 4. Subscripting is permitted. The pathname must be supplied in ASCII and must meet system defined formats. See Figure 18-1 for file specification examples, and below for pathname requirements for link types 2 through 4:

2 - Link to Another PDF File.

The full pathname of the link object is required.

3 - Link to a URL.

May be a full pathname to a Universal Resource Locator, or it may be the continuing and unique part of individual pathnames when a Base Uniform Resource Identifier has been issued. See type 4 below.

4 - Specification of a Base URI.

A Base Uniform Resource Identifier is the common leading part of many longer pathnames. If a Base URI has been issued, unique parts of individual pathnames may be issued. See Type 3, above. In cases of this sort, Acrobat concatenates the Base URI and link pathnames to generate full pathnames for individual links.

When Link Type is 4, no other keywords with related arguments than TYPE= and N= are required. The link simply establishes a URI which Acrobat may subsequently use.

If a URI has been established, the related argument to PATH= in all subsequent Type 3 LINK statements, that are NOT complete pathnames in and of themselves, will be concatenated with the current URI by Acrobat.

Note: that only one URI may be in force at any given time. The last URI specified, whether using a BOOKMARK statement or a LINK statement, will be in force.

Trailing blanks are not suppressed. Application programs should delimit this argument to save storage and, more importantly, output file size.

d - Destination Page Number

Specifies the sequential page number (not the page number that may be typeset on the page) in the target PDF file to which the link points; the first page of that document has the number one. May be a variable.

The next four keywords and arguments specify page-relative co-ordinates.

e - Annotation Rectangle, Lower-Left X

Specifies the horizontal position of the lower-left corner of the area to be designated as the link annotation, in tenth-point units.

f - Annotation Rectangle, Lower-Left Y

Specifies the vertical position of the lower-left corner of the area to be designated as the link annotation, in tenth-point units.

g - Annotation Rectangle, Upper-Right X

Specifies the horizontal position of the upper-right corner of the area to be designated as the link annotation, in tenth-point units.

h - Annotation Rectangle, Upper-Right Y

Specifies the vertical position of the upper-right corner of the area to be designated as the link annotation, in tenth-point units.

i - Annotation Border (Reserved)

A reserved keyword that must now be given the value zero.

j - Annotation Border (Reserved)

A reserved keyword that must now be given the value zero.

k - Annotation Border Rule Width

Specifies the thickness of the rules that make the box that marks the link annotation in tenth-point units. If the value specified is negative, the border will be invisible.

l - Annotation Border (Reserved)

A reserved keyword that must now be given the value zero.

m - Color of Annotation Border, Red Component

A 32-bit binary number in the range zero through 1000, both inclusive. Zero specifies no red contribution to the border color and 1000 specifies maximum intensity of that color.

n - Color of Annotation Border, Green Component

A 32-bit binary number in the range zero through 1000, both inclusive. Zero specifies no green contribution to the border color and 1000 specifies maximum intensity of that color.

o - Color of Annotation Border, Blue Component

A 32-bit binary number in the range zero through 1000, both inclusive. Zero specifies no blue contribution to the border color and 1000 specifies maximum intensity of that color.

The eight display-related view specifications provided are described below. One and only one may be specified when the LINK instruction is issued. If this keyword is omitted, the default is “F”.

1 - (Z,p,q,r)

The “left top zoom” option specifies the coordinates of the point on the page that is to occupy the upper-left corner of the window and the zoom factor to be in effect when the page is displayed. **Note:** The value zero for any of these three arguments specifies that no change from the preceding bookmark is to be made.

p - Left

Specifies the X-component of the point on the page that is to be at the top-left corner of the display window when the page is displayed under Acrobat. X-position must be stated in tenth point units.

q - Top

Specifies the Y-component of the point on the page that is to be at the top-left corner of the display window when the page is displayed under Acrobat. Vertical position must be stated in tenth point units.

r - Zoom

Specifies the zoom factor to be in force when the page is displayed in the window. The zoom value specified is a percentage number and must be in the range 8 to 3200, inclusive. These limits are not fixed; they vary with size of the page being displayed, as well as of pages previously viewed in the current file.

The value zero is allowed, as well, to specify that zoom is to be the same as the current page. When Zoom is set to zero, Acrobat ignores the left and top values supplied in this statement. The values used on the target page will be the same as those in effect for the current page.

A discussion of Acrobat General Preferences and Document Information, as they relate to document magnification, appears in the OPENACT description.

2 - (F)

Specifies that the page displayed is to be scaled to fit the window.

3 - (H,p)

Specifies that the page displayed is to be scaled such that the width fits the window. Full page depth may or may not be visible.

p - Top

Specifies the Y-position on the page that is to be at the top edge of the display window. Y-position must be stated in tenth point units.

4 - (V,p)

Specifies that the page displayed is to be scaled such that the height fits the window.

p - Left

Specifies the X-position on the page that is to be at the top edge of the display window. X-position must be stated in tenth point units.

5 - (R,p,q,r,s)

Specifies a rectangular area of the page that is to be displayed on the monitor when the page is opened. The area displayed is defined as the horizontal and vertical coordinates of its lower-left and upper-right corners.

p - Left

Specifies X-component of the point on the page that is to be at the lower-left corner of the display window when the page is displayed under Acrobat. X-position must be stated in tenth point units.

q - Bottom

Specifies Y-component of the point on the page that is to be at the lower-left corner of the display window when the page is displayed under Acrobat. Y-position must be stated in tenth point units.

r - Right

Specifies X-component of the point on the page that is to be at the upper-right corner of the display window when the page is displayed under Acrobat. X-position must be stated in tenth point units.

s - Top

Specifies Y-component of the point on the page that is to be at the upper-right corner of the display window when the page is displayed under Acrobat. Y-position must be stated in tenth point units.

6 - BF

Specifies that the page displayed is to be scaled such that its bounding box fits entirely within the window both horizontally and vertically.

7 - BH,p

Specifies that the page is to be displayed such that the width of its bounding box fits within the window. Full page depth may or may not be visible.

p - Top

Specifies the Y-position on the page that is to be at the top edge of the display window. Y-position must be stated in tenth point units.

8 - BV,p

Specifies that the page is to be displayed such that the depth of its bounding box fits within the window. Full page width may or may not be visible.

p - Left

Specifies the X-position on the page that is to be at the left edge of the display window. X-position must be stated in tenth point units.

Notes:

Not all of the keywords and their arguments are required for each link type. Some keywords have no meaning with some link types. The following notes apply:

- 1) No other keywords than TYPE= are necessary when link type is zero.
- 2) If TYPE=0, the System will return a zero value in CRC when the LINK statement is executed.

Adobe Acrobat Support LINK

- 3) If the View keyword is omitted, and was also omitted when the original link was issued, the view defaults to “fit the window”; that is, to V=(F).
- 4) The full pathname of the link object is required only for external link objects; this is the case when Type is 2.
- 5) Argument “d”, destination page number, is not used when the link object is a Universal Resource Locator, or a partial URL when a Uniform Resource Identifier has been issued; this is the case when Type is 3.
- 6) No other keywords and related arguments than TYPE= and N= are required when Type is 4.
- 7) If path, N=, is to be updated, the initial LINK statement must contain a dummy argument that is at least as long as the path name that is to be supplied in the updated LINK statement.
- 8) If argument “k” is given a negative value to specify that the link is invisible, the keyword C and its arguments serve no purpose.
- 9) If link annotations are to be made in justified text or in columns of text that are to be vertically justified, the co-ordinates of the lower-left and upper-right corners of the annotations are subject to adjustment:
 - Horizontal justification may change the X-components of the annotations.
 - Vertical justification may change the Y-components of the annotations.

The SAVPOS statement may be used to provide adjustment to annotation values:

- Updates to the X-components are made when the line in which they exist is closed.
- Updates to the Y-components are made when the page in which they exist is closed.

The link statement must be issued before a page is closed. If SAVEPOS is used to adjust the Y-component values of LINK annotations, the link must be updated following page close or at end of job.

- 10 The SAVPOS statement may be issued a maximum of 100 times per page.

Continued . . .

Using LINKs with the CLOSE !PAGE Statement

- 1) The sequence of statements is critical when links appear in a program that also uses the CLOSE !PAGE statement. The correct sequence of statements would be:

```

EPAGE                               Close last output page.
CLOSE      !PAGE
SPAGE

```

- 2) If the CLOSE !PAGE statement is issued after an SPAGE statement, some links may be lost. For example:

```

SPAGE                               First page, new output file.
COMPF
LINK                                 This link will be lost.
CLOSE      !PAGE                     Should close page before this statement.
LINK                                 This link will be OK.

```

File Specifications

A file location may be specified in terms that relate to a specific file system. The System-Dependent paths shown in Table 18-1 will work for the individual file systems noted. Acrobat uses a *simple* file specification method that is independent of operating system naming conventions. Note that PostScript (and Acrobat) uses the backslash character (\) as an escape character, each backslash character in a system-dependent DOS file specification must be preceded by a backslash. See examples below.

The format for a simple file specification divides the string into component parts separated by the slash (/) character. The slash is a generic component-separator that Acrobat maps to the appropriate separator when it generates a system-dependent filename.

TABLE 18-1. EXAMPLES OF FILE SPECIFICATIONS

System	System-Dependent Path*	Written as...
Mac	Macintosh HD:PDFDocs:spec.pdf	/Macintosh HD/PDFDocs/spec.pdf
DOS	\pdfdocs\spec.pdf (no drive)	//pdfdocs/spec.pdf
DOS	c:\pdfdocs\spec.pdf	/c/pdfdocs/spec.pdf
DOS	pcadobe/eng:\pdfdocs\spec.pdf	/pcadobe/eng/pdfdocs/spec.pdf
UNIX	/user/fred/pdfdocs/spec.pdf	/user/fred/pdfdocs/spec.pdf
UNIX	pdfdocs/spec.pdf (relative)	pdfdocs/spec.pdf

* The complete path name of the PDF file when link type is 2. When link type is 3 or 4, the destination is a URL, the path name is a protocol for the World Wide Web. The most expected protocols are "http" written, for example, http://www.spec.com/ and "ftp".

**Adobe Acrobat Support
LINK**

Examples:

1) `c:\pdfdocs\spec.pdf`

This path reaches the file `spec.pdf` in a directory named `pdfdocs` on the `c:` drive when Acrobat is running under DOS.

2) `/c/pdfdocs/spec.pdf`

This path reaches the same file as does Example 1, above. Acrobat maps the slash characters as shown under System-Dependent Path in the third line of the table when Acrobat is running under DOS.

Adobe Acrobat Support
OPENACT

e - Zoom

Specifies the zoom factor to be in force when the page is displayed in the window. The zoom value specified is a percentage number and must be in the range 8 to 3200, inclusive. These limits are not fixed; they vary with size of the page being displayed, as well as of pages previously viewed in the current file. See discussion, below, under Magnification: General Preferences & Document Information.

2 - F

Specifies that the page displayed is to be scaled to fit the window.

3 - H=c

Specifies that the page displayed is to be scaled such that the width fits the window. Full page depth may or may not be visible.

c - Top

Specifies the Y-position on the page that is to be at the top edge of the display window when the page is displayed under Acrobat. Y-position must be stated in tenth point units.

4 - V=c

Specifies that the page displayed is to be scaled such that the height fits the window. Full page width may or may not be visible.

c - Left

Specifies the X-position on the page that is to be at the left edge of the display window when the page is displayed under Acrobat. X-position must be stated in tenth point units.

5 - R=(c,d,e,f)

Specifies a rectangular area of the page is to be displayed on the monitor when the page is opened. The area displayed is defined as the horizontal and vertical coordinates of its lower-left and upper-right corners.

c - Left

Specifies the X-component of the point on the page that is to be at the lower-left corner of the display window when the page is displayed under Acrobat. X-position must be stated in tenth point units.

d - Bottom

Specifies the Y-component of the point on the page that is to be at the lower-left corner of the display window when the page is displayed under Acrobat. Y-position must be stated in tenth point units.

e - Right

Specifies the X-component of the point on the page that is to be at the upper-right corner of the display window when the page is displayed under Acrobat. X-position must be stated in tenth point units.

f - Top

Specifies the Y-component of the point on the page that is to be at the upper-right corner of the display window when the page is displayed under Acrobat. Y-position must be stated in tenth point units.

Acrobat will scale, subject to the 800% upper limit, as is necessary to fill the display area.

6 - BF

Specifies that the page displayed is to be scaled such that its bounding box fits entirely within the window both horizontally and vertically.

7 - BH=c

Specifies that the page is to be displayed such that the width of its bounding box fits within the window. Full page depth may or may not be visible.

c - Top

Specifies the Y-position on the page that is to be at the top edge of the display window. Y-position must be stated in tenth point units.

8 - BV=c

Specifies that the page is to be displayed such that the depth of its bounding box fits within the window. Full page width may or may not be visible.

c - Left

Specifies the X-position on the page that is to be at the left edge of the display window. X-position must be stated in tenth point units.

Magnification: General Preferences & Document Information

The default magnification when a document opened under Acrobat is specified in Reader and Exchange under Preferences, General. Choose *File > Preferences > General* to get the General Preferences dialog box. It provides for the specification of default page layout, default magnification and other basic options useful when viewing PDF files.

Initial magnification for a document to be opened under Acrobat may be specified using the OPENACT statement. This statement sets Document Information, Open to the zoom factor specified in the statement. With the document open, choose *File > Document Information > Open* to see this value.

If OPENACT is not issued in the application program or if OPENACT without one of the five display-related specifications is issued, the Document Information, Open magnification factor provided is default and the default specification specified under Preferences, above, will govern.

Without regard to General preferences or Document Information, zoom factor for a target page may be specified in BOOKMARK and LINK statements. If a BOOKMARK or a LINK statement specifies a magnification factor other than Default (General Preferences or Document Information, Open), the target page will be displayed at that magnification value. If any of the Next Page, Last Page, Previous Page, First Page navigation buttons is clicked, the magnification factor from the bookmarked or linked page will continue to be in effect. If another bookmark or link is clicked magnification of that target page will be determined by the magnification specified in that bookmark or link.

Adobe Acrobat Support

OPENACT

Examples:

```
OPENACT P=PNUM
```

This open action statement specifies the sequential page number that is to be displayed when the document is opened under Acrobat; it is the number stored in PNUM when the statement is executed. Since no display-related specification is cited, Acrobat will open the document using left-top-zoom. Left and top will both be zero and zoom will be the default zoom set in Reader or Exchange. Bookmarks, if the document has any, will be hidden; the default for the BO keyword.

```
BKOPN: DATA R 'Y'  
...  
OPENACT BO=BKOPN, P=PNUM, F
```

This open action statement, like the one above, specifies the sequential page number that is to be displayed when the document is opened under Acrobat. It also specifies that bookmarks are to be displayed when the file is opened under Acrobat and that the page is to be scaled to fit the display window.

Notes:

- 1) The OPENACT statement may be used with the PostScript, PDF and Dual drivers.
- 2) The OPENACT statement may be issued once or multiple times during the course of a composition job. Each execution overwrites the earlier one.
- 3) When the OPENACT statement is used with the PostScript driver, information conveyed in this statement is put into the composed file for use with Acrobat Distiller.
- 4) When the OPENACT statement is used with the Dual driver, information conveyed in this statement is put into the composed PostScript file, as well, for use with Acrobat Distiller.

VIEW DESCRIPTOR

Pages designed for printing may have “crop marks” that define the limits of the physical page. Such marks are used in trimming pages that are being prepared for printing. Pages designed for presentation screens may have white space around the “type page”, the part of the page that contains type and images. In both cases it may be desirable to suppress everything outside the type page, page margins and crop marks in the one case and margins in the other, when documents are viewed under Adobe Acrobat®. Otherwise the result is often smaller magnification and/or more scrolling to read the page on the monitor. Any marks or white space outside the view area specified will not be displayed under Acrobat.

Note: Suppressing parts of the pages using the \$VIEW descriptor affects only the pages displayed under and printed from Acrobat. When the pages are displayed under Acrobat, the image may be zoomed, magnified, to better use available monitor screen size. When pages with suppressed areas are printed from Acrobat, the entire physical page is printed, but any marks, such as crop marks that are outside the view area, are suppressed. The entire page, including crop marks, will be printed if the PostScript file is printed.

The View Descriptor is used to define the area on a page that encloses all the marks, type and images, that are to be displayed when the document is displayed under Acrobat. Position is specified as the upper-left corner of the area to be displayed. Size is specified by width and depth. All dimensions are scaled arguments specified to the nearest tenth point.

Only one View Descriptor is allowed in an application program. X- and Y-position of the upper-left corner of the view area and width and depth may be specified as constants or as variables. If the four values are specified as variables, the view area may be different from page to page. The values at the time the page is closed apply to the page being closed.

\$VIEW X=a,Y=b,W=c,D=d

a - X-Coordinate

A decimal constant or a variable that specifies the X-position of the upper-left corner of the view area with respect to the left edge of the page.

b - Y-Coordinate

A decimal constant or a variable that specifies the Y-position of the upper-left corner of the view area with respect to the top edge of the page.

c - Width

A decimal constant or a variable that specifies the width of the view area.

d - Depth

A decimal constant or a variable that specifies the depth of the view area.

Continued . . .

Adobe Acrobat Support
\$VIEW

Example:

1) **\$VIEW X=GV1,Y=GV2,W=GV3,D=GV4**

This View Descriptor specifies that the X,Y coordinates of the upper-left corner of the view area are GV1,GV2, that the width of the view area is GV3 and that the depth of the view area is GV4. The values in these four variables at the time each page is closed define the view area for that page. Any or all four of the variables may be changed to accommodate the page being closed.

Notes:

1) **\$VIEW** Descriptor present in an application program:

If all four arguments are zero and are not changed during program execution, there will be no crop boxes on any pages generated.

2) **\$VIEW** Descriptor present in an application program:

If any of the four arguments has a non-zero value, crop boxes will be issued for all pages closed under this condition using the values in effect as each page is closed. If all four values are subsequently set to zero, crop boxes will be issued using values derived from the \$PAGE descriptor for all pages closed under this condition.

DOCINFO STATEMENT

This statement is used to place descriptive information about a document being composed into a PDF file generated by the PDF driver or the Dual driver. Such information is available to the person viewing a document under Acrobat via the Document Information, General Information, option under the File pull-down menu.

The DOCINFO statement also places information lines in the PostScript file generated by the PostScript driver and the Dual driver that will be converted to PDF document information commands if the PostScript file is processed by the Acrobat Distiller. These information lines are ignored by PostScript printers and interpreters that may process the PostScript file. This statement may be used with the PostScript, PDF and Dual drivers.

Information that may be supplied using this statement is: Title, Subject, Author and Keywords. Versacomp supplies four other bits of information about the document that are displayed on Acrobat's General Information screen. These are:

- Creator: The word VERSACOMP and Release number.
- Producer: The name of the Versacomp program that produced the file.
- Created: The date and time at which application program execution began.
- Modified: Same as Created.

DOCINFO T=a[,S=b][,A=c][,K=d]

Each of the four arguments to the DOCINFO Statement may be a maximum of 240 characters in length and any character in the PDF Document Encoding scheme. See Appendix A. Trailing blanks in a data area are not stripped; they are passed as part of the caption.

A limited number of characters is visible when the Document information screen is displayed for a document open under Acrobat. If any of the four fields extends beyond its display area, the viewer may put the mouse pointer at the right end of the area and click to scroll the caption to see hidden characters.

a - Title

Specifies the information that will be displayed in the Title window of the display. Information to be provided may be in a data area, subscripts are allowed, or in a literal.

b - Subject

Specifies the information that will be displayed in the Subject window of the display. Information to be provided may be in a data area, subscripts are allowed, or in a literal.

c - Author

Specifies the information that will be displayed in the Author window of the display. Information to be provided may be in a data area, subscripts are allowed, or in a literal.

d - Keywords

Specifies the information that will be displayed in the Keywords window of the display. Information to be provided may be in a data area, subscripts are allowed, or in a literal.

**Adobe Acrobat Support
DOCINFO**

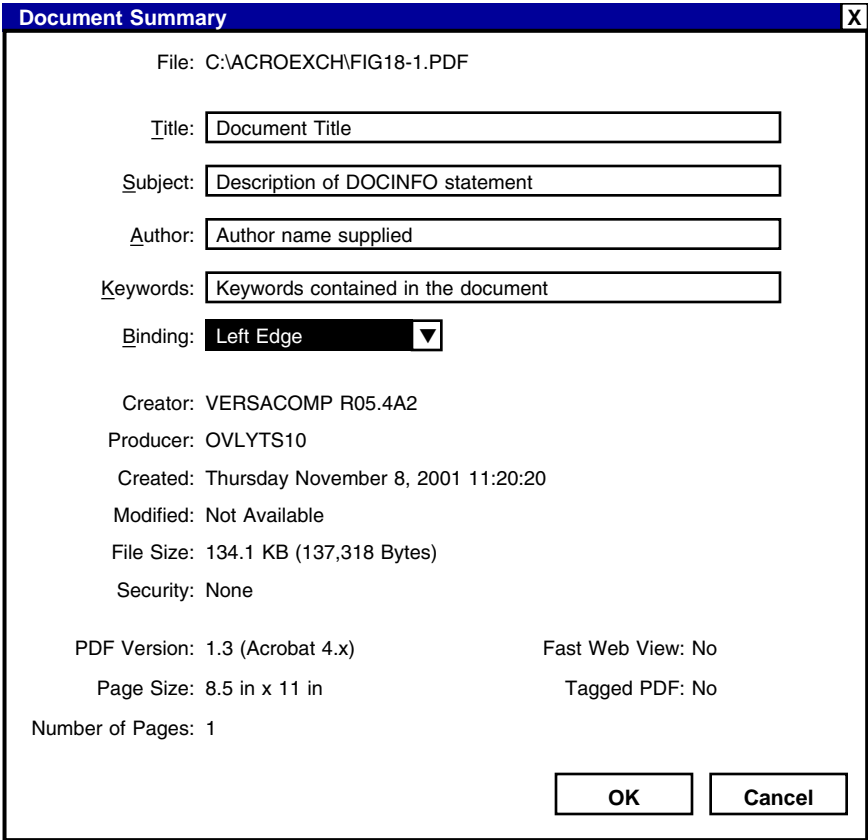
Notes:

- 1) This instruction may be issued any time before EXIT. The last DOCINFO statement issued in a composition run will be used.
- 2) All four caption areas are initialized to blanks on entry to the DOCINFO statement.
- 3) If the DOCINFO statement is used in conjunction with the CLOSE !PAGE statement, the DOCINFO areas in effect at execution of the CLOSE !PAGE statement apply to the file being closed. The DOCINFO areas are never erased; if new DOCINFO information is desired for the new file being opened, a new DOCINFO statement must be issued following the CLOSE !PAGE statement or before the next CLOSE !PAGE statement or the end of job.
- 4) The caption must contain only printable characters.
- 5) The T= operand replaces the Versacomp application name that now appears in the PostScript file following the comment %%Title:.

Example:

The sample code that follows will generate the form below when the Document Info option is invoked while the PDF document is viewed under Adobe Acrobat.

```
...  
DOCINFO      T=DOCTTL,S=SUBJECT,A=AUTHOR,K=KEYWORDS  
...  
...  
DOCTTL:     DATA      C'Document Title           '  
SUBJECT:    DATA      C'Description of DOCINFO statement '  
AUTHOR:     DATA      C'Author name supplied       '  
KEYWORDS:   DATA      C'Keywords contained in the document '  
...  
...
```



Note that, for purposes of this example, each of the captions is 35 characters in length. This allows users easily to post-process a PDF file and revise information by replacing fixed-length fields.

Figure 18-1. Document Information Display

SAVPOS STATEMENT

The horizontal and vertical positions on a page that are needed as arguments in a BOOKMARK statement may change when a line or a page is closed. Such positions may no longer be accurate when the generated PDF file is viewed using Adobe Acrobat™.

Rather than generating bookmarks “on the fly” as words and lines are composed, an application may be structured to save bookmark data in a buffer and delay the generation of bookmarks until a page is closed or until a job is finished.

This statement provides the means for Versacomp to adjust saved horizontal and vertical positions to account for horizontal movement made by COMPX and vertical movement caused by VERJUS at the close of lines and blocks.

```
SAVPOS [CX=(a,b)][,CY=(c,d)]  
      [BX=(a,b)][,PY=(c,d)]  
      [PX=(a,b)]
```

a - X-Position

Specifies the name of the storage area in which an x-position to be adjusted is to be stored. May be the name of a data storage area, a field, a variable or a table element. A pointer variable may be used. Subscripting is permitted. The cited data element must be exactly four bytes in length.

b - X-Position Adjustment

Specifies a (binary) value that is to be added to the x-position cited in argument “a” above before the value is stored. A positive value causes a rightward adjustment; a negative value causes a leftward adjustment of the value stored. May be a constant or a variable.

c - Y-Position

Specifies the name of the storage area in which an y-position to be adjusted is to be stored. May be the name of a data storage area, a field, a variable or a table element. A pointer variable may be used. Subscripting is permitted. The cited data element must be exactly four bytes in length.

d - Y-Position Adjustment

Specifies a (binary) value that is to be added to the y-position cited in argument “c” above before the value is stored. A positive value causes a downward adjustment; a negative value causes an upward adjustment of the value stored. May be a constant or a variable.

Notes:

- 1) The values of CX, CY, BX, PX and PY cited in the SAVPOS statement are the values of the related system variables at the time the statement is executed.

Adjustments to horizontal positions caused by line justification and adjustments to vertical positions caused by vertical justification are made as follows:

- Updates to the X-components are made when the line in which they exist is closed.

- Updates to the Y-components are made when the page in which they exist is closed.

The link statement must be issued before a page is closed. If SAVEPOS is used to adjust the Y-component values of LINK annotations, the link must be updated following page close or at end of job.

2) SAVPOS may be issued a maximum of 100 times per page.

Example 1:

The sample code that follows saves the information needed to generate a Table of Contents, and related bookmarks, for a volume. Since the pages of the volume are vertically justified, it is desirable that the PY values be adjusted at each page close.

The sample code is executed following the LINE or LEAD statement that establishes the baseline on which a first level side head will be composed and before composition is effected.

When the page is closed (EPAGE or SPAGE) Versacomp will store the adjusted value of PY into TOCSAVE at the displacement stored in GV50 at the time the SAVPOS statement was executed. Actually, the vertical position twelve and one-half points above the side head baseline will be adjusted and stored.

```

. . .
MOVE      TEXT (CC , RC ) , TOCELEM
BTOD      PN , TOCPAGE
MAKE      TOCLEVEL=2
MAKE      TOCPX=PX
MOVE      TOCTEXT , TOCSAVE (SV5 , 100)
MAKE      GV50=SV5+86
SAVPOS    PY= (TOCSAVE (GV50 , 4) , -12.5)
INC       SV5 , 100
. . .
TOCTEXT:  DATA      100C' '          TOC buffer.
TOCLEVEL: DS         LOC= (0 , 4)     TOC level.
TOCELEM:  DS         LOC= (5 , 75)    TOC element.
TOCPX:    DS         LOC= (81 , 4)    Related block PX.
TOCPY:    DS         LOC= (86 , 4)    Related page Y.
TOCPAGE:  DS         LOC= (96 , 4)    Related page number.
TOCSAVE:  DATA      40000C' '       TOC save buffer.

```

The application program may have procedures that compose a table of contents and generate a series of bookmarks based on the data stored in the data element TOCSAVE.

Generating and Using Versacomp Overlays

Overlays are generated by Versacomp programs. Multiple overlays may be generated by a single execution of the application program. The size of the program is, of course, a function of the complexity of the overlays being generated. Two statements, OVLYBEG and OVLYEND are used in the generation of an overlay. OVLYCALL is used to incorporate a Versacomp overlay into a page being composed.

A Versacomp overlay may not be displayed directly under Acrobat. For this reason, the OVLYEND statement has a test mode of its own. A "T" keyword provides for testing overlays being developed. When this mode is invoked, Versacomp generates a file that Acrobat will display. See Example 3.

OVLYBEG STATEMENT

This instruction starts the "overlays build" mode. It replaces the SPAGE instruction when a form is being built. Multiple overlays may be generated with a Versacomp program execution.

The Overlay facility may be used only with the PDF and Dual drivers.

OVLYBEG a

a - \$PAGE Descriptor Name

The name of the page descriptor that specifies the physical dimensions of the overlays to be generated.

An example of overlay generation follows the OVLYEND descriptor.

Notes:

- 1) The minimum allowed page size is 0.25 x 0.25 inch (18 x 18 points) and the maximum allowed page size is 200 x 200 inches (14400 x 14400 points). (*Portable Document Format Reference Manual*, Adobe Systems Reference Manual, page 203)
- 2) A maximum number of 1000 fonts may be used in all of the overlays in a single composition job.
- 3) A maximum number of 1000 Versacomp Overlays may be used in a single composition job.

OVLYEND STATEMENT

The OVLYEND statement terminates the overlay build mode. It replaces the EPAGE instruction when an overlay is being generated. Multiple overlays may be generated by a single execution of the application program.

The Overlay facility may be used only with the PDF and Dual drivers.

OVLYEND F=a,ID=b,N=c[,T]

a - \$FILE Descriptor Name

The name of the \$FILE descriptor that defines the characteristics of the overlay file to be created.

b - Data Area Name for Filename

Specifies the name of the data area that contains the filename under which the overlay being created is to be stored.

c - Data Area Name for Overlay Name

The name of the overlay. This name will be placed in the overlay file that is generated. If a given composition job will use more than one overlay, the overlay names assigned must each be unique. This name is required by PDF and the Versacomp OVLYCALL instruction. The maximum size allowed is 20 bytes.
Note: This name should be kept to the minimum length possible in the interest of speed in processing.

- Test Mode Indicator

This optional argument specifies the test mode. When this indicator is present, Versacomp will generate a PDF file that may be displayed under Acrobat. After an overlay design is complete, the application must be run without this argument in order to generate an overlay file that may be called by the OVLYCALL instruction.

Notes:

- 1) The name cited in the data area specified by argument "b" is the filename under which the overlay file will be written to disk.
- 2) The name cited in the data area specified by argument "c" is the name by which Acrobat will call the overlay.
- 3) In the sample programs of Examples 3, 5 and 7, the overlay file generated is on SU1. It must be specified to be a streaming file in PC systems and an undefined file in Mainframe systems.

Adobe Acrobat Support
OVLYEND

Example 3:

The sample overlay generator that follows uses the test option of the OVLYEND instruction. Under this option, Versacomp will generate a PDF file that may be immediately viewed and/or printed under Acrobat. This option simplifies debugging an overlay since a second program to call the overlay is not necessary during the design stage. **Note:** the test file to be displayed under Acrobat is MYOVLY.FRM, the name given the Named File on SU1.

To generate a usable overlay, the program must be run again without the "T" keyword. See the commented line in the sample program below.

```

NAME          OVLYBULD
$PAGE         OVLYPAGE,W=402,D=648
$BLOCK       OVLYBLK, LB=0, TB=0, W=402, D=648
$FONT        FNT2, TF=53, PS=14, BL=14, SW=14
$FILE        OVLYF, DEV=(SU1, NF), REC=(4096, 4096, U), @
EOF=FEOF, NF=OVLYD

OVLYD:      DATA      C'MYOVLYA.FRM'           Filename for overlay file
OVLYN:      DATA      C'MyOvlyA'             PDF Name for overlay

          OVLYBEG      OVLYPAGE               Replaces SPAGE
          SBLOCK       OVLYBLK               Require at least one block
* -----
          VRULE        L=574, W=1, X=(66.5, 133.5, @
200.5, 267.5, 334.5), Y=647
          HRULE        L=400, W=1, X=1, Y=(144, 216, @
288, 360, 432, 504, 576)
          LEAD         60.0
          COMPF        C'Sample Versacomp Overlay', M=CE, T=(0, 402), FNT2
          LEAD         8.0
          BOX1         L=402, H=24, X=0, T=1.5, R=6, SD=15
          LEAD         580.0
          BOX1         L=402, H=576, X=0, T=1.5, R=12, SD=15
          LEAD         0
* -----
          End of Overlay logic

          OVLYEND      F=OVLYF, ID=OVLYD, N=OVLYN, T
***          OVLYEND      F=OVLYF, ID=OVLYD, N=OVLYN
          EXIT

FEOF:      OPRMSG      C'*****End of File Error*****'
          END

```

OVLYINFO STATEMENT

The OVLYINFO statement is used to retrieve and store information about a Versacomp overlay that will later be merged using the OVLYCALL statement.

OVLYINFO F=a,ID=b,BUF=c

a - File Name

Specifies the name of the File Descriptor (\$FILE) for the overlay file. May be a name or a pointer variable.

b - Illustration ID

May Specify the name of a storage area or field that contains the file name of the overlay. The file name of the overlay may be up to 256 characters (EBCDIC) in length. Trailing blanks in the Overlay ID are stripped by the system. Depending on the operating system under which Versacomp is running, the form files may reside in:

- PC - For PC Systems, a number of Versacomp overlays may be maintained together in a directory. They may be accessed for determination of overlay name, width and depth and for merge using the Versacomp Named File facility. See Note 1.
- MVS - For MVS Systems, a number of Versacomp overlays may be maintained together in a Partitioned Data Set. They may be accessed for determination of form name, width and depth and for merge using the Versacomp Named File facility. See Note 2.
- VSE - For IBM/DOS-VSE Systems, a number of Versacomp overlays may be maintained together in a keyed VSAM File. They may be accessed for determination of overlay name, width and depth and for merge using the Versacomp Named File facility. See Note 3.

If the overlay file is not found, a return code of 20 will be loaded into the system variable CRC. A return code of 24 reports an invalid overlay format.

c - Buffer Name

The name of a data area into which returned information, about the overlay file interrogated, will be stored. This data area should be defined to be 100 bytes in length to allow for future expansion. The area may be subscripted. Data returned are stored as follows:

First four bytes: reserved.

Second four bytes: reports the returned horizontal size in decipoints.

Third four bytes: reports the returned vertical size in decipoints.

Next sixteen bytes: reserved.

Next twenty bytes: reports the overlay name in ASCII.

Next twenty bytes: reports the overlay name in EBCDIC.

Adobe Acrobat Support OVLYINFO

Notes:

- 1) For PC-Based Systems, each overlay must reside in a single, self-contained file. Consider an overlay file named OVLAY01.FRM which resides in the directory C:\VRS\FRMS. Files in this directory will be accessed for determination of size, and later merge as well, using the Named File facility. The Named File is cited in the BAT file as follows:

```
DEV SUnn TYP=DI REC=4096 FOR=S DSN=C:\VRS\FRMS\
```

and is defined in the Versacomp application program as follows:

```
        $FILE          OVLYSIN,DEV=(SU5,NF),REC=(4096,4096,U),@  
EOF=ERREOF,NF=OVLYID
```

OVLYID is a data area with length sufficient to hold the longest form filename to be interrogated for its size. The name of a form file to be interrogated is moved into OVLYID before the OVLYINFO statement is executed. Assume that OVLYNAM is the name of a field in a control file that relates form name keys to form filenames and file types in some way. Assume further that OVLYNAM is OVLAY01.FRM, the application program could define a data buffer and then:

```
OVLYINFO:  DATA          100C' '          Overlay buffer.  
OVLYRES1:  DS            LOC=(0,4)        Reserved.  
OVLYWID:   DS            LOC=(4,4)        Overlay width.  
OVLYDEP:   DS            LOC=(8,4)        Overlay depth.  
OVLYRES2:  DS            LOC=(12,16)      Reserved.  
OVLYASC:   DS            LOC=(28,20)      Overlay name, ASCII.  
OVLYEBC:   DS            LOC=(48,20)      Overlay name, EBCDIC.  
...  
...  
MOVE      OVLYNAM,OVLYID  
OVLYINFO  F=OVLYN, ID=OVLYID, BUF=OVLYINFO
```

Following execution of the OVLYINFO statement, the data segment OVLYWID will contain overlay width in decipoints, OVLYDEP will contain overlay depth in decipoints and OVLYASC will contain the name of the overlay in ASCII. OVLYEBC will contain the name of the overlay in EBCDIC.

OVLYEBC will be used in the OVLYCALL statement for the name of the overlay.

Alternatively, the entire path may be specified in OVLYID. In this latter event, the BAT file would show the DEV record as follows:

```
DEV SUnn TYP=DI REC=4096 FOR=S DSN=
```

and the entire path must be specified in OVLYID:

```
C:\VRS\FRMS\OVLAY01.FRM
```

- 2) For IBM/MVS Systems, each overlay must be a member of a Partitioned Data Set. Maximum record size is 4096 bytes. The name of the Partitioned Data Set that contains the Versacomp overlay files may be cited as follows:

```
//SUnn DD DSN=OVLYS,DISP=SHR,  
          DCB=(RECFM=U,LRECL=4096,BLKSIZE=4096)
```

The name of each overlay file (PDS Member name) then is supplied in the data area OVLYID.

- 3) For IBM/DOS-VSE Systems, each overlay must reside in a keyed VSAM file. Maximum record size is 4096. The name of each key is supplied in the data area OVLYID.

OVLYCALL STATEMENT

The OVLYCALL statement provides the means to call a previously stored Versacomp overlay for use in a page being composed.

The Overlay facility may be used only with the PDF and Dual drivers.

OVLYCALL F=a[,ID=b],N=c,O=d

a - \$FILE Descriptor Name

The name of the \$FILE descriptor that defines the characteristics of the overlay file to be retrieved.

b - Data Area Name for Filename

The name of the data area that contains the filename of the overlay file to be retrieved.

c - Data Area Name for Overlay Name

The name of the overlay to be retrieved. This name is required by PDF as well as by this instruction. The maximum size allowed is 20 bytes. See note under argument "c" of the OVLYEND instruction.

d - Operation

This argument may take one of the values 0, 1 or 2 to specify the following operations:

- 0 - Read the disk file and store the overlay in memory.
- 1 - Read the disk file and store the overlay in memory setting the overlay at the current x/y position in the current block.
- 2 - Fetch the overlay from memory and set it at the current x/y position in the current block.

Notes:

- 1) The maximum number of overlays that may be called in a composition job is 1000.
- 2) The maximum number of overlays in a job may be limited by the target printer's memory as well.
- 3) A maximum number of 1000 fonts may be used in all of the overlays in a single composition job.

Example 4:

The sample program below calls the overlay generated by Example 3 rerun without the “T” keyword. Use of Option 1 in the OVLYCALL statement specifies that the overlay named “MyOvlyA” which resides in the named file (or Data Set) called MYOVLYA.FRM is to be stored in memory and also placed in the current block on the current baseline and starting at the current left tab. The page generated by this program is shown in Figure 18.2.

After an overlay is stored in memory, using either option 0 or option 1, repeated calls on the same overlay may be made using option 2 which will retrieve the overlay from memory using the overlay name cited in argument “c”.

```

NAME          OVLYTEST
$PAGE         PAGEP,W=612,D=792
$BLOCK       BLK, LB=0, TB=0, W=612, D=792
$FONT        FNT2, TF=74, PS=10, BL=10, SW=10
$HJUST       WS=(20,33,50), LS=(0,0)
$FILE        OVLYFILE, DEV=(SU1,NF), REC=(4096,4096,U),@
EOF=FEOF,NF=OVLYDSN

OVLYDSN:     DATA      30C' '           Buffer: overlay filename
OVLYNAM:     DATA      20C' '           Buffer: overlay name
OVLY1DSN:    DATA      C'MYOVLYA.FRM'   Filename for overlay file
OVLY1NAM:    DATA      C'MyOvlyA'      PDF Name for overlay

$PAGE       PAGEP
$BLOCK      BLK
$TAB        105,507
$LEAD       702.0
MOVE        OVLY1DSN,OVLYDSN
MOVE        OVLY1NAM,OVLYNAM
OVLYCALL    F=OVLYFILE, ID=OVLYDSN,N=OVLYNAM,O=1
$BLOCK      BLK
$LEAD       726.0
COMPF      C'Figure 18-2. Use of MyOvlyA in a Page',@
M=CE,T=(0,612),F=FNT2
LINE
EXIT

FEOF:       OPRMSG     C'*****End of File Error*****'
EXIT
END

```

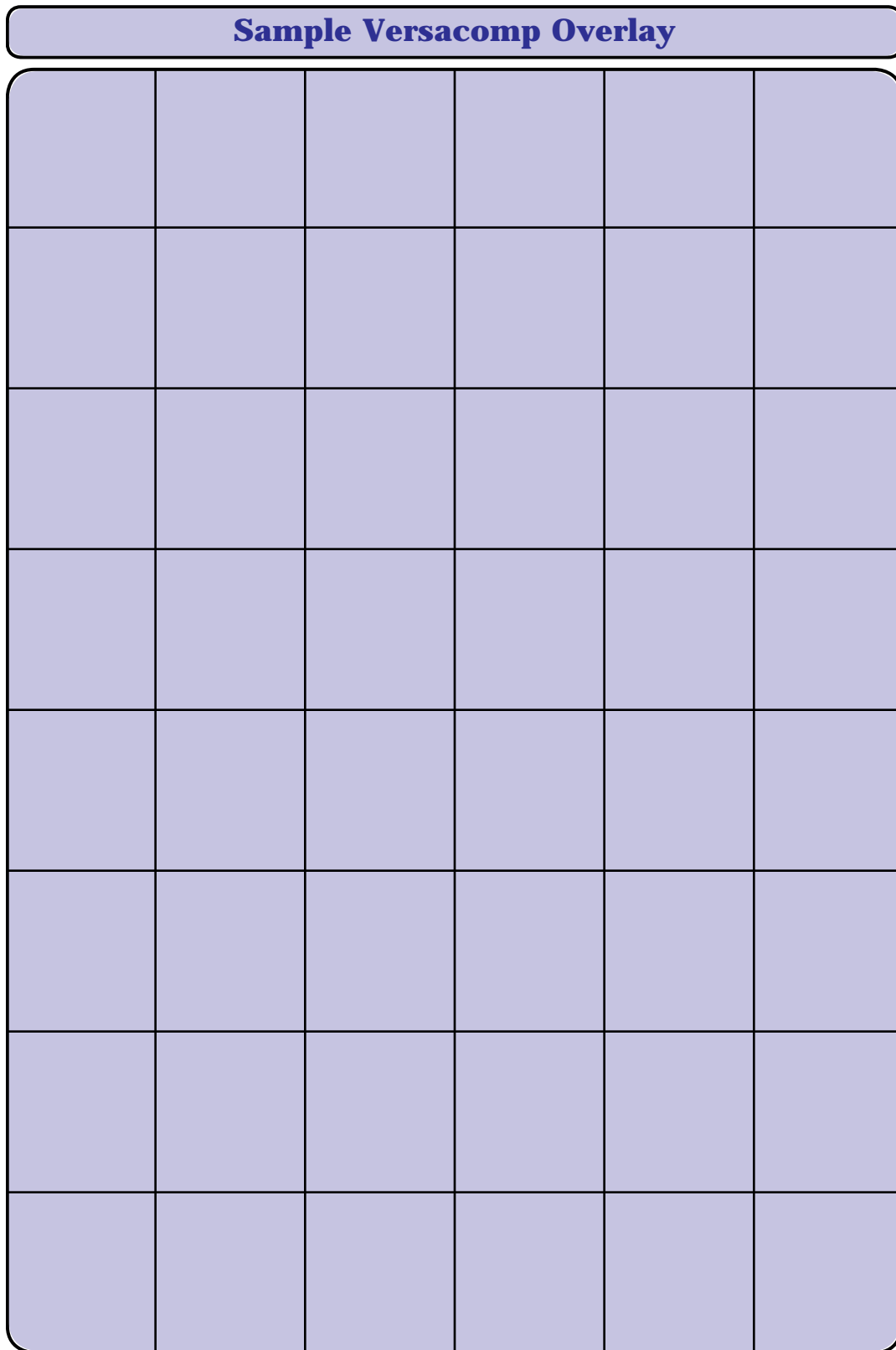


Figure 18-2. Use of MyFormA in a Page

Example 5:

The sample overlay generator that follows produces a form that is 192 points wide by 36 points high.

The overlay is simply a round-cornered box that is 152 points wide, is 36 points high and is centered on 192 point measure. The form has a 10 percent screen and a 2 point outline. It is designed for use in a publication whose columns are 192 points wide. It might be used in a directory behind classified headings.

```

NAME          OVLYBULD
$PAGE         OVLYPAGE,W=192,D=36
$BLOCK        OVLYBLK, LB=0, TB=0, W=192, D=36
$FILE         OVLYF, DEV=(SU1,NF), REC=(4096,4096,U), @
EOF=FEOF, NF=OVLYD

OVLYD:        DATA      C'MYOVLBYB.FRM'           Filename for overlay file
OVLYN:        DATA      C'MyOvlyB'              PDF Name for overlay

OVLYBEG       OVLYPAGE           Replaces SPAGE
SBLOCK        OVLYBLK           Require at least one block

* -----                               Start of Overlay logic
LEAD          36.0
BOX1          L=152,H=36,X=20,T=2.0,R=12,SD=10
* -----                               End of Overlay logic

OVLYEND       F=OVLYF, ID=OVLYD,N=OVLYN
EXIT

FEOF:         OPRMSG          C'*****End of File Error*****'
EXIT

END

```

Adobe Acrobat Support
OVLYCALL

Example 6:

The sample program below calls the overlay generated by the program shown in Example 5. Use of Option 0 in the first OVLYCALL statement specifies that the overlay named "MyOvlyB" which resides in the named file (or Data Set) called MYOVLYB.FRM is to be stored in memory. A page and block must be open when the OVLYCALL instruction is executed.

The code that generated the two column marker boxes is not shown, nor is any material composed in the columns. This example simply shows that an overlay stored in memory may be accessed from memory as many times as necessary. More important, the overlay is called only once and thereafter resides in printer memory.

```

NAME          OVLYTEST
$PAGE         PAGEP , W=612 , D=792
$BLOCK        BLK , LB=0 , TB=0 , W=612 , D=792
$BLOCK        BLKL , LB=105 , TB=92 , W=192 , D=618
$BLOCK        BLKR , LB=309 , TB=92 , W=192 , D=618
$FONT         FNT1 , TF=73 , PS=10 , BL=10 , SW=10
$FONT         FNT2 , TF=74 , PS=10 , BL=10 , SW=10
$FONT         FNT3 , TF=33 , PS=14 , BL=14 , SW=16
$HJUST        WS= ( 20 , 33 , 50 ) , LS= ( 0 , 0 )
$FILE         OVLYFILE , DEV= ( SU1 , NF ) , REC= ( 4096 , 4096 , U ) , @
EOF=FEOF , NF=OVLYDSN

OVLYDSN:      DATA      30C' '           Buffer: overlay filename
OVLYNAM:      DATA      20C' '           Buffer: overlay name
OVLY1DSN:     DATA      C'MYOVLYB.FRM'   Filename for overlay file
OVLY1NAM:     DATA      C'MyOvlyB'       PDF Name for overlay

SPAGE         PAGEP
SBLOCK        BLKL
MOVE          OVLY1DSN , OVLYDSN
MOVE          OVLY1NAM , OVLYNAM
OVLYCALL      F=OVLYFILE , ID=OVLYDSN , N=OVLYNAM , O=0
LEAD          23.0
COMPF         C'Classification' , M=CE , T= ( 0 , 192 ) , F=FNT3
LEAD          13.0
OVLYCALL      F=OVLYFILE , ID=OVLYDSN , N=OVLYNAM , O=2

```

**Compose material and swap columns as necessary
until another classification heading is found.**

**Adobe Acrobat Support
OVLYCALL**

```
SBLOCK      BLKR
LEAD        276.0
COMPF      C'Classification',M=CE,T=(0,192),F=FNT3
LEAD        13.0
OVLYCALL   F=OVLYFILE,ID=OVLYDSN,N=OVLYNAM,O=2
```

**Compose material and swap columns as necessary
until another classification heading is found or
until a page break is encountered.**

...
...

```
SBLOCK      BLK
LEAD        726.0
COMPF      C'Figure 18-3. Multiple Use of MyOvlyB @
in a Page',M=CE,T=(0,612),F=FNT2
SBLOCK      BLK
LEAD        68.0
COMPF      C'Adobe Acrobat Support',M=CE,T=(102,402),F=FNT1
LEAD        12.0
COMPF      C'OVLYCALL'
LEAD        684.0
COMPF      C'18-20'
```

Close page and resume composition.

...
...

EXIT

```
FEOF:      OPRMSG      C'*****End of File Error*****'
END
```

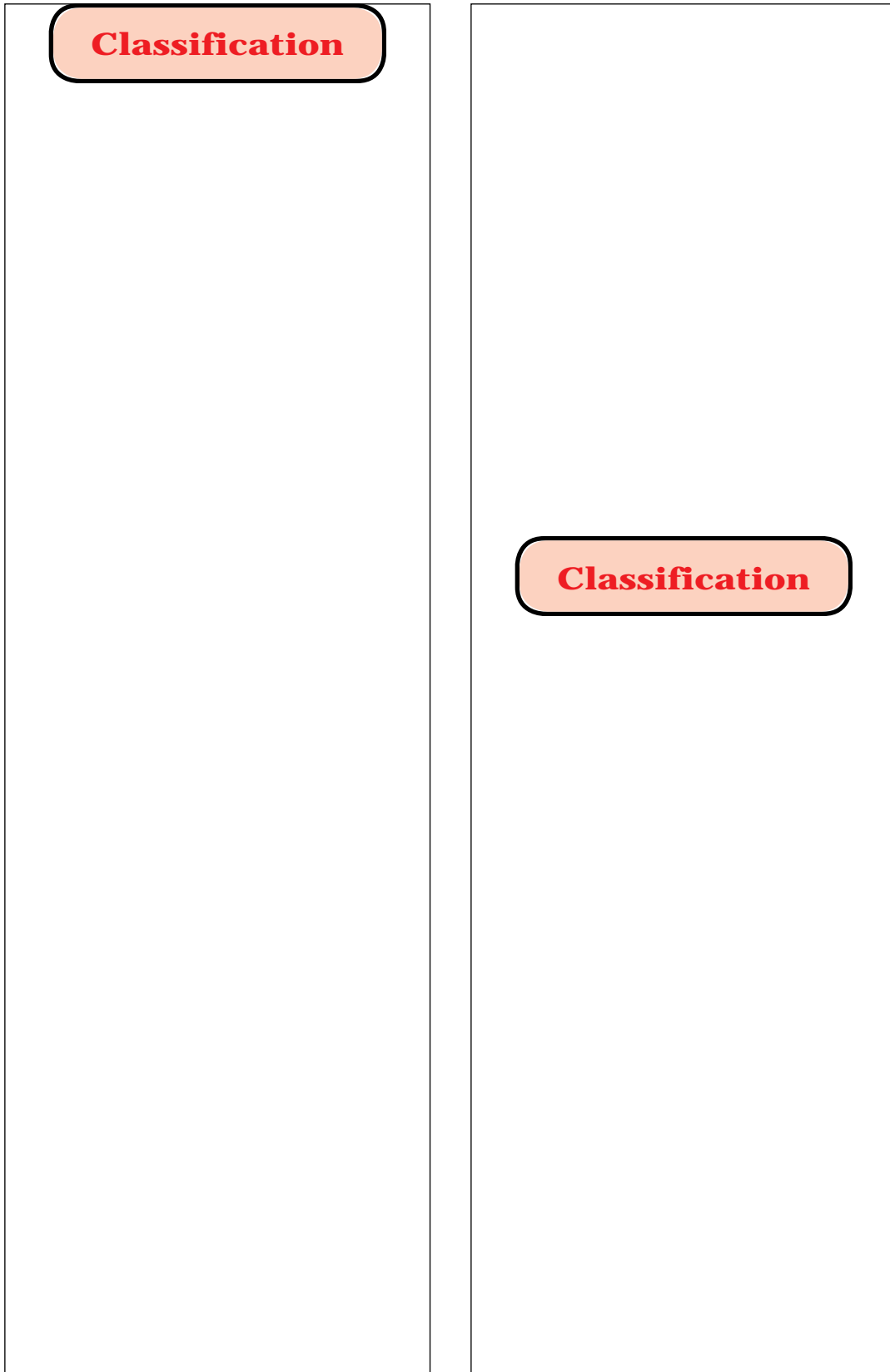


Figure 18-3. Multiple Use of MyFormB in a Page

Example 7:

The sample overlay generator that follows produces a landscape form. The body of the overlay is composed in a block that is 632 points wide and 384 points deep. The page has the same width and depth. This example demonstrates generation of the overlay. Example 8 will show its incorporation into a rotated block in a portrait page.

```

NAME          OVLYBULD
$PAGE         OVLYPAGE,W=632,D=384
$BLOCK       OVLYBLK, LB=0, TB=0, W=632, D=384
$FONT        FNT1, TF=73, PS=8.5, BL=10, SW=8.5
$FONT        FNT2, TF=53, PS=14, BL=14, SW=14
$FONT        FNT3, TF=71, PS=5, BL=5, SW=5
$FILE        OVLYF, DEV=(SU1,NF), REC=(4096,4096,U), @
EOF=FEOF, NF=OVLYD

OVLYD:      DATA      C'MYOVLy8.FRM'           Filename for overlay file
OVLYN:      DATA      C'MyOvly8'           PDF Name for overlay

          OVLYBEG      OVLYPAGE           Replaces SPAGE
          SBLOCK       OVLYBLK           Require at least one block
* -----
          Start of Overlay logic
GO:        LEAD        12.0
          HRULE        L=632,W=0.5,X=0
          LEAD        12.0
          PCOLOR       250,0,250,0           Shade of green
          BOX1         L=632,H=12,X=0,T=0.5,SD=10
          PCOLOR       Turn color off
          HRULE        L=632,W=0.5,X=0
          INC          GV1
          IF           GV1,LT,16,GO

          SBLOCK       OVLYBLK
          VRULE        L=384,W=0.5,X=(127.0,128.0,170.0,@
212.0,254.0,296.0,338.0,380.0,422.0,464.0,@
506.0,548.0,590.0),Y=384.0

```

**Adobe Acrobat Support
OVLYCALL**

```
LEAD          10.0
COMPF        C'Items of Interest',M=CE,T=(0,128),F=FNT1
MAKE         GV1=128.0
MAKE         GV2=170.0

GO1:         INC          TP
COMPF        MOTBL(TP),M=CE,T=(GV1,GV2),F=FNT1
IF           TP,GE,12,GO2
INC          GV1,42.0
INC          GV2,42.0
JUMP        GO1

GO2:         LEAD        374.0
BOX1         L=632,H=384,X=0,T=1.5
LEAD        6.0
COMPF        C'Overlay No. 123A-01',M=FL,T=(0,632),F=FNT3

* ----- End of overlay logic

OVLYEND      F=OVLYF,ID=OVLYD,N=OVLYN
EXIT

FEOF:        OPRMSG      C'*****End of File Error*****'
EXIT

MOTBL:       TABLE      C'Jan',C'Feb',C'Mar',C'Apr',C'May',@
C'June',C'July',C'Aug',C'Sept',C'Oct',C'Nov',C'Dec'

END
```

Example 8:

This program incorporates the overlay generated by the program of Example 7 into a portrait page.

```

NAME          OVLYTEST
$PAGE         PAGEL,W=612,D=792
$BLOCK        BLKL1,X=510,Y=88,W=632,D=408,ROT=270
$BLOCK        BLKL2,X=510,Y=88,W=632,D=414,ROT=270
$BLOCK        BLKR, LB=102, TB=60, W=408, D=720
$FONT         FNT1, TF=73, PS=10, BL=10, SW=10
$FONT         FNT2, TF=74, PS=10, BL=10, SW=10
$FONT         FNT14, TF=71, PS=4, BL=4, SW=4
$FILE         OVLYF, DEV=(SU1,NF), REC=(4096,4096,U), @
EOF=FEOF, NF=OVLYD

OVLYD:        DATA      C'MYOVLY8.FRM'           Filename for overlay file
OVLYN:        DATA      C'MyOvly8'           PDF Name for overlay
$PAGE         PAGEL
$BLOCK        BLKL1
TAB           0,632
LEAD          388.0
OVLYCALL      F=OVLYF, ID=OVLYD, N=OVLYN, O=1
$BLOCK        BLKL2
LEAD          406.0
COMPF         C'Figure 18-4. Use of Landscape MyOvly8',@
M=CE, T=(0,632), F=FNT2

$BLOCK        BLKR
COMPF         C'Adobe Acrobat Support', M=CE, T=(0,406), F=FNT2
LEAD          12.0
COMPF         C'OVLYCALL'
LEAD          684.0
COMPF         C'18-34'
LEAD          0
COMPF         C'OVLYTS11', M=FR, F=FNT14
EXIT
FEOF:        OPRMSG      C'*****End of File Error*****'
EXIT
END

```


COMMENT STATEMENT

The COMMENT instruction is used to place comment lines in PostScript and PDF files generated by the PostScript, PDF and the Dual drivers, such that the files generated may be post processed.

COMMENT P=a,C=b

a - Placement

Specifies locations in the generated files for placement of comments. May be a literal constant or a variable; the options are:

- 0 - Reset all comment placements to null.
- 1 - Place the comment near the beginning of the generated file.
- 2 - Place the comment near the end of the generated file.
- 3 - Place the comment near the beginning of each page.
- 4 - Place the comment near the end of each page.

Placements stay in effect for the entire job or until reset.

b - Comment

Specifies the name of a data area that will contain the comment or comments to be placed in generated files. The data area may be subscripted.

A comment line must begin with a percent sign (%). A number of comment lines, each started by a percent sign and terminated by a carriage return or line feed character may be concatenated in the data area. The maximum length of a comment line or collection of lines is 1024 bytes.

The buffer must be translated to ASCII before the COMMENT statement is issued.

Notes:

- 1) The COMMENT statement may be used only with the PostScript, PDF and Dual drivers.
- 2) Information lines are ignored by PostScript printers and interpreters that may process the PostScript and/or PDF files.
- 3) Placements stay in effect for the entire job or until reset.
- 4) All four placements may be in effect at the same time.
- 5) If file splitting is performed in the job, comments and placements will apply to all files generated or until comments are reset.
- 5) Any new placement will overlay an earlier one.
- 6) The user application program must translate comments to ASCII.
- 7) If a comment line contains a carriage return or a line feed character, the following character must be a percent sign.
- 8) If a PDF file that contains comments generated by this instruction is opened under Acrobat Exchange and is then saved, those comments will be lost.
- 9) If a PostScript file that contains comments generated by this instruction is processed using Adobe Distiller to produce a corresponding PDF file, those comments will not appear in the PDF file.

PDFMERGE STATEMENT

The PDFMERGE statement is used to merge a PDF file that contains one or more PDF pages into a PDF file being composed using the PDF driver. The page(s) being merged are merged between two pages.

PDFMERGE F=a,ID=b

a - File Name

Specifies the name of the file descriptor (\$FILE) for the PDF file that is to be merged.

b - PDF ID

Specifies the name of the data area that contains the name of the PDF file that is to be merged.

Notes:

- 1 - The System Variable PN is updated with each PDFMERGE statement issued to include all pages merged.
- 2 - The file descriptor named in "a" must cite a named file.

Caveats:

- 1) The PDF file to be merged may not exceed 1 MB in size.
- 2) The PDF file to be merged must have been generated by Versacomp.
- 3) If user has a PDF file generated by another system, please submit a sample file to Pagetec for study and possible implementation.