

VERSIÓN 18.0.0  
ABRIL DE 2023  
702P09007

# Lenguaje Xerox® VIPP®

Manual de referencia

© 2023 Xerox Corporation. Reservados todos los derechos. Xerox®, FreeFlow®, FreeFlow Makeready®, VIPP® y GlossMark® son marcas registradas de Xerox Corporation en los Estados Unidos y/o en otros países. También se reconocen las siguientes marcas comerciales de otras empresas:

Adobe®, el logotipo de Adobe, Acrobat®, el logotipo de Acrobat, Acrobat Reader®, Distiller®, Adobe PDF JobReady™, InDesign®, PostScript® y el logotipo de PostScript son marcas comerciales registradas de Adobe Systems Incorporated en los Estados Unidos y/o otros países. Todas las instancias del nombre PostScript que aparecen en el texto hacen referencia al lenguaje PostScript según lo define Adobe Systems Incorporated, a menos que se indique lo contrario. El nombre PostScript se usa como marca comercial de producto para la implementación en Adobe Systems del intérprete de lenguaje PostScript y otros productos de Adobe. Copyright 1987–2021 de Adobe Systems Incorporated y sus licenciarios. Reservados todos los derechos. Incluye las bibliotecas PDF de Adobe® y la tecnología Adobe Normalizer.

Intel®, Pentium®, Centrino® y Xeon® son marcas comerciales registradas de Intel Corporation. Intel Core™ Duo es una marca comercial de Intel Corporation.

Intelligent Mail® es una marca comercial registrada de United States Postal Service.

Macintosh®, Mac®, OS X® y macOS® son marcas comerciales registradas de Apple, Inc., registradas en los Estados Unidos y otros países. Los elementos de la Documentación técnica para el usuario de Apple se utilizan con permiso de Apple, Inc.

TIFF® es una marca comercial registrada de Aldus Corporation.

Windows®, Windows® 10, Windows® 11, Windows Server® 2016, Windows Server® 2019, Windows Server® 2022 e Internet Explorer son marcas comerciales de Microsoft Corporation; Microsoft® y MS-DOS® son marcas comerciales registradas de Microsoft Corporation.

Todos los otros nombres de productos y servicios mencionados en esta publicación son marcas comerciales o marcas comerciales registradas de sus respectivas empresas. Se usan en esta publicación en beneficio de esas empresas y no cumplen la función de demostrar respaldo u otro tipo de afiliación con la publicación.

Las empresas, nombres y datos usados en los ejemplos son ficticios, a menos que se indique lo contrario.

Si bien este material se ha preparado con gran cuidado, Xerox Corporation no acepta ningún tipo de responsabilidad como consecuencia de inexactitudes u omisiones.

Este documento se modifica periódicamente. Las modificaciones, inexactitudes técnicas y errores tipográficos se corregirán en ediciones subsiguientes.

Producido en los Estados Unidos de América.

BR38504

# Tabla de contenido

Descripción general del lenguaje VIPP® .....	19
Foro de clientes de VI Suite .....	20
Descarga de fuentes .....	21
Uso de recursos PDF con VIPP® .....	22
FreeFlow® VI Compose Open Edition.....	23
Lista de comandos VIPP® .....	24
Comandos de VIPP® .....	43
++ y – .....	52
ABSPOS .....	53
ACCLOG.....	54
ADD.....	56
ADVPAGE .....	58
AZTEC .....	59
BACKSP_off.....	61
BACKSPF_on.....	62
BACKSPP_on.....	63
BCALL.....	64
BEGINARBM.....	65
BEGINARBT .....	66
BEGINFRAME.....	68
BEGINIMP.....	69
BEGINPAGE .....	71
BEGINPCC.....	73
BEGINRPE .....	74
BEGINTABLE.....	75
BEGINXPD.....	76
BOOKLETRANGE.....	77
BOOKMARK .....	78
BSTRIP_off.....	80
BTA.....	81
BTS .....	83
CACHE.....	85
CASE .....	87
CHKPOINT .....	89
COLLATE_dbm .....	90
COLLATE_off .....	91
COLLATE_on.....	92
COPYRANGE.....	93
CUTMARK .....	94
DATAMATRIX.....	95

DEFINELAYOUT .....	97
DIV .....	98
DJDEBEGIN .....	99
DRAWB y DRAWBR .....	100
DRAWBAR .....	102
DRAWBC .....	104
DRAWBC (extensión para admitir GS1 Databar) .....	106
DRAWBM y DRAWBRM .....	110
DRAWC .....	111
DRAWCRV .....	112
DRAWPAR .....	114
DRAWPATH y DRAWPATHR .....	116
DRAWPFF .....	117
DRAWPIE .....	124
DRAWPOL .....	125
DRAWRDR .....	126
DUPLEX_off .....	128
DUPLEX_on .....	129
ENDARBM .....	130
ENDARBT .....	131
ENDBOOKLET .....	132
ENDCASE .....	133
ENDCLIP .....	134
ENDIFALL .....	135
ENDIMP .....	136
ENDJOB .....	137
ENDOFRUN .....	138
ENDOFSET .....	139
ENDPAGE .....	141
ENDPCC .....	142
ENDRPE .....	143
ENDTABLE .....	144
ENDXPD .....	145
ETA .....	146
ETCLIP .....	147
ETS .....	148
EXIST .....	149
EXIT .....	151
FBIND .....	152
FCALL .....	153
FILLOMR .....	154
FOREACH .....	156
FORMSHIFT .....	158
FROMLINE .....	159
Uso de FROMLINE para calcular la posición de impresión .....	161
FSHOW .....	165
GETDATE .....	166

GETFIELD.....	168
GETITEM .....	170
GOTOFRAME .....	172
ICALL .....	173
IF/ELSE/ELIF/ENDIF .....	176
IGNOREBT_off.....	178
IGNOREBT_on .....	179
ILAND .....	180
INDEXALIGN .....	181
INDEXBAT .....	182
INDEXCOLOR.....	183
INDEXFONT.....	185
INDEXKERN.....	187
INDEXLSP .....	189
INDEXOTL.....	190
INDEXPIF.....	191
INDEXRPE .....	193
INDEXSST .....	194
IPOINT.....	196
IREVERSE_off .....	197
IREVERSE_on.....	198
JOG_on y JOG_off .....	199
LAND .....	200
LMSKIP.....	201
MAKEVMFILE.....	202
MAKEVMFORM .....	203
MAXICODE.....	204
Requisitos de formato msgdata .....	205
MOVEH.....	209
MOVEHR.....	210
MOVETO.....	211
MSPP_on .....	212
MUL.....	213
NEWBACK.....	215
NEWFRAME.....	216
NEWFRONT .....	217
NEWGROUP .....	218
NEWPOS.....	219
NEWSIDE.....	220
NEWSTACK .....	221
NL.....	222
NMP_off.....	223
OMRINIT .....	224
Parámetro OMRconfig.....	226
OMRSHOW.....	229
ONEUP .....	230
ORIBL.....	231

ORITL.....	232
OVERPRINT_on.....	233
PAGEBRK.....	234
PAGERANGE .....	235
PDF417 .....	236
PDFBOUND .....	239
PDFDEST.....	240
PDFFORMOCG .....	241
PDFINFO.....	242
PDFOCG.....	243
PDFOPEN .....	244
PORT.....	245
PRECACHE.....	246
PROCESSDJE.....	247
QRCODE .....	249
QSTRIP_on.....	253
RELVAR .....	255
REPEAT .....	256
RESET .....	258
RPEDEF.....	259
RPEKEY .....	261
RSAVE .....	263
RUN.....	264
RUNDD.....	266
RUNPDF .....	270
RUNTIF .....	271
SAVEPP.....	273
SCALL.....	274
SETBAT .....	280
SETBFORM .....	283
SETBIDI.....	284
SETBUFSIZE .....	285
SETCJKENCMAP .....	286
SETCJKRULES.....	288
SETCOL.....	290
SETCOLWIDTH.....	294
SETCYCLECOPY .....	295
SETDATE .....	297
SETDBSEP .....	298
SETDLFILE.....	299
SETENCODING .....	300
SETEPATH.....	306
SETFINISHING.....	307
SETFONT.....	314
SETFORM.....	316
SETFPATH.....	319
SETFRAME .....	320

SETFTSP.....	321
SETFTSW .....	322
SETGEP.....	323
SETGRID .....	325
SETGUNIT.....	326
SETINDENT .....	327
SETIPATH.....	328
SETJDT .....	329
SETJPATH.....	331
SETKERN.....	332
SETLAYOUT .....	335
SETLFI.....	338
SETLKF .....	339
SETLMFILE .....	341
SETLSP .....	342
SETMARGIN .....	343
SETMAXBFORM.....	344
SETMAXCOPY .....	345
SETMAXFORM.....	346
SETMEDIA.....	347
SETMEDIAT .....	350
SETMPATH.....	351
SETMULTIUP .....	352
SETNMP.....	354
SETOBIN.....	355
SETOBINT .....	356
SETOTL.....	357
SETPAGEDEF.....	358
SETPAGENUMBER.....	359
SETPAGESIZE .....	361
SETPARAMS.....	362
SETPAT .....	363
SETPBRK .....	367
SETPCC.....	369
SETPCD .....	370
SETPIF.....	372
SETPPAT.....	376
SETPPATH .....	377
SETPROJECT .....	379
SETRCD .....	380
SETRES .....	382
SETRPE .....	383
SETRPEPREFIX.....	384
SETSKIP .....	386
SETTAB.....	388
SETTABS.....	389
SETTRAN .....	391

SETTPAT.....	392
SETTXB.....	394
SETTXC.....	395
SETTXS.....	398
SETUNIT.....	400
SETUTAB .....	401
SETV2HCONV .....	402
SETV2HTABLE.....	403
SETVAR.....	404
SETVFU .....	407
SETZEBRA .....	408
Comandos SHx.....	410
SHC y SHc .....	411
DESPLAZAMIENTO.....	412
SHIFTDATE.....	413
SHJ y SHj .....	414
SHL y SH .....	415
SHMF, SHMf y SHmf .....	416
SHP y SHp.....	419
SHPATH .....	425
SHPIT .....	427
SHR y SHr.....	429
SHROW.....	430
SHT y SHt.....	432
SHX.....	434
SKIPPAGE.....	436
SLIPSHEET.....	437
SOF_off .....	438
SORT .....	439
SPOOLNAME.....	440
STARTBOOKLET.....	441
STARTDBM.....	442
STARTLM .....	444
STARTOFSET.....	445
STARTXML .....	446
STOREVAR.....	447
SUB .....	449
TIFORI_off .....	451
TIFORI_on .....	452
TUMBLEDDUPLEX_off.....	453
TUMBLEDDUPLEX_on .....	454
TWOUP .....	455
UPDATE .....	456
USPS4CB .....	457
VPDISTPAT .....	460
XGF.....	461
XGFDEBUG.....	462



XGFEND.....	463
XGFENTRY.....	464
XGFRESDEF .....	465
XMLATL .....	467
XMLSYN.....	468
ZSORT .....	469
Marcadores.....	475
% .....	476
% %.....	477
%!.....	478
% %BoundingBox .....	479
% %EOD_XGF .....	480
% %EOF .....	481
% %PagesPerBooklet .....	482
% %XGF .....	483
\$\$name.....	484
[=name=] .....	485
BBOX.....	486
EXPAND .....	487
EXTVAR.....	488
Funciones de transformación.....	489
2OF5.....	491
64TO256 .....	492
BIDIRECCIONAL.....	493
BSTRIP .....	495
BTRIM .....	496
CASELOW.....	497
CASETI .....	498
CASEUP .....	499
CODE39.....	500
CODE128 y EAN128 .....	501
CS.....	504
DÍAS .....	505
EAN13 y EAN8.....	506
F2S .....	508
FORMAT.....	509
GETINTV.....	511
HMS.....	513
NOHYPHEN .....	514
POSTJPN.....	515
POSTNET .....	517
QSTRIP.....	518
REEMPLAZAR .....	519
REDONDEAR.....	520

SUBSTFONT .....	521
TRÍO .....	522
UPCA .....	524
UTF8TOLOC.....	525
VSUB.....	526
VSUB2.....	528
VSUB3.....	529
VSUB4.....	530
Variables.....	531
Variables de VIPP® por tipo .....	534
AUTOGRID .....	538
ATRÁS .....	539
BCOUNT .....	540
BLGRID.....	541
BPCOUNT .....	542
CLIP .....	543
COLW.....	544
CPCOUNT .....	545
CURLINE.....	546
D_DD .....	547
D_DOY .....	548
D_DWL .....	549
D_DWS.....	550
D_MO .....	551
D_MOL.....	552
D_MOS.....	553
D_YY.....	554
D_YYYY .....	555
DEVRES .....	556
DJDECMD .....	557
DJDEPAR.....	558
FRCOUNT .....	559
FRLEFT .....	560
GLT .....	561
GRIDSKIP.....	563
HCOLOR.....	564
HDISP .....	565
HPOS.....	566
HPOS2.....	567
IHEIGHT .....	568
IWIDTH .....	569
LNCOUNT .....	570
LPCOUNT.....	571
LPINDEX.....	572
LSP .....	573

MPR .....	574
OTCLIP y ITCLIP.....	575
PAGEH.....	576
PAGEW .....	577
PDFDEVICE .....	578
PDFPAGES.....	579
PLINES .....	580
PPCOUNT .....	583
PREV y NEXT.....	584
PRODUCT .....	585
PSIZE .....	586
RPCOUNT .....	589
RPEPOS.....	590
RPLEFT .....	592
SHEETH .....	593
SHEETW.....	594
SHPOS.....	595
SLENGTH .....	596
SSIZE .....	597
SVPOS .....	598
T_AMPM.....	599
T_HH .....	600
T_HH2.....	601
T_MM .....	602
T_SS .....	603
T_TZN.....	604
TLENGTH.....	605
TLGRID.....	606
TPATH .....	607
UV2L para efecto UV de dos capas .....	608
VARDataFileName .....	609
VDISP.....	610
VPOS.....	611
XGFVER .....	612
XMLATN.....	613
XMLATV .....	614
XMLDTH .....	615
XMLPAR .....	616
XMLPATH .....	617
XMLTAG.....	618
XMLVAL .....	619
YINIT .....	620
ZSPAGE.....	621
ZSRECNUM .....	622
ZSREPCNT.....	623
ZSREPIDX.....	624

Parámetros.....	625
Categorías del parámetro.....	626
Descripción del parámetro.....	627
/3D .....	627
/3DAngle.....	627
/3DThickness.....	628
/Across.....	628
/Align.....	628
/AlignChar .....	628
/AmPm .....	629
/BarSpace.....	629
/BarSpace2.....	629
/BGColor .....	629
/BGLineColor.....	630
/BookletMismatch .....	630
/BottomBleed.....	631
/BurstList .....	631
/CacheICALL .....	631
/Caching .....	631
/CellImage.....	632
/CellFill .....	632
/CellStroke .....	632
/CellText .....	633
/ChartDir .....	633
/ChartOrder .....	633
/CheckLabelOverlap.....	633
/ChkResources .....	634
/CJKunitcount.....	634
/ClearSubst .....	634
/ClusterMode .....	635
/ColorCycle .....	636
/ColorTable .....	636
/DaylightSaving.....	636
/DaysLong .....	636
/DaysShort.....	637
/DecimalPoint.....	637
/DefaultDate.....	637
/DefinedDate .....	637
/DotsPerModule.....	638
/Down.....	638
/DrawMode.....	638
/EmptyJobReport .....	639
/ELevel .....	639
/ExtraSpace .....	639
/FDecimalPoint.....	640
/FDigit .....	640
/FillOrder .....	640
/FitSpace.....	641
/FLZDigit .....	641
/FNSign.....	641
/Format.....	642
/FormatV.....	642

/FormatVI.....	642
/FormatPC .....	642
/FPSign .....	643
/FPunctuation.....	643
/HalfPie.....	643
/Height .....	643
/HGutter.....	644
/ImageDefRes.....	644
/IMPmode: .....	644
/Interpolate .....	644
/KeepRatio.....	645
/LabelColw.....	645
/LabelDashColor .....	645
/LabelDashWidth .....	646
/LabelOffset.....	646
/LayoutMarks .....	646
/LCDSmode.....	647
/LeftBleed.....	647
/LineDash .....	648
/LocalToUTF8.....	648
/Margins.....	649
/MarkLength .....	649
/MarkOffset .....	650
/MarkWidth .....	650
/MaxHeight .....	650
/MaxVal .....	650
/MediaSubst .....	650
/MergeValue .....	651
/MinDim.....	651
/MinVal.....	651
/MixPlexCount.....	651
/MonthsLong.....	652
/MonthsShort.....	652
/MUPduplex.....	652
/NSign .....	652
/OffsetValue.....	653
/OMRconfig .....	653
/OMRDir.....	654
/OMRHdisp .....	654
/OMRHskip .....	654
/OMRMap.....	654
/OMRMode.....	655
/OMRSlugChar.....	655
/OMRSlugFont.....	656
/OMRSlugSize.....	656
/OMRVdisp.....	656
/OMRVskip.....	657
/OMRWriteResp .....	657
/OriLine .....	657
/PageClip.....	658
/PageHeight.....	658
/PageRange .....	658
/PagesPerBooklet .....	659

/PageWidth.....	659
/PDFCropping.....	659
/PDFPage.....	660
/PDFXembed.....	661
/PDFXObject.....	661
/PlotSymbol .....	662
/PrintLabel .....	662
/PrintScale .....	663
/PrintValue.....	663
/ResCaseSense.....	664
/ResolvePath .....	664
/RightBleed.....	665
/Rotate.....	665
/RowHeight.....	665
/ScaleStep .....	665
/ShadeAdjust .....	666
/SHPWrap.....	666
/SliceBurst.....	667
/SliceSepColor .....	667
/SliceSepWidth .....	667
/SpotLabels.....	668
/SpotOffset.....	668
/SpotSize.....	668
/Stack .....	669
/TableStroke.....	669
/TextAtt.....	669
/TextFilter .....	669
/TextFit .....	670
/TimeZone.....	670
/TimeZoneName.....	670
/TopBleed .....	671
/TransWeight .....	671
/ValueColor .....	671
/VGutter.....	671
/Width .....	672
/XFloat.....	672
/XMLMisVal.....	672
/ZSRepeatField .....	672
Combinaciones compuestas.....	675
Expresiones aritméticas .....	676
Operadores de prueba y expresiones condicionales.....	679
Ejemplos .....	679
CIEQ y CINE .....	679
/HOLD .....	680
Mensajes de error.....	681
Gestión de errores.....	682
Errores de PostScript .....	682
Hoja de errores.....	682
Ejecución de archivos de enlace.....	682
Encontrar la causa de un error .....	682

Lista de errores de VIPP® .....	684
VIPP_access_denied.....	684
VIPP_AFM_parsing_error.....	684
VIPP_ambiguous_name in _name .....	684
VIPP_buffer_overflow .....	684
VIPP_corrupted_or_unsupported_image_file .....	684
VIPP_invalid_align en SHMF .....	684
VIPP_invalid_align en SHP.....	684
VIPP_invalid_align en SHX.....	685
VIPP_invalid_booklet_length.....	685
VIPP_invalid_color .....	685
VIPP_invalid_combination in Multi-up_COLLATE_off .....	685
VIPP_invalid_combination in STOREVAR_file_must_exist .....	685
VIPP_invalid_combination in STOREVAR_VIeCmulti .....	685
VIPP_invalid_combination en UV2L .....	685
VIPP_invalid_contents in ENDPAGE .....	685
VIPP_invalid_font .....	685
VIPP_invalid_license_file .....	685
VIPP_invalid_PIF_type.....	686
VIPP_invalid_PN_option.....	686
VIPP_invalid_syntax en RPE .....	686
VIPP_invalid_syntax en SETBAT .....	686
VIPP_invalid_syntax in SETMULTIUP .....	686
VIPP_invalid_syntax en SHP.....	686
VIPP_invalid_variable_name.....	686
VIPP_invalid_VSUB .....	686
VIPP_length_error in ENDPCC .....	686
VIPP_length_error in RPE.....	687
VIPP_license_failed .....	687
VIPP_misplaced en SETPAGESIZE.....	687
VIPP_OMR_invalid_response_string .....	687
VIPP_PDF417_data_limit_exceeded .....	687
VIPP_PDF417_invalid_mode.....	687
VIPP_plane_number_out_of_range .....	687
VIPP_POSTNET_invalid_digit .....	687
VIPP_POSTNET_invalid_length.....	687
VIPP_RPE_invalid_prefix.....	688
VIPP_RPE_invalid_prefix_length .....	688
VIPP_SETVAR_invalid_name .....	688
VIPP_unable_to_locate.....	688
Otros errores .....	689
Error de limitcheck.....	689
PostScript .....	689
Error PostScript Error - dictfull / Offending Command: def .....	689
Páginas seleccionadas: <first> <last> .....	689
Folletos seleccionados: <first> <last> .....	690
Error de desbordamiento de la pila en Ghostscript.....	690
Sugerencias de programación.....	691
Consumir frente a ejecutar .....	692
Origen 0,0 para la posición de la página lógica y objetos .....	693
Control de cyclecopy .....	696

Uso de formularios y JDT de ciclos .....	696
Fecha y hora .....	697
Diseño y depuración .....	698
Diseño de la aplicación .....	698
Herramientas de depuración y documentación .....	698
Creación y mantenimiento de recursos .....	699
Fuentes y colores .....	701
Aplicar atributos para las fuentes .....	701
Tonos de color .....	702
Transparencia de color .....	702
Colores directos frente a colores de procesamiento .....	703
Fuentes de códigos de barras y fuentes fijas .....	704
Espaciado .....	704
Fuentes de múltiples bytes .....	705
Fuentes recodificadas e instrucciones de run o RUN .....	705
Configuración de tamaño automático de las fuentes .....	706
Configuración de negrita en el centro de un registro o de una línea .....	706
Simulación de color sólido estucado y sin estucar .....	706
Impresión especializada .....	707
Fuentes GlossMark y de correlación (GL y CR) .....	708
Uso de fuentes de varios bytes (CJK) .....	709
Uso de archivos PostScript con fuentes integradas como formularios .....	709
Marcadores .....	711
Comandos y funciones varios .....	712
Creación de un archivo PostScript independiente de VIPP® .....	712
Incorporación de color de realce en formularios de descomposición .....	712
Inclusión de un objeto no VIPP® en un trabajo VIPP® .....	712
Texto imprimible o referencia ( ) .....	713
Impresión con caracteres especiales .....	713
VI Compose y EBCDIC .....	713
Control del dispositivo de salida .....	715
Control de página .....	715
Diseño de página .....	715
Marca de página .....	722
Funciones interactivas de PDF .....	731
Marcadores .....	731
Enlaces .....	731
Notas .....	731
Claves predeterminadas y palabras clave .....	733
Agregación de GEPkeys .....	733
Mejora de la diferenciación de sombreado mediante DDG .....	733
Palabras clave reservadas .....	733
VIPP® Colorkeys .....	734
Imprimir procesamiento de archivos .....	735
Compatibilidad de folletos .....	735
Soporte de materiales de impresión .....	736
ZSORT y agrupación de registros .....	737
Control de carro de la impresora .....	738
Control de recursos .....	739
Elementos de RPE .....	740



Ignorar el número de líneas al principio de cada archivo .....	740
Sobreimpresión de una línea por lograr el efecto de fuente negrita .....	740
Información de comando RPE .....	741
Procedimiento de alineación .....	744
Uso del color de realce para imprimir números negativos .....	744
Funciones de transformación .....	746



# Descripción general del lenguaje VIPP®

Este capítulo incluye:

Foro de clientes de VI Suite .....	20
Descarga de fuentes .....	21
Uso de recursos PDF con VIPP® .....	22
FreeFlow® VI Compose Open Edition .....	23
Lista de comandos VIPP® .....	24

Este manual documenta el lenguaje de programación Variable Information Production PrintWare (VIPP®). VIPP® es un lenguaje de programación para el diseño de aplicaciones de información variable (VI). El lenguaje VIPP® permite describir la estructura y la lógica de los datos variables que se procesarán, y producir la apariencia de las páginas. El lenguaje VIPP® es un lenguaje de composición variable de documentos que proporciona una capacidad y flexibilidad casi ilimitadas a los usuarios.

El *Manual de referencia del lenguaje VIPP®* contiene información acerca de los comandos, los marcadores, las funciones de transformación, las variables, los parámetros y los mensajes de error de VIPP® y da respuesta a muchas preguntas frecuentes acerca del lenguaje VIPP®. Para obtener información complementaria y descripciones de las utilidades, los archivos y los recursos de VIPP®, y para obtener información sobre FreeFlow VI Compose, consulte la *Guía del usuario de VI Compose de FreeFlow®*.

La descripción de cada comando de VIPP® incluye la siguiente información:

- Sintaxis de comandos
- Modos aplicables por el uso
- Comandos relacionados

La información acerca del lenguaje VIPP® y los componentes se agrupa en estos temas:

- Comandos de VIPP®
- Marcadores
- Transformar funciones
- Variables
- Parámetros
- Combinaciones compuestas

## Foro de clientes de VI Suite

Xerox proporciona un foro de asistencia técnica a la comunidad. El foro de clientes de VI Suite ahora es parte de un foro de asistencia más grande, que le permite publicar y revisar información acerca de los productos y servicios de Xerox desde una sola ubicación. Tómese un minuto para iniciar sesión en la comunidad del foro de clientes: <https://VIPPSupport.xerox.com>.

## Descarga de fuentes

Para descargar las fuentes de Impresión especializada y Códigos de barras, vaya a [www.xerox.com/support](http://www.xerox.com/support) y busque **VIPP**. Seleccione **Software**.

Algunos programas de información variable, fuentes de impresión especiales y fuentes de códigos de barras pueden ser adquiridos por los clientes en los Estados Unidos, mediante una tarjeta de crédito. Los productos descargables se encuentran en la Xerox eStore <https://buy.xerox.com>.



Nota: Cuando descargue fuentes, se le redireccionará a una página donde deberá leer el Acuerdo de licencia para el usuario final. Para descargar las fuentes, revise y acepte el Acuerdo de licencia del usuario final. Si no acepta el Acuerdo de licencia para el usuario final, el programa abandonará la página de descarga de fuentes.

## Uso de recursos PDF con VIPP®

VI Compose admite recursos PDF en el antiguo RIP PostScript, Adobe PDF Print Engine (APPE) y en las diversas funcionalidades de salida de PDF de FreeFlow VI Suite. Para obtener más información, consulte *Uso de recursos PDF con opciones VIPP®, APPE, PDF/VT y EPS integrado* en la *Guía del usuario de FreeFlow VI Compose*.

## FreeFlow® VI Compose Open Edition

El software FreeFlow VI Compose Open Edition o VIC(OE) es una versión modificada del software VI Compose admitida en motores de impresión Xerox. VIC(OE) ha sido modificado para que pueda instalarse en equipos de otras marcas y para que su licencia pueda activarse a través de los canales de Xerox habituales.

Debido a ciertas diferencias de funcionamiento interno entre los equipos Xerox y de otras marcas, pueden existir algunas limitaciones. Estas limitaciones afectan principalmente a las áreas de alimentación y acabado, pero pueden existir también en otras áreas. Por ejemplo, en los trabajos donde VIPP® intenta escribir información de posición del archivo en el dispositivo. Xerox no tiene control sobre los dispositivos de otros fabricantes. Las operaciones que son habituales en equipos de Xerox podrían no estar permitidas en equipos de otras marcas. Debido a estas posibles limitaciones, se recomienda la validación total de los trabajos que se vayan a realizar en equipos de terceros antes de ejecutar trabajos en modo de producción.

Xerox no dispone de asistencia técnica para equipos de producción de otras marcas. Cualquier problema comunicado será validado en un equipo de producción de Xerox similar. Si el equipo de producción Xerox experimenta el mismo problema, Xerox lo puede solucionar con un parche de software que proporcionará al cliente para su instalación y verificación. Si no se soluciona el problema en el dispositivo de producción no de Xerox, es posible que Xerox no pueda proporcionar más asistencia.

Para obtener más información, consulte la descripción general y la instalación de *FreeFlow® VI Compose (Open Edition)*.

## Lista de comandos VIPP®

La tabla contiene una lista de todos los comandos VIPP®, los marcadores, las funciones de transformación y las variables en orden alfabético. Además del tipo de comando, la información incluye su función. Si usa este documento en formato PDF o Ayuda, puede hacer clic en cualquiera de los nombres de comandos en la lista para acceder a la descripción del comando.

Las funciones de los comandos se agrupan en las categorías descritas aquí. Las descripciones de las categorías solo proporcionan ejemplos generales y no son inclusivas.

- Los comandos Cyclecopy controlan el número de copias y cómo son gestionadas por la impresora.
- Las funciones de fecha y hora permiten configurar la información de fecha y hora.
- Los comandos de diseño y depuración se utilizan para diseñar los trabajos VIPP® y para facilitar la depuración de las aplicaciones que cree.
- Los comandos de fuente y color controlan el tipo y la apariencia de las fuentes y el fondo del texto de los trabajos VIPP®.
- Los comandos de control del dispositivo de salida controlan las opciones en el dispositivo de salida. Estos comandos permiten configurar el grapado de juegos, el desplazamiento, las opciones de producción a 2 caras, los requisitos de materiales de impresión, la resolución de salida, etc.
- Los comandos de control de la página controlan elementos como el salto de página, la imposición de páginas, la omisión y la impresión de páginas, la inserción de hojas intermedias, etc.
- Los comandos de diseño de página determinan la apariencia del trabajo. Los comandos de diseño de página incluyen comandos para configurar la anchura y el margen de las columnas, definir los formularios utilizados, configurar la orientación de la página, etc.
- Los comandos de marcación de páginas permiten insertar imágenes, segmentos, textos, glifos, etc. También permiten configurar DDG y parámetros de formato, y diseñar gráficos
- Los comandos de procesamiento de PCC permiten configurar opciones de carácter de control de la impresora.
- Los comandos de control de PDF son un conjunto de comandos de VIPP® que permiten crear elementos interactivos cuando un trabajo VIPP® se representa en un documento PDF.
- Los comandos de procesamiento de archivos de impresión permiten determinar cómo se procesan los archivos y configurar los modos nativos, de línea, de base de datos y XML, etc.
- Los comandos de control de recursos permiten usar las opciones de almacenamiento en caché y precaché, configurar rutas y patrones de los trabajos y proyectos de VI, y almacenar datos en la memoria como un archivo virtual.
- El procesamiento de RPE permite utilizar las opciones de procesamiento de datos de entrada de procesamiento de registros.
- Los comandos de control de transformación permiten cambiar el texto en el documento, quitar espacios, crear códigos de barras, etc.



VARIABLES, FUNCIONES DE TRANSFORMACIÓN, MARCADORES Y COMANDOS VIPP®	TIPO: C = COMANDO, M = MARCADOR, T = FUNCIÓN DE TRANSFORMACIÓN, V = VARIABLE	FUNCIÓN
%	M	
% %	M	
%!	M	
% %BoundingBox	M	
% %EOD XGF	M	
% %EOF	M	
% %PagesPerBooklet	M	
% %XGF	M	
\$\$name.	M	
++/-	C	Varios
[=name=]	M	
2OF5	T	Transformación de control
64TO256	T	Transformación de control
ABSPPOS	C	Marca de página
ACCLOG	C	Varios
ADD	C	Varios
ADVPAGE	C	Control de página
AUTOGRID	V	Diseño y depuración
AZTEC	C	Marca de página
ACK	V	Control del dispositivo de salida
ACKSP off	C	Imprimir procesamiento de archivos
ACKSPF on	C	Imprimir procesamiento de archivos
ACKSPP on	C	Imprimir procesamiento de archivos
BCALL	C	Marca de página

VARIABLES, FUNCIONES DE TRANSFORMACIÓN, MARCADORES Y COMANDOS VIPP®	TIPO: C = COMANDO, M = MARCADOR, T = FUNCIÓN DE TRANSFORMACIÓN, V = VARIABLE	FUNCIÓN
BBOX	M	
BCOUNT	V	Varios
BEGINARBM	C	Imprimir procesamiento de archivos
BEGINARBT	C	Imprimir procesamiento de archivos
BEGINFRAME	C	Varios
BEGINIMP	C	Control de página
BEGINPAGE	C	Varios
BEGINPCC	C	Procesamiento de PCC
BEGINRPE	C	Procesamiento de RPE
BEGINTABLE	C	Marca de página
BEGINXPD	C	Imprimir procesamiento de archivos
BIDIRECCIONAL	T	Imprimir procesamiento de archivos
BLGRID	V	Diseño y depuración
BOOKLETRANGE	C	Imprimir procesamiento de archivos
BOOKMARK	C	Control de PDF
BPCOUNT	V	Varios
BSTRIP	T	Transformación de control
BSTRIP off	C	Imprimir procesamiento de archivos
BTA	C	Imprimir procesamiento de archivos
BTRIM	T	Transformación de control
BTS	C	Imprimir procesamiento de archivos
CACHE	C	Control de recursos

VARIABLES, FUNCIONES DE TRANSFORMACIÓN, MARCADORES Y COMANDOS VIPP®	TIPO: C = COMANDO, M = MARCADOR, T = FUNCIÓN DE TRANSFORMACIÓN, V = VARIABLE	FUNCIÓN
CASE	C	Varios
CASELOW	T	Transformación de control
CASETI	T	Transformación de control
CASEUP	T	Transformación de control
CHKPOINT	C	Cyclecopy
CLIP	V	Marca de página
CODE39	T	Transformación de control
CODE128 y EAN128	T	Transformación de control
COLLATE dbm	C	Cyclecopy
COLLATE off	C	Cyclecopy
COLLATE on	C	Cyclecopy
COLW	V	Diseño de página
COPYRANGE	C	Cyclecopy
CPCOUNT	V	Varios
CS	T	Transformación de control
CURLINE	V	Procesamiento de RPE
CUTMARK	C	Marca de página
D DD	V	Fecha y hora
D DOY	V	Fecha y hora
D DWL	V	Fecha y hora
D DWS	V	Fecha y hora
D MO	V	Fecha y hora
D MOL	V	Fecha y hora

VARIABLES, FUNCIONES DE TRANSFORMACIÓN, MARCADORES Y COMANDOS VIPP®	TIPO: C = COMANDO, M = MARCADOR, T = FUNCIÓN DE TRANSFORMACIÓN, V = VARIABLE	FUNCIÓN
D MOS	V	Fecha y hora
D YY	V	Fecha y hora
D YYYY	V	Fecha y hora
DATAMATRIX	C	Marca de página
DÍAS	T	Fecha y hora
DEVRES	V	Control del dispositivo de salida
DIV	C	Varios
DJDEBEGIN	C	Imprimir procesamiento de archivos
DJDECMD	V	Imprimir procesamiento de archivos
DJDEPAR	V	Imprimir procesamiento de archivos
DRAWB y DRAWBR	C	Marca de página
DRAWBAR	C	Marca de página
DRAWBC	C	Marca de página
DRAWBM y DRAWBRM	C	Marca de página
DRAWC	C	Marca de página
DRAWCRV	C	Marca de página
DRAWPAR	C	Marca de página
DRAWDRAW y DRAWDRAWHR	C	Marca de página
DRAWPFF	C	Marca de página
DRAWPIE	C	Marca de página
DRAWPOL	C	Marca de página
DRAWRDR	C	Marca de página
DUPLEX_off	C	Control del dispositivo de salida
DUPLEX_on	C	Control del dispositivo

VARIABLES, FUNCIONES DE TRANSFORMACIÓN, MARCADORES Y COMANDOS VIPP®	TIPO: C = COMANDO, M = MARCADOR, T = FUNCIÓN DE TRANSFORMACIÓN, V = VARIABLE	FUNCIÓN
		de salida
EAN13/EAN8	T	Transformación de control
ENDARBM	C	Imprimir procesamiento de archivos
ENDARBT	C	Imprimir procesamiento de archivos
ENDBOOKLET	C	Imprimir procesamiento de archivos
ENDCASE	C	Varios
ENDCLIP	C	Marca de página
ENDIFALL	C	Procesamiento de RPE
ENDIMP	C	Control de página
ENDJOB	C	Imprimir procesamiento de archivos
ENDOFRUN	C	Control del dispositivo de salida
ENDOFSET	C	Control del dispositivo de salida
ENDPAGE	C	Varios
ENDPCC	C	Procesamiento de PCC
ENDRPE	C	Procesamiento de RPE
ENDTABLE	C	Marca de página
ENDXPD	C	Imprimir procesamiento de archivos
ETA	C	Imprimir procesamiento de archivos
ETCLIP	C	Marca de página
ETS	C	Imprimir procesamiento de archivos
EXIST	C	Control de recursos
EXIT	C	Imprimir

VARIABLES, FUNCIONES DE TRANSFORMACIÓN, MARCADORES Y COMANDOS VIPP®	TIPO: C = COMANDO, M = MARCADOR, T = FUNCIÓN DE TRANSFORMACIÓN, V = VARIABLE	FUNCIÓN
		procesamiento de archivos
EXPAND	M	
EXTVAR	M	
F2S	T	Transformación de control
FBIND	C	Control de recursos
FCALL	C	Marca de página
FILLOMR	C	Marca de página
FOREACH	C	Varios
FORMAT	T	Transformación de control
FORMSHIFT	C	Diseño de página
FRCOUNT	V	Imprimir procesamiento de archivos
FRLEFT	V	Imprimir procesamiento de archivos
FROMLINE	C	Procesamiento de RPE
FSHOW	C	Diseño y depuración
GETDATE	C	Fecha y hora
GETFIELD	C	Varios
GETINTV	T	Transformación de control
GETITEM	C	Varios
GLT	V	Fuente y color
GOTOFRAME	C	Control de página
GRIDSKIP	V	Control de página
HCOLOR	V	Fuente y color
HDISP	V	Marca de página
HMS	T	Transformación de

VARIABLES, FUNCIONES DE TRANSFORMACIÓN, MARCADORES Y COMANDOS VIPP®	TIPO: C = COMANDO, M = MARCADOR, T = FUNCIÓN DE TRANSFORMACIÓN, V = VARIABLE	FUNCIÓN
		control
HPOS	V	Diseño de página
HPOS2	V	Diseño de página
ICALL	C	Marca de página
IF/ELSE/ELIF/ENDIF	C	Varios
IGNOREBT off	C	Varios
IGNOREBT on	C	Varios
IHEIGHT	V	Diseño de página
ILAND	C	Diseño de página
INDEXALIGN	C	Fuente y color
INDEXBAT	C	Fuente y color
INDEXCOLOR	C	Fuente y color
INDEXFONT	C	Fuente y color
INDEXKERN	C	Fuente y color
INDEXLSP	C	Fuente y color
INDEXOTL	C	Fuente y color
INDEXPIF	C	Imprimir procesamiento de archivos
INDEXRPE	C	Procesamiento de RPE
INDEXSST	C	Fuente y color
IPOINT	C	Diseño de página
IREVERSE off	C	Marca de página
IREVERSE on	C	Marca de página
IWIDTH	V	Diseño de página
JOG_on y JOG_off	C	Control del dispositivo de salida
LAND	C	Diseño de página
LMSKIP	C	Imprimir

VARIABLES, FUNCIONES DE TRANSFORMACIÓN, MARCADORES Y COMANDOS VIPP®	TIPO: C = COMANDO, M = MARCADOR, T = FUNCIÓN DE TRANSFORMACIÓN, V = VARIABLE	FUNCIÓN
		procesamiento de archivos
LNCOUNT	V	Varios
LPCOUNT	V	Varios
LPINDEX	V	Varios
LSP	V	Diseño de página
MAKEVMFILE	C	Control de recursos
MAKEVMFORM	C	Control de recursos
MAXICODE	C	Marca de página
MOVEH	C	Marca de página
MOVEHR	C	Marca de página
MOVETO	C	Marca de página
MPR	V	Fuente y color
MSPP on	C	Varios
MUL	C	Varios
NEWBACK	C	Control de página
NEWFRAME	C	Control de página
NEWFRONT	C	Control de página
NEWGROUP	C	Procesamiento de RPE
NEWPOS	C	Procesamiento de RPE
NEWSIDE	C	Control de página
NEWSTACK	C	Control de página
NL	C	Marca de página
NMP off	C	Diseño y depuración
NOHYPHEN	T	Transformación de control
OMRINIT	C	Marca de página
OMRSHOW	C	Marca de página



VARIABLES, FUNCIONES DE TRANSFORMACIÓN, MARCADORES Y COMANDOS VIPP®	TIPO: C = COMANDO, M = MARCADOR, T = FUNCIÓN DE TRANSFORMACIÓN, V = VARIABLE	FUNCIÓN
ONEUP	C	Diseño de página
ORIBL	C	Diseño de página
ORITL	C	Diseño de página
OTCLIP y ITCLIP	V	Marca de página
OVERPRINT on	C	Imprimir procesamiento de archivos
PAGEBRK	C	Control de página
PAGEH	V	Diseño de página
PAGERANGE	C	Imprimir procesamiento de archivos
PAGEW	V	Diseño de página
PDF417	C	Marca de página
PDFBOUND	C	Control de PDF
PDFDEST	C	Control de PDF
PDFDEVICE	V	Control de PDF
PDFFORMOCG	C	Control de PDF
PDFINFO	C	Control de PDF
PDFOCG	C	Control de PDF
PDFOPEN	C	Control de PDF
PDFPAGES	V	Control de PDF
PLINES	V	Diseño de página
PORT	C	Diseño de página
POSTJPN	T	Transformación de control
POSTNET	T	Transformación de control
PPCOUNT	V	Imprimir procesamiento de archivos
PRECACHE	C	Control de recursos
PREV y NEXT	V	Varios

VARIABLES, FUNCIONES DE TRANSFORMACIÓN, MARCADORES Y COMANDOS VIPP®	TIPO: C = COMANDO, M = MARCADOR, T = FUNCIÓN DE TRANSFORMACIÓN, V = VARIABLE	FUNCIÓN
PROCESSDJE	C	Imprimir procesamiento de archivos
PRODUCT	V	Varios
PSIZE	V	Diseño de página
QRCODE	C	Marca de página
QSTRIP	T	Transformación de control
QSTRIP on	C	Imprimir procesamiento de archivos
RELVAR	C	Varios
REPEAT	C	Cyclecopy
REEMPLAZAR	T	Transformación de control
RESET	C	Varios
REDONDEAR	T	Varios
RPCOUNT	V	Cyclecopy
RPEDEF	C	Procesamiento de RPE
RPEKEY	C	Procesamiento de RPE
RPEPOS	V	Procesamiento de RPE
RPLEFT	V	Cyclecopy
RSAVE	C	Varios
RUN	C	Varios
RUNDD	C	Varios
RUNPDF	C	Varios
RUNTIF	C	Varios
Comandos de VIPP®	C	Marca de página
SCALL	C	Marca de página
SETBAT	C	Fuente y color
SETBFORM	C	Diseño de página

VARIABLES, FUNCIONES DE TRANSFORMACIÓN, MARCADORES Y COMANDOS VIPP®	TIPO: C = COMANDO, M = MARCADOR, T = FUNCIÓN DE TRANSFORMACIÓN, V = VARIABLE	FUNCIÓN
SETBIDI	C	Imprimir procesamiento de archivos
SETBUFSIZE	C	Imprimir procesamiento de archivos
SETCJKENCMAP	C	Fuente y color
SETCJKRULES	C	Fuente y color
SETCOL	C	Fuente y color
SETCOLWIDTH	C	Diseño de página
SETCYCLECOPY	C	Cyclecopy
SETDATE	C	Fecha y hora
SETDBSEP	C	Imprimir procesamiento de archivos
SETDLFILE	C	Imprimir procesamiento de archivos
SETENCODING	C	Fuente y color
SETEPATH	C	Control de recursos
SETFINISHING	C	Control del dispositivo de salida
SETFONT	C	Fuente y color
SETFORM	C	Diseño de página
SETFPATH	C	Control de recursos
SETFRAME	C	Diseño de página
SETFTSP	C	Fuente y color
SETFTSW	C	Fuente y color
SETGEP	C	Fuente y color
SETGRID	C	Diseño de página
SETGUNIT	C	Diseño de página
SETINDENT	C	Marca de página
SETIPATH	C	Control de recursos

VARIABLES, FUNCIONES DE TRANSFORMACIÓN, MARCADORES Y COMANDOS VIPP®	TIPO: C = COMANDO, M = MARCADOR, T = FUNCIÓN DE TRANSFORMACIÓN, V = VARIABLE	FUNCIÓN
SETJDT	C	Imprimir procesamiento de archivos
SETJPATH	C	Control de recursos
SETKERN	C	Fuente y color
SETLAYOUT	C	Marca de página
SETLFI	C	Diseño de página
SETLKF	C	Diseño de página
SETLMFILE	C	Imprimir procesamiento de archivos
SETLSP	C	Diseño de página
SETMARGIN	C	Diseño de página
SETMAXBFORM	C	Diseño de página
SETMAXCOPY	C	Cyclecopy
SETMAXFORM	C	Diseño de página
SETMEDIA	C	Control del dispositivo de salida
SETMEDIAT	C	Control del dispositivo de salida
SETMPATH	C	Control de recursos
SETMULTIUP	C	Diseño de página
SETNMP	C	Imprimir procesamiento de archivos
SETOBIN	C	Control del dispositivo de salida
SETOBINT	C	Control del dispositivo de salida
SETOTL	C	Fuente y color
SETPAGEDEF	C	Control de recursos
SETPAGENUMBER	C	Diseño de página
SETPAGESIZE	C	Diseño de página

VARIABLES, FUNCIONES DE TRANSFORMACIÓN, MARCADORES Y COMANDOS VIPP®	TIPO: C = COMANDO, M = MARCADOR, T = FUNCIÓN DE TRANSFORMACIÓN, V = VARIABLE	FUNCIÓN
SETPARAMS	C	Imprimir procesamiento de archivos
SETPAT	C	Fuente y color
SETPBRK	C	Control de página
SETPCC	C	Procesamiento de PCC
SETPCD	C	Varios
SETPIF	C	Control de PDF
SETPPAT	C	Fuente y color
SETPPATH	C	Control de recursos
SETPROJECT	C	Control de recursos
SETRCD	C	Procesamiento de RPE
SETRES	C	Control del dispositivo de salida
SETRPE	C	Procesamiento de RPE
SETRPEPREFIX	C	Procesamiento de RPE
SETSKIP	C	Procesamiento de PCC
SETTAB	C	Diseño de página
SETTABS	C	Diseño de página
SETTPAT	C	Fuente y color
SETTRAN	C	Fuente y color
SETTXB	C	Fuente y color
SETTXC	C	Fuente y color
SETTXS	C	Fuente y color
SETUNIT	C	Diseño de página
SETUTAB	C	Diseño de página
SETV2HCONV	C	Fuente y color
SETV2HTABLE	C	Fuente y color
SETVAR	C	Varios

VARIABLES, FUNCIONES DE TRANSFORMACIÓN, MARCADORES Y COMANDOS VIPP®	TIPO: C = COMANDO, M = MARCADOR, T = FUNCIÓN DE TRANSFORMACIÓN, V = VARIABLE	FUNCIÓN
SETVFU	C	Procesamiento de PCC
SETZEBRA	C	Diseño de página
SHC y SHc	C	Marca de página
SHEETH	V	Diseño de página
SHEETW	V	Diseño de página
DESPLAZAMIENTO	C	Diseño de página
SHIFTDATE	C	Fecha y hora
SHJ y SHj	C	Marca de página
SHL y SH	C	Marca de página
SHMF, SHMf y SHmf	C	Marca de página
SHP y SHp	C	Marca de página
SHPATH	C	Marca de página
SHPIT	C	Marca de página
SHPOS	V	Marca de página
SHR y SHr	C	Marca de página
SHROW	C	Marca de página
SHT y SHt	C	Marca de página
SHX	C	Marca de página
SKIPPAGE	C	Control de página
SLENGTH	V	Diseño de página
SLIPSHEET	C	Control de página
SOF off	C	Imprimir procesamiento de archivos
SORT	C	Varios
SPOOLNAME	C	Cyclecopy
SSIZE	V	Diseño de página
STARTBOOKLET	C	Imprimir

VARIABLES, FUNCIONES DE TRANSFORMACIÓN, MARCADORES Y COMANDOS VIPP®	TIPO: C = COMANDO, M = MARCADOR, T = FUNCIÓN DE TRANSFORMACIÓN, V = VARIABLE	FUNCIÓN
		procesamiento de archivos
STARTDBM	C	Imprimir procesamiento de archivos
STARTLM	C	Imprimir procesamiento de archivos
STARTOFSET	C	Control del dispositivo de salida
STARTXML	C	Imprimir procesamiento de archivos
STOREVAR	C	Control de recursos
SUB	C	Varios
SUBSTFONT	T	Control de recursos
SVPOS	V	Marca de página
T AMPM	V	Fecha y hora
T HH	V	Fecha y hora
T HH2	V	Fecha y hora
T MM	V	Fecha y hora
T SS	V	Fecha y hora
T TZN	V	Fecha y hora
TIFORI off	C	Diseño de página
TIFORI on	C	Diseño de página
TLENGTH	V	Diseño y depuración
TLGRID	V	Diseño y depuración
TPATH	V	Marca de página
TRÍO	T	Imprimir procesamiento de archivos
TUMBLEDDUPLEX_off	C	Control del dispositivo de salida
TUMBLEDDUPLEX_on	C	Control del dispositivo de salida

VARIABLES, FUNCIONES DE TRANSFORMACIÓN, MARCADORES Y COMANDOS VIPP®	TIPO: C = COMANDO, M = MARCADOR, T = FUNCIÓN DE TRANSFORMACIÓN, V = VARIABLE	FUNCIÓN
TWOUP	C	Diseño de página
UPCA	T	Transformación de control
UPDATE	C	Varios
USPS4CB	C	Marca de página
UTF8TOLOC	T	Transformación de control
UV2L para efecto UV de dos capas	V	Imprimir procesamiento de archivos
VARDataFileName	V	Marca de página
VDISP	V	Marca de página
VPDISTPAT	C	Fuente y color
VPOS	V	Diseño de página
VSUB	T	Transformación de control
VSUB2	T	Transformación de control
VSUB3	T	Transformación de control
VSUB4	T	Transformación de control
XGF	C	Varios
XGFDEBUG	C	Diseño y depuración
XGFEND	C	Imprimir procesamiento de archivos
XGFENTRY	C	Varios
XGFRESDEF	C	Control de recursos
XGFVER	V	Varios
XMLATL	C	Imprimir procesamiento de archivos
XMLATN	V	Imprimir procesamiento de archivos



VARIABLES, FUNCIONES DE TRANSFORMACIÓN, MARCADORES Y COMANDOS VIPP®	TIPO: C = COMANDO, M = MARCADOR, T = FUNCIÓN DE TRANSFORMACIÓN, V = VARIABLE	FUNCIÓN
XMLATV	V	Imprimir procesamiento de archivos
XMLDTH	V	Imprimir procesamiento de archivos
XMLPAR	V	Imprimir procesamiento de archivos
XMLPATH	V	Imprimir procesamiento de archivos
XMLSYN	C	Imprimir procesamiento de archivos
XMLTAG	V	Imprimir procesamiento de archivos
XMLVAL	V	Imprimir procesamiento de archivos
YINIT	V	Diseño de página
ZSORT	C	Imprimir procesamiento de archivos
ZSPAGE	V	Imprimir procesamiento de archivos
ZSRECNUM	V	Imprimir procesamiento de archivos
ZSREPCNT	V	Imprimir procesamiento de archivos
ZSREPIDX	V	Imprimir procesamiento de archivos



# Comandos de VIPP®

Este capítulo incluye:

++ y – .....	52
ABSPOS .....	53
ACCLOG .....	54
ADD .....	56
ADV PAGE .....	58
AZTEC .....	59
BACKSP_off .....	61
BACKSPF_on .....	62
BACKSPP_on .....	63
BCALL .....	64
BEGINARBM .....	65
BEGINARBT .....	66
BEGINFRAME .....	68
BEGINIMP .....	69
BEGINPAGE .....	71
BEGINPCC .....	73
BEGINRPE .....	74
BEGINTABLE .....	75
BEGINXPD .....	76
BOOKLETRANGE .....	77
BOOKMARK .....	78
BSTRIP_off .....	80
BTA .....	81
BTS .....	83
CACHE .....	85
CASE .....	87
CHKPOINT .....	89
COLLATE_dbm .....	90
COLLATE_off .....	91
COLLATE_on .....	92
COPYRANGE .....	93
CUTMARK .....	94

DATAMATRIX.....	95
DEFINELAYOUT .....	97
DIV.....	98
DJDEBEGIN.....	99
DRAWB y DRAWBR.....	100
DRAWBAR.....	102
DRAWBC .....	104
DRAWBC (extensión para admitir GS1 Databar).....	106
DRAWBM y DRAWBRM .....	110
DRAWC .....	111
DRAWCRV.....	112
DRAWPAR .....	114
DRAWPATH y DRAWPATHR .....	116
DRAWPFF .....	117
DRAWPIE.....	124
DRAWPOL.....	125
DRAWRDR .....	126
DUPLEX_off .....	128
DUPLEX_on.....	129
ENDARBM.....	130
ENDARBT .....	131
ENDBOOKLET .....	132
ENDCASE.....	133
ENDCLIP.....	134
ENDIFALL .....	135
ENDIMP .....	136
ENDJOB.....	137
ENDOFRUN.....	138
ENDOFSET .....	139
ENDPAGE.....	141
ENDPCC .....	142
ENDRPE.....	143
ENDTABLE .....	144
ENDXPD .....	145

ETA .....	146
ETCLIP .....	147
ETS.....	148
EXIST.....	149
EXIT .....	151
FBIND.....	152
FCALL.....	153
FILLOMR.....	154
FOREACH.....	156
FORMSHIFT .....	158
FROMLINE .....	159
FSHOW .....	165
GETDATE .....	166
GETFIELD .....	168
GETITEM.....	170
GOTOFRAME .....	172
ICALL.....	173
IF/ELSE/ELIF/ENDIF .....	176
IGNOREBT_off.....	178
IGNOREBT_on.....	179
ILAND.....	180
INDEXALIGN.....	181
INDEXBAT.....	182
INDEXCOLOR .....	183
INDEXFONT .....	185
INDEXKERN .....	187
INDEXLSP .....	189
INDEXOTL.....	190
INDEXPIF .....	191
INDEXRPE.....	193
INDEXSST .....	194
IPORT .....	196
IREVERSE_off.....	197
IREVERSE_on.....	198

JOG_on y JOG_off.....	199
LAND .....	200
LMSKIP.....	201
MAKEVMFILE .....	202
MAKEVMFORM .....	203
MAXICODE .....	204
MOVEH .....	209
MOVEHR .....	210
MOVETO.....	211
MSPP_on .....	212
MUL .....	213
NEWBACK.....	215
NEWFRAME .....	216
NEWFRONT.....	217
NEWGROUP .....	218
NEWPOS.....	219
NEWSIDE.....	220
NEWSTACK.....	221
NL .....	222
NMP_off.....	223
OMRINIT .....	224
OMRSHOW .....	229
ONEUP.....	230
ORIBL .....	231
ORITL.....	232
OVERPRINT_on.....	233
PAGEBRK .....	234
PAGERANGE .....	235
PDF417 .....	236
PDFBOUND.....	239
PDFDEST .....	240
PDFFORMOCG .....	241
PDFINFO .....	242
PDFOCG .....	243

PDFOPEN.....	244
PORT.....	245
PRECACHE .....	246
PROCESSDJE .....	247
QRCODE.....	249
QSTRIP_on.....	253
RELVAR.....	255
REPEAT.....	256
RESET .....	258
RPEDEF .....	259
RPEKEY.....	261
RSAVE.....	263
RUN.....	264
RUNDD .....	266
RUNPDF .....	270
RUNTIF .....	271
SAVEPP.....	273
SCALL .....	274
SETBAT.....	280
SETBFORM.....	283
SETBIDI .....	284
SETBUFSIZE.....	285
SETCJKENCMAP.....	286
SETCJKRULES .....	288
SETCOL .....	290
SETCOLWIDTH.....	294
SETCYCLECOPY.....	295
SETDATE.....	297
SETDBSEP .....	298
SETDLFILE .....	299
SETENCODING .....	300
SETEPATH .....	306
SETFINISHING .....	307
SETFONT .....	314

SETFORM.....	316
SETFPATH .....	319
SETFRAME .....	320
SETFTSP .....	321
SETFTSW .....	322
SETGEP.....	323
SETGRID.....	325
SETGUNIT .....	326
SETINDENT.....	327
SETIPATH.....	328
SETJDT .....	329
SETJPATH.....	331
SETKERN .....	332
SETLAYOUT .....	335
SETLFI.....	338
SETLKF .....	339
SETLMFILE.....	341
SETLSP .....	342
SETMARGIN.....	343
SETMAXBFORM.....	344
SETMAXCOPY .....	345
SETMAXFORM .....	346
SETMEDIA .....	347
SETMEDIAT .....	350
SETMPATH .....	351
SETMULTIUP .....	352
SETNMP .....	354
SETOBIN.....	355
SETOBINT .....	356
SETOTL.....	357
SETPAGEDEF .....	358
SETPAGENUMBER .....	359
SETPAGESIZE.....	361
SETPARAMS .....	362



SETPAT .....	363
SETPBRK.....	367
SETPCC .....	369
SETPCD .....	370
SETPIF .....	372
SETPPAT .....	376
SETPPATH .....	377
SETPROJECT.....	379
SETRCD .....	380
SETRES.....	382
SETRPE.....	383
SETRPEPREFIX .....	384
SETSKIP.....	386
SETTAB .....	388
SETTABS.....	389
SETTRAN .....	391
SETTPAT .....	392
SETTXB .....	394
SETTXC .....	395
SETTXS.....	398
SETUNIT .....	400
SETUTAB.....	401
SETV2HCONV.....	402
SETV2HTABLE .....	403
SETVAR .....	404
SETVFU .....	407
SETZEBRA.....	408
Comandos SHx .....	410
SHC y SHc.....	411
DESPLAZAMIENTO .....	412
SHIFTDATE .....	413
SHJ y SHj .....	414
SHL y SH.....	415
SHMF, SHMf y SHmf .....	416

SHP y SHp.....	419
SHPATH .....	425
SHPIT.....	427
SHR y SHr .....	429
SHROW .....	430
SHT y SHt.....	432
SHX .....	434
SKIPPAGE.....	436
SLIPSHEET .....	437
SOF_off.....	438
SORT.....	439
SPOOLNAME .....	440
STARTBOOKLET .....	441
STARTDBM .....	442
STARTLM.....	444
STARTOFSET .....	445
STARTXML .....	446
STOREVAR.....	447
SUB .....	449
TIFORI_off.....	451
TIFORI_on .....	452
TUMBLEDUPLEX_off.....	453
TUMBLEDUPLEX_on .....	454
TWOUP .....	455
UPDATE.....	456
USPS4CB .....	457
VPDISTPAT.....	460
XGF .....	461
XGFDEBUG .....	462
XGFEND .....	463
XGFENTRY .....	464
XGFRESDEF .....	465
XMLATL .....	467
XMLSYN .....	468

**ZSORT** ..... 469

Un comando de VIPP® es una secuencia independiente formada por un número cualquiera de operandos y la palabra clave del comando VIPP® descrita en la sintaxis. Los operandos, si los hay, siempre se colocan delante del comando. Los comandos están escritos en mayúsculas para evitar conflictos con los operadores de PostScript.

**++ y –**

Estos comandos se pueden aplicar a variables numéricas definidas con **SETVAR** para aumentar o disminuirlas en uno. Se pueden utilizar para implementar un contador y acciones asociadas en combinación con IF/ELSE/ENDIF.

**Sintaxis**

- /VARname ++
- /VARname –

**Donde**

**VARname** debe haberse inicializado previamente con **SETVAR**.

**Ejemplos**

Utilice este ejemplo para grapar cada documento de diez páginas configurado en un servidor de impresión FreeFlow o NPS.

```
/Staple /ON SETFINISHING /VAR.CNT1 0 SETVAR { /VAR.CNT1 ++ IF VAR.CNT1 10 eq { ENDOFSET
```



Nota: El tipo de variable puede ser un número entero o una cadena. Una variable de cadena también puede imprimirse o combinarse con el comando **VSUB**.

Los contadores que usan un valor de cadena pueden contener números de hasta 25 dígitos.

Se conservan los ceros iniciales presentes en la inicialización cuando se imprime la cadena.

Ejemplo:

```
/VAR1 (0000) SETVAR /VAR1 ++ VAR1 SH
```

imprimirá 0001

**Modos**

Este comando es aplicable en todos los modos.

**Comandos relacionados**

- **ADD**
- **SETVAR**
- **SUB**

## ABSPOS

Use **ABSPOS** en el modo de varias en 1 para colocar los comandos siguientes, como **MOVETO** o **MOVEH**, en una posición en función de la página física, en vez de la página lógica actual.

En general, este comando se utiliza para convertir los flujos de datos LCDS mediante las instrucciones DJDE GRAPHIC= en el modo de varias en 1.

El efecto de **ABSPOS** se cancela automáticamente al final de la página o a través de comandos de orientación como **PORT**, **LAND**, **IPOINT** e **ILAND**.

### Sintaxis

ABSPOS

### Ejemplos

Este ejemplo imprime la página 1 y la página 2 lado a lado, con una reducción a una sola página física. El archivo image1.tif se imprime en x0, y1000 (unidades actuales) desde el origen de la página física, y se superpone en las dos páginas lógicas.

```
TWOUP Page 1 data/commands PAGEBRK Page 2 data/commands ABSPOS 0 1000 MOVETO (image1.tif)
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **MOVETO**
- **MOVEH**

## ACCLOG

Use **ACCLOG** para capturar los datos de procesamiento de un trabajo VIPP® y, a continuación, registre los datos en el archivo de salida de Demographics, por ejemplo, .vpr y/o .vpd.

Este comando solo es eficaz si la función Demographics se ha activado para el trabajo; consulte la *Guía del usuario de FreeFlow VI Compose*.

Los datos capturados por **ACCLOG** pueden incluir, entre otros:

- Información estadística detallada que se acumula durante el uso por parte del trabajo de variables exclusivas. Coloque los comandos **ACCLOG** en **ENDJOB** para volcarlas al final del trabajo.
- Información detallada sobre los registros o las páginas que se han procesado, que se selecciona para realizar comprobaciones de integridad del trabajo. Coloque los comandos **ACCLOG** en la DBM o en **BEGINPAGE**.
- Información de contabilidad acumulada durante el trabajo para realizar recargas.

### Sintaxis

TagName Contents ACCLOG

TagName () ACCLOG

() () ACCLOG

### Donde:

(TagName) es una cadena o una variable que representa una etiqueta para identificar la información en contents (contenidos).

(Contents) es una cadena o una variable que contiene la información que se recopilará.

Si la cadena está vacía (segunda sintaxis), TagName representa una etiqueta para un conjunto de información proporcionada en los comandos **ACCLOG** siguientes. Los grupos pueden estar anidados en varios niveles.

Si ambas cadenas están vacías (tercera sintaxis) el grupo actual de información, abierto anteriormente por un comando mediante la segunda sintaxis, se cierra.



**Precaución:** No ejecute **ACCLOG** después de la inicialización de la página. Coloque el comando **ACCLOG** antes del primer comando de marcado, por ejemplo, después del comando **PAGEBRK**, o en un procedimiento **BEGINPAGE** o **/P ENDPAGE**.

### Ejemplos

La secuencia siguiente puede colocarse al principio de un DBM para capturar la información seleccionada para el registro actual:

```
(Customer) () ACCLOG
(FullName) ($$FIRST.$$NAME.) VSUB ACCLOG
(Address) ADDRESS VSUB ACCLOG
(City) CITY ACCLOG
() () ACCLOG
```

Los datos capturados por **ACCLOG** se almacenan en el archivo XML y/o en el archivo de base de datos, en

función de las opciones de Demographics.

En el archivo XML, .vppd, los datos se implementan como un árbol (cada grupo representa a un nodo) con el nombre de etiqueta raíz <ACCLLOG>.

Los datos del archivo de la base de datos se vuelcan de forma secuencial con el número de prefijo 4.

**Modos**

Este comando es aplicable en todos los modos.

**Comandos relacionados**

Ninguno

## ADD

**ADD** agrega un valor a una variable definida por **SETVAR** o a una variable de XML. Si la variable es un número, ya sea una cadena numérica, un número real o un número entero, el operando debe ser un número y **ADD** realiza la adición matemática tradicional. Si la variable es una matriz, el operando también debe ser una matriz, y **ADD** incluye los elementos del operando como nuevos elementos en la matriz de variables.

### Sintaxis

```
/VARname additem ADD
/^XMLname additem ADD
```

### Donde

#### /VARname

Hace referencia a una variable numérica previamente inicializada por **SETVAR**.

#### /^XMLname

Hace referencia a una variable de XML. Una variable XML no necesita inicializarse explícitamente. VIPP® inicializa todas las variables XML en una cadena vacía equivalente a una cadena numérica igual a cero.

### additem

Es una de las siguientes:

- Un número positivo o negativo que se agrega a la variable si esta es, en sí misma, un número. Puede ser un número entero o real o una cadena numérica. Cuando se trata de un gran número, es obligatorio contar con una cadena numérica.
- Una matriz que se unirá a la matriz de variables.

Las cadenas numéricas admiten números grandes de hasta 40 dígitos, 25 dígitos para la parte entera y 15 dígitos para la parte decimal. En una cadena numérica, el signo negativo y el delimitador decimal se definen mediante los parámetros /DecimalPoint y /NSign y se pueden producir en cualquier lugar de la cadena.

Es obligatorio configurar estos parámetros con los valores correspondientes para garantizar unos resultados precisos. Los valores predeterminados se definen en el archivo /usr/xgf/src/xgf.def. Se omiten los caracteres en la cadena numérica que no sean estos dos más los dígitos 0–9.

La longitud inicial de la cadena definida por **SETVAR** se amplía automáticamente hasta 40 dígitos cuando es necesario.

Los números reales y enteros deben utilizarse solo para los valores pequeños <= 99999, por ejemplo, para la implementación de un contador. El delimitador decimal, si lo hubiera, siempre es el punto (.). El signo negativo, si lo hubiera, siempre es el signo menos (-) y debe ser el primer carácter.

### Ejemplos

```
/VAR.CNT1 0 SETVAR
/VAR.CNT1 12 ADD
/VAR.CNT1 -3 ADD
/VAR_SUM (0) SETVAR
/VAR_SUM (1'234'890'566,00-) ADD
```



Este ejemplo muestra cómo usar **ADD** para capturar los datos en un trabajo de modo de línea, y cómo producir un gráfico en la página.

```
{ /VAR_CHARTDATA [ ] SETVAR
{ IF RPCOUNT 10 ge RPCOUNT 40 le and
{ /VAR_LINE RPCOUNT 0 100 GETFIELD %lines 10 to 40
/VAR_CHARTDATA
[ VAR_LINE 15 10 GETINTV %extract label
VAR_LINE 30 10 GETINTV %extract value
] ADD %accumulate in CHARTDATA
} ENDIF
} LNCOUNT REPEAT
x y MOVETO VAR_CHARTDATA 800 DRAWPIE %draw a pie
} ENDPAGE
```

Este ejemplo muestra cómo utilizar **ADD** para acumular datos en una matriz y producir un gráfico con estos. Cada línea se incluye en una parte distinta del recurso con formato (JDT/DBM/XJT), inicialización habitual, **BEGIN-PAGE**, **ENDPAGE**.

```
/VAR_CHART [ ] SETVAR %initialize array
...
/VAR_CHART [ VAR_LAB VAR_VAL ] ADD %accumulate label/value pairs
...
x y MOVETO VAR_CHART 800 DRAWPIE %draw chart
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **++/-**
- **SETVAR**
- **SUB**
- **MUL**
- **DIV**

## ADVPAGE

**ADVPAGE** activa diversas opciones relacionadas con las página en blanco y el procesamiento de líneas vacías.

### Sintaxis

`option ADVPAGE`

### Donde:

<b>opción</b>	es un número que se puede combinar estos valores:
<b>+1</b>	imprimir páginas que constan de una sola línea vacía
<b>+2</b>	imprimir páginas que constan de una sola línea NMP
<b>+4</b>	procesar líneas de vacío para pruebas de SETRCD

Las páginas que constan de una sola NMP o una sola línea vacía se omitirán de manera predefinida. No se aplican pruebas SETRCD de líneas vacías de manera predefinida.

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

**SETPBRK**

## AZTEC

**AZTEC** crea e imprime códigos de barras de AZTEC en función de los datos de cadena y los parámetros especificados. No hay fuentes especiales obligatorias.

### Sintaxis

```
(data) AZTEC
(data) scale rotate align AZTEC
(data) [width] rotate align AZTEC
(data) [ /ELevel xx ... ] AZTEC
(data) [ /ELevel xx ... ] scale rotate align AZTEC
(data [ /ELevel xx ... ] [width] rotate align AZTEC
```

### Donde:

<b>(datos)</b>	es una cadena que contiene la información que se desea almacenar en el símbolo.
<b>escala</b>	es el valor de escala (el valor prefijado es 1).
<b>girar</b>	es el valor de giro (el valor prefijado es 0).
<b>alinear</b>	es uno de los códigos de alineación que indica el punto del código de barras que se alineará a la posición de impresión secundaria:
	<b>0</b> superior izquierda (predeterminado)
	<b>1</b> superior derecha
	<b>2</b> centro superior
	<b>10</b> inferior izquierda
	<b>11</b> inferior derecha
	<b>12</b> centro inferior
	<b>20</b> centro izquierda
	<b>21</b> centro derecha
	<b>22</b> centro centro

La matriz opcional contiene una lista de pares de teclas/valores que se pueden usar para especificar las siguientes opciones adicionales:

<b>/ELevel entero</b>	es el nivel de corrección de errores entre 0 y 99. El valor prefijado es 23.
<b>/MinDim entero</b>	es la dimensión mínima de los símbolos.

Otras opciones se pueden agregar en futuras versiones.

Las funciones avanzadas, como ECI y la anexión de estructuras, es posible que se admitan en futuras versiones.

### Ejemplos

(data) [ /ELevel 40 /MinDim 5 ] AZTEC

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- PDF417
- DATAMATRIX
- MAXICODE
- QRCODE
- MOVEH
- MOVEHR
- MOVETO

## BACKSP\_off

**BACKSP\_off** desactiva el procesamiento del retroceso.

### Sintaxis

`BACKSP_off`

### Modos

Este comando es aplicable en el modo de línea solamente sin RPE.

### Comandos relacionados

`BACKSP_on`

## BACKSPF\_on

**BACKSPF\_on** permite el procesamiento del retroceso solo con fuentes fijas.

### Sintaxis

BACKSPF\_on

### Modos

Este comando es aplicable en el modo de línea solamente sin RPE.

### Comandos relacionados

BACKSP\_off

## BACKSPP\_on

**BACKSPP\_on** permite el procesamiento del retroceso con fuentes fijas y proporcionales.

### Sintaxis

BACKSPP\_on

### Modos

Este comando es aplicable en el modo de línea solamente sin RPE.

### Comandos relacionados

BACKSP\_off, BACKSPF\_on

## BCALL

**BCALL** ejecuta un fragmento del código VIPP® encapsulado en un procedimiento. **BCALL** protege las opciones en el procedimiento de efectos externos. Las opciones en vigor antes de la instrucción **BCALL** permanecerán activas después de la instrucción.

Entre las opciones protegidas se incluyen:

- Fuente definida por **SETFONT** o a instancias de un índice de fuente
- Color definido por **SETTXC** o instancias de un índice de color
- Fondo definido por **SETTXB** o a instancias de un índice **BAT**
- Espaciado de línea definido por **SETLSP**
- Sangría definida por **SETINDENT**
- Esquema definido por **SETOTL** o a instancias de un índice **OTL**

### Sintaxis

```
{ VIPP code } BCALL
```

**Donde:**

**{ VIPP code }** es un fragmento de código VIPP® que pinta un elemento en la página.

### Ejemplos

```
{ /NHE 12 SETFONT RED SETTXC /UNDL SETTXB (text text text ... ) 3 SHP ] BCALL
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**FCALL**, **MOVETO**, **SCALL**



## BEGINARBM

**BEGINARBM** inicia una definición de combinación bidireccional que el comando **SETBIDI** seleccionará y será usada por la función de transformación BIDI.

La definición debe terminar con un comando **ENDARBM**.

### Sintaxis

```
/mrg_name BEGINARBM (char31) (char21) (char11) (charR1) ..... (char3N) (char2N) (char1N)
```

### Donde:

<b>/mrg_name</b>	es el nombre de la tabla que será usada por <b>SETBIDI</b>
<b>(char3X)</b>	es una cadena que representa el tercer carácter de un triplete que se combinará, o 0 si se trata de un par
<b>(char2X)</b>	es una cadena que representa el segundo carácter de un par o un triplete que se combinará
<b>(char1X)</b>	es una cadena que representa el primer carácter de un par o un triplete que se combinará
<b>(charRX)</b>	es una cadena que representa el carácter que se sustituirá en el par o el triplete



Nota: Se proporcionan tablas predefinidas para Windows1256 para árabe, Windows1255 para hebreo y UTF8 en el archivo de configuración bidireccional en `xgf/src/arb.def`.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **BIDIRECCIONAL**
- **SETBIDI**
- **BEGINARBT**
- **ENDARBT**

## BEGINARBT

**BEGINARBT** inicia una definición de contexto bidireccional que el comando **SETBIDI** seleccionará y será usada por la función de transformación BIDI.

La definición debe terminar con un comando **ENDARBT**.

### Sintaxis

```
/ctx_name BEGINARBT (input1) (begin1) (middle1) (end1) (isolated1) join_group1 .... (in
```

Donde:

<b>/ctx_name</b>	es el nombre de la tabla que será usada por <b>SETBIDI</b> .
<b>(inputX)</b>	es una cadena que representa un carácter en la cadena de datos de entrada procesada mediante la función de transformación BIDI.
<b>(beginX)</b>	es una cadena que se sustituirá si el carácter está al comienzo de una palabra. Para una cadena de entrada que representa un dígito (0-9), esta cadena define el dígito hindi alternativo.
<b>(middleX)</b>	es una cadena que se sustituirá si el carácter está en el centro de una palabra.
<b>(endX)</b>	es una cadena que se sustituirá si el carácter está al final de una palabra.
<b>(isolatedX)</b>	es una cadena que se sustituirá si el carácter está aislado.
<b>join_groupX</b>	es el grupo de unión al que este carácter pertenece. Una de: <ol style="list-style-type: none"> <li>1 Carácter derecha a izquierda aislado</li> <li>2 Carácter árabe conectado por la derecha</li> <li>3 Carácter árabe conectado por la derecha y la izquierda</li> <li>4 Carácter árabe conectado a la izquierda</li> <li>5 Carácter de izquierda a derecha</li> <li>6 caracteres especiales, derecha a izquierda aislado, o izquierda a derecha, según los caracteres adyacentes.</li> </ol>

Si un carácter no se ha definido en la tabla, no se sustituye y se asigna a un grupo de unión de 5.

Cada cadena sustituida está vinculada a una asignación de fuentes determinada. Por lo tanto, la tabla seleccionada por **SETBIDI** debe coincidir con la fuente seleccionada por **SETFONT** o **INDEXFONT**.



Nota: Se proporcionan tablas predefinidas para Windows1256 (árabe), Windows1255 (hebreo) y UTF8 en el archivo de configuración bidireccional en `xgf/src/arb.def`.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- BIDIRECCIONAL
- SETBIDI
- FCALL
- MOVETO
- SCALL
- ENDARBM

## BEGINFRAME

**BEGINFRAME** define acciones que debe realizarse al comienzo de cada trama en una página. Estas acciones son nulas de manera prefijada. **BEGINFRAME** permite comandos de marca como acciones, aunque normalmente se usa para configurar variables y lógicas condicionales para ejecutar determinadas acciones antes de representar la trama siguiente.

### Sintaxis

```
{ "start of frame" actions } BEGINFRAME
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**FRCOUNT**, **NEWFRAME**, **SETLKF**

## BEGINIMP

**BEGINIMP** activa la función de imposición. Use esta función para imprimir un documento que consta de una recopilación de imágenes, segmentos y/o archivos EPS para que las páginas impresas creen un folleto. Use este comando con una configuración varias en 1 como TWOUP y el modo a 2 caras como **DUPLEX\_on** o **TUMBLE-DUPLEX\_on**. Esto garantiza que el documento impreso se podrá plegar y grapar correctamente. **BEGINIMP** finaliza mediante **ENDIMP**.



Nota: No asocie este comando con funciones de imposición en el controlador de la impresora.

### Sintaxis

BEGINIMP

llama a los elementos de página utilizando ICALL o SCALL

ENDIMP

Utilice esta información para activar la impresión de la salida impuesta en el orden inverso:

- |                               |   |
|-------------------------------|---|
| <b>[/IMPmode 0] SETPARAMS</b> | el modo actual y el modo predeterminado producen páginas en orden 1 a N. El orden de las páginas es [última primera] [segunda última-1]... [centro + 1 centro]. |
| <b>[/IMPmode 1] SETPARAMS</b> | produce páginas en orden N a 1. El orden de las páginas es [centro + 1 centro] ... [segundo último -1] [último primero].  |

### Ejemplos

```
TWOUP
TUMBLE-DUPLEX_on
BEGINIMP
(report1.ps) RUNDD
ENDIMP
```

### Modos

Este comando es aplicable en todos los modos excepto el modo de línea.

### Comandos relacionados

- **ENDIMP**
- **DUPLEX\_off**
- **DUPLEX\_on**
- **ICALL**
- **PAGEBRK**
- **RUNDD**
- **RUNTIF**
- **RUNPDF**
- **TUMBLE-DUPLEX\_off**

- TUMBLEDUPLEX\_on
- TWOUP
- TUMBLEDUPLEX\_on
- SCALL

## BEGINPAGE

**BEGINPAGE** define acciones que deben realizarse al principio de cada página. Estas acciones son nulas de manera prefijada. No use **BEGINPAGE** para realizar comandos de marca, úselo para realizar pruebas en una página antes de imprimirla. Consulte **ENDPAGE** si desea obtener más información acerca de la marcación de páginas. Esto permite comprobar un valor y realizar una acción como, por ejemplo, la configuración de un nuevo archivo JDT.



Nota: Coloque el procedimiento **BEGINPAGE** antes de realizar cualquier comando de marcado, incluido el comando **MOVETO** en la página actual. Si coloca el comando **BEGINPAGE** después de un comando de marcado, el comando **BEGINPAGE** no se ejecuta para la página actual y se rechaza mediante **PAGEBRK** al final de la página.

### Sintaxis

- { start of page actions } **BEGINPAGE**
- { start of page actions } /M **BEGINPAGE**
- { start of page actions } /B **BEGINPAGE**
- { start of page actions } /Z **BEGINPAGE**

#### { start of page actions } **BEGINPAGE**

Define una acción en el nivel JDT secundario o básico.

#### { start of page actions } /M **BEGINPAGE**

Define una acción en el nivel JDT principal.

#### { start of page actions } /B **BEGINPAGE**

Define una acción en el nivel JDT emergente.

#### { start of page actions } /Z **BEGINPAGE**

Define un procedimiento que se ejecuta en el momento de la imposición en lugar de al realizar la composición.

Utilice el modificador /B al definir las acciones de **BEGINPAGE** en las páginas de pancarta de JDT. Utilice el modificador /M al definir las acciones de **BEGINPAGE** en el JDT principal. No se requiere ningún modificador para definir las acciones de **BEGINPAGE** en el JDT secundario.

El modificador /Z solo es relevante si **ZSORT** genérico está activo, y se usa para evitar la realización de la composición y la imposición de forma simultánea. Se usa principalmente en lógicas condicionales con la variable **PINDEX** dado que, si **ZSORT** genérico está activo, **LPINDEX** es siempre igual a 1 en el momento de la composición.

## Ejemplos

Este ejemplo muestra cómo grapar conjuntos de varias copias de las páginas si se utiliza el modo sin clasificación. Esto evita tener que colocar un comando ENDOFRUN al principio de cada página en los datos.

- /Staple /ON SETFINISHING
- 3 SETCYCLECOPY
- COLLATE\_off
- { ENDOFRUN } BEGINPAGE

Este ejemplo muestra cómo capturar el campo de datos después de **FORM=** en una portada, y usarlo para activar un JDT en las páginas siguientes.

```
/BANNER 10 9 0 5 /eq (FORM=) SETPCD {IF BANNER {/VARjdt 0 5 8 /BANNER GETFIELD ($$VARjdt
```

Para comprobar una página de portada de **START OF JOB** o **END OF JOB**, use un procedimiento **BEGINPAGE**. Este procedimiento buscará el texto **START OF JOB** o **END OF JOB** y, al encontrarlo, configura un formulario nulo (sin formulario).

```
/StartBanner 3 3 5 13 /eq (START OF JOB:) SETPCD /EndBanner 3 3 5 11 /eq (END OF JOB:
```

Utilice el operador “or” para definir la condición como verdadera si **StartBanner** o **EndBanner** es verdadero.

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

- ENDOFRUN
- ENDPAGE
- GETFIELD
- SLIPSHEET



## BEGINPCC

**BEGINPCC** inicia la definición de una tabla PCC que se usa más adelante con el comando **SETPCC**.

### Sintaxis

```
/Staple /ON SETFINISHING /pccname BEGINPCC value [ pre_skip print_action post_skip ] ..
```

### Donde:

<b>valor</b>	es el código PCC (0–255 o 16 #0–16 #FF).
<b>pre_skip</b>	es una de las siguientes: <ul style="list-style-type: none"> <li>• un número de líneas</li> <li>• una tecla que <b>SETVfu</b> asignará más adelante.</li> </ul>
<b>print_action</b>	es una de las siguientes: <ul style="list-style-type: none"> <li>• verdadero (para imprimir el registro)</li> <li>• falso (no se imprime el registro).</li> </ul>
<b>post_skip</b>	es una de las siguientes: <ul style="list-style-type: none"> <li>• un número de líneas</li> <li>• una tecla que <b>SETVfu</b> asignará más adelante.</li> </ul>

### Ejemplos

```
/ANSI BEGINPCC
16#20 [ 1 true 0 ] % (first entry is the default)
16#30 [ 2 true 0 ] % asa blank : space 1 line and print
16#20 [ 3 true 0 ] % asa 0 : space 2 line and print
16#20 [ 3 true 0 ] % asa - : space 3 line and print
16#2B [ 0 true 0 ] % asa + : print without spacing
16#31 [ /Sk1 true 0 ] % asa 1 : skip to channel 1 and print
....
ENDPCC
```

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

**ENDPCC**, **SETPCC**, **SETVfu**

## BEGINRPE

**BEGINRPE** inicia una definición de biblioteca RPE en un archivo JDT. Consulte [Información de comando RPE](#) y otros comandos de **RPE** relacionados para obtener más información.

### Sintaxis

```
Maxrpe BEGINRPE
```

### Donde:

**Maxrpe** debe ser igual o superior al número de los comandos **FROMLINE** o **RPEKEY** después de **BEGINRPE** y hasta **ENDRPE**.



Nota: En teoría, el número máximo de entradas en una biblioteca RPE o en una entrada RPE es 65 535. Sin embargo, un número grande puede afectar al rendimiento o causar errores de desbordamiento de la pila.

### Definir bibliotecas RPE en un JDT maestro

para obtener más información sobre la definición de varias bibliotecas RPE en un JDT maestro, consulte [INDEXRPE](#).

### Cancelar un RPE activo

Para cancelar un RPE activo al alternar entre archivos JDT, utilice la siguiente sintaxis sin **ENDRPE**:

```
0 BEGINRPE
```

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

- [ENDRPE](#)
- [FROMLINE](#)
- [INDEXRPE](#)
- [RPEKEY](#)
- [STARTLM](#)

## BEGINTABLE

Utilice **BEGINTABLE** para iniciar una tabla.xx

### Sintaxis

```
[ /param1 value1 ... /paramX ValueX ] BEGINTABLE
```

### Donde:

<b>/param</b>	puede ser uno de los siguientes:
<b>/Margins</b>	[superior inferior izquierda derecha] márgenes de celda predeterminados en unidades actuales
<b>/TableStroke</b>	GEPkey trazo del borde de la tabla
<b>/Width</b>	anchura predeterminada de las celdas en unidades actuales
<b>/Height</b>	altura mínima predeterminada de las celdas en unidades actuales
<b>/MaxHeight</b>	altura máxima predeterminada de las celdas en unidades actuales
<b>/CellFill</b>	color predeterminado para rellenar las celdas
<b>/CellStroke</b>	Gepkeys predeterminadas para trazar los bordes de las celdas
<b>/TextAtt</b>	código VIPP® predeterminado para configurar los atributos de texto
<b>/Align</b>	atributo de alineación predeterminado igual que <b>SHP</b>

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**ENDTABLE,SHROW**

## BEGINXPD

**BEGINXPD** inicia una tabla de definición de procesamiento de XML (XPD) en un archivo de ficha de trabajo de XML (XJT). Se debe combinar con un comando **ENDXPD**.

Entre las definiciones **BEGINXPD** y **ENDXPD** solo se admiten definiciones de acciones o sustituciones de etiquetas mediante comandos **BTA/ ETA** y **BTS/ETS**.

### Sintaxis

BEGINXPD

En las definiciones de BTA y BTS, están disponibles estas variables integradas:

<b>XMLATL</b>	lista de todos los atributos de una etiqueta XML
<b>XMLATN</b>	nombre del atributo
<b>XMLATV</b>	valor del atributo
<b>XMLTAG</b>	nombre del nodo actual
<b>XMLVAL</b>	contenidos del nodo actual
<b>XMLPAR</b>	nombre del nodo primario del nodo actual
<b>XMLPATH</b>	VXVpath del nodo actual
<b>XMLDTH</b>	profundidad del nodo actual

### Modos

Este comando es aplicable en el modo XML.

### Comandos relacionados

- **ENDXPD**
- **BTA**
- **BTS**
- **ETA**
- **ETS**
- **STARTXML**

## BOOKLETRANGE

**BOOKLETRANGE** es similar a **PAGERANGE** pero con orientación de folleto. **BOOKLETRANGE** funciona en combinación con **STARTBOOKLET/ENDBOOKLET** para permitir la selección de una serie de folletos que desea imprimir.

El comando **BOOKLETRANGE** debe colocarse en el trabajo VIPP®. Se recomienda agregar el comando al principio del archivo de envío.

### Sintaxis

```
startbooklet endbooklet BOOKLETRANGE
```

### Donde

**startbooklet** es el número del primer folleto que se va a imprimir.

**endbooklet** es el número del último folleto que se va a imprimir.

En función del parámetro /PageRange configurado por **SETPARAMS**, el trabajo se completará con normalidad o finalizará con el mensaje Folletos seleccionados: startbooklet endbooklet.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**STARTBOOKLET, ENDBOOKLET**


## BOOKMARK


**BOOKMARK** crea un marcador interactivo en un archivo PDF.

### Sintaxis

<b>input_string</b>	<p>es la cadena que se utiliza para crear el marcador. Por lo general es un campo en el modo de base de datos o datos extraídos mediante <b>GETFIELD</b> en modo de línea.</p> <p>Los datos de un solo byte deben cifrarse mediante ISO-8859-1. Los datos de varios bytes deben cifrarse mediante UTF8. VI Compose lo convertirá automáticamente en UTF16 para insertarlo en el PDF porque esta es la única codificación de varios bytes admitida por el formato PDF. Para desencadenar la conversión de datos UTF8 a UTF16, la fuente actual (seleccionada por SETFONT o INDEXFONT) debe tener una codificación UTF8.</p>
<b>recuento de</b>	<p>es el número de marcadores secundarios después de este marcador. (el valor predeterminado es 0). Puede ser una variable. Si es positivo, el marcador se abre al acceder inicialmente al archivo. Si es negativo, está cerrado.</p>
<b>color</b>	<p>es un Colorkey que define el color utilizado para mostrar el texto del marcador. (El valor predeterminado es negro). Se admiten solo RGB sin formato y color de escala de grises</p>
<b>estilo</b>	<p>es el estilo del texto marcador:</p> <p><b>0</b> normal (valor predeterminado)</p> <p><b>1</b> cursiva</p> <p><b>2</b> negrita</p> <p><b>3</b> negrita y cursiva</p>
<b>/EX</b>	<p>indica <b>BOOKMARK</b> que crea un marcador extendido en lugar de un marcador normal. Un marcador extendido es una nota PDF transparente que normalmente solo se visualiza desde la sección de comentarios del PDF completado. /EX es válido solo si se utiliza VI eCompose (VIeC).</p> <p>Los marcadores ampliados admiten grandes bloques de texto si se utiliza la función de envío de VIeC. (El contenido de un marcador regular está limitado a 256 caracteres. Los marcadores ampliados pueden contener hasta 64 KB). VIeC procesa los contenidos de los marcadores ampliados (en función de la opción 2) a través de su unión a los contenidos del marcador habitual. Un marcador ampliado siempre está vinculado a un marcador habitual y debe crearse después del mismo. Varios marcadores extendidos pueden crearse en un documento secundario (entre dos marcadores regulares consecutivos). Consulte la <i>Guía del usuario de FreeFlow VI eCompose</i> para obtener información detallada de VIeC y del separador de VIeC.</p>
<b>opt1</b>	<p>es un número entero que indica al separador VIeC cómo gestionar el archivo PDF secundario:</p> <p><b>0</b> No produce el archivo PDF secundario</p>

- 1** Genera el archivo PDF secundario pero no incluye el marcador en este.
- 2** Genera el archivo PDF secundario e incluyen el marcador en este. Esta es la opción prefijada.

 Nota: Si no hay ningún marcador ampliado, el separador VIEC funcionará como se describe si opt1 = 2.

 Nota: Todos los marcadores ampliados en un documento secundario requieren el mismo valor de opt1.


**opt2** es un número entero que indica al separador VIEC cómo identificar el identificador ampliado.

**0** No hacer nada con este marcador ampliado

**1** Combinar el marcador ampliado con el marcador normal asociado (para el procesamiento de archivos de índice), pero no incluirlo en el PDF secundario

**2** Combinar el marcador ampliado con el marcador normal asociado (para el procesamiento de archivos de índice) e incluirlo en el PDF secundario

 Nota: Para insertar comentarios o instrucciones en el archivo PDF principal, use opt2=0.

 Nota: La combinación opt1=0 y opt2=2 no es relevante. El software actúa como opt2=1.

## Ejemplos

Este ejemplo crea un marcador ampliado que hace que el separador VIEC produzca un PDF secundario que sin el marcador ampliado. Sin embargo, el marcado ampliado se incluirá en el archivo de índice de VIEC y, con un envío de VIEC apropiado y la configuración del servidor de correo electrónico, generará un correo electrónico a John Smith con el archivo PDF secundario adjuntado.

```
(John Smith@isp.com:April invoice:Dear John,\nAttached you will find your
invoice for April.\nBest regards,\nPaul Martin) [ /EX 2 1 ] BOOKMARK
```

## Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- GETFIELD
- PDFDEST
- SETPIF
- INDEXPIF
- PDFINFO
- PDFOPEN

## BSTRIP\_off

**BSTRIP\_off** desactiva la eliminación de espacios iniciales y finales en blanco en campos delimitados en el modo de base de datos.

### Sintaxis

**BSTRIP\_off**

**BSTRIP\_off** y otros comandos globales como **DUPLEX\_on**, **SETDBSEP** y **SETBUFSIZE** no deben codificarse en el archivo maestro de la base de datos. Deberán codificarse al principio del archivo de base de datos antes del comando **STARTDBM**, o en una ficha del descriptor de trabajos externo referenciada por un comando **SETJDT** colocado antes del comando **STARTDBM** en el archivo de base de datos.

### Ejemplos

```
%!
DUPLEX_on
(;) SETDBSEP
BSTRIP_off
(cas.dbm) STARTDBM
....

%! database file
(cas.jdt) SETJDT
(cas.dbm) STARTDBM
...

%!PS-Adobe-2.0
%%Title: cas.jdt
%%Creator: CAS/RXCH
....
DUPLEX_on
(;) SETDBSEP
BSTRIP_off
....
```

### Modos

Este comando es aplicable en el modo de base datos.

### Comandos relacionados

**BSTRIP,GETINTV,STARTDBM**



## BTA

**BTA** inicia una definición de acción de etiqueta. Se debe combinar con un comando **ETA**.

### Sintaxis

```
BTA /VXVkey
{ acción de inicio de etiqueta }
{ acción de final de etiqueta }
{ acción de etiqueta de contenidos parciales }
ETA
```

### Donde:

<b>/VXVkey</b>	es un VXVname que hace referencia a un nodo del archivo XML. Dado que VXVkey hace referencia a un nombre de nodo XML (en vez de a un VXVname que hace referencia a los contenidos del nodo) el “^” en la primera posición debe omitirse. A diferencia de las referencias independientes VXVname, los operandos VXVname BTA pueden ser ambiguos. Si esto ocurre, la definición se aplicará a todos los nodos cuyo VXVpath coincide con BTA VXVname.
<b>{ start tag action }</b>	es una secuencia de comandos VIPP® que se ejecuta cada vez que se encuentra la etiqueta de inicio del nodo.
<b>{ end tag action }</b>	es una secuencia de comandos VIPP® que se ejecuta cada vez que se encuentra la etiqueta de final del nodo.
<b>{ partial contents tag action }</b>	es una secuencia de comandos de VIPP® que se ejecuta cada vez que se detecta una etiqueta de inicio mientras algunos contenidos del nodo de su elemento primario están pendientes.

### Ejemplos

En este ejemplo, los contenidos del nodo parcial `is an y writer born in` desencadenarán la ejecución del proceso. Este operando es opcional. Si no se especifica, el operando `{ end tag action }` se utiliza en su lugar.

```
<author>
<name>John Smith</name> is an <nat>English</nat>writer born in <birth>1961</
birth>.
</author>
```

Es un ejemplo de BTA utilizado con los datos XML que se muestran en el ejemplo anterior.

```
BTA /author
{ }
{ x y MOVETO ^author 0 SHP }
ETA
```

### Modos

Este comando es aplicable en el modo XML.

### Comandos relacionados

- BEGINXPD
- ENDXPD
- BTS
- ETA
- ETS
- STARTXML

## BTS

**BTS** inicia una definición de sustitución de etiqueta. Se debe combinar con un comando **ETS**.

### Sintaxis

```
BTS /VXVname
(start tag substitution string) | { start tag substitution action }
(end tag substitution string) |
{ end tag substitution action }
ETS:
```

### Donde

<b>/VXVname</b>	es un VXVname que hace referencia a un nodo del archivo XML. Dado que VXVkey hace referencia a un nombre de nodo XML (en vez de a un VXVname que hace referencia a los contenidos del nodo) el ^ en la primera posición debe omitirse. A diferencia de las referencias a VXVname independientes, los operandos VXVname BTS pueden ser ambiguos. Si esto ocurre, la definición se aplicará a todos los nodos cuyo VXVpath coincide con BTS VXVname.
<b>(iniciar la cadena de sustitución de etiqueta )</b>	define una cadena que se sustituirá con el nombre de la etiqueta inicial, lo que la hará parte de los contenidos del nodo de su elemento primario.
<b>{ start tag substitution action }</b>	es una secuencia de comandos VIPP® que se ejecuta cada vez que se encuentra la etiqueta de inicio del nodo. define una cadena que se sustituirá con el nombre de la etiqueta inicial, lo que la hará parte de los contenidos del nodo de su elemento primario.
<b>(iniciar la cadena de sustitución de etiqueta)</b>	define una cadena que se sustituirá con el nombre de la etiqueta final, lo que la hará parte de los contenidos del nodo de su elemento primario.
<b>{ end tag substitution action }</b>	es una secuencia de comandos VIPP® que se ejecuta cada vez que se encuentra la etiqueta de final del nodo. Esta secuencia debe proporcionar una cadena que se sustituirá con el nombre de la etiqueta final, lo que la hará parte de los contenidos del nodo de su elemento primario.



Nota: Las sustituciones de **BTS** tienen prioridad sobre las acciones **BTA**. Si un nodo coincide con una entrada **BTS**, ninguna acción **BTA** coincidente se pasa por alto.



Consejo: Para sustituir las etiquetas con la fuente, el color, el fondo y los índices **SST** o **PIF**, utilice las definiciones **BTS** y **ETS**.

### Ejemplos

Se asumen los fragmentos XML siguientes:

```
<author>
<name>John Smith</name> is an English writer born in 1961.
</author>
```

Esta codificación imprimirá John Smith is an English writer born in 1961.

```
/U /UNDL INDEXBAT  
/u null INDEXBAT  
BEGINXPD  
BTS /name (//U) (//u) ETS  
BTA /author {} { ^author 0 SHP } ETA  
ENDXPD
```

### Modos

Este comando es aplicable en el modo XML.

### Comandos relacionados

- BEGINXPD
- ENDXPD
- BTA
- ETA
- ETS
- STARTXML

## CACHE

**CACHE** permite el almacenamiento en caché de los recursos. **CACHE** convierte un recurso en un diccionario de formularios PostScript que pueden ser usados por los comandos **SCALL**, **SETFORM** o **SETBFORM**. **CACHE** siempre se combina con uno de estos comandos mediante la sustitución de (segmentname) o (formname) por (rname) **CACHE**, como se describe más abajo.

Para recursos PDF, **CACHE** utiliza **CropBox** o **MediaBox** para determinar el cuadro de límite de la imagen. **Trimbox** (se usa en vez de CropBox hasta Vi Compose 12.0) ya no se usa.

### Sintaxis

```
(rname) CACHE ... SCALL (rname) CACHE ... SETFORM (rname) CACHE ... SETBFORM [(rname) C
```

Para obtener descripciones adicionales de la sintaxis, consulte [SETFORM](#), [SETBFORM](#) y [SCALL](#).

### Donde:

**rname** puede representar un segmento VIPP®, un archivo EPS, un archivo PostScript, un archivo JPEG o un TIFF.

Si se usa un segmento VIPP® con **SETFORM/SETBFORM**, su origen figura en la esquina inferior izquierda de la página. Para especificar otro origen, utilice **CACHE/SCALL** en una definición de formulario en línea como en este ejemplo.

```
{ 50 80 MOVETO (logo1.seg) CACHE SCALL } SETFORM
```

Los formularios VIPP® encapsulados entre llaves { } no están permitidos como operandos de **CACHE**. Si se utiliza **FSHOW** en dichos formularios, implícitamente se aplica **CACHE**.

**CACHE** puede utilizar archivos ubicados en estos directorios de recursos:

- formlib (definido por SETFPATH)
- imglib (definido por SETIPATH)
- mislib (definido por SETMPATH)

### Ejemplos

```
(car1.eps) CACHE .6 SCALL (car1.eps) CACHE [600 300] 0 22 SCALL %for fit-in-box option
```



Nota: El comando **CACHE** tiene una implementación de PostScript compatible con dispositivos que usan el almacenamiento en caché de formularios de Adobe Red Book PostScript de nivel 2. Con este método de caché, los recursos son cargados y recuperados de la memoria caché mediante el formato de rasterización del dispositivo.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- [SCALL](#)
- [SETFORM](#)

- SETBFORM
- SETFPATH
- SETIPATH
- SETMPATH

## CASE

**CASE** es una alternativa a IF/ELSE/ENDIF en varias pruebas simultáneas similares. Utilice siempre **ENDCASE** para cerrar la lista CASE.

### Sintaxis

```
CASE reference_variable { default action }
  choice1 { action 1 }
  choice2 { action 2 }
  ...
  choicen { action n }
ENDCASE

CASE reference_variable { default action }
  [ choice1a choice1b ... choice1x ] { action 1 }
  choice2 { action 2 }
  ...
  choiceN { action N }
ENDCASE
```

### Donde:

<b>reference_variable</b>	es una variable que representa una cadena o un número.
<b>acción prefijada</b>	es una secuencia de comandos de modo nativo que se ejecuta si la referencia _variable no es igual a los elementos choicen.
<b>choicen</b>	es una cadena o un número, en comparación con reference_variable.
<b>action n</b>	es una secuencia de comandos de modo nativo que se ejecutan si choicen es igual a reference_variable. Puede incluir instrucciones IF/ELSE adicionales.
<b>choice1a choice1b ... choice1x</b>	es una lista de posibles opciones asociadas a una sola acción.

### Ejemplos

En este ejemplo, el saludo utilizado varía según el valor asignado a VAR.SCORE. Si VAR.SCORE no es 1, 2 o 3, se utiliza el valor predeterminado Dear Sirs.

```
CASE VAR.SCORE { (Dear Sirs,) SHL}
(1) { (Dear Mr. $$NAME.,) VSUB SHL}
(2) { (Dear Mrs. $$NAME.,) VSUB SHL}
(3) { (Dear Miss $$NAME.,) VSUB SHL}
ENDCASE
```

La sintaxis puede ser ampliada para incluir instrucciones IF/ELSE en la instrucción CASE.

```
CASE VARbranch { null SETFORM }          % branch unknown
(7481) { IF STATE (IL) eq
      { (Illinoisregion1) SETFORM }
      ELSE
      { (region1) SETFORM } ENDIF
(7483) { (region1) SETFORM }
(7496) { (region2) SETFORM ]
ENDCASE
```

### Comandos relacionados

ENDCASE,IF/ELSE/ELIF/ENDIF



## CHKPOINT

**CHKPOINT** define el final de un punto del ciclo de copia en modo de clasificación. De forma prefijado, el final del punto es el final del archivo en el modo de línea. No hay ningún final prefijado del punto del juego en modo nativo. Use **%%EOF** al final de cada conjunto de datos o archivo de base de datos para finalizar el archivo.

### Sintaxis

CHKPOINT

### Ejemplos

Este es un ejemplo de un conjunto de datos que usa **CHKPOINT** en modo nativo en un flujo de datos.

```
( ) STARTLM Page 1 Page 2 Page 3 %%XGF CHKPOINT PAGE 4 PAGE 5 PAGE 6 %%EOF
```

Use **CHKPOINT** en un comando **SETPCD** si hay alguna parte de los datos que se puede usar como desencadenante entre los conjuntos.

```
/SetStart 1 1 0 6 /eq (Page 1) SETPCD { IF SetStart { CHKPOINT } ENDIF } BEGINPAGE ( ) S
```

Use **CHKPOINT** con un contador si hay un número de páginas conocido y constante en cada conjunto (en este caso, tres).

```
/VARPageCount 1 SETVAR { IF VARPageCount 3 eq { CHKPOINT /VARPageCount 1 SETVAR % Reset
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**COLLATE\_on**, **SETCYCLECOPY**

## COLLATE\_dbm

**COLLATE\_dbm** activa un modo de clasificación en modo de base de datos. Si se activa el modo de clasificación, se realiza la llamada al archivo maestro de la base de datos para cada registro el número de veces especificado por SETCYCLECOPY.

### Sintaxis

COLLATE\_dbm

### Ejemplos

Este ejemplo imprime dos copias de un trabajo en modo de base de datos mediante **COLLATE\_dbm** y **SETCYCLECOPY**.

```
% DBM code

COLLATE_dbm
2 SETCYCLECOPY

/NTMR 14 SETFONT
300 2700 MOVETO
($$Fname. $$LName.) VSUB 0 SHP
% other code
PAGEBRK
```



Nota: Todos los comandos de clasificación antes del comando **SETCYCLECOPY** deben estar codificados.

### Modos

Este comando es aplicable en el modo de base datos.

### Comandos relacionados

SETCYCLECOPY

## COLLATE\_off

**COLLATE\_off** establece el modo **sin clasificación** del modo del ciclo de copia. **COLLATE\_on** es el valor prefijado. El modo **sin clasificación** indica que el número solicitado de copias se produce inmediatamente después de cada página lógica.

### Sintaxis

`COLLATE_off`



Nota: Todos los comandos de clasificación antes del comando **SETCYCLECOPY** deben estar codificados.



Consejo: Use este comando con **TWOUP** para producir dos copias reducidas de un informe en una hoja de tamaño normal en una pasada.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**COLLATE\_on**, **COLLATE\_dbm**, **SETCYCLECOPY**

## COLLATE\_on

**COLLATE\_on** configura el modo de clasificación del modo del ciclo de copia. Esta es la opción prefijada. En el modo de clasificación, se crean varias copias según el trabajo o juego correspondientes. Para obtener más información, consulte **CHKPOINT**.

### Sintaxis

COLLATE\_on



Nota: Todos los comandos de clasificación antes del comando **SETCYCLECOPY** deben estar codificados.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**CHKPOINT**, **COLLATE\_off**, **SETCYCLECOPY**

## COPYRANGE

**COPYRANGE** hace que la definición de la entrada de procesamiento de registros **FROMLINE** o **RPEKEY** asociada, así como las definiciones siguientes, se aplique solo a las copias especificadas [c1, c2, ... cn]. El número de copias es definido por **SETCYCLECOPY**. Consulte **FROM LINE** o **RPEKEY** para obtener información sobre la descripción de los parámetros de entrada de RPE.

### Sintaxis

```
Linenumber FROMLINE [ c1 c2 ... cn ]COPYRANGE /rpekeyname RPEKEY [ c1 c2 ... cn ]COPYRANGE
```



Nota: Utilice este comando solo con un comando **FROMLINE** o un comando **RPEKEY**.



Nota: Utilice este comando para configurar un diseño diferente para cada copia.

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

- **FROMLINE**
- **RPEKEY**
- **SETCYCLECOPY**
- **SETTXC**
- **SETPAT**

## CUTMARK

El comando **CUTMARK** imprime marcas para guiar el recorte en formatos de varias en 1 o en áreas fuera de la página final.

### Sintaxis

Xpox Ypos length width option CUTMARK

### Donde:

<b>XPos y Ypos</b>	son el origen de la marca en unidades actuales
<b>longitud</b>	es la longitud de la marca en unidades actuales
<b>anchura</b>	es la anchura de las líneas en unidades actuales.
<b>opción</b>	es un número entero que representa el tipo de marca:
	0 esquina superior izquierda
	1 esquina inferior izquierda
	2 esquina inferior derecha
	3 esquina superior derecha
	4 vertical hacia abajo
	5 vertical hacia arriba
	6 centro vertical
	7 horizontal derecha
	8 horizontal izquierda
	9 centro horizontal

### Ejemplos

```
ORITL 200 200 150 2 0 CUTMARK 200 PAGEH'-'200 150 2 1 CUTMARK PAGEW'-'200 PAGEH'-'200 1
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

SETMULTIUP, PAGEH, PAGEW

## DATAMATRIX

**DATAMATRIX** crea e imprime códigos de barras de Datamatrix ECC200 en función de los datos de cadena y parámetro especificados. No hay fuentes especiales obligatorias.

### Sintaxis

```
(data) DATAMATRIX (data) scale rotate align DATAMATRIX (data) [ /key1 value1 ... /keyN
```

### Donde:

<b>(datos)</b>	es una cadena que contiene la información que se desea almacenar en el símbolo.
<b>escala</b>	es el valor de escala (el valor prefijado es 1)
<b>anchura</b>	es la anchura solicitada del símbolo en unidades actuales. Con esta sintaxis, la anchura del símbolo se mantiene de forma continua mediante el aumento o la reducción correspondiente del ajuste a escala, independientemente del nivel de datos.



Nota: El código de barras se amplía o reduce proporcionalmente para ajustarse a la anchura especificada. Especifique una anchura capaz de mostrar un código de barras legible.

<b>girar</b>	es el valor de giro (el valor prefijado es 0)
<b>alinear</b>	es uno de los códigos de alineación (que indica el punto del código de barras que se puede alinear la posición de impresión secundaria):
	<b>0</b> superior izquierda (predeterminado)
	<b>1</b> superior derecha
	<b>2</b> centro superior
	<b>10</b> inferior izquierda
	<b>11</b> inferior derecha
	<b>12</b> centro inferior
	<b>20</b> centro izquierda
	<b>21</b> centro derecha
	<b>22</b> centro centro

La matriz opcional contiene una lista de pares de teclas o valores que se pueden usar para especificar las siguientes opciones adicionales:

<b>/MinDim</b>	Dimensión mínima del símbolo. Entero entre 8 y 144 (el valor prefijado es 8).
<b>/Rect entero</b>	Una de:  <b>0</b> siempre producir un símbolo cuadrado (predeterminado)  <b>1</b> genera un símbolo rectangular si los datos pueden ajustarse en uno de estos
<b>/Encod /enc_option</b>	Una de:  <b>/ASCII</b> Codificación ASCII (valor predeterminado)  <b>/Base256</b> Codificación Base256  <b>/C40</b> optimizado para dígitos, letras mayúsculas y espacios.  <b>/C40+</b> optimizado para dígitos, letras mayúsculas y espacios con compresión de secuencias de dígitos de más de 13 dígitos  <b>/Text</b> optimizado para dígitos, letras minúsculas y espacios  <b>/Text+</b> optimizado para dígitos, letras minúsculas y espacios con compresión de secuencias de dígitos de más de 13 dígitos.

### Ejemplos

```
(data...) [ /Rect 1 /Encod /Base256 ] DATAMATRIX (ABC 12345678998765432123) [ /Encod /C
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **AZTEC**
- **PDF417**
- **MAXICODE**
- **QRCODE**
- **MOVEH**
- **MOVEHR**
- **MOVETO**



## DEFINELAYOUT

**DEFINELAYOUT** crea y registra un diseño de varias en 1 para utilizarlo posteriormente con **SETLAYOUT**.

### Sintaxis

```
/layoutname
[ /PageWidth      pagewidth
  /PageHeight     pageheight
  /TopBleed       topbleed
  /LeftBleed      leftbleed
  /RightBleed     rightbleed
  /BottomBleed    bottombleed
  /HGutter        hgutter
  /VGutter        vgutter
  /Across         across
  /Down          down
  /Rotate         rotate
  /FillOrder      fillorder
  /LayoutMarks    markoption
  /MarkLength     marklength
  /MarkWidth      markwidth
  /MarkOffset     markoffset
] DEFINELAYOUT
```

Consulte **SETLAYOUT** para obtener una descripción de los parámetros.

### Ejemplos

```
/2x3Letter90
[ /PageWidth  216
  /PageHeight 270
  /HGutter    18
  /VGutter    18
  /Across     2
  /Down       3
  /Rotate     90
] DEFINELAYOUT
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETLAYOUT**

## DIV

**DIV** divide una variable numérica definida por **SETVAR** o una variable XML por un número.

### Sintaxis

```
/VARname number DIV
```

```
/^XMLname number DIV
```

### Donde:

<b>VARname</b>	hace referencia a una variable numérica previamente inicializada por <b>SETVAR</b> .
<b>/^XMLname</b>	hace referencia a una variable de XML. En general, las variables XML no necesitan inicializarse de forma explícita. VI Compose inicializa todas las variables XML en una cadena vacía, que equivale a una cadena numérica igual a cero.
<b>número</b>	es el número por el que se divide la variable. Puede ser un número entero, o real, o una cadena numérica. Cuando se trata de un gran número, es obligatorio contar con una cadena numérica.

Las cadenas `/Numeric` son compatibles con números grandes, de hasta 40 dígitos, 25 dígitos para la parte entera y 15 dígitos para la parte decimal. En una cadena numérica, el signo negativo y el delimitador decimal se definen mediante los parámetros `/DecimalPoint` y `/NSign` y se pueden producir en cualquier lugar de la cadena.

Es obligatorio configurar estos parámetros con los valores correspondientes para garantizar unos resultados precisos. Los valores predeterminados se definen en el archivo `/usr/xgf/src/xgf.def`. Se omiten los caracteres en la cadena numérica que no sean estos dos más los dígitos 0–9.

La longitud inicial de la cadena definida por **SETVAR** se amplía automáticamente hasta 40 dígitos cuando es necesario.

Los números reales y enteros deben utilizarse solo para los valores pequeños  $\leq 99999$ . Por ejemplo, para la implementación de un contador. El delimitador decimal, si lo hubiera, siempre es el punto ( `.` ). El signo negativo, si lo hubiera, siempre es el signo menos ( `-` ) y puede ser el primer carácter.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **ADD**
- **SUB**
- **MUL**
- **SETVAR**
- **++/-**

## DJDEBEGIN

Utilice el comando **DJDEBEGIN** para procesar el parámetro LCDS BEGIN en un procedimiento PROCESSDJDE.

### Sintaxis

```
( DJDE BEGIN parameter ) DJDEBEGIN
```

Se admiten las unidades LCDS IN y CM en la cadena de operandos. Se admiten varios comandos **BEGIN** a través de varios comandos **DJDEBEGIN**.

### Ejemplos

This is an example of DJDEBEGIN.

```
(1.2 CM, 5 CM) DJDEBEGIN
```

This is an example in PROCESSDJDE.

```
{ CASE DJDECMD
  ...
  (BEGIN) { DJDEPAR DJDEBEGIN }
  ...
ENDCASE
} 0 (DJDE) 3 PROCESSDJDE
```



Nota: Este comando realiza una llamada implícita a **SETMULTIUP** y, por lo tanto, **SETMULTIUP** no se puede usar explícitamente si se utiliza **DJDEBEGIN**.

### Modos

Este comando es aplicable solo en el modo de línea.

### Comandos relacionados

PROCESSDJDE, SETMULTIUP

## DRAWB y DRAWBR

**DRAWB** dibuja un cuadro con esquinas cuadradas. **DRAWBR** dibuja un cuadro con esquinas redondeadas. El cuadro es se describe y se rellena en función de GEPkey. Estos comandos admiten cadenas como operandos que permiten utilizar las variables DBM como operandos.



Nota: Utilice **DRAWC** para dibujar círculos o elipses.

### Sintaxis

Xpos Ypos width height GEPkey DRAWB

Xpos Ypos width height GEPkey radius DRAWBR

Donde:

<b>XPos y Ypos</b>	son la posición de la esquina superior izquierda.
<b>anchura</b>	es la anchura.
<b>altura</b>	es la altura.
<b>radio</b>	es la radio; ancho = altura = 2 x radio genera un círculo.

Las GEPkeys predefinidas son las siguientes:

FBLACK	S1	S2	S3	D1	D2	D3
XLT	XLT_S1	XLT_S2	XLT_S3	XLT_D1	XLT_D2	XLT_D3
LT	LT_S1	LT_S2	LT_S3	LT_D1	LT_D2	LT_D3
LMED	LMED_S1	LMED_S2	LMED_S3	LMED_D1	LMED_D2	LMED_D3
MED	MED_S1	MED_S2	MED_S3	MED_D1	MED_D2	MED_D3
DMED	DMED_S1	DMED_S2	DMED_S3	DMED_D1	DMED_D2	DMED_D3
DRK	DRK_S1	DRK_S2	DRK_S3	DRK_D1	DRK_D2	DRK_D3
XDRK	XDRK_S1	XDRK_S2	XDRK_S3	XDRK_D1	XDRK_D2	XDRK_D3

### Ejemplos

710	1250	1770	950	XLT	DRAWB	% shaded box
1250	900	1230	0	S1	DRAWB	% horizontal line
1250	900	0	1230	S1	DRAWB	% vertical line
710	1250	1770	950	LT_S1 30	DRAWBR	% shaded box + border + round corners
110	00	220	220	D1 110	DRAWBR	% circle

### Modos

Estos comandos son aplicables en todos los modos.

### Comandos relacionados

- DRAWC
- TPATH
- SHPATH
- OTCLIP / ITCLIP
- DRAWPATH y DRAWPATHR

## DRAWBAR

**DRAWBAR** dibuja un gráfico de barras. Este comando coloca el origen inferior izquierdo del gráfico de barras en la posición de impresión secundaria actual.

### Sintaxis

```
[ lab1 value1 lab2 value2 ... labN valueN ]width height DRAWBAR
[ lab1 value1 lab2 value2 ... labN valueN ]width height option DRAWBAR
[ lab1 [val11 val12 ..val1 1M] ... labN [valN1 valN2 ..valN NM] ] width height DRAWBAR
[ lab1 [val11 val12 ..val1 1M] ... labN [valN1 valN2 ..valN NM] ] width height option DRAWBAR
ddg_index width height DRAWBAR
ddg_index width height option DRAWBAR
```

### Donde:

<b>labX</b>	<p>es la etiqueta de la cadena de los elementos de la barra. Se pueden utilizar variables. Se admiten parámetros de atributos como, por ejemplo, la fuente y el color en el contenido de la cadena o de la variable.</p> <p>La posición de las etiquetas se determina mediante los parámetros /PrintLabel y /ChartDir.</p> <p>El ajuste de líneas de los gráficos de barras verticales se realiza en etiquetas según la anchura de las barras. El ajuste de líneas de los gráficos de barras horizontales se realiza según un porcentaje de la anchura del gráfico configurado por el parámetro /LabelColw.</p>
<b>valueX</b>	<p>es el valor (cadena o real) de un elemento de barra. Se pueden utilizar variables. Para ver las opciones de formato, consulte <a href="#">/Format</a>.</p>
<b>[ valX1 valX2 .. valXM ]</b>	<p>es el conjunto de valores de cadena (cadena o real) de un elemento de barra. Se pueden utilizar variables.</p> <p>Cuando se proporciona un conjunto de valores, las barras relevantes se apilan entre sí. El número de valores debe ser el mismo en todos los conjuntos.</p> <p>El número de colores del parámetro <a href="#">/ColorTable</a> debe coincidir con el número de valores en un determinado conjunto. El etiquetado directo de estos colores puede proporcionarse mediante el parámetro /SpotLabels.</p> <p>Para ver las opciones de formato, consulte <a href="#">/Format</a>.</p>
<b>anchura</b>	<p>es la anchura del área del gráfico de barras.</p>
<b>altura</b>	<p>es la altura del área del gráfico de barras.</p>

<b>opción</b>	como se indica en <b>Descripción del parámetro</b> . Todos los parámetros tienen valores predefinidos y se pueden omitir. Todos los parámetros establecidos aquí pueden sustituir temporalmente el valor prefijado definido por <b>SETPARAMS</b> y solo se aplicarán a ese comando. Se pueden restaurar los valores predeterminados para comandos posteriores.
<b>ddg_index</b>	se refiere a una lista de etiquetas/valores capturada por una entrada de RPE; solo se admite cero actualmente. Consulte el parámetro de alineación <b>FROMLINE</b> .

## Ejemplos

```
[ (FF) 100 (US$) 250 (DM) 150 ] 300 200 DRAWBAR
[ (FF) 100 (US$) 250 (DM) 150 ] 300 200 /117 DRAWBAR
```

Esto es un ejemplo de **DRAWBAR** con varios valores de apilado.

```
[ (FF) [ 100 20 87 ] (US$) [ 250 120 350 ] (DM) [ 150 75 123 ] ]
300 200 [ /SpotLabels [ (Cash) (Checks) (Credit Cards) ] /ColorTable
[ RED BLUE GREEN ] ] DRAWBAR
```

Esto es un ejemplo de un gráfico de barras apiladas que usa **/OffsetValue** para mostrar los valores de la pila y los totales:

```
/I /NHEB 11 INDEXFONT
/V /NHEB 14 INDEXFONT
/L 60 INDEXLSP
/W WHITE INDEXCOLOR

[ labx valx ....]
width height
[ /OffsetValue      [-2 1.3]
  /FORMATV          (/V//LTOTAL\n$0000#.00)
  /FORMATVI         (/I//W$0000#)
]DRAWBAR
```

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

- **SETPARAMS**
- **RPEKEY**
- **FROMLINE**
- **DRAWPIE**
- **DRAWCRV**

## DRAWBC

El comando **DRAWBC** dibuja códigos de barras lineales compatibles sin depender de una fuente en particular. Se puede suministrar un nombre y un tamaño de fuente para el texto legible para el ojo humano antes del comando **DRAWBC** con SETFONT o un valor de índice de la fuente.

### Sintaxis

```
(data) /BCkey option DRAWBC
```

```
(data) /BCkey [scaleH scaleV] option DRAWBC
```

### Donde:

**(datos)** es una cadena numérica que contiene los datos de codificación en el código de barras.

**/BCkey** es una de:

**/EAN8**

**/EAN13**

**/EAN5**

**/UPCA**

**/UPCE**

**/DATABAR**. Para obtener más información, consulte [DRAWBC \(extensión para admitir GS1 Databar\)](#)

**scaleH** es un factor de escala horizontal opcional (el valor prefijado es 1).

**scaleV** es un factor de escala vertical opcional (el valor prefijado es 1).

**opción** es una de:

**0** no se imprimen los dígitos legibles por el ojo humano

**10** imprimir dígitos legibles por el ojo humano

### Ejemplos



(78858101497) /UPCA 10 DRAWBC



(0123456) /UPCE 10 DRAWBC



(1123456) /UPCE [.5 1] DRAWBC



(1123456) /UPCE [1 .5] DRAWBC



## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

UPCA

## DRAWBC (extensión para admitir GS1 Databar)

El comando **DRAWBC** se ha ampliado para incluir el soporte para los códigos de barras GS1 Databar.

### Descripción

El código de barras GS1 DataBar se basa en una familia de símbolos utilizada habitualmente en los cupones GS1 DataBar. Se trata de códigos de cupones usados habitualmente en entornos comerciales. GS1 DataBar era conocido anteriormente como Simbología de espacio reducido (RSS). Su nombre se ha modificado para que coincida con el nombre de la organización GS1.

Estos códigos de barras pueden codificar hasta 14 dígitos, lo que los hace compatibles con GTIN 8, 12, 13 y 14.

GS1 DataBar Expanded y GS1 DataBar Expanded Stacked pueden codificar hasta 74 caracteres numéricos o 41 caracteres alfanuméricos, y proporcionan la capacidad de utilizar todos los identificadores de aplicación GS1 (por ejemplo, la fecha de vencimiento, el número de serie y de lote, etc.). Estos códigos de barras se utilizan con frecuencia en cupones de fabricantes.

VI Suite admite siete códigos de barras GS1 DataBar. Para obtener más información acerca de los códigos de barras de GS1 DataBar, lea detenidamente la especificación de GS1 DataBar, disponible en Internet.

- Los datos de los códigos de barras que se enumeran aquí (opción Y = 0 - 3) pueden incluir 13 o 14 dígitos. Si se especifican 14 dígitos, el decimocuarto dígito se considera un marcador de posición y puede sustituirse automáticamente por un dígito de comprobación calculado.
  - GS1 DataBar omnidireccional
  - GS1 DataBar truncado
  - GS1 DataBar apilado
  - GS1 DataBar apilado omnidireccional
- Los datos del código de barras que se enumeran aquí (opción Y = 4) pueden incluir 13 o 14 dígitos. Si se especifican 14 dígitos, el decimocuarto dígito se considera un marcador de posición y puede sustituirse automáticamente por un dígito de comprobación calculado. El dígito indicador (primer dígito) debe ser un 0 o 1.
  - GS1 DataBar limitado
- Los códigos de barras indicados aquí (opción Y = 5 o 6) pueden codificar un GTIN de 14 dígitos (debe tener una longitud de 14 dígitos), y datos adicionales (hasta 74 caracteres numéricos o 41 caracteres alfabéticos). Consulte la especificación ISO de GS1 Databar si desea obtener más información.
  - GS1 DataBar ampliado
  - GS1 DataBar ampliado apilado

### Sintaxis de comandos Databar GS1 en DRAWBC

El comando **DRAWBC** tiene la sintaxis genérica siguiente:

```
(data) /BCkey option DRAWBC
```

o

```
(data) /BCkey [scaleH scaleV] option DRAWBC
```

**Donde:**

<b>/BCkey</b>	es /DATABAR
<b>Opción</b>	es un número de 4 dígitos (ZZXY) donde:  <b>ZZ</b> Segmentos por fila. Los valores admitidos son: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20. Se utilizan solo en código de barras GS1 DataBar ampliado apilado.  <b>X</b> : 0 no imprimir los dígitos legibles por el ojo humano 1 imprimir dígitos legibles por el ojo humano  <b>Y</b> : 0 GS1 DataBar omnidireccional 1 GS1 DataBar truncado 2 GS1 DataBar apilado 3 GS1 DataBar apilado omnidireccional 4 GS1 DataBar limitado 5 GS1 DataBar ampliado 6 GS1 DataBar ampliado apilado

**Ejemplo:**

( (01) 04412345678909) /DATABAR 10 DRAWBC % GS1 Databar omnidireccional con dígitos legibles por el ojo humano.

Donde opción se define actualmente como:

<b>1</b>	imprimir dígitos legibles
<b>0</b>	omnidireccional

**Entradas válidas****Para todos los códigos de barras excepto GS1 Databar ampliado y GS1 Databar ampliado apilado**

GS1 Databar acepta 13 o 14 dígitos (la opción 5 y 6 solo aceptan 14 dígitos más datos opcionales). Cuando se especifican 14 dígitos, el dígito 14 se sustituye con un dígito de verificación calculado. Por ejemplo:

(0123456789101112)

se truncan como:

(0123456789101).

Si la entrada es inferior a 13 dígitos, se pueden agregar 0 iniciales. Por ejemplo, la entrada válida de:

(123456789)

es:

(0000123456789).

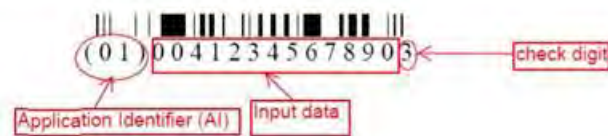
El generador de códigos de barras acepta entradas con o sin un identificador de aplicaciones (AI). AI no es una

entrada obligatoria. Sin embargo, la salida siempre tiene AI al principio en la pantalla HRI.

### Entrada con AI

El ejemplo siguiente incluye un identificador de aplicaciones ().

((01)00412345678909) /DATABAR 11 DRAWBC



Nota: En el ejemplo anterior, el decimocuarto carácter de la cadena de entrada (9) se puede descartar y sustituir por un dígito de comprobación calculado por el comando **DRAWBC**.

### Entrada sin AI

El ejemplo siguiente no incluye un identificador de la aplicación, ni tampoco incluye la interpretación legible por los usuarios (HRI).

(0041234567890) /DATABAR 01 DRAWBC



El ejemplo siguiente no incluye un identificador de la aplicación. Sin embargo, incluye la interpretación legible por los usuarios (HRI).

((0041234567890) /DATABAR 11 DRAWBC



### GS1 Databar ampliado y GS1 Databar ampliado apilado

GS1 DataBar ampliado es una simbología lineal de longitud variable capaz de codificar hasta 74 caracteres numéricos o 41 caracteres alfabéticos de datos de cadena de elementos de AI representados internamente como un número binario. Se requiere un identificador de aplicaciones de 14 dígitos en los códigos de barras GS1 Databar ampliado y ampliado apilado.

Ejemplos de entradas válidas:

(01) 90012345678908 (3103) 012233 (15) 991231

(01) 00012345678905 (10) ABC123

La entrada sin AI no es válida:

00012345678905 (10) ABC123

### Uso de GS1 Databar en DRAWBC

Valores de la opción:

00	GS1 DataBar omnidireccional
10	GS1 DataBar omnidireccional con HRI
01	GS1 DataBar truncado
11	GS1 DataBar truncado con HRI
02	GS1 DataBar apilado
12	GS1 DataBar apilado con HRI
03	GS1 DataBar apilado omnidireccional
13	GS1 DataBar apilado omnidireccional con HRI
04	GS1 DataBar limitado
14	GS1 DataBar limitado con HRI
05	GS1 DataBar ampliado
15	GS1 DataBar ampliado con HRI
06	GS1 DataBar ampliado apilado
16	GS1 DataBar ampliado apilado con HRI

```
((01)24012345678905) /DATABAR 10 DRAWBC    % Generate GS1 DataBar Omnidirectional with HRI
```

```
(24012345678905) /DATABAR 10 DRAWBC    % Generate same GS1 DataBar Omnidirectional with HRI as  
above
```

```
((01)15012345678907) /DATABAR [.5 1] 04 DRAWBC    %Generate GS1 DataBar Limited without HRI  
with scale x:0.5 , y:1
```

```
((01)00012345678905) /DATABAR [1.5 1.5] 02 DRAWBC    %Generate GS1 DataBar Stacked without HRI  
with scale x:1.5, y:1.5
```

## DRAWBM y DRAWBRM

**DRAWBM** dibuja varios cuadros con esquinas cuadradas. **DRAWBRM** dibuja varios cuadros con esquinas redondeadas. Los cuadros se describen y/o se rellena en función de GEPkey. Estos comandos también admiten cadenas como operandos que permiten utilizar las variables DBM como operandos.



Nota: Utilice **DRAWC** para dibujar círculos o elipses.

### Sintaxis

```
Xpos Ypos width height GEPkey repeat Xdispl Ydispl DRAWBM
```

```
Xpos Ypos width height GEPkey radius repeat Xdispl Ydispl DRAWBRM
```

### Donde:

<b>XPos y Ypos</b>	son las posiciones de la esquina superior izquierda del primer cuadro
<b>anchura</b>	es la anchura.
<b>altura</b>	es la altura.
<b>radio</b>	es la radio; ancho = altura = 2 x radio genera un círculo.
<b>repeticiones</b>	proporciona el número de cuadros que se dibujarán.
<b>Xdispl e Ydispl</b>	se agregan a los Xpos e Ypos del cuadro anterior para proporcionar la posición de la esquina superior izquierda del cuadro que se dibujará a continuación. El desplazamiento de Ydispl se realiza hacia arriba si se utiliza ORIBL o hacia abajo si se usa ORITL. Los números negativos invierten la dirección de desplazamiento. El valor prefijado es ORIBL.

### Ejemplos

```
120 360 1160 45 LT 20 0 100 DRAWBM % zebra
```

### Modos

Estos comandos son aplicables en todos los modos.

### Comandos relacionados

**DRAWC**

## DRAWC

**DRAWC** dibuja un círculo o una elipse. El círculo se describe y se rellena en función de GEPkey. Este comando admite cadenas como operandos, lo que le permite utilizar las variables DBM como operandos.

### Sintaxis

Xpos Ypos width height GEPkey DRAWC

### Donde

<b>XPos y Ypos</b>	son las posiciones de la esquina superior izquierda del cuadro de límite del círculo o la elipse en unidades actuales.
<b>anchura y altura</b>	son la anchura y la altura del círculo o la elipse en unidades actuales. Cuando la anchura y la altura son iguales, se dibuja un círculo. Cuando son distintas, se dibuja una elipse.
<b>GEPkey</b>	es la tecla de la propiedad del elemento gráfico que se usa para trazar el círculo.

### Ejemplos

710 1250 900 900 S1 DRAWC	% outlined circle
710 1250 900 900 XLT_S1 DRAWC	% shaded and outlined circle
710 1250 900 600 S1 DRAWC	% outlined horizontal ellipse
710 1250 600 900 XLT DRAWC	% shaded vertical ellipse

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- SETGEP
- DRAWB y DRAWBR
- TPATH
- SHPATH
- OTCLIP y ITCLIP

## DRAWCRV

**DRAWCRV** dibuja un gráfico de curva. Este comando coloca el origen inferior izquierdo del gráfico en la posición de impresión secundaria actual.

### Sintaxis

```
[ label/value list ] width height DRAWCRV
[ label/value list ] width height option DRAWCRV
ddg_index width height DRAWCRV
ddg_index width height option DRAWCRV
```

Donde:

etiqueta/lista de valores	es una lista de parejas de etiquetas y valores.
anchura	es la anchura del área del gráfico circular.
altura	es la altura del área del gráfico circular.
opción	como se indica en <a href="#">Descripción del parámetro</a> . Todos los parámetros tienen valores predefinidos y se pueden omitir. Todos los parámetros establecidos aquí pueden sustituir temporalmente el valor prefijado definido por SETPARAMS y solo se aplicarán a ese comando. Se pueden restaurar los valores predeterminados para comandos posteriores.
ddg_index	se refiere a una lista de etiquetas/valores capturada por una entrada de RPE (solo se admite cero actualmente). Consulte el parámetro de alineación <a href="#">FROMLINE</a> .

Varios gráficos dibujados con **DRAWCRV** pueden ser apilados mediante el parámetro /Stack. Al dibujar los gráficos apilados, se deben seguir las siguientes normas:

- Los parámetros /MaxVal y /MinVal pueden configurarse para garantizar una escala común para los gráficos apilados.
- Todos los gráficos de la pila pueden compartir la misma anchura y altura.
- Las etiquetas pueden especificarse solo en el primer gráfico.

### Ejemplos

```
[ (FF) 100 (US$) 250 (DM) 150 ] 300 200 DRAWCRV [ (FF) 100 (US$) 250 (DM) 150 ] 300 200
```

Este ejemplo muestra cómo dibujar tres gráficos apilados:

```
[/3D true /MaxVal 1000 /MinVal 500 /BGColor XLGREEN ] SETPARAMS [ (1998) 623 (1999) 556
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados



- SETPARAMS
- RPEKEY
- FROMLINE
- DRAWPIE
- DRAWBAR

## DRAWPAR

**DRAWPAR** es un comando gráfico utilizado para dibujar diagramas de Pareto. Un gráfico de Pareto combina un gráfico de barras y un gráfico lineal con el mismo conjunto de valores. Se utiliza para mostrar la importancia relativa de las diferencias entre los grupos de datos. El gráfico de barras muestra el conjunto de valores como haría **DRAWBAR**. El gráfico de líneas representa el porcentaje acumulativo relacionado con la suma de los valores. Los valores se esperan en orden descendente.

### Sintaxis

```
[ lab1 val1 ... labN valN ] width height DRAWPAR
[ lab1 val1 ... labN valN ] width height option1 DRAWPAR
[ lab1 val1 ... labN valN ] width height option1 option2 DRAWPAR
```

Donde:

<b>option1</b>	está vinculada al gráfico de barras
<b>option2</b>	está vinculada al gráfico de líneas

Todos los demás parámetros son similares a los del comando **DRAWBAR**.

### Ejemplos

```
/VAR_SAMPLE
[ (Traffic) 92 (Child Care) 55 (Public Transportation) 42
  (Weather) 30 (Overslept) 25 (Emergency) 15 (Other) 7
] SETVAR

VAR_SAMPLE 2500 1638 DRAWPAR % using default parameters

VAR_SAMPLE 2500 1638
[ /3D true
  /BGColor GR_SUN1H
  /BGLineColor WHITE
  /PrintValue true
  /ColorTable [XLGREEN]
  /BarSpace .1
]
[ /BarSpace .1 ]
DRAWPAR
```

Los siguientes parámetros prefijados son obligatorios para diagramas de Pareto:

- Gráfico de barras
  - /MaxVal sum\_of\_values
  - /ScaleStep (10)
  - /PrintScale 1
  - /ColorTable [MBLUE]
  - /BarSpace .03
  - /SliceSepWidth 0

- /BGLineColor LIGHT
- /3DThickness .1 /Format (@@@@@@@@@@#)
- Gráfico de líneas:
  - /BGColor null
  - /OriLine 21
  - /PrintScale 2
  - /MaxVal 100
  - /PlotSymbol [(l) current\_size RED]
  - /ColorTable [RED]
  - /BGLineColor WHITE
  - /Format (@@# %)

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- DRAWBAR
- DRAWPIE
- DRAWCRV
- DRAWDR

## DRAWPATH y DRAWPATHR

**DRAWPATH** dibuja una ruta mediante una combinación de líneas rectas y/o curvas de Bezier. **DRAWPATHR** dibuja la ruta con esquinas redondeadas.

### Sintaxis

```
[Xpos Ypos [Point1] [Point2] ... [PointN]] GEPkey DRAWPATH
[Xpos Ypos [Point1] [Point2] ... [PointN]] GEPkey radius DRAWPATHR
[ Xpos1 Ypos1 [Point11] [Point12] ... [Point1N]      % path 1
  Xpos2 Ypos2 [Point21] [Point22] ... [Point2N]      % path 2
] GEPkey DRAWPATH
```

### Donde

<b>Xpos Ypos</b>	son las coordenadas en unidades actuales del origen de la ruta.
<b>PointX</b>	define puntos consecutivos que unen el punto anterior con una línea recta o una curva de Bezier:  [Xpos1 Ypos1] son las coordenadas del final de un segmento lineal recto.  [Xpos1 Ypos1 Xpos2 Ypos2 Xpos3 Ypos3] son las coordenadas de las coordenadas de los puntos de control de la curva de Bezier.
<b>GEPkey</b>	define la tecla de la propiedad de elementos gráficos para el trazo o el relleno de la ruta.
<b>radio</b>	es el radio.

### Ejemplos

```
[200 200 [200 1200] [1400 200 300 300 400 450] [200 200]] S1 DRAWPATH
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **DRAWPOL**
- **DRAWB y DRAWBR**
- **SETGEP**
- **TPATH**
- **SHPATH**
- **OTCLIP y ITCLIP**

## DRAWPFF

**DRAWPFF** traza e inserta un campo de formulario de PDF en un documento PDF en la posición de impresión secundaria opcional. Los campos de formulario PDF serán completados por un usuario (el destinatario) y el PDF será enviado a un proceso diseñado para extraer el contenido del campo y almacenarlo en una base de datos.

El comando es muy efectivo cuando se procesa el trabajo VIPP® en un documento PDF. Cuando el trabajo se imprime de forma simple en pantalla o en el papel, solo se traza la imagen inicial del campo como se muestra cuando el documento se abre.

DRAWPFF puede crear varios tipos distintos de campos de formulario, como se indica en la sintaxis. Los campos de formulario se incluyen en el PDF usando las especificaciones de AcroForm. Para obtener más información, consulte la documentación Formularios interactivos de Adobe.

### Sintaxis

```
/type width height align [options] DRAWPFF
```

### Donde

<b>/type</b>	/TL - campo de texto (una sola línea)
	/TB - cuadro de texto (varias líneas)
	/CB - la casilla de verificación
	/RB - botón de radio
	/PL - lista desplegable
	/CL - cuadro combinado (libre escritura permitida)
	/LB - cuadro de lista
	/RS - botón de restablecimiento del formulario
	/PR - botón de imprimir el formulario
	/SF - botón de envío de formulario
	/SI - Firma digital

**anchura** es la anchura del rectángulo del campo en unidades actuales.

**altura** es la altura del rectángulo del campo en unidades actuales.

## **alinear**

Indica qué punto del campo puede estar alineado en la posición de impresión secundaria

0: superior izquierda (prefijado)

1: superior derecha

2: centro superior

10: inferior izquierdo

11: inferior derecho

12: centro inferior

20: centro izquierda

21: centro derecha

22: centro centro

## **opciones**

es una matriz de parejas las clave/valor opcionales siguientes:

**/FName** Nombre del campo

Introduzca la cadena

Valor predeterminado ninguno

Descripción Un nombre para identificar el campo en el momento de extracción. Deben ser diferentes para cada campo a excepción de un conjunto de botones de radio que deben compartir el mismo nombre

**/FDesc** Descripción alternativa del campo

Introduzca la cadena

Valor predeterminado ninguno

Descripción Un nombre de campo alternativo que debe ser usado en lugar del nombre del campo real cuando el campo puede ser identificado en la interfaz de usuario (como mensajes de estado o error del campo). Este texto también es útil al extraer el contenido del documento para facilitar la accesibilidad de usuarios con discapacidades u otros propósitos.

**/FMap** Nombre de asignación de campo

Tipo cadena

Ningún valor predeterminado

Descripción: El nombre de asignación que se utilizará al exportar datos del campo del formulario PDF

**/ReadOnly** Marca de solo lectura

Tipo entero

Valor predeterminado 0

Descripción 0 - el campo puede modificarse

1 - el campo no se puede modificar

**/Required** Marca obligatoria

Tipo entero

Valor predeterminado 0

Descripción 0 - el campo se puede dejar vacío

1 - el campo no se puede dejar vacío

**/NoExport** Marca de exportación

Tipo entero

Valor predeterminado 0

Descripción 0 - el campo se exportará

1 - el campo no se exportará

**/NoPrint** Indicador de impresión

Introduzca número entero

Valor predeterminado 0

Descripción 0: el campo se imprimirá

1: el campo no se puede modificar

**/MulSel** Marca de selección múltiple (solo LB)

Introduzca número entero

Valor predeterminado 0

Descripción 0 - solo se permite la selección de un elemento

1 - se admite la selección de varios elementos

**/FValue** Valor del campo (TL, TB, PL, CL, LB solamente)

Introduzca la cadena

Valor predeterminado ninguno

Descripción Valor predeterminado que se muestra cuando se abre el PDF.

**/DValue** Valor predeterminado de campo

Introduzca la cadena

Valor predeterminado ninguno

Descripción Valor que se muestra al pulsar el botón Restaurar.

**/IState** El estado predefinido de las casillas de verificación y los botones de

radio (deben codificarse solo con el primer botón de radio del conjunto).

Introduzca número entero

Valor predeterminado 0

Descripción 0 - estado desactivado

1 - en el estado de la casilla de verificación o del primer botón de radio

> 1 - en estado para otros botones de radio

(número de orden del botón)

**/FChoices** Matriz de opciones de valor (solo PL, CL, LB)

Tipo Matriz de cadenas O matriz de cadenas de [exportar pantalla]

Ningún valor predeterminado

Descripción Lista de opciones que se presentan al usuario. Si se usa el formato [exportar pantalla], las cadenas de exportación se usan para la exportación y las cadenas de visualización se usan para la presentación en la pantalla.

**/TAlign** Alineación de texto del campo en el campo

Introduzca número entero

Valor predeterminado 0

Descripción 0 - izquierda

1 - derecha

2 - centro

**/BColor** Color del borde de campo

Introduzca la clave de color (solo color gris, RGB y CMYK)

Ningún valor predeterminado (transparente)

Descripción Color del borde de campo

**/FColor** Color de fondo de campo

Introduzca la clave de color (solo color gris, RGB, CMYK, degradado y patrón)

Ningún valor predeterminado (transparente)

Descripción Color del borde de campo

**/VColor** Color del valor de campo

Tipo Tecla de color (solo color gris, RGB y CMYK)

Ningún valor predeterminado (transparente)



Descripción Color del valor de campo

**/BStyle** Estilo del borde de campo

Introduzca [ancho /estilo ]

Valor predeterminado [1 /s]

Estilo del borde de Descripción:

- anchura del borde en puntos
- el estilo del borde es una de las opciones siguientes:

/S sólido

/D con guiones

/B gofrado

/I grabado

/U subrayado

**/FCaption** Título del campo (solo CB, RB, RS y SF)

Tipo cadena

Valor predeterminado signo de marca (CB) o viñeta (RB)

Descripción Para la casilla de verificación y el botón de radio:

<hex code> (consulte la siguiente tabla de leyenda) para restablecer y enviar los botones: (cadena de texto)

**/TSplit** Dividir el campo de texto (solo TL, TB)

Tipo entero

Valor predeterminado 0

Descripción 0 - sin dividir

> 0 - número de divisiones y longitud máxima

**/To** Destino del envío (solo SF)

Tipo cadena URL

Ningún valor predeterminado

Correo electrónico de descripción o URL de servidor web a donde se envía la extracción del formulario (en formato PDF o FDF).

**/SubmitPDF** Formato del envío (solo SF)

Tipo entero

Valor predeterminado 0

Descripción 0 - enviar en formato FDF

- 1 - enviar como formato PDF
- 2 - enviar como formato HTML
- 3 - enviar como formato XFDF

**/TFName** Un nombre de fuente para el valor del campo. Se limita a la siguiente lista: /Helv /HeBo /HeOb /HeBO(Helvetica) /TiRo /TiBo /Tilt /TiBI (Times-Roman) /Cour /CoBo /CoOb /CoBO (Courier).

Tipo /nombre

Valor predeterminado /Helv

**/TFSize** Un nombre de fuente para el valor del campo.

Tipo entero (puntos)

Valor predeterminado 0

Descripción 0 - escalable automáticamente

>0 - tamaño de fuente fijo

**/FImage** Una imagen de fondo de campo

Tipo Cadena (nombre de imagen de TIFF, JPEG o EPS)

Ningún valor predeterminado









































































































Descripción Imagen que se utilizará como fondo del campo (alternativa a FColor)

























































































## Ejemplos

```
% text box
/TL 600 100 0
[ /FName (FirstName)
  /TAlign 2
  /FValue (John)
  /BColor ORANGE
  /BStyle [1 /B]
] DRAWPFF
```

```
% check box:
/CB 100 100 0
[ /FName (CheckBox1)
  /BStyle [1 /B]
  /FColor XLGREEN
  /VColor ORANGE
  /IState 1
] DRAWPFF
```

Tabla de títulos de la opción FCaption:

						
30	40	50	60	70	80	
						
21	31	41	51	61	71	81
						
22	32	42	52	62	72	82
						
23	33	43	53	63	73	83
						
24	34	44	54	64	74	84
						
25	35	45	55	65	75	85
						
26	36	46	56	66	76	86
						
27	37	47	57	67	77	87
						
28	38	48	58	68	78	88
						
29	39	49	59	69	79	89
						
2A	3A	4A	5A	6A	7A	8A
						
2B	3B	4B	5B	6B	7B	8B
						
2C	3C	4C	5C	6C	7C	8C
						
2D	3D	4D	5D	6D	7D	8D
						
2E	3E	4E	5E	6E	7E	
2F	3F	4F	5F	6F		

					
B0	C0	D0	E0		
					
A1	B1	C1	D1	E1	F1
					
A2	B2	C2	D2	E2	F2
					
A3	B3	C3	D3	E3	F3
					
A4	B4	C4	D4	E4	F4
					
A5	B5	C5	D5	E5	F5
					
A6	B6	C6	D6	E6	F6
					
A7	B7	C7	D7	E7	F7
					
A8	B8	C8	D8	E8	F8
					
A9	B9	C9	D9	E9	F9
					
AA	BA	CA	DA	EA	FA
					
AB	BB	CB	DB	EB	FB
					
AC	BC	CC	DC	EC	FC
					
AD	BD	CD	DD	ED	FD
					
AE	BE	CE	DE	EE	FE
AF	BF	CF	DF	EF	

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

SETPIF

## DRAWPIE

**DRAWPIE** dibuja un gráfico circular. Este comando coloca el centro de un gráfico circular en la posición de impresión secundaria actual.

### Sintaxis

```
[ label/value list ] radius DRAWPIE
[ label/value list ] radius option DRAWPIE
ddg_index radius DRAWPIE
ddg_index radius option DRAWPIE
```

### Donde:

<b>etiqueta/lista de valores</b>	es una lista de parejas de etiquetas y valores.
<b>radio</b>	es el radio del gráfico circular. Si /FitSpace > 0, el radio es el máximo radio aceptable.
<b>opción</b>	como se indica en <a href="#">Descripción del parámetro</a> . Todos los parámetros tienen valores predefinidos y se pueden omitir. Todos los parámetros establecidos aquí sustituirán temporalmente el valor prefijado definido por <b>SETPARAMS</b> y solo se aplicarán a ese comando. Se pueden restaurar los valores predeterminados para comandos posteriores.
<b>ddg_index</b>	se refiere a una lista de etiquetas/valores capturada por una entrada de RPE (solo se admite cero actualmente). Consulte el parámetro de alineación <a href="#">FROMLINE</a> .

### Ejemplos

```
[ (FF) 100 (US$) 250 (DM) 150 ] 200 DRAWPIE
[ (FF) 100 (US$) 250 (DM) 150 ] 200 [/3D true /3DThickness .8
/ColorTable [BLUE GREEN RED] ] DRAWPIE
[ (FF) 100 (US$) 250 (DM) 150 ] 200 /29 DRAWPIE
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- [SETPARAMS](#)
- [FROMLINE](#)
- [RPEKEY](#)
- [DRAWBAR](#)
- [DRAWCRV](#)

## DRAWPOL

**DRAWPOL** dibuja un polígono. Este comando admite cadenas como operandos, lo que le permite utilizar las variables DBM como operandos.

### Sintaxis

```
[ Xpos1 Ypos1 Xpos2 Ypos2 . . . Xposn Yposn ] GEPkey DRAWPOL
```

### Donde:

**XPos y Ypos** proporciona las coordenadas siguientes de un polígono, que se traza, rellena o ambas opciones según la GEPkey correspondiente.

### Ejemplos

```
[ 200 200 1240 3300 2280 200 ] LMED_S1 DRAWPOL % draws a triangle
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **DRAWB y DRAWBR**
- **TPATH**
- **SHPATH**
- **OTCLIP y ITCLIP**
- **DRAWPATH y DRAWPATHR**

## DRAWRDR

**DRAWRDR** es un comando gráfico utilizado para dibujar diagramas radiales.

### Sintaxis

```
[ etiqueta1 val1 etiqueta2 val2 ... etiquetaN valN ] radio DRAWRDR
[ etiqueta1 val1 etiqueta2 val2 ... etiquetaN valN ] radio opción DRAWRDR
[ etiqueta1 [val11 val12 ..val 1M] ... etiquetaN [valN1 valN2 ..val NM] ] radio
DRAWRDR
[ etiqueta1 [val11 val12 ..val 1M] ... etiquetaN [valN1 valN2 ..val NM] ] radio op-
ción DRAWRDR
```

### Donde:

#### labX

es la etiqueta de la cadena de los elementos de radar. Se pueden utilizar variables. Las etiquetas no ajustan las palabras.

#### valueX

es el valor, cadena o real, de un elemento de radar. Se pueden utilizar variables. Consulte el parámetro /Format para ver las opciones de formato.

#### [ valX1 valX2 ..valXM ]

es el conjunto de valores de cadena, cadena o real, de varios elementos de radar. Se pueden utilizar variables.

Cuando se proporciona un conjunto de valores, los puntos equivalentes se marcan a lo largo del mismo eje. El número de valores debe ser el mismo en todos los conjuntos.

El número de colores del parámetro /ColorTable debe coincidir con el número de valores en un determinado conjunto. El etiquetado directo de estos colores puede proporcionarse mediante el parámetro /SpotLabels. Consulte el parámetro /Format para ver las opciones de formato.

### Ejemplos

```
[ (ItemA) 20 (ItemB) 11 (ItemC) 42 (ItemD) 35 (ItemE) 95 (ItemF) 63 ] 800 DRAWRDR
```

Los siguientes parámetros predeterminados son obligatorios para diagramas radiales:

- /BGLineColor DARK
- /ColorTable [ RED ]
- /PlotSymbol [ (u) current\_size+4 null ]

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- DRAWBARDRAWPIE
- DRAWCRV
- DRAWPAR

## DUPLEX\_off

**DUPLEX\_off** desactiva la impresión a 2 caras. Esta es la opción prefijada.

Para retrasar el cambio del procesamiento a 2 caras al modo a 1 cara, y evitar el deterioro de los trabajos que realizan este cambio de forma frecuente, use **SETPARAMS** para configurar el parámetro MixPlexCount mediante estos valores:

- Si el modo de mezcla de caras impresas y el valor de MixPlexCount es mayor que 0, el valor es el número de páginas que pueden imprimir con páginas posteriores en blanco después de la ejecución del comando **DUPLEX\_off**. El modo a 1 cara exclusivo solo se activa después de ese número de páginas.
- Si el valor de mix-plex y MixPlexCount es igual a 0, se activa el modo a 1 cara exclusivo después de la ejecución de **DUPLEX\_off**. Este es el valor prefijado para las impresoras del servidor de impresión FreeFlow.
- Si el valor de MixPlexCount es igual a -1, nunca se activa el modo a 1 cara exclusivo en el modo de mezcla de caras impresas. Para garantizar la compatibilidad con versiones anteriores, este es el valor prefijado para las impresoras NPS. Si se configura este valor, la velocidad de impresión del documento sigue siendo la misma en el modo a 2 caras porque se imprime una página en blanco en la parte posterior de cada página impresa a 1 cara.

### Sintaxis

DUPLEX\_off

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **BCALL**
- **DUPLEX\_on**
- **ENDIMP**
- **TUMBLEDUPLEX\_off**
- **TUMBLEDUPLEX\_on**



## DUPLEX\_on

**DUPLEX\_on** activa la impresión a 2 caras. El valor prefijado es **DUPLEX\_off**.

### Sintaxis

DUPLEX\_on

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **BCALL**
- **DUPLEX\_off**
- **ENDIMP**
- **TUMBLEDUPLEX\_off**
- **TUMBLEDUPLEX\_on**

## ENDARBM

Utilice el comando **ENDARBM** para terminar una definición de combinación bidireccional, **BEGINARBM**.

### Sintaxis

ENDARM

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**FCALL**, **MOVETO**, **SCALL**

## ENDARBT

Utilice el comando **ENDARBT** para terminar una definición de contexto bidireccional (BEGINARBT).

### Sintaxis

ENDARBT

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**BEGINARBT**

## ENDBOOKLET

Utilice el comando **ENDBOOKLET** para terminar un folleto. Este comando debe codificarse después del comando **PAGEBRK** en la última página de los folletos.

### Sintaxis

ENDBOOKLET

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

SETPARAMS, STARTBOOKLET

## ENDCASE

**ENDCASE** puede cerrar la lista CASE. Utilice este comando solo con el comando **CASE**.

### Sintaxis

ENDCASE

### Comandos relacionados

**CASE**

## ENDCLIP

**ENDCLIP** cancela el área de recorte definida por un comando anterior mediante una GEPkey CLIP.

### Sintaxis

ENDCLIP

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

COLLATE\_dbm

## ENDIFALL

**ENDIFALL** es un comando secundario **RPE** que se utiliza cuando se anidan condiciones en cualquier nivel. **/ENDIFALL** permite cerrar todas las instrucciones **/IF** pendientes en un comando, en vez de tener que codificar todas las instrucciones **/ENDIF**. Consulte [SETRCD](#) y [SETPCD](#). Para obtener más información, consulte también [Información de comando RPE](#) y otros comandos **RPE** relacionados.

### Ejemplos

```

1 FROMLINE
  /IF_CND1
    [ .... rpe entry 1 .... ]
    [ .... rpe entry 2 .... ]
  /IF_CND2
    [ .... rpe entry 3 .... ]
  /ELSE
    [ .... rpe entry 4 .... ]
  /ENDIF
  /ELSE
    [ .... rpe entry 5 .... ]
    [ .... rpe entry 6 .... ]
  /IF_CND3
    .... rpe entry 7 .... ]
  /ENDIF
/ENDIF

10 FROMLINE
  /IF_CND4
    [ .... rpe entry 11 .... ]
    [ .... rpe entry 12 .... ]
  /ELSE /IF_CND5
    [ .... rpe entry 13 .... ]
  /ELSE /IF_CND6
    [ .... rpe entry 14 .... ]
  /ELSE /IF_CND7
    [ .... rpe entry 15 .... ]
    [ .... rpe entry 16 .... ]
  /ELSE
    [ .... rpe entry 17 .... ]
  /ENDIFALL

```

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

Ninguno

## ENDIMP

**ENDIMP** desactiva la función de imposición iniciada por **BEGINIMP**. Estos 2 comandos siempre deben estar codificados como un par.

### Sintaxis

```
BEGINIMP  
llamada a los archivos de documento tiff  
ENDIMP
```

### Ejemplos

```
TWOUP  
TUMBLEDDUPLEX_on  
ENDIMP  
BEGINIMP (report1.ps) RUNDD
```

### Modos

Este comando es aplicable en todos los modos excepto el modo de línea.

### Comandos relacionados

**BEGINIMP**



## ENDJOB

Utilice **ENDJOB** para definir una secuencia de acciones que se ejecutarán al final de un trabajo.

### Sintaxis

```
{additional actions} ENDJOB
```

### Donde:

**{acciones adicionales}** es una secuencia de comandos VIPP® que puede ejecutarse al final del trabajo. Las acciones adicionales pueden incluir una marcación adicional en la última página o en las páginas adicionales, en función de si se usa **PAGEBRK** en la secuencia, y de cómo se usa.

### Ejemplos

```
{ PAGEBRK
  /NHE 30 SETFONT
  200 3000 MOVETO
  (This job printed $$PPCOUNT. pages) VSUB 0 SHP
  PAGEBRK
} ENDJOB
```

### Modos

○Este comando es aplicable en modo de línea y de base de datos.

### Comandos relacionados

**STARTDBM, STARTLM**

## ENDOFRUN

**ENDOFRUN** actúa como un subconjunto de un determinado delimitador. Cuando se inserta en el archivo de impresión, este comando debe colocarse al principio de la última página del subconjunto.

### Sintaxis

ENDOFRUN

Especifique **ENDOFRUN** en el flujo de datos (en un NMP en modo de línea) o en un procedimiento **BEGINPAGE** basado en una condición detectada en la página como se ilustra en este ejemplo.

```
/LASTPAGE 1 60 0 --length-- /eq (endofrun_marker) SETPCD {IF LASTPAGE { ENDOFRUN } ENDI
```

Esto permite colocar un `endofrun_marker` en cualquier ubicación de la última página del conjunto.

**SETPBRK** se puede usar para configurar un delimitador de página que combina un `endofpage_marker` con un `endofrun_marker`. Si se usa el avance de página como un `endofpage_marker`, combine la configuración anterior con este ejemplo.

```
<0C> 021 SETPBRK
```

El flujo de datos se muestra como se indica a continuación:

```
%! (xyz.jdt)STARTLM primera página <FF> segunda página <FF> última página del trabajo <
```

En este ejemplo, el `endofrun_marker` es parte de la última página del trabajo. Esto puede facilitar la migración de aplicaciones de LCDS.



Nota: Esta función solo es eficaz en sistemas DocuPrint NPS con la función de conjunto múltiple activada y en el servidor de impresión FreeFlow. Para realizar una determinada acción en el subconjunto, programe los comandos **SETFINISHING** en el JDT o al principio del trabajo. En impresoras del Servidor de impresión FreeFlow, los comandos **ENDOFSET** y **ENDOFRUN** son equivalentes. No hay ninguna diferencia entre un set y un run.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

BEGINPAGE,SETPCD,SETFINISHING

## ENDOFSET

**ENDOFSET** actúa como un delimitador del conjunto. Cuando se inserta en el archivo de impresión, este comando debe colocarse al principio de la última página del conjunto.



Nota: Este comando se puede usar en un servidor de impresión FreeFlow (FFPS) sin usar el comando **SETFINISHING**. Para configurar el acabado en el FFPS, utilice una cola con la opción Salida de subjuegos. Ajuste la opción Salida de subjuegos en Recuperado de PDL.

### Sintaxis

**ENDOFSET**

Especifique **ENDOFSET** en el flujo de datos (en un NMP en modo de línea) o en un procedimiento **BEGINPAGE** basado en una condición detectada en la página como se indica a continuación.

```
/LASTPAGE 1 60 0 -length- /eq (endofset_marker) SETPCD
{ IF LASTPAGE { ENDOFSET } ENDIF } BEGINPAGE
```

Esto permite colocar un `endofset_marker` en cualquier ubicación de la última página del conjunto. **SETPBRK** ayuda a configurar un delimitador de página que combina un `endofpage_marker` con un `endofset_marker`.

Si se usa el avance de página como un `endofpage_marker`, combine la configuración anterior con este ejemplo.

```
<0C> 021 SETPBRK
```

El flujo de datos se muestra como se indica a continuación:

```
%!
(xyz.jdt)STARTLM
..... first page
<FF>
..... 2nd page
<FF>
..... last page of set
<FF>endofset_marker
..... first page of new set
```

En este ejemplo, el `endofset_marker` es parte de la última página del conjunto. Esto puede facilitar la migración de aplicaciones de LCDS.



Nota: Esta función solo es eficaz en sistemas NPS DocuPrint con la función de conjunto múltiple activada y en el servidor de impresión FreeFlow. Para realizar una determinada acción en el subconjunto, programe los comandos **/ON SETFINISHING** en el JDT o al principio del trabajo. En impresoras del Servidor de impresión FreeFlow, los comandos **ENDOFSET** y **ENDOFRUN** son equivalentes. No hay ninguna diferencia entre un set y un run.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- BEGINPAGE
- SETPCD
- SETFINISHING,
- STARTOFSET

## ENDPAGE

Utilice **ENDPAGE** para definir las acciones que el sistema lleva a cabo al final de cada página. El valor prefijado es sin acciones.

### Sintaxis

```
{end of page actions} ENDPAGE
{end of page actions} count ENDPAGE
{end of page actions} /P ENDPAGE
```

### Donde

**recuento de** es un operando opcional que define el número de páginas consecutivas en el que se pueden ejecutar las acciones. El valor prefijado del operando contador es 999999.

Los dos primeros ejemplos de sintaxis definen las acciones que el sistema ejecuta antes de que la página sea procesada. En general, estas constan de marcas finales como, por ejemplo, los datos de impresión capturados previamente mediante **GETFIELD**.

El tercer ejemplo de sintaxis define las acciones que el sistema ejecuta después de procesar la página. En general, utilice esta sintaxis para cambiar la configuración de las páginas siguientes, en función de las condiciones evaluadas por **SETPCD**. No se pueden utilizar los comandos de marca en esta sintaxis.

### Ejemplos

```
{ /NHE 20 SETFONT
  100 3000 MOVETO
  VAR.CUST SHC
} ENDPAGE

{ IF BANNER
  { /VARjdt 0 5 8 /BANNER GETFIELD
    ($$VARjdt...jdt) VSUB SETJDT
  }
  ENDIF
} /P ENDPAGE
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **BEGINPAGE**
- **SETPCD**
- **GETFIELD**
- **SLIPSHEET**

## ENDPCC

**ENDPCC** termina una definición PCC. Se puede combinar con **BEGINPCC**.

### Sintaxis

ENDPCC

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

**BEGINPCC**, **SETPCC**, **SETVFU**

## ENDRPE

**ENDRPE** termina una definición de biblioteca RPE. Se puede combinar con **BEGINRPE**. Consulte [Información de comando RPE](#) y otros comandos de RPE relacionados.

### Sintaxis

ENDRPE

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

- [BEGINRPE](#)
- [FROMLINE](#)
- [INDEXRPE](#)
- [RPEKEY](#)

## ENDTABLE

Utilice **ENDTABLE** para finalizar una tabla.

### Sintaxis

ENDTABLE

### Ejemplos

```
[ /Margins [10 10 10 10] /TableStroke B_S1 ] BEGINTABLE
[ [ /width 200 /CellText (R1/C1 Hello World) /CellFill RED ]
[ /width 300 /CellText (R1/C2 Hello Earth) /CellFill YELLOW]
[ /width 80 /CellText (R1/C3 Hello Man) /CellFill GREEN]
] SHROW
[ [ /width 200 /CellText (R2/C1 Hello World) /CellFill RED ]
[ /width 300 /CellText (R2/C2 Hello Earth) /CellFill YELLOW]
[ /width 80 /CellText (R2/C3 Hello Man) /CellFill GREEN]
] SHROW
ENDTABLE
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- ENDRPE
- FROMLINE
- INDEXRPE
- RPEKEY
- STARTLM
- SHROW
- BEGINTABLE



## ENDXPD

**ENDXPD** termina una tabla de definición de procesamiento de XML (XPD) en un archivo de ficha de trabajo de XML (XJT). Se puede combinar con un comando **BEGINXPD**.

### Sintaxis

ENDXPD

### Modos

Este comando es aplicable en el modo XML.

### Comandos relacionados

- **BEGINXPD**
- **BTA**
- **BTS**
- **ETA**
- **ETS**
- **STARTXML**

## ETA

**ETA** termina una definición de acción de modificación Se puede combinar con un comando **BTA**.

### Sintaxis

ETA

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

- **BEGINXPD**
- **BTA**
- **BTS**
- **ETS**
- **STARTXML**

## ETCLIP

Utilice **ETCLIP** para eliminar una ruta previamente configurada por las GEPkeys **TPATH**, **OTCLIP** o **ITCLIP**.

### Sintaxis

ETCLIP

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

TPATH, OTCLIP y ITCLIP

## ETS

ETS termina una definición de sustitución de modificación. Se puede combinar con un comando **BTS**.

### Sintaxis

ETS

### Modos

Este comando es aplicable en el modo XML.

### Comandos relacionados

- **BEGINXPD**
- **ENDXPD**
- **BTA**
- **BTS**
- **ETA**
- **STARTXML**

## EXIST

El comando **EXIST** comprueba la existencia de un recurso VIPP® interno o externo. El comando **EXIST** requiere el uso con el comando **IF**.

### Sintaxis

```
IF (resource_name) /rec_type EXIST
```

### Donde:

<b>resource_name</b>	Es el nombre del recurso que se comprueba.  Si resource_name es una cadena vacía, el comando <b>EXIST</b> devuelve false.
<b>rec_type</b>	Es una de las siguientes:  <b>frm</b> : Archivos de formulario VIPP®, archivos FRM y EPS, en <code>xgfc/formlib</code> . <b>dbm</b> : Archivos VIPP® maestros de base de datos (DBM) en <code>xgfc/formlib</code> . <b>seg</b> : Segmentos de PostScript de VIPP® en <code>xgfc/formlib</code> . <b>img</b> : Archivos TIFF y JPEG en <code>xgfc/formlib</code> . <b>jdt</b> : Archivos de la ficha de descripción de trabajo VIPP® en <code>xgfc/formlib</code> . <b>enc</b> : Tablas de codificación de fuentes en <code>xgfc/encoding</code> . <b>fnt</b> : Archivos de fuente. <b>mis</b> : Archivos variados. La ruta se establece mediante el comando <b>SEMPATH</b> . <b>dd</b> : Archivos de descomposición.



Nota: Las rutas de todos los archivos se definen en el archivo `xgf/src/xgfunix.run` o `xgf/src/xgdos.run`.

### Ejemplos

```
IF (truk.tif) /img EXIST
{ 100 200 MOVETO (truk.tif) 1 0 ICALL }
ENDIF
```

En el siguiente ejemplo, cuando la condición es `true`, se coloca la imagen `truk.tif`. De lo contrario, coloca una imagen `default.tif`.

```
IF (truk.tif) /img EXIST
{ 100 200 MOVETO (truk.tif) 1 0 ICALL
}
ELSE
{ 100 200 MOVETO (default.tif) 1 0 ICALL
}
ENDIF
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- ICALL
- SETFORM
- SCALL
- SETJDT
- SETENCODING
- RUN
- RUNDD

## EXIT

Utilice **EXIT** para salir del bucle REPEAT.

### Sintaxis

EXIT

### Ejemplos

Este ejemplo puede imprimir todos los documentos de servicios de descomposición en orden docu001.txt, docu002.txt, etc. y se cerrará cuando un archivo con un determinado nombre no exista, con un número máximo de 999 documentos.

```
{ /VAR1 RPCOUNT (###) FORMAT SETVAR
  IF (docu$$VAR1..txt) VSUB /dd EXIST
  { BEGINIMP
    (docu$$VAR1..txt) RUNDD
    ENDIMP
  }
  ELSE
    {EXIT}
  ENDIF
} 999 REPEAT
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

REPEAT, EXIST, IF/ELSE/ELIF/ENDIF

## FBIND

**FBIND** se utiliza exclusivamente antes de **XGFRESDEF** al incorporar un formulario de VIPP® en un trabajo VIPP®.

### Sintaxis

```
{ form contents } FBIND XGFRESDEF
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**XGFRESDEF**



## FCALL

**FCALL** ejecuta un segmento en el contexto actual, como la fuente, la impresión o la posición.

### Sintaxis

(Segmentname) **FCALL**

Los segmentos deben estar codificados en el modo nativo VIPP® y deben almacenarse en una de las bibliotecas formlib definidas por **SETFPATH** en el archivo `/usr/xgf/src/xgf`. Se recomienda el uso de la extensión `.seg`.



Nota: Al contrario que el comando **SCALL**, el comando **FCALL** no se encapsula. Al sustituir el parámetro Segmentname en **FCALL** con el contenido del segmento, obtiene el mismo resultado. Tenga en cuenta que pueden experimentarse posibles efectos secundarios. El uso de código ineficaz de PostScript puede afectar al rendimiento.

El comando **FCALL** puede ejecutar una **PAGEBRK** como parte de la definición de segmentos sin afectar al rendimiento.



Consejo: **FCALL** puede utilizarse para almacenar los fragmentos de código VIPP® que se utilizan con frecuencia o para imprimir párrafos en la página.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SCALL**, **SETFPATH**

## FILLOMR

**FILLOMR** rellena los bocadillos y los cuadros de respuesta de una cuadrícula de lectura de marca óptica (OMR).

### Sintaxis

```
(response string) FILLOMR
```

```
(response string) [ parameters ] FILLOMR
```

### Donde:

[ parameters ] puede ser uno de los parámetros enumerados aquí:

**/OMRMap (cadena de asignación)**

**/OMRDir /H o /V**

**/OMRHskip entero**

**/OMRVskip entero**

**/OMRHdisp entero**

**/OMRVdisp entero**

**/OMRSlugFont /Fontname**

**/OMRSlugSize entero**

**/OMRSlugChar cadena**

**/OMRWriteResp booleano**

**/OMRMode entero**

Los parámetros se describen detalladamente en [Descripción del parámetro](#).

Todos los parámetros tienen valores predeterminados integrados que se pueden modificar mediante **SETPARAMS**. La matriz de FILLOMR puede proporcionarse con solo los parámetros que se diferencian de las opciones predeterminadas.

La cadena de respuesta puede estar formada de caracteres en la cadena OMRMap. La fuente actual definida por **SETFONT** o un índice de fuente se utiliza para la impresión de la cadena de respuesta.

La fuente actual definida por **SETFONT** o un índice de fuente se utiliza para la impresión de la cadena de respuesta.

### Ejemplos

```

/Helvetica 15 SETFONT          % Font for response string
100 1000 MOVETO                % The origin of the top left corner of the grid.
                                % More specifically, it is the upper left corner
                                % of the grid cell (OMRHdisp X OMRVdisp) in
                                % which the first grid bubble is centered.

(398400562874234) [ /OMRMap (0123456789)
    /OMRDir /H
    /OMRHskip 1
    /OMRVskip 2
    /OMRHdisp 6
    /OMRVdisp 5
    /OMRslugFont /XOMR
    /OMRslugSize 12
    /OMRslugChar (A)
    /OMRwriteResp true
] FILLMR
100 2000 MOVETO

(4523456) [ /OMRMode 1 /OMRMap (8421) ] FILLMR
(234523456) [ /OMRMode 2 ] FILLMR

```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

SETPARAMS, MOVETO, SETFONT

## FOREACH

**FOREACH** aplica **GETITEM** a cada entrada de una matriz de variables y ejecuta el procedimiento suministrado. La matriz de variables se define previamente y se rellena con **SETVAR** y **ADD**.

### Sintaxis:

```
{proc } [ tabla ] FOREACH
```

```
{proc } [ tabla ] /MF FOREACH
```

### Donde:

<b>{Secuencia de comandos VIPP}</b>	es cualquier secuencia de comandos del modo nativo que proporciona un segmento de una página, una página completa o diversas páginas. Todos los nombres de variables definidos en la matriz de variables se pueden utilizar en esta secuencia.
<b>variable_array_name</b>	es un nombre que hace referencia a una matriz de variables en el formato adecuado para el comando <b>GETITEM</b> .
<b>/MF</b>	Cuando se utiliza la opción <b>/MF</b> , <b>SETLKF</b> , <b>FOREACH</b> y <b>SHROW</b> pueden interactuar entre sí se realiza una llamada a <b>SHROW</b> en el bucle <b>FOREACH</b> . Se evalúa el tamaño total de todos los <b>SHROW</b> invocados en el procedimiento <b>FOREACH</b> . Si no encaja en el marco, las filas no se imprimen, se realiza una llamada a <b>NEWFRAME</b> y el procedimiento se ejecuta por segunda vez con los mismos valores de entrada de la tabla. Esta opción pretende solucionar problemas de varios marcos con distintos formatos, para que en la segunda ejecución se pueda realizar una llamada diferente a <b>SHROW</b> .



Nota: En un trabajo VIPP® en modo nativo, coloque un bucle **FOREACH** con el comando **PAGEBRK** en una instrucción **BCALL** o deben procesarse en modo de marco vinculado.

```
{ x y MOVETO
  { ... PAGEBRK ... } table_name FOREACH
} BCALL
```

```
[[ x y w h 0 ]] SETLKF
{ ... } table_name FOREACH
```

### Ejemplos

Se imprimirán en estos ejemplos:

```
John Smith Paul Martin Mary White
```

```

/VAR_names
[ [ /VAR_fname /VAR_name ]
[ (John) (Smith) ]
[ (Paul) (Martin) ]
[ (Mary) (White) ]
] SETVAR

{ ($$VAR_fname. $$VAR_name.) VSUB SHL } VAR_names FOREACH

```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- GETITEM
- SETVAR
- ADD
- REPEAT
- UPDATE
- SORT

## FORMSHIFT

Utilice **FORMSHIFT** para ajustar el origen en un formulario VIPP®.

### Sintaxis

X Y FORMSHIFT

#### Donde:

X es el ajuste horizontal en unidades actuales. Puede ser negativo.

Y es el ajuste vertical en unidades actuales. Puede ser negativo.



Nota: **FORMSHIFT** solo tiene efecto en el interior de la definición de un formulario, entre corchetes {} y después del comando de orientación, si lo hay.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

DESPLAZAMIENTO

## FROMLINE

**FROMLINE** establece una definición de RPE que se aplicará en `linenumber`. Para obtener más información sobre cómo ampliar los comandos **FROMLINE** y **RPEKEY** mediante el uso del procesamiento condicional, texto fijo y procedimientos de alineación, y sobre otros comandos RPE relacionados, consulte [Elementos de RPE](#).

### Sintaxis

```
linenumber FROMLINE
[align rotate Xinit Xdisp Yinit Ydisp recpos length
 /font Colorkey]

[align rotate Xinit Xdisp Yinit Ydisp field_nr /FN
 /font Colorkey]

[ {proc} rotate Xinit Xdisp Yinit Ydisp recpos length
 /font Colorkey]

[ {proc} rotate Xinit Xdisp Yinit Ydisp field_nr /FN
 /font Colorkey]

[ {proc} rotate Xinit Xdisp Yinit Ydisp 0 (resource_name)
 /font Colorkey]

[ align rotate Xinit Xdisp Yinit Ydisp 0 (my text here)
 /font Colorkey]
```

### Donde:

<b>linenumber</b>	hace referencia a la posición del registro en el archivo de impresión desde el último delimitador de página. El comando <b>FROMLINE</b> es seguido a continuación por una o varias entradas RPE encapsuladas entre corchetes [ ].
-------------------	---

Cada entrada RPE tiene 10 parámetros que definen el campo procesado en el registro y los atributos de formato de la manera siguiente:

<b>alinear</b>	<p>puede incluir:</p> <ul style="list-style-type: none"> <li><b>0</b> alinear izquierda</li> <li><b>1</b> alinear derecha</li> <li><b>2</b> alinear centro</li> <li><b>[ Colwidth 3 ]</b> justificar en Colwidth</li> <li><b>4</b> alinear a la izquierda con espacio en blanco de inicio</li> <li><b>5</b> alinear a la derecha con espacio en blanco al final</li> <li><b>[ (string) 6 ]</b> alinear en la cadena (por ejemplo, un punto decimal)</li> <li><b>7</b> mostrar en (agregar a la cadena anterior).</li> </ul> <p>Si se amplía un parámetro de alineación con un decimal, el campo RPE relacionado puede ser capturado para ser usado posteriormente en un Data Driven Graphic (DDG). Para obtener más información, consulte <a href="#">DRAWPIE</a>, <a href="#">DRAWBAR</a> o <a href="#">DRAWCRV</a>. Al ampliar la alineación con un número decimal:</p> <ul style="list-style-type: none"> <li>x.1 - captura de un campo de valor (obligatorio)</li> <li>x.2 - captura de un campo de etiqueta (opcional)</li> </ul> <p>Por ejemplo, 1.1 puede capturar un campo de valor e imprimirlo con alineación a la derecha, [(.) 6.2] puede capturar un campo de etiqueta y alinearlo en el punto decimal.</p> <p>El parámetro alinear también se puede sustituir a través del cuerpo de un procedimiento {proceso}. Consulte Procedimiento de alineación en <a href="#">Sugerencias de programación</a>.</p>
<b>girar</b>	es el ángulo de giro en grados (los valores positivos indican un giro en sentido izquierdo).
<b>Xinit</b>	es la posición inicial horizontal en la página calculada desde el margen izquierdo.
<b>Xdispl</b>	se agrega al Xinit en registros consecutivos mediante la misma definición de RPE.
<b>Yinit</b>	es la posición inicial vertical en la página calculada desde el margen superior. En una definición de RPE, el origen se encuentra siempre en la esquina superior izquierda de la página. Este parámetro también puede expresarse como una variable, /YINIT.
<b>Ydispl</b>	se agrega al Yinit en registros consecutivos mediante la misma definición de RPE. Este parámetro también puede expresarse como una variable, /LSP.
<b>recpos</b>	es la posición de registro del campo que se va a seleccionar (empezando por 0).
<b>longitud</b>	es la longitud del campo. La longitud se puede reemplazar con un valor de cadena.
<b>field_nr</b>	es el número de campo empezando por 0. Se aplica a los registros con una estructura de campos delimitados. /FN indica que el parámetro anterior es un número de



	campo. El delimitador de campo se define con <b>SETDBSEP</b> . El valor predeterminado es <b>:</b> .
<b>fuente</b>	es el índice alfabético de fuentes definido por <b>INDEXFONT</b> . <b>/CRFT</b> es el nombre de índice reservado que se puede usar para hacer referencia a la fuente usada en la línea anterior. Se debe proporcionar un valor inicial mediante <b>INDEXFONT</b> .
<b>Colorkey</b>	se define en el archivo <code>/usr/xgf/src/xgf.gcp</code> . Para obtener más información, consulte Listas estándar, tablas, claves y atributos en la <i>Guía del usuario de FreeFlow VI Compose</i> .
<b>resource_name</b>	es el nombre del recurso que se usará con un procedimiento de alineación; consulte <a href="#">Sugerencias de programación</a> .
<b>mi texto aquí</b>	es el texto fijo que se imprimirá; consulte <a href="#">Sugerencias de programación</a> .

## Ejemplos

En este ejemplo, el valor Smith es obtenido por la entrada RPE desde el registro que sigue.

```
/ADR0
[ 0 0 100 null 200 50 3 /FN /F1 BLACK ]
ADR0:Robert:W.:Smith:New York
```

Este ejemplo muestra cómo capturar los datos (CASH/32.70 y SHARE/45.28) impresos mediante una definición FROMLINE y cómo usar los datos para crear un gráfico gestionado por datos.

```
Data File
CASH          37,586          32.70
SHARE         26,879          45.28
-----

/GraphData  14  5  /eq  (-----) SETRCD
2 BEGINRPE
1 FROMLINE
[ 0.2  0  150 null  150  50  0  10  /F4 BLACK ]
[ 0.1  0  600 null    0  0  24  5  /F4 BLACK ]

/GraphData
[ {SCALL} 0 1500 null 300 50 0 (PIE0) /F3 BLACK ]
/ENDIF
ENDRPE

/PIE0 { 0 300 DRAWPIE } XGFRESDEF
```

## USO DE FROMLINE PARA CALCULAR LA POSICIÓN DE IMPRESIÓN

La posición de la impresión de líneas consecutivas formateadas por la misma definición de RPE, o grupo de RPE si se usa RPEKEY, se computa mediante uno de estos métodos:

- Espaciado fijo de línea
- Espaciado de línea variable

El espaciado fijo de línea permite la compatibilidad con versiones anteriores. Si no se usa la clave nula en

ningún campo Xdispl o Ydispl en la definición de RPE, la posición de impresión se computa de la manera siguiente:

$$Xpos = Xinit + Xdispl * LN$$

$$Ypos = Yinit + Ydispl * LN$$

**Donde:**

**LN** es el número, a partir de cero, de la línea en el grupo actual. Cuando el grupo cambia, el valor de LN se restablece a cero.

**XPos y Ypos** se reinician con los valores Xinit e Yinit del nuevo grupo.

**Xinit, Yinit, Xdispl e Ydispl** por lo general pueden ser idénticos en todas las entradas de un grupo de RPE. Xinit o Yinit puede cambiar solo cuando el desplazamiento de X o Y correspondiente se configura como cero.

### Ejemplos

Este es un ejemplo de espaciado fijo de línea.

```
% RPE definition
5 BEGINRPE

% align rot. Xinit  Xdispl  Yinit  Ydispl  recpos  length  font  color

1  FROMLINE
   [  2  0      835    0    300    0      00    99    /F4    BLACK ]
2  FROMLINE
   [  2  0      615    0    445    0      00    99    /F1    WHITE ]
3  FROMLINE
   [  2  0     1199   318    445    0      00    99    /F1    WHITE ]
9  FROMLINE
   [  2  0     3140    0    445    0      00    99    /F5    WHITE ]
10 FROMLINE
   [  0  0       230    0    560    75     00    33    /F2    BLACK ]
   [  1  0     1345    0    560    75     33    12    /F2    BLACK ]
   [  1  0     1658    0    560    75     45    12    /F2    BLACK ]
   [  1  0     1976    0    560    75     57    12    /F2    BLACK ]
   [  1  0     3290    0    560    75    105    12    /F3    BLACK ]
   [  1  0     2286    0    560    75     69    12    /F2    BLACK ]
ENDRPE
```

Con espaciado variable de línea, Xdispl e Ydispl pueden variar en un grupo RPE o en una definición de RPE. Si las líneas que se están procesando coinciden con una definición o un grupo RPE, la posición de impresión (X e Y) resultante de la última entrada de RPE se guarda en la memoria y se usa como posición de impresión inicial de la entrada RPE siguiente. Estos valores se inicializan con Xinit e Yinit la primera vez que una definición o un grupo RPE actúan en una determinada página. En los accesos siguientes a cualquier tipo de entrada en esta definición de RPE, la posición de impresión se computa de la manera siguiente:

- Si se usa la palabra clave Null para Xdispl o Ydispl, se usan los valores Xinit o Yinit correspondientes de la entrada RPE actual.
- Si no se usa la palabra clave Null para Xdispl o Ydispl, los valores Xdispl o Ydispl se agregan a los valores en memoria. Si se activa el procesamiento de PCC, los valores de Xdispl o Ydispl se agregan antes o después de imprimir la porción correspondiente del registro en función de la definición de PCC.

Por lo tanto, si se divide un registro en varios campos que pueden imprimirse en la misma posición horizontal, Yinit e Ydispl solo son relevantes en la primera entrada RPE. Todas las demás entradas pueden tener estos parámetros configurados con el valor cero. Además, Xinit varía para reflejar las distintas posiciones horizontales de los campos y Xdispl siempre es nulo.

Este es un ejemplo del espaciado de línea variable en el que los registros LFA1 usan una fuente (F2) que requiere un espaciado de línea (Ydispl = 80) mayor que el necesario (Ydispl = 50) con la fuente (F1) usada para los registros LFA0.

```
%
/LFA0 RPEKEY
[ 0 0 70 null 910 50 5 2 /F1 BLACK]
[ 0 0 130 null 0 0 7 22 /F1 BLACK]
[ 1 0 920 null 0 0 29 6 /F1 BLACK]
[ 1 0 1170 null 0 0 35 9 /F1 BLACK]
[ 1 0 1420 null 0 0 44 8 /F1 BLACK]
[ 1 0 1620 null 0 0 52 6 /F1 BLACK]
[ 1 0 1870 null 0 0 58 9 /F1 BLACK]
[ 1 0 2100 null 0 0 67 7 /F1 BLACK]
[ 1 0 2410 null 0 0 74 11 /F1 BLACK]
/LFA1 RPEKEY
[ 0 0 70 null 910 80 5 24 /F2 BLACK]
[ 1 0 2410 null 0 0 74 11 /F2 BLACK]
```

Los registros LFA0 y LFA1 en el archivo de datos se pueden mezclar en cualquier orden; el espaciado se ajusta automáticamente.

En este ejemplo se muestra el uso del subcomando NEWPOS. Use NEWPOS si se requiere una nueva posición de impresión para que un grupo de entradas de RPE impriman el mismo campo en distintas ubicaciones en la página.

```
%
/LFA0 RPEKEY
[ 0 0 70 null 910 50 0 30 /F1 BLACK]
NEWPOS
[ 0 0 1200 null 200 80 0 30 /F3 BLACK]
```

NEWPOS hace que se guarde en la memoria un nuevo conjunto independiente de valores de posición de la impresión.

Además, con esta nueva sintaxis, los registros del mismo grupo de RPE pueden no ser consecutivos en el archivo de datos. Las sintaxis nuevas y antiguas son exclusivas de la definición de RPE especificada configurada a través de un comando RPEKEY o FROMLINE. Sin embargo, las definiciones de RPE con ambas sintaxis se pueden mezclar en bibliotecas de RPE compuestas de diversos comandos RPEKEY o FROMLINE situados entre BEGINRPE y ENDRPE.



Consejo: TLGRID puede ser una herramienta útil para configurar las definiciones de RPE.

## Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

- BEGINRPE
- COPYRANGE
- ENDRPE
- INDEXRPE
- LSP
- RPEKEY
- SETPAT
- SETPCD
- SETRCD
- SETTXC

## FSHOW

Use **FSHOW** para enviar un formulario como un archivo de impresión normal, y para obtener una muestra del formulario como salida impresa o por la pantalla.

FSHOW permite automáticamente el almacenamiento en caché si **SETFORM** o **SETBFORM** invocan al formulario.

FSHOW toma a FORMSHIFT en cuenta si el formulario se envía individualmente.



Nota: Para invocar el almacenamiento automático en caché de formularios sin elementos variables, use el comando **FSHOW**. Si se almacena en caché un formulario con **FSHOW** y el formulario contiene elementos variables, solo el primer objeto variable figurará en la memoria caché o se usará en las llamadas siguientes.

### Sintaxis

% !

```
{ form contents } FSHOW
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

Ninguno

## GETDATE

**GETDATE** establece o actualiza estas variables VIPP® relacionadas con la fecha y la hora:

<b>D_DWL</b>	Día de la semana (largo)	Domingo, lunes, martes, etc.
<b>D_DWS</b>	Día de la semana (corto)	Dom, lun, mar, etc.
<b>D_DD</b>	Día del mes	1-31
<b>D_MOL</b>	Mes (alfanumérico largo)	Enero, febrero, marzo, etc.
<b>D_MOS</b>	Mes (alfanumérico corto)	Ene, feb, mar, etc.
<b>D_MO</b>	Número de mes	01-12
<b>D_YY</b>	Año (2 dígitos)	00-99
<b>D_YYYY</b>	Año (4 dígitos)	1970-9999
<b>T_HH</b>	Horas (24)	00-23
<b>T_HH2</b>	Horas (12)	1-12
<b>T_MM</b>	Minutos	00-59
<b>T_SS</b>	Segundos	00-59
<b>T_AMPM</b>	AM/PM	am, pm
<b>T_TZN</b>	TimeZone	PST, PDT
<b>D_DOY</b>	DayOfYear	1-366

### Sintaxis

**GETDATE**

**GETDATE** se llama inicialmente en `xgf.def` y, a continuación, se llama implícitamente en cada inicialización de página. En la mayoría de casos, no debe llamar al **GETDATE** de forma explícita a menos que desee forzar una actualización inmediata de las variables de datos.

Las variables de fecha pueden utilizarse en combinación con **VSUB** para crear una marca de hora apropiada. El ejemplo siguiente imprimirá 01/24/2003 16:16:56:

```
( $$D_MO./$$D_DD./$$D_YYYY. $$T_HH.:$$T_MM.:$$T_SS. ) VSUB SH
```

El comportamiento y el resultado de **GETDATE** se ven afectados por estos parámetros:

<b>/TimeZone</b>	configurar la zona horaria, +-minutos desde UTC (-480 = PST)
<b>/DaylightSaving</b>	configurar el inicio y el final configurados del horario de verano
<b>/DaysLong</b>	lista de nombres de los días (largo)
<b>/DaysShort</b>	lista de nombres de los días (corto)

<b>/MonthsLong</b>	lista de nombres de los meses (largo)
<b>/MonthsShort</b>	lista de nombres de los meses (corto)
<b>/TimeZoneName</b>	nombres de las zonas horarias: Hora estándar, horario de verano
<b>/AmPm</b>	Designaciones de a. m./p. m., por ejemplo: [(a.m.) (p.m.)]
<b>/DefaultDate</b>	fecha que se utiliza si no hay ningún sistema de archivos disponible
<b>/DefinedDate</b>	si está presente, esta fecha sustituye la fecha del sistema



Nota: Indique la hora en formato de 24 horas para **DefaultDate** y **DefinedDate**.

Para obtener información detallada sobre estos parámetros, consulte [Descripción del parámetro](#).

Los valores predeterminados de estos parámetros, excepto **/DefinedDate**, se definen en `xf.def`. Para encontrar las variables de fecha y hora del área, y para devolver valores en un idioma que no sea inglés, es preciso editar el archivo `xf.def`. El ejemplo mostrado contiene los valores para las instalaciones en el oeste de Estados Unidos.

```

/TimeZone -480 % Pacific Standard
/DaylightSaving [ % set start, end times for Daylight Saving Time
[ 2003 60 96 120 299 120 ] % year adj. sday stime endday endtime
[ 2004 60 95 120 305 120 ] % 2004 +1Hr day95 0200Hr day305 0200Hr (4 Apr - 26 Oct)
[ 2005 60 93 120 303 120 ] % 2005 +1Hr day93 0200Hr day303 0200Hr (3 Apr - 30 Oct)
[ 2006 60 92 120 302 120 ] % 2006 +1Hr day92 0200Hr day302 0200Hr (2 Apr - 29 Oct)
[ 2007 60 70 120 308 120 ] % 2007 +1Hr day70 0200Hr day308 0200Hr (11 Mar - 4 Nov)
[ 2008 60 69 120 307 120 ] % 2008 +1Hr day69 0200Hr day307 0200Hr (9 Mar - 2 Nov)
[ 2009 60 67 120 305 120 ] % 2009 +1Hr day67 0200Hr day305 0200Hr (8 Mar - 1 Nov)
[ 2010 60 73 120 311 120 ] % 2010 +1Hr day73 0200Hr day311 0200Hr (14 Mar - 7 Nov)
[ 2011 60 72 120 311 120 ] % 2011 +1Hr day72 0200Hr day310 0200Hr (13 Mar - 6 Nov)
[ 2012 60 71 120 311 120 ] % 2012 +1Hr day71 0200Hr day309 0200Hr (11 Mar - 4 Nov)
]
/DaysLong [(Sunday)(Monday)(Tuesday)(Wednesday)(Thursday)(Friday)(Saturday)]
/DaysShort [(Sun)(Mon)(Tue)(Wed)(Thu)(Fri)(Sat)]
/MonthsLong [(January)(February)(March)(April)(May)(June)
(July)(August)(September)(October)(November)(December)]
/MonthsShort [(Jan)(Feb)(Mar)(Apr)(May)(Jun)(Jul)(Aug)(Sep)(Oct)(Nov)(Dec)]
/TimeZoneName [(PST)(PDT)]
/AmPm [(a.m.)(p.m.)]
/DefaultDate [ 2003 1 1 00 00 00 0 ] % 2003 Jan 1 00:00:00 Std time

```

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

**SPOOLNAME, SETPARAMS, SHIFTDATE**

## GETFIELD

El comando **GETFIELD** captura una parte del registro o un campo de una determinada línea y le asigna su valor a la variable especificada.



Nota: Si se usan bytes PCC en entradas **FROMLINE** o **RPEKEY**, donde `recpos=0` hace referencia al primer byte de datos que omite el PCC, la columna uno del archivo de datos original no se tratará como datos de usuario. Los datos originales se utilizan como el índice de PCC. Sin embargo, esto no es así para **GETFIELD**, **SETRCD** y **SETPCD**, donde `recpos=0` siempre hace referencia al primer byte del registro, independientemente de si **SETPCC** se codifica o no.

### Sintaxis

```
/VARname line_nr recpos length GETFIELD
/VARname line_off recpos length /PCDkey GETFIELD
/VARname line_nr field_nr /FN GETFIELD
/VARname line_off field_nr /FN /PCDkey GETFIELD
```

### Donde:

<b>line_nr</b>	es el número de la línea en la que se van a capturar los datos.
<b>recpos</b>	es la posición de la parte del registro que se desea capturar.
<b>longitud</b>	es la longitud de la parte del registro que se desea capturar.
<b>field_nr</b>	es el número del campo que se desea capturar en el registro.
<b>PCDkey</b>	determina la línea desde la que se van a capturar datos.
<b>line_off</b>	es un desplazamiento en la línea identificada por <b>PCDkey</b> . Por ejemplo: <b>0</b> selecciona la línea en la que <b>PCDkey</b> es verdadero <b>1</b> selecciona la línea siguiente <b>-1</b> selecciona la línea anterior

Si la condición es false, o si `line_nr` o `line_off` está fuera del rango, **GETFIELD** devuelve una cadena vacía.



Nota: **GETFIELD** elimina los espacios en blanco a la izquierda y la derecha en el campo extraído antes de asignarlo a la variable.

### Ejemplos

Este ejemplo muestra cómo capturar el campo de datos después de **FORM=** en una portada, y usarlo para activar un JDT en las páginas siguientes. (FORM=INV01 calls INV01.jdt)/**BANNER** debe ser verdadero para asignar a `/VARjdt` el valor capturado por **GETFIELD**.



```

/BANNER 10 9 0 5 /eq (FORM=) SETPCD
{ IF BANNER
  { /VARjdt 0 5 8 /BANNER GETFIELD
    ($$VARjdt..jdt) VSUB 1 SETJDT
  }
ENDIF
} /B BEGINPAGE

```

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

- [BEGINPAGE](#)
- [ENDPAGE](#)
- [SETPCD](#)
- [SLIPSHEET](#)

## GETITEM

Use el comando **GETITEM** para asignar un determinado conjunto de valores referenciados por un índice a un conjunto de nombres de campos. Los valores y nombres de los campos se guardan en una tabla previamente definida por **SETVAR**. La primera entrada de la tabla contiene los nombres de los campos, mientras que las siguientes entradas contienen los conjuntos de valores. **GETITEM** se utiliza en un patrón de base de datos para evitar una larga lista de **SETVAR**. Normalmente este comando se utiliza para asignar las variables de varios idiomas en función de un índice en un campo DBF.

### Sintaxis

VAR\_itemtable index GETITEM

Donde:

**VAR\_itemtable** es una tabla previamente definida con esta sintaxis:

```
/VAR_itemtable
[ [ /VAR_name1 /VAR_name2 /VAR_name3 .../VAR_nameN ]
  [ (value11) (value12) (value13) .... (value1N) ]
  [ (value21) (value22) (value23) .... (value2N) ]
  .....
  [ (valueM1) (valueM2) (valueM3) .... (valueMN) ]
] SETVAR
```

**índice** es un número entero en el rango 1 a M (longitud de la tabla).

### Ejemplos

Este ejemplo muestra cómo crear y utilizar una tabla de varios idiomas para correo de varios idiomas. En la tabla LCODE se encuentra el código de idioma con valores de 1 a 4. GENDER es el género con valores de 1 a 3.

```
/VAR_LANGUAGE
[ [ /VAR_G1 /VAR_G2 /VAR_G3 /VAR_LETTER ]
  [(Dear Sir,)(Dear Madam,)(Dear Miss,)(letter_en.ps)] % 1. English
  [(Querido Señor,)(Querida Señora,)(Querida Señorita,)(letter_fr.ps)] % 2. Spanish
  [(Cher Monsieur,)(Chère Madame,)(Chère Mademoiselle,)(letter_sp.ps)] % 3. French
  [(Sehr geehrter Herr,)(Sehr geehrte Frau,)(Sehr geehrtes Fraulein,)(letter_ge.ps)] % 4. German ]
/INI SETVAR
VAR_LANGUAGE LCODE GETITEM
VAR_LETTER CACHE SETFORM
x y MOVETO
(VAR_G$$GENDER.) VSUB2 SHL
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **SETVAR**
- **REPEAT**
- **IF/ELSE/ELIF/ENDIF**

- CASE

## GOTOFRAME

Utilice el comando **GOTOFRAME** con un número de marco para comenzar a colocar los elementos siguientes en el marco especificado. El número del marco debe ser mayor que el número de marco actual, e inferior o igual al número de marco máximo, de lo contrario la solicitud puede ignorarse.

Sintaxis

Donde:

**frame\_number** es el número de marco en el marco de destino (empezando por 1).

### Ejemplos

Este ejemplo inicia la colocación de un párrafo del texto en el marco 2.

```
2 GOTOFRAME
```

```
(My paragraph of text) 0 SHP
```

### Modos

Este comando es aplicable en modos nativo y de base de datos.

### Comandos relacionados

**FRCOUNT,NEWFRAME,SETLKF**

## ICALL

ICALL coloca un archivo TIFF 6.0 o JFIF/JPEG (con formato de línea de base codificado) en la posición de impresión actual secundaria. Las imágenes TIFF a escala de grises o de dos niveles utilizan la configuración de color actual de SETTXC.

### Sintaxis

```
(imagenname) scale rotation ICALL
() scale rotation ICALL
(imagenname) scale rotation align ICALL
() scale rotation align ICALL
```

### Donde:


<b>ImageName</b>	es el nombre del archivo TIFF o JPEG que se desea imprimir.
<b>escala</b>	es el factor de escala (1 = 100 % o sin escala). Un valor especial (2D3) está disponible para imprimir imágenes escaneadas de 200 PPP a 300 PPP con el mejor rendimiento. El valor prefijado es sin escala.
<b>rotación</b>	es el valor de giro en grados (los valores positivos indican un giro en sentido antihorario).
<b>alinear</b>	indica qué punto de la imagen puede alinearse en la posición de impresión secundaria, mediante estos valores:  <b>0</b> superior izquierda (predeterminado) <b>1</b> superior derecha <b>2</b> centro superior <b>10</b> inferior izquierda <b>11</b> inferior derecha <b>12</b> centro inferior <b>20</b> centro izquierda <b>21</b> centro derecha <b>22</b> centro centro

Las imágenes puede almacenarse en una de las bibliotecas definidas por **SETIPATH**. Se recomienda el uso de la extensión .tif o .jpg.

Esta compatibilidad se proporciona para archivos JPEG o TIFF:

- Una imagen para cada archivo. Solo se procesa la primera imagen
- Dos niveles, impresión transparente, sin compresión o comprimidos con esquemas de codificación CCITT G4/G3 o PackBits.
- Escala de grises o cuatricromía, sin comprimir o comprimido con esquemas de codificación LZW, impresión opaca

- Dos dimensiones (2D)
- Una franja o imágenes de varias franjas
- El valor de FillOrder (etiqueta 266) puede ser 1 (MSB). Se admite un valor de FillOrder de 2 (LSB) en las impresoras Docuprint NPS
- Compatibilidad con el espacio de color CIE (PhotometricInterpretation=8).
- Compatibilidad con densidad parameters=0 en archivos JPEG
- Compatibilidad con la resolución disponible en el marcador Exif de algunos archivos JPEG
- Se admite la línea de base JPEG, no se admite la codificación progresiva

 **Importante:** El marcador JPEG/Exif contiene información acerca de la resolución que se ignoró hasta la versión 4.0 de FreeFlow. En las versiones anteriores de VIPP® puede haber tenido un valor predeterminado de 300 ppp y posiblemente se haya procesado la imagen con un tamaño incorrecto. Dado que la imagen ahora se representa con el tamaño correcto, los trabajos de VIPP® existentes que hacen referencia a archivos JPEG con marcadores Exif sufrirán un problema de incompatibilidad con versiones anteriores. Para solucionar este problema de escala, es preciso modificar el comando **ICALL**. Si el cambio tiene algún efecto en la producción, el comportamiento anterior se restablecerá a través de la adición temporal de la instrucción siguiente en el JDT, o /usr/xgf/src/xgf.def: [ /ProcessExif false ] SETPARAMS

La presencia de varias franjas, escalas de grises, imágenes en color, modificaciones de escala y giros afectarán negativamente al rendimiento de la impresión.

Se recomienda la sintaxis **XGFRESDEF**.

## Ejemplos

Esto es un ejemplo de cómo llamar a un archivo TIFF interno en el flujo de datos.

```
500 900 MOVETO
( ) 1 0 ICALL
<TIFF file contents>
%%%%%%%%%%
200 300 MOVETO ...
```

En DocuPrint NPS, el archivo /var/db/PS.prefix.read puede contener las rutas de acceso a las bibliotecas de imagen o a sus elementos primarios.

Aunque habitualmente no se admiten en las especificaciones de TIFF, los archivos TFF con etiquetas de resolución (282 y 283) configuradas con el valor cero asumen el valor predeterminado 300 ppp. La etiqueta 262 (PhotometricInterpretation) asume el valor predeterminado cero. Se admiten distintas etiquetas de resolución XY.

 **Nota:** Para almacenar imágenes en caché, use **SCALL** en lugar de **ICALL**, ya que **CACHE** no se puede combinar con **ICALL**.

Dado que **ICALL** y **SCALL** tienen distintas alineaciones predefinidas, al convertir una sintaxis de **ICALL** sin opción de alineación, la sintaxis **SCALL** deberá especificar la opción de alineación 0 para garantizar la colocación correcta de la imagen. Por ejemplo:

```
(myfile.tif) 1 0 ICALL
```

se debe convertir a:

```
(myfile.tif) CACHE 1 0 0 SCALL
```

**Modos**

Este comando es aplicable en todos los modos.

**Comandos relacionados**

- **BCALL**
- **ENDIMP**
- **IGNOREBT\_off**
- **IGNOREBT\_on**
- **IREVERSE\_off**
- **IREVERSE\_on**
- **MOVEH**
- **MOVETO**
- **RUNTIF**
- **SETTXC**
- **TIFORI\_off**
- **TIFORI\_on**

## IF/ELSE/ELIF/ENDIF

Los comandos **IF**, **ELSE** y **ENDIF** presentan el modo nativo y la lógica condicional de la base de datos maestra. **ELIF** complementa el conjunto de instrucciones condicionales **IF/ELSE/ENDIF**. Combina **ELSE** e **IF** en una instrucción para evitar anidaciones complejas.

### Sintaxis

```
IF condition { ... true action ... } ENDIF

IF condition { ... true action ... } ELSE          { ... false action ... } ENDIF

IF condition1  { ... action if condition1 is true ... } ELIF condition2 { ... action i
IF condition1  { ... action if condition1 is true ... } ELIF condition2 { ... action i
```

### Donde

La **condición**: puede ser uno de los siguientes:

- **elemento1 elemento2 test\_op**
- **PCDkey**
- **condición1 condición2 bool\_op**
- **item1 e item2** pueden ser cadenas fijas, números enteros, números reales, nombres de variables definidas por **SETVAR** o **GETFIELD**, nombres de campos de bases de datos o etiquetas XML.
- **test\_op** es un operador de prueba como:
  - **eq** igual
  - **ne** distinto
  - **gt** mayor que
  - **ge** mayor o igual que
  - **lt** menor que
  - **le** menor o igual que
- **CIEQ** igual sin diferenciar entre mayúsculas y minúsculas
- **CINE** desigual sin diferenciar entre mayúsculas y minúsculas
- **HOLD** busca una cadena en cualquier parte del campo seleccionado, del área definida o de los caracteres en una línea.

Para obtener más información, consulte [Operadores de prueba y expresiones condicionales](#).

- **bool\_op** es un operador booleano como:
  - **or** verdadero si cualquier condición es verdadera
  - **and** verdadero si se cumplen las dos condiciones



Para obtener más información, consulte [Operadores de prueba y expresiones condicionales](#).

- **PCDkey** es una condición definida por SETPCD

### Ejemplos

Este ejemplo imprime Your gift will be a pocket razor si el contenido de FIELD1 es Sir, and Your gift will be a hand bag mirror si FIELD1 contiene otro texto.

```
IF FIELD (Sir) eq { (Your gift will be a pocket razor) SHL } ELSE { (Your gift will be
```

Use este ejemplo para imprimir la cadena Is life good or what!! solo para hombres de California de más de 40 años de edad.

```
IF State (California) eq Age (40) ge and Gender (Male) eq and { Is life good or what!!)
```

### Modos

Estos comandos son aplicables en modo nativo, el modo de línea y el modo de base de datos.

### Comandos relacionados

[SETVAR](#), [GETFIELD](#)

## IGNOREBT\_off

### IGNOREBT\_off

IGNOREBT\_off hace que el comando **ICALL** cancele el trabajo en archivos TIFF no admitidos o que están dañados. **IGNOREBT\_off** es el valor predeterminado.

### IGNOREBT

IGNOREBT significa que se ignoran los TIFF incorrectos.

### Sintaxis

IGNOREBT\_off

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

IGNOREBT\_on, ICALL

## IGNOREBT\_on

IGNOREBT\_on hace que los archivos TIFF dañados o no admitidos sean omitidos por el comando **ICALL**. No se imprime el archivo TIFF relacionado y el trabajo se finaliza. Se produce una lista de los archivos TIFF omitidos en la página de errores al final del trabajo. De manera predeterminada, el comando **ICALL** cancela el trabajo en archivos TIFF que están dañados o no son admitidos.

El valor prefijado es IGNOREBT\_off.

### Sintaxis

IGNOREBT\_on

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

IGNOREBT\_off, ICALL

## ILAND

ILAND establece la orientación horizontal inversa. La orientación horizontal inversa se obtiene mediante la rotación de una hoja vertical 90 grados en sentido horario.

### Sintaxis

ILAND

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

LAND,IPORT,PORT

## INDEXALIGN

INDEXALIGN asocia un valor de alineación con un ALIindex. Una vez definido por INDEXALIGN, un **ALIindex** sigue las mismas reglas y los mismos comportamientos que los índices definidos por un comando **INDEX**. Sin embargo, **INDEXALIGN** solo es efectivo en los bloques de texto que se imprimen con **SHP**.

Para obtener más información, consulte [INDEXCOLOR](#).

### Sintaxis

```
/ALIindex align INDEXALIGN
```

### Donde

#### ALIindex

Es una cadena alfanumérica.

#### Alinear

es uno de estos códigos de alineación:

- **0** izquierda
- **1** derecha
- **2** centro
- **3** justificar con la última línea alineada a la izquierda
- **4** justificar con la última línea alineada a la derecha
- **5** justificar con la última línea centrada
- **6** justificar todas las líneas

### Ejemplos

```
(//) 2 SETFTSW /F1 /NHE 12 INDEXFONT /A2 2 INDEXALIGN /A3 3 INDEXALIGN (centra la palab
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

[SETLSP](#), [SHP](#) y [SHp](#), [SETFTSW](#)

## INDEXBAT

**INDEXBAT** asocia un BATkey con un BATindex.

Una vez definido por INDEXBAT, BATindex sigue las mismas reglas y comportamientos que un Colorindex definido por INDEXCOLOR. Para obtener más información, consulte [INDEXCOLOR](#).

### Sintaxis

```
/BATindex /BATkey INDEXBAT /BATindex null INDEXBAT
```

### Donde

#### BATindex

Es una cadena alfanumérica.

#### BATkey

Es un **BATkey** definido por SETBAT.

#### nulo

Define un **BATindex** para cancelar el actual.

Para obtener más información sobre **BATkeys** predefinidas, consulte *Listas estándar, tablas, claves y atributos* en la *Guía del usuario de FreeFlow VI Compose*.

### Ejemplos

Este ejemplo imprime Switching from underlined text to regular.

```
/U /UNDL INDEXBAT /N null INDEXBAT (Switching from //U underlined text//N to regular) 0
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- [INDEXFONT](#)
- [INDEXCOLOR](#)
- [SHMF](#), [SHMf](#) y [SHmf](#)
- [SHX](#)
- [SETFTSW](#)

# INDEXCOLOR

**INDEXCOLOR** asocia un color y el patrón con la tecla **Colorindex**.

Una vez definido por **INDEXCOLOR**, **Colorindex** se puede usar externamente como un comando independiente entre comandos **SHX** posteriores o después de un prefijo de cambio de fuente en datos de impresión procesados por **SHMF** o **SHP**. El prefijo de cambio de fuente se define mediante **SETFTSW** (el valor predeterminado es //).

Cuando se utiliza de forma externa, **Colorindex** puede ser una cadena alfanumérica que empieza con un carácter alfabético. Cuando se usa en datos imprimibles, **Colorindex** puede ser una cadena de caracteres alfanuméricos cuya longitud está definida por **SETFTSW** (el valor predeterminado es 1).


Colorkeys se definen en /usr/xgf/src/xgf.gep.

## Sintaxis

```
/Colorindex Colorkey INDEXCOLOR /Colorindex /Colorkey INDEXCOLOR /Colorindex [ Colorkey
```

Donde:

ColorIndex	es una cadena alfanumérica.
Colorkey	es un Colorkey definido en /usr/xgf/src/xgf.gep.
PATkey	es una clave de trama definida por SETPAT.
nulo	sale del área transparente en lugar de rellenarse con un color.
Tintlevel	es un número real (entre 0 y 1) que especifica el nivel de tinta.
TRlevel	es un número real (entre 0 y 1) que especifica el nivel de transparencia.  TRlevel sólo es efectivo en VI eCompose y APPE RIP. En un PS RIP se emula mediante un matiz opaco.

 Nota: Para obtener más información sobre claves de patrones y colores predefinidos, consulte *Listas estándar, tablas, claves y atributos y Tonos de color* en la *Guía del usuario de FreeFlow VI Compose*.

## Ejemplos

Este ejemplo muestra cómo aplicar índices de color de tono y transparencia.

```
/T (RED~.5) INDEXCOLOR /R (RED#.5) INDEXCOLOR (//TText using tinted red //RTest using t
```

Este ejemplo muestra dos cadenas de texto que se imprimen mediante una combinación de índices de fuentes, colores, tonos y transparencias.

```
/1 /NHE 18 INDEXFONT  
/2 /NHE 24 INDEXFONT  
/3 /NHE 18 INDEXFONT  
/A BLACK INDEXCOLOR  
/B BLUE INDEXCOLOR  
/Y (RED ~.5) INDEXCOLOR      % sets 50% tint  
/Z (BLUE #.75) INDEXCOLOR    % sets 75% transparency
```

```
200 200 MOVETO  
(//1//Atext using font 1 black //2//Bswitch to font 2 blue //3and Font 3) 0 SHMF
```

```
200 400 MOVETO  
(//1//Atext using font 1 black //2//Yswitch to font 2 red 50% tint //3//2and Font 3 with 75% transparency) 0 SHMF
```

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

- INDEXBAT
- SHMF, SHMf y SHmf
- SETFTSW
- SETPAT
- SETTXC
- SHP y SHp
- SETCOL
- SETTRAN



## INDEXFONT

**INDEXFONT** asocia un nombre y un tamaño de fuente con **Fontindex**.

Una vez definido por **INDEXFONT**, **Fontindex** se puede usar externamente como un comando independiente entre comandos **SHx** posteriores o después de un prefijo de cambio de fuente en datos de impresión procesados por **SHMF** o **SHP**. El prefijo de cambio de fuente se define mediante **SETFTSW** (el valor predeterminado es //). Para obtener más información, consulte [SHMF](#), [SHMF](#) y [SHmf](#).

Cuando se utiliza de forma externa, **Fontindex** puede ser una cadena alfanumérica que empieza con un carácter alfabético. Cuando se usa en datos imprimibles, **Fontindex** puede ser una cadena de caracteres alfanuméricos cuya longitud está definida por **SETFTSW** (el valor predefinido es 1).

Para obtener más información, consulte [Espaciado](#).

### Sintaxis

```
/Fontindex /Fontname size INDEXFONT
```

```
/Fontindex /Fontname sizex sizey INDEXFONT
```

### Donde:

<b>Fontindex</b>	es una cadena alfanumérica. El índice integrado de la fuente, /CRFT, hace referencia a la fuente actual y tiene como objetivo las tablas RPE.
<b>Fontname</b>	<p>es una de las siguientes:</p> <p>El nombre de una fuente elegida en las listas de fuentes VIPP® activadas por <b>SETENCODING</b> en el archivo <code>/usr/xgf/src/xgf.def</code>. Para obtener más información, consulte “Atributos, claves, tablas y listas estándar” en la <i>Guía del usuario de FreeFlow VI Compose</i>.</p> <p>Un Fontname definido para facilitar el tipo de fuente a otro miembro de la misma familia de fuentes sin tener que volver a especificar el Fontname:</p> <p><b>/~REG</b> para normal</p> <p><b>/~BLD</b> para negrita</p> <p><b>/~ITL</b> para cursiva</p> <p><b>/~ BDI</b> para negrita cursiva</p> <p><b>/~ CUR</b> para la fuente actual</p> <p>Por ejemplo, si la fuente actual es Helvetica, 12 puntos, como se define a través de <b>/NHE 12 SETFONT</b>, un <b>INDEXFONT</b> en el que se especifica el Fontname <b>/~BLD</b> especial cambia la fuente a Helvetica-Bold. Cuando la fuente actual es Times cursiva, la misma definición de <b>INDEXFONT</b> cambia la fuente a Times negrita. El Fontname especial <b>/~CUR</b> se puede usar para cambiar el tamaño de una fuente y mantener el Fontname sin cambios. La especificación de un tamaño de la fuente como nulo mantiene el tamaño del punto.</p> <p>Para obtener más información, consulte <a href="#">Aplicar atributos para las fuentes</a>.</p>

**tamaño**

es una de las siguientes:

**Un número real o entero** que especifica el tamaño de la fuente en unidades de 1/72 de pulgada (puntos). Si se da un tamaño igual a 0, la escala de la fuente se ajusta automáticamente según los márgenes y la cuadrícula definidos por **SETMARGIN** y **SETGRID**. En este caso, debe utilizarse una fuente fija (por ejemplo, Courier).

**Nulo** para especificar el tamaño del punto actual.

**sizeX**

permite especificar el tamaño de la fuente en la dirección de X en puntos. Los valores son los mismos tal como se define en el tamaño, sin embargo, un valor de 0 no se permite.

**sizeY**

permite especificar el tamaño de la fuente en la dirección de Y en puntos. Los valores son los mismos tal como se define en el tamaño, sin embargo, un valor de 0 no se permite.

Cuando se utiliza la segunda sintaxis, la fuente se ajusta a escala con valores distintos en los ejes X e Y.



Nota: Las variables integradas, *GLT* y *MPR*, que se usan en impresiones especiales, se pueden usar como opciones en este comando.

Para obtener más información, consulte [Impresión especializada](#).

**Ejemplos**

Este ejemplo muestra el uso de un índice de fuentes de tres caracteres con un comando **SHMF**.

```
(//) 3 SETFTSW
/H10 /NHE 10 INDEXFONT
/H12 /NHE 12 INDEXFONT
/BLD /~BLD null INDEXFONT
/CUR /~CUR 10 INDEXFONT
(//H10 use Helvetica 10 //H12 use Helvetica 12) 0 SHMF
(//BLD use Helvetica Bold 12 //CUR use Helvetica Bold 10) 0 SHMF
```

**Modos**

Este comando es aplicable en todos los modos.

**Comandos relacionados**

- [INDEXBAT](#)
- [SETENCODING](#)
- [SETFONT](#)
- [SHMF](#), [SHMf](#) y [SHmf](#)
- [SHP](#) y [SHp](#)
- [SETFTSW](#)

## INDEXKERN

**INDEXKERN** asocia opciones de espaciado con una clave **Kernindex**.

Una vez definido por **INDEXKERN**, **Kernindex** se puede usar externamente como un comando independiente entre comandos **SHx** posteriores o después de un prefijo de cambio de fuente en datos de impresión procesados por **SHMF** o **SHP**. El prefijo de cambio de fuente se define mediante **SETFTSW** (el valor predeterminado es //).

Cuando se utiliza de forma externa, **Kernindex** puede ser una cadena alfanumérica que empieza con un carácter alfabético. Cuando se usa en datos imprimibles, **Kernindex** puede ser una cadena de caracteres alfanuméricos cuya longitud está definida por **SETFTSW** (el valor predefinido es 1).

### Sintaxis

```
/Kernindex [ PW_opt TG_opt TK_deg ] INDEXKERN
```

### Donde:

<b>Kernindex</b>	es una cadena alfanumérica.
<b>PW_opt</b>	define la opción de espaciado de pares PW_opt puede tener uno de estos valores:  <b>0</b> desactiva el espaciado de pares.  <b>not 0</b> activa espaciado de pares a través de la multiplicación de valores de espaciado proporcionados por el archivo AFM (entradas KP, KPX o KPY). El valor recomendado es 1 y puede ser un número real.  <b>null</b> mantiene la opción de espaciado de pares actual en vigor.
<b>TG_opt</b>	define la opción de espaciado de seguimiento genérico. TG_opt puede tener uno de estos valores:  <b>0</b> desactiva el espaciado de seguimiento genérico  <b>not 0</b> activa espaciado de seguimiento a través de la multiplicación de los valores de espaciado de seguimiento genéricos definidos por VI Compose. El intervalo de valores recomendado es de -3 a + 3 y puede ser un número real.  <b>null</b> mantiene la opción de espaciado de seguimiento genérico en vigor.
<b>TK_deg</b>	define el grado de espaciado de seguimiento. TK_deg puede tener uno de estos valores:  <b>0</b> desactiva el grado de espaciado de seguimiento.  <b>not 0</b> activa el espaciado de seguimiento mediante la selección de valores del grado de espaciado de seguimiento más próximo definido en las entradas TrackKern del archivo AFM. El grado de espaciado de seguimiento va normalmente de -3 a +3, y debe ser un entero.  <b>null</b> mantiene el grado de espaciado de seguimiento actual.

### Modos

Este comando es aplicable en todos los modos.

**Comandos relacionados**

SHMF, SHMf y SHmf, SHP y SHp, SETKERN

## INDEXLSP

**INDEXLSP** asocia un valor de espaciado de línea con un **LSPindex**. una vez definido por **INDEXLSP**, un **LSPindex** sigue las mismas reglas y los mismos comportamientos que los índices definidos por un comando **INDEX**. Para obtener más información, consulte **INDEXCOLOR**.

**INDEXLSP** tiene como objetivo principal ser usado en bloques de texto impresos con **SHP** en combinación con distintas fuentes y tamaños.

### Sintaxis

```
/LSPindex LSPval INDEXLSP
```

### Donde:

<b>LSPindex</b>	es una cadena alfanumérica.
<b>LSPval</b>	es un valor del espaciado de línea. El número real es en unidades actuales.

### Ejemplos

```
POINT SETUNIT
(//) 2 SETFTSW
/F1 /NHE 12 INDEXFONT
/L1 14.4 INDEXLSP
/F2 /NHE 20 INDEXFONT
/L2 24 INDEXLSP
(//F1//L1switching from small to //F2//L2larger text
//F1//L1and back to small) 0 SHP
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETLSP**, **SHP** y **SHp**, **SETFTSW**

## INDEXOTL

**INDEXOTL** asocia parámetros de contorno de texto con un índice **OTL**. Una vez definido por **INDEXOTL**, un **OTL** sigue las mismas reglas y los mismos comportamientos que los índices definidos por un comando **INDEX**. Para obtener más información, consulte **INDEXCOLOR**.

### Sintaxis

```
/OTLindex [ LineWidth Colorkey ] INDEXOTL
```

### Donde:

<b>LineWidth</b>	es la anchura de la línea en unidades actuales.
<b>Colorkey</b>	define el color de la línea

### Ejemplos

```
/OTL1 [ 2 RED ] INDEXOTL
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETOTL, SHP y SHp, SETFTSW**

## INDEXPIF

**INDEXPIF** asocia una nota o un destino **PIF** con un índice **PIF**. Una vez definido por **INDEXPIF**, un índice **PIF** sigue las mismas reglas y los mismos comportamientos que un **Colorindex** definido por un comando **INDEX-COLOR**. Para obtener más información, consulte **INDEXCOLOR**.

### Sintaxis

#### Donde:

<b>PIFindex</b>	es una cadena alfanumérica.
<b>PIFtype</b>	puede ser uno de los siguientes: <ul style="list-style-type: none"> <li>• <b>PAGE</b> una página en el documento</li> <li>• <b>DEST</b> un destino con nombre definido por PDFDEST</li> <li>• <b>XPAGE</b> una página en otro documento PDF</li> <li>• <b>XDEST</b> un destino con nombre en otro documento PDF</li> <li>• <b>FILE</b> un documento no PDF</li> <li>• <b>URI</b> un sitio de Internet o Intranet o un documento</li> <li>• <b>NOTE</b> una nota</li> </ul>
<b>paramX</b>	en función de PIFtype, deben suministrarse estos parámetros: <ul style="list-style-type: none"> <li>• [ /PAGE pagenum view ]</li> <li>• [ /DEST /destname ]</li> <li>• [ /XPAGE (fileref) pagenum view ]</li> <li>• [ /XDEST (fileref) /destname ]</li> <li>• [ /FILE (fileref) ]</li> <li>• [ /URI ]</li> <li>• [ /URI (URIstring) ]</li> <li>• [ /NOTE (title) (contents) ]</li> <li>• [ /NOTE (title) (contents) notetype color option ]</li> </ul>



Nota: Para obtener una lista de los valores posibles, consulte **SETPIF**.

#### nulo

define un índice PIF para cancelar el índice PIF actual.



Nota: Dicho índice solo resulta útil cuando el índice PIF se utiliza en un fragmento de texto en una cadena que se imprime mediante SHP o SHMF. Para cualquier otro uso, el índice PIF se cancela automáticamente si se asocia a un elemento.



Nota: Los datos de un solo byte pueden cifrarse mediante ISO-8859-1. Los datos de varios bytes pueden cifrarse mediante UTF8. VI Compose los puede convertir automáticamente en UTF16 para insertarlos en el PDF porque esta es la única codificación de varios bytes admitida por el formato PDF. Para activar la conversión de datos UTF8 a UTF16, la fuente actual seleccionada por **SETFONT** o **INDEXFONT** puede tener una codificación UTF8.

### Ejemplos

Este ejemplo muestra cómo asociar una dirección URL con un logotipo y un marcador mediante un índice PIF:

```
/LX [ /URI (http://www.xerox.com) ] INDEXPIF
```

```
/LX [ /URI (http://www.xerox.com) ] INDEXPIF
```

```
LX (Xerox web site) BOOKMARK
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- PDFDEST
- SETPIF
- BOOKMARK
- PDFOPEN
- PDFINFO



## INDEXRPE

**INDEXRPE** registra el RPE actual bajo una clave.

En el ejemplo de sintaxis más abajo, **INDEXRPE** registra el **RPE** actual bajo el **RPEname** usado posteriormente como un comando de ejecución automática en un JDT o un prefijo de modo nativo (NMP). **INDEXRPE** puede estar codificado inmediatamente después de **ENDRPE**.

El último **RPE** definido por **ENDRPE** permanece activo.

Para obtener más información, consulte [Información de comando RPE](#) y otros comandos **RPE** relacionados.

### Sintaxis

```
/RPEname INDEXRPE
```

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

- **BEGINRPE**
- **ENDRPE**
- **FROMLINE**
- **RPEKEY**

## INDEXSST

**INDEXSST** asocia una secuencia de **SST\_param** con un **SSTindex**. Una vez definido por **INDEXSST**, un índice **SSTindex** sigue las mismas reglas y los mismos comportamientos que un **Colorindex** definido por un comando **INDEXCOLOR**.

Este índice se utiliza para activar atributos de superíndice y subíndice. Para obtener más información, consulte **SETTXS**.

### Sintaxis

```
/SSTindex sst_param INDEXSST
/SSTindex [ Ydispl Fshrink ] INDEXSST
```

### Donde:

<b>SSTindex</b>	es una cadena alfanumérica.
<b>sst_param</b>	se define en <b>SETTXS</b> . Consulte <b>SETTXS</b> para obtener una descripción de este operando.
<b>Ydispl</b>	define el desplazamiento vertical como un factor del tamaño de fuente actual. Pueden ser positivo, mostrado como superíndice, o negativo, mostrado como subíndice. El intervalo de valores esperado es [-1 +1].
<b>Fshrink</b>	define el factor de reducción que se aplica a la fuente actual. El intervalo de valores esperado es [>0 +1].

### Ejemplos

Ambos ejemplos imprimen *March, 17<sup>th</sup>*.

```
/1 /NHE 16 INDEXFONT
/2 /NHE 8 INDEXFONT
/3 20 INDEXSST
/4 null INDEXSST
100 3000 MOVETO
(//1March, 17//2//3th//4//1) 0 SHMF

/S /SUP INDEXSST
/N null INDEXSST
/NHE 16 SETFONT
100 3000 MOVETO
(March, 17//sth//N) 0 SHMF
```

El ejemplo establece un superíndice con un desplazamiento vertical igual al 40 % del tamaño actual, y la reducción de fuente actual en un 60 %.

```
/ss1 [.4.6] INDEXSST
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- SETFTSW
- SETTXS
- SHMF, SHMf y SHmf
- SHP y SHp

## I PORT

**I PORT** establece la orientación vertical inversa. La orientación vertical inversa se obtiene mediante un giro de 180 grados de una hoja vertical.

### Sintaxis

I PORT

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

ILAND, LAND, PORT

## IREVERSE\_off

**IREVERSE\_off** hace que se impriman imágenes en dos niveles en modo normal. Esta es la opción prefijada.

Las imágenes de dos niveles son transparentes. Las partes blanco no se superponen a las marcas anteriores en la página.

### Sintaxis

```
IREVERSE_off
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

ICALL, IREVERSE\_on

## IREVERSE\_on

**IREVERSE\_on** hace que se impriman imágenes en dos niveles en modo al revés. El valor prefijado es **IREVERSE\_off**.

Las imágenes de dos niveles son transparentes. Las partes blanco no se superponen a las marcas anteriores en la página.

### Sintaxis

IREVERSE\_on

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

ICALL, IREVERSE\_off

## JOG\_on y JOG\_off

Use **JOG\_on** y **JOG\_off** para desplazar las páginas una por una, en vez de desplazarlas en grupo como ocurre con la opción **ENDOFSET**. Si **JOG\_on** se codifica al principio de una página, se produce un desplazamiento de la página siguiente y de todas las páginas posteriores. **JOG\_off** deja de aplicar el desplazamiento a partir de la página siguiente. El comando sustituido, **OFFSET\_on**, no se puede utilizar si se utiliza **JOG\_on**.

El uso de **JOG\_on** y **JOG\_off** es más eficiente que el uso de **ENDOFSET**.



Nota: Estos comandos solo son efectivos en sistemas DocuPrint NPS y en el servidor de impresión FreeFlow.

### Sintaxis

`JOG_on`

`JOG_off`

### Ejemplos

Utilice este ejemplo para terminar un **DBM** de varias páginas para que desplace los documentos **DBM** a partir del siguiente.

```

****
PAGEBRK          % end of last-but-one page
JOG_on
*****
PAGEBRK          % last page
JOG_off

```

### Modos

Estos comandos son aplicables en todos los modos.

### Comandos relacionados

**ENDOFRUN**, **ENDOFSET**

## LAND

**LAND** configura la orientación horizontal. La orientación horizontal se obtiene mediante la rotación de una hoja vertical 90 grados hacia la izquierda. La orientación predeterminada es **PORT** y se configura en el archivo `/usr/xgf/src/xgf.def`.

### Sintaxis

LAND

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

ILAND, IPORT, PORT



## LMSKIP

Use **LMSKIP** para omitir un número de líneas, un número de caracteres o una matriz de bytes del principio de los datos del modo de línea. El comando **LMSKIP** se puede colocar en la JDT antes del comando **STARTLM**.

### Sintaxis

```
item option LMSKIP
```

#### Donde:

##### elemento

es:

- Un número de líneas o bytes para las opciones /L o /C
- Una matriz de números ASCII para la opción /B

##### opción

es:

- /L para omitir líneas
- /C para omitir caracteres
- /B para omitir bytes

### Ejemplos

```
3 /L LMSKIP
10 /C LMSKIP
[ 16#0A 16#0D 16#0C ] /B LMSKIP
[ 10 12 13 ] /B LMSKIP
```

```
% skip first 3 lines
% skip first 10 characters
% skip all bytes whose ASCII
% value is either 10, 13 or 12
% skip all bytes whose ASCII value
% is either 10, 13 or 12
```

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

**STARTLM**

## MAKEVMFILE

**MAKEVMFILE** almacena datos, por lo general el contenido de un archivo, en la memoria como un archivo virtual. Use **MAKEVMFILE** en combinación con **XGFRESDEF** para incorporar los archivos externos usados por **RUN**, **SETLMFILE** y **SETDLFILE** en un archivo VIPP® PostScript independiente.

### Sintaxis

```
MAKEVMFILE data ..... %%EOD_XGF
```

### Donde:

**data .....** representa el contenido de un archivo.

**% %%EOD\_XGF** puede agregarse al final para indicar el final de los datos.

### Ejemplos

Este ejemplo muestra cómo incrustar un archivo PostScript en el archivo VIPP® independiente.

```
/doc1.ps MAKEVMFILE data ..... %%EOD_XGF XGFRESDEF
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**XGFRESDEF**

## MAKEVMFORM

**MAKEVMFORM** crea un procedimiento adecuado para su uso con los comandos **SETFORM** o **SETBFORM**. Utilice **MAKEVMFORM** para evitar el encapsulado entre llaves ( {.....} ). Este modo crea formularios VIPP® fuera de archivos PostScript generados por aplicaciones de procesamiento de documentos. Cuando se utiliza este formato, se evitan todas las limitaciones de encapsulado de llaves a excepción de las limitaciones de memoria.

### Sintaxis

```
MAKEVMFILE
data .....
%%EOD_XGF
```

### Donde:

<b>data .....</b>	representa el contenido de un archivo.
<b>% %EOD_XGF</b>	debe agregarse al final para indicar el final de los datos.

### Ejemplos

Este ejemplo muestra cómo crear un formulario de VIPP® a través de un archivo PostScript.

```
%! %%Title: form1.ps MAKEVMFORM PS code ..... %%EOD_XGF
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETFORM**, **SETBFORM**

## MAXICODE

**MAXICODE** crea e imprime un código de barras **MaxiCode** basado en la cadena especificada y los datos de los parámetros. No hay fuentes especiales obligatorias.



Nota: No se permiten **PAGEBRK** entre los códigos de barras de una secuencia de unión estructurada.

### Sintaxis

```
(msgdata) mode MAXICODE (msgdata) mode align MAXICODE (msgdata) mode [ posval totval ]
```

### Donde:

<b>(msgdata)</b>	es una cadena con los datos que se codificarán. Para obtener más información, consulte <a href="#">requisitos de formato msgdata</a>
<b>modo</b>	<p>define la estructura de los datos del código de barras y la corrección de errores en el símbolo. Hay dos modos obsoletos y cinco modos compatibles:</p> <p><b>0</b> obsoleto.</p> <p><b>1</b> obsoleto.</p> <p><b>2</b> crea un mensaje de transporte con una estructura estadounidense que el sector del transporte utiliza como símbolo de clasificación del destino.</p> <p><b>3</b> crea un mensaje de transporte con una estructura internacional que el sector del transporte utiliza como símbolo de clasificación del destino.</p> <p><b>4</b> crea un símbolo que codifica información con usos diferentes al del sector del transporte. El modo 4 codifica un máximo de 93 caracteres o 138 dígitos, pero contiene menos correcciones/detecciones de errores que el modo 5.</p> <p><b>5</b> crea un símbolo que codifica información con usos diferentes al del sector del transporte. El modo 5 codifica un máximo de 77 caracteres o 113 dígitos, pero contiene menos correcciones o detecciones de errores que el modo 4.</p> <p><b>6</b> crea un símbolo que codifica un mensaje que se utiliza para programar lectores de códigos de barras (escáneres) El modo 6 codifica un máximo de 93 caracteres o 138 dígitos.</p> <p>Los modos 2, 3, 4 y 6 usan la Corrección mejorada de errores (EEC) para el mensaje principal, y la corrección de errores estándar (SEC) para el mensaje secundario. El modo 5 utiliza CEE para los mensajes principales y secundarios.</p>

<b>alinear</b>	indica qué punto del código de barras se alinea en la posición de impresión secundaria, mediante estos valores:  <b>0</b> superior izquierda (predeterminado) <b>1</b> superior derecha <b>2</b> centro superior <b>10</b> inferior izquierda <b>11</b> inferior derecha <b>12</b> centro inferior <b>20</b> centro izquierda <b>21</b> centro derecha <b>22</b> centro centro
<b>[ posval totval ]</b>	es una matriz de unión estructurada opcional. La matriz se especifica si un número de códigos de barras MaxiCode superior a uno e inferior a ocho se va a unir en un formato estructurado. El valor posval es la posición del código de barras. El valor totval es el número total de códigos de barras unidos.

## REQUISITOS DE FORMATO MSGDATA

El formato msgdata depende del modo del código de barras como se explica más abajo.

### Todos los modos:

Si se especifica este encabezado opcional del mensaje, debe estar situado al principio de msgdata:

```
[ ] ><RS>01<GS>YY
```

(En este encabezado, YY es una representación del año en dos dígitos).

### Modos 2 y 3 solamente:

msgdata contiene un código postal (consulte el modo 2 y 3 para obtener más información), un código de país de 3 dígitos y una clase de servicio de 3 dígitos, en ese orden. El código postal sigue al encabezado del mensaje, si se especifica en msgdata. De lo contrario, el código postal puede colocarse al principio de msgdata.

### En una secuencia de agregación estructurada:

Los códigos de barras de una secuencia de unión estructurada contiene el mismo modo, código postal, código de país y clase de servicio.

### Modo 2 solamente:

El código postal puede tener de 2 a 9 dígitos. No se permiten guiones ni otros caracteres. Un código postal de 5 dígitos con el código de país 840 se puede rellenar con 4 ceros. Los demás códigos postales con una longitud de menos de 9 dígitos se pueden rellenar con ceros.

### Modo 3 solamente:

El código postal puede tener entre 2 y 6 caracteres, pero un código postal con menos de 6 caracteres se rellena con 6 caracteres iguales. Cada carácter puede ser un número o una letra mayúscula.

#### Modos 4, 5 y 6 solamente:

msgdata no contiene código postal, código de país ni clase de servicio.

#### En una secuencia de agregación estructurada:

Cada código de barras de una determinada secuencia de unión estructurada contiene el mismo modo.

#### Ejemplos

Este es un ejemplo de un código de barras de modo 2:

```
( [ \ ] ) > < RS > 01 < GS > 96152382802 < GS > 840 < GS > 001 < GS > 1Z00004951 < GS > UPSN  
< GS > 06X610 < GS > 159 < GS > 1234567 < GS > 1/1 < GS > < GS > Y < GS > 634 ALPHA DRIVE  
< GS > PITTSBURGH < GS > PA < RS > < EOT > ) 2 MAXICODE
```

#### Donde:

<GS>	es una secuencia de caracteres especiales para separar campos
<RS>	es una secuencia de caracteres especiales que indica el final del formato
<EOT>	es una secuencia de caracteres especiales que indica el final de la transmisión
<FS>	es una secuencia de caracteres especiales para separar los números de dirección principales y secundarios que no se muestra en el ejemplo anterior.



Nota: Son necesarios los corchetes en las secuencias de caracteres especiales para diferenciarlos de caracteres en mayúscula normales.

[ ] > < RS > 01 < GS > 96	encabezado del mensaje, datos de transporte y encabezado de formato; 96 indica los últimos 2 dígitos del año
152382802	Código postal
840	Código de país, puede tener 3 dígitos
001	Clase de servicio, puede tener 3 dígitos
1Z00004951	Número de seguimiento
UPSN	SCAC
06X610	Número de cuenta UPS
159	Día de recogida
1234567	Número de identificador de envío
1/1	Paquete n/x
Y	Validación de la dirección
634 ALPHA DRIVE	Calle de la dirección de envío

**PITTSBURGH** Ciudad de envío

**PA** Estado del envío

Este es un ejemplo de un código de barras de modo 3:

```
([\])><RS>01<GS>V6J5G3<GS>124<GS>001<GS>1Z00004951<GS>UPSN
<GS>06X610<GS>159<GS>1234567<GS>1/1<GS><GS>Y<GS>1090 W PENDER ST
<GS>VANCOUVER<GS>BC<RS><EOT>) 3 MAXICODE
```

En el que la estructura de msgdata para los códigos de barras de modo 2 y 3 es muy similar. Solo los requisitos de código postal son distintos.

Este es un ejemplo de códigos de barras de modo 2 en una secuencia de unión estructurada:

```
100 700 MOVETO
([\])><RS>01<GS>96152382802<GS>840<GS>001<GS>1Z00004951<GS>UPSN<GS>
06X610<GS>159<GS>1234567<GS>1/1<GS><GS>Y<GS>634 ALPHA DRIVE<GS>
PITTSBURGH<GS>PA<RS><EOT>) 2 [1 2] MAXICODE
100 1100 MOVETO
([\])><RS>01<GS>96152382802<GS>840<GS>001<GS>05<GS>400123456789<RS>
<EOT>) 2 [2 2] MAXICODE
```

Este es un ejemplo de códigos de barras de modo 5 en una secuencia de unión estructurada:

```
100 700 MOVETO
(This is message 1 of a structured append sequence) 5 [1 8] MAXICODE
100 1100 MOVETO
(This is message 2 of a structured append sequence) 5 [2 8] MAXICODE
100 1500 MOVETO
(This is message 3 of a structured append sequence) 5 [3 8] MAXICODE
100 1900 MOVETO
(This is message 4 of a structured append sequence) 5 [4 8] MAXICODE
500 700 MOVETO
(This is message 5 of a structured append sequence) 5 [5 8] MAXICODE
500 1100 MOVETO
(This is message 6 of a structured append sequence) 5 [6 8] MAXICODE
500 1500 MOVETO
(This is message 7 of a structured append sequence) 5 [7 8] MAXICODE
500 1900 MOVETO
(This is message 8 of a structured append sequence) 5 [8 8] MAXICODE
```

Se requiere un comando **MOVETO** entre comandos **MAXICODE** de una secuencia de unión estructurada para evitar la superposición de imágenes.

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

- **AZTEC**
- **DATAMATRIX**
- **PDF417**
- **QRCODE**
- **MOVEH**
- **MOVEHR**

- **MOVETO**



## MOVEH

**MOVEH** establece la posición de impresión horizontal secundaria (PP) con puntos iniciales opcionales.

### Sintaxis

`hpos MOVEH`

`[ hpos GEPkey ] MOVEH`

### Donde:

<b>hpos</b>	es la posición horizontal secundaria en unidades actuales. Cuando se utiliza <b>MOVEH</b> , hpos se mide desde el borde izquierdo de la página.
<b>GEPkey</b>	es la <b>GEPkey</b> utilizada para definir la apariencia de los puntos de entrada. Si hay un valor <b>GEPkey</b> , se dibuja una línea desde la posición de impresión horizontal secundaria actual hasta la izquierda del siguiente elemento de texto impreso con un comando <b>SHx</b> .

### Ejemplos

Este ejemplo imprime:

`Introduction.1`

`100 MOVEH (Introduction) SH [1500 D1] MOVEH (1) SH`

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **MOVETO**
- **MOVEHR**
- **NL**
- **SETLKF**

## MOVEHR

El comando **MOVEHR** configura la posición de impresión horizontal secundaria en función de la última posición principal horizontal definida en el último **MOVETO** o la trama actual con puntos iniciales opcionales.

### Sintaxis

```
hrpos MOVEHR
```

```
[ hpos GEPkey ] MOVEHR
```

### Donde:

#### hrpos

es la distancia de desplazamiento a lo largo del eje horizontal desde la última posición principal horizontal definida en el último **MOVETO** o la trama actual. Si no se especifica, la PP inicial es 0,0. hrpos también puede ser una cadena o una variable. Esto permite utilizar variables DBM como operandos hrpos.

El origen (0,0) se encuentra en la esquina inferior izquierda de la página, o en la esquina superior izquierda si se especifica **ORITL**.

#### GEPkey

es la **GEPkey** utilizada para definir la apariencia de los puntos de entrada. Si hay un valor **GEPkey**, se dibuja una línea desde la posición de impresión horizontal secundaria actual hasta la izquierda del siguiente elemento de texto impreso con un comando **SHx**.

### Ejemplos

Este ejemplo imprime:

```
Introduction ..... 1
```

```
(Introduction) SH [1500 D1] MOVEHR (1) SH
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **MOVEH**
- **MOVETO**
- **NL**
- **ORIBL**
- **ORITL**
- **SETLKF**

## MOVETO

El comando **MOVETO** establece las posiciones de impresión principales y secundarias. En el siguiente ejemplo de sintaxis, **MOVETO** establece las PP principales y secundarias como *hpos* y *vpos*.

El comando **MOVETO** hace que la página se inicialice.

### Sintaxis

*hpos vpos MOVETO*

### Donde:

**hpos**                                      está en el eje horizontal.

**vpos**                                      está en el eje vertical.

Cuando no se especifica, la posición de impresión inicial es 0,0. Los valores *vpos* y *hpos* también pueden ser cadenas o variables. Esto permite utilizar variables **DBM** como operandos *vpos* y *hpos*.

El origen 0,0 se encuentra en la esquina inferior izquierda de la página, o en la esquina superior izquierda si se especifica **ORITL**.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **MOVEH**
- **NL**
- **ORIBL**
- **ORITL**
- **SETLKF**

## MSPP\_on

En modo de varias en 1, el comando **MSPP\_on** permite colocar a **ENDOFSET**, **ENDOFRUN** y **JOG\_on/JOG\_off** al principio de la última página física, en vez de empezar al final de la última página lógica.

De manera predeterminada, VI Compose hace que la última página lógica en una nueva página física ejecute el comando **ENDOFx**. Si bien el juego se mantiene, esto puede producir una división no deseada de la página. **MSPP\_on** permite evitar esta circunstancia.

### Sintaxis

MSPP\_on



Nota: Al usar **MSPP\_on**, es la responsabilidad de la aplicación del cliente insertar comandos en la posición que identifica la primera página lógica de la última página física del conjunto.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **ENDOFRUN**
- **ENDOFSET**
- **TWOUP**
- **SETMULTIUP**
- **JOG\_on** y **JOG\_off**

## MUL

**MUL** multiplica una variable numérica definida por **SETVAR** o una variable XML por un número.

### Sintaxis

```
/VARname number MUL
```

```
/^XMLname number MUL
```

### Donde:

<b>/VARname</b>	hace referencia a una variable numérica previamente inicializada por <b>SETVAR</b> .
<b>/^XMLname</b>	hace referencia a una variable de XML. En general, las variables XML no necesitan inicializarse de forma explícita. VI Compose inicializa todas las variables XML en una cadena vacía, que equivale a una cadena numérica igual a cero.
<b>número</b>	es el número por el que se multiplica la variable. Puede ser un número entero, o real, o una cadena numérica. Cuando se trata de un gran número, es obligatorio contar con una cadena numérica.

Las cadenas numéricas admiten números grandes de hasta 40 dígitos, 25 dígitos para la parte entera y 15 dígitos para la parte decimal. En una cadena numérica, el signo negativo y el delimitador decimal se definen mediante los parámetros `/DecimalPoint` y `/NSign` y se pueden producir en cualquier lugar de la cadena.

Es obligatorio configurar estos parámetros con los valores correspondientes para garantizar unos resultados precisos. Los valores predeterminados se definen en el archivo `/usr/xgf/src/xgf.def`. Se omiten los caracteres en la cadena numérica que no sean estos dos más los dígitos 0-9.

La longitud inicial de la cadena definida por **SETVAR** se amplía automáticamente hasta 40 dígitos cuando es necesario.

Los números reales y enteros deben utilizarse solo para los valores pequeños  $\leq 99999$ . Por ejemplo, para la implementación de un contador. El delimitador decimal, si lo hubiera, siempre es el punto (.). El signo negativo, si lo hubiera, siempre es el signo menos (-) y puede ser el primer carácter.

### Ejemplos

```
/VAR.CNT1 123 SETVAR /VAR.CNT1 12 MUL /VAR.CNT1 -3 MUL /VAR_SUM (1,234,890,566,00-) SET
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **ADD**
- **SUB**
- **DIV**
- **SETVAR**

- ++ y –

## NEWBACK

En el modo a 2 caras, **NEWBACK** hace que la página actual imprima en la siguiente contraportada disponible de una hoja. Puede codificar **NEWBACK** después de un delimitador de página, como **PAGEBRK**, avance de formulario o salto para garantizar la canalización.

### Sintaxis

NEWBACK

En el modo Varias en 1, también se realiza una operación **NEWSIDE**.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **DUPLEX\_on**
- **NEWFRONT**
- **NEWSIDE**
- **PAGEBRK**
- **SETMULTIUP**
- **TWOUP**

## NEWFRAME

Utilice el comando **NEWFRAME** para comenzar a colocar elementos posteriores en el marco disponible siguiente. El uso de este comando puede producir una transición de página.

### Sintaxis

NEWFRAME

### Modos

Este comando es aplicable en modos nativo y de base de datos.

### Comandos relacionados

FRCOUNT, GOTOFRAME, SETLKF



## NEWFRONT

**NEWFRONT** hace que la página actual imprima en la parte delantera de una nueva hoja. Puede codificar **NEWFRONT** después de un delimitador de página como **PAGEBRK**, **Form Feed** o **Skip** para garantizar la canalización.

### Sintaxis

NEWFRONT

En el modo Varias en 1, también se realiza una operación **NEWSIDE**.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **DUPLEX\_on**
- **NEWBACK**
- **NEWSIDE**
- **PAGEBRK**
- **SETMULTIUP**
- **TWOUP**

## NEWGROUP

**NEWGROUP** es un comando secundario **RPE** que permite crear grupos **RPE** en una definición de **RPE**. El comando se usa si un flujo de datos usa prefijos que no siguen la regla de último dígito para la agrupación de líneas. Todas las definiciones de **RPEKEY** que pertenecen a un grupo se deben agrupar y deben ser precedidas por un comando **NEWGROUP**.

Consulte [Información de comando RPE](#) y otros comandos de **RPE** relacionados.

### Ejemplos

```
6 SETRPEPREFIX
10 BEGINRPE
  NEWGROUP
    /LINECR RPEKEY [ .... ]
    /LINEDB RPEKEY [ .... ]
  NEWGROUP
    /NAME01 RPEKEY [ .... ]
    /NAME02 RPEKEY [ .... ]
    /ADRE01 RPEKEY [ .... ]
    /ADRE01 RPEKEY [ .... ]
    /ZPCITY RPEKEY [ .... ]
  ENDRPE
```

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

**BEGINRPE**, **ENDRPE**

## NEWPOS

**NEWPOS**, un comando secundario de **RPE**, hace que se mantenga en memoria un nuevo conjunto independiente de valores de posición de impresión. Se usa si se requiere una nueva posición de impresión para que un grupo de entradas de **RPE** imprima el mismo campo en distintas ubicaciones en la página.

### Ejemplos

Además, con esta nueva sintaxis, los registros del mismo grupo de RPE pueden no ser consecutivos en el archivo de datos. Las sintaxis nuevas y antiguas son exclusivas de la definición de **RPE** especificada configurada a través de un comando **RPEKEY** o **FROMLINE**. Sin embargo, las definiciones de **RPE** con ambas sintaxis se pueden mezclar en bibliotecas de RPE compuestas de diversos comandos **RPEKEY** o **FROMLINE** situados entre **BEGINRPE** y **ENDRPE**.

Para obtener más información, consulte [Información de comando RPE](#) y otros comandos RPE relacionados.

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

**RPEKEY**

## NEWSIDE

En el modo Varias en 1, **NEWSIDE** hace que la página lógica actual imprima en la siguiente página física. Puede codificar **NEWSIDE** después de un delimitador de página como **PAGEBRK**, **Form Feed** o **Skip** para garantizar la canalización.

### Sintaxis

NEWSIDE

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **NEWBACK**
- **NEWFRONT**
- **PAGEBRK**
- **SETMULTIUP**
- **TWOUP**

## NEWSTACK

Este comando solo está disponible si se utiliza el modo genérico de **ZSORT**. Consulte **ZSORT** en este documento.

Este comando finaliza la pila **ZSORT** actual con todos los registros procesados hasta el momento y comienza una nueva pila. Pretende ser usado en una declaración condicional para iniciar una nueva pila cuando la condición es verdadera.

**NEWSTACK** puede colocarse antes de la inicialización de la página antes de cualquier **MOVETO** o marca en la página.

### Sintaxis:

NEWSTACK

### Ejemplo:

```
IF Fieldx NEXT_FieldX ne { NEWSTACK } ENDIF
```

### Modos

Este comando es aplicable en todos los modos al combinarse con un **ZSORT** genérico.

### Comandos relacionados

**ZSORT**

## NL

El comando **NL** restablece la posición de impresión horizontal (PP) y reenvía la posición de impresión vertical.

En el ejemplo de sintaxis más abajo, NL restablece las posiciones de impresión horizontales principales y secundarias con los últimos valores especificados por **MOVETO** y reenvía la posición de impresión vertical de los valores **SETLSP** o **LSPval**.

### Sintaxis

NL

LSPval NL

### Donde:

<b>LSPval</b>	es el valor (en unidades actuales) que puede agregarse a la posición vertical actual. Si no está presente, se utiliza el valor prefijado de <b>SETLSP</b> .
---------------	---

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**MOVEH,MOVETO**

## NMP\_off

**NMP\_off** desactiva el procesamiento de registros de prefijo de modo nativo (NMP).

### Sintaxis

`NMP_off`

### Modos

○Este comando es aplicable en modo de línea y de base de datos.

### Comandos relacionados

`% %XGF, SETNMP`

## OMRINIT

**OMRINIT** inicializa el procesamiento de código **OMR**. Un código de **OMR** es una secuencia de pequeñas barras verticales utilizada para dar un sistema de correo automatizado. La altura, el grosor, el espaciado y la configuración de las barras se pueden definir mediante el parámetro **/OMRconfig**.

Al inicializarse, el código **OMR** se puede imprimir en cada página mediante el comando **OMRSHOW**.

Este comando puede codificarse en un trabajo de modo nativo, en un **JDT** o en un **NMP**. Se puede invocar una sola vez en un trabajo, o varias veces si la configuración debe modificarse durante el trabajo, por ejemplo, entre conjuntos.

Es preciso codificar **OMRINIT** antes de la página se inicialice, antes de la aparición de comandos de marcaje como **MOVETO**.

### Sintaxis

```
/plex feed_count fold_count annexes OMRINIT
```

#### Donde:

**/plex**

puede ser uno de los siguientes:

**/F** El código de **OMR** está pensado para las páginas frontales

**/B** El código de **OMR** está pensado para las páginas posteriores

**feed\_count**

puede ser uno de los siguientes:

**1–255** alimentación de sobres después del número de páginas

**0** no hay alimentación para las páginas siguientes

La barra de alimentación se coloca en la posición I o i en la cadena de configuración.

**fold\_count**

puede ser uno de los siguientes:

**1–255** pliegue intermedio después del número de páginas

**0** no hay pliegues intermedios de las páginas siguientes

La barra de pliegue se coloca en la posición F en la cadena de configuración.



**anexos**

es un entero cuyo componente al cuadrado representa una barra para desencadenar una inserción o una acción lateral. Ejemplo:

**+1** insertar anexo 1

**+2** insertar anexo 2

**+4** insertar anexo 3

**+8** no cerrar sobres

**+16** etc.



Nota: Bar +1 se coloca en la posición  $\times$  o  $\times$  en la cadena de configuración, seguido por el resto de barras. El número máximo de barras depende de la configuración del sistema postal.

**Ejemplos**

El ejemplo inicializa el código **OMR** que se imprimirá en la página frontal, introducirá 7 páginas en el sobre, realizará un pliegue cada 2 páginas e insertará los anexos 1 y 3.

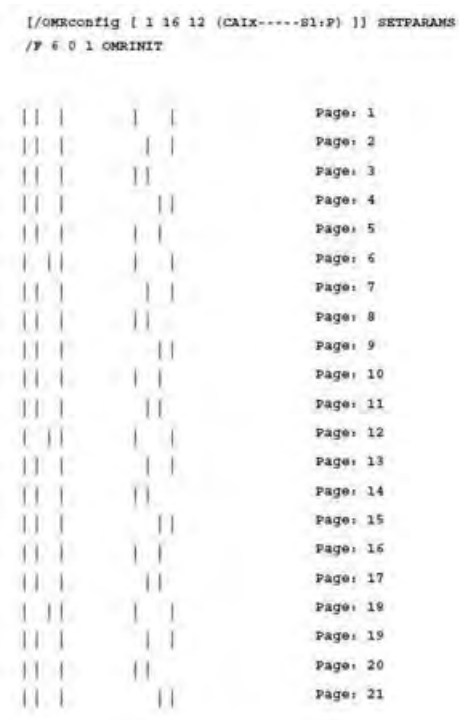
/F 7 2 5 OMRINIT



Nota: Los parámetros `feed_count` y `fold_count` indican un número de páginas de datos. En formularios de contraportadas a 1 o 2 caras, este recuento equivale a la cantidad de hojas. En el modo a 2 caras sin formulario de contraportada, esto equivale al número de caras impresas.

/B es obligatorio si se realiza una llamada a **OMRSHOW** en un formulario de contraportada.

El gráfico más abajo es un ejemplo de las opciones de /OMRconfig y **OMRINIT** seguidas por una recopilación de códigos **OMR** como se imprimirían en 21 páginas consecutivas según estas opciones:



### PARÁMETRO OMRCONFIG

Cada sistema de correo electrónico requiere una configuración de OMR diferente y que se describe en su documentación. Este parámetro puede utilizarse para definir la configuración del código OMR. Dado que un determinado sitio normalmente tiene solo un tipo de sistema de correo, se puede agregar a xgf.def para que solo sea definido una vez. También se puede colocar en el propio trabajo (JDT, XJT, etc.)

#### Sintaxis

```
[/OMRconfig [ width height spacing (config_string) ]] SETPARAMS
```

#### Donde:

- anchura** es la anchura de una barra en puntos.
- altura** es la altura de una barra en puntos.

<b>espaciado</b>	es el espacio entre las barras en puntos.
<b>config_string</b>	es una cadena que puede incluir los símbolos siguientes: <b>C</b> barra de control (presente en cada código OMR) <b>I</b> insertar en la última página <b>i</b> insertar en la primera página <b>A</b> alimentación (cuando no hay inserción) <b>P</b> paridad impar <b>p</b> paridad par <b>Sab</b> Un número de secuencia sucesiva (3 posiciones) <b>Donde:</b> <b>S</b> fijo debe ser S <b>Ningún número entre 0 y 7</b> <b>1</b> número entre 1 y 7 <b>7</b> números entre 7 y 1 <b>b</b> - no restaurar <b>+</b> restaurar en la primera página <b>:</b> restaurar en la última página <b>F</b> plegar <b>L</b> última página <b>X</b> primera barra para anexos (en la página de inserción solamente) <b>x</b> primera barra para los anexos (en todas las páginas) <b>-</b> espacio o marcador vacío

A excepción de C y -, los símbolos pueden utilizarse una sola vez en una cadena de configuración. Los pares P/p, I/i y X/x son mutuamente excluyentes.

### Ejemplos

Este ejemplo define un código de barras 16, con posiciones de inserción, alimentación y pliegue, paridad impar, secuencias sucesivas de 1 a 7 sin restablecimiento y hasta 6 anexos (de +1 a +32).

```
[ /OMRconfig 1 16 12 (C-CPAIS1-FX—) ] SETPARAMS
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

OMRSHOW,DUPLEX\_on,DUPLEX\_off

## OMRSHOW

**OMRSHOW** imprime un código OMR previamente inicializado por **OMRINIT** en la posición de impresión principal. Se puede incluir en una definición de formulario.

### Sintaxis

OMRSHOW

```
{ imaging proc } /BC OMRSHOW
```

### Donde:

La primera sintaxis imprime una secuencia de barras de OMR.

La segunda sintaxis imprime la secuencia de OMR como una serie de ceros y unos, posiblemente mediante una fuente de código de barras o un código de barras 2D.

### Ejemplos

Para imprimir una secuencia de barras:

```
{ 1450 3330 MOVETO OMRSHOW } SETFORM
```

Para imprimir la secuencia de OMR como un código de barras 2OF5:

```
{ /2of5_font size SETFONT { 2OF5 SH } /BC OMRSHOW } SETFORM
```

Para imprimir la secuencia OMR como un símbolo de Datamatrix:

```
{ { DATAMATRIX } /BC OMRSHOW } SETFORM
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**OMRINIT**, **SETFORM**, **SETBFORM**

## ONEUP

**ONEUP** restablece el modo de varias en 1. El modo Varias en 1 permite la impresión de una sola página lógica en cada página física. **ONEUP** es el valor prefijado.

### Sintaxis

ONEUP

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

SETMULTIUP,TWOUP

## ORIBL

**ORIBL** establece el origen de todas las coordenadas, excepto **RPE**, en la esquina inferior izquierda de la página. Este comando puede ser el primer comando en un archivo VIPP®. **ORIBL** es el valor prefijado.

### Sintaxis

ORIBL

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

ORITL

## ORITL

**ORITL** establece el origen de todas las coordenadas en la esquina superior izquierda de la página. Debe ser el primer comando en un archivo VIPP®.

### Sintaxis

ORITL

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

ORIBL



## OVERPRINT\_on

**OVERPRINT\_on** permite el procesamiento de la sobreimpresión en modo de línea y en modo de base de datos. Use este comando para imprimir líneas de datos entre sí si las líneas finalizan con un solo retorno de carro (CR) en vez de con **LF** o **CR/LF**. Cuando se colocan antes de **STARTDBM** en combinación con **QSTRIP\_on LF** o **CR/LF**, pueden ser integradas en campos entrecomillados. A continuación, se pueden imprimir los campos mediante **SHP** con la opción + 20.

### Sintaxis

OVERPRINT\_on



Importante: Utilice este comando solamente cuando sea necesario ya que puede afectar al rendimiento.

### Modos

Este comando es aplicable en modo de línea y modo de base de datos.

### Comandos relacionados

**STARTLM**

## PAGEBRK

El comando **PAGEBRK** imprime la página actual y restaura las posiciones de impresión primarias y secundarias (PP) a 0,0. Este comando es el único final de la marca de página en el modo nativo. En el modo de línea, el final de la página también se produce cuando se encuentra una omisión de canales o un avance del formulario. Para obtener más información, consulte *Flujos de datos de VIPP®* en la *Guía del usuario de FreeFlow VI Compose*.

En el modo Varias en 1, **PAGEBRK** pasa a la siguiente página lógica. Una página física solo se imprime si se llega a la última página lógica, a menos que se usen **NEWFRONT**, **NEWBACK**, **NEWFRAME** o **NEWSIDE**.

### Sintaxis

PAGEBRK

Cuando nada se procesa en la página, **PAGEBRK** no produce una página en blanco. Para producir una página en blanco debe utilizar, como mínimo, **NL PAGEBRK**.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **BCALL**
- **ENDIMP**
- **NEWBACK**
- **NEWFRAME**
- **NEWFRONT**
- **NEWSIDE**
- **SETLKF**

## PAGERANGE

**PAGERANGE** especifica el intervalo de páginas que se imprimirá del trabajo actual.



Nota: Ejecute este comando al principio del trabajo, antes de que las páginas sean procesadas.

### Sintaxis

```
startpage stoppage PAGERANGE
```

### Donde:

**startpage** es el número de la página inicial.

**stoppage** es el número de la página final.

### Ejemplos

Este ejemplo solo imprime de la página 50 a la 100

```
50 100 PAGERANGE
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETPARAMS**

## PDF417

**PDF417** crea e imprime un código de barras **PDF417** basado en las cadenas especificadas y los datos de los parámetros. No hay fuentes especiales obligatorias.

### Sintaxis

```
[/TC (string) /BC (string) /NC (string)] PDF417
[/TC (string) /BC (string) /NC (string)] align PDF417
[/TC (string) /BC (string) /NC (string)] [/Rows RValue /Columns CValue
/ARatio [HValue WValue] /ELevel EValue ] PDF417
[/TC (string) /BC (string) /NC (string)] [/Rows RValue /Columns CValue
/ARatio [HValue WValue] /ELevel EValue ] align PDF417
[/TC (string) ...] scale rotate align PDF417
[/TC (string) ...] [/Rows RValue...]scale rotate align PDF417
[/TC (string) ...] [fit-in-width] rotate align PDF417
[/TC (string) ...] [/Rows RValue...] [fit-in-width] rotate align PDF417
[/TC (string) /BC (string) /NC (string)] [/Rows RValue /Columns CValue /ARatio
[HValue WValue] /ELevel EValue /Truncate TValue] PDF417
[/TC (string) /BC (string) /NC (string)] [/Rows RValue /Columns CValue /ARatio
[HValue WValue] /ELevel EValue /Truncate TValue] align PDF417
```

### Donde:

**/TC** es el modo de compactación de texto. Los caracteres permitidos son caracteres imprimibles con los valores decimales 32 - 127 y los siguientes caracteres de control:

**(\n)** nueva línea o avance de línea

**(\r)** retorno de carro

**(\t)** ficha horizontal

**(\\)** barra invertida



Nota: Para utilizar caracteres fuera del intervalo 32-127, utilice el modo /BC. El tamaño del código de barras es más grande y permite toda la gama de códigos.

**/BC** es el modo de compactación de bytes. Permite la codificación de cualquier valor de 8 bits, de 0 a 255.

**/NC** es el modo de compactación numérico. Se debe utilizar para codificar cadenas largas de dígitos numéricos consecutivos. Aunque se puede realizar una llamada al modo de compactación numérica para cualquier longitud de dígitos, Xerox recomienda que se usa cuando hay 13 o más dígitos consecutivos.

**(cadena)** son datos de texto, bytes o datos numéricos en el modo de compactación anterior.

**alinear** indica qué punto del código de barras se alinea en la posición de impresión secundaria, mediante estos valores:

0 superior izquierda (predeterminado)

1 superior derecha

2 centro superior

10 inferior izquierda

11 inferior derecha

12 centro inferior

20 centro izquierda

21 centro derecha

22 centro centro

**/Rows** es el número de filas en el código de barras.

**RValue** debe tener un valor entre 3 y 90.

**/Columns** es el número de columnas de datos en el código de barras.

**CValue** debe tener un valor entre 1 y 30.

**/ARatio** es la proporción del código de barras.



Nota: El **/ARatio** predefinido es [1 1]. Un **/ARatio** de [1 1] no produce un código de barras cuadrado. El número de columnas por código de barras es **WValue** más cuatro, para las columnas de inicio y final, más los indicadores de fila derecha e izquierda.

El **/ARatio** necesario para crear un código de barras cuadrado depende de varios factores, como los valores pasados a los parámetros **SETPARAMS**, **DotsPerModule** y **RowHeight**, el valor de corrección/detección, el número de filas o columnas, y el número de caracteres de cadena. La mejor manera de crear un determinado tamaño de código de barras es a través de la realización de pruebas y errores. Si un **/ARatio** de 1 a 1 crea un código de barras que es más ancho que alto, como sucede si se utilizan los parámetros predefinidos, para crear un código de barras cuadrado, aumente **/HValue** o reduzca **/WValue**, o ambos. Por ejemplo: **/ARatio [5 1]**.

**HValue** es la altura de las filas o el número de filas en el código de barras.

**WValue** es la anchura o el número de columnas de datos.

**/ELevel** es el nivel de corrección/detección de errores.

**EValue** debe estar en el rango entre 0 y 8, donde 0 es el valor mínimo y 8 es la cantidad máxima de corrección.

**/Truncate** especifica el código de barras **PDF417** truncado.

**/TValue** es el estado del código de barras **PDF417** truncado:

**true** = activado

**false** = desactivado

/Rows, /Columns y /ARatio son opcionales. No se puede especificar /Rows y /Columns en el mismo comando. Cuando se especifican ambos, solo se aceptan /Columns y /Rows se actualizan. Si ni /Rows ni /Columns se especifican, ambos se determinarán automáticamente según su valor predefinido o el especificado por /ARatio.

/ELevel también es opcional. El nivel de error predefinido será un mínimo de dos y un máximo de cinco, en función del nivel de codificación de datos.

Un código de barras **PDF417** puede codificar aproximadamente 1850 caracteres de texto, 1110 bytes o 2710 dígitos, con un nivel de error de 0. Para un nivel de error de 8, los límites son aproximadamente 830 caracteres de texto, 498 bytes o 1215 dígitos.

El código de barras **PDF417** también admite dos parámetros nuevos en **SETPARAMS**:

```
[ /DotsPerModule integer
/RowHeight integer
] SETPARAMS
```

Donde:

<b>/DotsPerModule</b>	es el número de puntos por barra o espacio en el código de barras. El valor predeterminado es 3.
<b>/RowHeight</b>	es la altura de una fila en el código de barras. El valor predefinido es 4 si el nivel de errores es bajo dado el nivel de datos codificado, y 3 si el nivel de errores es correcto para la cantidad de datos codificada.

Estos parámetros son modificables para admitir una amplia gama de impresoras y escáneres. Sin embargo, Xerox recomienda que los parámetros mantengan sus valores predefinidos a menos que los problemas del escáner o la resolución de la impresora requieran barras mayores.

### Ejemplos

```
[ /BC <01 02 03 04 05 06 07 08 09> /TC (Test) /NC (00246812345678) ] PDF417
[ /BC (PDF417 symbology) ] [ /ARatio [1 2] /ELevel 1 ] PDF417
[ /NC (1234567890123456789012345) ] [ /Rows 3 ] PDF417
[ /TC (John Doe\n1405 Ocean Drive\nEl Segundo, CA 90245) ] [ /Columns 3 ] PDF417
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **AZTEC**
- **DATAMATRIX**
- **MAXICODE**
- **QRCODE**
- **MOVEH**
- **MOVEHR**
- **MOVETO**

## PDFBOUND

**PDFBOUND** activa la creación de cuadros opcionales de límite de página en un PDF de salida creado por un trabajo VIPP®. Este comando no altera el contenido de PDF aparte de agregar los cuadros de límite adicionales en la página. Las aplicaciones que realicen la reproducción o el procesamiento descendiente del PDF en el flujo de trabajo pueden procesarlo u omitirlo. Si la salida del trabajo VIPP® no es un archivo PDF, el comando se ignora y no tendrá efecto.

**PDFBOUND** debe colocarse antes de la inicialización de la página antes de cualquier **MOVETO** o marca en la página.

### Sintaxis:

```
/xxxBox [top bottom left right] PDFBOUND
```

### Donde:

**/xxxBox**

es una de:

- /CropBox
- /BleedBox
- /TrimBox
- /ArtBox

**superior, inferior, izquierda y derecha**

son valores de desplazamiento al centro de la página (en unidades actuales) desde MediaBox, según lo definido por **SETPAGESIZE**.

### Ejemplo:

```
/CropBox [50 50 50 50] PDFBOUND
```

### Modo

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETPAGESIZE**

## PDFDEST

**PDFDEST** define un destino con nombre. Se puede hacer referencia a un destino con nombre en una definición de PIF de tipo DEST o XDEST.

### Sintaxis

```
/destname PDFDEST
```

```
/destname [ pagenum view ] PDFDEST
```

### Donde:

<b>destname</b>	es el nombre de destino (cadena alfanumérica).
<b>pagenum</b>	es el número de página del destino empezando por 1. El predeterminado es el número de página actual. Puede ser una variable.
<b>ver</b>	define cómo ajustar la vista para el destino de página. Consulte <b>SETPIF</b> para obtener una lista de posibles valores.

### Ejemplos

Este ejemplo muestra cómo configurar un enlace a un destino (/NOTES) definido más adelante en el trabajo:

```
[ /DEST /NOTES [/Fit] ] SETPIF 100 500 MOVETO (Go to Notes) 2 SHmf .... PAGEBRK .... /N
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **BOOKMARK**
- **SETPIF**
- **INDEXPIF**
- **PDFOPEN**
- **PDFINFO**



## PDFFORMOCG

**PDFFORMOCG** permite el contenido opcional del PDF a nivel de formulario.

El contenido opcional de PDF es la funcionalidad proporcionada por algunos visualizadores de PDF, como Adobe Acrobat y Reader, de mostrar u ocultar algunas partes del documento PDF cuando se visualiza o se imprime. Estos elementos aparecen en la barra de herramientas Capas cuando se abre un archivo PDF.

**PDFFORMOCG** se usa para asignar cada **planenumber** de formulario VIPP® con un nombre de capa de PDF, para que pueda aparecer en la barra de herramientas Capa. Un uso habitual consiste en permitir a un formulario de fondo figurar en el PDF, pero desactivarse si el documento se imprime en papel preimpreso.

### Sintaxis

```
[ [ (layerName1) istate1 ] ... [ (layerNameN) istateN ] ] PDFFORMOCG
```

### Donde:

<b>[ (layerNameX) istateX ]</b>	es el nombre y el estado inicial de cada número de plano del formulario. Se puede sustituir por null si la capa no está sujeta a contenido opcional.
<b>(LayerNameX)</b>	es una cadena que contiene el nombre de la capa
<b>istateX</b>	es un valor booleano que indica el estado inicial de la capa al abrir el documento: <ul style="list-style-type: none"> <li>- verdadero visible</li> <li>- falso oculto</li> </ul>

### Ejemplos

```
Asigne una capa opcional a 2 de 3 planos de formulario: (bvr.frm) 0 SETFORM (bill.frm)
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

PDFOCG, SETFORM, SETMAXBFORM

## PDFINFO

**PDFINFO** llena la sección Resumen del documento del archivo PDF con información.

### Sintaxis

```
[ /Author (Autor del documento) /Creator (Creador del documento) /Title (Título del documento)
```

### Donde:

<b>Author of the document</b>	contiene el nombre del autor.
<b>Creator of the document</b>	contiene el nombre del creador del documento.
<b>Title of the document</b>	contiene el título del documento.
<b>Subject of the document</b>	contiene el sujeto del documento.
<b>List of Keywords</b>	contiene las palabras clave del documento.



Nota: Los datos de un solo byte pueden cifrarse mediante ISO-8859-1. Los datos de varios bytes pueden cifrarse mediante UTF8. VI Compose los puede convertir automáticamente en UTF16 para insertarlos en el PDF porque esta es la única codificación de varios bytes admitida por el formato PDF. Para activar la conversión de datos UTF8 a UTF16, la fuente actual seleccionada por **SETFONT** o **INDEXFONT** puede tener una codificación UTF8.

### Ejemplos

```
[ /Author (John Smith) /Creator () /Title (VIPP PIF Samples) /Subject (Samples of VIPP)
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**PDFOPEN**

## PDFOCG

**PDFOCG** activa el contenido opcional del PDF a nivel de elemento.

El contenido opcional de PDF es la funcionalidad proporcionada por algunos visualizadores de PDF, como Adobe Acrobat y Reader, de mostrar u ocultar algunas partes del documento PDF cuando se visualiza o se imprime. Estos elementos aparecen en la barra de herramientas Capas cuando se abre un archivo PDF.

**PDFOCG** se usa para encapsular porciones del código VIPP® que contienen los comandos de marca y asignarles un nombre, por lo que aparecen en la barra de herramientas Capa.

### Sintaxis

```
(LayerName) istate PDFOCG % begin optional content group
null PDFOCG % end optional content group
```

### Donde:

<b>(LayerName)</b>	es una cadena que contiene el nombre de la capa
<b>istate</b>	es un valor booleano que indica el estado inicial de la capa al abrir el documento:
	- verdadero visible
	- falso oculto

### Ejemplos

```
Place the logo in a specific optional layer: (Logo) true PDFOCG (xslogo.tif) .4 0 1 ICA
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**PDFFORMOCG**

## PDFOPEN

**PDFOPEN** define cómo se puede abrir un documento PDF. Cuando se asocia con un destino PIF también se define el destino que se utilizará cuando se abre el documento.

### Sintaxis

```
/openmode PDFOPEN
```

### Donde:

#### openmode

selecciona el modo que se utiliza cuando se abre el documento. puede ser uno de los siguientes:

- **/UseOutlines** abrir y mostrar los marcadores
- **/UseThumbs** abrir y mostrar imágenes en miniatura
- **/UseNone** no abrir ni mostrar nada de lo anterior (predeterminado)
- **/FullScreen** abrir en el modo de pantalla completa

### Ejemplos

Este ejemplo muestra cómo utilizar **PDFOPEN** para mostrar marcadores cuando se abre el archivo PDF:

```
/UseOutlines PDFOPEN
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

PDFINFO

## PORT

**PORT** configura la orientación vertical. La orientación vertical indica que el borde corto de la hoja está en el eje horizontal (X) y el borde largo está en el eje vertical (Y). Esta es la opción prefijada.

### Sintaxis

PORT

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- ILAND
- IPORT
- LAND
- SETPAGESIZE

## PRECACHE

**PRECACHE** permite el almacenamiento previo en caché de los recursos. Solo es eficaz en intérpretes de PostScript que funcionan de acuerdo a las especificaciones de Xerox para el almacenamiento previo en caché. Si desea obtener más información, póngase en contacto con un representante de Xerox.

Los recursos almacenados previamente en caché se crean en el almacenamiento externo del controlador de la impresora. **PRECACHE** no produce ninguna marca en la página. Solo las llamadas siguientes al recurso, posiblemente en trabajos posteriores, mediante **SCALL** o **SETFORM/SETBFORM** imprimirán el recurso en la página.



**Nota:** **PRECACHE** no se admite en todos los controladores. Para determinar si el controlador es compatible con este comando, póngase en contacto con un representante de Xerox.

### Sintaxis

```
(rname) [rot1 scale1 rot2 scale2...rotN scaleN] PRECACHE
```

### Donde:

<b>rname</b>	puede ser un segmento VIPP®, un archivo EPS, PostScript o TIFF.
<b>rotN scaleN</b>	los pares indican el giro y las combinaciones de escala que necesitan ser representadas previamente.

**PRECACHE** puede utilizar archivos ubicados en estos directorios de recursos de VIPP®:

- formlib (definido por **SETFPATH**)
- imglib (definido por **SETIPATH**)
- mislib (definido por **SETMPATH**)

### Ejemplos

```
(car1.eps) [0 1 90 1 0 .5 90 .5] PRECACHE
```

### Comandos relacionados

**SETFPATH**, **SETIPATH**, **SETMPATH**

## PROCESSDJDE

Utilice **PROCESSDJDE** para simplificar el procesamiento de trabajos **DJDE** (flujo de datos LCDS) de VI Compose. **PROCESSDJDE** elimina los **DJDE** de los datos imprimibles y realiza una llamada a un procedimiento definido por el usuario para cada par de palabra clave/parámetro en la línea de **DJDE**.

### Sintaxis

```
{ djde_proc } position (djde_prefix) option PROCESSDJDE
```

### Donde:

<b>djde_proc</b>	es un procedimiento que se ejecuta para cada par palabra clave/parámetro en la línea <b>DJDE</b> . Es una secuencia de comandos de VIPP® que realiza las acciones correspondientes según <b>DJDECMD</b> con la palabra clave <b>DJDE</b> y <b>DJDEPAR</b> con el parámetro relacionado DJDE. Se espera el uso de los comandos <b>CASE</b> y/o <b>IF/ELSE/ENDIF</b> en este procedimiento.
<b>posición</b>	es la posición del prefijo <b>DJDE</b> en el registro, empezando por 0.
<b>djde_prefix</b>	es la cadena de prefijo de <b>DJDE</b> .
<b>opción</b>	es un número que se puede combinar estos valores: <b>+1</b> fuerza una nueva página en el primer <b>DJDE</b> en un paquete <b>DJDE</b> <b>+2</b> cancela la omisión previa en la primera línea de datos después de <b>DJDE</b> <b>+4</b> cancela la omisión previa de la línea en la línea de <b>DJDE</b> <b>+8</b> cancela la omisión posterior de la línea de <b>DJDE</b>

### Ejemplos

```
{ CASE DJDECMD {}% default action = none
  (JDL) {($DJDEPAR..JDT) VSUB SETJDT }
  (FORM) {($DJDEPAR..FRM) VSUB SETFORM }
  (FEED) {CASE DJDEPAR {}
    (AUX){(Preprinted) SETMEDIA}
    (MAIN){(Plain) SETMEDIA}
  ENDCASE
}
ENDCASE
} 0 ($DJDE$) 3 PROCESSDJDE
```

Este ejemplo puede procesar la instrucción DJDE asignando exclusivamente los valores siguientes:

```
$DJDE$ JDL=JDL23, FORM=BILL2. FEED=MAIN, END; DJDECMD=(JDL), DJDEPAR=(JDL23) DJDECMD=(F
```

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

CASE, IF/ELSE/ELIF/ENDIF, DJDEBEGIN



## QRCODE

**QRCODE** crea e imprime códigos de barras QR 2 según las cadenas especificadas y los datos del parámetro. No hay fuentes especiales obligatorias.

### Sintaxis

```
[/AC (string) /BC (string) /KC (string) /MC (string) /NC (string)] QRCODE
[/AC (string) /BC (string) /KC (string) /MC (string) /NC (string)]
scale rotate align QRCODE
[/AC (string) /BC (string) /KC (string) /MC (string) /NC (string)]
[/ELevel EValue /SAppend [Posval Totval] /QRver version] QRCODE
[/AC (string) /BC (string) /KC (string) /MC (string) /NC (string)]
[/ELevel EValue /SAppend [Posval Totval] /QRver version] scale rotate align QRCODE
[ /cmode (data)...] [fit-in-width] rotate align QRCODE
[ /cmode (data)...] [/ELevel EValue...] [fit-in-width] rotate align QRCODE
```

### Donde:

<b>/AC</b>	es el modo de compactación de datos alfanuméricos. Los datos están codificados en una densidad de 2 caracteres por cada 11 bits. Hay 45 caracteres permitidos:  <b>10</b> dígitos numéricos (0-9) <b>26</b> caracteres alfabéticos (A-Z) y <b>9</b> símbolos (SP, \$, %, *, +, -, ., /, :)
<b>/BC</b>	es el modo de compactación de los datos de bytes de 8 bits. El modo codifica el conjunto de caracteres Latin/Kana de 8 bits configurados con JIS X 0201 (valores de caracteres de 00 a FF hex) con una densidad de 8 bits por carácter.
<b>/KC</b>	es el modo de compactación de datos Kanji. El modo Kanji codifica los caracteres Kanji según el sistema Shift JIS en función de JIS X 0208 con una densidad de un carácter de dos bytes por cada 13 bits. Los valores de Shift JIS se desplazan desde los valores de JIS X 0208.
<b>/MC</b>	es el modo de compactación del flujo de datos mixto. Este modo codifica secuencias de datos en una combinación de cualquiera de los modos de compactación.
<b>/NC</b>	es el modo de compactación de datos numéricos. Este modo codifica los datos del conjunto de dígitos decimales (0-9) a una densidad de 3 caracteres de datos por cada 10 bits.
<b>(cadena)</b>	son datos alfanuméricos, byte, Kanji, numéricos o mixtos en función del modo de compactación anterior.



Nota: Se pueden especificar combinaciones de parejas modo y cadena en cualquier orden, aunque un modo siempre debe preceder a su cadena correspondiente.

<b>escala</b>	cambia el tamaño del código de barras (el prefijado es 1). Si la escala es 1, el tamaño del módulo (la celda o el componente más pequeño del código de barras) es 0,254 mm, 0,01 pulgadas, o 6 × 6 puntos en impresoras de 600 ppp.
<b>girar</b>	cambia la orientación del código de barras según un ángulo (el valor prefijado es 0, es decir, no hay giro). Un valor positivo gira el código de barras en sentido antihorario con respecto a su orientación anterior. Por ejemplo, si rotate = -90, el código de barras puede girar 90 grados en sentido horario.
<b>alinear</b>	indica qué punto del código de barras se alinea en la posición de impresión secundaria, mediante estos valores:  <b>0</b> superior izquierda (predeterminado) <b>1</b> superior derecha <b>2</b> centro superior <b>10</b> inferior izquierda <b>11</b> inferior derecha <b>12</b> centro inferior <b>20</b> centro izquierda <b>21</b> centro derecha <b>22</b> centro centro

La matriz opcional [/ELevel EValue /SAppend [Posval Totval] /QRver version] contiene una lista de pares de claves o valores que se pueden usar para especificar las siguientes opciones:

<b>/ELevel</b>	es el nivel de corrección/detección de errores.
<b>EValue</b>	puede estar en el rango entre 1 y 4, donde 1 es el valor mínimo y 4 es la cantidad máxima de corrección. El rango 1 a 4 corresponde a los niveles de L, M, Q y H respectivamente. Si no se especifica ELevel, el nivel 2 o M es el prefijado.
<b>/MPattern</b>	es un parámetro de mejora del rendimiento. Especifica un subconjunto de las ocho tramas de máscara usadas para evaluar y crear el código de barras QR.
<b>MValue</b>	es un número entero en el intervalo 1 - 8 o una matriz de 1 a 8 enteros.

Aunque la especificación del código QR en teoría requiere 8 tramas de máscara distintas para los datos de códigos de barras y la selección de la máscara que resulta en el código de barras idóneo, las pruebas indican que cada trama de máscara resultó en un código de barras legible. Dado que la evaluación de tramas de máscara es una tarea difícil, la opción de usar un solo conjunto de las 8 tramas de máscara se proporciona para mejorar el rendimiento.


<b>/QRopt</b>	es un parámetro de mejora del rendimiento. Activa o desactiva accesos directos en el análisis de tramas de máscara.
<b>opt</b>	es -1 (prefijado) o cualquier otro entero. -1 permite accesos directos; otros números desactivan los accesos directos.

Si se activan los accesos directos, un porcentaje más pequeño de datos de códigos de barras se utiliza en la evaluación de tramas de máscara. Dado que las pruebas indican que cada trama de máscara resulta en un código de barras legible, no es necesario realizar un análisis en profundidad de las tramas de máscara.

Ejemplos

```
[/AC (AC-42)] [/ELevel 4 /MPattern 1] QRCODE % mejorar rendimiento
[/AC (AC-42)] [/ELevel 4 /MPattern [2 3 6] QRCODE % mejorar rendimiento
[/AC (AC-42)] [/ELevel 4 /QRopt 1] QRCODE % desactivar accesos directos, % sin me-
jora de rendimientos
```

<b>/SAppend</b>	es una matriz de unión estructurada opcional que se usa cuando los códigos de barras de más de 1 y hasta 16 QR se tienen que unir en un formato es- tructurado. La unión contiene dos elementos y puede usarse de la siguiente manera:
<b>Posval</b>	es la posición del código de barras dentro del conjunto de códigos de barras unidos.
<b>Totval</b>	es el número total de códigos de barras unidos.
<b>/QRver</b>	es la versión o el tamaño del código de barras QR.
<b>version</b>	es un número entre 1 y 40. Si no se especifica un número de versión, la ver- sión más adecuada se puede determinar automáticamente a partir del nú- mero de caracteres de datos codificados y el valor de <b>EValue</b> . Si <b>QRver</b> y <b>ELevel</b> están especificados, pero <b>QRver</b> no tiene la capacidad de mantener los datos proporcionados en <b>ELevel</b> , se puede informar de que ha habido un error. Si <b>QRver</b> está especificado y <b>ELevel</b> no lo está, se puede informar de que ha habido un error si <b>QRver</b> no tiene la capacidad de mantener los da- tos proporcionados en el <b>ELevel</b> prefijado.

 Nota: Se admiten estos conjuntos de caracteres en el código de barras QR.

JIS X 0201 (JIS 8-bit) JIS X 0208-1997 (Japanese Graphic, Annex 1 Shift Coded Represent

QRCODE y PAGEBRK

QRCODE no es compatible con un código de barras QR que abarque un PAGEBRK. No se permiten PAGEBRK entre los códigos de barras de una sintaxis de unión estructurada.

Carácter Kanji 5C

Si el segundo byte de un carácter kanji es 5C en hexadecimal, carácter de barra diagonal inversa, debe estar precedido por otro 5C o será tratado como un carácter de escape y se extraerá antes de llegar a VI Compose. El

5C adicional debe ser insertada en la cadena de datos por la aplicación que genera el código VIPP®.

### Limitaciones

No se admite el código QR de modelo 1. El modelo 1 es la versión original (más restringida) del modelo 2.

No se admite FNC1 ya que apenas se utiliza. FNC1 es un indicador de modo que identifica símbolos que codifican mensajes con formato según las especificaciones de aplicaciones o del sector.

### Ejemplos

```
[/AC (HTTP://WWW.FUJIXEROX.CO.JP) ] QRCODE  
[/NC (12345678901234567890) ] 1 -90 0 QRCODE  
[/BC (ハンカケモジ) ] 1 0 0 QRCODE  
[/KC (富士ゼロックスー申し込み) ] [/ELEVEL 3 /QRver 8 ] 1 0 0 QRCODE  
[/MC (ABCD1234567 ハンカケモジ富士ゼロックスー申し込み ) ] [/ELEVEL 2 /QRver 4 ] 2 0 0 QRCODE  
[/AC (THIS IS MESSAGE 1 OF A STRUCTURED APPEND SEQUENCE.)] [/SAppend [1 2]] QRCODE  
[/AC (THIS IS MESSAGE 1 OF A STRUCTURED APPEND SEQUENCE.)] [/SAppend [2 2]] QRCODE
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- AZTEC
- DATAMATRIX
- PDF417
- MAXICODE
- QRCODE
- MOVEH
- MOVEHR
- MOVETO

## QSTRIP\_on

**QSTRIP\_on** quita las comillas de apertura y cierre individuales o dobles, si las hay, de todos los campos delimitados en el modo de base de datos. Se pueden combinar campos entrecomillados y sin entrecomillar en un registro. Los delimitadores de campos, como se define mediante **SETDBSEP**, en los campos entrecomillados no se extraen, sino que se retienen como parte de los contenidos del campo.

### Sintaxis

**QSTRIP\_on**

No codifique comandos globales como **QSTRIP\_on**, **DUPLEX\_on**, **SETDBSEP** y **SETBUFSIZE** en el archivo principal de la base de datos. Coloque los comandos globales al principio del archivo de base de datos antes del comando **STARTDBM**. Como alternativa, los comandos globales se pueden colocar en un archivo de ficha del descriptor de trabajos externo referenciado por un comando **SETJDT** colocado antes del comando **STARTDBM** en el archivo de base de datos.

### Ejemplos

```
%!
DUPLEX_on
(;) SETDBSEP
BSTRIP_off
QSTRIP_on
(cas.dbm) STARTDBM
....

%! database file
(cas.jdt) SETJDT
(cas.dbm) STARTDBM
....

%!PS-Adobe-2.0
%%Title: cas.jdt
%%Creator: CAS/RXCH
....
DUPLEX_on
(;) SETDBSEP
QSTRIP_on
BSTRIP_off
....

%!
XGF
QSTRIP_on
(,) SETDBSEP
(dbm1.dbm) STARTDBM
FNAME,NAME,ADDRESS
"John","Martin","23, wall Street"
.....
```

Este ejemplo asigna las variables como se indica a continuación:

FNAME=John

NAME=Martin

ADDRESS=23, Wall Street

Los nombres de campo también se pueden definir mediante el uso de comillas como se indica a continuación:

FNAME, NAME, ADDRESS

John, Martin, 23, Wall Street

### **Modos**

Este comando es aplicable solo en el modo de base datos.

### **Comandos relacionados**

STARTDBM

## RELVAR

**RELVAR** extrae una definición **SETVAR** previa para que pueda ser configurada de nuevo a través de una instrucción **SETVAR** con la opción */INI*.

### Sintaxis

```
/VARxyz RELVAR
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETVAR**

## REPEAT

Utilice **REPEAT** para ejecutar una secuencia de comandos varias veces. **REPEAT** puede utilizarse en un archivo maestro de base de datos si no se puede utilizar el comando **SETCYCLECOPY**. **REPEAT** ejecuta el comando equivalente a **SETCYCLECOPY** en el modo de base de datos.

### Sintaxis

```
{sequence of VIPP commands} count REPEAT
```

### Donde:

<b>{Secuencia de comandos VIPP}</b>	es cualquier secuencia de comandos del modo nativo que proporciona un segmento de una página, una página completa o diversas páginas.
<b>recuento de</b>	es el número de veces que se ejecuta el procedimiento. Este operando puede ser un nombre de campo del archivo de base de datos si el nombre del campo contiene un valor numérico.

En el funcionamiento normal **REPEAT** terminará cuando se alcanza el total. El comando **EXIT** puede utilizarse en una prueba condicional para salir del bucle **REPEAT**.

Use las dos variables enteras de VIPP®, **RPCOUNT** y **RPLEFT**, en el procedimiento **REPEAT** para comprobar qué iteración se ejecuta.

Para facilitar el control, utilice estas variables con **IF/ELSE/ENDIF**:

<b>RPCOUNT</b>	indica el número de iteraciones actuales en el comando <b>REPEAT</b>
<b>RPLEFT</b>	indica el número de iteraciones restantes en el comando <b>REPEAT</b>



Nota: El uso de **RSAVE** y **RESET** en **REPEAT** no se recomienda ya que pueden producir resultados impredecibles.

### Ejemplos

Este ejemplo usa **REPEAT** para imprimir varias copias del formulario del archivo maestro de base de datos para cada registro en el archivo de base de datos, cambia el número de copias entre registros y produce un **ENDOFSET** en la última página.

```
IF RPLEFT 1 eq
  {ENDOFSET}
ENDIF
{ 0 2400 MOVETO
  ($$GREETING..tif) VSUB 1 0 ICALL
  PAGEBRK
} NUMBER REPEAT
```

Este ejemplo muestra cómo se cambia la fuente de la última repetición de un bucle **REPEAT**.



```

300 2500 MOVETO
{
(I am looping) 0 SHP
IF RPCOUNT 9 gt
  { /NCRB 30 SETFONT } ENDIF
} 10 REPEAT

```

Este ejemplo muestra cómo salir de un bucle si la posición horizontal ha alcanzado el borde de la página.

```

/VARHoz 300 SETVAR
/VARVer 300 SETVAR
/VARNumber 0 SETVAR

{
VARHoz VARVer MOVETO
VARNumber 50 2 SHP
/VARNumber ++
/VARHor 50 ADD
IF VARHoz 2550 ge { EXIT} ENDIF
} 24 REPEAT

```

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

- IF/ELSE/ELIF/ENDIF
- STARTDBM

## RESET

**RESET** restaura el contexto de VIPP® según el valor prefijado inicial o el valor almacenado por el último **RSAVE**. Cancela todas las opciones, formularios, fuentes, etc., desde el último **RSAVE**.

### Sintaxis

RESET

### Colocación

Especifique este comando solo después de un comando de delimitador de página, por ejemplo, **PAGEBRK**, avance de formulario u omitir canal.

### No usar

No utilice **RESET** en un comando **REPEAT** ni después del comando **SETCYCLECOPY**. De lo contrario, pueden producirse resultados impredecibles.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**RSAVE**

## RPEDEF

**RPEDEF** proporciona una manera rápida de generar definiciones RPE simplificadas que procesarán los registros con un índice de fuente. Las definiciones de RPE simplificadas no tienen ninguna selección de campos, alineación izquierda ni giros.

Utilice **RPEDEF** en registros NMP ( % %XGF) si es necesaria la conversión de las instrucciones **LCDS DJDE FONTS=**. **RPEDEF** también puede codificarse en las fichas del descriptor de trabajos.

Para obtener más información sobre otros comandos relacionados con RPE, consulte [Información de comando RPE](#).

### Sintaxis

```
[ [ /font0 Ydispl0 ] [ /font1 Ydispl1 ] ... ] RPEDEF
```

### Donde:

<b>fontN</b>	es un índice de fuente VIPP® previamente definido por <b>INDEXFONT</b> , que se utilizará con el índice de fuente del registro N.
<b>YdisplN</b>	es el espaciado de línea, en unidades actuales, asociada a esta fuente.
<b>Colorkey</b>	es una Colorkey VIPP®.
<b>Fheight</b>	especifica un valor de altura de la fuente expresado en unidades actuales para cada fuente en la lista. Esto hace que el margen superior configurado por <b>SETMARGIN</b> haga referencia a la parte superior de la celda en vez de la línea base de manera predeterminada para la primera línea de la página. Esta opción se utiliza principalmente para convertir los flujos de datos LCDS mediante el índice de fuentes.

### Ejemplos

Este ejemplo equivale a la codificación RPE más abajo.

```
[ [ /F0 50 ] [ /F1 80 RED ] [ /F2 [ 60 40 ] ] RPEDEF 1 SETRPEPREFIX 2 BEGINRPE /0 RPEKE
```

Esta información se debe definir en el archivo xgf.def, en la JDT o en un registro NMP anterior:

- Posición inicial (valor de **DJDE BEGIN=**) de **SETMARGIN**
- buf\_size de **SETBUFSIZE**
- /font0, /font1, ... con **INDEXFONT**

El valor del primer byte del registro, o el segundo byte si el modo PCC está activo, se usa para seleccionar el **SETMARGIN** simplificado definido por **RPEDEF**. La selección se basa en la posición de la definición de RPE en el comando **RPEDEF**.

Este ejemplo imprime hello world con la fuente /F1 y un desplazamiento vertical de 20 unidades. *I am still alive* se imprime con la fuente /F2 con un desplazamiento vertical de 30.

```
%%XGF [ [/F1 20 ] [/F2 30] ] RPEDEF 0hello world 1I am still alive
```

Los valores de índice de fuente empiezan con cero y pertenecen a un intervalo de 0 a 9 y de A a Z. Si la lista del

cliente empieza con 1, la lista **RPEDEF** empieza con una entrada ficticia.

Si el índice de la fuente no es el primer byte del registro, o el segundo byte del registro si se usa PCC, **SETRPEPREFIX** puede codificarse antes de **RPEDEF**. Si se transforma un flujo de datos **EBCDIC**, el comando **INDEXFONT** usado para definir la fuente puede codificarse antes del comando **RPEDEF**.

En este ejemplo, la porción extraída del registro como datos imprimibles (de 0 a 131) se acaba antes del byte de índice de fuentes.

[ 1 131 ] SETRPEPREFIX [ ..... ] RPEDEF
---

### Comandos relacionados

- INDEXFONT
- SETBUFSIZE
- SETMARGIN
- SETPAT
- SETTXC
- SETRPEPREFIX
- RPEKEY

## RPEKEY

**RPEKEY** empieza con una definición de RPE que el prefijo de línea invocará si se usa **SETRPEPREFIX** o mediante un comando **SETRPE** en el flujo de datos. Los nombres de clave RPE que coinciden con todos menos el último carácter, formarán un grupo. Consulte **SETRPEPREFIX** para obtener más información, consulte también Información de comando RPE y el resto de comandos RPE relacionados.

### Sintaxis

```
/rpekeyname RPEKEY [ align rotate Xinit Xdispl Yinit Ydispl recpos length /font Colorke
```

### Donde:

**rpekeyname** es un prefijo de línea que coincide con la definición de **SETRPEPREFIX**. Si el prefijo contiene uno o más espacios, la sintaxis debe ser rpekeyname en lugar de /rpekeyname.



Nota: Siempre que un prefijo RPE contiene espacios, debe programarse entre paréntesis. Por ejemplo:  
(REC1 45 C) RPEKEY [ ... RPE entry ... ]

Para obtener más información, consulte **FROMLINE** para obtener una descripción completa de los parámetros de entrada de RPE y Ampliación de los comandos FROMLINE y RPEKEY.

### Ejemplos

```
% set RPE fonts
/F1 /NHEB 9 13 INDEXFONT
/F2 /NHEB 10 15 INDEXFONT
/F3 /NHEB 13 15 INDEXFONT
/F4 /NHEB 18 INDEXFONT
/F5 /NHEB 13 INDEXFONT

% RPE definition
5 BEGINRPE

% almt rot. Xinit Xdispl Yinit Ydispl recpos length font color
/HDRO RPEKEY
[ 2 0 835 0 300 0 00 99 /F4 BLACK ]
/CL10 RPEKEY
[ 2 0 615 0 445 0 00 99 /F1 WHITE ]
/CL20 RPEKEY
[ 2 0 1199 318 445 0 00 99 /F1 WHITE ]
/CL30 RPEKEY
[ 2 0 3140 0 445 0 00 99 /F5 WHITE ]
/BODY RPEKEY
[ 0 0 230 0 560 75 00 33 /F2 BLACK ]
[ 1 0 1345 0 560 75 33 12 /F2 BLACK ]
[ 1 0 1658 0 560 75 45 12 /F2 BLACK ]
[ 1 0 1976 0 560 75 57 12 /F2 BLACK ]
[ 1 0 2286 0 560 75 69 12 /F2 BLACK ]
[ 1 0 2610 0 560 75 81 12 /F2 BLACK ]
[ 1 0 2945 0 560 75 93 12 /F2 BLACK ]
[ 1 0 3290 0 560 75 105 12 /F3 BLACK ]
ENDRPE
```

## Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

- BEGINRPE
- COPYRANGE
- ENDRPE
- FROMLINE
- INDEXRPE
- SETPAT
- SETPCD
- SETRCD
- SETRPEPREFIX
- SETTXC

## RSAVE

**RSAVE** guarda el contexto actual, que se restaura más tarde mediante el comando **RESET**.

### Sintaxis

RSAVE

### Colocación

Este comando solo se puede producir antes del primer comando de marca (por ejemplo, **SHx**, **DRAWB**, **DRAWBR**, **DRAWBM**, **DRAWBRM**, **ICALL** y **SCALL**) en una página y después de un delimitador de página si no es la primera página. De lo contrario, los efectos se limitan a la página actual.

### No usar

No utilice **RSAVE** en un comando **REPEAT** ni después del comando **SETCYCLECOPY**. De lo contrario, pueden producirse resultados impredecibles.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**%%EOF** y **RESET**.

## RUN

El comando **RUN** ejecuta el código de VIPP® o PostScript contenido en el archivo de referencia. El archivo referenciado debe almacenarse en una de las bibliotecas definidas por **SETMPATH**, o en un repositorio de software Xerox® FreeFlow Makeready® con tecnología VIPP®.

### Sintaxis

```
(file name) RUN
```

```
(file name) option RUN
```

### Donde:

#### opción

puede incluir:

**0** ninguna acción especial, predeterminado

**1** guardar/restaurar encapsulación para proteger el consumo de memoria

**2** guardar/restaurar + interacción VIPP®/PostScript básica

**3** guardar/restaurar + interacción completa VIPP®/PostScript limitada a archivos PostScript creados por algunos controladores.

**0** ninguna acción especial, predeterminado

guardar/restaurar encapsulación para proteger el consumo de memoria

guardar/restaurar + interacción VIPP®/PostScript básica

guardar/restaurar + interacción completa VIPP®/PostScript limitada a archivos PostScript creados por algunos controladores.

La opción **0** (o sin opción) se utiliza con archivos con el código VIPP®, y es la única opción posible con estos archivos.

Para las aplicaciones que deben proteger la ejecución de **RUN**, se puede especificar la opción **1**. Además, las opciones **2** y **3** activan una interacción entre VIPP® y PostScript en la que las funciones relacionadas con páginas, como **SETFORM**, **SETPAGENUMBER** y **SETFRAME**, permanecen activas durante la ejecución del trabajo de PostScript.

La opción **2** aplica funciones VI Compose al final de la ejecución de la página de PostScript. Por este motivo, no se permite **SETMEDIA** con esta opción.

La opción **3** aplica funciones VI Compose al principio de la ejecución de la página de PostScript. Se puede usar **SETMEDIA**. Esta opción solo es compatible con archivos PostScript generados por algunos controladores de PostScript. Se recomiendan los controladores Apple LaserWriter NTX II o Adobe.

En el modo varias en 1, si se usa un formulario, se imprimirá en la página lógica siguiente a partir de la última página si el lado de la hoja no se llena completamente. Las opciones **2** y **3** también actualizan la información del contexto, como el número de página y la etiqueta frontal o posterior de **NEWFRONT/NEWBACK**.

No hay ninguna garantía que las opciones **2** y **3** funcionen con cada archivo PostScript, dada la naturaleza de PostScript, pero deben funcionar con la mayoría de estos.



## RUN y Demographics

Para utilizar el comando RUN con la función Demographics, puede utilizar la opción 2 o 3.

## Run y Normalize

No se puede normalizar el código de PostScript invocado por RUN. Para más información, consulte la *Guía del usuario de FreeFlow VI Compose*.

Mediante la utilización de una referencia vacía, el comando RUN se puede colocar después de las opciones de VIPP® y antes del principio de un documento PostScript, para que no sea necesario un archivo de envío de VIPP® independiente. Si el código VIPP® siguiente se anexa después del código PS, el marcador **% %EOD\_XGF** se coloca entre el final del código PS y el código VIPP® para terminar correctamente el comando RUN y reanudar la ejecución de código VIPP® habitual.

## Ejemplos

Este ejemplo muestra una referencia vacía en el comando **RUN**.

```
% !
XGF
(copy.frm) SETFORM
() 3 RUN
%!PS_Adobe-3.0
.....
```



Nota: El uso de código ineficaz de PostScript puede afectar al rendimiento.

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

**RESET**, **SETMPATH**

## RUNDD

**RUNDD** imprime un documento previamente procesado por el servicio de descomposición en DocuPrint NPS o controladores del servidor de impresión FreeFlow.

### Sintaxis

```
(document_name) RUNDD (document_name) /ENDOFSET RUNDD (document_name) /VARxyz RUNDD
```

### Donde:

<b>(document_name)</b>	es el nombre de un documento previamente procesado por el servicio de descomposición.
<b>/ENDOFSET</b>	se aplica el acabado previamente definido por <b>SETFINISHING</b> en el documento.
<b>/VARxyz</b>	hace referencia al procedimiento definido anteriormente con <b>SETVAR</b> que se ejecutará al principio de la última página del documento.

**RUNDD** encuentra la página inicial relacionada con el documento en las bibliotecas definidas por **SETIPATH** o **SETFPATH** y, a continuación, encuentra todas las páginas consecutivas en la misma biblioteca asociada al documento.

**RUNDD** se puede combinar con **SETFORM** y **SETMEDIA**, normalmente con la sintaxis de ciclo de estos comandos, para asignar formularios y materiales de impresión con determinadas páginas del documento.

Utilice **SETFORM** o **ENDPAGE** para colocar datos variables o estáticos en todas las páginas del documento descompuesto. **SETFORM** imprime los datos en la página descompuesta (las áreas blancas son transparentes), **ENDPAGE** los imprime en la parte superior. Para colocar determinados datos en ciertas páginas, implemente un contador y use las instrucciones **IF** o **CASE** en el formulario o en un procedimiento **ENDPAGE**. La codificación de diseños más complejos también puede realizarse mediante **SETPAGEDEF**.

Puede usar **RUNDD** con **SETCYCLECOPY/COLLATE\_off**.

### Compatibilidad con la aplicación DocuPrint NPS y el servidor de impresión de FreeFlow

El comando **RUNDD** realizará la búsqueda en bibliotecas **SETIPATH** de imágenes TIFF con nombres que se ajustan a la convención de servicios de descomposición de DocuPrint NPS:

```
document_name.pnnnn.c.tif
```

### Donde:

<b>document_name</b>	es el nombre del archivo enviado a los servicios de descomposición.
<b>nnnn</b>	es el número de página de cuatro dígitos.
<b>c</b>	es:
	<b>b</b> negro
	<b>h</b> color de realce

Si no se encuentran imágenes TIFF, esta nomenclatura de servicios de descomposición del servidor de impresión FreeFlow se utiliza para volver a realizar la búsqueda:

```
document_name_dir/document_name.pnnnnnnnn.tif
```

Donde:

<b>document_name</b>	es el nombre del archivo enviado a los servicios de descomposición.
<b>nnnnnnnn</b>	es el número de página de ocho dígitos.
<b>document_name_dir</b>	es el nombre del subdirectorio que contiene las imágenes TIFF.

En ambos casos, cuando se encuentran imágenes TIFF, **RUNDD** las imprime en páginas independientes mediante el comando ICALL.

### Solo DocuPrint NPS

Todas las páginas **.b.tif** se imprimen en negro, se aplica **0 setgray**, y las imágenes **.h.tif** se superponen en la imagen en negro mediante el color definido anteriormente por **SETTXC**.

Si las imágenes TIFF no se encuentran en los pasos anteriores, los formularios nombrados según la nomenclatura de servicios de descomposición de DocuPrint NPS se buscan en las bibliotecas de **SETFPATH**:

```
document_name.pnnnn.ps
```

Donde:

<b>document_name</b>	es el nombre del archivo enviado a los servicios de descomposición.
<b>nnnn</b>	es el número de página de cuatro dígitos.

Si se encuentran los formularios, **RUNDD** los imprime en una página individual mediante el operador de ejecución de PostScript. Todas las páginas del formulario se imprimen en blanco y el color de realce actual.

La opción de formularios solo se admite en DocuPrint NPS, y es necesaria si se utiliza la descomposición **Save-MaskBC** de bytes comprimidos.

### Solo el servidor de impresión FreeFlow

Si la ubicación predefinida de las imágenes de descomposición del servidor de impresión FreeFlow se modifica, la nueva ruta/ubicación puede agregarse a **SETIPATH** en **xgunix.run**. NO cambie ni quite la ruta a la ubicación predeterminada original.

### Ejemplos

Este ejemplo imprime el documento descompuesto **doc1** de forma secuencial mediante el conjunto de formularios y materiales de impresión definidos por los comandos **SETFORM** y **SETMEDIA** de la manera siguiente:  
**doc1(page1)+form1+med1, doc1(page2)+form2+med2, ...**

```
[ (form1) (form2) .... (formn) ] SETFORM [ (med1) (med2) .... (medn) ] SETMEDIA (doc1) F
```

Al usar un documento descompuesto como formulario, combine **RUNDD** con **SETFORM** para eliminar las diferencias de impresión entre la familia de impresoras DocuPrint NPS y los controladores del servidor de impresión

FreeFlow. Ejemplo de sintaxis (se supone que doc1.ps es un documento de una página):

```
{ (doc1.ps) RUNDD } SETFORM
```

Esta solución permite a los formularios descompuestos imprimir lo mismo en las tres familias de impresoras.

El ejemplo imprime This page belongs to name (campo del archivo de la base de datos) en cada página del documento descompuesto:

```
{ 1275 50 MOVETO (This page belongs to $$NAME. ) VSUB 1200 2 SHP } SETFORM (Doc1) RUNDD
```

Este es un ejemplo del uso de **ENDPAGE** al imprimir This page belongs to name solo en las tres primeras páginas.

```
{ 1275 50 MOVETO (This page belongs to $$NAME. ) VSUB 1200 2 SHP } 3 ENDPAGE (Doc1) RUNDD
```

El usuario define el 3, y representa el número de páginas en las que el procedimiento permanece activo.

Este ejemplo imprime los campos Name y Address del archivo de la base de datos en las páginas 2 y 5 del documento descompuesto.

```
/VARpage (00) SETVAR
{ /VARpage ++
  IF VARpage (02) eq VARpage (05) eq or
  { 1275 50 MOVETO
    NAME SHL
    ADDRESS SHL
  }
  ENDIF
} SETFORM
(Doc1) RUNDD
```

Esto es un ejemplo de la impresión de un número de página secuencial en una página de descomposición.

```
%!
XGF
DUPLEX_on
PORT
/VARPageCount (00001) SETVAR % Starting number for page.
/NHEB 10 SETFONT
{
1275 50 MOVETO
(Page $$VARPageCount.) VSUB 900 2 SHP
} SETFORM
```

Esto es un ejemplo de la funcionalidad de juego. Grapa todos los juegos.

```
/Staple /ON SETFINISHING (FirtsSection.ps) /ENDOFSET RUNDD (SecondSection.ps) /ENDOFSET
```

En este ejemplo se grapan todas las secciones. Sin embargo, no se grapa la cubierta.

```
150 3000 MOVETO  
(Cover Page) SH  
PAGEBRK  
/Staple /ON SETFINISHING  
(FirstSection.ps) RUNDD  
(SecondSection.ps) RUNDD  
(ThirdSection.ps) /ENDOFSET RUNDD
```

## Modos

Este comando es aplicable en modo nativo y de base de datos.

## Comandos relacionados

- [BCALL](#)
- [ENDIMP](#)
- [ENDOFRUN](#)
- [ENDOFSET](#)
- [SETFPATH](#)
- [SETIPATH](#)
- [RUNPDF](#)
- [RUNTIF](#)

## RUNPDF

**RUNPDF** imprime un documento PDF que contiene una o más páginas y es compatible con páginas de varios tamaños y orientaciones. Se imprime cada página del archivo PDF en una página lógica independiente similar a **RUNDD**.

### Sintaxis

(document1.pdf) RUNPDF



Nota: En el procesamiento RIP de PostScript, las páginas PDF se representan como un rectángulo con los nombres de PDF a menos que la información de EPS haya sido incorporada en el contenido del archivo PDF. Se realiza automáticamente cuando el archivo PDF se utiliza como un recurso de *VI Design Pro* o *VI Design Express*. En procesamientos de PDF de VI eCompose, no es necesario normalizar este paso.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **RUNDD**
- **RUNTIF**
- **BEGINIMP**
- **ENDIMP**



- BCALL
- ENDIMP
- RUNDD
- RUNPDF



## SAVEPP

**SAVEPP** guarda la posición de impresión secundaria actual para un uso posterior mediante **HDISP**, **VDISP**, **SHPOS** y **SVPOS**. Hay diversos usos de este comando, pero el uso más habitual de **SAVEPP** es dibujar cuadros o líneas variables entre párrafos de longitud variable.

### Sintaxis

SAVEPP

### Ejemplos

Utilice este ejemplo para dibujar una línea entre los párrafos mediante **SAVEPP**.

```
150 3000 MOVETO
400 0 SHP   %This is a variable length paragraph which could have many lines
25 NL
SAVEPP
SHPOS SVPOS 400 0 S1 DRAWB
25 NL
400 0 SHP   %This is the next paragraph
```

Para dibujar un cuadro en lugar de una línea, utilice los comandos **VDISP** y **HDISP**.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **HDISP**
- **SHPOS**
- **SVPOS**
- **VDISP**

## SCALL

**SCALL** imprime un objeto o un segmento en la página actual. El origen (0,0) del objeto o del segmento se coloca en la posición de impresión secundaria. Un objeto o un segmento puede ser:

- Un segmento codificado mediante el modo nativo VIPP® o un simple PostScript
- Un archivo TIFF, JPEG, PostScript, PostScript encapsulado (EPS) o XObject
- Un archivo PDF de una página.
- En procesamientos RIP de PostScript, el PDF se representa como un rectángulo y relleno con el nombre de PDF a menos que la información de EPS haya sido incorporada en el contenido del archivo PDF. Esta acción se realiza automáticamente cuando el archivo PDF se utiliza como un recurso de VI Design Pro o VI Design Express. En procesamientos de PDF como VI eCompose o Normalizer, no es necesario este paso.
- Sintaxis:
  - Cuando se realiza una llamada a un segmento, todas las variantes de la sintaxis que se muestran a continuación pueden utilizarse con o sin el comando **CACHE**.
  - Al realizar una llamada a otro formato como TIFF, JPEG, EPS, PS o PDF con el comando **CACHE** se pueden usar todas las variantes de sintaxis.
  - Al realizar una llamada a otro formato como TIFF, JPEG, EPS, PS o PDF sin el comando **CACHE**, las tres opciones scale, rotate y align son obligatorias.
- Ubicación:
  - Para cualquier sintaxis, TIFF y JPEG son obligatorios en imglib con el comando **SETIPATH**, o en directorios de proyectos con el comando **SETPPATH**.
  - Para un segmento sin el comando **CACHE** y sin la opción align, el segmento es necesario en formlib con el comando **SETFPATH**, en directorios de proyectos con el comando **SETPPATH**.
  - En cualquier otro caso, la ubicación es necesaria en imglib con el comando **SETIPATH**, en formlib con el comando **SETFPATH** y en mislib con el comando **SETMPATH**, o en directorios de proyectos con el comando **SETPPATH**.
- Para ver la compatibilidad y limitaciones del formato de imagen, consulte el comando **ICALL**.

### Sintaxis

```

(Segmentname) SCALL
(Segmentname) scale SCALL
(Segmentname) scale rotation1 SCALL
(Objectname) scale rotation1 align1 SCALL
(Objectname) [width height] rotation2 align2 SCALL
(Objectname) CACHE scale rotation1 align1 SCALL
(Objectname) CACHE [width height] rotation2 align2 SCALL
{ segment contents } SCALL
{ segment contents } scale SCALL
{ segment contents } scale rotation1 SCALL

```

Donde:

**Segmentname** es el nombre de un segmento.

**Objectname** es el nombre de un segmento, TIFF, JPEG, PostScript, archivo EPS, PDF o XObject.

**escala** es:

Un **solo factor de escala del mismo formato** (número real, puede ser negativo)

Una **matriz** del formato: [ scaleH scaleV /A ]

**Donde:**

**scaleH scaleV** son 2 factores de escala anamórfica (números reales, pueden ser negativos).

**Números negativos** se utilizan para el efecto espejo (horizontal, verticalmente o ambos)

**rotation1** es el ángulo del giro del segmento, en sentido antihorario, alrededor de su punto de alineación. El valor predeterminado es 0.

**align1** es la opción de alineación que define qué punto de la imagen debe ser alineado en la posición de impresión secundaria. (Utilice el marcador BBOX para segmentos en línea). Debe ser uno de los siguientes:

- 0 superior izquierda
- 1 superior derecha
- 2 centro superior
- 10 inferior (predeterminado)
- 11 inferior derecha
- 12 centro inferior

	<b>20</b> centro izquierda
	<b>21</b> centro derecha
	<b>22</b> centro centro
<b>ancho alto</b>	son el tamaño, en unidades actuales, de un cuadro en el que se ajustará el objeto. Esto se conoce como la opción de ajuste al cuadro. La esquina inferior izquierda del cuadro se coloca en la posición de impresión secundaria.
<b>rotation2</b>	es el giro del objeto en el interior del cuadro (solo se admiten los valores 0, 90, 180 y 270).
<b>align2</b>	indica cómo se alineará la imagen en el interior del cuadro. Los valores son los mismos que align1.  Tres pesos opcionales pueden combinarse con las opciones de align2:  <b>+100</b> opción de cuadro de relleno Cuando esta opción se activa, la imagen se ajusta a escala de forma isomórfica para que se rellene todo el cuadro. Si las proporciones de anchura/altura del cuadro y la imagen son distintas, la parte que sale del cuadro se recorta. En este caso, las opciones de alineación se usan para seleccionar qué parte de la imagen (esquinas, lado izquierdo, derecho, superior, inferior o el centro) se mantiene visible en el cuadro.  <b>+200</b> opción de cuadro flexible. Cuando esta opción se activa, la imagen se ajusta a escala de forma anamórfica para que se rellene todo el cuadro. Si las proporciones de anchura/altura del cuadro y la imagen son distintas, la imagen se estira vertical u horizontalmente. En este caso, las opciones de alineación no son útiles.  <b>+300</b> colocación en el cuadro. Cuando esta opción se activa, la imagen no se ajusta a escala. Se coloca en el cuadro según las opciones de alineación. La parte de la imagen que está fuera del cuadro, si la hay, se recorta.
<b>contenidos del segmento</b>	representa un segmento pequeño integrado en el flujo de datos. Dicho segmento, que no hace referencia a ningún archivo externo, se conoce como un segmento en línea. Cuando se utiliza esta sintaxis, no utilice <b>CACHE</b> .  Los segmentos deben estar codificados en el modo nativo VIPP® o PostScript simple y deben almacenarse en una de las bibliotecas definidas por <b>SETFPATH</b> . Se recomienda el uso de la extensión .seg.

**CACHE** es opcional después de segmentname en los primeros tres ejemplos de sintaxis. Si el comando **CACHE** se usa con un segmento, debe usar una instrucción **% %BoundingBox** al principio del segmento para definir el tamaño de la imagen y su posición en función del origen.

En un segmento, las posiciones de impresión dependen del origen. La posición de impresión inicial se configura

de forma predefinida como 0,0, es decir, el origen del segmento. En otras palabras, **0 0 MOVETO** se ejecuta implícitamente al principio de un segmento.

Los archivos EPS y PostScript de una página creados por aplicaciones de procesamiento de documentos mediante el uso de controladores habituales, normalmente, archivos `.prn`, además de los archivos TIFF y JPEG, se pueden imprimir mediante **SCALL**. Estos objetos deben almacenarse en una biblioteca referenciada por los comandos **SETFPATH**, **SETIPATH**, **SETMPATH** o **SETPPATH**.

#### Sintaxis para los archivos PDF:

Las siguientes son las únicas sintaxis compatibles con archivos PDF:

```
(xyz.pdf) CACHE SCALL
```

```
(xyz.pdf) scale rotate align SCALL
```

#### Por ejemplo:

```
(xyz) .pdf SCALL
```

se debe convertir en:

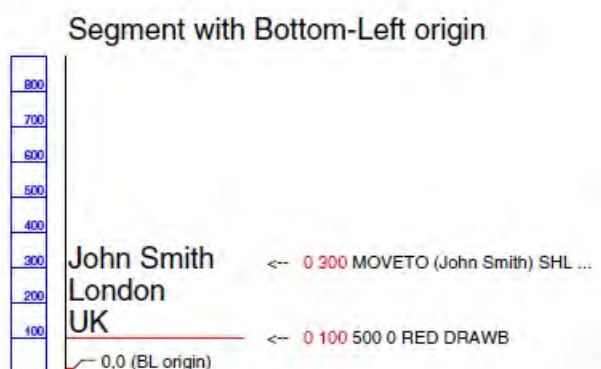
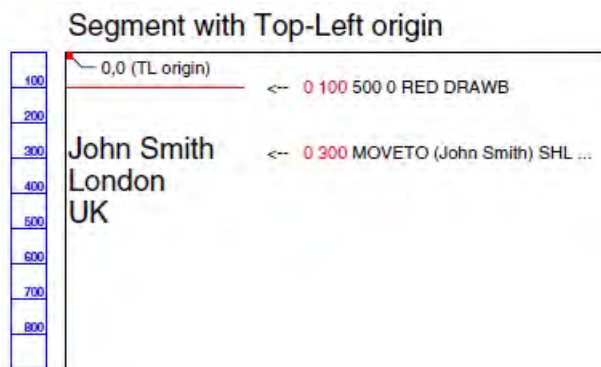
```
(xyz.pdf) 1 0 10 SCALL
```

#### Origen de los segmentos codificados mediante el lenguaje VIPP®:

El origen de las colocaciones en un determinado segmento codificado con VIPP® es la posición secundaria de impresión cuando se ejecuta **SCALL**.

A continuación, se considera esa posición:

- inferior izquierda (las inserciones verticales se miden hacia arriba) de forma prefijada o si se codifica **ORIBL**
- superior izquierda (las inserciones verticales se miden hacia abajo) si se codifica **ORITL**



### Cuadro de límite:

Si **SCALL** se usa en combinación con **CACHE**, la instrucción cuadro de límite

```
(%%BoundingBox: llx lly urx ury)
```

se utiliza para calcular el tamaño y el recorte de la imagen del segmento. **llx lly** se considera el origen de la imagen y coincide con la posición de impresión secundaria.

### Ejemplos

```
(dcxlogo.seg) SCALL
```

```
(doc1.eps) CACHE .6 SCALL
```

```
{ /NHE 18 SETFONT (XY Corp) SHc 25 NL (xy.tif) 1 0 2 ICALL } SCALL
```

Este ejemplo utiliza la opción de cuadro flexible:

```
(picture1.jpg) CACHE [ 500 1000 ] 0 222 SCALL
```

En el siguiente ejemplo...

```
(XObject) CACHE 1 90 0 SCALL
```

La ventaja es la eficacia y la optimización de la productividad, porque **SCALL** ya no tiene que imprimir el objeto.



Nota: El marcador JPEG/Exif contiene información acerca de la resolución que se ignoró hasta la versión 4.0 de FreeFlow. En las versiones anteriores de VIPP® puede haber tenido un valor predefinido de 300 ppp y posiblemente se haya procesado la imagen con un tamaño incorrecto. Dado que la imagen ahora se representará con el tamaño correcto, los trabajos de VIPP® existentes que hacen referencia a archivos JPEG con marcadores Exif sufrirán un problema de incompatibilidad con versiones anteriores. Para solucionar este problema, es preciso modificar la opción de escala de los comandos **SCALL**. Si el cambio tiene algún efecto en la producción, el comportamiento anterior se restablecerá a través de la adición temporal de la instrucción siguiente en el JDT, o `/usr/xgf/src/xgf.def: [ /ProcessExif false ] SETPARAMS`.

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

- **ICALL**
- **CACHE**
- **FCALL**
- **MOVETO**
- **MOVEH**
- **SETFPATH**
- **SETIPATH**
- **SETMPATH**

## SETBAT

**SETBAT** define una clave de atributo de fondo. Si **SETTXB** o **INDEXBAT** la activa, una **BATkey** pinta un fondo bajo todo el texto siguiente impreso mediante comandos **SHx**. El fondo del texto puede constar de cualquier combinación de líneas y cuadros (trazados o rellenos), denominados elementos. Un conjunto de BATkeys estándar se proporciona en `/usr/xgf/src/xgf.bat`. Se pueden probar mediante la impresión de estos archivos:

- `/usr/xgf/demo/sambat.ps` (para BATkeys grises)
- `/usr/xgf/demo/sambatr.ps` (para BATkeys rojas)
- `/usr/xgf/demo/sambatg.ps` (para BATkeys verdes)
- `/usr/xgf/demo/sambatb.ps` (para BATkeys azules)

### Sintaxis

```
/BATkey [Col_GEP LineW WVal WC HVal HC Xpos XC Ypos YC Corner CC UR .....] SETBAT
```

Donde:

<b>Col_GEP</b>	es una Colorkey o una <b>GEPkey</b> como se definen en <code>xgf.gep</code> .
<b>LineW</b>	es un porcentaje de la altura de la fuente usada para configurar la anchura (el grosor) de la línea de este elemento (por ejemplo, 0.06 con una fuente de 20 puntos configurará una anchura de línea de 1.2 puntos). Si es cero, la anchura de la línea configurada por GEPkey se usará independientemente del tamaño de la fuente.
<b>WVal</b>	se utiliza para calcular la anchura del elemento en función del código de anchura.
<b>WC</b>	es el código de anchura: <b>0</b> WVal es un valor fijo en unidades actuales <b>1</b> WVal es un porcentaje de la altura de la cadena que se agregará a la anchura de la cadena <b>2</b> WVal es un valor fijo que se agregará a la anchura de la cadena <b>3</b> WVal es un porcentaje del valor máximo de la anchura de la cadena y de la altura de la cadena
<b>HVal</b>	se utiliza para calcular la altura del elemento en función del código de altura.
<b>HC</b>	es el código de altura: <b>0</b> HVal es un valor fijo en unidades actuales <b>1</b> HVal es un porcentaje de la altura de la cadena <b>2</b> HVal es un valor fijo que se agregará a la altura de la cadena <b>3</b> HVal es un porcentaje del valor máximo de la anchura de la cadena y de la altura de la cadena



<b>XPos</b>	se utiliza para calcular la posición horizontal del elemento en función del código X.
<b>XC</b>	<p>es el código X:</p> <p><b>0</b> Xpos al es un valor fijo en unidades actuales</p> <p><b>1</b> Xpos es un porcentaje de la altura de la cadena agregado a la posición de la cadena horizontal</p> <p><b>2</b> XPos es un valor fijo que se agregará a la posición de la cadena horizontal</p> <p><b>3</b> el elemento se centra horizontalmente de forma automática en el texto. Además, Xpos es un porcentaje del valor máximo de la anchura de la cadena y de la altura de la cadena agregado a la posición del elemento horizontal</p>
<b>YC</b>	<p>es el código Y:</p> <p><b>0</b> Ypos es un valor fijo en unidades actuales</p> <p><b>1</b> Ypos es un porcentaje de la altura de la cadena agregado a la posición de la cadena vertical</p> <p><b>2</b> Ypos es un valor fijo que se agregará a la posición de la cadena vertical</p> <p><b>3</b> el elemento se centra verticalmente de forma automática en el texto. Además, Ypos es un porcentaje del valor máximo de la anchura y de la altura de la cadena agregado a la posición del elemento vertical</p>
<b>Corner</b>	se utiliza para calcular el radio de la esquina del elemento en función del código de esquina.
<b>CC</b>	<p>es el código de esquina:</p> <p><b>0</b> Corner es un valor fijo en unidades actuales</p> <p><b>1</b> Corner es un porcentaje de la altura de la cadena</p> <p><b>2</b> Corner es un valor fijo que se agregará a la altura de la cadena</p> <p><b>3</b> Corner es un porcentaje del valor máximo de la anchura y la altura de la cadena</p>
<b>UR</b>	<p>establece la altura de la cadena:</p> <p><b>0</b> la altura de la cadena es la altura de la X mayúscula</p> <p><b>1</b> la altura de la cadena es la altura real de la cadena (las posiciones se ajustarán en trazos ascendentes y descendentes)</p>

## Ejemplos

Este ejemplo define una **BATkey** de subrayado. La línea será negra, tendrá un grosor igual al 6 % de la altura de la fuente, una anchura igual a la anchura del texto, y se colocará a una altura del 25 % de la altura de la fuente,

bajo la línea base.

```
/UNDL [ BLACK .06 0 2 0 0 0 2 -.25 1 0 0 0 ] SETBAT
```



Nota: Si el código es 0, WVal, HVal, Xpos, Ypos y Corner se pueden especificar como una variable (por ejemplo: anchura /VAR). Se proporciona una variable interna (/COLW) para representar la anchura de la columna definida por **SETCOLWIDTH**.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- SETCOLWIDTH
- SETPAT
- SETTXB
- SETTXC
- SHX

## SETBFORM

**SETBFORM** selecciona un formulario para imprimirlo en las caras posteriores de las hojas sin datos variables en las hojas actuales y las hojas siguientes. Este comando debe especificarse antes de cualquier comando de impresión en la hoja. De lo contrario, este comando genera los mismos resultados que **SETFORM**. Consulte **SETFORM** para ejemplos y descripciones de la sintaxis.



Nota: **SETBFORM** genera un formulario posterior en cada página. El comando null **SETBFORM** no cancela el formulario posterior, sino que en lugar de ello produce un formulario en blanco en la cara posterior de cada página. Si desea cancelar el formulario posterior, use 0 SETMAXBFORM.

### Sintaxis

```
Form_ref SETBFORM
```

```
Form_ref planenumber SETBFORM
```

```
Form_ref [ c1 c2 ... cn ] SETBFORM
```

```
Form_ref planenumber [ c1 c2 ... cn ] SETBFORM
```

Se puede usar un archivo EPS, JPEG o TIFF como formulario mediante el comando **CACHE** después del nombre de recurso. Consulte **CACHE** para obtener más información.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **SETFORM**
- **SETFPATH**
- **SETMAXBFORM**
- **CACHE**

## SETBIDI

**SETBIDI** se usa para configurar la función de transformación BIDI. Este comando reemplaza el comando obsoleto **SETARB**.

### Sintaxis

```
ctx_table mrg_table (bi-directional index) (left-to-right index) SETBIDI
```

### Donde:

<b>ctx_table</b>	es una definición de análisis de contexto definida por <b>BEGINARBT/ENDARBT</b>
<b>mrg_table</b>	es una definición de combinación definida por <b>BEGINARBM/ENDARBM</b>
<b>(bi-directional index)</b>	es un índice de fuente, definido por <b>INDEXFONT</b> , para ser utilizado por BIDI con la opción 1
<b>(índice de izquierda a derecha)</b>	es un índice de fuente, definido por <b>INDEXFONT</b> , para ser utilizado por BIDI con la opción 1

### Ejemplos

```
ARB1256_2 ARB1256_M1 (A) (E) SETBIDI
```

Una configuración predeterminada se codifica al final del archivo de configuración bidireccional en `xgf/src/arb.def`

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **BIDIRECCIONAL**
- **BEGINARBT**
- **ENDARBT**
- **FCALL**
- **MOVETO**
- **SCALL**
- **ENDARBM**

## SETBUFSIZE

El comando **SETBUFSIZE** define el tamaño del búfer de línea o de registro de los archivos de datos que son procesados por **STARTLM** o **STARTDBM**.

### Sintaxis

```
size SETBUFSIZE
```

```
size (F) SETBUFSIZE
```

### Donde:

<b>tamaño</b>	define la longitud máxima de línea. Utilice esta sintaxis para procesar los archivos de líneas que terminan con los delimitadores de línea <b>LF</b> , <b>CR</b> o <b>CRLF</b> .
<b>tamaño (F)</b>	define una longitud de registro fija. Use esta sintaxis para procesar archivos compuestos por registros de longitud fija sin delimitadores como <b>LF</b> , <b>CR</b> o <b>CRLF</b> .

El valor predeterminado se define en el archivo `/usr/xgf/src/xgf.def`.



Nota: Si el tamaño del búfer es demasiado pequeño, el trabajo se cancela con el mensaje de error de comprobación del rango en la línea de lectura.



Consejo: Configure el valor predefinido en el archivo `/usr/xgf/src/xgf.def` según el valor máximo requerido por las aplicaciones como, por ejemplo, 3000 **SETBUFSIZE**. El valor predeterminado es 2000.

### Modos

○Este comando es aplicable en modo de línea y de base de datos.

### Comandos relacionados

Ninguno

## SETCJKENCMAP

**SETCJKENCMAP** define la tabla usada para obtener la codificación de una fuente del nombre de la misma si esta se selecciona con **SETFONT** o **INDEXFONT**. El comando **SETCJKENCMAP** se utiliza con fuentes de varios bytes.

Una tabla predeterminada se define en el archivo configuración `xfp/src/cjk.def`.

### Sintaxis

```
[ (guess string1) /encoding-name1 /wrapping-rules1 /v2h_cvname1
(guess string2) /encoding-name2 /wrapping-rules2 /v2h_cvname2
....
] switch SETCJKENCMAP
```

### Donde:

(guess stringX)

/encoding-nameX

La codificación asociada a esa fuente. Seleccione el nombre de la codificación en la siguiente lista:

- /ASCII
- /SJIS
- /EUC-JP
- /EUC-TW
- /GB18030
- /EUC-CN
- /EUC-KR
- /Johab
- /UHC
- /BIG5
- /GBK
- /EUC-KRpc
- /EUC-CNpc
- /BIG5pc
- /UTF8
- /UTF16
- /UTF32
- /UCS2
- /ISO-2022

wrapping-rulesX

El nombre de la tabla de ajuste definida mediante el comando **SETCJKRULES**. El nombre de la tabla de ajuste no está restringido. Una sola tabla puede ser referenciada por diversas codificaciones.

**/v2h\_cvnameX**

El nombre de la tabla de conversión definida mediante el comando **SETV2H-CONV**. El nombre de la tabla de conversión no está restringido. Una sola tabla puede ser referenciada por diversas codificaciones.

La tabla solo se utiliza con las fuentes CJK: /FontType=0. La tabla no se utiliza para fuentes Roman.

**switch**

Opcional. Si es igual a /INI se crea una nueva tabla. Si se omite, las entradas se agregan en la parte superior de la tabla actual.



Nota:

- El comando **SETCJKENCMAP** principalmente está ideado para usarse en el archivo de configuración de CJK `xgf/src/cjk.def`. Aunque **SETCJKENCMAP** se puede utilizar directamente en un trabajo VIPP®, no se recomienda.
- Las búsquedas en la tabla de conversión se realizan de arriba abajo. La primera entrada coincidente se guarda. Por ejemplo, **GB-EUC** se debe situar antes que **EUC**. La última entrada de la tabla es el valor prefijado, por lo que la cadena de estimación de la entrada debe estar vacía.

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

- **SETFONT**
- **INDEXFONT**
- **SHP** y **SHp**
- **SHMF**, **SHMf** y **SHmf**

## SETCJKRULES

**SETCJKRULES** define la lista de caracteres que pueden influir en el ajuste de texto con el comando **SHF** de una determinada familia de codificación.

Una tabla predeterminada se define en el archivo configuración `xgf/src/cjk.def`.

### Sintaxis

```
[ /wrapping-rules1
  [ opening_char_list1 ]
  [ closing_char_list1 ]
  [ punct_char_list1 ]
  /wrapping-rules2
  [ opening_char_list2 ]
  [ closing_char_list2 ]
  [ punct_char_list2 ]
  ...
] SETCJKRULES
```

### Donde:

<b>/wrapping-rulesX</b>	es un nombre no restringido que puede utilizarse como tercer parámetro en una entrada del comando <b>SETCJKENCMAP</b> .
<b>opening_char_listX</b>	es una lista de cadenas que representan el conjunto de caracteres de apertura asociados a una determinada codificación.
<b>closing_char_listX</b>	es una lista de cadenas que representan el juego de caracteres contiguos o de cierre asociados a una determinada codificación.
<b>punct_char_listX</b>	es una lista de cadenas que representan el conjunto de caracteres de puntuación asociados a una determinada codificación.  Xerox recomienda la notación hexadecimal para estos caracteres, un valor entre paréntesis angulares de cierre en cada lista.



Nota: Este comando principalmente está ideado para usarse en el archivo de configuración de CJK, `xgf/src/cjk.def`. Aunque se puede utilizar directamente en un trabajo VIPP®, no se recomienda. Para obtener un ejemplo del comando, consulte el archivo `xgf/src/cjk.def`.

### Ejemplos



```

/SJIS
% Opening
[
<816f> <8165> <8167> <8171> <8173> <a2> <8175> <8177> <8179>
<816b> <8197> <40> <8198> <81a7> <8194> <23>
]
% Contiguous/Closing
[
<21> <8149> <3f> <8148> <3a> <8146> <3b> <8147> <29> <816a>
<5d> <816e> <7d> <8170> <818c> <8166> <8168> <818d> <8172>
<8174> <a3> <8176> <8178> <817a> <816c>
]
% Punctuation
[ <8141> <8142> <2c> <8143> <2e> <8144> ]
/UTF8
% Opening
[
<7b> <efbd9b> <e28098> <e2809c> <e38088> <e3808a> <efbda2>
<e3808c> <e3808e> <e38090> <e38094> <efbca0> <40> <c2a7>
]
% Contiguous/Closing
[
<21> <efbc81> <3f> <efbc9f> <3a> <efbc9a> <3b> <efbc9b> <29>
<efbc89> <5d> <efbc9d> <7d> <efbd9d> <e280b2> <e28099> <e2809d>
<e280b3> <e38089> <e3808b> <efbda3> <e3808d> <e3808f> <e38091>
]
% Punctuation
[ <e38081> <e38082> <2c> <efbc8c> <2e> <efbc8e> ]
] SETCJKRULES

```

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

SETFONT, INDEXFONT, SHP y SHp

## SETCOL

El comando **SETCOL** asigna un nombre, referido como un **Colorkey**, a una definición de color. Una definición de color puede ser de uno de los tipos siguientes:

- Escala de grises
- RGB
- CMYK
- Gradiente es un conjunto de hasta tres colores que se aplican gradualmente a un elemento gráfico como, por ejemplo, un cuadro, un círculo, un polígono o una prueba
- Espacio de color de separación CMYK, un subconjunto de componentes CMYK

### Sintaxis

```

/Colorkey [ c m y k ] SETCOL      % for CMYK color definition
/Colorkey [ r g b ] SETCOL        % for RGB color definition
/Colorkey g SETCOL                % for grayscale color definition
/Colorkey [ color1 color2 color3 dispcode tint direction ] SETCOL
                                   % for gradient color definition

/Colorkey [ c /SCS_C ] SETCOL
/Colorkey [ m /SCS_M ] SETCOL
/Colorkey [ y /SCS_Y ] SETCOL
/Colorkey [ k /SCS_K ] SETCOL
/Colorkey [ c m /SCS_CM ] SETCOL
/Colorkey [ c y /SCS_CY ] SETCOL
/Colorkey [ c k /SCS_CK ] SETCOL
/Colorkey [ m y /SCS_MY ] SETCOL
/Colorkey [ m k /SCS_MK ] SETCOL
/Colorkey [ y k /SCS_YK ] SETCOL
/Colorkey [ c m y /SCS_CMY ] SETCOL
/Colorkey [ c m k /SCS_CMK ] SETCOL
/Colorkey [ c y k /SCS_CYK ] SETCOL
/Colorkey [ m y k /SCS_MYK ] SETCOL
(spotcolorname) [ c m y k ] SETCOL      % spot color definition
/Colorkey [ v /SCS_V ] SETCOL           % standalone Clear Dry Ink
/Colorkey [ c m y k v /SCS_CMYKV ] SETCOL % Clear Dry Ink combined
                                           % with a CMYK color

/Colorkey [ (spotcolorname) v /SCS_SV ] SETCOL % Clear Dry Ink combined
                                           % with a spot color

/Colorkey [ c m y k o /SCS_CMYKO ] SETCOL % color space key to support Orange 5th color
                                           %Extended Gamut combined with CMYK
/Colorkey [ c m y k b /SCS_CMYKB ] SETCOL % color space key to support Blue 5th color
                                           % Extended Gamut combined with CMYK

/Colorkey [ c m y k g /SCS_CMYKG ] SETCOL % color space key to support Green 5th color
                                           % Extended Gamut combined with CMYK

/Colorkey [ v /SCS_MICR ] SETCOL         % color space key to support the "Xerox MICR"
                                           % spot color.

```

Donde:

<b>/Colorkey</b>	es el nombre del <b>Colorkey</b> . El nombre es definido por el usuario y no debe ser una palabra reservada.
<b>[c m y k]</b>	es una definición de color CMYK cian, magenta, amarillo, negro. Son necesarios cuatro valores reales en la matriz Corchetes. Cada valor debe estar en el rango de 0 a 1.
<b>[r g b]</b>	es una definición de color RGB rojo, verde, azul. Son necesarios tres valores reales en la matriz Corchetes. Cada valor debe estar en el rango de 0 a 1.
<b>g</b>	es una definición de color de nivel gris. Es un valor real de 0 a 1. 0 es negro y 1 es blanco.
<b>v</b>	es un número entero entre 0 y 1 que representa la cobertura de tintas especiales de Xerox, de 0 a 100 %.
<b>color1/color2/color3</b>	son 3 Colorkeys distintas que se utilizan para pintar, de forma progresiva, de un color a otro.
<b>dispcode</b>	<p>es el código de envío; un código de 3 dígitos que define cómo se extiende la graduación en el relleno. Cada dígito tiene un peso de 0 o 1 a 9.</p> <p>El <b>primer dígito</b> (1–9) define el peso de la gradación del color 1 al color 2.</p> <p>El <b>segundo dígito</b> (0-9) define cuánto tiempo se mantiene el color 2 inalterado en el medio, útil para efectos de borde.</p> <p>El <b>último dígito</b> (0-9) define el peso de la gradación del color 2 al color 3. Si es igual a cero, solo se usan los colores 1 y 2.</p>
<b>tono</b>	<p>es un número entre 0 y 2 que se usa para aplicar un tono a toda la graduación:</p> <p><b>0–1</b>: tonos claros de blancos a neutros</p> <p><b>1</b> color normal.</p> <p><b>1–2</b>: tonos oscuros de neutros a negros</p>
<b>direction</b>	<p>es una de:</p> <p><b>/H</b> la graduación se aplica horizontalmente</p> <p><b>/V</b> la graduación se aplica verticalmente</p>
<b>spotcolorname</b>	es el nombre de un color directo definido en el DFE de la impresora. Los valores CMYK asociados en la instrucción <b>SETCOL</b> solo se usan si no se ha definido ningún color directo con ese nombre en el dispositivo. Esos colores no se ven afectados por el proceso de administración del color posiblemente activo en el RIP de destino. Si spotcolorname se combina con tintas especiales de Xerox, deberá haberse definido anteriormente con una instrucción <b>SETCOL</b> de color de punto de tramado.

Se puede usar un color con relleno degradado como **FillColorkey** para definir una **GEPkey** o como un parámetro **/ColorTable** de **DRAWBAR** o **DRAWPIE**.

Un relleno degradado también puede sustituir a una **GEPkey** como parámetro de un comando **DRAWBx**, **DRAWPOL** o **SHx**.



Nota: Las tintas especiales 100 % independientes de colores predefinidos siguientes se han agregado al `/src/xgf.gep`:

`/CLEAR [ 1 /SCS_V ] SETCOL`. Un nuevo parámetro **/ClearSubst** ha sido agregado para configurar la representación en plataformas sin tóner de tinta especial de Xerox. Consulte la sección Parámetros para obtener más detalles.

## Limitaciones

Las definiciones de relleno degradado no son aplicables en todas las instancias en las que se puede especificar un color o una **GEPkey**. En particular, no se puede aplicar a **SETTxC**, **INDEXCOLOR**, **SETPIF/INDEXPIF**, **SETFRAME**, **SETZEBRA**, **MOVEH** y **RPE**. No se pueden combinar con tramas.

## Ejemplos

```
/RB_BLUE1 [ DBLUE MBLUE LBLUE 252 1 /V ] SETCOL
/RB_GREEN1 [ DGREEN LGREEN DGREEN 505 1 /V ] SETCOL
/RB_RED1 [ DRED RED LRED 282 .7 /V ] SETCOL
/RB_BLUE_RED1 [ MBLUE LBLUE MRED 904 1 /V ] SETCOL

/GEP1 0 WHITE 0 RB_BLUE1 SETGEP
228 3124 2061 150 GEP1 50 DRAWBR

/GEP1 0 WHITE 0 RB_GREEN1 SETGEP
228 2836 2061 150 GEP1 DRAWB

228 2561 2061 150 RB_RED1 DRAWB

/GEP1 5 GREEN 0 RB_BLUE_RED1 SETGEP
228 2252 1039 600 GEP1 DRAWC

/NAGD 85 120 SETFONT
(Sample Text) 0 GEP1 0 SHX

/CLEAR50 [ .5 /SCS_V ] SETCOL

/CYANCLEAR [ 1 0 0 0 1 /SCS_V ] SETCOL

/PAN200CLEAR [ (PANTONE 200 CS) 1 /SCS_SV ] SETCOL
```

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

- **SETTxC**
- **INDEXCOLOR**
- **SETGEP**
- **SETTRAN**

## SETCOLWIDTH

El comando **SETCOLWIDTH** establece la anchura de la columna para su uso por comandos siguientes con la opción justificar.

### Sintaxis

```
colwidth SETCOLWIDTH
```

### Donde:

**colwidth** es la anchura de la columna en unidades actuales.

El valor predeterminado se define en el archivo `/usr/xgf/src/xgf.def`.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- COLW
- SHJ y SHj
- SHMF, SHMf y SHmf
- SHP y SHp
- SETLKF

## SETCYCLECOPY

El comando **SETCYCLECOPY** establece el número de copias del ciclo. El ciclo de copias es una alternativa al mecanismo de copia múltiple estándar que permite la especificación de los intervalos de formularios, materiales de impresión y otros elementos de RPE como se describe en los comandos relacionados.

### Sintaxis

```
copynumber SETCYCLECOPY
```

### Ejemplos

Este ejemplo imprime tres copias de cada página, y usa FORM\_A en las copias 1 y 2, y FORM\_B en la copia 3.

```
COLLATE_off
```

```
3 SETCYCLECOPY
```

```
(FORM_A.frm) [1 2] SETFORM
```

```
(FORM_B.frm) [3] SETFORM
```

- El archivo de datos se pone en la cola de forma temporal. En impresoras descentralizadas, cuando se usa el modo de clasificación **COLLATE\_on** y cuando no hay ningún disco local disponible, el archivo de datos se pone en la cola de forma temporal en el disco local o en la memoria.
  - Si se coloca en la cola del disco, el comando **SPOOLNAME** se usa para definir la ruta al archivo de cola temporal que se encuentra de manera predefinida en `/usr/xgf/src/xgfunix.run` o `x:\xgf\src\xgfdos.run`.
  - Cuando el archivo de datos se manda a la cola de la memoria, y si la memoria se agota, es posible que los trabajos de gran tamaño se anulen. El límite de memoria es probable que tenga un valor entre 1 y 3 MB, según la cantidad de memoria disponible.
- Posición del comando de clasificación: Todos los comandos de clasificación antes del comando **SETCYCLECOPY** deben estar codificados.

Coloque un comando **SETMAXCOPY** al principio del JDT o XJT para especificar el número máximo de copias si se usa **SETCYCLECOPY** para cambiar de forma dinámica el número de copias en:

<b>Modo de línea</b>	ya sea mediante NMP o en un procedimiento <b>BEGINPAGE</b> o <b>PROCESSDJE</b>
<b>Modo de base de datos</b>	en una DBM
<b>Modo XML</b>	en una BTA

Inicializa las estructuras de memoria y aplica la colocación en la cola del archivo de datos de forma preventiva en impresoras de escritorio. En estos casos, **SETCYCLECOPY** ejecuta un **CHKPOINT** implícito, no es necesario codificar **CHKPOINT**.



Nota: No utilice un comando **REPEAT** o **RSAVE** después del comando **SETCYCLECOPY**, porque de lo contrario pueden producirse resultados impredecibles.



Consejo: Combine este comando con otros como **DUPLEX\_on** o **SETMULTIUP** para llevar a cabo funciones avanzadas de varias copias.

Si **SETCYCLECOPY** se usa en modo nativo con **SETCYCLECOPY**, el comando **CHKPOINT** es obligatorio al final del trabajo.

Si selecciona **SETCYCLECOPY** en un JDT en modo de base de datos, es obligatorio colocarlo como último comando del JDT. Así se garantiza el funcionamiento correcto en los DFE sin un flujo de entrada recolocable.

Este comando no restaura las variables antes de cada pasada de copia. Dado que los valores de las variables permanecen inalteradas de un paso al siguiente, el usuario es responsable de reinicializarlas como corresponda.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **CHKPOINT**
- **COLLATE\_dbm**
- **COLLATE\_off**
- **COLLATE\_on**
- **COPYRANGE**
- **SETBFORM**
- **SETDLFILE**
- **SETFORM**
- **SETMEDIA**
- **SPOOLNAME**
- **REPEAT**



## SETDATE

**SETDATE** establece temporalmente las variables según la fecha proporcionada. Puede utilizarse en combinación con **GETDATE**, **SHIFTDATE** y **DAYS** para calcular distintos valores de variación de la fecha.

### Sintaxis

```
[ YYYY MO DD ] SETDATE
```

```
[ YYYY MO DD HH MM SS ] SETDATE
```

```
days SETDATE
```

### Donde:

<b>AAAA</b>	es el año (> 1970). Debe ser un número entero.
<b>MO</b>	es el mes (1–12). Debe ser un número entero.
<b>DD</b>	es el día (1–31). Debe ser un número entero.
<b>HH</b>	es la hora (0–23). Debe ser un número entero.
<b>MM</b>	son los minutos (0–59). Debe ser un número entero.
<b>SS</b>	son los segundos (0–59). Debe ser un número entero.
<b>días</b>	representa una fecha como el número de días desde 1.1.1970. Es un número entero.

Las variables de fecha se restauran según la fecha del día mediante una inicialización de página o un **GETDATE** explícito. Por esta razón, debe situarse lo más cerca posible al comando que usa las variables de datos, o las variables deben ser capturadas en una cadena justo después de **SETDATE** mediante **VSUB** y **SETVAR**.

Con la primera y la tercera sintaxis, la hora actual permanece inalterada.

### Ejemplos

Este ejemplo asume que Year, Month y Day son campos de un archivo de DBF.

```
[ Year Month Day ] SETDATE
/VARdate1 ($$D_DWL. $$D_MO./$$D_DD./$$D_YYYY.) VSUB SETVAR
/VARstart [ Year Month Day ] DAYS SETVAR
VARstart+'55 SETDATE
/VARdate2 ($$D_DWL. $$D_MO./$$D_DD./$$D_YYYY.) VSUB SETVAR
GETDATE
(From $$VARdate1. to $$VARdate2. you will be given the opportunity to visit our new shopping
center and purchase any article with a 50% discount.) VSUB 0 SHP
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**, **SHIFTDATE**, **DÍAS**

## SETDBSEP

**SETDBSEP** establece el separador de campos que se usarán en el modo de base de datos, o con entradas RPE con la opción /FN. Los comandos **STARTDBM** posteriores utilizan este separador de campos para escanear campos en un archivo de la base de datos. Para obtener más información, consulte Flujos de datos de VIPP® en la *Guía del usuario de FreeFlow VI Compose*. El valor predeterminado : se define en el archivo `/usr/xgf/src/xf.def`.

Para cambiar el valor de separadores, cambie el archivo, que aplica el nuevo separador a todas las aplicaciones, o use el comando **SETDBSEP** en el JDT o antes del comando **STARTDBM**.

### Sintaxis

```
(field separator) SETDBSEP
```

### Donde:

**Separador de campos** es una cadena alfanumérica de uno o más caracteres.

### Ejemplos

Este ejemplo define : como separador de campos.

```
( :) SETDBSEP
```

Los valores hexadecimales u octales pueden utilizarse para definir el separador de campos. Utilice este ejemplo para configurar el carácter de tabulación como el separador de campos.

```
<09> SETDBSEP % Hex value
```

```
(\011) SETDBSEP % Octal value
```



Importante: No coloque este comando en el archivo maestro de base de datos.

### Modos

○Este comando es aplicable en modo de línea y de base de datos.

### Comandos relacionados

- **GETINTV**
- **STARTDBM**
- **FROMLINE**
- **RPEKEY**

## SETDLFILE

**SETDLFILE** activa el modo de varias copias con una lista de distribución para el trabajo actual.

Como alternativa a **SETCYCLECOPY**, se pueden obtener varias copias de un documento con la función de la lista de distribución. Esto se conoce como “etiquetado de conjuntos”.

**SETDLFILE** asocia una lista de distribución y un JDT para producir portadas delante de cada juego de documentos. Estas páginas se crean al procesar el archivo de listas de distribución (DL) mediante el JDT asociado de la misma manera que las procesa **STARTLM**. Por lo tanto, el número de copias realizadas es igual al número de páginas del archivo DL. En este caso, no puede utilizarse **SETCYCLECOPY**.

**SETDLFILE** se utiliza en un archivo de modo nativo, en un JDT o en un archivo de envío.

El archivo DL debe encontrarse en una de las bibliotecas referenciadas por **SETMPATH**.

### Sintaxis

```
(DL file name) (jdtname) SETDLFILE
```

### Ejemplos

Este ejemplo imprime el archivo `dist23.lst` mediante el JDT **dist.jdt**. El archivo `report57.dat` se imprime entre cada portada mediante el JDT `report.jd`. El JDT `dist.jdt` contiene un formulario de llamada y diseño de página para las portadas.

```
%!
```

```
(dist23.lst) (dist.jdt) SETDLFILE
```

```
(report57.dat) SETLMFILE
```

```
(report.jdt) STARTLM
```

### Modos

Este comando es aplicable en modo nativo y el modo de línea.

### Comandos relacionados

- **SETCYCLECOPY**
- **SETJPATH**
- **SETMPATH**
- **STARTLM**

## SETENCODING

**SETENCODING** se usa para recodificar una fuente PostScript base a través de la asociación de una fuente base y una tabla de codificación.

Una fuente PostScript base es una recopilación de caracteres, el conjunto de caracteres referenciado por nombres, el nombre del carácter o nombre de glifo. En la terminología de fuentes, una fuente base es una fuente PostScript con un formato tipo 1, como por ejemplo Times Roman, Courier, Helvetica y Symbol, fuentes definidas por el usuario tipo 3 o formato de tipo 42 (TTF).

Los datos de texto son un flujo de códigos de caracteres según una determinada codificación, por ejemplo, ASCII, ISO 8859-1, UTF8, etc. Un flujo de datos puede ser un solo byte, por ejemplo, ISO 8859-1 o varios bytes, por ejemplo, UTF8. Las fuentes base, que son fuentes de un solo byte, procesan los datos de un solo byte; los datos de varios bytes, por ejemplo, los compuestos y los que disponen de una clave CID procesan datos de varios bytes.

Una tabla de codificación es un archivo que se usa para asignar una lista de códigos de caracteres a los nombres de caracteres correspondientes según una determinada configuración. Una fuente PostScript base contiene una tabla de codificación prefijada, por lo general, StandardEncoding. La recodificación es un proceso para sustituir la tabla de codificación predefinida con una que coincida con el flujo de datos que se imprimirá. Si **SETENCODING** se usa para recodificar una fuente base, se crea una nueva fuente que es una copia de la fuente base pero con otra tabla de codificación.

La recodificación de una fuente base puede ser necesaria si un carácter no válido se muestra o imprime en vez del carácter esperado.



Nota: Es probable que se produzca un carácter no válido si se produce alguna de las siguientes opciones:

- Un carácter en su documento no se incluye en el conjunto de caracteres de la fuente actual. La fuente actual es la fuente especificada por el comando **SETFONT** o **INDEXFONT**.
- El código de caracteres en su documento asigna el nombre de carácter incorrecto en la fuente actual.
- El código de caracteres en su documento asigna el nombre de carácter incorrecto en la fuente actual.

Para volver a codificar una fuente base, coloque uno de los siguientes comandos **SETENCODING** en el envío de trabajos o el archivo de recursos. A continuación, haga referencia a la nueva fuente recodificada, en lugar de a la fuente base en un comando **SETFONT** o **INDEXFONT** posterior.



Nota:

- Para la recodificación que se efectúa en el primer comando **SETFONT** o **INDEXFONT**, la recodificación de fuentes no se produce para todas las fuentes a la vez en la primera aparición de **SETENCODING**. En su lugar, la recodificación de fuentes se realiza solamente la primera vez que aparece el comando **SETFONT** o **INDEXFONT**. Esta acción permite disminuir el consumo de memoria y el tiempo de arranque de VI Compose.
- Para fuentes de varios bytes, la lista de fuentes básicas **SETENCODING**, **SETENCODING** solo se aplica a fuentes básicas de un solo byte. Las fuentes de varios bytes no pueden aparecer en la lista de fuentes básicas **SETENCODING** porque usan su propio mecanismo de codificación, por ejemplo, recursos de CMAP de PostScript. Para utilizar una fuente de varios bytes, especifique el nombre de la fuente PostScript con el comando **SETFONT** o **INDEXFONT**.

### Sintaxis

```
(font list file) (encoding table) SETENCODING
```

```
[font_entry_1, font_entry_2, ..., font_entry_N] (encoding table) SETENCODING
```

Donde:

**archivo de lista de fuentes**

es un archivo que contiene una lista de fuentes según lo descrito más abajo.

**font\_entry\_X**

es una instrucción en cualquiera de los cuatro formatos siguientes:

```
/new_font_name /PS_font_name
```

```
/new_font_name (font_file_name)
```

```
/new_font_name [ /PS_font_name (AFM_file_name) ]
```

```
/new_font_name [ (font_file_name) (AFM_file_name) ]
```

Donde:

**new\_font\_name** es el nuevo nombre para la fuente recodificada.

**PS\_font\_name** es el nombre de la fuente PostScript residente original que se recodificará.

**font\_file\_name** es el nombre del archivo que contiene la fuente que se recodificará.

Si se usa la sintaxis con **font\_file\_name**, la fuente es procesada por VI Compose como un recurso VIPP® y no necesita volver a ser instalada como fuente residente de la impresora. Como tal, la fuente se incorporará a cualquier contenedor, VPC, ProofPrint, etc. producido por las diversas herramientas de Variable Information Suite. El archivo de la fuente debe almacenarse en una de las bibliotecas referenciadas por SETMPATH o SETPPATH en el modo de proyecto. Solo se admiten los formatos de fuente pfa y pfb. No es preciso especificar el nombre de la fuente PostScript original; VI Compose lo extrae del archivo de la fuente.

**AFM\_file\_name** es un archivo de Adobe Font Metrics (AFM) que proporciona la información de espaciado. Obtenga más información de espaciado más abajo.

**tabla de codificación**

es el nombre de archivo que contiene una lista de códigos de caracteres y pares de nombres de caracteres. Una tabla de codificación solamente puede contener caracteres, pares de nombres/códigos que requieren volver a realizar la codificación. Los caracteres que no están presentes en la tabla de codificación se copian en la tabla de codificación ASCII PostScript (StandardEncoding).

La tabla de codificación se puede sustituir por un valor nulo. En ese caso, la tabla de codificación predefinida se preserva, y el propósito de la recodificación es solo proveer un nombre de fuente más corto para la comodidad de los usuarios.

Los nombres de caracteres en una tabla de codificación deben estar en la fuente base que se está recodificando. Si no se encuentra ningún nombre coincidente, seleccione una fuente base distinta. Sin embargo, los códigos de caracteres pueden ser valores de un byte o de varios bytes. Por lo tanto, aunque SETENCODING solo se aplica a fuentes base de un solo byte, la fuente de recodificación puede ser una fuente base si la tabla de codificación solo contiene códigos de carácter de un solo byte o fuentes compuestas si la tabla de codificación contiene códigos de caracteres de varios bytes.

Además /STARTFF y /ENDFF pueden ser usados para encapsular hasta 4 entradas de fuentes para definir una familia de fuentes:

```
/STARTFF
  font_entry_reguar      (for regular font)
  font_entry_bold       (for bold font)
  font_entry_italic      (for italic font)
  font_entry_boldItalic  (for boldItalic font)
/ENDFF
```

Donde:

<b>STARTFF</b>	marca el principio de una familia de fuentes
<b>ENDFF</b>	marca el final de una familia de fuentes

El espaciado hace referencia a la capacidad de ajustar el nivel de espacio entre los caracteres al imprimir un bloque de texto con una fuente determinada. Si se requiere el espaciado, cree una entrada de fuentes mediante el formato `AFM_file_name`, para establecer un vínculo entre una determinada fuente y un archivo AFM y use los comandos **SETKERN** o **INDEXKERN**.

VI Compose se suministra junto a un conjunto de archivos de espaciado genéricos (.afk) que son un subconjunto de archivos AFM. Estos archivos contienen la información de espaciado genérica que se puede usar si un determinado archivo AFM de una fuente no está disponible. Los archivos AFK solo existen en el directorio `xgf/encoding` y son referenciados en `xgf/encoding/fontlist` si un usuario activa el espaciado con una de las fuentes de VIPP® indicadas. Hay dos conjuntos de archivos prefijados, uno para fuentes serif (`genericssr*`) y otro para fuentes sans serif (`genericss*`). Los archivos AFK pueden utilizarse con cualquier fuente. Seleccione una de estas en función del tipo, serif o sans serif y los atributos de la fuente, normal, negrita, cursiva, negrita-cursiva. Sin embargo, dado que son genéricos, el resultado no puede ser totalmente preciso. Para obtener resultados precisos, debe utilizarse el archivo AFM original proporcionado por Adobe.

Para obtener información adicional sobre espaciado, consulte [Espaciado](#).

A continuación figura un ejemplo del uso de los archivos AFK con la familia de fuentes Helvetica sans serif:

```
/STARTFF
  /NHE[/Helvetica (genericss.afk)]
  /NHEB[/Helvetica-Bold (genericss_bo.afk)]
  /NHEO[/Helvetica-Oblique (genericss_it.afk)]
  /NHEBO[/Helvetica-BoldOblique (genericss_bi.afk)]
/ENDFF
```

**Nota: Las fuentes de varios bytes y la lista de fuentes básicas SETENCODING**

**SETENCODING** solo se aplica a las fuentes básicas de un solo byte. Las fuentes de varios bytes nunca deben aparecer en la lista de fuentes básicas **SETENCODING** porque usan su propio mecanismo de codificación, por ejemplo, recursos de CMap de PostScript. Para utilizar una fuente de varios bytes, especifique el nombre de la fuente PostScript con el comando **SETFONT** o **INDEXFONT**.

**Ubicación de nombres de caracteres consecutivos**

Los nombres de caracteres consecutivos pueden colocarse después del código de carácter inicial.

Por ejemplo

```
16#80 /Adieresis /Aring /Ccedilla
is equivalent to:
16#80 /Adieresis
16#81 /Aring
16#82 /Ccedilla
```

Para volver a codificar datos de un byte en una fuente base, use códigos de carácter entre 0 y 255 (16#00 - 16#FF).

Por ejemplo:

```
16#80 /Adieresis      /Aring      /Ccedilla      /Eacute
16#84 /Ntilde        /Odieresis   /Udieresis     /aacute
16#88 /agrave        /acircumflex /adieresis     /atilde
```

Para agregar datos UTF-8 de varios bytes a una fuente recodificada, use los códigos de caracteres en uno de los intervalos de varios bytes UTF-8:

<b>1 byte</b>	<b>16#00 - 16#7F</b>
<b>2 bytes</b>	<b>16#C080 - 16#DFBF</b>
<b>3 bytes</b>	<b>16#E08080 - 16#EFBFBF</b>
<b>4 bytes</b>	<b>&lt;F0808080&gt; - &lt;F7BFBFBF&gt;</b>

Por ejemplo:

```
16#27 /quotesingle
16#C2A1 /exclamdown /cent /sterling /currency /yen /brokenbar
16#E28098 /quoteleft /quoteright /quotesinglbase /quotereversed
```

Tenga en cuenta que los códigos de caracteres en un intervalo de 4 bytes deben expresarse como cadenas hexadecimales. Por ejemplo:

```
<F090A080> /cypriot_syllable_A
```

Debe tenerse en cuenta que los intervalos de varios bytes UTF-8 contienen miles de caracteres, y solo se imprimen los códigos de caracteres que pueden asignarse a un nombre de carácter en la fuente base, normalmente mediante el conjunto de caracteres Roman estándar. Otros códigos de carácter asumen el signo de interrogación (?) como valor prefijado.

Se proporciona una tabla de codificación UTF-8 predefinida en `xgf/src/encoding/utf8`. A continuación figura un



ejemplo sobre cómo utilizarlo:

```
[ /CR-UTF8 /Courier
  /HE-UTF8 /Helvetica
  /TM-UTF8 /Times-Roman ] (utf8) SETENCODING
```

El uso de -UTF8 en el nuevo nombre de fuente no es obligatorio pero se recomienda encarecidamente para beneficiarse del reconocimiento de límites de caracteres en comandos VIPP®, **RPE/GETFIELD/ SETRCD/SETPCD/ GETINTV**. La causa es que -UTF8 está registrado como cadena de suposición en `cjk.def`.

Para obtener más información sobre las tablas de codificación y las listas de fuentes, consulte Listas estándar, tablas, claves y atributos y Recursos de VIPP® en la *Guía del usuario de FreeFlow VI Compose*.

El archivo `/usr/xgf/src/xgf.def` contiene dos instrucciones **SETENCODING** que proporcionan una lista predefinida de fuentes recodificadas.

### Ejemplos

```
(fontlist) (sun8) SETENCODING
[ /EHE /Helvetica ] (ebcdic) SETENCODING
[ /CD128 (mb034.pfb) ] null SETENCODING
[ /FN115F (fn115f.pfa) ] null SETENCODING
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- INDEXFONT
- SETEPATH
- SETFONT
- SETMPATH
- SETPPATH

## SETEPATH

**SETEPATH** define una biblioteca o una lista de bibliotecas en las listas de fuentes y tablas de codificación. Se utilizan las bibliotecas especificadas por **SETENCODING** para localizar listas de fuentes y tablas de codificación. El valor predeterminado se define en el archivo `/usr/xgf/src/xgfunix.run` o `x:xgf\src\xgfdos.run`.

Utilice los comandos **SETEPATH** solo en los archivos `xgfunix.run` o `xgfdos.run`. La agregación de **SETEPATH** a un trabajo VIPP® pone en riesgo la portabilidad de este. La agregación de **SETEPATH** a un proyecto VI pone en riesgo la portabilidad y la organización del proyecto.

### Sintaxis

```
(path to enc. library) SETEPATH
```

```
[ (path to enc. library 1) (path to enc. library 2) ... ] SETEPATH
```

Si se especifica una lista de bibliotecas, como en la segunda sintaxis más arriba, se realizan búsquedas en las mismas en el orden en el que aparecen en la lista.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETENCODING**

## SETFINISHING

Use el comando **SETFINISHING** para seleccionar una función de acabado y una opción de acabado asociada para todo el trabajo, o para subconjuntos del trabajo en los dispositivos que admiten el acabado de subconjuntos.

El comando **SETFINISHING** consolida en un solo comando unificado los comandos relacionados con el acabado, tanto nuevos como antiguos, admitidos por las impresoras de Xerox.

Solo se admiten los comandos **SETFINISHING** en equipos de producción de Xerox. Es posible disponer de niveles limitados de compatibilidad si se utilizan dispositivos no de Xerox. Compruebe que el dispositivo admite comandos **SETFINISHING**. **SETFINISHING** sustituye a los siguientes comandos antiguos, que ya no se utilizan, pero que todavía se admiten para garantizar la compatibilidad con versiones anteriores:

- **STAPLE\_on/off**
- **STAPLEDETAILS**
- **OFFSET\_on/off**
- **BIND\_on/off**
- **BINDDETAILS**

### Sintaxis

```
/Feature /ON SETFINISHING
```

```
/Feature /OFF SETFINISHING
```

```
/Feature (option-spec) SETFINISHING
```



Nota: El comando **SETFINISHING** debe especificarse al principio de la página antes de especificar cualquier comando de marca.

<b>Función</b>	es el nombre de una función de acabado aplicable al dispositivo de acabado de la impresora. Las funciones de acabado que se pueden activar mediante <b>SETFINISHING</b> son:
	<b>/Staple</b>
	<b>/Offset</b>
	<b>/Bind</b>
	<b>/Fold</b>
	<b>/MakeBooklet</b>
	<b>/Punch</b>
<b>/ON</b>	activa la función de acabado en el dispositivo de acabado mediante la última opción seleccionada o la opción predefinida.
<b>/OFF</b>	desactiva la función de acabado en el dispositivo de acabado.

**(option-spec)**

activa la función de acabado en el dispositivo de acabado mediante la opción especificada. El uso de esta sintaxis implícitamente activa la función. No es preciso codificar la opción /ON.

(option-spec) puede codificarse para incluir diversas acciones secundarias separadas por dos puntos, como se muestra a continuación:

```
/Punch (TopPortrait:Two) SETFINISHING.
```

Cualquier opción secundaria omitida asumirá el valor predeterminado del sistema o el de la última opción secundaria especificada.



Nota: Si la función de opción-especificación está activada, puede desactivar la función. Use la sintaxis /OFF o /ON como se muestra en el ejemplo.

La tabla siguiente consta de ejemplos de funciones y opciones disponibles según el dispositivo y para diversas impresoras. Úsela como referencia para la inclusión como valores (opción - espec). Consulte también el manual del dispositivo correspondiente para obtener información sobre las determinadas opciones que no aparecen en la tabla.

FUNCIÓN		OPCIÓN - ESPEC.	61XX	DP 65/75/ 90 2045/ 2060 12/ 5252/ 6060/8000	NU- VERA	FFPS 4110, 4112, 700 Y 770	TODOS LOS FFPS (LAS OP- CIONES DE ACA- BADO DEPEN- DEN DEL DISPOSI- TIVO DE IMPRESIÓN)	EFI EXP4110
Encuaderna- ción		LeftPortrait	X					
		RightPortrait	X					
		Longedge						
		Shortedge						
Desplazamien- to		Desplazamien- to	X	X	X			
Ple- gado	Fol- dEdge	CFoldOutside				X	X	X
		CFoldInside			X	X	X	X
		ZFoldOutside				X	X	X
		ZFoldInside			X	X	X	X
		ZFoldRightHalf				X	X	X

FUNCIÓN		OPCIÓN - ESPEC.	61XX	DP 65/75/ 90 2045/ 2060 12/ 5252/ 6060/8000	NU- VERA	FFPS 4110, 4112, 700 Y 770	TODOS LOS FFPS (LAS OP- CIONES DE ACA- BADO DEPEN- DEN DEL DISPOSI- TIVO DE IMPRE- SIÓN)	EFI EXP4110
		HalfFoldOutside				X		X
		HalfFoldInside				X		X
		BookletSquareFoldAndTrim*				X		
		BookletSaddleSquareFoldAndTrim*				X		
		CFold			X	X	X	X
		ZFold			X	X	X	X
		BiFoldInsideAndTrim*				X		
	Trim- Position	un valor real en puntos				X		
	Squa- reFol- dLevel	Weak2				X		
		Weak1				X		
		Normal				X		
		Strong1				X		
		Strong2				X		
		Off				X		
		* Las opciones TrimPosition y SquareFoldLevel solo se aplican a 3 posibilidades opción-especificación FoldEdge: BookletSquareFoldAndTrim, BookletSaddleSquareFoldAndTrim, BiFoldInsideAndTrim.						
MakeBooklet	BookletFold				X	X	X	X
					X	X	X	X

FUNCIÓN	OPCIÓN - ESPEC.	61XX	DP 65/75/ 90 2045/ 2060 12/ 5252/ 6060/8000	NU- VERA	FFPS 4110, 4112, 700 Y 770	TODOS LOS FFPS (LAS OP- CIONES DE ACA- BADO DEPEN- DEN DEL DISPOSI- TIVO DE IMPRE- SIÓN)	EFI EXP4110
	BookletSad- dleStitch						
<b>Grapas</b>	SinglePortrait	X	X	X	X	X	X
	RightPortrait	X	X	X	X	X	X
	BottomLeftPor- trait				X	X	X
	BottomRight- Portrait				X	X	X
	SingleLandsca- pe	X	X	X	X		X
	RightLandscape	X	X	X	X	X	X
	BottomLef- tLandscape				X	X	X
	BottomRigh- tLandscape				X	X	X
	DualLeftPortrait	X	X	X	X	X	X
	DualRightPor- trait	X	X	X	X	X	X
	+DualTopPor- trait			X	X	X	X
	DualBottom- Portrait			X	X	X	X
	DualLeftLands- cape			X	X	X	X
	DualRigh- tLandscape			X	X	X	X

FUNCIÓN		OPCIÓN - ESPEC.	61XX	DP 65/75/ 90 2045/ 2060 12/ 5252/ 6060/8000	NU- VERA	FFPS 4110, 4112, 700 Y 770	TODOS LOS FFPS (LAS OP- CIONES DE ACA- BADO DEPEN- DEN DEL DISPOSI- TIVO DE IMPRESI- ÓN)	EFI EXP4110
		DualLandscape	X	X	X	X	X	X
		DualBottom- Landscape					X	X
		CenterLeftPor- trait				X		X
		CenterRightPor- trait				X		X
		CenterTopPor- trait				X		X
		CenterBottom- Portrait				X		X
		CenterLef- tLandscape				X		X
		CenterRigh- tLandscape				X		X
		CenterTo- pLandscape				X		X
		CenterBottom- Landscape				X		X
Perfo- ra- ción	Ubica- ción	TopPortrait				X	X	X
		BottomPortrait				X	X	X
		RightPortrait				X	X	X
		LeftPortrait				X	X	X
		TopLandscape				X	X	X
		BottomLands- cape				X	X	X

FUNCIÓN		OPCIÓN - ESPEC.	61XX	DP 65/75/ 90 2045/ 2060 12/ 5252/ 6060/8000	NU- VERA	FFPS 4110, 4112, 700 Y 770	TODOS LOS FFPS (LAS OP- CIONES DE ACA- BADO DEPEN- DEN DEL DISPOSI- TIVO DE IMPRE- SIÓN)	EFI EXP4110
		RightLandscape				X	X	X
		LeftLandscape				X	X	X
	Núme- ro de perfo- racio- nes	Dos				X	X	X
		Tres*				X	X	X
		Cuatro*				X	X	X
						X	X	X
	* Se puede configurar solo un par: Dos y tres, o dos y cuatro.							



Nota: **SETFINISHING** activa el acabado disponible en el dispositivo. Por ejemplo, si el dispositivo no dispone de una opción de grapas en línea, el comando de grapas se pasará por alto. No se pueden controlar dispositivos de acabado externos a través del código VIPP®. La única opción disponible es desplazar la bandeja al final del juego. Es la opción que se debe usar si la acabadora no detecta el trabajo de la bandeja. Si el trabajo consta de un número constante de páginas, la acabadora externa puede a menudo configurarse para que se active en un recuento de páginas determinado.

Consulte la Guía del anexo de PostScript del dispositivo o póngase en contacto con su analista local de Xerox para confirmar las opciones de acabado admitidas por el dispositivo en función de su configuración.

### Ejemplos

El código en el ejemplo activa las grapas para el primer y el último conjuntos, y las desactiva para el resto de conjuntos de un trabajo de varios juegos.



```

/staple (SinglePortrait) SETFINISHING
/staple /ON SETFINISHING
STARTOFSET
. . . Pages of first set
PAGEBRK
/staple /OFF SETFINISHING
STARTOFSET
. . . Pages of second set
PAGEBRK
STARTOFSET
. . . Pages of third and subsequent sets up to last set
PAGEBRK
/staple /ON SETFINISHING
STARTOFSET
. . . Pages of last set
PAGEBRK
%%EOF

```

En el servidor de impresión FreeFlow, este comando solamente hace referencia a la estación interna de grapas. Para configurar una acabadora externa como destino de acabado de subjugos, utilice el comando **SETOBIN**, por ejemplo, (SBM) **SETOBIN**.

Ejemplos de uso de **FoldEdge** con **TrimPosition** y **SquareFoldEdge**:

- /Fold (BookletSquareFoldAndTrim:13.2:Strong1) SETFINISHING
- /Fold (BiFoldInsideAndTrim:13.2) SETFINISHING

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **STARTBOOKLET**
- **ENDBOOKLET**
- **STARTOFSET**
- **ENDOFSET**

## SETFONT

**SETFONT** selecciona y ajusta a escala una fuente. Todos los datos después de este comando se imprimen con la fuente seleccionada hasta que se detecte un nuevo comando **SETFONT** o se invoque un índice de fuentes definido por **SETFONT**.

Para obtener más información, consulte [Espaciado](#)

### Sintaxis

```
/Fontname size SETFONT
/Fontname sizeX sizeY SETFONT
/glossfont_name GLT SETFONT
/microfont_name MPR SETFONT
```

### Donde:

#### Fontname

es una de las siguientes:

El nombre de una fuente elegida en las listas de fuentes VIPP® activadas por **SE-TENCODING** en el archivo `/usr/xgf/src/xgf.def`. Para obtener más información, consulte Listas estándar, tablas, claves y atributos en la *Guía del usuario de FreeFlow VI Compose*.

Un Fontname definido para facilitar el tipo de fuente a otro miembro de la misma familia de fuentes sin tener que volver a especificar el Fontname:

**/~REG** para normal

**/~BLD** para negrita

**/~ITL** para cursiva

**/~ BDI** para negrita cursiva

**/~ CUR** para la fuente actual

Por ejemplo, si la fuente actual es Helvetica, 12 puntos, como se define a través de `/NHE 12 SETFONT`, un **SETFONT** en el que se especifica el Fontname **/~BLD** especial cambia la fuente a Helvetica-Bold. Cuando la fuente actual es Times cursiva, la misma definición de **SETFONT** cambia la fuente a Times negrita. El Fontname especial Fontname **/~CUR** se puede usar para cambiar el tamaño de una fuente y mantener el Fontname sin cambios. La especificación de un tamaño de la fuente como nulo mantiene el tamaño del punto inalterado.

Para obtener más información, consulte [Aplicar atributos para las fuentes](#).

#### tamaño

Un **número entero o real** que especifica el tamaño de la fuente en unidades de 1/72 de pulgada (puntos). Si se da un tamaño igual a 0, la escala de la fuente se ajusta automáticamente según los márgenes y la cuadrícula definidos por **SETMARGIN** y **SETGRID**. En este caso, use una fuente fija como, por ejemplo, Courier.

**null** especifica que el tamaño del punto actual se mantiene inalterado.

<b>sizeX</b>	permite especificar el tamaño de la fuente en la dirección de X en puntos. Los valores son los mismos tal como se define en el tamaño, sin embargo, un valor de 0 no se permite.
<b>sizeY</b>	permite especificar el tamaño de la fuente en la dirección de Y en puntos. Los valores son los mismos tal como se define en el tamaño, sin embargo, un valor de 0 no se permite.
<b>glossfont_name</b>	es el nombre de la fuente brillante utilizada en una aplicación especial de la impresión. Consulte también Impresión especializada con VIC en la <i>Guía del usuario de FreeFlow VI Compose</i> .
<b>microfont_name</b>	es el nombre de la microfuentes utilizada en una aplicación especial de la impresión. Consulte también Impresión especializada con VIC en la <i>Guía del usuario de FreeFlow VI Compose</i> .
<b>GLT/MPR</b>	son variables integradas que proporcionan el tamaño de la fuente y configuran el espaciado de línea correspondiente para una fuente de impresión especial determinada.

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

- INDEXFONT
- SETENCODING
- GLT
- MPR

## SETFORM

**SETFORM** selecciona un formulario para que se imprima en la página actual y en todas las páginas siguientes. Este comando debe especificarse antes de cualquier comando de marca en la página.

Los formularios deben estar codificados en el modo nativo VIPP® o PostScript y deben almacenarse como procedimientos en una de las bibliotecas referenciadas por **SETFPATH**. Se recomienda el uso de la extensión .frm. Los procedimientos están encapsulados entre llaves "{ }".

### Sintaxis

```
Form_ref SETFORM Form_ref númeroplano SETFORM Form_ref [ c1 c2 ... cn ] SETFORM Form_ref
```

Donde:

#### Form\_ref

Alguna de estas opciones:

- **(Formname)** el nombre de un formulario entre paréntesis
- **null** para desactivar el formulario (predeterminado)
- **{ form contents }** código VIPP® entre llaves para formar un pequeño formulario en línea en lugar de una codificación de un archivo externo.
- **[ FormRef1 FormRef2 ... FormRefn ]** una lista de **FormRef** entre corchetes que se utilizará de forma secuencial y cíclica, ciclo de formularios.

#### planenumber

es el índice de planenumber (el valor prefijado es 0). Hace referencia a la capacidad de imprimir diversos formularios uno encima del otro. El valor de planenumber está entre 0 y maxplanenumber -1. El valor prefijado de planenumber es 0.

El valor prefijado para maxplanenumber configurado por **SETMAXFORM** es 1. Esto también determina el orden en el que los formularios se imprimen.

Un formulario con un planenumber de 0 se imprime antes de un formulario con un planenumber igual a 1. Este orden debe realizarse con cuidado debido a que los elementos de PostScript son opacos.

#### [ c1, c2, ... cn ]

Define una selección de intervalo de copia. Se debe utilizar con **SETCYCLECOPY**. La selección de copias específica en qué copias se imprime el formulario. Por ejemplo [ 1 5 ] indica que el formulario solo se imprime en las copias 1 y 5. Si no se especifica, el formulario se imprime en todas las copias.

Se puede usar un archivo EPS, JPEG o TIFF como formulario mediante el comando **CACHE** después del nombre de recurso. Para obtener más información, consulte *CACHE*.

Origen del formulario de los formularios codificados mediante el lenguaje VIPP®:

- La esquina inferior izquierda de la página de manera prefijado o si se codifica **ORIBL**
- La esquina superior izquierda de la página si se codifica **ORITL**

#### Cuadro de límite:

Si SETFORM se usa en combinación con **CACHE** la instrucción de cuadro delimitador **% %BoundingBox: llx lly**

urx ury, si la hay, se utiliza para determinar el tamaño y el recorte de la imagen del formulario. lxx lly se considera el origen de la imagen en relación a la esquina inferior izquierda de la página.

### Ejemplos

```
(form1.frm) SETFORM
(form1.frm) 0 SETFORM
(form2.frm) 1 SETFORM
(copy.frm) 3 [2 3] SETFORM
null SETFORM
null 3 SETFORM
{ PORT 100 3200 MOVETO (image.tif) 1 0 ICALL } SETFORM
[ (form1.frm) (form2.frm) (form3.frm) ] SETFORM
```

En estos ejemplos se utiliza **CACHE SETFORM**:

```
(form1.ps) CACHE SETFORM
(logo.eps) CACHE SETFORM
(image3.tif) CACHE SETFORM
```

Combinar SETFORM, números de plano e instrucciones condicionales para llevar a cabo funciones únicas. En este ejemplo se usa un formulario en línea para colocar otra imagen en la página, en función de un valor en un registro.

```
{ /VAR_FIRST8 1 0 8 GETFIELD } BEGINPAGE
{ x y MOVETO
CASE VAR_FIRST8 {}
  (xxxxxxx) { (image1.tif) 1 0 ICALL }
  (yyyyyyy) { (image2.tif) 1 0 ICALL }
  ....
ENDCASE
} SETFORM
```



Nota: Utilice siempre el comando **SETMAXFORM** si utiliza varios formularios. Si no utiliza **SETMAXFORM** con varios formularios, se produce un error de comprobación de la serie.



Nota: Para obtener información relacionada con la sintaxis sobre los servicios de descomposición en DocuPrint, NPS y los sistemas del servidor de impresión FreeFlow, consulte *Consejos y sugerencias de los servicios de descomposición* en la *Guía del usuario de FreeFlow VI Compose*.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **SETBFORM**
- **SETCYCLECOPY**
- **SETFPATH**
- **SETMAXFORM**
- **SLIPSHEET**

- CACHE

## SETFPATH

**SETFPATH** define una biblioteca o una lista de bibliotecas de formularios. El valor predeterminado se define en el archivo `/usr/xgf/src/xgfunix.run` o `x:\xgf\src\xgfdos.run`.

Utilice los comandos **SETFPATH** solo en los archivos `xgfunix.run` o `xgfdos.run`. La agregación de **SETFPATH** a un trabajo VIPP® pone en riesgo la portabilidad de este. La agregación de **SETFPATH** a un proyecto VI pone en riesgo la organización y la portabilidad del proyecto.

### Sintaxis

```
(path to form library) SETFPATH
```

```
[ (path to form library 1) (path to form library 2) ... ] SETFPATH
```

Las bibliotecas especificadas con **SETFPATH** son usadas por **SETFORM**, **SETBFORM**, **SCALL**, **FCALL** y **STARTDBM** para encontrar formularios, segmentos y archivos maestros de la base de datos (DBM).

Si se especifica una lista de bibliotecas, se buscan en el orden en el que aparecen en la lista.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **FCALL**
- **SCALL**
- **SETBFORM**
- **SETFORM**
- **STARTDBM**
- **CACHE**

## SETFRAME

**SETFRAME** traza un marco general alrededor de la página actual y de todas las páginas siguientes.

### Sintaxis

```
linewidth offset Colorkey cradius SETFRAME
```

### Donde:

<b>linewidth</b>	es la anchura de la línea en unidades actuales. 0 indica que no hay marco.
<b>desplazamiento</b>	es la distancia en unidades actuales de los márgenes.
<b>Colorkey</b>	es el color.
<b>cradius</b>	es el radio de la esquina del marco en unidades actuales. 0 indica cuadrado.

El valor predeterminado se especifica en el archivo `/usr/xgf/src/xgf.def`.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETMARGIN**, **SETPAT**, **SETTXC**



## SETFTSP

**SETFTSP** selecciona y ajusta a escala una fuente. Además, establece el espaciado de línea como tamaño del punto \* 1,2, que es el valor de espaciado más habitual. Por lo demás, la sintaxis es igual a **SETFONT**.

### Sintaxis

```
/Fontname size SETFTSP
```

```
/Fontname sizeX sizeY SETFTSP
```

Para obtener una descripción de los operandos, consulte **SETFONT**.

### Ejemplo:

```
/NHE 12 SETFTSP
```

Si **POINT SETUNIT**, el ejemplo anterior equivale a:

```
/NHE 12 SETFONT 14.4 SETLSP
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETFONT SETLSP**

## SETFTSW

El comando **SETFTSW** establece el cambio de fuente/color para los comandos **SHMF** o **SHP y SHp** posteriores.

### Sintaxis

```
(ftsw) SETFTSW
```

```
(ftsw) index_length SETFTSW
```

### Donde:

<b>ftsw</b>	es una cadena alfanumérica que desencadenará un cambio de fuente o color. Se recomienda elegir una secuencia de caracteres que no es probable que aparezcan en los datos que se imprimirán.
<b>index_length</b>	indica el número de bytes que siguen a ftsw en los datos que representan el índice de fuente/color. Si no se indica index_length, el operando asume el valor predeterminado 1.

El valor predeterminado (//) se define en el archivo `/usr/xgf/src/xgf.def`.

### Ejemplos

Este ejemplo muestra el uso de un índice de fuentes de tres caracteres con un comando **SHMF**.

```
(//) 3 SETFTSW
/H10 /NHE 10 INDEXFONT
/H12 /NHE 12 INDEXFONT
(//H10 use Helvetica 10 //H12 use Helvetica 12) 0 SHMF
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- INDEXCOLOR
- INDEXFONT
- INDEXSST
- SETTXS
- SHMF, SHMf y SHmf
- INDEXBAT
- SHP y SHp
- INDEXALIGN
- INDEXLSP

## SETGEP

**SETGEP** define una tecla de propiedad de elementos gráficos (**GEPkey**) utilizada en los comandos de dibujo posteriores.

### Sintaxis

```
/GEPkey LineWidth LineColorkey LineDash FillColorkey SETGEP
```

### Donde:

<b>GEPkey</b>	es cualquier cadena alfanumérica que empieza por un carácter alfabético.
<b>LineWidth</b>	define la anchura del borde de contorno. 0 indica que no hay borde.
<b>LineColorkey</b>	define un Colorkey para el borde de contorno.
<b>LineDash</b>	define la trama de guiones para el borde de contorno. 0 indica una línea sólida. Puede ser un número individual que define la anchura igual de las partes rellenas y sin relleno, o una lista que especifique el desplazamiento inicial y una secuencia de anchura [ offset gap1 gap2 ... gapn ].
<b>FillColorkey</b>	define un Colorkey para rellenar el interior.

Las unidades de LineWidth y **LineDash** se definen mediante **SETGUNIT**.

### Ejemplos

Este ejemplo especifica un contorno negro con guiones equidistantes definido como sigue:

- El borde del contorno tiene una anchura de cuatro unidades
- La trama consta de seis unidades negras que se alternan con seis unidades blancas
- Se utiliza un **LMEDIUM** medio en el interior de la trama de relleno

```
/GEPK1 4 BLACK 6 LMEDIUM SETGEP
```

Este ejemplo especifica un contorno negro con guiones aleatorios.

```
/GEPK2 4 BLACK [ 0 10 5 5 5 ] LMEDIUM SETGEP
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **DRAWB** y **DRAWBR**
- **DRAWBM** y **DRAWBRM**
- **DRAWPATH** y **DRAWPATHR**
- **DRAWPOL**
- **SETGUNIT**
- **SETPAT**

- SETTXC
- SHX
- SETCOL
- SETTRAN

## SETGRID

**SETGRID** establece el número de caracteres por línea (cpl) y el número de líneas por página (lpp).

### Sintaxis

```
cpl lpp SETGRID
```

### Donde:

**cpl y lpp** puede ser números o los valores numérico de cadena.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

LSP, SETFONT

## SETGUNIT

**SETGUNIT** establece las unidades de medida de todos los comandos **SETGEP** posteriores.

### Sintaxis

```
unit SETGUNIT
```

### Donde:

**unit**

puede incluir:

**DOT3** (1/300 de pulgada) valor predeterminado

**PELS** (1/240 de pulgada)

**POINT** (1/72 de pulgada)

**CM** (centímetro)

**MM** (milímetro)

**INCH** (pulgada)

**PELS** puede facilitar la conversión de recursos de AFP a recursos VIPP®.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETGEP**, **SETUNIT**

## SETINDENT

**SETINDENT** establece la sangría de comandos **SHP and SHp** posteriores. Después de configurarse, la sangría hace que la primera línea de cada párrafo que se imprime con **SHP** y **SHp** se imprima con desplazamiento horizontal.

### Sintaxis

```
indent SETINDENT
```

### Donde:

**indent** es el valor de sangría en unidades actuales. El valor predeterminado es 0.

Cuando se utiliza SETINDENT en combinación con **SHP y SHp**, la opción +20 se aplica a cada subpárrafo.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SHP y SHp**

## SETIPATH

**SETIPATH** define una biblioteca o una lista de bibliotecas de imágenes. El valor predeterminado se define en los archivos `/usr/xgf/src/xgfunix.run` o `x:\xgf\src\xgfdos.run`. Se utilizan las bibliotecas especificadas por **ICALL** para localizar las imágenes.

Utilice los comandos **SETIPATH** solo en los archivos `xgfunix.run` o `xgfdos.run`. La agregación de **SETIPATH** a un trabajo VIPP® pone en riesgo la portabilidad de este. La agregación de **SETIPATH** a un proyecto VI pone en riesgo la portabilidad y la organización del proyecto.

### Sintaxis

```
(path to image library) SETIPATH
```

```
[ (path to image library 1) (path to image library 2) ... ] SETIPATH
```

Si se especifica una lista de bibliotecas, se buscan en el orden en el que aparecen en la lista.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**ICALL**, **RUNTIF**, **CACHE**



## SETJDT

**SETJDT** activa las opciones en el JDT para las siguientes páginas. Las opciones no especificadas se heredan de las opciones anteriores, no de los valores predefinidos básicos.

JDT debe ser codificado con los comandos de modo nativo VIPP® y almacenado en una de las bibliotecas referenciadas por **SETJPATH**. Se recomienda el uso de la extensión `.jdt`. Consulte Flujos de datos de VIPP® en la *Guía del usuario de FreeFlow VI Compose*, y el comando **STARTLM** para obtener más información sobre el procesamiento de JDT.

### Sintaxis

```
jdt_ref SETJDT
```

```
jdt_ref count SETJDT
```

### Donde:

<b>jdt_ref</b>	puede ser uno de los siguientes:  ( <b>JDTname</b> ) el nombre de un JDT entre paréntesis  [ ( <b>jdtname1</b> ) { <b>JDT contents</b> }... ( <b>jdtnameN</b> ) ] una lista entre corchetes que consiste en jdtnames o código VIPP® (entre llaves) que se usará de forma secuencial y cíclica (ciclo JDT). El código VIPP® entre llaves se puede utilizar para crear un pequeño JDT en línea en lugar de usar la codificación de un archivo externo.
<b>recuento de</b>	indica el número de páginas después del cual se aplica el JDT (es similar al operando de contador de <b>STARTLM</b> ). Este operando retrasa la aplicación de un JDT en páginas siguientes. Consulte el ejemplo <b>GETFIELD</b> para obtener más información.



Nota: En el interior de un JDT, los comandos de marca solo se permiten en un formulario en línea o el procedimiento **ENDPAGE**.

### CONSEJO

Use este comando con un registro (`% %XGF`) de prefijo de modo nativo (**NMP**) para cambiar el JDT página a página. El JDT referenciado por **SETJDT** se conoce como JDT secundario, y contiene solo las opciones que se cambian página a página. El JDT principal referenciado por **STARTLM** al principio del trabajo contiene todas las opciones globales.

El comando **BEGINPAGE** ha sido diseñado para facilitar el uso de JDT principales, secundarios y portadas en la misma aplicación. Consulte **BEGINPAGE** para obtener más información.



Importante: El número máximo de JDT que se pueden incluir en un solo trabajo es 65535. El número máximo de JDT que se pueden incluir en un comando **SETJDT** es 65535. Sin embargo, tantos JDT en un único comando **SETJDT** puede perjudicar al rendimiento del sistema y agotar la memoria virtual. Por lo tanto, Xerox recomienda que se incluyan menos de 1000 JDT en un comando **SETJDT**.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- SETJPATH
- STARTLM
- STARTDBM
- SETPAGEDEF

## SETJPATH

**SETJPATH** define una biblioteca o una lista de bibliotecas de JDT. El valor predeterminado se define en el archivo `/usr/xgf/src/xgfunix.run` o `x:\xgf\src\xgfdos.run`.

Utilice los comandos **SETJPATH** solo en los archivos `xgfunix.run` o `xgfdos.run`. La agregación de **SETJPATH** a un trabajo VIPP® pone en riesgo la portabilidad de este. La agregación de **SETJPATH** a un proyecto VI pone en riesgo la portabilidad y la organización del proyecto.

### Sintaxis

```
(path to JDT library) SETJPATH
```

```
[ (path to JDT library 1) (path to JDT library 2) ... ] SETJPATH
```

Se utilizan las bibliotecas especificadas por **STARTLM**, **SETJDT** y **SETDLFILE** para localizar los JDT.

Si se especifica una lista de bibliotecas, como en la segunda sintaxis más arriba, se realizan búsquedas en las mismas en el orden en el que aparecen en la lista.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETDLFILE**, **SETJDT**, **STARTLM**

## SETKERN

Los comandos **SETKERN** configuran las opciones de espaciado del texto siguiente impreso mediante uno de los comandos **SHx**.

### Sintaxis

```
[ PW_opt TG_opt TK_deg ] SETKERN
```

### Donde:

<b>PW_opt</b>	<p>define la opción de espaciado de pares PW_opt puede tener uno de estos valores:</p> <p><b>0</b> desactiva el espaciado de pares.</p> <p><b>not 0</b> activa espaciado de pares a través de la multiplicación de valores de espaciado proporcionados por el archivo AFM (entradas KP, KPX o KPY). El valor recomendado es 1 y puede ser un número real.</p> <p><b>null</b> mantiene la opción de espaciado de pares actual en vigor.</p>
<b>TG_opt</b>	<p>define la opción de espaciado de seguimiento genérico. TG_opt puede tener uno de estos valores:</p> <p><b>0</b> desactivar espaciado de seguimiento genérico.</p> <p><b>not 0</b> activar espaciado de seguimiento a través de la multiplicación de los valores de espaciado de seguimiento genéricos definidos por VIPP®. El intervalo de valores recomendado es de -3 a + 3 y puede ser un número real.</p> <p><b>null</b> mantiene la opción de espaciado de seguimiento genérico en vigor.</p>
<b>TK_deg</b>	<p>define el grado de espaciado de seguimiento. TK_deg puede tener uno de estos valores:</p> <p><b>0</b> desactiva el grado de espaciado de seguimiento.</p> <p><b>not 0</b> activa el espaciado de seguimiento mediante la selección de valores del grado de espaciado de seguimiento más próximo definido en las entradas TrackKern del archivo AFM. El grado de espaciado de seguimiento va normalmente de -3 a +3, y debe ser un entero.</p> <p><b>null</b> mantiene el grado de espaciado de seguimiento actual.</p> <p>Si TK_deg no es cero y las entradas de TrackKern existen en el archivo AFM, se omite TG_opt.</p> <p>Si TK_deg no es cero y las entradas de TrackKern existen en el archivo AFM, se usa TG_opt.</p> <p>PW_opt y TK_deg se ignoran si no se asocia ningún archivo AFM con la fuente actual o no hay información de espaciado presente en el mismo.</p> <p>TG_opt no requiere información del archivo AFM, por lo que puede usarse incluso si no hay ningún archivo AFM asociado a la fuente actual.</p> <p>Use [0 0 0] SETKERN para desactivar todas las opciones de espaciado (prefijado).</p>

## Ejemplos

Utilice este ejemplo para activar el espaciado entre pares:

```
[ 1 0 0 ] SETKERN
```

Utilice este ejemplo para activar el espaciado de seguimiento medio AFM:

```
[ 0 0 -2 ] SETKERN
```

Utilice este ejemplo para activar el espaciado de seguimiento medio AFM entre pares:

```
[ 1 0 -2 ] SETKERN
```

Utilice este ejemplo para activar el espaciado de seguimiento medio AFM o genérico:

```
[ 0 -2 -2 ] SETKERN
```

Utilice este ejemplo para activar el espaciado de seguimiento genérico ligero y entre pares:

```
[ 1 1 0 ] SETKERN
```

### **Modos**

Este comando es aplicable en todos los modos.

### **Comandos relacionados**

SHx, INDEXKERN

## SETLAYOUT

**SETLAYOUT** es una alternativa a **SETMULTIUP** para establecer un diseño par de varias en 1.

### Sintaxis

```
/layoutname SETLAYOUT
[ /PageWidth      pagewidth
  /PageHeight     pageheight
  /TopBleed       topbleed
  /LeftBleed      leftbleed
  /RightBleed     rightbleed
  /BottomBleed    bottombleed
  /HGutter        hgutter
  /VGutter        vgutter
  /Across         across
  /Down           down
  /Rotate rotate | [ rotate1 rotate2 ... rotateN ]
  /FillOrder      fillorder
  /LayoutMarks    markoption
  /MarkLength     marklength
  /MarkWidth      markwidth
  /MarkOffset     markoffset
] SETLAYOUT
```

### Donde:

Utilice unidades actuales cuando sea necesario.

<b>layoutname</b>	es el nombre de un diseño definido por <b>DEFINELAYOUT</b>
<b>pagewidth</b>	es la anchura de las páginas lógicas
<b>pageheight</b>	es la altura de las páginas lógicas
<b>topbleed</b>	es el sangrado superior (prefijado: 0)
<b>leftbleed</b>	es el sangrado izquierdo (prefijado: 0)
<b>rightbleed</b>	es el sangrado derecho (prefijado: 0)
<b>bottombleed</b>	es el sangrado inferior (prefijado: 0)
<b>hgutter</b>	es el margen interno horizontal (prefijado: 0)
<b>vgutter</b>	es el margen interno vertical (prefijado: 0)
<b>a través</b>	es el número de páginas lógicas en la hoja (prefijado: 1)
<b>abajo</b>	es el número de páginas lógicas en la parte inferior de la hoja (prefijado: 1)
<b>girar</b>	es el giro de las páginas lógicas; valores posibles: 0 (predefinido), 90, 180, 270 (o -90). Si se utiliza una matriz se aplican los valores de giro en secuencia cíclica en el orden de las páginas.
<b>fillorder</b>	es el orden de llenado de las páginas lógicas (prefijado: /RD)

	<b>/RD</b> (derecha-abajo) de izquierda a derecha, a continuación, de arriba a abajo
	<b>/LD</b> (izquierda-abajo) de derecha a izquierda y, a continuación, de arriba a abajo
	<b>/RU</b> (derecha-arriba) de izquierda a derecha y, a continuación, de abajo a arriba
	<b>/LU</b> (izquierda-arriba) de derecha a izquierda y, a continuación, de abajo a arriba
	<b>/DR</b> (abajo-derecha) de arriba a abajo y, a continuación, de izquierda a derecha
	<b>/UR</b> (arriba-derecha) de abajo a arriba y, a continuación, de izquierda a derecha
	<b>/DL</b> (abajo-izquierda) de arriba a abajo, a continuación, de derecha a izquierda
	<b>/UL</b> (arriba-izquierda) de abajo a arriba y, a continuación, de derecha a izquierda
<b>markoption</b>	es la opción de las marcas de diseño en las páginas lógicas:
	<b>0</b> ninguna marca (predeterminado)
	<b>1</b> marcas de recorte
	<b>2</b> marcas de sangrado
	<b>3</b> marcas de sangrado y recorte
	<b>+</b> opción para imprimir las marcas en la portada, la contraportada o en ambas
	<b>+0</b> marcas de impresión solo en la portada
	<b>+10</b> marcas de impresión solo en la contraportada
	<b>+20</b> marcas de impresión en la portada y la contraportada
	<b>+100</b> desactivar marcas de recorte internas
	<b>+200</b> desactivar marcas de sangrado internas
	<b>+300</b> desactivar las marcas de sangrado y recorte internas
<b>marklength</b>	es la longitud de las marcas en unidades actuales (0 prefijado)
<b>markwidth</b>	es la anchura de las marcas en unidades actuales (0 prefijado)
<b>markoffset</b>	es la marca de desplazamiento desde la esquina en unidades actuales (0 prefijado)

Se puede omitir cualquier parámetro con un prefijado. **SETPARAMS** se utiliza para cambiar los valores prefijados o asignar valores prefijados a los parámetros que no tienen un valor.

Un simulador de diseño `/xgf/demo/layoutsimulator.nm` está disponible para ayudar a implementar el comando **SETLAYOUT**.

## Ejemplos



```
POINT SETUNIT
612 792 SETPAGESIZE
[ /PageWidth 216
  /PageHeight 270
  /HGutter 18
  /VGutter 18
  /Across 2
  /Down 3
  /Rotate 90
] SETLAYOUT
```

```
[...
  /Rotate [ 0 180 0 180 ]
] SETLAYOUT
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

DEFINELAYOUT, SETMULTIUP

## SETLFI

El comando **SETLFI** establece el incremento de salto de línea.

### Sintaxis

```
lfi SETLFI
```

### Donde:

**lfi** es el número de líneas que se saltarán después de cada registro cuando sean positivos o antes de cada registro si son negativos. El valor debe ser un número entero. El valor pre-determinado es 1.

El valor de espaciado de línea se define mediante **SETGRID** o **SETLSP**.

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

**SETGRID, SETLSP**

## SETLKF

El comando **SETLKF** activa el modo de tramas vinculado (volver a flujo de texto). El modo permite definir una recopilación de tramas rectangulares en la página en la que el texto o los elementos gráficos se colocarán mediante comandos VIPP® como **SHx**, **ICALL** o **SCALL**. Las tramas se rellenarán en el orden definido por **SETLKF**. Al llenar la última trama, se produce un **PAGEBRK** implícito y el elemento siguiente (o la parte restante del elemento de texto en caso de **SHP** y **SHp**) se colocará en la primera trama de una nueva página.

No utilice el comando **MOVETO** si el modo **LKF** está activado. Solo pueden programarse ubicaciones relativas mediante **MOVEH**, **MOVEHR**, **NL** y **SETLSP**. Los elementos que deben situarse en ubicaciones fijas deben colocarse mediante una definición de formulario (**SETFORM**) en línea o externa.

Todas las alineaciones se calcularán según la trama actual (el centro significa el centro de la trama, la derecha significa el lado derecho de la trama, la anchura de la columna se ajusta a la anchura de la trama, etc.). La posición de impresión secundaria (como se define mediante **MOVEH** o **MOVEHR**) se aplicará en la trama si difiere de la posición de impresión principal.

### SETLKF, FOREACH y SHROW

**SETLKF**, **FOREACH** y **SHROW** pueden interactuar entre sí se realiza una llamada a **SHROW** en el bucle **FOREACH**. Este comportamiento se activa a través de la sintaxis **FOREACH** siguiente:

```
{ sequence of VIPP commands } variable_array /MF FOREACH
```

Cuando se utiliza la opción /MF, se evalúa el tamaño total de todos los **SHROW** invocados en el procedimiento **FOREACH**. Si no encaja en el marco, las filas no se imprimen, se realiza una llamada a **NEWFRAME** y el procedimiento se ejecuta por segunda vez con los mismos valores de entrada de la tabla. Esta opción pretende solucionar problemas de varios marcos con distintos formatos, para que en la segunda ejecución se pueda realizar una llamada diferente a **SHROW**.

### Sintaxis

```
[ [ Hor1 Ver1 Anch1 Alt1 Girar1] [ Hor2 Ver2 Anch2 Alt2 girar2 ] .... [ [ Hor1 Ver1 Anch1 Alt1 Girar1] [ Hor2 Ver2 Anch2 Alt2 girar2 ] .... ]
```

Esta sintaxis se utiliza para desactivar el modo de trama vinculada:

```
[ ] SETLKF
```

Donde:

#### HorX y VerX

define las coordenadas vertical y horizontal (en unidades actuales) de la esquina superior izquierda de la trama relacionada al origen actual (TL o BL) de la página lógica.

#### WidthX y HeightX

defina el tamaño de la trama (en unidades actuales).

#### rotateX

define los valores de giro. El valor admitido es 0, VI Compose ignora cualquier otro valor.



Nota: La colocación vertical (VPOS) en una trama inicial en la parte superior de la trama. En un bloque de texto en una trama, VPOS no es la línea de base de la fuente, pero es la parte superior de la celda del carácter (suponiendo un tamaño de fuente similar y LSP). La línea de base de la fuente está colocada hacia abajo un 70 % de LSP. Esto permite a los trazos ascendentes de texto (y a los descendentes en la parte inferior de la trama) ajustarse a la trama. Por ejemplo, el carácter T en la primera línea no se truncará en el parte superior de la trama y el carácter “p” no se truncará en la parte inferior de la trama.

La transición de tramas se genera cuando el espacio restante es inferior al espaciado de la línea. LSP debe representar el espacio mínimo suficiente para incluir un carácter (70 % para trazos ascendentes y 30 % para trazos descendentes).

Esta regla no se aplica a gráficos, imágenes ni segmentos. El origen debe figurar en función de la anchura, la altura y la alineación del elemento, posiblemente mediante los comandos **NL** y **MOVEHR**, si el VPOS actual no coincide.



Consejo: Seleccione un LSP próximo al tamaño de la fuente y, a continuación, puede realizar ajustes verticales de menor magnitud mediante NL. Para realizar el ajuste según el LSP, puede utilizar una expresión aritmética.

### Ejemplo:

```
LSP'*.2 NL % move down 20% of the current LSP
```

### Ejemplos

```
LSP'*.2 NL % move down 20% of the current LSP
```

[ [ 200 200 1025 2900 0 ]	% Columna izquierda [ 1325 200 1025 2900 0 ]
---------------------------	--

% column



Nota: Para permitir el flujo automático a través de los límites de página, escriba **SETLKF** antes de cualquier comando de marca en la página. Si no es el caso (por ejemplo, si **SETLKF** se codifica en la definición del formulario) no se producirá ninguna transición de página. En su lugar, la última trama se desbordará más allá de su borde inferior y, posiblemente, más allá del borde de la página. Es posible definir distintos conjuntos de tramas y formularios de páginas consecutivas mediante el comando **SETPAGEDEF**.

### Modos

Este comando es aplicable en los modos nativo y de base de datos.

### Comandos relacionados

- **FRCOUNT**
- **FRLEFT**
- **GOTOFRAME**
- **NEWFRAME**
- **SETGRID**
- **SETLSP**
- **SETMARGIN**

## SETLMFILE

**SETLMFILE** hace que los comandos siguientes **STARTLM** o **STARTDBM** lean datos de un determinado archivo, que debe estar ubicado en una de las bibliotecas referenciadas por **SEMPATH**.

### Sintaxis

```
(file name) SETLMFILE
```

```
(file name) skiplines SETLMFILE
```

La segunda sintaxis hace que las primeras skiplines en el archivo se ignoren.

### Ejemplos

El envío de la sintaxis en este ejemplo hace que el archivo de datos `invoice.dat` se imprima mediante el diseño descrito en `invoice.jdt`.

```
%!
```

```
(invoice.dat) SETLMFILE
```

```
(invoice.jdt) STARTLM
```

### Modos

○Este comando es aplicable en modo de línea y de base de datos.

### Comandos relacionados

**SEMPATH**, **STARTDBM**, **STARTLM**

## SETLSP

El comando **SETLSP** configura el espaciado de línea. El espaciado de línea se calcula desde los márgenes y las especificaciones de cuadrícula prefijadas.

### Sintaxis

LSPval SETLSP

### Donde:

**LSPval** es el valor del espaciado de línea.



Nota: La especificación de un valor de espaciado de línea negativo permitirá realizar un desplazamiento hacia arriba de la posición de impresión en la página. A menudo se usa para imprimir un bloque de direcciones donde el código postal debe ubicarse en una posición X e Y, y el resto de las líneas de la dirección se espaciarán desde esa posición.

Es importante restablecer el espaciado de línea con un valor positivo para el resto de la aplicación.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- SETGRID
- SETMARGIN
- SHC y SHc
- SHJ y SHj
- SHL y SH
- SHR y SHr
- SHX
- SHMF, SHMf y SHmf
- SHP y SHp
- INDEXALIGN
- INDEXLSP

## SETMARGIN

**SETMARGIN** configura los márgenes superior, inferior, izquierdo y derecho mediante las unidades actuales. El valor predeterminado se especifica en el archivo `/usr/xgf/src/xgf.def`.

### Sintaxis

```
top bottom left right SETMARGIN
```

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

**SETFONT**, **SETGRID**

## SETMAXBFORM

**SETMAXBFORM** establece el número máximo de planos de formularios posteriores en una sola página. Los intervalos de números de plano desde 0 a `maxplanenumber-1` y los formularios posteriores asociados se imprimen en este orden. Los elementos de PostScript son opacos.

**SETMAXBFORM** se debe utilizar si la aplicación requiere más de un formulario de contraportada.

### Sintaxis

```
maxplanenumber SETMAXBFORM
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETBFORM**



## SETMAXCOPY

**SETMAXCOPY** establece el número máximo de copias de un trabajo en el valor especificado. Utilice este comando si **SETCYCLECOPY** se configura dinámicamente mediante **PROCESSDJDE** o **BEGINPAGE**. **SETMAXCOPY** no configura el número de copias hasta que se ejecuta un comando **SETCYCLECOPY**. El valor de **SETCYCLECOPY** no puede ser mayor que el valor especificado para **SETMAXCOPY**.

### Sintaxis

number SETMAXCOPY

### Donde:

número es un valor definido por el usuario.

### Ejemplos

```
10 SETMAXCOPY
```

```
{CASE DJDECMD {} (COPIES) {DJDEPAR SETCYCLECOPY} ENDCASE } 0 ($DJDE) 3
```

```
PROCESSDJDE
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETCYCLECOPY**, **PROCESSDJDE**

## SETMAXFORM

**SETMAXFORM** establece el número máximo de planos de formularios en una sola página. Los intervalos de números de plano desde 0 a `maxplanenumber-1` y los formularios asociados se imprimen en este orden. Los elementos de PostScript son opacos. El valor predeterminado se especifica en el archivo `/usr/xgf/src/xgf.def`.

### Sintaxis

```
maxplanenumber SETMAXFORM
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETFORM**

## SETMEDIA

El comando **SETMEDIA** establece los requisitos de materiales de impresión.

En los ejemplos de sintaxis más abajo, **SETMEDIA MediaType, MediaColor y MediaWeight** como los requisitos de tipo de materiales de impresión para las páginas siguientes.

### Sintaxis

```
(MediaType:MediaColor:MediaWeight) SETMEDIA
```

```
(MediaType:MediaColor:MediaWeight) [c1 c2... cn] SETMEDIA
```

```
[ (Med req1) (Med req2) ... (Med reqN) ] SETMEDIA
```

Este requisito de materiales de impresión se compara a los valores relacionados configurados por el comando **SetTray** en la interfaz de usuario DocuPrint NPS o por la utilidad trayload de VI Compose. Si una o diversas bandejas satisfacen este requisito, las hojas para la página actual y las páginas siguientes, hasta el siguiente comando **SETMEDIA**, se alimentan desde estas bandejas.

La segunda sintaxis establece el requisito de los materiales de impresión de la selección especificada de copias [c1, c2... cn].

La tercera sintaxis define una lista de requisitos de materiales de impresión que se usarán de forma secuencial cíclicamente (ciclo de materiales de impresión).

Tenga en cuenta esta circunstancia al usar **SETMEDIA**:

- Si alguno de los atributos del papel se especifican como nulo, se omiten estos atributos en las siguientes selecciones de materiales de impresión. Este ejemplo omite MediaColor.

```
(Drilled:null:100) SETMEDIA
```

- Si alguno de los atributos de los materiales de impresión, como el tipo, color o peso se omiten, la última especificación o el valor predefinido del atributo seguirá vigente.
- Los ":" finales se pueden omitir como se indica en este ejemplo.

```
(Plain::) SETMEDIA
```

```
(Plain:) SETMEDIA
```

```
(Plain) SETMEDIA
```

Este comando no configura el atributo de los materiales de impresión de **PageSize**, sino que los define el comando **SETPAGESIZE**.



Nota: Para obtener información adicional relacionada con los dispositivos que no admiten la selección de materiales de impresión PostScript estándar, consulte [Soporte de materiales de impresión](#).



Consejo: Es necesaria la codificación de un registro %%DocumentMedia al principio del archivo de datos en DocuPrint NPS para que las bandejas relacionadas estén disponibles para el trabajo. Consulte [%%](#) para obtener más información.

Consulte también Consideraciones de VIC para iGen en la *Guía del usuario de FreeFlow VI Compose* para obtener más información.

### Solo para el servidor de impresión FreeFlow

La sintaxis de **SETMEDIA** se ha ampliado para admitir la selección de estucado para la cara anterior y la posterior.

Puede usar la opción de materiales de impresión **postnoimage** para extraer papel de una bandeja, referenciada a veces como bandeja especial o de imposición, que omite la ruta de impresión. Se utiliza para extraer formularios preimpresos que omiten la ruta de impresión. Este ejemplo inserta una única inserción:

```
(postnoimage:pink) SETMEDIA
```

```
0 NL PAGEBRK
```

Este ejemplo inserta cinco inserciones ordenadas:

```
(postnoimage/5:white) SETMEDIA
```

```
{ 0 NL PAGEBRK } 5 REPEAT
```

### Sintaxis

```
(MediaType:MediaColor:MediaWeight:MediaFrontCoating:MediaBackCoating)  
SETMEDIA
```

### Donde:

#### MediaFrontCoating y MediaBackCoating

hace referencia a los atributos de estucado de la cara anterior y la posterior de un tipo de papel determinado, y puede tener los siguientes valores:

**Ninguno (sin estucar)**

**Glossy (Brillante)**

**HighGloss (Muy brillante)**

**SemiGloss (Semibrillante)**

**Satin (Satinado)**

**Matte (Mate)**

Estos nuevos atributos siguen las mismas normas de los atributos de los soportes de impresión descritos en el comando **SETMEDIA**, y se pueden aplicar a todos los formularios de la sintaxis **SETMEDIA**.

### Ejemplos

```
(Cover1:::Glossy:None) SETMEDIA
```

```
(Cover2:::120:Matte:Matte) SETMEDIA
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **%%**
- **SETCYCLECOPY**

- **SETPAGESIZE**
- **SETMEDIAT**
- **SLIPSHEET**

## SETMEDIAT

**SETMEDIAT** establece temporalmente los requisitos de materiales de impresión de la página actual. La página siguiente restaura el material de impresión seleccionado anteriormente, o el papel necesario para satisfacer los materiales de impresión del ciclo.

### Sintaxis

```
(MediaType:mediaColor:MediaWeight) SETMEDIAT
```

Igual que con el comando **SETMEDIA**, los valores de los materiales de impresión pueden mantenerse inalterados si se dejan vacíos los campos correspondientes.

### Ejemplos

Este ejemplo selecciona un material con *Tipo = Perforado*, *Peso = 120* y color igual a la configuración de la página actual. Sin embargo, las opciones anteriores se restauran en la página posterior.

```
(drilled::120) SETMEDIAT
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**% %**, **SETMEDIA**, **SETPAGESIZE**

## SEMPATH

**SEMPATH** define una biblioteca o una lista de bibliotecas de varios archivos. El valor predeterminado se define en el archivo `/usr/xgf/src/xgfunix.run` o `x:\xgf\src\xgfdos.run`.

Utilice los comandos **SEMPATH** solo en los archivos `xgfunix.run` o `xgfdos.run`. La agregación de **SEMPATH** a un trabajo VIPP® pone en riesgo la portabilidad de este. La agregación de **SEMPATH** a un proyecto VI pone en riesgo la portabilidad y la organización del proyecto.

### Sintaxis

```
(path to misc. library) SEMPATH
```

```
[ (path to misc. library 1) (path to misc. library 2) ... ] SEMPATH
```

Las bibliotecas especificadas son usadas por **SETLMFILE**, **SETDLFILE**, **CACHE** y **RUN** para encontrar los archivos referenciados por estos comandos.

Si se especifica una lista de bibliotecas, como en la segunda sintaxis, se realizan búsquedas en las mismas en el orden en el que aparecen en la lista.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **RUN**
- **SETDLFILE**
- **SETLMFILE**
- **STOREVAR**
- **CACHE**

## SETMULTIUP

**SETMULTIUP** permite la impresión de varias en 1. La impresión de varias en 1 es la impresión de varias páginas lógicas en una página física.

### Sintaxis

En esta sintaxis cada página lógica se define por una entrada.

```
[ Hor1 Ver1 girar1 EscalaH1 EscalaV1 % configuración de página lógica 1 Hor2 Ver2 girar2 EscalaH2 EscalaV2 ]
[ [ Hor1 Anch1 ] [ Ver1 Altat1 ] girar1 EscalaH1 EscalaV1 [ Hor2 Anch2 ] [ Ver2 Alt2 ]
```

Donde:

<b>Horizontal y vertical</b>	define los orígenes horizontales y verticales de las páginas lógicas. Estos se miden siempre desde la esquina inferior izquierda de la página física.
<b>Anchura y altura</b>	define el tamaño de la página original. El tamaño de página original es el tamaño para el que el trabajo se ha diseñado originalmente. Si no se especifica, el tamaño de la página original es igual al tamaño de la página física actual (según lo predefinido por la impresora o la configuración de <b>SETPAGESIZE</b> ).  El tamaño de la página lógica se calculará mediante la aplicación de los factores de escala (Hscale y Vscale) al tamaño de página original.  La especificación del tamaño de la página original es útil si el tamaño de las páginas originales difiere de la página física (por ejemplo, si se imprimen 2 páginas A4 en papel A3).
<b>girar</b>	define los ángulos de giro.
<b>Hscale y Vscale</b>	define el ajuste a escala horizontal y vertical.

Si se activa la opción varios en 1, cada salto de página que ocurre después de **PAGEBRK**, la alimentación de formularios, el paso al canal uno o cuando se llega a la última línea provoca un salto a la nueva página lógica. La página física se imprime solo si la última página lógica se completa o si se llega al final del trabajo.

En el modo Varias en 1, tenga en cuenta que si no se aplica ninguna reducción correcta (reducción = escalado < 1), solo la parte de la página lógica que se superpone a la página física se imprime, y se requieren ajustes de los valores de colocación.

Este comando debe usarse como el primer comando después de la línea **%!** en el archivo de impresión o como el primer comando de un JDT.

Utilice **NEWFRONT**, **NEWBACK** y **NEWSIDE** para generar una nueva página física.

### Ejemplos

Este ejemplo define la impresión 2 en 1 en papel A4 con unidades DOT3.

```
[ 0 3500 -90 .7 .7
```



```
0 1750 -90 .7 .7 ] SETMULTIUP
```

Este ejemplo define la impresión a 2 caras A4 en una hoja A3 de un documento que ha sido sometido a un proceso RIP por el servicio de descomposición.

```
4960 3500 SETPAGESIZE           % size of the sheet
[ [ 0 2480][0 3500] 0 1 1       % original A4 page
  [2480 2480][0 3500] 0 1 1     % original A4 page
] SETMULTIUP
(doc1.ps) RUNDD
```



Nota: El uso de la reducción y del modo de varias en 1 puede afectar al rendimiento de la impresora, especialmente si se realiza la llamada a las imágenes.

La colocación del comando **SETMULTIUP** determina el resultado. Por ejemplo, si una instrucción 4 en 1 se codifica en el archivo DBM, se imprimirán 4 copias del registro que se está procesando actualmente. Cuatro copias de la misma página física. Si se codifica la opción varias en 1 en el archivo de datos, o para crear y realizar una llamada a un JDT con una instrucción **SETMULTIUP** 4 en 1, la página resultante contiene los registros 1, 2, 3 y 4.



Consejo: Use este comando con **SETCYCLECOPY** y **COLLATE\_off** para producir varias copias de documentos más pequeños en una hoja de papel. Las coordenadas de cada página lógica están relacionadas con la esquina inferior izquierda de la página física, y definen la esquina inferior izquierda de la página lógica.

Las opciones de Varias en 1 predefinidas globalmente en un determinado sitio deben colocarse en el archivo `/usr/xgf/src/xgf.mup`. Consulte Listas estándar, tablas, claves y atributos en la *Guía del usuario de Free-Flow VI Compose* para obtener más información. Para invocarlos, use esta sintaxis:

```
VAR.4UP SETMULTIUP
```

Al crear un trabajo que puede no usar la totalidad de la página, para usar **SETMULTIUP**, por ejemplo, una aplicación de comprobación que se imprimirá en formato 4 en 1 en una página, cree una página física maestra con la imagen comprobada y los datos de variables en la parte inferior de la página, y no en la superior. Esto facilita la configuración de las posiciones horizontal y vertical usadas en el comando **SETMULTIUP**, dado que la esquina inferior izquierda de la página máster es la coordenada 0,0 que se usa para colocar la segunda, tercera y cuarta comprobación en la definición de varias en 1.

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

- **NEWBACK**
- **NEWFRONT**
- **NEWSIDE**
- **ONEUP**
- **TWOUP**
- **SETLAYOUT**

## SETNMP

**SETNMP** define la cadena de prefijo de modo nativo (NMP). Para obtener más información acerca de NMP, consulte Flujos de datos VIPP® en la *Guía del usuario de FreeFlow VI Compose* y %%XGF.

### Sintaxis

```
(NMP string) SETNMP
```

%%XGF es el valor predeterminado de NMP definido en el archivo `/usr/xgf/src/xgf.def`.

### Modos

Este comando es aplicable en el modo de línea

### Comandos relacionados

%%XGF, NMP\_off, STARTLM

## SETOBIN

**SETOBIN** establece el destino de la bandeja de salida.

### Sintaxis

```
(OutputType) SETOBIN
```

```
(OutputType) [ c1 c2 ... cn ] SETOBIN
```

```
[ (OType1) (OType2) ... (OTypeN) ] SETOBIN
```

La primera sintaxis permite seleccionar una determinada bandeja de salida. La segunda y tercera sintaxis permiten seleccionar otra bandeja de salida para las copias de un determinado documento, o definir un ciclo de salida de cada página en una bandeja distinta, igual que la sintaxis del ciclo **SETMEDIA**.

OutputType depende del dispositivo y se puede asignar a las claves estándares con **SETVAR**.

### Ejemplos

```
/VARTT (TopTray) SETVAR
```

```
VARTT SETOBIN
```

Cada impresora varía la descripción de las bandejas de salida. Compruebe los valores de OutputType en la documentación de la impresora.

SETOBIN depende del diccionario OutputAttributes, que es una función de nivel 2 de PostScript no implementada en la totalidad de los dispositivos de nivel 2.



**Nota:** En el servidor de impresión FreeFlow, puede usar este comando para configurar una acabadora externa como un destino de acabado de subconjuntos. Por ejemplo: (SBM) SETOBIN. En esta situación, no use **/ON SETFINISHING**.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETOBINT**

## SETOBINT

**SETOBINT** configura temporalmente el destino de la bandeja de salida de la página actual. La página siguiente vuelve a asumir el destino de bandeja de salida anterior o la siguiente bandeja de salida en el ciclo.

### Sintaxis

(OutputType) SETOBINT

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

SETOBIN

## SETOTL

El comando **SETOTL** establece el contorno del texto para todo el texto impreso posterior. De manera prefijada, el contorno del texto se desactiva.

### Sintaxis

```
[ LineWidth Colorkey ] SETOTL
```

### Donde:

**LineWidth** es la anchura de la línea en unidades actuales.

**Colorkey** define el color de la línea.

### Ejemplos

```
[ 2 RED ] SETOTL
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**INDEXOTL**

## SETPAGEDEF

Utilice el comando **SETPAGEDEF** para definir distintos conjuntos de tramas, formularios, materiales de impresión y otras opciones de diseño de páginas consecutivas. Los diseños se aplicarán en el orden en el que se definen de manera cíclica a menos que el diseño sea seguido por la clave /R. Cuando se utiliza /R, se aplicará el último diseño indefinidamente.



Nota: Si usa el comando **SETPAGEDEF**, no puede usar la sintaxis del ciclo de comandos como **SETFORM** o **SETJDT**, porque usan el mismo índice. Para recorrer los formularios, utilice una rutina **BEGINPAGE** e instrucciones **CASE**. La primera definición de diseño se activa durante la ejecución de **SETPAGEDEF**. Las definiciones de diseño posteriores se activan al final del **PAGEBRK** para la página siguiente. Para cancelar **SETPAGEDEF**, utilice la instrucción siguiente: [ { } ] **SETPAGEDEF**

### Sintaxis

```
[ { layout_definition1 } { layout_definition1 } ... ] SETPAGEDEF
[ { layout_definition1 } { layout_definition1 } ... /R ] SETPAGEDEF
```

### Donde:

**layout\_definitionX**

define el diseño de una página en particular. Puede contener cualquier comando VIPP® relacionado con la configuración de diseño, por ejemplo, **SETLKF**, **SETFORM**, **SETMEDIA** y **SETJDT**.

### Ejemplos

```
[ { [ [ 200 500 990 1330 0 ]      % layout for page 1
      [ 200 2400 990 900 0 ]
      [ 1290 500 990 430 0 ]
      [ 1290 2300 990 1000 0 ]
    ] SETLKF
    (form1.frm) SETFORM
  }
  { [ [ 200 200 990 3100 0 ]      % layout for page 2
      [ 1290 200 990 3100 0 ]
    ] SETLKF
    (form2.frm) SETFORM
  }
  /R                                     % repeat layout 2
] SETPAGEDEF
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETLKF**, **SETJDT**

## SETPAGENUMBER

El comando **SETPAGENUMBER** establece la numeración de páginas de todas las páginas siguientes.

### Sintaxis

```
(format) start pos SETPAGENUMBER
```

```
(format) start rotate pos SETPAGENUMBER
```

```
(format) start hpos vpos align SETPAGENUMBER
```

```
(format) start hpos vpos rotate align SETPAGENUMBER
```

### Donde:

<b>formato</b>	es cualquier cadena donde # representa el marcador de posición de los números de página. Utilice varios caracteres # para imprimir los ceros iniciales.
<b>inicio</b>	especifica el número de la primera página. Este valor puede ser cero o negativo. Cuando se utiliza un valor negativo o cero, los números de página inferiores a uno no se imprimen. El valor máximo es 999999.
<b>girar</b>	especifica el ángulo de giro. Los valores positivos giran en sentido antihorario.
<b>pos</b>	proporciona la posición como se indica a continuación:  <b>0</b> no se imprimirá el número de página (desactivación temporal) <b>1</b> centro inferior <b>2</b> inferior derecha <b>3</b> centro superior <b>4</b> superior derecha
<b>hpos y vpos</b>	ofrecen las coordenadas horizontal y vertical absolutas.
<b>alinear</b>	proporciona la alineación como se indica a continuación:  <b>5</b> izquierda <b>6</b> derecha <b>7</b> centro

La numeración predeterminada de la página, la fuente (índice reservado de la fuente /PNFT) y el color (índice reservado del color /PNCL) usados para imprimir los números de página se especifican en el archivo `/usr/xgf/src/xgf.def`.

Para retener los valores actuales durante el cambio de otras opciones de numeración de páginas de las páginas siguientes y actuales, use un valor nulo en los operandos de inicio y formato. El operando rotate (girar) es, en este caso, obligatorio. Los comandos **SETPAGENUMBER** deben preceder a cualquier comando de marca de página.

## Ejemplos

Este ejemplo imprime Page 001 en la esquina inferior derecha de la primera página.

```
(Page ###) 1 2 SETPAGENUMBER
```

Este ejemplo se imprime en la parte superior central con el mismo formato y la misma continuación de página.

```
null null 0 3 SETPAGENUMBER
```

Este ejemplo se imprime en la parte inferior central con el mismo formato, y reinicia la numeración de páginas en 1.

```
null 1 0 1 SETPAGENUMBER
```

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

**SETMARGIN**



## SETPAGESIZE

El comando **SETPAGESIZE** define los requisitos de tamaño de página para todas las páginas siguientes.

### Sintaxis

```
pagewidth pageheight SETPAGESIZE
```

### Donde:

<b>pagewidth</b>	es la anchura de la página física en unidades actuales. Se presupone que la anchura de la página se refiere a la dimensión corta de la hoja, el eje horizontal en la orientación vertical.
<b>pageheight</b>	es la altura de la página física en unidades actuales. Se presupone que la altura de la página se refiere a la dimensión larga de la hoja, el eje vertical en la orientación vertical.

Los comandos de VIPP® **PORT**, **LAND**, **IPOINT** e **ILAND** dependen de esta convención para colocar elementos en la página. Si no se sigue, se puede generar un error de coincidencia de orientación.

Todas las opciones del diseño de página, como los márgenes, la cuadrícula y la orientación, se basan en estos valores. Si no se especifica, el valor prefijado es el tamaño de página prefijado del dispositivo de impresión. Se produce un error cuando este comando se coloca incorrectamente, por ejemplo, en el medio de una página.

En las impresoras de nivel 2, este comando hace que las páginas siguientes impriman mediante el material de impresión adecuado.

### Ejemplos

Este ejemplo establece el tamaño de la página para imprimir material de impresión de tamaño Carta EE. UU. (8,5 x 11 pulg.) con unidades actuales de DOT3.

```
DOT3 SETUNIT
```

```
2550 3300 SETPAGESIZE
```

Este ejemplo establece el tamaño de la página como vertical A3, mediante unidades actuales de MM.

```
MM SETUNIT
```

```
297 420 SETPAGESIZE
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**% %XGF, NMP\_off**

## SETPARAMS

**SETPARAMS** establece parámetros constantes en todo el trabajo. Después de configurar un parámetro con **SETPARAMS**, el parámetro se aplica a todos los comandos siguientes a los que está relacionado. Para obtener más información, consulte [Descripción del parámetro](#).

### Sintaxis

```
[parameters] SETPARAMS
[ /Name1 val1 /Name2 val2 ... /NameN valN ] SETPARAMS
[ /code /Name1 val1 /Name2 val2 ... /NameN valN ] SETPARAMS
```

### Donde:

<b>NameX</b>	es un nombre de parámetro tomado de la tabla en <a href="#">Descripción del parámetro</a> .
<b>valX</b>	es el valor que se asignará a este parámetro de acuerdo con la información proporcionada en la tabla.
<b>code</b>	es un número entero que permite configurar varios parámetros a la vez. Se calcula agregando todos los códigos relacionados con los parámetros necesarios. Consulte <a href="#">Descripción del parámetro</a> . No todos los parámetros tienen un código de asociado.

### Ejemplos

```
[/3D true /SliceBurst .1] SETPARAMS
```

Este ejemplo muestra el uso de `/Pagerange` al principio de un archivo de datos:

```
%!
XGF
[ /PageRange 1 ] SETPARAMS
1 10 PAGERANGE
...
```



Nota: **SETPARAMS** reemplaza al comando **SETDDGPARAMS**, pero **SETDDGPARAMS** es compatible con versiones anteriores.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- [DRAWPIE](#)
- [DRAWBAR](#)
- [DRAWCRV](#)
- [FORMAT](#)
- [Parámetros](#)

## SETPAT

Utilice **SETPAT** para crear una instancia de una trama de un prototipo definido mediante **SETPPAT**. Una trama es un dibujo pequeño que se puede utilizar para rellenar un área de forma repetida. Una vez definido, puede utilizarse en lugar de un color, o en combinación con el mismo.

### Sintaxis

```
/PATkey /PPATkey Htrans Vtrans rotate Hscale Yscale SETPAT
/PATkey (imagenname) Htrans Vtrans rotate Hscale Yscale SETPAT
/PATkey [ /VPGL lwidth color1 color2 ] Htrans Vtrans rotate Hscale Yscale SETPAT
/PATkey [ /VPCR lwidth color1 color2 ] Htrans Vtrans rotate Hscale Yscale SETPAT
/PATkey [ /VPPG1 textPixelFrequency backgroundPixelFrequency color1 color2 ]
Htrans Vtrans rotate Hscale Yscale SETPAT
```

### Donde:

<b>PATkey</b>	es el nombre de la trama (clave de la trama) para definir cualquier cadena alfanumérica que empieza por un carácter alfabético.
<b>PPATkey</b>	es el nombre de la trama de prototipo en la que se va a crear la trama.
<b>/VPGL</b>	es una <b>PPATkey</b> integrada que se utiliza para definir un efecto de brillo de tramas de vectores.
<b>/VPCR</b>	es una <b>PPATkey</b> integrada que se utiliza para definir efectos de correlación de tramas de vectores (1 o 2 capas).
<b>/VPPG1</b>	es una <b>PPATkey</b> integrada que se utiliza para definir un efecto de pantógrafo vacío.
<b>textPixelFrequency</b>	define el tamaño (frecuencia) del píxel de texto que se debe ocultar.
<b>backgroundPixelFrequency</b>	define el tamaño (frecuencia) del píxel del fondo que oculta el texto. Se supone que es mayor que el doble del valor de textPixelFrequency.
<b>lwidth</b>	es la anchura de la línea de trama en píxeles de un efecto de brillo o correlación. Los valores más apropiados comprendidos entre 1 y 3 (real). 1 es la opción más común.
<b>color1</b>	es el color de fondo de un efecto de brillo o correlación. Este es el color de fondo de los píxeles de fondo para el efecto de pantógrafo vacío
<b>color2</b>	es el color de trama de un efecto de correlación de 1 capa o brillo. Este es el color de fondo de los píxeles de texto para el efecto de pantógrafo vacío
<b>color3</b>	es el color de trama de la segunda capa de un efecto de correlación de 2

	capas.
<b>Htrans</b>	es la traducción horizontal inicial de la primera celda en puntos.
<b>Vtrans</b>	es la traducción vertical inicial de la primera celda en puntos.
<b>girar</b>	es el giro de la celda en grados.
<b>Hscale</b>	es el factor de escala horizontal de la celda.
<b>Vscale</b>	es el factor de escala vertical de la celda.
<b>ImageName</b>	puede ser un archivo JPEG, TIF o EPS.

Después de ser definida por **SETPAT**, es posible usar una clave de trama en lugar de **Colorkey** con cualquier comando VIPP® que lo requiera. Es decir:

- **SETTXXC**
- **INDEXCOLOR**
- **FROMLINE/RPEKEY**
- **SETPARAMS**: ColorTable, BGColor
- **SETGEP**: FillColorkey

Si el tipo de patrón es sin color (PaintType=2), se pintará con el último color normal definido en el trabajo de VIPP®. Otra posibilidad es utilizar la clave de trama en combinación con una Colorkey con esta sintaxis:

```
[ Colorkey PATkey ]
```

## Ejemplos

```
/Star /PStar 5 -5 0 .8 .8 SETPAT % pattern definition
Star SETTXXC % invoke pattern with current color
[ BLUE STAR ] SETTXXC % invoke pattern with color BLUE
/VP01 [ /VPGL 1 OWHITE GL_Black ] 0 0 0 1 1 SETPAT % define a Gloss effect
/VP02 [ /VPCR 1 OWHITE CR_Red50 ] 0 0 0 1 1 SETPAT % define a Correlation effect (1
layer)
/VP03 [ /VPCR 1 OWHITE BLUE CYAN ] 0 0 0 1 1 SETPAT % define a Correlation effect (2
layers)
/VP04 [ /VPPG1 12 36 OWHITE BLACK ] 0 0 0 1 1 SETPAT % defines voidpantograph effect
```

Se proporciona un conjunto de patrones predeterminados en el archivo `src/xgf.gep`. Para obtener una muestra, imprima `xgf/demo/sampat.nm`.

## Uso de la trama de vectores GL/CR

Los efectos de la trama de vectores de correlación y brillo tienen como objetivo sustituir los efectos de correlación y brillo heredados mediante fuentes dedicadas. Con efectos de tramas de vectores se puede utilizar cualquier fuente normal (excepto las fuentes de tipo 3). Después de definir el efecto de la trama de vectores

mediante **SETPAT**, el uso es similar al efecto de infrarrojos o fluorescente. Se puede usar un patrón de vectores GL/CR para rellenar un cuadro y la impresión de texto siguiente o una imagen TIFF de dos niveles en la parte superior del cuadro activará el efecto.

#### Ejemplo:

```
500 3000 1750 300 VP02 DRAWB /NHEB 50 SETFONT 575 2770 MOVETO (Correlation Text) SHL
```

#### Ejemplo de efecto GL en imagen de dos niveles (truk1.tif)

```
226 2400 1000 530 VP01 DRAWB 309 2380 MOVETO (truk.tif) 1.5 0 ICALL
```

#### Ejemplo de efecto CR en imagen de dos niveles (truk.tif)

```
1347 2400 1000 530 VP02 DRAWB 1468 2380 MOVETO (truk.tif) 1.5 0 ICALL
```

Para una correlación de 2 capas, se puede colocar un segundo texto encima del primero.

#### Ejemplo:

```
500 3000 1750 300 VP03 DRAWB /NHEB 50 SETFONT 575 2770 MOVETO (Correlation Text) SHL 65
```

Los colores GL o CR se recomiendan para efectos de tramas de vectores, pero no son obligatorios. Se recomienda realizar las pruebas correspondientes para determinar los colores que proporcionan los mejores efectos en una determinada impresora. Se proporcionan 3 VPC en la carpeta `xgf/dem` para mostrar los efectos y facilitar la determinación de los mejores:

- SI\_VP\_GlossMark.vpc
- SI\_VP\_Correlation\_1L.vpc
- SI\_VP\_Correlation\_2L.vpc

#### Superposición de transparencia de claves de correlación

Para crear una transparencia de clave de correlación de patrones de vectores, cree un archivo simple de modo nativo de VIPP® con los contenidos siguientes:

```
/VPKey [ /VPCR 1 WHITE BLACK ] 0 0 0 1 1 SETPAT 100 100 2000 3000 VPKey DRAWB PAGEBRK
```

e imprímalo en una transparencia. Para la segunda capa, la clave debe girarse 90 grados (girar = 90). Este tipo de archivo se proporciona en la carpeta `xgf/demo`:

```
SI_VP_Correlation_key.nm
```

#### Variantes personalizadas de correlación

Tenga en cuenta que mediante la variación de los parámetros de **SETPAT** `Htrans`, `Vtrans`, `rotate`, `Hscale` e `Yscale` es posible crear variantes de correlación personalizadas que solo funcionarán con la transparencia de clave asociada mediante el uso de los mismos parámetros.

#### Uso del pantógrafo vacío

El efecto de pantógrafo vacío tiene la finalidad de asegurar y proteger los documentos originales frente a copia o manipulación. El efecto de pantógrafo vacío puede utilizar cualquier fuente normal excepto fuentes de tipo 3.

El efecto de pantógrafo vacío se ha definido utilizando el comando **SETPAT**. Se puede utilizar un pantógrafo vacío para rellenar un cuadro y cualquier texto a continuación con píxeles.

#### Ejemplo:

```
4 3499 2479 172 VP04 DRAWB 8 3350 MOVETO /NHEB 50 SETFONT % (Helvetica Bold) (Void Void
```

Se proporcionan `SI_VP_VoidPanto.vpc` en la carpeta `xgf/dem` para mostrar los efectos y facilitar la determinación de los mejores:

#### Modos

Este comando es aplicable en todos los modos.

#### Comandos relacionados

- **SETPPAT**
- **SETTXC**
- **INDEXCOLOR**
- **SETTPAT**

## SETPBRK

Utilice **SETPBRK** para especificar cualquier cadena como un delimitador de página. Cuando se detecta esta cadena, la línea en la que aparece puede ser una de las siguientes:

- La última línea de la página actual
- La primera línea de una página nueva
- Delimitador de página no imprimible
- Dividir en una parte izquierda y una parte derecha

En el último caso, el proceso puede repetirse de forma recursiva en el lado derecho.

### Sintaxis

```
(delimiter string) option SETPBRK
```

### Donde:

**cadena de delimitación** es la cadena que identifica una delimitación de página.

**opción** es una serie de tres dígitos con los siguientes valores:

Primer dígito:

**0** la línea coincidente es la última línea

**1** la línea coincidente es la primera línea

Segundo dígito:

**0** imprimir la parte izquierda si no está vacía

**1** imprimir la parte izquierda

**2** imprimir la parte derecha

**3** imprimir toda la línea

**4** no imprimir ninguna

Tercer dígito:

**0** realizar bucle en el lado derecho. Comprueba si hay más de una instancia de la cadena de prueba en el registro.

**1** no realizar un bucle en la parte derecha. No comprueba si hay más de una instancia de la cadena de prueba en el registro.

### Ejemplos

Esta es la opción prefijada para la compatibilidad con versiones anteriores:

```
<0C> 000 SETPBRK
```

Este ejemplo configura el avance de formulario como delimitador predefinido de la página, permite el procesamiento de varios avances de formulario en una sola línea, y evita páginas vacías en varios avances de

formulario adyacentes.

```
<0C> 010 SETPBRK      % full ASCII printer compatibility
<0C> 001 SETPBRK      % FF at the end of the last line
                        % suppresses extra blank line on the next page
(Page) 131 SETPBRK     % assumes that a line with Page is the first line
($$INDEX) 141 SETPBRK % detects a nonprintable delimiter
                        % this line may contain additional information such as
                        % archive indexing
```

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

PAGEBRK, XGFDEBUG



## SETPCC

**SETPCC** permite el procesamiento PCC del trabajo actual. Si PCC se activa, los espacios verticales son controlados por el primer byte de cada registro (byte PCC), cuyas acciones se definen en la definición de PCC. Consulte **BEGINPCC** para obtener más información.



Nota: Si se usan bytes PCC en entradas **FROMLINE/RPEKEY**, donde **recpos=0** hace referencia al primer byte de datos (omitiendo el PCC), la columna uno del archivo de datos original no se tratará como datos de usuario. Se utiliza como el índice de PCC.

Sin embargo, esto no es así para **GETFIELD**, **SETRCD** y **SETPCD**, donde **recpos=0** siempre hace referencia al primer byte del registro, independientemente de si **SETPCC** se codifica o no.

### Sintaxis

```
/pccname SETPCC
```

### Donde:

**pccname** se refiere a una definición de PCC previamente definida por **BEGINPCC**.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**BEGINPCC**, **ENDPCC**, **SETVFU**

## SETPCD

El comando **SETPCD** establece una definición de criterios de la página. Aunque este comando es similar a **SETRCD**, la condición se evalúa a nivel de página en vez de a nivel de registro. Las pruebas mediante **SETPCD** tienen un resultado verdadero o falso.

Todos los **PCD** se evalúan para cada página antes de la composición de páginas. Por lo tanto, las **PCDkeys** se pueden usar en procedimientos **BEGINPAGE** o **ENDPAGE**, en formularios con pruebas de modo nativo **IF**, **ELSE** y **ENDIF**, o en una definición **RPE** de la misma manera que se usan las pruebas **RCD**.



Nota: Si se usan bytes **PCC** en entradas **FROMLINE/RPEKEY**, donde **recpos=0** hace referencia al primer byte de datos, omitiendo el **PCC**, la columna uno del archivo de datos original no se tratará como datos de usuario. Se utiliza como el índice de **PCC**. Sin embargo, esto no es así para **GETFIELD**, **SETRCD** y **SETPCD**, donde **recpos=0** siempre hace referencia al primer byte del registro, independientemente de si **SETPCC** se codifica o no.

### Sintaxis

- `/PCDkey line_nr line_ct recpos length /cnd (comp.str) SETPCD`
- `/PCDkey line_nr line_ct field_nr /FN /cnd (comp.str) SETPCD`

### Donde

- **PCDkey** es el nombre **PCD**.
- **line\_nr** es el número de línea en las que se puede evaluar la condición a partir de 1. Si **line\_nr** está fuera del intervalo, no se producirá ningún error, pero la prueba tendrá un resultado falso.
- **line\_ct** es el número de líneas, empezando por **line\_nr**, en las que se evalúa la condición. Si **line\_nr + line\_ct** es mayor que el número de líneas de la página actual, no se producirá ningún error y **line\_ct** asumirá el valor predefinido del número máximo de líneas.
- **recpos/length** selecciona la parte de registro de la línea que va a compararse utilizando la cadena de comparación. **recpos** comienza por 0.
- **field\_nr** especifica el número del campo para compararlo con la cadena de comparación. **field\_nr** empieza por 0 y se aplica a todos los registros con una estructura con delimitación de campos. El delimitador de campo se define con **SETDBSEP**. El valor predeterminado son dos puntos (:).
- **/cnd** especifica el operador de prueba, por ejemplo, **/eq**, **/ne**, **/ge**, **/gt**, **/le**, **/lt**, **/CIEQ**, **/CINE** y **/HOLD**. **CIEQ** y **CINE** comparan la parte de registro seleccionada (con **comp.str**) independientemente de si se trata de mayúsculas o minúsculas, **/HOLD** busca la cadena de comparación en cualquier lugar de la parte de registro seleccionada.

Para obtener más información, consulte [Operadores de prueba y expresiones condicionales](#).

- **comp.str** es la cadena de referencia para la prueba. Para las cadenas largas que repiten el mismo carácter, utilice un valor de recuento. Por ejemplo, `100 (*)`.

Utilice **/PREV** para hacer referencia a la cadena equivalente en la página anterior. **/ne/PREV** siempre es **TRUE** en la primera página. Cuando se utiliza **/PREV**, el operando **line\_ct** debe ser 1.

**comp.str** también puede ser una variable con el formato **/VARxxx**. Permite cambiar la cadena de comparación

durante el trabajo.



Nota: **SETPCD** también se aplica a registros NMP. Permite comprobar los datos no imprimibles integrados en comentarios NMP, por ejemplo: %%XGF% marker 1.

## Ejemplos

- En este ejemplo, la condición es verdadera si la cadena `FORM=` aparece en las primeras 132 posiciones de cualquier línea, de la 10 a la 18.

```
/BANNER 10 9 0 132 /HOLD (FORM=) SETPCD
```

- En este ejemplo, la condición es verdadera si las seis primeras posiciones de la línea 10 son distintas a las de la página anterior. En la primera página, la condición siempre es verdadera.

```
/NEWDEPT 10 1 0 6 /ne /PREV SETPCD
```

- En este ejemplo, la condición es verdadera si la cadena `START OF JOB :` aparece en la posición 5 de un total de 13 posiciones de caracteres en cualquier línea, de la 3 a la 5.

```
/STARTBANNER 3 3 5 13 /eq (START OF JOB :) SETPCD
```

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

- **BEGINPAGE**
- **ENDPAGE**
- **GETFIELD**
- **IF/ELSE/ELIF/ENDIF**
- **SETRCD**

## SETPIF

**SETPIF** define un destino o nota de **PIF** que se asociará con el siguiente y único elemento impreso en la página usando **SHP**, **SHMF**, **SHX**, **ICALL**, **SCALL**, **DRAWB** o **BOOKMARK**.

### Sintaxis

```
[ /PIFtype param1 param2 . . . . ] SETPIF
```

```
null SETPIF
```

### Donde:

<b>PIFtype</b>	puede ser uno de los siguientes:
<b>PAGE</b>	una página en el documento
<b>DEST</b>	un destino con nombre definido por <b>PDFDEST</b>
<b>XPAGE</b>	una página en otro documento PDF
<b>XDEST</b>	un destino con nombre en otro documento PDF
<b>FILE</b>	un documento no PDF
<b>URI</b>	un sitio de Internet o Intranet o un documento
<b>NOTE</b>	una nota
<b>nulo</b>	cancela las definiciones de PIF activas.



Nota: Dado que los PIF activos se cancelan automáticamente si se asocian con un elemento, el uso de la sintaxis nula pocas veces es necesaria.

En función de PIFtype, deben suministrarse estos parámetros:

- [ /PAGE pagenum view ]
- [ /PAGE {/Prev} view ]
- [ /PAGE {/Next} view ]
- [ /DEST /destname ]
- [ /XPAGE (fileref) pagenum view ]
- [ /XDEST (fileref) /destname ]
- [ /FILE (fileref) ]
- [ /URI ]
- [ /URI (URIstring) ]
- [ /NOTE (title) (contents) ]
- [ /NOTE (title) (contents) notetype color option ]

### Donde:

<b>pagenum</b>	es el número de página del destino empezando por 1. Puede ser una variable.
<b>{/Prev}</b>	es el destino de la página anterior
<b>{/Next}</b>	es el destino de la página siguiente
<b>ver</b>	define cómo ajustar la vista para un destino de página. Puede ser uno de los siguientes:  <b>null</b> utilizar la vista actual.  <b>[/Fit]</b> ajustar la página a la ventana.  <b>[/FitB]</b> ajustar el cuadro delimitador del contenido de la página a la ventana.  <b>[/FitR left bottom right top]</b> el rectángulo de la página especificado por el cuadro delimitador izquierda abajo derecha arriba se amplía y se centra en la ventana del visualizador.  <b>[/FitH top]</b> ajustar la anchura de la página a la ventana y coloca su origen superior encima de las unidades actuales del origen de la página.  <b>[/FitV left]</b> ajustar la altura de la página a la ventana y coloca su origen izquierdo a la izquierda de las unidades actuales del origen de la página.  <b>[/XYZ left top zoom]</b> colocar el origen de la ventana en unidades actuales en el lado superior izquierdo del origen de la página.
<b>destname</b>	es un nombre de destino. Puede ser una variable.
<b>fileref</b>	es la ruta completa o relativa y el nombre de un archivo externo. Consulte el parámetro <a href="#">/ResolvePath</a> para configurar el modo de resolución de la ruta.  Para obtener más información, consulte <a href="#">Operadores de prueba y expresiones condicionales</a> .
<b>URIstring</b>	es un enlace a un nombre de documento o sitio de Internet o Intranet. Si se omite, el texto del elemento asociado (impreso por <b>SHP</b> o <b>SHMF</b> ) se utiliza en su lugar.
<b>title</b>	es el título de la nota.
<b>contents</b>	es el contenido de la nota. Puede ser una variable.  Los datos de un solo byte pueden cifrarse mediante ISO-8859-1. Los datos de varios bytes pueden cifrarse mediante UTF8. VI Compose los puede convertir automáticamente en UTF16 para insertarlos en el PDF porque esta es la única codificación de varios bytes admitida por el formato PDF. Para activar la conversión de datos UTF8 a UTF16, la fuente actual seleccionada por <b>SETFONT</b> o <b>INDEXFONT</b> puede tener una codificación UTF8.
<b>notetype</b>	es una de estas claves por categoría:  <b>Iconos</b> <ul style="list-style-type: none"><li>• <b>/Nota</b></li><li>• <b>/Comentario</b></li><li>• <b>/Ayuda</b></li></ul>

- /Insertar
- /Clave
- /NuevoPárrafo
- /Párrafo

#### **sellos**

- /Borrador
- /Aprobado
- /Experimental
- /No aprobado
- /AsIs
- /Caducado
- /NotForPublicRelease
- /Confidencial
- /Final
- /Vendido
- /Departamentos
- /Para comentar
- /Máximo secreto
- /Para publicación pública

**free text** /FT

**markup** /MK

#### **color**

es una de las siguientes:

Una **Colorkey** que define el color de la nota

**null** si no se requiere ningún color. Se admiten solo RGB sin formato y color de escala de grises.

#### **opción**

es una de las siguientes:

**+1** la nota se presenta abierta (solo se aplica a iconos). El valor predeterminado es cerrado.

**+2** notetype es una marca.

**+4** centrar el icono de nota en el elemento delimitador.

Cuando se omiten notetype, color y option, /Note, YELLOW y 0 se configuran de forma prefijada.

#### **Modos**

Este comando es aplicable en todos los modos.

#### **Comandos relacionados**

- PDFDEST
- INDEXPIF

- BOOKMARK
- PDFINFO
- PDFOPEN

## SETPPAT

El comando **SETPPAT** define una trama de prototipos para su uso con el comando **SETPAT**. Una trama es un dibujo pequeño que se puede utilizar para rellenar un área de forma repetida. Una vez definido, puede utilizarse en lugar de un color, o en combinación con el mismo.

### Sintaxis

```
/PPATkey PaintType [BBox] Xstep Ystep {PaintProc} SETPPAT
```

### Donde:

<b>PPATkey</b>	es el nombre de la trama de prototipo para definir cualquier cadena alfanumérica que empieza por un carácter alfabético.
<b>PaintType</b>	es un número entero que especifica el tipo de dibujo como uno de los siguientes: <ol style="list-style-type: none"> <li>1 Patrón de color. El color se especifica en PaintProc.</li> <li>2 Patrón sin color. El color se puede definir antes del uso del patrón.</li> </ol>
<b>BBox</b>	es el cuadro de límites, una matriz de 4 números en puntos que proporcionan la posición X inferior izquierda, Y inferior izquierda, X superior derecha, Y superior derecha de la celda de la trama.
<b>Xstep</b>	es el espaciado horizontal entre las celdas en puntos.
<b>Ystep</b>	es el espaciado vertical entre las celdas en puntos.
<b>PaintProc</b>	es un procedimiento de lenguaje PostScript para dibujar celdas.

### Ejemplos

```
/PStar 2 [-5 -5 5 5] 10 10{pop 0 5 moveto 4 {144 rotate 0 5 lineto} repeat  
closepath fill} SETPPAT
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETPAT**, **SETTPAT**



## SETPPATH

El comando **SETPPATH** define una lista de bibliotecas de proyectos. Las rutas de bibliotecas pueden contener las secuencias de claves **\$\$FOLDER** y **\$\$PROJECT** como marcadores de posición de carpetas y nombres de proyectos. Los valores predeterminados se definen en el archivo `/usr/xgf/src/xgfunix.run` o `x:\xgf\src\xgfdos.run`.

Utilice los comandos **SETPPATH** solo en los archivos `xgfunix.run` o `xgfdos.run`. La agregación de **SETPPATH** a un trabajo VIPP® pone en riesgo la portabilidad de este. La agregación de **SETPPATH** a un proyecto VI pone en riesgo la portabilidad y la organización del proyecto.

### Sintaxis

```
[ (path to project library 1) (path to project library 1) ... ] SETPPATH
```

Los parámetros definidos por **SETPPATH** se aplican si un solo comando **SETPROJECT** se coloca al principio del trabajo VIPP®. Solo debe haber un comando **SETPROJECT** por trabajo, y puede estar colocado al principio del trabajo, antes de otros comandos que hacen referencia a un origen.

Al utilizar **SETPPATH** y **SETPROJECT**, todas las bibliotecas de recursos se redefinen a aquellas definidas por **SETPPATH** para todos los tipos de recursos; todas las definiciones anteriores **SETEPATH**, **SETFPATH**, **SETIPATH**, **SETJPATH** y **SETMPATH** se sustituyen por la definiciones **SETPPATH**.

Las rutas de proyectos se dividen en estas tres categorías (ámbitos):

<b>Ámbito local</b>	rutas con <b>\$\$FOLDER</b> y <b>\$\$PROJECT</b> . Estas bibliotecas pueden contener los recursos pertenecientes solo al proyecto.
<b>Ámbito de la carpeta</b>	rutas con solo <b>\$\$FOLDER</b> . Estas bibliotecas pueden contener bibliotecas y recursos del proyecto compartidos por proyectos que pertenecen a la misma carpeta.
<b>Ámbito global</b>	rutas sin <b>\$\$FOLDER</b> ni <b>\$\$PROJECT</b> . Estas bibliotecas pueden contener recursos compartidos por todos los proyectos.

### Reglas y limitaciones:

- En la categoría de ámbito Local, **\$\$PROJECT** sigue inmediatamente a **\$\$FOLDER**.
- No se admiten rutas con **\$\$PROJECT** sin **\$\$FOLDER**.
- Si están presentes, **\$\$FOLDER** y **\$\$PROJECT** aparecen solo una vez en cada ruta.
- No se admiten componentes de ruta adicionales después de **\$\$PROJECT**.
- Una ruta que termine con **\$\$PROJECT** no es válida.
- Puede haber al menos una ruta para cada categoría.
- Hay diversas rutas en cada categoría, pero pueden definirse y agruparse por categoría (local, carpeta, global) en la lista **SETPPATH**.
- Aparece un nombre de carpeta o proyecto una sola vez en los árboles de directorios cubiertos por **SETPPATH**.

- Cuando hay un recurso con el mismo nombre en más de un ámbito, el orden de prioridad es: local, carpeta, global.
- Para mejorar la portabilidad entre plataformas, Xerox recomienda que los nombres de CARPETA y de PROYECTO no contengan más de 32 caracteres, y solo usen los caracteres desde a hasta z, 0 a 9, punto (.), guión (-) y guión bajo (\_).

### Ejemplos

```
[ (c:\\xgfc\\$$FOLDER.\\$$PROJECT.\\)
  (c:\\xgfc\\$$FOLDER.\\fshared\\)
  (c:\\xgfc\\gshared\\)
] SETPPATH
```

Se admiten solo las resoluciones compatibles con el dispositivo actual y disponibles a través de PostScript. Compruebe los valores de resolución legales en la documentación de la impresora. Este comando se admite solamente en dispositivos de nivel 2.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

SETPROJECT

## SETPROJECT

El comando **SETPROJECT** activa la definición de **SETPPATH** con el proyecto y la carpeta relacionados para todos los accesos de recursos posteriores. Utilice el comando **SETPPATH** solo en los archivos `xgfunix.run` o `xgfdos.run`.

Si se usa **SETPROJECT**, puede definirse solo una vez y colocarse al principio del trabajo antes de los comandos que hacen referencia a un recurso.



**Nota:** Para mejorar la portabilidad a través de plataformas, se recomienda que los nombres de carpeta y de proyecto no contengan más de 32 caracteres, y solo usen los caracteres a–z, 0–9, . punto, – guión y \_ subrayado.

### Sintaxis

```
[ (folder_name) (project_name) ] SETPROJECT
```

### Donde:

**folder\_name** es el valor de sustitución de `$$FOLDER`.  
**project\_name** es el valor de sustitución de `$$PROJECT`.

### Ejemplos

```
%!
```

```
[(projects) (holidays)] SETPROJECT
```

```
(holi1.dbm) STARTDBM
```

```
.....
```

Se admiten solo las resoluciones compatibles con el dispositivo actual y disponibles a través de PostScript. Compruebe los valores de resolución legales en la documentación de la impresora. Este comando se admite solamente en dispositivos de nivel 2.

Si se asocia con el ejemplo presentado en **SETPPATH**, el comando **SETPROJECT** puede asignar estas bibliotecas para realizar la búsqueda de recursos durante el resto del trabajo:

- `c:\xgfc\projects\holidays`
- `c:\xgfc\projects\fshared`
- `c:\xgfc\gshared`

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETPPAT**

## SETRCD

**SETRCD** establece una definición de criterios de registro (RCD) para su uso en las definiciones de RPE posteriores. Para obtener más información, consulte [Información de comando RPE](#) y otros comandos RPE relacionados.



Nota: Si se usan bytes PCC en entradas **FROMLINE/RPEKEY**, donde **recpos=0** hace referencia al primer byte de datos (omitiendo el PCC), la columna uno del archivo de datos original no se tratará como datos de usuario. Se utiliza como el índice de PCC. Sin embargo, esto no es así para **GETFIELD**, **SETRCD** y **SETPCD**, donde **recpos=0** siempre hace referencia al primer byte del registro, independientemente de si **SETPCC** se codifica o no.

### Sintaxis

- `/RCDkey recpos length /cond (compare string) SETRCD`
- `/RCDkey field_nr /FN /cond (compare string) SETRCD`
- `/RCDkey [ RCDkey1 RCDkey2 /bool_op ] SETRCD`
- `/RCDkey { condition statement } SETRCD`

### Donde

- **RCDkey** es el nombre RCD usado en la definición de RPE.
- **recpos/length** selecciona la parte de registro de la línea que va a compararse utilizando la cadena de comparación. **recpos** comienza por 0.
- **field\_nr** especifica el número del campo para compararlo con la cadena de comparación. **field\_nr** empieza por 0 y se aplica a todos los registros con una estructura con delimitación de campos. El delimitador de campo se define con **SETDBSEP**. El valor predeterminado son dos puntos ( : ).
- **cond** especifica el operador de prueba, por ejemplo, `/eq`, `/ne`, `/ge`, `/gt`, `/le`, `/lt`, `/CIEQ`, `/CINE` y `/HOLD`. **CIEQ** y **CINE** comparan la parte de registro seleccionada (con `comp.str`) independientemente de si se trata de mayúsculas o minúsculas, `/HOLD` busca la cadena de comparación en cualquier lugar de la parte de registro seleccionada.

Para obtener más información, consulte [Operadores de prueba y expresiones condicionales](#).

- **compare string** es la cadena de referencia para la prueba. Puede ser expresado mediante un valor de contador. Por ejemplo, `100 (*)`.
- **/bool op** es un operador booleano, como `/or`, `/and` y `/not`, utilizado para combinar varias definiciones de RCD.
- **{ condition statement }**
- es la secuencia de comandos VIPP® esperada para producir un booleano. Es probable que se usen variables integradas o personalizadas en esta instrucción.

Después de definirla, la RCDkey se puede colocar antes de la entrada de RPE, lo que hace que esta entrada solo sea procesada si la condición resultante es verdadera. Si la siguiente entrada está precedida por `/ELSE`, se procesa solamente si el RCD anterior es falso.

Cuando se utiliza `/ENDIF`, la condición y el siguiente `/ELSE` se pueden aplicar a varias entradas de RPE, y pueden

anidarse. Consulte los ejemplos de las extensiones **FROMLINE** para obtener más información.

## Ejemplos

Este ejemplo muestra cómo se imprime una línea que comienza con la palabra **TOTAL** mediante una fuente diferente.

```
/IF_TOTAL 0 5 /eq (TOTAL) SETRCD 5 BEGINRPE % align rotate init displ Yinit Ydispl
```

Este ejemplo define un nuevo RCD que es verdadero si **IF\_CND1 RCD** o **IF\_CND2 RCD** es verdadero.

```
/IF_CND3 [ IF_CND1 IF_CND2 /or ] SETRCD
```

Este ejemplo define un nuevo RCD que es verdadero si **IF\_CND1 RCD** o **IF\_CND2 RCD** es verdadero e **IF\_CND3 RCD** también lo es. Las combinaciones complejas pueden afectar al rendimiento.

```
/IF_CND4 [ IF_CND1 IF_CND2 /or IF_CND3 /and ] SETRCD
```

Estos dos ejemplos son equivalentes.

- `/IF_CND1 0 10 /eq 10 (*) SETRCD`
- `/IF_CND1 0 10 /eq (*****) SETRCD`

Este ejemplo es verdadero cuando la palabra **DIVISION** aparece en cualquiera de las primeras 100 posiciones del registro. La búsqueda de una cadena en los registros y en cada byte del registro puede afectar el rendimiento.

```
/IF_CND1 0 100 /HOLD (DIVISION) SETRCD
```

Esto es un ejemplo con la variable integrada **CURLINE**.

```
/IF_1STLINE { CURLINE 1 eq } SETRCD 2 BEGINRPE /HEAD RPEKEY /IF_1STLINE [ ..... ] % e
```

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

- **CURLINE**
- **FROMLINE**
- **LNCOUNT**
- **RPEKEY**
- **SETPCD**

## SETRES

El comando **SETRES** establece la resolución del dispositivo en la página actual y la posterior.

### Sintaxis

Res SETRES

### Ejemplos

600 SETRES

Se admiten solo las resoluciones compatibles con el dispositivo actual y disponibles a través de PostScript. Compruebe los valores de resolución legales en la documentación de la impresora. Este comando se admite solamente en dispositivos de nivel 2.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

Ninguno

## SETRPE

El comando **SETRPE** llama a una definición de RPE configurada por **RPEKEY** para las líneas que siguen al comando. Este comando está integrado en el flujo de datos con un prefijo de modo nativo (NMP). Para obtener más información, consulte [Información de comando RPE](#) y otros comandos RPE relacionados.

### Sintaxis

```
/rpekeyname SETRPE
```

**SETRPE** y **SETRPEPREFIX** son excluyentes entre sí.

### Ejemplos

Es un ejemplo que llama a ADR0 para las siguientes líneas hasta que se encuentra el siguiente **SETRPE**.

```
%%XGF /ADR0 SETRPE
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

[RPEKEY](#), [SETRPEPREFIX](#)

## SETRPEPREFIX

El comando **SETRPEPREFIX** activa el modo de prefijo RPE y la longitud del prefijo. Para obtener más información, consulte [Información de comando RPE](#) y otros comandos RPE relacionados.

### Sintaxis

```
prefixlength SETRPEPREFIX
```

```
[prefixlength position] SETRPEPREFIX
```

### Donde:

<b>prefixlength</b>	es la longitud del prefijo.
<b>posición</b>	es la posición del prefijo de los registros empezando por 0 (el valor prefijado es cero).

La segunda sintaxis es compatible con un prefijo empezando por cualquier posición en el registro. Use esta opción para migrar aplicaciones **LCDS** que usan un índice de fuentes para asignar el byte de índice de la fuente a una definición de **RPEKEY**. Consulte [RPEDEF](#).

Si el modo de prefijo de RPE se activa, todas las líneas en el archivo de datos empiezan con un prefijo que invoca la definición de RPE coincidente definida por **RPEKEY** en el JDT para esta línea.

Si no se define el prefijo de una determinada línea:

- Se saltará la línea si **prefixlength** tiene un valor positivo
- El trabajo se cancela con un mensaje de error de **VIPP\_RPE\_invalid\_prefix** en lugar de **/rpekeyname**.



**Nota:** Siempre que un prefijo RPE contenga espacios puede programarse entre paréntesis, en lugar de ser precedido por un carácter /. Por ejemplo:

```
(REC1 45 C) RPEKEY
```

```
[ ... RPE entry ... ]
```

Un grupo RPE está formado por líneas consecutivas que utilizan el mismo prefijo pero tienen un último dígito distinto. Esto indica que solo el último dígito del prefijo cambia con respecto a la línea anterior. La línea pertenece al mismo grupo RPE.

Un prefijo RPE puede tener como mínimo 2 bytes de longitud, a menos que el prefijo RPE se use para procesar un flujo de datos LCDS mediante **fontindex**. Para obtener más información, consulte [RPEDEF](#). Esto se debe a la posibilidad de usar grupos RPE. Si el prefijo tiene solo un byte de longitud, todas las definiciones de RPE pertenecen al mismo grupo único. Si se da el caso, se procesan todos los registros de la misma manera.

Use distintos prefijos en un grupo para especificar varios campos de atributos como, por ejemplo, la fuente, el color, la alineación y la posición, si distintos tipos de registros aparecen en orden aleatorio. Por ejemplo, en un extracto bancario, las líneas de crédito pueden tener el prefijo **LINEC**, y las líneas de débito pueden tener el prefijo **LINED**. Esto permite la impresión de la cantidad con un color diferente o en una columna diferente.

Cuando un flujo de datos usa prefijos que no siguen la regla del último dígito, utilice **NEWGROUP**.



Para obtener más información sobre el procesamiento de RPE, consulte [RPEKEY](#) y [FROMLINE](#)

Ejemplos

Este es un ejemplo de un flujo de datos mediante registros prefijados.

```
%!  
4 SETRPEPREFIX          % should normally be coded in the JDT  
(bill.jdt) STARTLM  
PER0 01.11.1989 - 31.12.1989  
REF0 14153 01764960  
  
ADR0 M. MARCEL DUPONT  
ADR0 RESIDENCE "LES MIMOSAS"  
ADR0 PLACE DE LA GARE, 44  
ADR0 2323 SAGEX  
  
DFA0 23.01.1990  
DPY0 23.02.1990  
DRE0 7.12.1989  
CRN0 068.025.000 COMMUNS IMMEUBLE  
  
LFA0 14 COMMUNS          81978  30248  30144   104          44,40  
LFA0 41 EAU TARIF I      491    12869  12714   155  0,2800    43,40  
LFA0   TAXE DE BASE                2,0  7,692    15,35  
LFA0   LOCATION COMPTEUR            2,0  2,00     4,00  
LFA0 82 GAZ CHAUFFAGE    3500    40972  40126   8798  0,0330   290,35  
LFA0   TAXE DE BASE                2,0 16,666    33,35  
LFA1 TOTAL S.I.                                430,85  
LFA0 41 ASSAINISSEMENT    491                155   0,25    38,75  
  
MNT0 MONTANT A PAYER      469,60  
BRE0 00 00014 15301 76496 00120 01019  
OLI0 0100000469609>000001415301764960012001019+  
%%EOF
```

Este ejemplo de [posición de longitud de prefijo] **SETRPEPREFIX**, define el prefijo como de 1 byte de longitud en la posición 132 del registro. La posición predefinida es cero.

```
[ 1 131 ] SETRPEPREFIX
```

Modos

Este comando es aplicable en todos los modos.

Comandos relacionados

[RPEKEY](#), [SETRPE](#)

## SETSKIP

El comando **SETSKIP** es una alternativa a **SETPCC**. Además, proporciona un mecanismo para gestionar situaciones de impresiones huérfanas y ventanas. **SETSKIP** y **SETPCC** son exclusivos.

### Sintaxis

```
[ skip_cond1 [ pre-skip print-action post-skip BOF TOF ]
  skip_cond2 [ pre-skip print-action post-skip BOF TOF ]
  .....
  skip_condn [ pre-skip print-action post-skip BOF TOF ]
  ()         [ pre-skip print-action post-skip BOF TOF ] %default
] SETSKIP
```

### Donde:

**skip\_condx**

es una de las siguientes:

Una **cadena** de cualquier longitud entre paréntesis. Si el principio de una línea coincide con la cadena, se produce la acción de omisión asociada. La cadena permanece como parte de los datos de impresión.

Una **RCDkey**. Si la condición de RCD es verdadera para esa línea, se produce la acción de omisión asociada.

**pre\_skip**

es una de las siguientes:

Un **número de líneas**

Una **tecla** que se asignará más adelante con **SETVFU**

**print\_action**

es una de las siguientes:

**True**, imprime el registro

**False**, no se imprime el registro

**post\_skip**

es una de las siguientes:

Un **número de líneas**

Una **tecla** que se asignará más adelante con **SETVFU**

**BOF**

define el umbral inferior. Se produce un salto a la página siguiente, antes de la impresión, si el número de líneas restantes en la página después de la omisión previa es inferior al número. Se define el número de líneas por página mediante el comando **SETGRID**.

**TOF**

define el umbral superior. Si no es cero, **TOF** hace que la línea sea la línea objetivo (independientemente de la omisión previa) si se produce una transición de página durante el proceso de omisión previa.

### Ejemplos

En este ejemplo cualquier línea que empiece por 0001 puede generar un salto al canal uno.

```
[ (0001) [ /SK1 true 0 0 0 ]  
( ) [ 1 true 0 0 0 ]  
] SETSKIP
```

**Modos**

Este comando es aplicable en todos los modos.

**Comandos relacionados**

- SETGRID
- SETPBRK
- SETPCC
- SETVFU

## SETTAB

El comando **SETTAB** permite el procesamiento de tabuladores y define el espaciado de los tabuladores en unidades de carácter.



Nota: **SETTABS** se utiliza para definir la anchura de la ficha en unidades actuales. La anchura de la ficha es el espacio asignado cada vez que se procesa el carácter de la ficha. **SETTABS** se usa para configurar una serie de tabulaciones para su uso en bloques de texto, por ejemplo, para alinear columnas. Con **SETTABS**, la posición de impresión pasa a la siguiente tabulación.

### Sintaxis

```
charcount SETTAB
```

### Donde:

**charcount** es el número de caracteres que se utilizan para la colocación de tabuladores.

### Ejemplos

Este ejemplo define una pausa de tabulación cada ocho caracteres.

```
8 SETTAB
```



Nota: Use SETTAB solamente cuando sea necesario porque puede afectar al rendimiento.



Nota: Utilice **SETUTAB** con fuentes proporcionales.

### Modos

Este comando es aplicable en el modo de línea básica (sin RPE).


### Comandos relacionados

**SETUTAB**, **SETTABS**

## SETTABS

**SETTABS** activa el procesamiento de tabuladores en bloques de texto que se imprimen con **SHP** o **SHMF**. Si se ha codificado el comando **SETTABS**, **SHP** o **SHMF** pueden garantizar la integridad de los caracteres de tabulación en la cadena.

- Si los tabuladores se activan, solo se admite la alineación izquierda para **SHP** y **SHMF**. Con **SHP**, el ajuste de líneas no tiene en cuenta a los tabuladores, por lo que cada línea con tabuladores puede ajustarse a la anchura de la columna definida.
- Si configura la opción +20, varias líneas pueden colocarse en un bloque de texto.

 Nota: **SETTABS** se utiliza para definir la anchura de la ficha en unidades actuales. La anchura de la ficha es el espacio asignado cada vez que se procesa el carácter de la ficha. **SETTABS** se usa para configurar una serie de tabulaciones para su uso en bloques de texto, por ejemplo, para alinear columnas. Con **SETTABS**, la posición de impresión pasa a la siguiente tabulación.

### Sintaxis

[ [ tab\_stop1 ] [ tab\_stop2 ] .... default\_tab\_space ] SETTABS

Donde:

[ tab_stopX ]	Puede tener el formato:  [ align position ]: Sin carácter de entrada ni de alineación  [ align position null null ]: Sin carácter de entrada ni de alineación  [ align position (leader) null ]: Carácter de entrada, sin carácter de alineación  [ 1 position null (align_char) ]: Sin carácter de entrada, carácter de alineación  [ 1 position (leader) (align_char) ]: Carácter de entrada, carácter de alineación
alinear	es una de:  0: Tabulador izquierdo  1: Tabulador derecho  2: Tabulador centro
posición	Posición de tabulación, en unidades actuales, en relación a la posición de impresión actual principal.
default_tab_space	Se aplica si se han agotado todos los puntos de tabulación.

### Ejemplos

[ [ 0 20 ] [ 1 50 ] [ 1 100 null (.) ] [ 2 150 (\*-\*) null ] [ 1 100 (\*-\*) (.) ] ]

### Modos

Este comando es aplicable en todos los modos.

**Comandos relacionados**

SETTAB, SETUTAB

## SETTRAN

**SETTRAN** establece la transparencia de todos los comandos de marca posteriores, incluidas las imágenes. El comando solo funciona en VI eCompose y APPE RIP. En un PS RIP se emula mediante un matiz opaco.

### Sintaxis

TRlevel SETTRAN

### Donde:

TRlevel	es un número real entre 0 y 1 que especifica el nivel de transparencia. Solo funciona en VI eCompose y APPE RIP. En un PS RIP se emula mediante un matiz opaco.
---------	---

### Ejemplos

Este ejemplo establece la transparencia al 50%:

```
. 5 SETTRAN
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

SETCOL, SETTXC, SETGEP

## SETTPAT

Utilice **SETTPAT** para crear una trama de tinta con diseños que incluye una cadena de texto repetida. La cadena de texto puede contener texto variable. Una trama de texto rellena un área repitiendo el texto. Una vez definida, se puede utilizar en lugar de un color para dibujar un objeto.

### Sintaxis

**Para definir una tinta con patrón sin color de fondo:**

```
/tpatname (text) /fname fsize htr vtr rot Tcolor SETTPAT
```

**Para definir una tinta con patrón con color de fondo:**

```
/tpatname (text) /fname fsize htr vtr rot [Tcolor Bcolor] SETTPAT
```

**Donde:**

<b>/tpatname</b>	es el nombre de la trama de texto
<b>(text)</b>	es el texto que se extiende
<b>/fname</b>	es el nombre de la fuente que se usará
<b>fsize</b>	es el tamaño de la fuente en puntos
<b>htr</b>	es un ajuste horizontal en puntos
<b>vtr</b>	es un ajuste vertical en puntos
<b>rot</b>	es el ángulo de giro en grados del texto de la trama
<b>Tcolor</b>	es el color del texto
<b>Bcolor</b>	es el color del fondo

Si la trama se invoca para rellenar una forma, esta se pinta en primer lugar con el Bcolor y la trama de texto se pinta mediante Tcolor sobre ella.

Si se usa la primera sintaxis, Bcolor no se especifica, por lo que no se pintará ningún fondo, a excepción de los siguientes casos:

- Si la fuente es una fuente de GL o CR, se pinta un fondo blanco
- Si Tcolor es una tinta UV/IR, el fondo se pinta mediante tinta alternativa
- Si Tcolor es ARTBLACK\_A o ARTBLACK\_B, el fondo se pinta con negro de contraste

### Ejemplos

**Patrón de texto regular:**

```
/TVipp (VIPP) /NHEB 20 15 15 45 [ RED YELLOW ] SETTPAT
```

**Patrón de texto fluorescente:**

```
/TVipp (VIPP) /NHEB 28 5 2 45 UV_OLIVE1 SETTPAT
```



**Patrón de texto GlossText:**

```
/TVipp (VIPP) /NeueClassic-GL-24 GLT 0 0 45 GL_Magenta SETTPAT
```

**Patrón de texto negro de imágenes:**

```
/TVipp (VIPP) /NHEB 20 5 5 45 ARTBLACK_A SETTPAT
```

El ejemplo siguiente rellena el cuadro con el texto VIPP® con un giro de 45 grados:

```
100 200 400 150 TVipp DRAWB
```

**Modos**

Este comando es aplicable en todos los modos.

**Comandos relacionados**

SETPPAT, SETPAT

## SETTXB

El comando **SETTXB** activa un fondo para todo el texto impreso siguiente mediante un comando **SHx**.

### Sintaxis

```
/BATkey SETTXB  
null SETTXB
```

### Donde:

**BATkey** representa cualquier tecla definida anteriormente con **SETBAT**.

**nulo** desactiva una **BATkey** activa.

### Ejemplos

Este ejemplo muestra cómo activar el gris ligero y el subrayado.

```
UNDL SETTXB  
PB_LT SETTXB
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETBAT**, **SHX**, **INDEXBAT**

# SETTXC


El comando **SETTXC** establece el color y el patrón de todo el texto impreso y los comandos **ICALL** siguientes.

## Sintaxis

`Colorkey SETTXC PATkey SETTXC [ Colorkey PATkey ] SETTXC null SETTXC (ColorKey~Tintlevel`

## Donde:

Colorkey	es el nombre del color utilizado para el texto y los comandos <b>ICALL</b> posteriores.
PATkey	es el nombre del patrón utilizado para el texto y los comandos <b>ICALL</b> posteriores. El color usado es el último color normal definido en el trabajo VIPP®.
[ Colorkey PATkey ]	establece el color y el patrón utilizados para el texto y los comandos <b>ICALL</b> posteriores.  Las Colorkey y los patrones se definen en el archivo <code>/usr/xcfg/src/xcfg.gcp</code> , en Listas estándar, tablas, claves y atributos en la <i>Guía del usuario de FreeFlow VI Compose</i> .
nulo	establece el área que se pinta de color transparente, en vez de rellenarse con un color.
Tintlevel	es un número real, entre 0 y 1, que especifica el nivel de tinta.
TRlevel	es un número real entre 0 y 1 que especifica el nivel de transparencia.

 Nota: TRlevel solo es efectivo en VI eCompose y APPE RIP. En un PS RIP, TRlevel se emula mediante un matiz opaco.

El color prefijado es negro. Las Colorkeys predefinidas incluyen:

- NEGRO
- XLIGHT
- LMEDIUM
- OSCURO
- ROJO
- BLANCO
- CLARO
- MEDIO
- XDARK
- AZUL
- DMEDIUM

- VERDE

Para definir Colorkeys nuevas, edite el archivo `/usr/xgf/src/xgf.gcp` y agregue líneas con esta sintaxis:

```
/Colorkey color_definition SETCOL
```

**Donde:**

**color\_definition** puede ser un número entre 0 (negro) y 1 (blanco) para definir un nivel de gris:

```
/LIGHT .9 SETCOL
```

Consulte la descripción [SETCOL](#) para posibles sintaxis relacionadas con color\_definition.

La trama prefijada es sin trama. Las claves de trama se definen utilizando los comandos **SETPAT** y **SETPPAT**. Entre los patrones predefinidos se incluyen:

- Diamond (Diamante)
- Diamond2 (Diamante 2)
- Cross (Cruz)
- Grid (Cuadrícula)
- Honey (Miel)
- VIPP®
- Strike (Raya)
- Spot (Punto)
- Spot2 (Punto2)
- Star (Estrella)
- Star1 (Estrella1)
- Wave (Onda)

Para obtener más información acerca de los tonos de los colores, consulte [Tonos de color](#).

## Ejemplos

Estos ejemplos muestran cómo aplicar la transparencia a un color.

```
(RED#.5) SETTXC 20 20 300 400 (BLUE#.7) DRAWB
```

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

- [INDEXCOLOR](#)
- [SETPAT](#)
- [SETCOL](#)

- SETTRAN

## SETTXS

Utilice el comando **SETTXS** para activar o desactivar el superíndice y el subíndice según el caso entre comandos **SH**. El subíndice y el superíndice se tratan como atributos de texto, como la fuente, el color y el fondo.

### Sintaxis

```
sst_param SETTXS
[ Ydispl1 Fstring ] SETTXS
```

### Donde:

<b>sst_param</b>	<p>puede ser uno de los siguientes:</p> <p><b>Un valor numérico</b> que representa un desplazamiento vertical en unidades actuales desde la posición vertical actual, positivo para el superíndice y negativo para el subíndice. En este caso, el tamaño de la fuente se gestiona mediante los comandos de la fuente (<b>SETFONT</b>, <b>INDEXFONT</b>).</p> <p><b>/SUP</b> para la activación automática de superíndices. Las fuentes más pequeñas y el desplazamiento se calculan mediante la configuración de la fuente actual.</p> <p><b>/SUB</b> para la activación automática de subíndices. Las fuentes más pequeñas y el desplazamiento se calculan mediante la configuración de la fuente actual.</p> <p><b>null</b> indica el final del subíndice/superíndice.</p>
<b>Ydispl</b>	define el desplazamiento vertical como un factor del tamaño de fuente actual. Pueden ser positivo (superíndice) o negativo (subíndice). El intervalo de valores esperado es [-1 +1].
<b>Fshrink</b>	define el factor de reducción que se aplica a la fuente actual. El intervalo de valores esperado es [>0 +1].

### Ejemplos

En estos ejemplos, imprime March, 17th.

```
100 3000 MOVETO
/NHE 16 SETFONT
(March, 17) SH
/NHE 8 SETFONT
20 SETTXS (th) SH null SETTXS
```

```
100 3000 MOVETO
/NHE 16 SETFONT
(March, 17) SH
/NHE 8 SETFONT
/SUP SETTXS (th) SH null SETTXS %uses VIPP computed placement
```

El ejemplo establece un superíndice con un desplazamiento vertical igual al 40 % del tamaño actual, y la reducción de fuente actual en un 60 %.

```
[ .4.6 ] SETTXS
```

### Modos

Este comando es aplicable en todos los modos.

**Comandos relacionados**

**INDEXSST, SETFTSW**

## SETUNIT

**SETUNIT** establece las unidades de medida de todos los comandos posteriores excepto **SETGEP**.

### Sintaxis

```
unit SETUNIT
```

### Donde:

**unit**

Puede ser uno de estos valores:

**DOT3**: El valor predeterminado es 1/300 pulg.

**PELS**: 1/240 pulg.

**POINT**: 1/72 pulg.

**CM**: (centímetros)

**MM**: Milímetros

**INCH**: pulg.

**PELS** puede facilitar la conversión de recursos de AFP a recursos VIPP®.



Importante: **SETUNIT** no es un comando de arranque de VIPP®. Cuando el trabajo empieza con una instrucción **SETUNIT**, coloque un comando **XGF** antes de la instrucción **SETUNIT**.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETGUNIT**



## SETUTAB

El comando **SETUTAB** activa el procesamiento de la tabulación y define la longitud de esta en unidades actuales.

### Sintaxis

```
unitcount SETUTAB
```

### Donde:

**unitcount** es el número de unidades que se utilizan para la colocación de tabulaciones.

### Ejemplos

Este ejemplo define un punto de tabulación cada 50 unidades actuales.

```
50 SETUTAB
```



Nota: Use SETUTAB solamente cuando sea necesario, porque puede afectar al rendimiento.



Nota: Utilice **SETUTAB** con fuentes proporcionales.

### Modos

Este comando es aplicable en el modo de línea básica (sin RPE).

### Comandos relacionados

SETTAB, SETTABS

## SETV2HCONV

**SETV2HCONV** define un juego de tablas de conversión que se utilizan cuando se configura la opción apropiada de **SHP** or **SHMF** (+4000). Este comando se utiliza con fuentes de varios bytes.

### Sintaxis

```
[ /v2h_cvname1 [ (char1) (conv1) ... ] /v2h_cvname2 [ (char1) (conv1) ... ] ..... ] SETV2HCONV
```

### Donde:

**/v2h\_cvname1** es un nombre no restringido que puede utilizarse como cuarto parámetro en una entrada del comando **SETCJKENCMAP**.

**(charX) (convX)** define los pares de conversión. Si se encuentra charX en el argumento de la cadena de **SHP/SHMF**, se sustituye por convX.

### Ejemplos

```
[ /SJIS [ (1) <88EA> (2) <93F1> (3) <8E4F> (-) <815C> ] ] SETV2HCONV
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **SETFONT**
- **INDEXFONT**
- **SHP** y **SHp**
- **SHMF**, **SHMf** y **SHmf**

## SETV2HTABLE

**SETV2HTABLE** define una lista de caracteres que se procesarán horizontalmente si una fuente vertical está activa y se configuran las opciones adecuadas de **SHP** o **SHMF** (+1000/+2000).

### Sintaxis

```
[ def_scale (char1) [(char2) sca2] ... ] SETV2HTABLE
```

### Donde:

<b>def_scale</b>	es un factor de escala predefinida que se aplicará a la anchura horizontal de cada carácter procesado horizontalmente con el método 0.
<b>charX</b>	es un carácter que se procesará horizontalmente.
<b>scaX</b>	es un factor de ajuste a escala individual opcional que se sustituirá según el factor prefijado solo para ese carácter.

### Ejemplos

```
[ 1.0 [(0) 1.2] (1) (2) (3) (4) (5) (6) (7) (8) (9) ] SETV2HTABLE
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SHP**, **SHMF**, **SHMf** y **SHmf**

## SETVAR

El comando **SETVAR** establece una variable. Si está definida, la referencia de una determinada variable se usa en un comando en lugar de un operando de matriz, número o cadena.

### Sintaxis

```
/VARname value SETVAR /VARname value /INI SETVAR /^name value SETVAR value /VARname /SW
```

### Donde:

<b>VARname</b>	puede ser cualquier cadena alfanumérica ASCII que empiece por el prefijo VAR. La longitud máxima es de 20 caracteres. El guión - y el guión bajo _ se pueden utilizar. Sin embargo, el punto . solo está disponible si la variable no se utiliza con <b>VSUB</b> .
<b>valor</b>	es una cadena, un número o una matriz.
<b>/INI</b>	Si se utiliza esta opción, la variable se configurará solo si todavía no existe. No se inicializarán las variables que ya existen si se utiliza la opción /INI. Utilice esta opción en un DBM para inicializar una variable solo durante la primera llamada.
<b>^nombre</b>	es un nombre de variable que suele asociarse con una ficha de trabajo XML.
<b>/SWP</b>	/SWP es un parámetro que permite obtener el valor de la pila. Tiene como objetivo ser usado en un procedimiento de alineación <b>FROMLINE/RPEKEY</b> para capturar y usar la cadena RPE en la pila, como en este ejemplo:

```
... [ { /VARqrc /SWP SETVAR [ /AC VARqrc ] QRCODE } rot Xi Xd Yi Yd pos ]
```

En este ejemplo de una entrada RPE, **SETVAR** captura la cadena de datos extraída por pos length, y usa la variable para producir un símbolo **QRCODE**.

Las cadenas numéricas admiten números grandes de hasta 40 dígitos, 25 dígitos para la parte entera y 15 dígitos para la parte decimal. En una cadena numérica, el signo negativo y el delimitador decimal se definen mediante los parámetros /DecimalPoint y /NSign y se pueden producir en cualquier lugar de la cadena.

Es obligatorio configurar estos parámetros con los valores correspondientes para garantizar unos resultados precisos. Los valores predeterminados se definen en el archivo /usr/xgf/src/xgf.def. Si la variable es procesada por **ADD**, **SUB**, **MUL** o **DIV**, los caracteres en la cadena numérica aparte de estos dos, más los dígitos 0-9, se descartan.

La longitud inicial de la cadena definida por **SETVAR** se amplía automáticamente hasta 40 dígitos cuando es necesario.

Los números reales y enteros se utilizan solo para los valores pequeños <= 99999 por ejemplo, para la implementación de un contador. El delimitador decimal, si lo hubiera, siempre es el punto ( . ). El signo negativo, si lo hubiera, siempre es el signo menos ( - ) y es el primer carácter.

Los números reales y enteros se pueden convertir en cadenas mediante el comando **VSUB**. A continuación, se pueden formatear para la impresión mediante el comando **FORMAT**.

En un trabajo XML, se asigna un nombre completo a las variables XML de forma automática en función del nombre y la ruta XML actual. Además se restauran automáticamente cuando se cruzan los nodos primarios.

Las variables XML empiezan por ^ y han sido diseñados para su uso en una ficha de trabajo XML (.xjt). No requieren una inicialización explícita. Si no se inicializan, se configurarán de manera predefinida como una cadena vacía o con valor cero cuando parezcan haberse actualizado mediante ++, –, ADD o SUB.

Si la ruta XML actual comienza con: ^invoices^invoice^customer, a la variable: ^customer^area se le asignará un nombre completo de variable:

^invoices^invoice^customer^area y se restablecerá de forma automática si se cruza una nueva etiqueta <cliente> o <factura>.

Como las variables de árbol XML, las variables XML se pueden acceder mediante su nombre completo o mediante un subnombre no ambiguo cuando se realiza la referencia.

### Ejemplos

```
/VAR.date (December, 12th. 1993) SETVAR
/VARX1 200 SETVAR
/VARY1 300 SETVAR
VARX1 VARY1 MOVETO
VAR.date SHL
($$VARX1.) VSUB SHL
($$VARX1.) VSUB (:#####:) FORMAT SHL
```

En este ejemplo, **SETVAR** inicializa una variable en un DBM. Dado que /INI se usó para definir la variable, no se reinicializará para cada registro, lo que permite a **/VAR.COUNT ++** incrementar el contador. Si no se utiliza /INI, el recuento se configura como cero para cada nuevo registro.

```
/VAR.COUNT 0 /INI SETVAR
IF VAR.COUNT 50 eq
  { (slipsheet) SETMEDIAT
    0 NL PAGEBRK
    /VAR.COUNT 0 SETVAR
  }
ENDIF
/VAR.COUNT ++
.....
```

En este ejemplo, **SETVAR** se utiliza para definir un procedimiento. Este procedimiento imprime un bloque de direcciones. Se ejecuta cada vez que el nombre de la variable se codifica en el código VIPP®.

Este ejemplo también muestra el uso de **SHP** para imprimir las líneas de dirección Addr1 y Addr2. Cuando una de las líneas de dirección está vacía, VI Compose no moverá la posición de la impresión hacia abajo. Esto permite que el bloque de dirección se ajuste a las líneas de vacías.

```

/VARadd_block
{
  ($$Fname. $$Lname.) VSUB SHL
  Addr1 0 SHP
  Addr2 0 SHP
  ($$City. $$State. $$Zip.) VSUB SHL
} SETVAR

% when needed
300 3000 MOVETO
VARadd_block

```

Permite que el lenguaje VIPP® aumente y disminuya las variables. Existen diferencias en la aplicación a enteros y valores de cadena.

Enteros:

```

/VARNumber 0 SETVAR      % Value equals 0
/VARNumber ++           % Value now equals 1
/VARNumber 3 ADD        % Value now equals 4
/VARNumber --           % Value now equals 3

```

Valores de cadena:

- Si quiere que el número se llene con ceros, declare un número con el número de ceros adecuado.
- Si no desea números rellenos con ceros, utilice FORMAT para dar formato al número.

```

/VARNumber (000) SETVAR      % value is 000
/VARNumber ++               % Value is now 001
/VARNumber 7 ADD            % Value is now 008
/VARNumber --               % Value is now 007
/VARNumber -2 ADD           % Value is now 005
/VARNumber (7) ADD          % Value is now 012

```

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

SETMPATH, STOREVAR, VSUB

## SETVFU

El comando **SETVFU** define una tabla de unidad de formato vertical (VFU), con omisión de canales a asignaciones de número de línea para la definición de PCC actual.

### Sintaxis

```
[ /skip-key1 line-number1 /skip-key2 line-number2 ... ] SETVFU [ ... /skip-keyx [line-numberx ... ]
```

### Donde:

- |                    |   |
|--------------------|---|
| <b>/skip-key</b>   | son las claves de omisión como se define en la definición de PCC. Para obtener más información, consulte Listas estándar, tablas, claves y atributos en la <i>Guía del usuario de FreeFlow VI Compose</i> .   |
| <b>line-number</b> | los números de línea relevantes para el trabajo actual. Un número negativo de línea implementa un salto en la línea especificada en una nueva página. Se asocian varios números de línea con una sola clave de omisión. Las líneas se omitirán en orden, a partir de la primera que sea más grande o igual que la línea actual. |

Consulte **BEGINPCC** para obtener más información sobre las definiciones de PCC.

**SETVFU** debe programarse en un JDT.

### Ejemplos

```
[ /SK1 -1 /SK2 35 /SK9 66 ] SETVFU
```

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

**BEGINPCC**, **ENDPCC**, **SETPCC**

## SETZEBRA

El comando **SETZEBRA** permite la impresión de cebra, o franjas verdes, en todas las páginas siguientes. La impresión cebra es la impresión de zonas sombreadas de fondo.

### Sintaxis

```
Colorkey lines-with lines-without SETZEBRA
Colorkey lines-with lines-without /V SETZEBRA
```

### Donde:

<b>Colorkey</b>	define el color. WHITE indica que no se realiza la impresión de cebra.
<b>lines-with y lines-without</b>	define la secuencia de líneas con o sin la impresión en cebra.
<b>/V</b>	varía el número de líneas de cebra según el número de líneas impresas en la página. Sin /V, el número de líneas de cebra es regular y es controlado por el número de líneas por página definidas a través del comando <b>SETGRID</b> .

Los cuadros de cebra se calculan desde la cuadrícula y los márgenes actuales. El patrón de cebra o de franjas verdes predeterminado se configura en el archivo `/usr/xgf/src/xgf.def`. Las Colorkeys se definen en el archivo `/usr/xgf/src/xgf.gep`, en Listas estándar, tablas, claves y atributos en la *Guía del usuario de FreeFlow VI Compose*. Las claves predefinidas incluyen:

- NEGRO
- XLIGHT
- LMEDIUM
- OSCURO
- ROJO
- BLANCO
- CLARO
- MEDIO
- XDARK
- AZUL
- DMEDIUM
- VERDE

Edite el archivo `/usr/xgf/src/xgf.gep` para agregar nuevas Colorkeys

### Ejemplos

```
LIGHT 3 3 SETZEBRA
```

### Modos

Este comando es aplicable en todos los modos.



#### Comandos relacionados

- SETGRID
- SETMARGIN
- SETPAT
- SETTXC

## Comandos SHx

El término genérico SHx se utiliza en esta documentación y en otra documentación de Variable Information Suite para indicar que uno o más comandos de los indicados aquí se pueden usar:

- SHC y SHc
- SHJ y SHj
- SHL y SH
- SHMF, SHMf y SHmf
- SHP y SHp
- SHR y SHr
- SHT y SHt
- SHX

## SHC y SHc

**SHC** y **SHc** imprimen datos centrados en la posición de impresión principal (**SHC**) o secundaria (**SHc**). **SHC** restaura la posición de impresión horizontal principal al último valor especificado por **MOVETO**, y envía la posición de impresión vertical según el valor **SETLSP**. **SHc** establece la posición de impresión secundaria en el punto al que se llegó después de imprimir los datos.

### Sintaxis

(printable data) SHC

(printable data) SHc

### Modos

Estos comandos son aplicables en modo nativo, el modo de línea y el modo de base de datos.

### Comandos relacionados

- **MOVETO**
- **MOVEH**
- **MOVEHR**
- **SETLSP**
- **SHJ** y **SHj**
- **SHL** y **SH**
- **SHMF**, **SHMf** y **SHmf**
- **SHR** y **SHr**

## DESPLAZAMIENTO

El comando **SHIFT** define los valores de desplazamiento horizontal (X) y vertical (Y) en unidades actuales de las páginas de portada y contraportada. Todo el contenido de la página se desplaza del origen inferior izquierdo en el eje vertical y horizontal según los valores especificados (positivos o negativos). Estos valores siempre hacen referencia al origen inferior izquierdo de la hoja vertical independientemente de la orientación actual.

### Sintaxis

```
Xfront Yfront Xback Yback SHIFT
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

SETPARAMS, SETVAR, SHX

## SHIFTDATE

**SHIFTDATE** establece las variables de la fecha actual según la fecha actual aumentada por el valor especificado. Por lo tanto, con un valor de cero, **SHIFTDATE** equivale a **GETDATE**. La fecha actual se restaura mediante un nuevo **GETDATE**, o bien el siguiente **PAGEBRK**. Consulte **GETDATE** para obtener más información.

### Sintaxis

```
shift /unit SHIFTDATE
```

### Donde:

<b>desplazar</b>	es un número entero que indica el valor en unidades especificadas por /unit
<b>/unit</b>	es una de estos valores:
	<b>/S</b> segundos
	<b>/M</b> minutos
	<b>/H</b> horas
	<b>/D</b> días
	<b>/W</b> semanas

Si se utiliza **SHIFTDATE**:

- Los valores negativos son aceptables.
- Varios comandos **SHIFTDATE** pueden colocarse en una sola página, calculado desde la fecha actual.
- **SHIFTDATE** no es compatible con años bisiestos.
- Utilice **PAGEBRK** o **GETDATE** para restaurar las variables de fecha según la fecha actual del sistema.

### Ejemplos

```
15 /D SHIFTDATE          % set current date plus 15 days
($$DD_MO./$$D_DD./$$D_YYYY.) VSUB SH  % print date plus 15 days
```

Se imprimirá el siguiente ejemplo: This will expire in 30 days, on 07/28/2005.

```
30 /D SHIFTDATE
(This will expire in 30 days, on $$D_MOL./$$D_DD./$$D_YYYY.) VSUB 0 SHP
PAGEBRK
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**GETDATE, SPOOLNAME, SETPARAMS**

## SHJ y SHj

**SHJ** y **SHj** imprimen datos justificados desde la posición de impresión principal (**SHJ**) o secundaria (**SHj**) según la anchura especificada por el último **SETCOLWIDTH**.

La justificación entre palabras se aplica si la cadena tiene espacios. De lo contrario, se aplica justificación entre caracteres.

**SHJ** restaura la posición de impresión horizontal principal al último valor especificado por **MOVETO** y envía la posición de impresión vertical según el valor **SETLSP**.

**SHj** establece la posición de impresión secundaria en el punto al que se llegó después de imprimir los datos.

### Sintaxis

```
(printable data) SHJ (printable data) SHj
```



Nota: Si la anchura configurada por el comando **SETCOLWIDTH** no se ajusta a la longitud de los datos imprimibles, el texto impreso puede solaparse.

### Modos

Estos comandos son aplicables en todos los modos.

### Comandos relacionados

- **MOVETO**
- **MOVEH**
- **MOVEHR**
- **SETCOLWIDTH**
- **SETLSP**
- **SHL** y **SH**
- **SHC** y **SHc**
- **SHMF**, **SHMf** y **SHmf**
- **SHR** y **SHr**

## SHL y SH

**SHL** y **SH** imprimen datos con alineación a la izquierda en la posición de impresión principal (**SHL**) o secundaria (**SH**).

**SHL** restaura la posición de impresión horizontal principal al último valor especificado por **MOVETO** y envía la posición de impresión vertical según el valor **SETLSP**.

**SH** establece la posición de impresión secundaria en el punto al que se llegó después de imprimir los datos.

### Sintaxis

```
(printable data) SHL
(printable data) SH
```

### Ejemplos

Este ejemplo imprime: Hello World it's me.

```
(Hello World) SH ( it's me) SH
```

### Modos

Estos comandos son aplicables en todos los modos.

### Comandos relacionados

- **CASETI**
- **GETINTV**
- **MOVETO**
- **MOVEH**
- **MOVEHR**
- **SETLSP**
- **SHC y SHc**
- **SHJ y SHj**
- **SHMF, SHMf y SHmf**
- **SHR y SHr**

## SHMF, SHMf y SHmf

**SHMF**, **SHMf** y **SHmf** imprimen los datos en la posición de impresión principal (**SHMF/SHMf**) o secundaria (**SHmf**). Utilice estos comandos para seleccionar cualquiera de las fuentes y los colores de los datos de impresión mediante el uso de una secuencia de cambio de la fuente/color. El prefijo de cambio de fuente/color se define en el archivo de `/usr/xgf/src/xgf.def` con **SETFTSW**. // es el valor prefijado.

**SHMF** restaura la posición de impresión horizontal principal al último valor especificado por **MOVETO**, y envía la posición de impresión vertical según el valor **SETLSP**.

**SHmf** establece la posición de impresión secundaria en el punto al que se llegó después de imprimir los datos.

**SHMF** permite imprimir los flujos de datos del modo de línea con secuencias de cambio de fuentes incorporadas. Esto puede ser útil para migrar flujos de datos **XES** a VIPP®.

**SHmf** funciona como **SHMF**. Sin embargo, como el comando **SHP**, no reenvía la posición vertical de impresión cuando los datos imprimibles son una cadena vacía.

### Sintaxis

```
(printable data) align SHMF (printable data) colwidth align SHMF
```

### Donde:

<b>alinear</b>	es la opción de alineación y se puede especificar como uno de estos valores; el signo más (+) indica que el valor se puede combinar con otros valores como corresponda, no es parte de la sintaxis:
<b>0</b>	izquierda
<b>1</b>	derecha
<b>2</b>	centro
<b>3</b>	justificar
<b>4</b>	alineación de punto decimal (el separador decimal es el carácter actualmente definido por el parámetro /DecimalPoint)
<b>+100</b>	ajustar a la anchura
<b>+200</b>	estirar en la anchura





Nota:

- **ajustar a la anchura:** Si se activa esta opción, se ajusta la escala de la fuente actual en el eje horizontal para que las cadenas impresas se ajusten a la anchura de la columna actual, según lo especificado por el comando **SETCOLWIDTH** o por el operando **colwidth**. El ajuste a escala solo se realiza si la anchura natural supera la anchura de la columna. Si la anchura natural no supera la anchura de la columna, no se aplica el ajuste a escala.
- **estirar en la anchura:** Si se activa esta opción, se ajusta la escala de la fuente actual en el eje horizontal para que las cadenas impresas se ajusten a la anchura de la columna actual, según lo especificado por el comando **SETCOLWIDTH** o por el operando **colwidth**.

El ejemplo siguiente imprime el nombre en un cuadro de dirección con alineación a la izquierda y con un límite de anchura de 500 puntos:

```
Mrs. Mary-Eleonore de Bourbon l'Archambaud 500 100 SHMF
```

Para textos de impresión especial con brillo y correlación, para controlar la anchura del efecto SI, utilice la opción de **estiramiento en anchura**. El texto se ajusta a la anchura definida de la columna. Para obtener más información, consulte [Fuentes de correlación y GlossMark \(GL/CR\)](#) en el tema *Sugerencias de programación*.

- **+1000:** Configurar el procesamiento vertical a horizontal, método 0, horizontal en vertical
- **+2000:** Configurar el procesamiento vertical a horizontal, método 1, giro 90 grados en sentido horario
- **+4000:** Establecer conversión
- **+5000:** Configurar la conversión vertical a horizontal, método 0, horizontal en vertical
- **+6000:** Configurar la conversión vertical a horizontal, método 1, giro 90 grados en sentido horario
- **+10000:** Aplicar transformación bidireccional en cada línea
- **+20000:** Aplicar la transformación bidireccional en cada línea y cambie la fuente de los caracteres de izquierda a derecha
- **+30000:** Aplicar la transformación bidireccional en cada línea y sustituir dígitos europeos por hindi



Nota: La lista de caracteres que se cambiará de vertical a horizontal se configura anteriormente a través del comando **SETV2HTABLE**. La tabla de conversión de la codificación relacionada se configura anteriormente a través del comando **SETV2HCONV**. Las listas y las tablas predeterminadas se definen en el archivo de configuración: `xgf/src/cjk.def`.

#### **colwidth**

es la anchura de la columna para la justificación (alineación = 3). Cuando no se especifica **colwidth**, se utiliza el valor definido por un **SETCOLWIDTH** anterior. Cuando se especifica **colwidth**, se sustituye y reemplaza el valor definido por un **SETCOLWIDTH** anterior.

### Ejemplos

Este ejemplo imprime Texto con fuente 1 cambie a fuente 2 azul y fuente 3.

```
/1 /NHE 18 INDEXFONT  
/2 /NHE 24 INDEXFONT  
/3 /NHEO 18 INDEXFONT  
/A BLACK INDEXCOLOR  
/B BLUE INDEXCOLOR  
200 200 MOVETO  
(//1Text using font 1 //2//BSwitch to font 2 blue //3and Font 3) 0 SHMF
```



Nota: Para obtener más información, consulte **INDEXFONT** y **SETFTSW**.

## CONSEJO

Use el comando **SHx** correspondiente si el cambio de color o fuente no es necesario.

## Modos

Estos comandos son aplicables en todos los modos.

## Comandos relacionados

- **INDEXBAT**
- **INDEXFONT**
- **MOVETO**
- **MOVEH**
- **MOVEHR**
- **SHP y SHp**
- **SETCOLWIDTH**
- **SETFTSW**
- **SETLSP**
- **SHL y SH**
- **SHC y SHc**
- **SHJ y SHj**
- **SHR y SHr**

## SHP y SHp

El comando **SHP** imprime datos en la posición de impresión principal, y **SHp** imprime datos en la posición de impresión secundaria con ajuste de línea.

Estos comandos funcionan igual que **SHMF**. Además, realizan el ajuste de palabras en espacios y guiones en datos imprimibles según la anchura definida por **SETCOLWIDTH** y el espaciado definido por **SETLSP**. Los datos imprimibles pueden ser una cadena larga de hasta 65 535 caracteres con formato dinámico en un párrafo de **SHP**.

**SHP** restaura la posición de impresión horizontal principal al último valor especificado por **MOVETO**, y envía la posición de impresión vertical alcanzada después de la última línea según el valor **SETLSP**.

### Sintaxis

```
(printable data) align SHP
(printable data) colwidth align SHP
[ (printable data1) (printable data2) .... ] align SHP
[ (printable data1) (printable data2) .... ] colwidth align SHP
(printable data) [width height spacing] align SHP
[ count (printable data)) ] align SHP
```

Donde:

**alinear**

es la opción de alineación y se puede especificar como uno de estos valores; el signo más (+) indica que el valor se puede combinar con otros valores como corresponda, no es parte de la sintaxis:

**0** izquierda

**1** derecha

**2** centro

**3** justificar con la última línea alineada a la izquierda (volver a habilitar)

**4** justificar con la última línea alineada a la derecha

**5** justificar con la última línea centrada

**6** justificar todas las líneas

**+00** tratar caracteres de nueva línea (0×0A) como espacios

**+10** eliminar espacios en blanco duplicados entre palabras solo al final de las líneas

**+20** tratar caracteres de nueva línea (0×0A) como final de línea

**+30** Eliminar espacios en blanco duplicados entre las palabras y tratar los caracteres de nueva línea (0×0A) como final de línea

**+40** no ajustar en el carácter de guión

**+000** ajuste de las palabras latinas y entre caracteres asiáticos

**+100** ajuste según las reglas de Asia

**+200** ajustar según las reglas de Asia

**+300** ajuste + comillas voladas las reglas de Asia

**+400** comillas voladas según las reglas de Asia

**+600** aplicar transformación bidireccional en cada línea

**+700** aplicar la transformación bidireccional en cada línea y cambie la fuente de los caracteres de izquierda a derecha

**+800** aplicar la transformación bidireccional en cada línea y sustituir dígitos europeos por hindi

**+1000** configurar el procesamiento vertical a horizontal, método 0, horizontal en vertical

**+2000** configurar el procesamiento vertical a horizontal, método 1, giro 90 grados en sentido horario

**+4000** configurar la conversión

**+5000** configurar la conversión vertical a horizontal, método 0, horizontal en vertical

**+6000** configurar la conversión vertical a horizontal, método 1, giro 90 grados en

sentido horario



Nota: Las reglas de idiomas y formato de Asia para la codificación relacionada deben configurarse anteriormente a través de los comandos **SETCJKENCMAP** y **SETCJKRULES**.

La lista de caracteres que se cambiará de vertical a horizontal se configura anteriormente a través del comando **SETV2HTABLE**. La tabla de conversión de la codificación relacionada se configura anteriormente a través del comando **SETV2HCONV**.

Las listas y las tablas predeterminadas se definen en el archivo de configuración `xgf/src/cjk.def`.

Para realizar la alineación vertical del párrafo en la posición de impresión actual, use las siguientes opciones:

- +0 alineación superior; predeterminada
- +10000 alineación inferior
- +20000 alineación centro

Para ajustar o adecuar el texto en un objeto o área, use las opciones siguientes:

- +100000 ajustar la anchura de la línea
- +200000 estirar la anchura de la línea

Las dos opciones precedentes equivalen a las opciones **SHMF** +100 y +200, y deben ser usadas en combinación con la opción +20. Cada línea que está delimitada por los caracteres 0×0A de nueva línea se comprime en función de la opción en lugar de ser ajustada.

- +300000 ajustar la anchura de la línea, conservar el ajuste
- +400000 estirar la anchura de la línea, conservar el ajuste

Las dos opciones anteriores son similares a las opciones +100000 y +200000, con la salvedad de que el ajuste se aplica antes de la compresión.

Para textos de impresión especial con brillo y correlación, para controlar la anchura del efecto SI, utilice la opción de estiramiento en anchura. Para controlar el relleno vertical, use la sintaxis de texto en cuadro con `spacing=0`. El texto se ajusta a la anchura y altura definidas del cuadro. Para obtener más información, en el tema Sugerencias de programación, consulte [Fuentes GlossMark y de correlación \(GL y CR\)](#)

<b>colwidth</b>	es la anchura de columna utilizada para el ajuste de palabras. Cuando no se especifica <b>colwidth</b> , se utiliza el valor definido por un <b>SETCOLWIDTH</b> anterior. Cuando se especifica <b>colwidth</b> , se sustituye y reemplaza el valor definido por un <b>SETCOLWIDTH</b> anterior.
<b>anchura</b>	es la restricción horizontal utilizada como <b>colwidth</b> .
<b>altura</b>	es la restricción vertical adicional.

<b>espaciado</b>	es el factor de espaciado de línea en relación al tamaño de la fuente; ejemplos:  <b>0</b> justificación vertical Esta opción hace que el espaciado de línea se ajuste para que el texto rellene completamente el cuadro.  <b>1</b> una sola línea  <b>1.5</b> una y media  <b>2</b> espaciado doble  <b>.5</b> media línea
<b>recuento de</b>	es un número entero que se utiliza para indicar el número de veces que los datos imprimibles se duplican.

En el tercer y cuarto ejemplos de sintaxis, **SHP** une la lista especificada de cadenas y formatos, y da formato a la cadena resultante, la suma de las longitudes no puede superar los 65 535 caracteres.

La sintaxis printable data [width height spacing] align **SHP** ajusta un bloque de texto en un cuadro predefinido mediante el ajuste automático del tamaño de la fuente.

Al usar la alineación 3, cualquier palabra que hace que la línea supere la anchura de la columna, pero en un determinado porcentaje, se imprime con la línea. El umbral se configura de manera predefinida en un 0,05 % de la anchura de la columna. Este valor se almacena en una variable denominada *VAR. SHP\_OVER*. Utilice el comando **SETVAR** para cambiar este valor.



Nota: Si el área de datos imprimibles está vacía, la posición de impresión vertical no se envía. Para que el texto fluya entre las tramas o las páginas, combine un comando **SHP** con un comando **SETLKF**.

## Ejemplos

Este ejemplo ilustra el uso de los registros de base de datos para imprimir un bloque de dirección y nombre que puede contener campos en blanco. Estos son los registros de base de datos de ejemplo:

Nombre,Apellido,dir1,dir2,ciudad,provincia,códigopostal

David,Kirk,101 Continental Blvd., Suite 1, El Segundo, CA, 90245

Mary, Smith, 5 Euclid Lane, Santa Monica, CA 90403

El ejemplo anterior imprime esta información:

David Kirk

101 Continental Blvd.

Suite 1

El Segundo, CA 90245

Mary Smith

5 Euclid Lane

Santa Monica, CA 90403

El ejemplo siguiente utiliza parámetros de alineación adicionales:

```
DOT3 SETUNIT 200 200 MOVETO /VARtextexample ( The lazy brown dog jumped over the sleepi
```

Insertar esta sintaxis en el ejemplo.

```
VARtextexample 300 0 SHP
```

Crea un párrafo de una pulgada de anchura similar a este:

```
The lazy brown dog jumped over the sleeping fox
```

Insertar esta sintaxis en el ejemplo...

```
VARtextexample 300 10 SHP
```

Crea un párrafo de una pulgada de anchura similar a este:

```
The lazy brown dog jumped over the sleeping fox
```

Este ejemplo produce el nombre y la dirección como se indica a continuación:

```
(John R. Doe\n1405 Ocean Drive\nEl Segundo, CA 90245) 20 SHP
```

John R. Doe

1405 Ocean Drive

El Segundo, CA 90245

Este ejemplo repite "Hello World" 50 veces, ajustado en un párrafo:

```
[ 50 (Hello World) ] 0 SHP
```

Estos ejemplos se pueden combinar con otras opciones de alineación:

```
(... text block ...) 20002 SHP % align vertically and horizontally (... text block ...)
```

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

- **MOVETO**
- **MOVETO**
- **MOVEH**
- **SCALL**
- **SETCOLWIDTH**
- **SETLSP**
- **SETVAR**
- **SHMF, SHMf y SHmf**
- **VSUB**

- INDEXALIGN
- INDEXLSP



## SHPATH

**SHPATH** imprime una cadena de texto en una ruta definida anteriormente por comandos mediante **TPATH** **GEPkey**. Los atributos de texto actuales, fuente, color, fondo, etc. se utilizan para imprimir el texto. Los parámetros de atributo están permitidos en la cadena de texto.

### Sintaxis

```
(string) baseline_offset text_start align SHPATH (string) [baseline_offset path_offset]
```

### Donde:

<b>(cadena)</b>	es la cadena de texto que se imprimirá
<b>baseline_offset</b>	es el desplazamiento en unidades actuales, positivas o negativas, entre la línea de base del texto y la ruta
<b>path_offset</b>	se parece a baseline_offset pero desplaza la ruta en lugar de los caracteres.
<b>text_start</b>	es el desplazamiento en unidades actuales desde el principio de la ruta hasta el punto en el que se inicia el texto. Si es nulo, el texto se inicia donde terminó el texto anterior.
<b>text_end</b>	es el desplazamiento en unidades actuales desde el principio de la ruta hasta el punto en el que el texto finaliza.  Si text_end se omite:  Si se configura la opción de alineación +40 (normalmente para una ruta cerrada) la totalidad de la ruta está disponible para texto, y text_start solo define el punto de inicio.  Si la opción de alineación +40 no se configura (normalmente para rutas abiertas) la ruta está disponible desde text_start al final de la ruta.
<b>alinear</b>	es uno de estos códigos de alineación:  0 alinear en el inicio de la ruta  1 alinear al final de la ruta  2 centrar en la ruta  3 justificar texto en la ruta  +20 dirección de ruta al revés  +40 permite al texto continuar al comienzo de la ruta cuando supera el final de esta

### Ejemplos

```
(text to print on a path) 0 0 0 SHPATH
```

### Modos

Este comando es aplicable en todos los modos.

**Comandos relacionados**

- DRAWB y DRAWBR
- DRAWC
- DRAWPOL
- DRAWPATH y DRAWPATHR

# SHPIT

Utilice **SHPIT** para imprimir texto distorsionado en un cuadrilátero. El texto distorsionado se rellenará o tomará la forma del cuadrilátero. El texto puede o no tener un efecto de sombra paralela aplicado. También se puede dibujar el cuadrilátero mediante la codificación de un **DRAWPOL** adicional. Este comando se puede usar para la personalización de imágenes mediante la impresión del texto en la parte superior de una imagen como, por ejemplo, texto en un muro o en la arena. Para un mejor efecto, sin bordes, puede reemplazar los colores de las sombras y del texto con patrones de imagen con un pequeño patrón extraído de la imagen, consulte **SETPAT** para determinar cómo crear un patrón de imagen. La imagen pequeña se puede crear mediante cualquier herramienta de impresión comercial.

## Sintaxis

### Sin efecto de sombra:

```
(variable string) VSUB [ llx lly ulx uly urx ury lrx lry ] [text_color] XYZ SHPIT
```

### Con efecto de sombra:

```
(variable string) VSUB [ llx lly ulx uly urx ury lrx lry ] [text_color shadow_color Xof
```

### Donde:

llx y lly	son el punto inferior izquierdo del cuadrilátero
ulx y uly	son el punto superior izquierdo del cuadrilátero
urx y ury	son el punto superior derecho del cuadrilátero
lrx y lry	son el punto inferior derecho del cuadrilátero
X	es una de las siguientes: <b>0</b> sin efecto espejo ni giro <b>1</b> girar 90 grados <b>2</b> girar 180 grados <b>3</b> girar 270 grados <b>4</b> efecto espejo <b>5</b> efecto espejo y giro de 90 grados <b>6</b> efecto espejo y giro de 180 grados <b>7</b> efecto espejo y giro de 270 grados
Y	es una de las siguientes: <b>0</b> sin ajuste <b>1</b> ajustar a la anchura <b>2</b> estirar en la anchura

**Z**

es una de las siguientes:

**0** alinear izquierda

**1** alinear derecha

**2** alinear centro

**Ejemplos XYZ:**

200 = girar 180 grados, sin ajuste ni alineación izquierda

401 = espejo, sin ajuste y alineación a la derecha

622 = espejo, girar 180 grados, estirar en anchura y alinear al centro

**color del texto**

puede ser un color normal, una trama o **GEPKey**

**color de sombra**

puede ser un color normal, una trama o **GEPKey**

**Xoff**

desplazamiento, efecto sombra horizontal en unidades actuales (pueden ser negativos).

**Yoff**

desplazamiento, efecto sombra vertical en unidades actuales (pueden ser negativos).

**Ejemplos**

```
([=FirstName=] [=LastName=] ) VSUB [ 78.0 346.8 78.0 182.966 384.0 43.2 384.0 103.2 ]
```

**Modos**

Este comando es aplicable en todos los modos.

**Comandos relacionados**

**SETPAT,DRAWPOL**

## SHR y SHr

**SHR** y **SHr** imprimen datos con alineación a la derecha en la posición de impresión principal (**SHR**) o secundaria (**SHr**).

**SHR** restaura la posición de impresión horizontal principal al último valor especificado por **MOVETO**, y envía la posición de impresión vertical según el valor **SETLSP**.

**SHr** establece la posición de impresión secundaria en el punto al que se llegó después de imprimir los datos.

### Sintaxis

(printable data) SHR

(printable data) SHr

### Modos

Estos comandos son aplicables en todos los modos.

### Comandos relacionados

- **MOVETO**
- **MOVEH**
- **MOVEHR**
- **SETLSP**
- **SHL y SH**
- **SHC y SHc**
- **SHJ y SHj**
- **SHMF, SHMf y SHmf**

## SHROW

Utilice **SHROW** para imprimir una fila de una tabla.

### Sintaxis

```
[ [ /param1 value1 ... /paramX valueX ]      % cell 1
  ...
  [ /param1 value1 ... /paramX valueX ]      % cell N
] SHROW

[ [ /param1 value1 ... /paramX valueX ]      % (without /width): default params applicable to all cells
  ...
  [ /param1 value1 ... /paramX valueX ]      % cell 1
  ...
  [ /param1 value1 ... /paramX valueX ]      % cell N
] SHROW
```

### Donde:

**/param** puede ser uno de los siguientes:

**/Width** anchura de las celdas en unidades actuales

**/Height** altura mínima de la celda en unidades actuales (opcional)

**/MaxHeight** altura máxima de la celda en unidades actuales (opcional)

**/Margins** márgenes de celda en unidades actuales

**/CellFill** color para rellenar la celda. `null` es el valor predeterminado es “null” e indica que la celda no se llena con ningún color.

**/CellStroke** GEPkeys para trazar los bordes de la celda

**/TextAtt** Código VIPP® para configurar los atributos de texto

**/Align** atributo de alineación (igual que SHP)

**/CellText** texto que se debe colocar en la celda

**/CellImage** imagen que se debe colocar en la celda

**/FixHeight** la altura de la celda es fija, y se ajusta la escala del texto en la celda, lo que se recomienda para mejorar el rendimiento

Atributo de alineación **/IAlign** para **CellImage**, igual que **ICALL**. Si no está presente, la imagen se alinea según **/Align** pero queda restringida a los valores de alineación de **ICALL**.



Nota: Los comandos **SETLKF**, **FOREACH** y **SHROW** pueden interactuar entre sí se realiza una llamada a **SHROW** en el bucle **FOREACH**. Este comportamiento se activa a través de la sintaxis **FOREACH** siguiente: { sequence of VIPP commands } `variable_array` /MF **FOREACH**. Cuando se utiliza la opción /MF, se evalúa el tamaño total de todos los comandos **SHROW** invocados en el procedimiento **FOREACH**. Si el tamaño total de los comandos **SHROW** no encaja en el marco, las filas no se imprimen, se realiza una llamada a **NEWFRAME** y el procedimiento se ejecuta por segunda vez con los mismos valores de entrada de la tabla. Esta opción pretende solucionar problemas de varios marcos con distintos formatos, para que en la segunda ejecución se pueda realizar una llamada diferente a **SHROW**.

**Ejemplo:**

```
[ % default cell parameters
  [ /FixHeight 18
    /Margins [ 4 4 4 4 ]
    /CellStroke [null LT null null]
    /CellFill null
    /TextAtt { F004 L003 C003 }
  ]
  % cells
  [ /width 52   /Align 000020 /CellText Date ]
  [ /width 53   /Align 000020 /CellText Time ]
  [ /width 146  /Align 000020 /CellText Place ]
  [ /width 126  /Align 000020 /CellText Number ]
  [ /width 48   /Align 000020 /CellText Rate ]
  [ /width 55   /Align 000021 /CellText Minutes ]
  [ /width 60   /Align 000021 /CellText Amount ]
] SHROW
```

**Modos**

Este comando es aplicable en todos los modos.

**Comandos relacionados**

- ENDRPE
- FROMLINE
- INDEXRPE
- RPEKEY
- STARTLM
- BEGINTABLE
- ENDTABLE

## SHT y SHt

**SHT** y **SHt** imprimen datos posiblemente truncados en la posición de impresión principal (**SHT**) o secundaria (**SHt**). La cantidad de datos impresos se limita a la longitud definida por la anchura de la columna. Estos comandos funcionan como **SHP** y **SHp** pero imprimen solo la primera línea del párrafo. Si los datos se truncan, se imprimen unos puntos suspensivos, . . . al final de la línea para indicar el truncamiento.

### Sintaxis

```
(printable data) align SHT (printable data) colwidth align SHT [ (printable data1) (pri
```

### Donde:

**alinear** es la opción de alineación y puede especificarse como uno de estos valores:

**0** izquierda

**1** derecha

**2** centro

**3** XYcenter (mayúsculas)

**+000** ajuste de las palabras latinas y entre caracteres asiáticos

**+100** ajuste según las reglas de Asia

**+200** ajustar según las reglas de Asia

**+300** ajuste + comillas voladas las reglas de Asia

**+400** comillas voladas según las reglas de Asia



Nota: Las reglas de idiomas y formato de Asia para la codificación relacionada deben configurarse anteriormente a través de los comandos **SETCJKNMAP** y **SETCJKRULES**.

**colwidth** es la anchura de columna utilizada para el ajuste de palabras. Cuando no se especifica **colwidth**, se utiliza el valor definido por un **SETCOLWIDTH** anterior. Cuando se especifica **colwidth**, se sustituye y reemplaza el valor definido por un **SETCOLWIDTH** anterior.

Con los dos últimos ejemplos de sintaxis, **SHT** une la lista especificada de cadenas y da formato a la cadena resultante; la suma de las longitudes no puede superar los 65 535 caracteres.

Al usar la alineación 3, cualquier palabra que hace que la línea supere la anchura de la columna, pero en un determinado porcentaje, se imprime con la línea. El umbral se configura de manera predefinida en un 0,05 % de la anchura de la columna. Este valor se almacena en una variable denominada **VAR.SHT\_OVER**. Utilice el comando **SETVAR** para cambiar este valor.

### Modos

Estos comandos son aplicables en todos los modos.

### Comandos relacionados



- MOVETO
- MOVEH
- MOVEHR
- SCALL
- SETCOLWIDTH
- SETLSP
- SETVAR
- SHMF, SHMf y SHmf
- VSUB

## SHX

El comando **SHX** imprime datos con un **GEPkey** y gira los datos.

**SHX** restaura la posición de impresión horizontal principal al último valor especificado por **MOVETO**, y envía la posición de impresión vertical según el valor **SETLSP**.

### Sintaxis

```
(printable data) rotate GEPkey align SHX
```

### Donde:

<b>girar</b>	es el ángulo en grados de giro del texto. Positivo es en sentido antihorario.
<b>GEPkey</b>	hace referencia a una <b>GEPkey</b> que se aplicará a cada carácter (contorno y/o relleno).
<b>alinear</b>	define el número de registros que es preciso leer para el procesamiento de la DBM. El valor predeterminado es 1. La opción de agrupación de registros hace que los registros del archivo de base de datos se procesen en paquetes, el primero de los cuales contiene los nombres de los campos. Esto es útil para registros largos con muchos campos.
	0 izquierda
	1 derecha
	2 centro
	4 XYcenter (mayúsculas)
	5 centro XY (minúsculas)
<b>XYcenter</b>	indica que el centrado se lleva a cabo no solo horizontalmente, según la longitud de la cadena, sino que también verticalmente, en función de la altura de caracteres.



Nota: **SHX** puede asociarse con un fondo; consulte **SETTXB** y **INDEXBAT** solamente si se utiliza un **GEPkey** sin trazado. Un **GEPkey** no trazado tiene LineWidth igual a 0; consulte **SETGEP**.

### CONSEJO

Este comando puede utilizarse para colocar texto trazado a través de la página con una fuente grande.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **INDEXBAT**
- **SETGEP**
- **MOVETO**
- **SETBAT**
- **SETLSP**

- SETTXB

## SKIPPAGE

**SKIPPAGE** hace que la página actual se omita, no se imprime. Este comando solo se utiliza en un procedimiento **BEGINPAGE** bajo el control de una instrucción **IF/ENDIF**.

### Sintaxis

SKIPPAGE

### Ejemplos

Este ejemplo muestra cómo hacer que las pancartas, las páginas con las líneas de asteriscos se omitan.

```
/BANNER 1 10 /eq 132 (*) SETPCD { IF BANNER {SKIPPAGE} ENDIF } BEGINPAGE
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**BEGINPAGE**, **IF/ELSE/ELIF/ENDIF**, **SETPCD**

## SLIPSHEET

**SLIPSHEET** inserta una página adicional, pensada para separar juegos, en el medio de un trabajo. Aunque está disponible para todos los modos, tiene como objetivo principal el modo de línea, porque este modo permite generar páginas adicionales además de páginas de datos. En el modo a 2 caras o en el modo varias en 1, **NEW-FRONT** se codifica antes y después de **SLIPSHEET** si se necesita la página en una hoja independiente, a menos que se codifique un cambio de materiales de impresión en el procedimiento de hojas intermedias.

### Sintaxis

```
{ slipsheet proc } SLIPSHEET
```

### Donde:

**{ slipsheet proc }** puede contener los comandos necesarios para especificar el formato de la hoja intermedia, por ejemplo, **SETMEDIA** o **SETFORM**.

El procedimiento entre llaves no afectará al resto del trabajo de impresión, por ejemplo a la numeración de páginas. Las hojas intermedias se imprimen utilizando el formulario actual de forma prefijada. Para cancelarlas o para seleccionar otras, utilice **SETFORM** en el procedimiento.

### Ejemplos

Esto es un ejemplo de codificación que selecciona papel azul y un determinado formulario.

```
/SLIP 1 5 0 10 /eq (criteria string) SETPCD
{ IF SLIP
  { { (:blue) SETMEDIA
    (slipform.frm) SETFORM
  } SLIPSHEET
  }ENDIF
} BEGINPAGE
```



Nota: En modo nativo o de base de datos, use siempre **SLIPSHEET** inmediatamente después de **PAGEBRK** o antes del principio de la primera página.

### CONSEJO

Si se usa en modo de línea, este comando debe ser llamado en **BEGINPAGE** o en un procedimiento **/P ENDPAGE** en una prueba de PCD. Use **GETFIELD** para obtener los datos de la página actual o de páginas anteriores e imprímalos en las hojas intermedias mediante un formulario en línea.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **BEGINPAGE**
- **ENDPAGE**
- **GETFIELD**
- **SETFORM**
- **SETMEDIA**

## SOF\_off

**SOF\_off** desactiva el modo de final de línea cuando se detecta %!

### Sintaxis

`SOF_off`

### Modos

○Este comando es aplicable en modo de línea y de base de datos.

### Comandos relacionados

% %EOF, %!

## SORT

**SORT** ordena una tabla en orden ascendente o descendente según una clave de campo.

### Sintaxis

`VARtable /VARkeyX /opt SORT`

### Donde:

<b>VAR_table</b>	tiene el formato:
	<code>[ [/VARkey1 /VARkey2 ..] [val11 val12 ..] .. [valN1 valN2 ..]</code>
<b>/VARkeyX</b>	es una de las claves de <code>[ /VARkey1 /VARkey2 ... ]</code>
	es una de:
	<b>/A</b> ascendente
	<b>/D</b> descendente

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **SETVAR**
- **ADD**
- **FOREACH**
- **UPDATE**

## SPOOLNAME

**SPOOLNAME** especifica un nombre y una ruta para un archivo temporal para almacenar los datos en modo cyclecopy; consulte **SETCYCLECOPY**. Utilice este comando en el archivo `/usr/xgf/src/xgfunix.run` o `x:\xgf\src\xgfdos.run`, donde se especifica el valor prefijado.

### Sintaxis

`SOF_off`



Nota: Este archivo no se utiliza en DocuPrint NPS ni en los productos del servidor de impresión FreeFlow, excepto en el modo de transmisión por secuencias.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETCYCLECOPY**, **SETMAXCOPY**



## STARTBOOKLET

Utilice el comando **STARTBOOKLET** para iniciar un folleto. Este comando está codificado antes de los comandos de marca en la primera página del folleto. Si está presente, **STARTBOOKLET** ejecuta el procedimiento **VIPPStartBooklet**, la implementación del cual depende del dispositivo, y preverá ejecutar la acción correspondiente en el dispositivo para inicializar un folleto. Si no se admite el folleto en el dispositivo, el procedimiento está vacío, y este es el valor prefijado. Para obtener más información, consulte [Compatibilidad de folletos](#).

### Sintaxis

STARTBOOKLET

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

ENDBOOKLET,SETPARAMS

## STARTDBM

**STARTDBM** inicia el modo de base de datos. Este comando puede preceder los datos en el archivo de base de datos. Se proporciona un nombre de archivo DBM como operando. Se almacena en una de las bibliotecas definidas por **SETFPATH** o en los directorios de proyectos usados por el trabajo. Se recomienda el uso de la extensión `.dbm`. El DBM contiene las instrucciones de procesamiento y de diseño de la página que permiten imprimir un documento para cada registro del archivo de base de datos.

Si se usa **SETCYCLECOPY** en un archivo maestro de base de datos, se ejecuta un **CHKPOINT** implícitamente. Si se combina con una instrucción condicional **IF/ELSE/ENDIF** en determinados registros, se permitirá el procesamiento de un conjunto de registros como un conjunto de documentos con un determinado número de copias.

Los comandos en un DBM pueden ser cualquier comando de modo nativo. Además, se utiliza **VSUB** para sustituir los contenidos de los campos.

Para obtener más información sobre el modo de base de datos, consulte Flujos de datos de VIPP® en la *Guía del usuario de FreeFlow VI Compose*.

- Para configurar un trabajo de varias en 1 a doble cara: Antes de un comando **STARTDBM**, o en el JDT asociado, coloque una instrucción **ZSORT** con `stacksize=1`. Para obtener más información, consulte **ZSORT**.
- Procesamiento correcto de los datos si el archivo de la base de datos contiene un flujo de datos de varios bytes  
Antes de un comando **STARTDBM**, coloque un comando **SETFONT** con una llamada a la fuente de varios bytes adecuada.

### Sintaxis

```
(Data Base Master name) STARTDBM<EOL> (Data Base Master name) record-grouping STARTDBM<
```

### Donde:

<b>(Data Base Master name)</b>	es el nombre DBM que se usará. Si no se suministra ningún DBM a <b>STARTDBM</b> , por ejemplo, () <b>STARTDBM</b> , se suministra un nombre DBM con cada registro en un campo llamado DBM_NAME. Esto permite que el uso de distintos DBM para cada registro.
<b>&lt;EOL&gt;</b>	es el marcador de final de línea, por ejemplo, LF, CRLF o CR. Debe seguir inmediatamente a la palabra clave <b>STARTDBM</b> . Se producirá un error del trabajo si hay espacios adicionales entre <b>STARTDBM</b> y <b>&lt;EOL&gt;</b>
<b>agrupación de registros</b>	define el número de registros que es preciso leer para el procesamiento de la DBM. El valor predeterminado es 1. La opción de agrupación de registros hace que los registros del archivo de base de datos se procesen en paquetes, el primero de los cuales contiene los nombres de los campos. Esto es útil para registros largos con muchos campos.

Dado que DBM se ejecuta para cada registro, para evitar efectos secundarios y en aras de una mayor eficiencia, puede colocar comandos generales de configuración como, por ejemplo, **DUPLEX\_on**, **SETMULTIUP**, **SETDBSEP** y otros, antes del comando **STARTDBM**. Estos comandos se pueden agrupar en un JDT y pueden ser llamados a través de **SETJDT** antes de **STARTDBM**. Por último, configure una sección de inicialización al principio de la DBM como se indica en este ejemplo:

```
/VAR_INI true /INI SETVAR
% this section executes only on the first call to the DBM
IF VAR_INI
{ TWOUP
  DUPLEX_on
  2480 3500 SETPAGESIZE
  /VAR_INI false SETVAR
} ENDIF
```

### Modos

Este comando es aplicable en el modo de base datos.

### Comandos relacionados

- **SETFPATH**
- **SETJDT**
- **STARTLM**
- **VSUB**

## STARTLM

**STARTLM** inicia el modo de línea. Para obtener más información sobre el modo de línea, consulte Flujos de datos de VIPP® en la Guía del usuario de *FreeFlow VI Compose*.

Este comando precede los datos de modo de línea. Se proporciona un nombre de archivo JDT como operando. Se almacena en una de las bibliotecas definidas por **SETJPATH** o en los directorios de proyectos usados por el trabajo. Se recomienda el uso de la extensión .jdt. Este archivo JDT contiene todas las instrucciones de procesamiento y de diseño de las páginas del trabajo.

En un JDT, use solo los comandos relacionados con la configuración del diseño de la página, como la orientación, **SETGRID**, **SETMARGIN**, **SETFONT**, **SETFORM** y definiciones de RPE. Los comandos que imprimen elementos en la página como **SHX**, **ICALL** y **DRAWB** no se pueden usar excepto en definiciones del formulario en línea o en procedimientos **ENDPAGE**.

Si un determinado comando no se especifica en el JDT, el trabajo usa la configuración predefinida en el archivo `/usr/xgf/src/xgf.def`.

### Sintaxis

```
(jdtname) STARTLM<EOL>
```

```
(jdtname2) (jdtname1) count STARTLM<EOL>
```

### Donde:

<b>jdtname</b>	es el nombre JDT que se usará. Si no se especifica ninguna referencia de JDT, por ejemplo, () <b>STARTLM</b> , no se realiza ninguna búsqueda del JDT y los comandos se incorporan entre las instrucciones %! y <b>STARTLM</b> .
<b>&lt;EOL&gt;</b>	es el marcador de final de línea; por ejemplo, LF, CRLF o CR. Debe seguir inmediatamente a la palabra clave <b>STARTLM</b> . Cualquier espacio adicional entre <b>STARTLM</b> y <EOL> hace que aparezca una línea en blanco en la parte superior de la primera página.
<b>jdtname1</b>	se aplicará a las páginas de recuento antes de aplicar jdtname2. Esto es muy útil para la impresión de páginas de pancarta con un determinado JDT.

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

- **LMSKIP**
- **SETJDT**
- **SETJPATH**
- **STARTDBM**

## STARTOFSET

**STARTOFSET** actúa como delimitador de juegos solo para trabajos de impresión del servidor de impresión de FreeFlow. Utilice este comando con el servidor de impresión FreeFlow. Este comando se puede colocar al principio de la primera página del juego.

Para obtener más información sobre la configuración y el uso de las funciones de acabado de subconjuntos de los productos del servidor de impresión FreeFlow, consulte la publicación del servidor de impresión FreeFlow.



Nota: Este comando se puede utilizar en FFPS sin **SETFINISHING**. El acabado puede definirse en FFPS mediante una cola con la opción de salida de subjuegos configurada como Recuperado de PDL.

### Sintaxis

STARTOFSET

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **ENDOFSET**
- **SETFINISHING**
- **SETOBIN**

## STARTXML

**STARTXML** inicia el modo XML. Este comando precede a los datos XML o se utiliza después de un comando **SETLMFILE** que hace referencia a un archivo XML. Toma como un operando una referencia a un archivo de ficha de trabajo de XML (XJT). El archivo XJT se almacena en una de las bibliotecas definidas por **SETJPATH** o en los directorios de proyectos usados por el trabajo. Se recomienda el uso de la extensión `.xjt`.

Si el operando XJT es una cadena vacía, VI Compose crea el nombre XJT mediante la unión de `.xjt` y el nombre de la etiqueta raíz del archivo XML. Si no existe este XJT, VI Compose utiliza un XJT denominado `xml.dump.xjt` ubicado en `xf/jdtlib`.

El archivo XJT contiene instrucciones sobre cómo procesar y ordenar los datos XML en un documento. Es similar a un JDT en modo de línea, y las mismas reglas y limitaciones se aplican a sus contenidos; consulte **STARTLM**. Contiene todas las definiciones de formato global del documento, la orientación, los formularios, los materiales de impresión, las tramas, las fuentes, los colores además de una tabla de definiciones de procesamiento de XML (XPD) que describe determinadas acciones que deberán realizarse en ciertas etiquetas XML.

**STARTXML** analiza y procesa el archivo XML, a la vez que rellena el directorio `VXVpath` con contenidos del nodo y realiza acciones o sustituciones según lo definido en la tabla XPD.

Si **SETCYCLECOPY** se usa en un proceso de inicio de BTA para una determinada etiqueta, se ejecuta de forma implícita un **CHKPOINT** cuando se detecta la etiqueta de finalización. Esto permite que todo el árbol secundario XML se procese como un documento (juego) con un número determinado de copias.

**STARTXML** termina después de la etiqueta raíz de final. En ese momento VI Compose vuelve al modo nativo.

Se admiten los archivos XML codificados mediante UTF-16 si los datos XML comienzan con un BOM UTF-16 (0xFFFE o 0xFEFF). Tenga en cuenta que los datos XML de UTF-16 se convierten en UTF-8 automáticamente, por lo que los contenidos del nodo se imprimen con fuentes con codificación UTF-8.

### Sintaxis

```
(xjtname) STARTXML
```

```
() STARTXML
```

### Modos

Este comando es aplicable en el modo XML.

### Comandos relacionados

- **BEGINXPD**
- **ENDXPD**
- **BTA**
- **ETA**
- **BTS**
- **ETS**
- **SETJPATH**

## STOREVAR


Utilice **STOREVAR** para almacenar datos en un archivo para usarlos posteriormente en la misma aplicación o en otra aplicación distinta. Los contadores duraderos o los datos compartidos por diferentes aplicaciones son ejemplos de datos que querrá almacenar mediante **STOREVAR**.

Para que **STOREVAR** sea capaz de escribir en un archivo, es preciso satisfacer tres condiciones:

- El archivo y el directorio deben tener acceso de escritura a nivel de sistema de archivos
- FFPS, si este es el objetivo, puede permitir a un programa PS escribir en el directorio en el que el archivo se escribe
- En el modo de proyecto, el archivo **STOREVAR** debe existir antes de enviar el trabajo

Utilice **STOREVAR** para definir en qué archivo se colocarán los datos que se van a almacenar. El archivo de almacenamiento real se define antes de utilizar **STOREVAR** en el modo de proyecto. Esto es necesario porque en el modo de proyecto, los recursos deben ser asignados a uno de los tres niveles de ámbito, local, carpeta, global, y es importante porque si diversos proyectos usan **STOREVAR** con el mismo nombre de archivo, el objetivo podría ser compartir dicho archivo, ámbito de carpeta o global, por lo que todos los proyectos escriben en el mismo archivo, o mantener una instancia del archivo para cada proyecto, ámbito local. Mediante la creación del archivo en el directorio correspondiente, por ejemplo, en `/usr/xgfc/FolderX/ProjectY`, `/usr/xgfc/FolderX/fshared` o `/usr/xgfc/gshared`, antes de enviar los trabajos que escriben en él, el usuario indica de forma implícita el ámbito del archivo.

Cuando se crea un VPC, es preciso que el archivo **STOREVAR** sea parte del VPC, incluso si está vacío, para que se cree al implementarse el proyecto. La impresión de un VPC mediante **STOREVAR** en modo impresión y olvidar impedirá que **STOREVAR** cumpla con su cometido, compartir información entre trabajos, ya que el directorio de proyectos `vpcpfd` se destruye al finalizarse el trabajo.

 Importante: Este comando almacena datos en el sistema de archivos del controlador de la impresora en un área que suele estar accesible por parte de otras aplicaciones o usuarios. Utilice el comando solamente según lo previsto. Para las variables internas de un determinado trabajo VIPP®, utilice el comando **SETVAR**. No utilice el comando **STOREVAR**.

 Nota: Se ha mejorado **STOREVAR** en VI Explorer y VI Design Pro:

**STOREVAR** se ha desactivado en VIE.

En VDP, **STOREVAR** se desactiva al navegar por las páginas, con los botones de página Siguiente y Anterior. Cuando se pulsa el botón Actualizar, el efecto de **STOREVAR** se puede visualizar, pero no se mantiene. La actualización se realiza en una copia temporal del archivo y se descarta al salir.

### Sintaxis

```
data (store_file) STOREVAR
data (store_file) /option STOREVAR
```

Donde:

- datos** puede ser una cadena entre paréntesis, un entero, un número real o una variable de uno de estos tipos.
- store\_file** es el nombre del archivo en el que se guardan los datos. Se debe encontrar en una de las bibliotecas referenciadas por **SETMPATH**. Si el archivo existe, los datos sobrescriben su contenido. Si no existe, se crea en la primera biblioteca referenciada por **SETMPATH**. La biblioteca y el archivo deben tener los permisos de escritura correspondientes.
- opción** puede ser cualquier combinación de:
- M** permitir a **STOREVAR** en el modo de varias instancias VIEC. Si se usa esta opción, los usuarios deben asegurarse de que el nombre `/store_file` es único en cada ejecución de **STOREVAR**, y que dos trabajos que se ejecutan en paralelo en distintas instancias de VIEC nunca ejecutan **STOREVAR** con el mismo nombre de archivo.
  - P** en el modo de proyecto, permite crear `store_file` en el primer directorio **SETPPATH** si este no existe. Si se usa esta opción, los usuarios deben asegurarse de que el directorio de destino coincide con el flujo de trabajo.
  - A** se abre `store_file` en el modo de agregar. Si se utiliza esta opción el contenido de `store_file` no se sobrescribe. En su lugar, los datos se agregan.
  - T** si los datos son de tipo cadena, se escriben en el archivo sin paréntesis, modo de texto frente a modo de PostScript.

Ejemplo:

```
(Hello\nMr Smith\n) (textXYZ.txt) /AT STOREVAR
```

agrega las dos líneas siguientes al contenido de `textXYZ.txt`:

```
Hello Mr. Smith
```

## Ejemplos

Este ejemplo muestra cómo almacenar, recuperar e incrementar una variable.

```
VAR.NUMBER (number.var) STOREVAR      (this stores the variable)
/VAR.NUMBER (number.var) RUN SETVAR    (this retrieves the variable)
/VAR.NUMBER ++                        (this increments the variable)
```

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

**SETMPATH, SETVAR**



## SUB

**SUB** resta un valor de una variable numérica definida por **SETVAR** o una variable de XML.

### Sintaxis

```
/VARname number SUB
/^XMLname number SUB
```

### Donde:

<b>/VARname</b>	hace referencia a una variable numérica previamente inicializada por <b>SETVAR</b> .
<b>/^XMLname</b>	hace referencia a una variable de XML que no requiere su inicialización explícita. VI Compose inicializa todas las variables XML en una cadena vacía, que equivale a una cadena numérica igual a cero.
<b>número</b>	es el número que se resta de la variable. Puede ser un número entero, o real, o una cadena numérica. Cuando se trata de un gran número, es obligatorio contar con una cadena numérica.

Las cadenas numéricas admiten números grandes de hasta 40 dígitos, 25 dígitos para la parte entera y 15 dígitos para la parte decimal. En una cadena numérica, el signo negativo y el delimitador decimal se definen mediante los parámetros /DecimalPoint y /NSign y se pueden producir en cualquier lugar de la cadena.

Es obligatorio configurar estos parámetros con los valores correspondientes para garantizar unos resultados precisos. Los valores predeterminados se definen en el archivo `/usr/xgff/src/xgff.def`. Se omiten los caracteres en la cadena numérica que no sean estos dos más los dígitos 0–9.

La longitud inicial de la cadena definida por **SETVAR** se amplía automáticamente hasta 40 dígitos cuando es necesario.

Los números reales y enteros se utilizan solo para los valores pequeños  $\leq 99999$  por ejemplo, para la implementación de un contador. El delimitador decimal, si lo hubiera, siempre es el punto (.). El signo negativo, si lo hubiera, siempre es el signo menos (-) y es el primer carácter.

Las variables integradas están disponibles en formularios o procedimientos **ENDPAGE** si se usa **RUNTIF**:

<b>TIFPAGE</b>	número de página de la página actual
<b>TIFLAST</b>	número de página de la última página

### Ejemplos

```
/VAR.CNT1 0 SETVAR /VAR.CNT1 12 SUB /VAR.CNT1 -3 SUB
```

```
/VAR.SUM (0) SETVAR /VAR.SUM (1,234,890,566,00-) SUB
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- ADD
- SETVAR
- ++ y –
- DIV

## TIFORI\_off

**TIFORI\_off** desactiva el procesamiento de la etiqueta 274 Orientación en **ICALL**. **TIFORI\_off** es el valor prefijado.

### Sintaxis

`TIFORI_off`

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

`TIFORI_on,ICALL`

## TIFORI\_on

**TIFORI\_on** activa el procesamiento de la etiqueta 274 Orientación en **ICALL**. El valor predeterminado de **ICALL** no procesa la etiqueta de orientación **TIFORI\_off** y asume que la orientación es 1, el valor predeterminado de los archivos TIFF. El procesamiento de la orientación se proporciona como una opción para garantizar la compatibilidad con versiones anteriores.

### Sintaxis

TIFORI\_on

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

TIFORI\_off,ICALL

## TUMBLEDDUPLEX\_off

**TUMBLEDDUPLEX\_off** desactiva la impresión a 2 caras. **TUMBLEDDUPLEX\_off** es el valor prefijado. Este comando equivale a **DUPLEX\_off** y solo está disponible para garantizar la compatibilidad con versiones anteriores.

### Sintaxis

`TUMBLEDDUPLEX_off`

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **BCALL**
- **DUPLEX\_off**
- **DUPLEX\_on**
- **ENDIMP**
- **TUMBLEDDUPLEX\_on**

## TUMBLEDDUPLEX\_on

**TUMBLEDDUPLEX\_on** permite la impresión a 2 caras con giro, giro de 180° en la cara posterior. El valor prefijado es **TUMBLEDDUPLEX\_off**.

### Sintaxis

TUMBLEDDUPLEX\_on

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **BCALL**
- **DUPLEX\_off**
- **DUPLEX\_on**
- **ENDIMP**
- **TUMBLEDDUPLEX\_off**

## TWOUP

**TWOUP** activa el modo 2 en 1. Dos páginas lógicas de tamaño reducido se imprimen una encima de la otra, **LAND** e **ILAND** o una al lado de la otra **PORT** e **IPOINT** en la misma página física.

Este comando debe usarse como el primer comando después de la línea %! en el archivo de impresión o como la primera línea en un JDT.

### Sintaxis

TWOUP

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **BCALL**
- **ENDIMP**
- **ONEUP**
- **SETMULTIUP**

## UPDATE

**UPDATE** se usa de forma exclusiva en un bucle **FOREACH** para actualizar la entrada de la tabla después actualizar uno o diversos valores en esta entrada.

### Sintaxis

UPDATE

### Ejemplos

Este ejemplo agrega 1 a cada valor *VAR1* en la tabla:

```
{ /VAR1 ++ UPDATE } VARtable FOREACH
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **FOREACH**
- **SETVAR**
- **ADD**
- **SORT**



## USPS4CB

**USPS4CB** crea e imprime códigos de barras de 4 estados USPS, Intelligent Mail® a partir de los datos de seguimiento y las cadenas de códigos de ruta especificados. No se requiere ninguna fuente especial.

Las dimensiones de las barras cumplen las especificaciones de las revisiones D y E de la especificación del código de barras *USPS Intelligent Mail® USPS-B-3200, código de barras del cliente de 4 estados* de 24/07/07 y 30/10/07.

### Sintaxis

```
(tracking data) (routing code) USPS4CB (tracking data) (routing code) align USPS4CB
```

Donde:

**datos de seguimiento**

es la cadena de datos de seguimiento de 20 dígitos que consiste en los campos secundarios siguientes:

Identificador de código de barras de 2 dígitos. El segundo dígito debe ser del 0 al 4.

Identificador de tipo de servicio de 3 dígitos.

Identificador de correo de 6 dígitos.

Número de serie de 9 dígitos.

Esta cadena puede tener espacios que coinciden con el formato humano legible, que puede ser requerido por algunos usos del código de barras de 4 barras y para mejorar la legibilidad.

**código de ruta**

es el código postal de punto de entrega. Puede ser en cualquiera de las 4 formas aceptadas por USPS:

( ) Una cadena vacía o nula para ningún código postal.

**(12345)** Un código postal de 5 dígitos.

**(12345-6789)** Un código de 9 dígitos ZIP+4

**(12345 -6789 01)** Un código postal ZIP+4 de once dígitos + DPC de 2 dígitos

Esta cadena puede tener espacios (que coincidan con el formato legible por el ojo humano) que pueden ser necesarios para algunos usos del código de barras de 4 estados o guiones (-) para mejorar la legibilidad.

**alinear**

indica qué punto del código de barras se alinea en la posición de impresión secundaria mediante estos valores:

**0** superior izquierda

**1** superior derecha

**2** centro superior

**10** inferior (predeterminado)

**11** inferior derecha

**12** centro inferior

**20** centro izquierda

**21** centro derecha

**22** centro centro

Los datos de seguimiento y las cadenas de datos de enrutamiento pueden tener espacios, guiones, u otros caracteres imprimibles integrados en las cadenas de caracteres. Estos caracteres se eliminarán antes de que se cree el código de barras. Después de eliminarlos, la cadena de datos de seguimiento debe tener exactamente 20 dígitos, el segundo dígito debe figurar en el intervalo 0-4, y la cadena de código de enrutamiento estará

vacía o tendrá 5, 9 u 11 dígitos. El código de barras se imprimirá en la posición secundaria actual, alineada según lo especificado. Es responsabilidad del usuario asegurarse de que se mantienen las zonas transparentes correspondientes alrededor del código de barras.

### Ejemplos

```
(01 234 567094 987654321) ( ) USPS4CB
(01234567094 987654321) (01234) 0 USPS4CB
(01 234 567094987654321) (01234-5678) 20 USPS4CB
(01 234 567094 987654321 ) (01234 5678 91) 10 USPS4CB
```

Las cadenas de datos de enrutamiento y los datos de seguimiento en el último ejemplo tienen el formato requerido para la impresión de datos en forma legible por el ojo humano como sea necesario para determinados usos del código de barras de 4 estados USPS Intelligent Mail®.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

MOVETO , MOVEH,MOVEHR

## VPDISTPAT

Cuando se utiliza la nueva función de patrón de vectores de correlación, puede que el texto aparezca visible a simple vista en algunos motores de impresión dependiendo de los colores seleccionados para el efecto.

En esta situación es posible utilizar VPDISTPAT para aplicar un patrón de distracción sobre el efecto.

### Sintaxis

`option VPDISTPAT`

### Donde

La opción puede ser uno de los siguientes números enteros:

- **0** ningún patrón de distracción, igual que si no se utiliza VPDISTPAT
- **1** patrón de distracción ligero, 1 o 2 capas
- **2** patrón de distracción intenso, 1 o 2 capas
- **11** patrón de distracción ligero utilizando solo col2, 2 capas
- **21** patrón de distracción ligero utilizando solo col3, 2 capas
- **12** patrón de distracción intenso utilizando solo col2, 2 capas
- **22** patrón de distracción intenso utilizando solo col3, 2 capas

### Ejemplos

```
500 3000 1750 300 VP02 DRAWB
/NHEB 50 SETFONT
575 2770 MOVETO (Correlation Text) SHL
1 VPDISTPAT
```

### Modos

Todos los modos.

### Comandos relacionados

**SETPAT**

## XGF

XGF hace que el intérprete de PostScript cargue VI Compose. VI Compose se carga implícitamente a través de estos comandos:

DUPLEX_on	RUN	SETLMFILE	STARTDBM
FSHOW	SETDBSEP	SETMARGIN	STARTLM
ILAND	SETDLFILE	SETMEDIA	STARTXML
INDEXFONT	SETEPATH	SETMEDIAT	TUMBLEDDUPLEX_on
IPOINT	SETFONT	SETMPATH	TWOUP
LAND	SETFORM	SETMULTIUP	XGF
MAKEVMFILE	SETEPATH	SETPARAMS	XGFDEBUG
MAKEVMFORM	SETGRID	SETPROJECT	XGFRESDEF
ORITL	SETIPATH	SETTAB	
PAGERANGE	SETJDTSETJPATH	SETUTAB	
PORT	SETJPATH	DESPLAZAMIENTO	

Para que el intérprete de PostScript cargue VI Compose, el archivo VIPP® enviado para la impresión debe comenzar con uno de los comandos indicados más arriba.

Si debe empezar el archivo VIPP® con otro comando, debe estar precedido por XGF o registrado como un comando inicial mediante **XGFENTRY** en el archivo `/usr/xgf/src/xgf` y el archivo `/usr/xgf/src/xgfunix.run` después del final de línea.

### Sintaxis

XGF

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**XGFENTRY**

## XGFDEBUG

**XGFDEBUG** fuerza la impresión de prefijo modo nativo (NMP) y el avance de formulario por motivos de documentación y de depuración. **Los comandos NMP** se procesan como datos de impresión y no se interpretan como comandos VIPP®. Los avances de formulario se imprimen como <FF> y no tienen ningún efecto de omisión.

Este comando puede programarse en un JDT o antes del comando **STARTLM**.

### Sintaxis

XGFDEBUG

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

Ninguno

## XGFEND

**XGFEND** es obligatorio al final del trabajo de un modo nativo si el servicio de demografía o normalización está activo, o se usa el **ZSORT** genérico. De lo contrario, este comando realiza ninguna acción.

## XGFENTRY

**XGFENTRY** registra un comando como un comando inicial. Para obtener más información, consulte XGF.

### Sintaxis

```
/command XGFENTRY
```

### Ejemplos

Este ejemplo registra **MOVETO** como comando inicial.

```
/MOVETO XGFENTRY
```



Nota: Utilice el comando solo en el archivo `/usr/xgf/src/xgf` y el archivo `usr/xgf/src/xgfunix.run` después de la línea final.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**XGF**



## XGFRESDEF

**XGFRESDEF** registra los recursos especificados en la memoria del intérprete de PostScript. Use este comando para integrar recursos en el flujo de datos para preparar trabajos VIPP® para su impresión en una impresora descentralizada. Esto se conoce como un flujo de datos VIPP® autónomos. La sintaxis del comando varía según el tipo de recurso. Para obtener más información, consulte los recursos VIPP® en la *Guía del usuario de FreeFlow VI Compose*.

### Sintaxis

```
/formname form_data FBIND XGFRESDEF /resourcename { resource_data } XGFRESDEF /imagenam
```

Donde:

<b>formname</b>	es el nombre de un formulario VIPP®.
<b>form_data</b>	es el contenido del formulario, incluidas las llaves de encapsulación (por lo tanto, no se muestran en la sintaxis).
<b>resourcename</b>	es el nombre de un segmento, un DBM, un JDT, una lista de fuentes o una tabla de codificación.
<b>resource_data</b>	es el contenido exacto del recurso.
<b>ImageName</b>	es el nombre de un archivo TIFF o JPEG.
<b>image_data</b>	es el contenido exacto del archivo TIFF o JPEG que se convierte en registros de longitud fija hexadecimal encapsulados entre < y >. El último registro puede ser más corto. La longitud de registro debe ser tal que el número de registros hexadecimales no exceda 65535. En general, 128 es suficiente.
<b>psname</b>	es el nombre de un archivo PostScript creado por un controlador de PostScript en una aplicación de procesamiento de documentos.
<b>ps_data</b>	es el código PostScript creado por el controlador.
<b>ilname</b>	es el nombre de un recurso en línea que debe almacenarse en la memoria caché. Normalmente, un recurso en línea no tiene ningún nombre, por lo que no se puede almacenar en la memoria caché. Utilice esta sintaxis para darle un nombre y usarlo posteriormente con <b>SCALL</b> , <b>SETFORM</b> o <b>SETBFORM</b> .
<b>VIPP_code</b>	es el contenido de los recursos en línea.
<b>llx lly urx ury</b>	es la cadena de límite del recurso en línea en las unidades VIPP® actuales, para obtener más información, consulte <b>%%BoundingBox</b> . Si no se especifica, el cuadro de límite predeterminado es el tamaño de la página física.

### Ejemplos

Este ejemplo ilustra distintos usos de **XGFRESDEF**.

```
/bill.frm { PORT 100 1150 MOVETO (Xerox) SHL ... } FBIND XGFRESDEF /a1b.jdt { ILAND 120
```



Nota: Los recursos incorporados aparecen en el flujo de datos antes de ser referenciados por un comando VIPP.



Consejo: **SETLMFILE** y **SETDLFILE** se pueden integrar en el flujo de datos mediante **XGFRESDEF**.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **CACHE**
- **FCALL**
- **ICALL**
- **RUN**
- **SCALL**
- **SETBFORM**
- **SETDLFILE**
- **SETENCODING**
- **SETFORM**
- **SETJDT**
- **STARTDBM**
- **STARTLM**

## XMLATL

**XMLATL** proporciona una manera de procesar los atributos de una etiqueta XML. Debe colocarse en el interior de un par de comandos **BTA** y **ETA** o **BTS** y **ETS**. Toma un proceso como argumento y realiza una llamada a este para cada atributo después de configurar los valores correspondientes en **XMLATN** y **XMLATV** para cada atributo. Se espera que el procedimiento realice alguna acción, imprimir, almacenar, definir variables, etc. con cada par **XMLATM** o **XMLATV**.

### Sintaxis

```
{ VIPP commands and XMLATN and XMLATV variables } XMLATL
```

### Donde

<b>XMLATN</b>	es el nombre del atributo.
<b>XMLATV</b>	es el valor del atributo.

### Modos

Este comando es aplicable en el modo XML.

### Comandos relacionados

- **BTA**
- **ETA**
- **BTS**
- **ETS**

## XMLSYN

**XMLSYN** define sinónimos de las etiquetas XML. A continuación, se deben utilizar estos sinónimos en lugar de los nombres de originales en las definiciones de XPD y **VXVpaths** XML.

El comando permitirá solucionar problemas de nombres de etiquetas largos en estructuras XML complejas que pueden hacer que **VXVpath** exceda los 128 caracteres, que es la longitud máxima de **VXVpaths**. Ahora se puede asignar un sinónimo para el nombre de etiqueta XML y referenciar el nombre asignado en VIPP®.

### Sintaxis

```
[ /tag_name1 /tag_syn1 /tag_name2 /tag_syn2 ... ] XMLSYN
```

### Donde

**tag\_nameX** es el nombre de una etiqueta del árbol de datos XML actual.

**tag\_synX** es el sinónimo que se sustituirá para esa etiqueta.

### Ejemplos

```
[ /ACCOUNT_INVOICE /AINV /ACCOUNT_INFORMATION /AINF /ACCOUNT_UNITS /AU /ACCOUNT_USAGE_C
```

### Modos

Este comando es aplicable solo en el modo XML.

### Comandos relacionados

- **BEGINXPD**
- **ENDXPD**
- **BTA**
- **ETA**
- **BTS**
- **ETS**

## ZSORT

**ZSORT** permite imprimir documentos de varias en 1, a 1 y 2 caras, en un orden de imposición de norte a sur o en pilas. Dichas pilas se pueden llevar a una cortadora para cortar y apilarlas, manteniéndose así el orden de registro a lo largo de la pila. Esto suele usarse para mantener el orden del código postal. Además, se puede crear e insertar una hoja intermedia entre las pilas de la salida para identificar los límites de la pila. **ZSORT** también permite mantener la alineación de registro anterior y posterior cuando los datos variables deben colocarse en las caras anterior y posterior de un documento de varias páginas en una.

**ZSORT** se debe ejecutar tan pronto como sea posible en el trabajo. Por ejemplo, se puede colocar al principio del archivo de envío antes del comando **START** que precede a los datos o en el **JDT** o el **XJT** invocado antes del comando **START**.

### Sintaxis

```
{ slipsheet proc } stacksize options ZSORT { slipsheet proc } stacksize [options groupi
```

### Donde

<b>slipsheet proc</b>	es el código VIPP® que se ejecutará al final de una pila para producir una hoja intermedia. Este procedimiento se procesa mediante el comando <b>SLIPSHEET</b> y debe ser un procedimiento válido para ese comando. Este procedimiento puede estar vacío si no es la hoja intermedia que se desea.
<b>stacksize</b>	es el número máximo de hojas en una pila. El tamaño de la pila normalmente se basa en el número máximo de hojas que se puede cortar mediante una cortadora de postproceso. Este tamaño debe ajustarse en función de la capacidad de almacenamiento de la impresora y el nivel medio de datos por pila. Un tamaño de pila razonable es entre 500 y 1000 hojas.

**opciones**

es un número entero con la forma XXXY, donde:

XXX es el número de páginas del documento menos 1.

Y puede ser una de:

**0 solo modo de base de datos:** reduce la última pila para ajustar el número de registros con doble cara continua

Si el documento tiene un número impar de páginas y la última página está en la página frontal, la página posterior contendrá la primera página del documento siguiente.

**1 solo modo de base de datos:** agregue páginas lógicas en blanco a la última pila para que coincida con el tamaño de la pila con a 2 caras continuo

**2 solo modo de base de datos:** reduzca la última pila para ajustar el número de registros con a 2 caras discontinuo.

Cuando el documento tiene una cantidad impar de páginas, el reverso de la última página se deja en blanco.

**3 solo modo de base de datos:** agrega páginas lógicas en blanco a la última pila para que coincida con el tamaño de la pila con doble cara no continua.

**4 todos los modos:** reduce la última pila para ajustar el número de registros.

**5 todos los modos:** agregue páginas lógicas en blanco a la última pila para que coincida con el tamaño de la pila.

**grouping**

es un número entero que debe ser un separador del número de varias en 1. Permite que varias páginas lógicas consecutivas en la hoja se traten secuencialmente en lugar de a través de la pila. La selección y el orden de las páginas se determina a través del orden de llenado Varios en 1, /FillOrder para SETLAYOUT u orden de secuencia para SETMULTIUP.

**Modo de base de datos y ZSORT genérico**

Las opciones Y=0-3 se aplican al modo de base de datos solamente mientras las opciones 4-5 se aplican a todos los modos ZSORT genéricos.

Las opciones de 0–3 están optimizadas para el modo de base de datos, por lo que se recomienda utilizarlas cuando sea posible. Su comportamiento es un poco diferente de las opciones 4–5 y disponen de algunos detalles y limitaciones detalladas a continuación.

**Detalles y limitaciones de las opciones Y=0-3, modo de base de datos**

- Llamadas DBM y documentos de diversas páginas**

Cada registro será procesado XXX+1 veces. El DBM será llamado ese número de veces, e incluirá lógica condicional para que cada llamada produzca una única página a la vez. Para ello, se proporciona el número de página del documento en cada llamada realizada por la variable ZSPAGE integrada. Para garantizar la compatibilidad con versiones anteriores, se puede usar la variable BACK en documentos a 2 caras de 2 páginas.

Por lo tanto, aunque **ZSORT** puede utilizarse para documentos de varias páginas, cada registro debe generar el mismo número de páginas. Por ejemplo, si imprime un boletín de cuatro páginas, todos los registros deben generar cuatro páginas. Si el número de páginas por documento no es fijo, deben utilizarse las opciones 4–5.

- **Variables integradas**

Se dispone de 4 variables integradas que solo pueden utilizarse en la DBM:

- **ZSPAGE** proporciona el número de páginas del documento incrementadas desde 1.
- **ZSRECNUM** produce al número de registros en el orden del archivo de datos. Este es el número de registro actual, aumentado desde el valor de 0.
- **ZSREPCNT** produce un número de repeticiones continuo, empezando por 0.
- **ZSREPIDX** ofrece un índice de repetición de registros. **ZSREPIDX** se incrementa para cada instancia pero se restablece a 0 al principio del siguiente registro.

- **Opción de repetición del registro de variables**

**ZSORT** puede tener en cuenta un valor de repetición asociado a cada registro. Esto se logra a través de la declaración del campo repetir en el JDT o antes de **STARTDBM** con el parámetro **/ZSRepeatField**, como se muestra más abajo. El campo de repetición debe contener un valor numérico.

```
[ /ZSRepeatField /RecRepeat ] SETPARAMS
```

Por ejemplo, al imprimir tarjetas de presentación, es posible que tenga un registro por cliente, cada uno de los cuales dispone de un campo de repetición llamado *CardVolume*. Al definir **ZSRepeatField** según *CardVolume*, **ZSORT** procesará cada registro el número de veces indicado por *CardVolume*. Por ejemplo, un registro podría producir 250 tarjetas de presentación, y el registro siguiente produce 100.

```
[ /ZSRepeatField /CardVolume ] SETPARAMS
```

También es posible insertar una hoja intermedia de repetición después de cada conjunto de repetición mediante la sintaxis siguiente:

```
[ { stack slipsheet } { repeat slipsheet } ] stacksize options ZSORT
```

- **Limitación de tamaño de la pila**

**ZSORT** lee y almacena en un búfer el número de registros necesarios para imprimir una pila completa en la memoria virtual (máquina virtual). La cantidad de máquina virtual requerida para una pila puede calcularse como se indica a continuación:

$\text{Número} \times \text{tamaño de pila} \times \text{tamaño de búfer/páginas por documento}$

En función del dispositivo de destino, una asignación de máquina virtual muy grande puede hacer que el trabajo se ejecute con un mal rendimiento y, posiblemente, se anulará con el error de máquina virtual agotada. Xerox recomienda reducir los valores más arriba, especialmente el tamaño del búfer y de la pila, a números razonables y apropiados.

**Por ejemplo:**

El trabajo configurado más arriba consumirá 60 MB ( $8 \times 2500 \times 3000$ ) de la máquina virtual y puede producir el error de máquina virtual agotada. Si los registros no superan los 300 bytes y la cortadora no puede aceptar más de 1000 hojas, configure el trabajo de la manera siguiente:

```
8-up
300SETBUFSIZE
{ }100010ZSORT
```

Esto reducirá la asignación de la máquina virtual a 2.4 MB y garantizará la velocidad nominal y una finalización sin errores.

- **Configuración de FFPS y PPR**

En FFPS, si un trabajo que usa **ZSORT** se envía a un flujo de trabajo RIP paralelo (PPR) cada segmento de PPR debe producir una pila completa, incluidas las hojas intermedias, si las hay. Se recomienda normalizar el trabajo VIPP® “crear un archivo a.vpn” antes de la impresión o como parte del proceso de impresión de trabajos. Si el trabajo no se ha normalizado en primer lugar, y la división de PPR se realiza directamente en el archivo de la base de datos, el segmento PPR deberá configurarse según el número de registros necesario para generar una pila completa.

Por ejemplo, si el tamaño necesario de la pila es 500 hojas, debe configurar el tamaño de segmento PPR según el número de registros que creará pilas de 500 hojas. Si el trabajo es una aplicación de postales 4 en 1 y una página por registro, el segmento debe contener 2000 registros (500\*4). Si se repite el registro, necesitará menos registros para realizar la pila. Con un valor de repetición de 10 solo necesitarían 200 registros (500\*4/10).

### **ZSORT genérico (opciones Y=4–5)**

El **ZSORT** genérico se puede aplicar a cualquier trabajo VIPP®. No requiere ninguna otra configuración especial e implica el uso del servicio de normalization de forma interna. Se crea un archivo normalizado temporal y se elimina de cada pila. Se puede utilizar cuando no se puede aplicar **ZSORT** de la base de datos.

### **Notas y recomendaciones**

Ajuste el tamaño de la pila según un múltiplo del número de páginas por documento, dividido por 2 si es a 2 caras. Si este no es el caso, VIPP® automáticamente realizará el ajuste, y el tamaño final de la pila será significativamente diferente del codificado en la instrucción **ZSORT**.

Las páginas lógicas se pueden ajustar completamente en la página física. Si **SETMULTIUP** se usa para definir el diseño de varias en 1, debe usarse la sintaxis que especifica el tamaño de cada página lógica. Si no se utiliza la sintaxis correcta, las inserciones en la cara posterior serán incorrectas.

### **Ejemplos**

Esto es un ejemplo de codificación de **ZSORT** en un documento de 3 páginas:

```
{ }50020ZSORT
```

La DBM puede codificarse como:

```
IF ZSPAGE 1 eq { code to produce page 1 } ENDIF IF ZSPAGE 2 eq { code to produce page 2 }
```

Este ejemplo muestra la codificación para la impresión de postales 4 en 1 en papel de carta EE. UU. en pilas de 100 hojas, con un cambio a material de impresión rojo para páginas con las palabras “Hoja intermedia” al final de cada pila; se supone que se usan unidades POINT. La pila se acorta si el número de registros de entrada no es un múltiplo de 400.



```

612 792 SETPAGESIZE
[ /PageWidth 306
  /PageHeight 396
  /Across 2
  /Down 2
] SETLAYOUT

{ ONEUP (:red) SETMEDIA
{/NHEB 24 SETFONT 300 1650 MOVETO (SlipSheet) SH } SETFORM
} 100 00 ZSORT

```

Esta es la instrucción **SETMULTIUP** para definir el diseño de 4 en 1 igual que el ejemplo anterior:

```

[ [ 0 306] [396 396] 0 1 1 [306 306] [396 396] 0 1 1
[ 0 306] [ 0 396] 0 1 1
[306 306] [ 0 396] 0 1 1
] SETMULTIUP

```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- SETLAYOUT
- SETMULTIUP
- DUPLEX\_on
- DUPLEX\_off
- SETBUFSIZE



## Marcadores

Este capítulo incluye:

%.....	476
% % .....	477
%!.....	478
% %BoundingBox .....	479
% %EOD_XGF .....	480
% %EOF.....	481
% %PagesPerBooklet .....	482
% %XGF.....	483
\$\$name.....	484
[=name=].....	485
BBOX .....	486
EXPAND.....	487
EXTVAR .....	488

Los marcadores son cadenas o secuencias reservadas que, cuando se incrustan en los lugares adecuados de un trabajo VIPP®, pueden afectar al comportamiento de determinados comandos.

%

La aparición del carácter % fuera de los paréntesis permite agregar un comentario. El comentario consiste en los caracteres entre el carácter % y un carácter de final de línea (EOL) como LF, CR o CRLF.

### **Sintaxis**

% comment

### **Modos**

Este comando es aplicable en todos los modos.

### **Comandos relacionados**

Ninguno

% %

El comando % % indica una instrucción de la convención de estructura de documentos (DSC, por sus siglas en inglés). Algunos servidores de impresión realizan determinadas acciones en subconjuntos de instrucciones de DSC. Por lo tanto, para satisfacer las especificaciones de DSC, todos los archivos VIPP® deben empezar con un encabezado DSC estándar, como se muestra en este ejemplo.

% !

```
%%Title: xxxxxxxx.ps
```

```
%%DocumentMedia: dmFAC 596 841 0 white
```

```
%%+ dmBVR 596 841 0 white BVR
```

```
%%EndComments
```

La instrucción % %DocumentMedia permite a DocuPrint NPS comprobar la carga correcta del material de impresión especificado en las bandejas de la impresora antes de procesar el trabajo. Si el material de impresión cargado no coincide con el requerido por % %DocumentMedia, el trabajo se pone en el estado MediaWait hasta que se coloca el material de impresión especificado.

Esta instrucción es obligatoria si se usa el comando SETMEDIA en el trabajo para cambiar el material de impresión según la página.



Nota: Si se omite la instrucción % %DocumentMedia, el trabajo se imprime en el material de impresión predeterminado y, por lo tanto, la selección del material de impresión a través de SETMEDIA puede producir unos resultados impredecibles.

## Sintaxis

```
%%DSC statement
```

Consulte la documentación correspondiente de Adobe para obtener más información sobre las instrucciones de DSC.

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

SETMEDIA

%!

El comando %! indica el inicio de un archivo PostScript o VIPP®. Una línea que comience con %! siempre debe ser la primera línea de un trabajo VIPP®. Permite al servidor del trabajo en el dispositivo de impresión identificar un flujo de datos PostScript y realizar una llamada al intérprete de PostScript.

### Sintaxis

%! comment



Nota: Si un determinado archivo no empieza con %!, es posible que el trabajo se imprima a través de un intérprete incorrecto; por ejemplo, que se imprima el trabajo como texto ASCII por error.



Consejo: %! solo es obligatorio al principio del archivo enviado. No es obligatorio al principio de los archivos de recursos.

%! también actúa como un nuevo delimitador del modo de línea a menos que se use SOF\_off. Consulte *Flujos de datos de VIPP®* en la *Guía del usuario de FreeFlow VI Compose* para obtener más información.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- %%EOF
- SOF\_off
- STARTDBM
- STARTLM

## % %BoundingBox

**% %BoundingBox** define el tamaño y la posición en relación al origen del área de imagen de un segmento. Se utiliza al principio de un segmento cuando se realiza una llamada con **CACHE/SCALL**. Cualquier marca fuera del área de la imagen no puede aparecer en la página.

Xerox recomienda que la declaración se agregue al principio de un trabajo VIPP® para informar a los controladores de la impresora y los servicios de gestión de salida sobre el área de las páginas del trabajo. Por ejemplo, utilice la siguiente muestra de código cuando el trabajo se vaya a imprimir en papel Carta EE. UU.:

```
%%BoundingBox: 0 0 612 792
```

### Sintaxis

```
%%BoundingBox: llx lly urx ury
```

### Donde:

<b>llx lly</b>	son las coordenadas en puntos, en función del origen del segmento, de la esquina inferior izquierda del área de imagen.
<b>urx ury</b>	son las coordenadas en puntos, en función del origen del segmento, de la esquina superior derecha del área de imagen.

### Ejemplos

Este ejemplo muestra un segmento con dos círculos concéntricos con la correspondiente instrucción %%BoundingBox.

```
%%BoundingBox: -100 -100 100 100
```

```
POINT SETUNIT
```

```
-100 100 200 200 LT_S1 100 DRAWBR
```

```
-50 50 100 100 OW_S1 50 DRAWBR
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**CACHE, SCALL**

% %EOD\_XGF

% %EOD\_XGF es un marcador de fin de datos que finaliza los comandos MAKEVMFILE, MAKEVMFORM o () RUN precedentes.

#### Sintaxis

% %EOD\_XGF

#### Modos

Este comando es aplicable en todos los modos.

#### Comandos relacionados

MAKEVMFILE, RUN



## % %EOF

El comando % %EOF es una instrucción de final de archivo de la convención de estructura de documentos (DSC, por sus siglas en inglés). En modo nativo, VI Compose no realiza ninguna acción específica en este. No obstante, se recomienda seguir las especificaciones de DSC.

En el modo de línea, % %EOF hace que VI Compose realice una eyección de página y un RESET para salir del modo de línea y reanudar el modo nativo. VI Compose actúa de la misma manera cuando se llega al final del archivo físico sin encontrar ningún % %EOF. Sin embargo, se recomienda incluir una instrucción % %EOF al final de los datos del modo de línea para realizar una salida correcta de este. Para obtener más información, consulte *Flujos de datos de VIPP®* en la *Guía del usuario de FreeFlow VI Compose*.

### Sintaxis

% %EOF

Instrucción DSC de fin de archivo.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- %!
- RESET
- STARTDBM
- STARTLM

## % %PagesPerBooklet

% %PagesPerBooklet indica el número de páginas por folleto para los trabajos de folletos con una cantidad fija de páginas. Coloque el marcador al principio de un trabajo VIPP® para controladores de la impresora que lo necesitan para gestionar correctamente los folletos del trabajo.

Sintaxis

`%%PagesPerBooklet: Number`

**Donde:**

**número** es el número de páginas por folleto.

**Modos**

Este comando es aplicable en todos los modos.

**Comandos relacionados**

**STARTBOOKLET, ENDBOOKLET**

## %%XGF

%%XGF identifica un registro de prefijo de modo nativo, NMP, en el modo de línea de manera prefijada. Los comandos NMP solo se aplican a impresoras de línea o a registros prefijados.

Los registros NMP permiten que los comandos en modo nativo se integren en un flujo de datos del modo de línea para cambiar dinámicamente el diseño en función de la página. Es preciso colocar un NMP (%%XGF) en la primera posición del registro, o en la segunda posición si se activa el procesamiento PCC. Para obtener más información, consulte *Flujos de datos de VIPP®* en la *Guía del usuario de FreeFlow VI Compose*.

%%XGF es el valor predeterminado definido en `/usr/xgf/src/xgf.def`. Cambie el valor predeterminado mediante la edición de este archivo o mediante el uso del comando SETNMP en el archivo JDT.

### Sintaxis

%%XGF comando de modo nativo

### Ejemplos

```
%! (p1.jdt) STARTLM ..... última línea de una página <Form Feed>%%
```



Nota: El uso de registros NMP puede implicar que la producción y la presentación de los datos dejen de ser independientes.



Consejo: Los registros NMP también pueden utilizarse para incorporar comentarios o información en los datos del modo de línea. Por ejemplo, no se imprimen los comentarios contenidos en este ejemplo.

```
%%XGF % comment
```

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

NMP\_off, SETNMP

`$$name.`

La cadena de sustitución de variable `$Nombre` consta de un nombre de variable encapsulado entre `$$` y `.` Un nombre de variable es una variable integrada o una variable configurada por SETVAR, o a través del procesamiento de una base de datos o archivo XML. Un nombre de variable definido en un archivo de base de datos puede incluir espacios.

Para obtener más información, consulte la descripción de la función de transformación **VSUB** y *Flujos de datos de VIPP®* en la *Guía del usuario de FreeFlow VI Compose*.

## [=name=]

La cadena de sustitución de variable [=name=] consta de un nombre de variable encapsulado entre [= and =]. Un nombre de variable es una variable integrada o una variable configurada a través de SETVAR o del procesamiento de una base de datos o un archivo XML. Para obtener más información, consulte *Flujos de datos de VIPP®* en la *Guía del usuario de FreeFlow VI Compose*. Un nombre de variable definido en un archivo de base de datos puede incluir espacios.

Para obtener más información, consulte la descripción de la función de transformación VSUB y *Flujos de datos de VIPP®* en la *Guía del usuario de FreeFlow VI Compose*.

## BBOX

El marcador BBOX se utiliza para proporcionar información del cuadro delimitador a un segmento en línea. Se coloca al principio de un segmento en línea cuando se utiliza el parámetro alineal.

### Sintaxis

```
{ llx lly urx ury } BBOX
```

### Donde:

**llx lly urx ury** son las coordenadas en puntos, en función del origen del segmento, si se usan las coordenadas x e y del lado inferior izquierdo (ll), y las coordenadas x e y del lado superior derecho (ur) del área de la imagen.

### Ejemplos

```
DOT3 SETUNIT 1200 2800 MOVETO { { 0 0 240 219 } BBOX 15 15 MOVETO (VIPPin.jpg) 1 0 10
```

## EXPAND

El marcador EXPAND se utiliza para facilitar a VIPP® Normalization el procesamiento de trabajos con segmentos VIPP®. Si EXPAND se coloca al principio de un segmento VIPP®, se sustituye una llamada al segmento en el archivo Normalized con un segmento en línea que contiene solo los comandos de marcaje útiles para la llamada.

### Sintaxis

#### EXPAND

Si el segmento tiene diversas variables externas y procesos lógicos anidados complejos, es más eficaz expandir solo el código relevante de cada llamada, en vez de realizar una llamada al segmento externo precedida por todas las definiciones externas de variables.

## EXTVAR

El marcador EXTVAR se utiliza para facilitar a VIPP® Normalization el procesamiento de trabajos con segmentos VIPP®. Puede colocar EXTVAR al principio de un segmento VIPP® para declarar la lista de variables externas.

### Sintaxis

```
{ /VARname1 /VARname2 . . . /VARnameN } EXTVAR
```

De manera predefinida, Normalization examina el código del segmento para determinar la lista de variables externas. Si el código es complejo, este método puede no ser preciso. EXTVAR proporciona una manera precisa de declarar de forma explícita la lista de variables externas.



## Funciones de transformación

Este capítulo incluye:

2OF5 .....	491
64TO256 .....	492
BIDIRECCIONAL .....	493
BSTRIP .....	495
BTRIM .....	496
CASELOW .....	497
CASETI .....	498
CASEUP .....	499
CODE39 .....	500
CODE128 y EAN128 .....	501
CS .....	504
DÍAS .....	505
EAN13 y EAN8 .....	506
F2S .....	508
FORMAT .....	509
GETINTV .....	511
HMS .....	513
NOHYPHEN .....	514
POSTJPN .....	515
POSTNET .....	517
QSTRIP .....	518
REEMPLAZAR .....	519
REDONDEAR .....	520
SUBSTFONT .....	521
TRÍO .....	522
UPCA .....	524
UTF8TOLOC .....	525
VSUB .....	526
VSUB2 .....	528
VSUB3 .....	529
VSUB4 .....	530

Una función de transformación de VIPP® es una secuencia formada por uno o varios operandos y una palabra clave de función de transformación de VIPP®. Una función de transformación VIPP® efectúa transformaciones en uno de los operandos y sustituye toda la secuencia por el operando transformado. La secuencia puede sustituir a cualquier operando de cualquier comando o función de transformación siempre que el valor del resultado sea apropiado para sustituir a ese operando. Esta acción permite combinar funciones de transformación.

Las funciones de transformación, excepto F2S y SUBSTFONT, pueden utilizarse en el procedimiento de alineación de una entrada RPE. Para obtener más información, consulte [Información de comando RPE](#) y las descripciones de [FROMLINE](#) y [RPEKEY](#).

## 20F5

**20F5** cambia una cadena para la impresión de una secuencia de código de barras con intercalación 2 de 5. Inserte **20F5** entre la cadena básica y el operador de impresión. Ha seleccionado previamente la fuente de código de barras mediante **SETFONT** o **INDEXFONT**.

### Sintaxis

```
(string) 20F5 SHx
```

### Donde:

**(cadena)** es la cadena numérica que se formateará.

**SHx** es uno de los comandos de visualización válidos



### Nota:

- El código de barras **20F5** no acepta caracteres alfanuméricos ni espacios, y produce un error si uno de estos caracteres se procesa en la cadena de entrada.
- Utilice solamente los comandos de tipo SH básicos al imprimir un código de barras. Para evitar la interacción no deseada entre los valores del código de barras, las transformaciones del código de barras y los comandos **DRAW**, utilice solamente los comandos de tipo SH básicos al imprimir un código de barras. El uso de **SHP** o **SHMF** puede conllevar una interacción inesperada si hay caracteres en el código de barras que representan el parámetro del atributo **SETFTSW** de los comandos **SHP** o **SHMF**.

### Ejemplos

```
(1340224715) 20F5 SHL
```

El ejemplo proporciona este resultado:

```
!0K9R2"
```

La fuente del código de barras correspondiente debe coincidir con una determinada asignación de caracteres. Para descargar fuentes, consulte [Descarga de fuentes](#).

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- [CODE39](#)
- [CODE128 / EAN128](#)
- [INDEXFONT](#)
- [SETFONT](#)
- [SHX](#)
- [UPCA](#)
- [POSTNET](#)

## 64TO256

**64TO256** es una función de transformación que convierte una cadena con codificación Base64 en una cadena binaria. Tiene como objetivo proporcionar una manera de colocar datos binarios (como imágenes pequeñas) en archivos de datos para, a continuación, imprimirlos con el comando **ICALL** o **SCALL**.

Ejemplo de uso:

Si se asume que `ImgString` es un campo de base de datos con un archivo de imagen convertido a Base64, la imagen se puede imprimir en el DBM mediante la sintaxis siguiente:

```
/VARImage [ ImgString 64TO256 ] XGFRESDEF x y MOVETO (VARImage) 1 0 ICALL
```



Nota: Esta función no puede procesar cadenas de más de 65,535 caracteres. Para procesar imágenes de un tamaño superior a 65 000, divida las imágenes en fragmentos de igual tamaño, excepto el último fragmento, y a continuación coloque los fragmentos en varios campos. Cada fragmento puede llamarse de forma secuencial en la instrucción **XGFRESDEF**.

### Ejemplos

```
/VARImage [ VARchk1 64TO256 VARchk2 64TO256 ... ] XGFRESDEF
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**XGFRESDEF**, **ICALL**, **SCALL**

## BIDIRECCIONAL

**BIDI** procesa una cadena de datos bidireccional para la impresión:

- El flujo de caracteres se invierte para la impresión de derecha a izquierda.
- El análisis de contexto y la sustitución de glifos se realizan según la colocación del carácter en la palabra: al principio, en el medio, al final, aislado.
- La unión se admite a través de la identificación y la sustitución de dos o tres dígitos especificados.
- Se proporciona un modificador opcional para los caracteres de izquierda a derecha.
- Se proporciona un modificador opcional a otras fuentes para dígitos europeos o hindi.

Este comando ha sustituido al comando **ARABIC**.

### Sintaxis

```
(Bi-directional data string) opt BIDI (Print-ready string)
```

Donde:

<b>(Cadena de datos bidireccional)</b>	es la cadena de entrada con los datos bidireccionales almacenados de izquierda a derecha y que aún no se han procesado para la impresión
<b>opt</b>	es uno de los siguientes, y puede omitirse: <ul style="list-style-type: none"> <li>0 función base; valor predeterminado si se omite</li> <li>1 cambiar a una fuente distinta para caracteres de izquierda a derecha</li> <li>2 sustituir dígitos europeos por dígitos hindi</li> </ul>
<b>(Cadena preparada para imprimir)</b>	es la cadena es entregada por la función; la cadena está a punto de ser procesada por un comando de impresión SHx, utilizando la fuente correspondiente.

### Ejemplos

```
(Bi-directional data string) BIDI SHR
```

La función de transformación **BIDI** se configura mediante el comando **SETBIDI**. Una configuración predeterminada se codifica al final del archivo de configuración bidireccional en `xf/src/arb.def`.

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **SETBIDI**
- **BEGINARBT**
- **ENDARBT**
- **FCALL**

- MOVETO
- SCALL
- ENDARBM

## BSTRIP

Puede configurar **BSTRIP** para eliminar o no los espacios en blanco adicionales que se encuentran delante o detrás de una cadena.

### Sintaxis

```
(string) option BSTRIP
(string) BSTRIP
```

### Donde:

Opción	es una de:
0	no eliminar nada
1	eliminar los espacios en blanco de delante
2	eliminar los espacios en blanco de detrás
3	quitar espacios en blanco iniciales y finales; valor predeterminado si se omite

### Ejemplos

Estas son instrucciones equivalentes:

```
( Text with extra blanks ) BSTRIP SHL
(Text with extra blanks) SHL
```

Estas son instrucciones equivalentes:

```
( Text with leading blanks) 1 BSTRIP SHL
(Text with leading blanks) SHL
```

Estas son instrucciones equivalentes:

```
(Text with trailing blanks ) 2 BSTRIP SHL
(Text with trailing blanks) SHL
```

Estas son instrucciones equivalentes:

```
( Text with extra blanks ) 0 BSTRIP SHL
( Text with extra blanks ) SHL
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**BSTRIP\_off**

## BTRIM

**BTRIM** es una función de transformación que elimina espacios en blanco iniciales, posteriores y duplicados de una cadena.

### Sintaxis

(string) BTRIM

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

BSTRIP, BCOUNT



## CASELOW

**CASELOW** cambia todas las letras en una cadena a minúsculas.

**CASELOW** es compatible con el conjunto de caracteres internacionales. La asignación se basa en el primer carácter de los nombres de carácter de fuentes activas. La fuente activa es la que se seleccionó por última vez antes de la ejecución de la función. Antes de codificar **CASELOW**, si la fuente activa no dispone de nombres de carácter apropiados para la asignación de mayúsculas o minúsculas, por ejemplo: /a/A, /acute /Acute, seleccione una fuente apropiada mediante **SETFONT** o un índice de fuentes.

### Sintaxis

```
string CASELOW
```

### Ejemplos

Este ejemplo imprime john r. martin:

```
(John R. Martin) CASELOW SHL
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**CASEUP**, **CASETI**

## CASETI

**CASETI** cambia todas las letras en una cadena al tipo correcto de letra (mayúsculas y minúsculas) para un título. Todas las letras se convierten en minúsculas, con la excepción de la primera letra de la cadena, y de cualquier letra precedida por un espacio, un guión, comillas simples o dobles, o una barra.

**CASETI** es compatible con el conjunto de caracteres internacionales. La asignación se basa en el primer carácter de los nombres de carácter de fuentes activas. La fuente activa es la fuente que se seleccionó por última vez antes de la ejecución de la función, lo que permite seleccionar la fuente apropiada, mediante SETFONT o un índice de fuentes antes de codificar CASEUP, si la fuente activa no dispone de nombres de carácter apropiados para la asignación de mayúsculas o minúsculas. Por ejemplo, /a/A, /acute /Aacute, etc.).

### Sintaxis

```
string CASETI
```

### Ejemplos

Este ejemplo imprime John R. Martin.

```
(JOHN R. MARTIN) CASETI SHL
```

Este ejemplo imprime Scarlett O'Hara

```
(SCARLETT O' HARA) CASETI SH
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- CASELOW
- CASEUP
- SHL y SH
- SHx

## CASEUP

**CASEUP** cambia todas las letras en una cadena a mayúsculas.

**CASEUP** es compatible con el conjunto de caracteres internacionales. La asignación se basa en el primer carácter de los nombres de carácter de fuentes activas. La fuente activa es la fuente que se seleccionó por última vez antes de la ejecución de la función, lo que permite seleccionar la fuente apropiada mediante **SETFONT** o un índice de fuentes antes de codificar **CASEUP** si la fuente activa no dispone de nombre de carácter apropiados para la asignación de mayúsculas o minúsculas. Por ejemplo, /a/A, /acute, /Acute, etc.

### Sintaxis

```
string CASEUP
```

### Ejemplos

Este ejemplo imprime JOHN MARTIN.

```
(John Martin) CASEUP SHL
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- CASELOW
- CASETI
- SHL y SH
- SHx

## CODE39

**CODE39** cambia el formato de una cadena para la impresión de una secuencia de código de barras de código 39. **CODE39** se inserta entre la cadena básica y el operador de impresión. La fuente de código de barras se ha seleccionado previamente mediante **SETFONT** o **INDEXFONT**.

### Sintaxis

```
(string) CODE39 SHx
```

### Donde:

(cadena) es una cadena alfanumérica para formatear

SHx es un comando Show válido.



Nota: Para evitar la interacción no deseada entre los valores del código de barras, las transformaciones del código de barras y los comandos **DRAW**, utilice solamente los comandos de tipo **SH** básicos al imprimir un código de barras. El uso de **SHP** o **SHMF** puede conllevar una interacción inesperada si hay caracteres en el código de barras que representan el cambio del atributo **SETFTSW** a los comandos **SHP** o **SHMF**.

### Ejemplos

Este ejemplo muestra la sintaxis y el resultado impreso. Dispone de la fuente de código de barras para imprimir el código de barras real.

```
(800273400024) CODE39 SHL
*800273400024*
```

Este comando también se utiliza en las entradas de RPE. Para obtener más información, consulte [FROMLINE RPEKEY](#).

La fuente del código de barras correspondiente debe coincidir con una determinada asignación de caracteres. Para descargar fuentes, consulte [Descarga de fuentes](#).

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- [2OF5](#)
- [EAN13/EAN8](#)
- [INDEXFONT](#)
- [SETFONT](#)
- [SHx](#)
- [UPCA](#)
- [POSTNET](#)

## CODE128 y EAN128

**CODE128** y **EAN128** cambian el formato de las cadenas para imprimir secuencias de códigos de barras CODE 128 o EAN 128. **CODE128** o **EAN128** se inserta entre la cadena básica y el operador de impresión. La fuente de código de barras se ha seleccionado previamente mediante **SETFONT** o **INDEXFONT**.

### Sintaxis

```
(string) CODE128 SHx
```

```
(string) integer CODE128 SHx
```

```
(string) EAN128 SHx
```

```
[ (fix_length_string) (var_length_string) <F1> ... ] EAN128 SHx
```

### Donde:

<b>cadena</b>	es una cadena alfanumérica para dar formato. La longitud de la cadena de entrada es variable. La cadena de entrada puede consistir en la serie completa de ASCII (0-127). Se admiten los conjuntos de códigos A, B y C. Los caracteres de inicio y final no se pueden incluir en la entrada.
<b>SHx</b>	es un comando Show válido.
<b>entero</b>	es un operando opcional. Los valores pueden ser:  <b>0</b> Esta es la opción predeterminada. CODE128 utiliza el conjunto de códigos C cuando sea posible (comprime una secuencia de dígitos). Comprime los caracteres numéricos cuando detecte cuatro o más dígitos adyacentes  <b>1</b> CODE128 NO comprime ninguna secuencia de dígitos (siempre usa el conjunto de códigos A o B)
<b>(fix_length_string)</b>	es un campo de longitud fija EAN (el ID del campo forma parte de la cadena).
<b>(var_length_string)&lt;F1&gt;</b>	es un campo de longitud variable EAN (el ID del campo forma parte de la cadena). La ID del campo se incluye en los campos.

La cadena de salida consiste en esta información: car\_ini + sec\_code128 + dig\_compr + car\_fin, en la que:

- car\_ini es un carácter de inicio del conjunto de códigos (A, B o C) correspondiente.
- code128\_sequence es una cadena de entrada comprimida si es posible, si partes de la cadena de entrada se comprimieron, la salida incluye caracteres modificadores.
- dig\_compr es un carácter de suma de comprobación.
- car\_final es un carácter de final.

Se admiten los conjuntos de códigos A, B y C. La codificación de la cadena de salida asigna la codificación Xerox (x20-7F, A1-AB) o BearRock, x20-7F, 95-DF. El comando realiza una detección automática para seleccionar la codificación correspondiente (la selección del conjunto de códigos A o B se realiza a partir de los datos; el conjunto de códigos C es la única opción seleccionable por el usuario).

Estas normas generales se usan para determinar la selección del conjunto de códigos:

- El conjunto de códigos A se utiliza para codificar todos los caracteres alfanuméricos en mayúsculas estándares, además de los caracteres de control.
- El conjunto de códigos B se utiliza para codificar todos los caracteres alfanuméricos en mayúsculas estándares, además de los caracteres alfabéticos en minúsculas.
- El conjunto de códigos C se usa para codificar parejas de caracteres numéricos (00-99) como un carácter, lo que hace que el código de barras sea mucho más compacto si contiene principalmente caracteres numéricos.
- Si el conjunto de códigos A o B se puede usar (gracias a que los datos solo contienen caracteres alfanuméricos en mayúsculas), se usa el conjunto de códigos B CODE128.

Este comando también puede utilizarse en las entradas de RPE. Para obtener más información, consulte [RPE command information](#) and [FROMLINE](#) y las descripciones de [RPEKEY](#).

La fuente del código de barras correspondiente debe coincidir con una determinada asignación de caracteres. Para descargar fuentes, consulte [Descarga de fuentes](#).



Nota: Para evitar la interacción no deseada entre los valores del código de barras, las transformaciones del código de barras y los comandos **DRAW**, al imprimir un código de barras, utilice solamente los comandos de tipo **SH** básicos. El uso de **SHP** o **SHMF** puede conllevar una interacción inesperada si hay caracteres en el código de barras que representan el parámetro del atributo **SETFTSW** de los comandos **SHP** o **SHMF**.

## Ejemplos

En estos ejemplos básicos de CODE128 y EAN128 se asume que se dispone de un campo llamado FIELD1:

```
/MB034 12 SETFONT
FIELD1 CODE128 SHL
FIELD1 EAN128 SHL
```

Este es un ejemplo de JDT:

```
/F128 /MB034 12 INDEXFONT
...
3 FROMLINE
[{BSTRIP 1 CODE128 SH} 0 300 0 1000 150 60 15 /F128 BLACK]
4 FROMLINE [...]
```

En el ejemplo anterior, una parte del registro (15 caracteres en la posición 60) se extrae de la línea 3. Después de la eliminación de los espacios en blanco, se imprime como un código de barras 128 en la posición “300 1000” con la fuente F128.

## Información y ejemplos de EAN128

Para imprimir un símbolo EAN128, la aplicación debe proporcionar como mínimo un identificador (AI) y una sección de datos (DATA). Si los tiene como un solo campo (campo1 = AI+DATA) utilice este código:

```
Field1 EAN128 SH
```

Si se usan dos campos para AI y DATA, use una de estas muestras de código:

```
( $$Field2. $$Field3. ) VSUB EAN128 SH
[ Field2 Field3 ] EAN128 SH
```

Para concatenar varios campos AI+DATA en un solo símbolo, tenga en cuenta que DATA puede tener datos de longitud fija o variable.

Use estos ejemplos si DATA tiene una longitud fija para concatenar dos campos como, por ejemplo, FieldF1=AI1+DATA1 y FieldF2=AI2+DATA2 mediante la sintaxis de VSUB, o una matriz:

```
( $$FieldF1. $$FieldF2. ) VSUB EAN128 SH
[ FieldF1 FieldF2 ] EAN128 SH
```

Use estas muestras si DATA tiene una longitud variable para concatenar 3 campos incluido el campo de longitud variable FieldV3=AI3+DATA3 mediante la sintaxis de matriz:

```
[ FieldF1 FieldV3 <F1> FieldF2 ] EAN128 SH
[ FieldF1 FieldF2 FieldV3 ] EAN128 SH
```

Si no es el campo final, el campo de longitud variable va seguido del código function1, que debe representarse mediante <F1>. Aunque <F1> es la cadena de bytes con el valor 0xF1, se convierte en el código apropiado por la función EAN128.

Este ejemplo ilustra una combinación de varios AI y DATA separados en varios campos, con AI codificado de forma integrada en el código VIPP®:

```
[ (01) Data01 (21) Data21 <F1> (10) Data10 ] EAN128 SH
```

## Modos

Este comando es aplicable en todos los modos.

## Comandos relacionados

- INDEXFONT
- SETFONT
- SHx
- UPCA
- POSTNET

## CS

Utilice CS para concatenar dos cadenas en una.

### Sintaxis

```
(string1) (string2) CS
```

### Ejemplo:

Este ejemplo imprime Dear Mr. Martin.

```
(Dear Mr.) (Martin) CS SHL
```

### Modo

Este comando es aplicable en todos los modos.

### Comandos relacionados

**VSUB, SETVAR**



## DÍAS

La función de transformación integrada produce un entero que representa el número de días de la fecha suministrada desde el 1 de enero de 1970. Se puede usar conjuntamente con **GETDATE**, **SHIFTDATE** y **SETDATE** para calcular distintos valores de variación de la fecha.

### Sintaxis

```
[ YYYY MO DD ] DAYS
```

### Donde:

<b>AAAA</b>	es el año (> 1970). Puede ser un número entero o una cadena numérica.
<b>MO</b>	es el mes (1-12). Puede ser un número entero o una cadena numérica.
<b>DD</b>	es el día. Puede ser un número entero o una cadena numérica.

### Ejemplo:

Asume que Year, Month y Day son campos de un campo de DBF:

```
[ Year Month Day ] SETDATE
/VARdate1 ($$D_DWL. $$D_MO./$$D_DD./$$D_YYYY.) VSUB SETVAR
/VARstart [ Year Month Day ] DAYS SETVAR
VARstart'+55 SETDATE
/VARdate2 ($$D_DWL. $$D_MO./$$D_DD./$$D_YYYY.) VSUB SETVAR
GETDATE
(From $$VARdate1. to $$VARdate2. you will be given the opportunity to visit our new shopping
center and purchase any article with a 50% discount.) VSUB 0 SHP
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**, **SHIFTDATE**, **SETDATE**

## EAN13 y EAN8

**DRAWBC** con sus opciones **/EAN13** y **/EAN8** sustituye a los antiguos comandos **EAN13** y **EAN8**, que están ya obsoletos, aunque sigan siendo compatibles con versiones anteriores. **EAN13** y **EAN8** cambian el formato de una cadena para la impresión de una secuencia de códigos de barras EAN. **EAN13** y **EAN8** se insertan entre la cadena básica y el operador de impresión. La fuente correspondiente se ha seleccionado previamente mediante **SETFONT** o **INDEXFONT**.

### Sintaxis

(SXXXXXXXXXXXX) EAN13 SHx

(XXXXXXX) EAN8 SHx

### Donde:

(SXXXXXXXXXXXX)

es una cadena de 12 dígitos que consta de:

**S** el sistema numérico

**XXXXXX** la parte izquierda

**YYYYY** la parte derecha

(XXXXXXX)

es una cadena de siete dígitos que consta de:

**XXXX** la parte izquierda

**YYY** la parte derecha

**SHx**

es un comando Show válido.



Nota: Las funciones automáticamente calculan y agregan un dígito de verificación al final de la cadena devuelta. Se aceptan cadenas de entrada que finalizan con un dígito de comprobación calculado previamente de 13 dígitos o cadenas de 8 dígitos, pero el dígito de comprobación se ignora y se sustituye por el calculado por la función.



Nota: Para evitar la interacción no deseada entre los valores del código de barras, las transformaciones del código de barras y los comandos **DRAW**, utilice solamente los comandos de tipo **SH** básicos al imprimir un código de barras. El uso de **SHP** o **SHMF** puede conllevar una interacción inesperada si hay caracteres en el código de barras que representan el parámetro del atributo **SETFTSW** de los comandos **SHP** o **SHMF**.

### Ejemplos

Este ejemplo muestra la sintaxis EAN13 y el resultado impreso.

```
(800273400024) EAN13 SHL
8<ALCSOE=aaacec>
```

Este ejemplo muestra la sintaxis EAN8 y el resultado impreso.

```
(7616424) EAN8 SHL
<HGBG=ecei>
```

Utilice también estos comandos en el procedimiento de alineación de una entrada RPE. Para obtener más

información, consulte [Información de comando RPE](#) y las descripciones de [FROMLINE](#) y [RPEKEY](#).

El dígito de verificación se calcula a través de las funciones. La fuente del código de barras correspondiente debe coincidir con una determinada asignación de caracteres. Para descargar fuentes, consulte [Descarga de fuentes](#).

### Modos

Estos comandos son aplicables en todos los modos.

### Comandos relacionados

- [2OF5](#)
- [CODE39](#)
- [INDEXFONT](#)
- [SETFONT](#)
- [SHx](#)
- [UPCA](#)
- [POSTNET](#)

## F2S

Utilice F2S para almacenar un registro delimitado por campos en una tabla mediante el comando **ADD**.

### Sintaxis

```
(fd11:fd12:...:fdln) (delim) F2S (fd11) (fd12) ... (fdln)
(fd11:fd12:...:fdln) (delim) { transform code } F2S (fd11) (fd12) ... (fdln)
```

### Ejemplos

Este ejemplo de F2S produce (John) (Smith) (Seattle).

```
(John:Smith:Seattle) (:) F2S
```

En este ejemplo, con ADD, se da por supuesto que RECORD es una cadena de datos delimitados por campos:

```
/VARtable [[ RECORD (:) F2S ]] ADD
```

En este ejemplo todos los campos se convierten en mayúsculas:

```
[ (aa:Bc:cC) (:) { CASEUP } F2S ]
```

Produce:

```
[ (AA) (BB) (CC) ]
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**ADD**, **GETITEM**, **FOREACH**

## FORMAT

Use **FORMAT** para dar formato a una cadena numérica para la impresión por parte de uno de los comandos **SHx**. **FORMAT** admite la encapsulación de números negativos entre paréntesis. A tal efecto, un par de paréntesis debe estar presente en la cadena de formato, y el parámetro **/FNSign** debe asignarse a los paréntesis de cierre (41).

### Sintaxis

```
(numeric data) (format) FORMAT SHx
```

```
(numeric data) (format) [ /param1 value1 /param2 value2 ... ] FORMAT SHx
```

### Donde:

**datos numéricos** es una cadena que contiene datos numéricos; se omitirán los caracteres no numéricos.

**formato** es una cadena con caracteres de metadatos que describe la salida deseada. También se pueden incluir caracteres estáticos en la cadena. Los metacaracteres se definen mediante determinados parámetros, y se pueden configurar de forma predefinida mediante el comando **SETPARAMS** o a través de su inclusión en la matriz opcional descrita más abajo.

Los parámetros de formato son:

- **/DecimalPoint** 4 delimitador decimal en datos numéricos (predeterminado: .)
- **/NSign** 45 signo negativo en datos numéricos (predeterminado: -)
- **/FDecimalPoint** 46 delimitador decimal en formato (predeterminado: .)
- **/FNSign** 45 signo negativo en formato (predeterminado: -)
- **/FPSign** 43 signo positivo en formato (predeterminado: +)
- **/FPunctuation** 44 delimitador de miles en formato (predeterminado: ,)
- **/FDigit** 35 marcador de posición de dígitos en formato (predeterminado: #)
- **/FLZDigit** 64 marcador de posición de dígitos en formato (predeterminado: @) que se sustituirá por espacios si el dígito es un cero inicial



Nota: El valor de cada parámetro es el valor decimal ASCII del carácter. Un valor nulo indica que el carácter correspondiente no está definido. Si se usa **FNSign** con datos numéricos positivos, genera un espacio. Si se usa **FPSign** con datos numéricos negativos, se sustituye por **FNSign**. En la cadena de salida, todos los caracteres de metadatos se sustituyen, eliminan o conservan según corresponda. Los caracteres de metadatos no se conservan. Los caracteres de metadatos se definen en `/usr/xgf/src/xgf.def` y varían en función de la opción de material de impresión predeterminada durante la instalación. Si el trabajo **VIPP®** se va a reproducir a través de diversas plataformas, se recomienda que los metacaracteres se definan en el trabajo.

**[ /paramx valuelx ... ]** La matriz de parámetros opcionales sigue los parámetros de formato que se configurarán temporalmente solo para esta instrucción, sin tener efecto en el resto del código.

### Ejemplos

Este ejemplo contiene un signo negativo (-) después del último carácter de número (#). Este ejemplo imprime

\$1,234.56-

(-00001234.56) (\$@,@@@,@@#.##-) FORMAT SHr

Estos son ejemplos de FORMAT con alineación en el punto decimal:

[ /FNSign 41 ] SETPARAMS

(123456.78) ((@@@,@@@,@@@,@@#.##) ) FORMAT 4 SHMF

imprime: 123.456,78

(-123456.78) ((@@@,@@@,@@@,@@#.##) ) FORMAT 4 SHMF

imprime: (123.456,78)

En este ejemplo:

(1234567890) (@@,@@@,@@#.##) [ /FDecimalPoint null ] FORMAT SHx

imprime: 12.345.678,90

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

SETPARAMS, SETVAR, SHx

## GETINTV

**GETINTV** extrae una cadena secundaria o un campo de una cadena.

### Sintaxis

```
(string) recpos length GETINTV
(string) field_nr (field_sep) GETINTV
```

### Donde:

<b>cadena</b>	es la cadena de origen de la extracción.
<b>recpos</b>	es la posición del primer carácter que se extraerá (empezando por cero).
<b>longitud</b>	es la longitud de la subcadena. Si la longitud es negativa, la extracción se realiza a partir del último carácter de derecha a izquierda, en vez de izquierda a derecha.
<b>field_nr</b>	es el número de campo empezando por 0.
<b>field_sep</b>	es la cadena del separador de campos. field_sep puede especificar otra cadena de separadores que la indicada por el comando <b>SETDBSEP</b> actual.

### Ejemplos

Para imprimir solo el nombre de la cadena Mr. John Martin, utilice esta opción.

```
(Mr. John Martin) 1 ( ) GETINTV SHL
```

Este ejemplo genera la cadena ABC de la cadena ABCDEF.

```
(ABCDEF) 0 3 GETINTV
```

Este ejemplo genera la cadena DEF de la cadena ABCDEF.

```
(ABCDEF) 0 -3 GETINTV
```

Este ejemplo selecciona el mes de la cadena 10/02/00 y, por lo tanto, imprime 10. Un comando **GETINTV** o **GETFIELD** anterior puede haberse usado para capturar este campo de una cadena o un campo mayores.

```
(10/02/00) 0 (/) GETINTV SHL
```

En este ejemplo la fecha se divide en el mes, el día y el año, y asigna las variables para usarlas posteriormente.

```
/VAR_month (10/02/00) 0 (/) GETINTV SETVAR
/VAR_day (10/02/00) 1 (/) GETINTV SETVAR
/VAR_year (10/02/00) 2 (/) GETINTV SETVAR
```

**GETINTV** también puede utilizarse en un DBM para procesar líneas con campos de longitud fija, normalmente datos en modo de línea. Este ejemplo muestra la creación de un archivo de base de datos utilizando los datos del modo de línea.

```
%!
(fixed.dbm) STARTDBM
MY_RECORD
John Mary Snow
```

```
Tim Victor Sand
%%EOF
```

En la DBM, los campos se separan y se asignan a las variables para usarse posteriormente mediante el código más abajo. El ejemplo muestra el campo de longitud FIXED, en el que los nombres figuran en la columna 0 con 6 bytes, el segundo nombre figura en la columna 6 con 11 bytes, y el último nombre está en la columna 17 con 8 bytes. Es importante alinear los datos en este tipo de orden fijo de posición.

```
/VARname MY_RECORD 0 6 GETINTV SETVAR
/VARname2 MY_RECORD 6 11 GETINTV SETVAR
/VARsurface MY_RECORD 17 8 GETINTV SETVAR
```

### Expresión de subcadena

Como alternativa a **GETINTV**, una expresión de subcadena se puede aplicar a una variable o a un nombre de campo con la sintaxis siguiente:

```
variable_name|recpos,length|
variable_name|field_nr,field_sep|
variable_name|field_nr,0xhexfield_sep|
```

### Donde

**recpos, length, field\_nr, field\_sep** Tienen el mismo sentido que en la sintaxis de **GETINTV**.

Una expresión de subcadena no debe contener ninguno de los caracteres siguientes:

espacio / ( ) < > { } [ ] %

Se puede usar 0xhexfield\_sep si el separador contiene cualquiera de los caracteres anteriores mediante la codificación del valor hexadecimal de la cadena precedida por la etiqueta 0x.

### Ejemplos:

```
/VAR1 (date: 07/12/2012) SETVAR
VAR1|8,4| SH % prints (/12/)
VAR1|1,:| SH % prints ( 07/12/2012)
VAR1|2,0x2F| SH % prints (2012)
([=VAR1|2,0x2F|=]:[=VAR1|1,0x2F|=]:[=VAR1|6,2|=]) VSUB SH % prints (2012:12:07)
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SETDBSEP, GETFIELD**



## HMS

Utilice **HMS** para convertir el tiempo medido en segundos al formato HH:MM:SS.

### Sintaxis

```
(integer) HMS  
integer HMS
```

### Ejemplos

Este ejemplo imprime 4456 como 1:14:16.

```
(4456) HMS SHL
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SHx**

## NOHYPHEN

**NOHYPHEN** es una función de transformación que evita la aparición de guiones en espacios del texto impreso con los comandos **SHP** o **SHp**. Esta opción está pensada para aplicarse en variables internas en texto **SHP**.

### Sintaxis

```
string NOHYPHEN
```

### Ejemplos

```
/VAR_ DATE DATE NOHYPHEN SETVAR  
(... will be delivered on $$VAR_ DATE. no later than 16:00 ...) VSUB SHP
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SHP and SHp, SHT and SHt**

## POSTJPN

**POSTJPN** cambia el formato de una cadena para la impresión de secuencias de códigos de barras de tipo postal del Japón. Este comando se introduce entre la cadena de entrada y un comando de impresión de VIPP®. Seleccione una fuente de código de barras Postal del Japón utilizando **SETFONT** o **INDEXFONT** antes de insertar este comando.

### Sintaxis

```
(string) POSTJPN SHx
```

### Donde:

**Cadena** es una cadena alfanumérica de un código postal seguida de un código de dirección. Se pueden usar los caracteres siguientes:

**0-9**

**guion**

**A-Z**

**a-z (permitido, pero se convierten en mayúsculas)**

Cualquier otro carácter produce un error VIPP®.

El código postal tiene 7 caracteres numéricos. El código de dirección puede ser alfanumérico y no tiene limitaciones de longitud, pero se trunca en 13 módulos de código de barras postal. Un carácter numérico o un guión es un módulo, un carácter alfabético son 2 módulos.

**SHx** es uno de los comandos de VIPP® para la impresión de texto

### Satisfacer las especificaciones del servicio de correos de Japón

En Japón, la cadena de entrada debe cumplir las especificaciones del servicio de correos de Japón; consulte [http:// www.post.japanpost.jp/](http://www.post.japanpost.jp/) (solo en japonés).

La precisión del código postal está determinada por la creación de la aplicación del flujo de datos.

La cadena de salida consiste en:

```
start_code + barcode modules + check_digit + stop_code
```

La fuente del código de barras asociado coincide con una determinada asignación de caracteres. Para adquirir fuentes del código de barras del servicio postal japonés, póngase en contacto con el representante de Fuji Xerox (las fuentes solo están disponible en Japón).

### Utilice solamente los comandos de tipo SH básicos al imprimir un código de barras

Para evitar la interacción no deseada entre los valores del código de barras, las transformaciones del código de barras y los comandos **DRAW**, utilice solamente los comandos de tipo **SH** básicos al imprimir un código de barras. El uso de **SHP** o **SHMF** puede conllevar una interacción inesperada si hay caracteres en el código de barras que representan el parámetro del atributo (SETFTSW) de los comandos **SHP** o **SHMF**.

### Ejemplos

```
(100000131-3-2-503SMITH) POSTJPN SH
```

### **Modos**

Este comando es aplicable en todos los modos.

### **Comandos relacionados**

SETFONT, INDEXFONT, SHx

## POSTNET

**POSTNET** cambia una cadena para la impresión de una secuencia de código de barras PostNet. Este comando se introduce entre la cadena de entrada y un comando de impresión de VIPP®. Ha seleccionado previamente una fuente de código de barras PostNet mediante **SETFONT** o **INDEXFONT**.

### Sintaxis

```
(string) POSTNET SHx
```

### Donde:

- cadena** son una cadena numérica de longitud 5, 9 u 11. Las longitudes de 10 y 12 elementos también son compatibles con un guión en la sexta posición.
- SHx** es uno de los comandos de VIPP® para la impresión de texto

### Satisfacer las especificaciones del servicio de correos de EE. UU.

En los EE. UU., la cadena de entrada debe satisfacer las especificaciones del servicio postal estadounidense en lo referente al código postal (Zip) Zip, Zip+4 o Zip+4+DPBC. Para obtener información detallada, consulte [www.usps.gov](http://www.usps.gov).

La precisión del código postal está determinada por la creación de la aplicación del flujo de datos.

### Utilice solamente los comandos de tipo SH básicos al imprimir un código de barras

Para evitar la interacción no deseada entre los valores del código de barras, las transformaciones del código de barras y los comandos **DRAW**, utilice solamente los comandos de tipo **SH** básicos al imprimir un código de barras. El uso de **SHP** o **SHMF** puede conllevar una interacción inesperada si hay caracteres en el código de barras que representan el parámetro del atributo (SETFTSW) de los comandos **SHP** o **SHMF**.

La cadena de salida consiste en:

- car\_trama + cad\_entrada + digito\_compr + car\_trama
- car\_trama puede estar codificado en la posición 42 (asterisco)

La fuente del código de barras asociado coincide con una determinada asignación de caracteres. Para descargar fuentes, consulte [Descarga de fuentes](#).

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

[SETFONT](#), [INDEXFONT](#), [SHx](#)

## QSTRIP

**QSTRIP** elimina el primer y el último carácter de una cadena. Se puede usar como una alternativa a **QSTRIP\_on** en el modo de base de datos solo si es preciso eliminar algunos campos. Este comando no tiene ningún efecto si la longitud de la cadena es inferior a dos caracteres.

### Sintaxis

```
(string) QSTRIP
```

### Ejemplos

Este ejemplo imprime John Martin sin comillas.

```
("John Martin") QSTRIP SHL
```

### Modos

Esta función es aplicable en todos los modos.

### Comandos relacionados

- **BSTRIP**
- **BSTRIP\_off**
- **GETINTV**
- **QSTRIP\_on**
- **SHx**

## REEMPLAZAR

**REPLACE** es una función de transformación que se puede usar para sustituir instancias de una subcadena con otra subcadena en una cadena de entrada definida.

### Sintaxis

```
(input string) (str1) (str2) REPLACE (output string)
```

### Donde:

<b>cadena de entrada</b>	es la cadena de entrada original
<b>str1</b>	la subcadena que se reemplazará si se encuentra
<b>str2</b>	es la nueva subcadena que reemplazará a str1
<b>cadena de salida</b>	la cadena es entregada por la función

### Ejemplos

```
(Hello:dear:friends) (:) (|) REPLACE SH
```

se imprimirán:

```
Hello|dear|friends
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

Ninguno

## REDONDEAR

Utilice este comando para realizar el redondeo de variables numéricas según el dígito más cercano en una posición determinada.

### Sintaxis

```
/VARname round_digit ROUND
```

### Donde:

**VARName** es el nombre de una variable previamente inicializada con **SETVAR**.

**round\_digit** es la posición de redondeo desde el punto decimal. Puede ser positiva o negativa.

### Ejemplos

```
/VAR1 (1234.5678) SETVAR  
/VAR1 3 ROUND % VAR1 es ahora: (0001234.568)  
/VAR1 2 ROUND % VAR1 es ahora: (0001234.57)  
/VAR1 0 ROUND % VAR1 es ahora: (0001235)  
/VAR1 -2 ROUND % VAR1 es ahora: (0001200)
```

### Modos

Esta variable es aplicable en todos los modos.

### Comandos relacionados

**ADD, SUB, DIV**



## SUBSTFONT

**SUBSTFONT** se utiliza para las sustituciones de fuentes. Utilice este comando en las listas de fuentes almacenadas en uno de los directorios referenciados por **SETEPATH**. Solo es efectivo en dispositivos de nivel 2 de PostScript.

### Sintaxis

```
/font1 /font2 SUBSTFONT
```

**SUBSTFONT** produce /font1 si esta fuente está disponible en el dispositivo. Si no está disponible, devuelve /font2.

Defina las sustituciones para evitar la sustitución predefinida (normalmente con Courier) o que se produzca un error si la fuente no está disponible en el dispositivo. font2 es lo más parecido posible a font1 para que la salida de VIPP® solo se vea ligeramente afectada.

### Ejemplos

En este ejemplo, Helvetica light se utiliza si está disponible, de lo contrario, se utiliza Helvetica.

```
/Helvetica-light /Helvetica SUBSTFONT
```

### Modos

Esta función es aplicable en todos los modos.

### Comandos relacionados

**SETENCODING**

## TRÍO

**TRIO** es una función de transformación para corregir problemas de compatibilidad entre las fuentes **LCDS** y **PS/PCF** en el mercado chino. Solo es compatible con flujos de datos **EBCDIC**.

La cadena de entrada debe contener los bytes 0x0E y 0x0F que representan los cambios entre texto en chino y occidental, por ejemplo, inglés, para el cual **TRIO** agrega espacios **EBCDIC** (0x40) y, posiblemente, modificadores **INDEXFONT**.

**TRIO** está pensado para su uso en combinación con **SHMF**.

### Sintaxis

```
(EBCDIC string) TRIO
```

### Ejemplos

Este ejemplo de procesamiento:

```
(-eng-<0E>-chn-<0F>-eng-) TRIO
```

... produce:

```
(-eng-<40><0E>-chn-<40>-eng-)
```

... con una fuente interna en inglés.

Alternativamente, este ejemplo se puede usar con fuentes externas inglesas (asumiendo que el parámetro de la fuente se haya configurado como (//) en SETFTSW).

```
(-eng-<40>//FI0<0E>-chn-//FIx<40>-eng-)
```

Utilice las siguientes convenciones de las fuentes definidas por **INDEXFONT**:

Se asigna un índice a las fuentes chinas que terminan en 0 (cero). Se puede asociar en la misma cadena con otras fuentes inglesas con el mismo índice pero sin finalizar con 0. Por consiguiente:

- Si la **INDEXFONT** actual termina en cero, se utilizará la fuente interna en inglés con la fuente china.
- Si la **INDEXFONT** actual no termina en cero, se usará una fuente inglesa y el índice coincidente acabado en cero como fuente china.

```
(//) 3 SETFTSW
/F00 /MSung5550-Light 12 INDEXFONT % Chinese font
/F01 /NCR 10 INDEXFONT             % Western/English font
1 SETRPEPREFIX
10 BEGINRPE
...
/4 RPEKEY                          % line 4 (font index 4)
[{TRIO 0 SHMF} ..... /F00 BLACK] % use only F00
/5 RPEKEY                          % line 5 (font index 5)
[{TRIO 0 SHMF} ..... /F01 BLACK] % start with F00 and use F01 for English
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

SHMF, SHMf, and SHmf, INDEXFONT, SETFTSW

## UPCA

**UPCA** cambia el formato de una cadena para su uso cuando se imprime una secuencia de código de barras UPC versión A. Inserte **UPCA** entre la cadena básica y el operador de impresión. Se requiere una fuente de código de barras UPCA. Seleccione un código de barras UPCA mediante **SETFONT** o **INDEXFONT**.

### Sintaxis

```
(SXXXXXXXXYYY) UPCA SHx
```

### Donde:

(SXXXXXXXXYYY)

es una cadena de 11 dígitos que consta de estos elementos:

- S representa el sistema numérico
- XXXXX especifica el número de proveedor
- YYYYY especifica el número de producto

SHx

representa un comando de impresión válido

El dígito de verificación se calcula a través de la función. La fuente del código de barras correspondiente debe coincidir con una determinada asignación de caracteres. Para descargar fuentes, consulte [Descarga de fuentes](#).



Nota: Para evitar la interacción no deseada entre los valores del código de barras, las transformaciones del código de barras y los comandos **DRAW**, utilice solamente los comandos de tipo **SH** básicos al imprimir un código de barras. El uso de **SHP** o **SHMF** puede conllevar una interacción inesperada si hay caracteres en el código de barras que representan el parámetro del atributo **SETFTSW** de los comandos **SHP** o **SHMF**.

### Ejemplos

Estos son ejemplos de la estructura de comandos y la salida resultante:

```
(08978204466) UPCA SHL
```

```
0<kIjHIC=aeegw>2
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- [20F5](#)
- [CODE39](#)
- [EAN13/EAN8](#)
- [INDEXFONT](#)
- [SETFONT](#)
- [SHx](#)
- [POSTNET](#)

## UTF8TOLOC

**UTF8TOLOC** es una función de transformación que se utiliza para convertir una cadena de UTF8 a la codificación local.

### Sintaxis

```
(string) local_code UTF8TOLOC
```

Donde

El código `local` es uno de las siguientes:

- |    |   |
|----|---|
| 1  | ISO-8859-1 o Windows-1252 (Europeo occidental)    |
| 2  | ISO-8859-2 (Europeo Central)                      |
| 3  | Windows-1250 (centroeuropeo)                      |
| 4  | ISO-8859-9 (turco)                                |
| 5  | Windows-1251 (cirílico)                           |
| 6  | Windows-1258 (vietnamita)                         |
| 7  | ISO-8859-11 o Windows-CP874 o TIS-620 (Tailandés) |
| 8  | Windows-CP866 (cirílico)                          |
| 9  | ISO-8859-15 (Latin-9)                             |
| 10 | Mac OS Roman                                      |
| 11 | Windows-1256 (árabe)                              |
| 12 | Windows-1255 (hebreo)                             |

### Ejemplo:

```
ImageName 1 UTF8TOLOC ICALL
```

`ImageName` es el nombre del campo de una imagen variable en un archivo de base de datos con codificación UTF8.

### Modo

Este comando es aplicable en todos los modos.

### Comandos relacionados

[/LocalToUTF8](#)

## VSUB

**VSUB** sustituye las referencias del texto o las variables con sus contenidos en una cadena.

Use este comando con los comandos que usan una cadena como operando como, por ejemplo, **SHx**, **SETFORM**, **SETMEDIA**, etc., en DBM, un formulario VIPP® o un trabajo en modo nativo.

Una referencia de variable consiste en un nombre de variable encapsulado entre \$\$ y ., o entre [= y =]. Un nombre de variable es una variable integrada, o bien una variable configurada por **SETVAR**, o a través del procesamiento de una base de datos o archivo XML. Para obtener más información, consulte Flujos de datos de VIPP® en la *Guía del usuario de FreeFlow VI Compose*. Un nombre de variable definido en un archivo de base de datos puede incluir espacios.

Una referencia al archivo de texto está formada por un nombre de archivo de texto encapsulado entre [= y =].

Si una referencia no es una variable, se busca como nombre de archivo en los directorios definidos por los comandos de modo de proyecto **SETMPATH** o **SETPPATH**. Los archivos de texto referenciados de esta manera deben ser de texto normal y componerse de parámetros de atributos VIPP® válidos eventualmente, según lo definido por los comandos INDEXxxx y los marcadores de posición variables \$\$ y ., o [= y =]. El contenido del archivo de texto se inserta en la cadena entregada sin cambios, excepto para la sustitución de referencias de variables. Un uso habitual de **VSUB** con referencias a archivos de texto consiste en importar bloques de texto de una recopilación de archivos de texto externos e imprimirlos en la parte inferior de la página mediante **SETLKF** y **SHP**.

La longitud total de la cadena producida por **VSUB** no puede exceder los 65 535 caracteres.

### Sintaxis

```
(printable data with variable and/or text file references) VSUB
(string to merge) (string holding $$.) VSUB
```

### Ejemplos

Cuando **FNAME** and **LNAME** contienen John y Smith, los ejemplos siguientes producen Dear John Smith.

```
(Dear [=FNAME=] [=LNAME=],) VSUB 0 SHP
(Dear $$FNAME. $$LNAME.,) VSUB 0 SHP
(John Smith) (Dear $$.,) VSUB 0 SHP
```

En este ejemplo, **TITLE** and **NAME** han sido asignados a Mr. y Martin por el registro actual del archivo de base de datos, y esta secuencia se imprime como: As you know, Mr. Martin, you have won, ...

```
(As you know, $$TITLE. $$NAME., you have won, ...) VSUB SHL
```

Use **VSUB** en un procedimiento de alineación de entrada RPE para combinar el campo RPE en una cadena a través de la omisión del nombre de la variable entre \$\$ y .

En el ejemplo siguiente se utiliza **VSUB** en un procedimiento de alineación RPE para combinar un campo en una cadena e imprimirlo.

```
[ { (Amount: -$$.-) VSUB SH } ..... 25 8 /F1 BLACK ]
```

Si el campo especificado por el inicio de la posición 25 y la longitud de 8 caracteres en el ejemplo anterior contiene 1,234.50, esta entrada imprimirá:

Amount: -1234.50-

Estos ejemplos muestran cómo insertar un archivo de texto en un párrafo:

(Your conditions are: [=cnd23.txt=] and [=cnd54.txt=] .) VSUB 0 SHP

Si una variable contiene el nombre del archivo, como en los campos CND1 y CND2 en este ejemplo, la sustitución de archivos de texto y variables se puede combinar de la manera siguiente:

(Sus condiciones son: [= \$\$CND1.=] and [= \$\$CND2.=] .) VSUB 0 SHP

### Modos

Este comando es aplicable en todos los modos.

Comandos relacionados:

- SETVAR
- STARTDBM
- VSUB2
- VSUB3
- SETMPATH
- SETPPATH

## VSUB2

**VSUB2** ejecuta **VSUB**. **VSUB2** considera la cadena resultante como un nombre de variable y proporciona su valor. Si la variable no existe se produce un error indefinido.

Los nombres de las variables se encapsulan entre \$\$ y .., o entre [= y =].

### Sintaxis

```
(printable data with variable references) VSUB2
(string to merge) (string holding $$..) VSUB2
```

### Ejemplos

En este ejemplo, si el contenido de FIELD 7 es 5, el nombre resultante de la variable es VARcode5, y el contenido de la variable May se imprime.

```
/VARcode5 (May) SETVAR
....
(VARcode$$FIELD7..) VSUB2 SHL
```

Use **VSUB2** en un procedimiento de alineación de entrada RPE para combinar el campo RPE en una cadena a través de la omisión del nombre de la variable entre \$\$ y ..

En este ejemplo se utiliza VSUB2 en un procedimiento de alineación de RPE para combinar un campo en una cadena, recuperar el valor de la variable e imprimirlo. Si el campo especificado por el inicio de la posición 25 y la longitud de 2 caracteres en el ejemplo anterior contiene 02, esta entrada imprime February.

```
/VARmonth01 (January) SETVAR
/VARmonth02 (February) SETVAR
...
10 BEGINRPE
...
[ { (VARmonth$$..) VSUB2 SH } ..... 25 2 /F1 BLACK ]
```

### Modos

Este comando es aplicable en todos los modos.

Comandos relacionados:

- **SETVAR**
- **STARTDBM**
- **VSUB**
- **VSUB3**



## VSUB3

**VSUB3** ejecuta **VSUB** y a continuación **VSUB3** evalúa la cadena resultante como un nombre de variable y, si la variable no existe, **VSUB3** produce su valor. Si la variable no existe, **VSUB3** produce la cadena de entrada.

Los nombres de las variables se encapsulan entre \$\$ y ., o entre [= y =].

### Sintaxis

```
(input string) (string holding $$.) VSUB3
```

Use **VSUB3** en un procedimiento de alineación de entrada RPE para combinar el campo RPE en una cadena a través de la omisión del nombre de la variable entre \$\$ y .

### Ejemplos

```
/VARmonth01 (January) SETVAR  
(month01) (VAR$$.) VSUB3
```

produce January, mientras que

```
(month13) (VAR$$.) VSUB3
```

produce month13

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

- **VSUB**
- **VSUB2**
- **SETVAR**
- **STARTDBM**

## VSUB4

FreeFlow VI

**VSUB4** tiene el mismo comportamiento que **VSUB** pero, además, **VSUB4** omite las líneas en blanco de las variables del bloque de texto. Este comando está pensado para usarse en bloques de nombre y dirección, pero se puede usar en otros bloques de texto.

Una línea variable en blanco es una línea delimitada por un carácter de fin de línea que contiene una o diversas variables que se evalúan como en blanco, vacías o con espacios.



Nota: La opción SHP 0 está aún disponible y se puede utilizar para las líneas de impresión individuales de un bloque de direcciones. **VSUB4** ahora permite crear el bloque de dirección completa en un único comando **SHP/SHMF**.

### Sintaxis

(datos imprimibles con referencias del archivo de texto o variables) VSUB4 0 SHP

### Ejemplos

```
/VAR1 (David Kirk) SETVAR
/VAR2 (12 Baker Street) SETVAR
/VAR3 (Los Angeles, CA 90245) SETVAR
300 300 MOVETO
( $$VAR1.
$$VAR2.
$$VAR3 ) VSUB4 0 SHP
```

En este ejemplo se imprime el siguiente texto, sin supresiones, ya que no hay ninguna variable vacía:

David Kirk

12 Baker Street

Los Angeles, CA 90245

```
/VAR1 (David Kirk) SETVAR
/VAR2 ( ) SETVAR
/VAR3 (Los Angeles, CA 90245) SETVAR
300 300 MOVETO
( $$VAR1.
$$VAR2.
$$VAR3 ) VSUB4 0 SHP
```

En este ejemplo se imprime el siguiente texto, en el que se suprime **VAR2** porque está vacío:

David Kirk

Los Angeles, CA 90245

# Variables

Este capítulo incluye:

Variables de VIPP® por tipo .....	534
AUTOGRID.....	538
ATRÁS .....	539
BCOUNT .....	540
BLGRID .....	541
BPCOUNT .....	542
CLIP.....	543
COLW.....	544
CPCOUNT .....	545
CURLINE .....	546
D_DD.....	547
D_DOY.....	548
D_DWL.....	549
D_DWS .....	550
D_MO .....	551
D_MOL.....	552
D_MOS .....	553
D_YY .....	554
D_YYYY .....	555
DEVRES .....	556
DJDECMD .....	557
DJDEPAR .....	558
FRCOUNT .....	559
FRLEFT .....	560
GLT.....	561
GRIDSKIP .....	563
HCOLOR .....	564
HDISP .....	565
HPOS .....	566
HPOS2 .....	567
IHEIGHT.....	568
IWIDTH.....	569

LNCOUNT .....	570
LPCOUNT .....	571
LPINDEX.....	572
LSP .....	573
MPR.....	574
OTCLIP y ITCLIP .....	575
PAGEH.....	576
PAGEW .....	577
PDFDEVICE .....	578
PDFPAGES.....	579
PLINES .....	580
PPCOUNT .....	583
PREV y NEXT .....	584
PRODUCT .....	585
PSIZE .....	586
RPCOUNT .....	589
RPEPOS .....	590
RPLEFT .....	592
SHEETH.....	593
SHEETW .....	594
SHPOS.....	595
SLENGTH.....	596
SSIZE .....	597
SVPOS.....	598
T_AMPM .....	599
T_HH .....	600
T_HH2 .....	601
T_MM.....	602
T_SS.....	603
T_TZN.....	604
TLENGTH.....	605
TLGRID.....	606
TPATH .....	607
UV2L para efecto UV de dos capas.....	608

VARDataFileName.....	609
VDISP .....	610
VPOS .....	611
XGFVER.....	612
XMLATN .....	613
XMLATV.....	614
XMLDTH.....	615
XMLPAR.....	616
XMLPATH.....	617
XMLTAG .....	618
XMLVAL .....	619
YINIT .....	620
ZSPAGE .....	621
ZSRECNUM.....	622
ZSREPCNT.....	623
ZSREPIDX .....	624

Una variable de VIPP® es una palabra clave, integrada o definida mediante SETVAR, que representa un valor que puede cambiar durante el procesamiento de un trabajo. Puede sustituir a cualquier operando de cualquier comando o función de transformación siempre que el valor de la variable pueda sustituir a ese operando. Algunas variables integradas como, por ejemplo, RPEPOS, pueden necesitar sus propios operandos. Las variables pueden ser de estos tipos de PostScript, como se define en el lenguaje PostScript:

- Cadena
- Entero
- Número real
- Booleano
- Matriz
- Procedimiento
- Nombre

También pueden ser variables integradas de estos tipos de VIPP®:

- Formulario
- GEPkey
- Colorkey

## Variables de VIPP® por tipo

La siguiente tabla proporciona una lista de todas las variables VIPP® integradas y enumera el tipo de cada variable.

VARIABLES INTEGRADAS	TIPO						
	BOOLEA- NO	COLOR	FORMU- LARIO	GEPKEY	ENTERO	REAL	CADENA
AUTOGRID			X				
ATRÁS	X						
BCOUNT					X		
BLGRID			X				
BPCOUNT					X		
CLIP				X			
COLW						X	
CPCOUNT					X		
CURLINE					X		
D_DD						X	
D_DOY						X	
D_DWL						X	
D_DWS						X	
D_MO						X	
D_MOL						X	
D_MOS						X	
D_YY						X	
D_YYYY						X	
DEVRES					X		
DJDECMD							X
DJDEPAR							X
FRCOUNT					X		
FRLEFT						X	
GLT						X	
GRIDSKIP	X						

VARIABLES INTEGRADAS	TIPO						
	BOOLEA- NO	COLOR	FORMU- LARIO	GEPKEY	ENTERO	REAL	CADENA
HCOLOR		X					
HDISP						X	
HPOS						X	
HPOS2						X	
IHEIGHT						X	
IWIDTH						X	
LNCOUNT					X		
LPCOUNT					X		
LPINDEX					X		
LSP						X	
MPR						X	
OTCLIP y ITCLIP				X			
PAGEH						X	
PAGEW						X	
PDFDEVICE	X						
PDFPAGES					X		
PLINES					X		
PPCOUNT					X		
ANT / SIG							
PRODUCT							X
PSIZE						X	
RPCOUNT					X		
RPEPOS						X	
RPLEFT					X		
SHEETH						X	
SHEETW						X	
SHPOS						X	
SLENGTH					X		

VARIABLES INTEGRADAS	TIPO						
	BOOLEA- NO	COLOR	FORMU- LARIO	GEPKEY	ENTERO	REAL	CADENA
SSIZE						X	
SVPOS						X	
T_AMPM							X
T_HH							X
T_HH2							X
T_MM							X
T_SS							X
T_TZN							X
TLENGTH			X				
TLGRID			X				
TPATH				X			
UV2L (efecto UV de dos capas)		X					
VARDataFileName							
VDISP						X	
VPOS						X	
XGFVER							X
XMLATN							X
XMLATV							X
XMLDTH					X		
XMLPAR							X
XMLPATH							X
XMLTAG							X
XMLVAL							X
YINIT						X	
ZSPAGE					X		
ZSRECNUM					X		



VARIABLES INTEGRADAS	TIPO						
	BOOLEA- NO	COLOR	FORMU- LARIO	GEPKEY	ENTERO	REAL	CADENA
ZSREPCNT					X		
ZSREPIDX					X		

## AUTOGRID

El formulario integrado **AUTOGRID** permite la impresión de un archivo de prueba de modo de línea sobre una cuadrícula de líneas y columnas.

Utilice esta función para determinar la posición del registro en el archivo de impresión al codificar definiciones de **RPE**. **AUTOGRID** debe utilizarse en un **JDT** con la cuadrícula, el margen y las opciones de fuente correspondientes. Por ejemplo, consulte el archivo `/usr/xgf/jdtlib/autogrid.jdt`.

### Sintaxis

```
(AUTOGRID) SETFORM
```

### Ejemplos

```
80 70 SETGRID
130 130 130 130 SETMARGIN          % to allow line/column number printing
(AUTOGRID) SETFORM
/NCR 0 SETFONT
```



Consejo: Utilice **NMP\_off** para desactivar los efectos secundarios del prefijo de modo nativo (NMP), si los hay. La cuadrícula se imprime con el color de realce actual en los sistemas de color de realce.

### Modos

Este formulario integrado es aplicable en el modo de línea.

### Comandos relacionados

- **NMP\_off**
- **SETFONT**
- **SETFORM**
- **SETGRID**
- **SETMARGIN**
- **STARTLM**

## ATRÁS

**BACK** es una variable booleana integrada que es verdadera si la página actual figura en la parte posterior de la hoja actual, y falsa en caso contrario.

### Sintaxis

ATRÁS

### Ejemplos

- `IF BACK { ... action on back pages ... } ENDIF`
- `IF BACK not { ... action on front pages ... } ENDIF`



Nota: La información proporcionada por esta variable depende de la página actual cuando se usa en **BEGINPAGE** y de la página siguiente si se usa **/P ENDPAGE**.

### Modos

Esta variable es aplicable en todos los modos.

### Comandos relacionados

- `DUPLEX_on`
- `SETBFORM`
- `TUMBLEDDUPLEX_on`

## BCOUNT

**BCOUNT** ofrece el número de espacios en blanco en una cadena.

### Sintaxis

```
(string) BCOUNT
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**BSTRIP**, **BTRIM**

## BLGRID

El formulario integrado **BLGRID** activa la impresión de una muestra de trabajo VIPP® sobre una cuadrícula de unidades con el origen inferior izquierdo. Esta cuadrícula es útil en el diseño de formularios de modo nativo.

**BLGRID** permite ubicar las posiciones de impresión en la página al codificar un trabajo de VIPP® como un formulario, **RPE** o modo nativo. La unidad utilizada es la actual configurada por **SETUNIT**.

### Sintaxis

```
(BLGRID) SETFORM
```

El origen (0,0) de la cuadrícula está vinculado a los márgenes. Se recomienda la configuración de los márgenes a cero en la mayoría de los casos.

### Ejemplos

```
0 0 0 0 SETMARGIN
```

```
(BLGRID) SETFORM
```



Consejo: Aumente **SETMAXFORM** para imprimir la cuadrícula encima de otros formularios. En sistemas de color de realce, la cuadrícula se imprime con el color de realce actual.

### Modos

Este formulario integrado es aplicable en todos los modos.

### Comandos relacionados

- **SETFORM**
- **SETMARGIN**
- **SETMAXFORM**
- **SETUNIT**
- **TLGRID**

## BPCOUNT

Esta variable integrada proporciona el recuento de páginas del folleto actual. El número de páginas impresas desde el último comando **STARTBOOKLET**.

### Sintaxis

BPCOUNT

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

STARTBOOKLET, ENDBOOKLET

## CLIP

**CLIP** es una GEPkey integrada y reservada para el recorte. Use **CLIP** con un comando que usa una GEPkey, como **DRAWB**, **DRAWBR**, **DRAWPOL** o **SHx**. El área definida por este comando se convierte en la nueva área de recorte.

El área de recorte es el área en la página donde se activa la impresión. No se procesa ninguna marca fuera del área de recorte. El valor prefijado es toda la página.

Para desactivar el recorte y restaurar la impresión de la página completa, use **ENDCLIP**. Para desactivar el recorte solamente, use **PAGEBRK**.

### Sintaxis

CLIP

### Ejemplos

```
[ 200 200 1200 3400 2300 200 ] CLIP DRAWPOL
```

El contenido fuera del triángulo no se imprime después del comando.



Consejo: Puede usar la GEPkey integrada para recortar tramas. No pueden recortarse elementos ya marcados en la página.

### Modos

Este GEPkey integrado es aplicable en todos los modos.

### Comandos relacionados

- **DRAWB y DRAWBR**
- **DRAWPOL**
- **ENDCLIP**
- **SHx**

## COLW

**COLW** es una variable dinámica integrada que proporciona la anchura de la columna actual definida por **SETCOLWIDTH**. **COLW** puede utilizarse con las definiciones de BATkey.

### Sintaxis

COLW

### Ejemplos

Este ejemplo dibuja una línea horizontal con la misma longitud que la anchura de la columna actual.

```
100 200 COLW 0 S1 DRAWB
```

### Modos

Esta variable es aplicable en todos los modos.

### Comandos relacionados

**SETCOLWIDTH**



## CPCOUNT

**CPCOUNT** es una variable integrada que proporciona el número de copia actual. Utilice esta variable para realizar acciones en una determinada pasada de copia.

### Sintaxis

CPCOUNT

### Ejemplos

En este ejemplo, se cancela el formulario de la contraportada en la tercera copia; de lo contrario, el formulario de la contraportada produce contraportadas en blanco.

```
3 SETCYCLECOPY  
(bform.frm) [1 2] SETBFORM  
{ IF CPCOUNT 3 eq { 0 SETMAXFORM} ENDIF }BEGINPAGE
```

### Modos

Esta variable es aplicable en todos los modos.

### Comandos relacionados

- **BEGINPAGE**
- **ENDIF**
- **SETBFORM**
- **SETCYCLECOPY**
- **SETMAXBFORM**

## CURLINE

**CURLINE** es una variable integrada que proporciona el número de la línea actual que está siendo procesada por una entrada de **RPE**. Esta variable se utiliza solo para pruebas de **SETRCD**. Para otros comandos relacionados con **RPE**, y para más información, consulte [Información de comando RPE](#).

### Sintaxis

CURLINE

### Ejemplos

```
/IF_1STLINE { CURLINE 1 eq } SETRCD
```

### Modos

Esta variable es aplicable solo en el modo de línea.

### Comandos relacionados

[SETRCD](#)

## D\_DD

**D\_DD** se usa en combinación con el comando **GETDATE** para configurar el día del mes correspondiente 1-31 para la aplicación.

### Sintaxis

D\_DD

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**

## D\_DOY

**D\_DOY** se usa en combinación con el comando **GETDATE** para configurar el día numérico del año correspondiente 1-366 para la aplicación.

### Sintaxis

D\_DOY

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**

## D\_DWL

**D\_DWL** se usa en combinación con el comando **GETDATE** para configurar el nombre no abreviado correspondiente de la semana, por ejemplo lunes, martes, etc. de la aplicación.

### Sintaxis

D\_DWL

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**

## D\_DWS

**D\_DWS** se usa en combinación con el comando **GETDATE** para configurar el nombre abreviado correspondiente de la semana, por ejemplo lun, mar, etc. de la aplicación.

### Sintaxis

D\_DWS

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**

## D\_MO

**D\_MO** se usa en combinación con el comando **GETDATE** para configurar el nombre numérico del mes, por ejemplo 01-12 de la aplicación.

### Sintaxis

D\_MO

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**

## D\_MOL

**D\_MOL** se usa en combinación con el comando **GETDATE** para configurar el nombre no abreviado correspondiente del mes, por ejemplo enero, febrero, etc. de la aplicación.

### Sintaxis

D\_MOL

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**



## D\_MOS

**D\_MOS** se usa en combinación con el comando **GETDATE** para configurar el nombre abreviado correspondiente del mes, por ejemplo ene, feb, etc. de la aplicación.

### Sintaxis

D\_MOS

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**

## D\_YY

**D\_YY** se usa en combinación con el comando **GETDATE** para configurar el año de dos dígitos correspondiente para la aplicación.

### Sintaxis

D\_DYY

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**

## D\_YYYY

**D\_YYYY** se usa en combinación con el comando **GETDATE** para configurar el año de cuatro dígitos correspondiente para la aplicación.

### Sintaxis

D\_YYYY

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**

## DEVRES

**DEVRES** proporciona la resolución del dispositivo actual en ppp (puntos por pulgada).

### Sintaxis

DEVRES

### Ejemplos

(Resolución: \$\$DEVRES.) VSUB SH

### Modos

Esta variable integrada es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**

## DJDECMD

**DJDECMD** es una variable integrada que representa la palabra clave **DJDE** que ha sido extraída de un registro **DJDE** por el comando **PROCESSDJDE**. **DJDECMD** está disponible solamente en el interior de los operandos `djde_proc` de un comando **PROCESSDJDE**. **DJDECMD** siempre se asocia con la variable **DJDEPAR** correspondiente.

### Sintaxis

DJDECMD

### Ejemplos

```
IF DJDECMD (FORM) eq { ($$DJDEPAR..frm) VSUB SETFORM } ENDIF
```

### Modos

Esta variable es aplicable solo en el modo de línea.

### Comandos relacionados

- **DJDEPAR**
- **PROCESSDJDE**
- **CASE**
- **IF/ELSE/ELIF/ENDIF**

## DJDEPAR

**DJDEPAR** es una variable integrada que representa un parámetro **DJDE** que ha sido extraído de un registro **DJDE** por el comando **PROCESSDJDE**. **DJDEPAR** está disponible solamente en el interior de los operandos `djde_proc` de un comando **PROCESSDJDE**. **DJDEPAR** siempre se asocia con la variable **DJDECMD** correspondiente.

### Sintaxis

DJDEPAR

### Ejemplos

```
IF DJDECMD (FORM) eq { ($$DJDEPAR..frm) VSUB SETFORM } ENDIF
```

### Modos

Esta variable es aplicable solo en el modo de línea.

### Comandos relacionados

DJDECMD, PROCESSDJDE, VSUB

## FRCOUNT

**FRCOUNT** es una variable integrada que proporciona el número de marco actual. Utilice esta variable para realizar acciones según un determinado número de trama.

### Sintaxis

FRCOUNT

### Ejemplos

```
IF FRCOUNT 3 eq  
    ( Do this )  
ELSE  
    ( Do that }  
ENDIF
```



Nota: Los números de tramas tienen un valor entre 1 y el número de tramas establecidas por **SETLKF**. Si **FRCOUNT** es igual a 0, la página actual todavía no se ha inicializado, por lo que la trama 1 todavía no ha empezado a rellenarse.

### Modos

Este comando es aplicable en modos nativo y de base de datos.

### Comandos relacionados

- **FRLEFT**
- **GOTOFRAME**
- **NEWFRAME**
- **SETLKF**

## FRLEFT

**FRLEFT** es una variable integrada que proporciona el nivel de espacio vertical disponible de la posición de impresión actual hasta la parte inferior de la trama actual en la pila. El valor se expresa en unidades actuales.

### Sintaxis

FRLEFT

### Ejemplos

```
IF FRLEFT 100 lt  
{ PAGEBRK }  
ENDIF
```

### Modos

Este comando es aplicable en modos nativo y de base de datos.

### Comandos relacionados

- **FRCOUNT**
- **GOTOFRAME**
- **NEWFRAME**
- **SETLKF**



## GLT

**GLT** es una variable integrada que proporciona el tamaño de la fuente y configura el espaciado de línea correspondiente para una fuente de texto de correlación o GlossMark determinados. Dicho tipo de fuente requiere el uso de un tamaño de punto que forma parte de la definición de la fuente. Use **GLT** exclusivamente como el parámetro de tamaño de un comando **SETFONT**. Para obtener más información, consulte [Impresión especializada](#).

### Sintaxis

```
/glossfont_name GLT SETFONT
```

### Donde:

La fuente se instala primero en el dispositivo PostScript mediante el procedimiento normal para instalar una fuente.

### Limitaciones

La impresión GlossText requiere el uso exclusivo de uno de los colores predefinidos siguientes:

- GL\_Black
- GL\_Gray
- GL\_Red
- GL\_Green
- GL\_Blue
- GL\_Yellow
- GL\_Cyan
- GL\_Magenta
- GL\_Maroon
- GL\_lightBlue
- GL\_lightGreen
- GL\_Olive
- GL\_Peach

### Ejemplos

Este ejemplo usa una construcción **BCALL** que se recomienda para aislar la secuencia GlossText del resto del código de VIPP®:

```
100 500 MOVETO
{ GL_Magenta SETTXC
  /NeueModern-GL-30 GLT SETFONT
  (Hello world) SH
} BCALL
```

### Modos

Esta variable es aplicable en todos los modos.

Comandos relacionados:

- SETTXC
- INDEXCOLOR
- SETFONT
- INDEXFONT

## GRIDSKIP

**GRIDSKIP** es una variable VIPP® booleana integrada que se usa para determinar cómo se ha producido un **PAGEBRK** en modo de línea. **GRIDSKIP** es verdadero cuando se exceden las líneas por páginas, tal como están definidas mediante **SETGRID**. **GRIDSKIP** es falso por el contrario cuando se produce una condición **PCC**, **SETPBRK** o **SETSKIP**.

Se puede usar **GRIDSKIP** solo en modo de línea en un proceso **BEGINPAGE** o **ENDPAGE**.

### Ejemplos

```
{ IF GRIDSKIP
  { (continue.frm) SETFORM }
  ELSE
  { (main.frm) SETFORM }
  ENDIF
} BEGINPAGE
```

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

- **SETSKIP**
- **IF/ELSE/ELIF/ENDIF**
- **BEGINPAGE**
- **ENDPAGE**

## HCOLOR

La Colorkey integrada, **HCOLOR**, selecciona el color de realce actual en sistemas de color de realce de DocuPrint NPS.

**HCOLOR** será negro en otros dispositivos. **HCOLOR** requiere que utilice el comando con los elementos **SETTXC**, **INDEXCOLOR**, **SETGEP** y **RPE**.

### Sintaxis

HCOLOR

### Modos

Esta Colorkey integrada es aplicable en todos los modos.

### Comandos relacionados

- FROMLINE
- INDEXCOLOR
- SETGEP
- SETTXC
- RPEKEY

## HDISP

**HDISP** proporciona el desplazamiento entre la posición actual de impresión horizontal secundaria y las guardadas por **SAVEPP**. Use **HDISP** con **MOVETO**, **MOVEH**, **DRAWBx** y **DRAWPOL**.

### Sintaxis

HDISP

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

- **SAVEPP**
- **SHPOS**
- **SVPOS**
- **VDISP**

## HPOS

HPOS proporciona la posición de impresión principal horizontal actual.

### Sintaxis

HPOS

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

- HPOS2
- VPOS
- SVPOS
- HDISP
- SAVEPP
- SHPOS
- VDISP

## HPOS2

**HPOS2** proporciona la posición de impresión secundaria horizontal actual.

### Sintaxis

HPOS2

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

- **HPOS**
- **VPOS**
- **SVPOS**
- **HDISP**
- **SAVEPP**
- **SHPOS**
- **VDISP**

## IHEIGHT

**IHEIGHT** proporciona la altura de una imagen en las unidades actuales en TIFF, JPEG o EPS.

### Sintaxis

```
(image_name) IHEIGHT
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**ICALL**, **SCALL**



## IWIDTH

**IHEIGHT** proporciona la altura de una imagen en las unidades actuales en TIFF, JPEG o EPS.

### Sintaxis

```
(image_name) IWIDTH
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**ICALL**, **SCALL**

## LNCOUNT

**LNCOUNT** es una variable integrada que proporciona el número de líneas en la página actual. Use **LNCOUNT** solamente en los procedimientos **BEGINPAGE** o **ENDPAGE**.

### Sintaxis

LNCOUNT

### Ejemplos

```
{ IF LNCOUNT 50 gt
  { (jdt1.jdt) SETJDT }
ELSE
  { (jdt2.jdt) SETJDT }
ENDIF
...
BEGINPAGE
```

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

**BEGINPAGE**, **ENDPAGE**

## LPCOUNT

**LPCOUNT** es una variable integrada o un entero que proporciona el número de página lógica actual. Este es el número impreso por el comando **SETPAGENUMBER**. **LPCOUNT** se puede utilizar para capturar el número de página de una determinada página y hacer referencia al mismo posteriormente.

**LPCOUNT** empieza a partir de 0 de forma prefijada debido al **SETPAGENUMBER** predefinido en `xf.def`.

Para imprimir el **LPCOUNT** es necesaria una construcción **VSUB**.

### Sintaxis

LPCOUNT

### Ejemplos

```
(Page #) 1 1 SETPAGENUMBER IF ... { /VARpage LPCOUNT SETVAR } ENDIF IF ... { (as mentio
```



Nota: La información proporciona por esta variable depende de la página actual cuando se usa en **BEGINPAGE** y de la página siguiente si se usa **/P ENDPAGE**.

### Modos

Esta variable es aplicable en todos los modos.

### Comandos relacionados

**SETPAGENUMBER**

## LPINDEX

**LPINDEX** es una variable integrada (entero) que proporciona el índice de la página lógica actual en la página física actual. **LPINDEX** varía entre 1 y el número de entradas en la definición de varias en 1.

Para imprimir un **LPINDEX**, es necesaria una construcción **VSUB**.

### Sintaxis

LPINDEX

### Ejemplos

```
IF LPINDEX 1 eq { ... action on 1st logical page ... } ENDIF
```



Nota: La información proporcionada por esta variable depende de la página actual cuando se usa en **BEGINPAGE** y de la página siguiente si se usa **/P ENDPAGE**.

### Modos

Esta variable es aplicable en todos los modos.

### Comandos relacionados

**SETMULTIUP**

## LSP

**LSP** es una variable dinámica integrada que proporciona el espaciado de línea actual definido por **SETGRID** o **SETLSP** a una entrada **RPE** en una definición **FROMLINE**. La variable **LSP** permite la interacción entre **RPE** y **SETGRID**. **LSP** se utiliza con frecuencia en combinación con **YINIT**.

### Sintaxis

**LSP**

**/LSP**

Si **SETGRID** puede variar durante un trabajo, utilice la segunda sintaxis.

### Ejemplos

El siguiente es un ejemplo de configuración de formato, generada por **SETGRID**, que imprime las dos primeras líneas con fuente negrita en un fondo cebra.

```
80 60 SETGRID
LMEDIUM 1 1 SETZEBRA
100 200 200 200 SETMARGIN
/F1 /NCR 0 INDEXFONT
/F2 /NCRB 0 INDEXFONT
2 BEGINRPE
1 FROMLINE [ 0 0 0 nu]] YINIT LSP 0 80 /F2 BLACK ]
3 FROMLINE [ 0 0 0 nu]] YINIT LSP 0 80 /F1 BLACK ]
ENDRPE
```

### Modos

Esta variable es aplicable en el modo en línea y de base de datos.

### Comandos relacionados

**FROMLINE**, **SETGRID**, **YINIT**

## MPR

**MPR** es una variable integrada que proporciona el tamaño de la fuente y configura el espaciado de línea correspondiente para una fuente Microprint determinada. Una fuente Microprint requiere un tamaño de punto que forma parte de la definición de la fuente. Use **MPR** exclusivamente como el parámetro de tamaño de un comando **SETFONT**. Para obtener más información, consulte [Impresión especializada](#).

### Sintaxis

```
/microfont_name MPR SETFONT
```

La fuente se instala primero en el dispositivo PostScript mediante el procedimiento normal para instalar una fuente.

### Ejemplos

Este ejemplo usa una construcción **BCALL** que se recomienda para aislar la secuencia Microprinting del resto del código de VIPP®:

```
100 500 MOVETO
{ /micro_f7 MPR SETFONT
  (Hello world) SH
} BCALL
```

### Limitaciones

Microprinting usa exclusivamente el color sólido CMKY con los componentes iguales a 0 o 1. Antes de que **SETFONT** se ejecute con **MPR**, seleccione el color. Cualquier otro color que no sea un color CMYK sólido genera un error.



Consejo: Dado que la capacidad de leer microtexto puede variar en función de la calidad de los materiales de impresión, se recomienda que el texto se imprima de forma repetida. Puede utilizar la nueva sintaxis **SHP** utilizando un campo de recuento.

### Modos

Esta variable es aplicable en todos los modos.

### Comandos relacionados

- [SETTXC](#)
- [INDEXCOLOR](#)
- [SETFONT](#)
- [INDEXFONT](#)

## OTCLIP y ITCLIP

**OTCLIP** e **ITCLIP** son GEPkeys reservadas e integradas. **OTCLIP** y **ITCLIP** se utilizan para definir una ruta pensada para interactuar con comandos **SH**P posteriores. Use **OTCLIP** para definir una forma alrededor de la cual se ajusta el texto. Use **ITCLIP** para definir una forma en la que se colocará el texto.

Use **OTCLIP** e **ITCLIP** con un comando que requiere una GEPkey, como **DRAWB**, **DRAWC**, **DRAWBR**, **DRAWPOL** o **DRAWPATH**.

Se pueden definir varias rutas continuas o discontinuas antes de invocar a **SH**P.

Utilice **ETCLIP** para borrar la ruta. La ruta se borra automáticamente al principio de cada página.

### Sintaxis

#### Como GEPkey:

```
OTCLIP % defines a shape to place text around
ITCLIP % defines a shape to place text inside
[ GEPkey OTCLIP ] % the path is also painted by GEPkey
[ GEPkey ITCLIP ] % the path is also painted by GEPkey
```

### Ejemplos

```
ORITL
/NHE 20 SETFONT
100 SETLSP
800 300 200 200 [ S1 OTCLIP ] DRAWB
500 270 MOVETO
(Text to be printed around the box defined using DRAWC) 800 3 SHP
```

### Modos

Estas GEPkeys integradas son aplicables en todos los modos.

### Comandos relacionados

- **DRAWB** y **DRAWBR**
- **DRAWC**
- **DRAWPOL**
- **DRAWPATH** y **DRAWPATHR**
- **ETCLIP**

## PAGEH

**PAGEH** proporciona la altura de la página lógica actual.

### Sintaxis

PAGEH

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

PAGEW, SHEETW, SHEETH



## PAGEW

**PAGEW** proporciona la anchura de la página lógica actual.

### Sintaxis

PAGEW

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

PAGEW, SHEETW, SHEETH

## PDFDEVICE

**PDFDEVICE** es una variable booleana integrada que tiene el valor verdadero si el dispositivo de impresión actual es un productor de PDF, y falso en caso contrario. **PDFDEVICE** tiene como objetivo permitir a los trabajos VIPP® crear, de forma condicional, documentos en papel o PDF, o ambos, con algunas partes del trabajo.

### Sintaxis

PDFDEVICE

### Ejemplos

En el ejemplo siguiente se muestra cómo detectar determinados documentos en un trabajo VIPP®, y producir los documentos solo en formato de papel o de documento PDF.

```
{ /VAR_OUTPUT 1 20 3 GETFIELD
CASE VAR_OUTPUT {}
  (PRT) { IF PDFDEVICE { SKIPPAGE } ENDIF }      % print only
  (WEB) { IF PDFDEVICE not { SKIPPAGE } ENDIF }   % PDF only
ENDCASE
} BEGINPAGE
```

### Modos

Esta variable es aplicable en todos los modos.

### Comandos relacionados

- IF/ELSE/ELIF/ENDIF
- SETPCD
- GETFIELD
- SKIPPAGE
- BEGINPAGE

## PDFPAGES

**PDFPAGES** ofrece al número de páginas de un documento PDF.

### Sintaxis

```
(document1.pdf) PDFPAGES
```

### Ejemplos

En el siguiente ejemplo se muestra cómo imprimir un PDF excepto la última página.

```
/VARpages (pdf1.pdf) PDFPAGES SETVAR  
{ [ /PDFTpage RPCOUNT'- '1 ] SETPARAMS  
(pdf1.pdf) CACHE SCALL  
PAGEBRK  
} VARpages'- '1 REPEAT
```

### Modos

Esta variable es aplicable en todos los modos.

### Comandos relacionados

**SCALL**, **RUNPDF**

## PLINES

**PLINES** proporciona el número de líneas o columnas en un párrafo según la fuente y el modo de escritura actuales.

### Sintaxis

```
(string) align PLINES
```

```
(string) colwidth align PLINES
```

**Donde:**

**alignar** es la opción de alineación y puede especificarse como uno de los siguientes valores:

- 0: Izquierda
- 1: Derecha
- 2: Centro
- 3: Justificar
- +00: Tratar caracteres de nueva línea 0×0A como espacios
- +10: Eliminar espacios en blanco duplicados entre palabras
- +20: Tratar caracteres de nueva línea 0×0A como final de línea
- +30: Eliminar espacios en blanco duplicados entre las palabras y tratar los caracteres de nueva línea 0×0A como final de línea
- +000: Ceñir arriba las palabras latinas y entre caracteres asiáticos
- +100: Ceñir abajo según las reglas de Asia
- +200: Ceñir arriba según las reglas de Asia
- +300: Ceñir arriba + comillas voladas las reglas de Asia
- +400: Comillas voladas según las reglas de Asia
- +1000: Configurar el procesamiento vertical a horizontal, método 0, horizontal en vertical
- +2000: Configurar el procesamiento vertical a horizontal, método 1, giro 90 grados en sentido horario
- +4000: Establecer conversión
- +5000: Configurar la conversión vertical a horizontal, método 0, horizontal en vertical
- +6000: Configurar la conversión vertical a horizontal, método 1, giro 90 grados en sentido horario



Nota: Las reglas de idiomas y caracteres de Asia para la codificación relacionada deben configurarse anteriormente a través de los comandos **SETCJKENCMAP** y **SETCJKRULES**.

La lista de caracteres que se cambiará de vertical a horizontal se configura anteriormente a través del comando **SETV2HTABLE**. La tabla de conversión de la codificación relacionada se configura anteriormente a través del comando **SETV2HCONV**.

Las listas y las tablas predeterminadas se definen en el archivo de configuración `xf/src/cjk.def`.

**colwidth** Anchura de columna utilizada para el ajuste de palabras. Cuando no se especifica **colwidth**, se utiliza el valor definido por un **SETCOLWIDTH** anterior. Cuando se especifica **colwidth**, se sustituye y reemplaza el valor definido por un **SETCOLWIDTH** anterior.

## Modos

Este comando es aplicable en todos los modos.

**Comandos relacionados**

**SHP**, **PSIZE**

## PPCOUNT

**PPCOUNT** es una variable integrada (un entero) que proporciona el número absoluto de página física actual, también llamado imagen o impresión de cara, desde el inicio del trabajo. El entero de la variable se puede imprimir en cada página con una fuente pequeña y usarse posteriormente como una referencia para la reimpresión mediante **PAGERANGE**.

Para imprimir el **PPCOUNT** es necesaria una construcción **VSUB**.

### Sintaxis

PPCOUNT

### Ejemplos

```
{ /NHE 6 SETFONT 80 80 MOVETO ($$PPCOUNT.) VSUB SH } SETFORM
```



Nota: La información proporciona por esta variable depende de la página actual cuando se usa en **BEGINPAGE** y de la página siguiente si se usa **/P ENDPAGE**.

### Modos

Esta variable es aplicable en todos los modos.

### Comandos relacionados

**PAGERANGE**

## PREV y NEXT

Los prefijos **PREV\_** and **NEXT\_** proporcionan acceso integrado a los valores anteriores o siguientes de cualquier campo de base de datos en un **DBM** o de variables de tabla en un bucle **FOREACH**. Si no hay ningún valor especial anterior o siguiente se devuelve un valor **ninguno**.

### Sintaxis

PREV\_vname

NEXT\_vname

### Donde:

**vname** es una de las siguientes:

El nombre de un campo de base de datos que solo disponible en el modo de base de datos

El nombre de una variable de una tabla VIPP® procesado por un bucle **FOREACH** que está disponible en todos los modos

### Ejemplos

```
IF CustomerID PREV_CustomerID ne
  { ... action on CustomerID change ... }
ENDIF

IF NEXT_CustomerID (**none**) eq
  { ... action on last record ... }
ENDIF
```

### Comandos relacionados

STARTDBM, FOREACH



## PRODUCT

**PRODUCT** es una variable de cadena integrada que identifica al intérprete de PostScript donde se ejecuta VI Compose.

### Sintaxis

PRODUCT

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

Ninguno

## PSIZE

**PSIZE** proporciona la altura o la anchura en unidades actuales de un párrafo según el modo de escritura y las fuentes actuales. Cuando se utiliza **PSIZE**, la dimensión no definida siempre es igual a la anchura de la columna seleccionada.

### Sintaxis

```
(string) align PSIZE
```

```
(string) colwidth align PSIZE
```

### Donde:

**alinear** es la opción de alineación y puede especificarse como uno de estos valores:

- 0: Izquierda
- 1: Derecha
- 2: Centro
- 3: Justificar
- +00: Tratar caracteres de nueva línea 0×0A como espacios
- +10: Eliminar espacios en blanco duplicados entre palabras
- +20: Tratar caracteres de nueva línea 0×0A como final de línea
- +30: Eliminar espacios en blanco duplicados entre las palabras y tratar los caracteres de nueva línea 0×0A como final de línea
- +000: Ceñir las palabras latinas y entre caracteres asiáticos
- +100: Ceñir abajo según las reglas de Asia
- +200: Ceñir arriba según las reglas de Asia
- +300: Ceñir arriba y usar comillas voladas las reglas de Asia
- +400: Comillas voladas según las reglas de Asia
- +1000: Configurar el procesamiento vertical a horizontal, método 0, horizontal en vertical
- +2000: Configurar el procesamiento vertical a horizontal, método 1, giro 90 grados en sentido horario
- +4000: Establecer conversión
- +5000: Configurar la conversión vertical a horizontal, método 0, horizontal en vertical
- +6000: Configurar la conversión vertical a horizontal, método 1, giro 90 grados en sentido horario



Nota: Las reglas de idiomas y caracteres de Asia para la codificación relacionada deben configurarse anteriormente a través de los comandos **SETCJKENCMAP** y **SETCJKRULES**.

La lista de caracteres que se cambiará de vertical a horizontal se configura anteriormente a través del comando **SETV2HTABLE**. La tabla de conversión de la codificación relacionada se configura anteriormente a través del comando **SETV2HCONV**.

Las listas y las tablas predeterminadas se definen en el archivo de configuración `xgf/src/cjk.def`.

**colwidth** Anchura de columna utilizada para el ajuste de palabras. Cuando no se especifica **colwidth**, se utiliza el valor definido por un **SETCOLWIDTH** anterior. Cuando se especifica **colwidth**, se sustituye y reemplaza el valor definido por un **SETCOLWIDTH** anterior.

## Modos

Este comando es aplicable en todos los modos.

**Comandos relacionados**

**SHP, PLINES**

## RPCOUNT

**RPCOUNT** es una variable integrada que está disponible solamente en un procedimiento **REPEAT** que proporciona el número, empezando por 1, de la iteración actual del bucle **REPEAT**.

Utilice esta variable con **IF**, **ELSE**, **ENDIF** o **CASE** para realizar determinadas acciones en iteraciones específicas del procedimiento **REPEAT**

### Sintaxis

RPCOUNT

### Ejemplos

```
{ IF RPCOUNT 1 eq          % action on first iteration
  { do this }
  ENDIF
  ...
} 10 REPEAT
```

### Modos

Esta variable es aplicable en todos los modos.

### Comandos relacionados

**REPEAT**, **RPLEFT**

## RPEPOS

**RPEPOS** es una variable integrada que proporciona la posición de impresión inicial o final, o el desplazamiento de un determinado grupo **RPE** definido mediante **FROMLINE** o **RPEKEY**. **RPEPOS** requiere que lo use con **MOVERTO**, **DRAWB** y **DRAWBR**, **DRAWBM** o **DRAWPOL**.

Para trazar un cuadro o una línea, o para colocar una imagen en una posición en función de la última línea del grupo **RPE**, use **RPEPOS** con **ENDPAGE**.

Para otros comandos relacionados con **RPE**, y para más información, consulte [Información de comando RPE](#).

### Sintaxis

```
/rpe_def offset HV_option RPEPOS
```

Donde:

<b>rpe_def</b>	especifica el número de línea o la tecla <b>RPE</b> utilizada en la definición de <b>FROMLINE</b> o <b>RPEKEY</b> relacionada.
<b>desplazamiento</b>	especifica el valor del ajuste que se puede agregar o restar al valor recuperado.
<b>HV_option</b>	<b>HV_option</b> puede ser uno de los siguientes: <b>/HI</b> : La posición horizontal inicial <b>/VI</b> : La posición vertical inicial <b>/H</b> : La posición horizontal final <b>/V</b> : La posición vertical final <b>/HD</b> : El desplazamiento horizontal <b>/VD</b> : El desplazamiento vertical

### Ejemplos

Este ejemplo traza a una línea horizontal debajo de la última línea.

```
{ORITL
1 0 /H RPEPOS      % final hor. pos. of "1 FROMLINE"
1 15 /V RPEPOS     % final ver. pos. of "1 FROMLINE"
2000 0 S1 DRAWB
}ENDPAGE
```

Este ejemplo dibuja un cuadro alrededor del grupo de líneas /ADR.

```
{ORITL
/ADR0 -50 /HI RPEPOS % init. hor. pos. of "/ADR0 RPEKEY"
/ADR0 -50 /VI RPEPOS % init. ver. pos. of "/ADR0 RPEKEY"
1000 % fixed width
/ADR0 100 /VD RPEPOS % ver. displ. of "/ADR0 RPEKEY"
S1 DRAWB
}ENDPAGE
```



Nota: Utilice GEPkeys opacas cuidadosamente porque las GEPkeys se superponen a del cuadro.

## Modos

Esta variable es aplicable en el modo de línea.

## Comandos relacionados

- BEGINRPE
- COPYRANGE
- ENDRPE
- FROMLINE
- INDEXRPE
- SETRPEPREFIX

## RPLEFT

**RPLEFT** es una variable integrada de VIPP® que está disponible solamente en un procedimiento **REPEAT** que proporciona el número de iteraciones restantes, incluida la actual del bucle **REPEAT**.

Para realizar determinadas acciones en determinadas iteraciones del procedimiento **REPEAT**, utilice la variable **RPLEFT** con **IF**, **ELSE**, **ENDIF** o **CASE**.

### Sintaxis

**RPLEFT**

### Ejemplos

```
{ IF RPLEFT 1 eq      % action on last iteration
  { do this }
  ENDIF
  ...
} REPEAT
```

### Modos

Esta variable es aplicable en todos los modos.

### Comandos relacionados

**REPEAT**, **RPCOUNT**



## SHEETH

**SHEETH** proporciona la altura de la página física actual.

### Sintaxis

SHEETH

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

SHEETW, PAGEW, PAGEH

## SHEETW

**SHEETW** proporciona la anchura de la página física actual.

### Sintaxis

SHEETW

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

SHEETH, PAGEW, PAGEH

## SHPOS

La variable dinámica integrada **SHPOS** proporciona la posición de impresión horizontal secundaria guardada por **SAVEPP**. Use **SHPOS** con **MOVETO**, **MOVEH**, **DRAWB**, **DRAWBR**, **DRAWBM**, **DRAWBRM** y **DRAWPOL**.

### Sintaxis

SHPOS

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

- **HDISP**
- **SAVEPP**
- **SVPOS**
- **VDISP**

## SLENGTH

**SLENGTH** proporciona la longitud o número de caracteres en una cadena. La fuente actual determina cómo se deben delimitar y contar los caracteres, especialmente para cadenas de varios bytes.

### Sintaxis

```
(string) SLENGTH
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SSIZE**

## SSIZE

**SSIZE** proporciona la altura o la anchura en unidades actuales de una cadena según las fuentes actuales y el modo de escritura actual.

### Sintaxis

```
(string) SSIZE
```

### Modos

Este comando es aplicable en todos los modos.

### Comandos relacionados

**SLENGTH**

## SVPOS

**SVPOS** proporciona la posición de impresión vertical guardada por **SAVEPP**. Use **SVPOS** con **MOVETO**, **DRAWB**, **DRAWBR**, **DRAWBM**, **DRAWBRM** y **DRAWPOL**.

### Sintaxis

SVPOS

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

- **HDISP**
- **SAVEPP**
- **SHPOS**
- **VDISP**

## T\_AMPM

**T\_AMPM** se usa en combinación con el comando **GETDATE** para configurar la hora adecuada para la aplicación. Esta variable se usa para configurar relojes de 12 horas en designaciones AM o PM.

### Sintaxis

T\_AMPM

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**

## T\_HH

**T\_HH** se usa en combinación con el comando **GETDATE** para configurar la hora o las horas adecuada(s) para la aplicación. Esta variable se utiliza para configurar relojes de 24 horas.

### Sintaxis

T\_HH

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**



## T\_HH2

**T\_HH2** se usa en combinación con el comando **GETDATE** para configurar la hora o las horas adecuada(s) para la aplicación. Esta variable se utiliza para configurar relojes de 12 horas.

### Sintaxis

T\_HH2

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**

## T\_MM

**T\_MM** se usa en combinación con el comando **GETDATE** para configurar el tiempo o los minutos correspondientes, 00-59, para la aplicación.

### Sintaxis

T\_MM

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**

## T\_SS

T\_SS se usa en combinación con el comando **GETDATE** para configurar el tiempo en segundos correspondiente, 00-59 para la aplicación.

### Sintaxis

T\_SS

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**

## T\_TZN

**T\_TZN** se usa en combinación con el comando **GETDATE** para configurar la zona horaria, como por ejemplo PST o PDT para la aplicación.

### Sintaxis

T\_TZN

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

**GETDATE**

## TLENGTH

**TLENGTH** ofrece la longitud o el número de elementos de una tabla.

### Sintaxis

```
VARTable_name TLENGTH
```

### Modos

Esta variable es aplicable en todos los modos.

### Comandos relacionados

- **SETVAR**
- **ADD**
- **GETITEM**
- **FOREACH**

## TLGRID

**TLGRID** activa la impresión de una muestra de trabajo VIPP® sobre una cuadrícula de unidades con el origen superior izquierdo. Esta cuadrícula puede resultar útil para la colocación de datos en las definiciones de **RPE**.

Esta función permite ubicar las posiciones de impresión en la página al codificar un trabajo de VIPP® como un formulario, **RPE** o modo nativo. La unidad utilizada es la unidad actual configurada por **SETUNIT**.

El origen 0,0 de la cuadrícula está vinculado a los márgenes. En la mayoría de los casos, se recomienda la configuración de los márgenes a 0.

### Sintaxis

```
(TLGRID) SETFORM
```

### Ejemplos

```
0 0 0 0 SETMARGIN
```

```
(TLGRID) SETFORM
```



Consejo: Para imprimir la cuadrícula encima de otros formularios, aumente **SETMAXFORM**. En sistemas de color de realce, la cuadrícula se imprime con el color de realce actual.

### Modos

Este formulario integrado es aplicable en todos los modos.

### Comandos relacionados

- **BLGRID**
- **SETFORM**
- **SETMARGIN**
- **SETMAXFORM**
- **SETUNIT**

## TPATH

**TPATH** es una GEPkey integrada reservada. Utilice **TPATH** para definir una ruta a lo largo de la cual **SHPATH** imprime el texto. **TPATH** se usa con un comando que requiere una GEPkey, como **DRAWB**, **DRAWC**, **DRAWBR**, **DRAWPOL** o **DRAWPATH**.

Se pueden definir varias rutas continuas o discontinuas antes de invocar a **SHPATH**.

Para borrar la ruta, utilice **ETCLIP**. La ruta se borra automáticamente al principio de cada página.

### Sintaxis

Como GEPkey:

```
TPATH                % defines a transparent path
[ GEPkey TPATH ]    % defines a path also painted by GEPkey
```

### Ejemplos

```
[ 200 200 800 1200 1400 200 ] [S1 TPATH] DRAWPOL
(Text to be printed along the triangle defined by DRAWPOL) 0 0 0 SHPATH
```

### Modos

Este GEPkey integrado es aplicable en todos los modos.

### Comandos relacionados

- **DRAWB y DRAWBR**
- **DRAWC**
- **DRAWPOL**
- **DRAWPATH y DRAWPATHR**
- **ETCLIP**

## UV2L para efecto UV de dos capas

**UV2L** es una opción de ocultación en el comando **SETTXC** que se usa para crear un efecto UV de dos capas encima del efecto UV habitual.

### Ejemplos

```
103 3342 1116 149 UV_GOLDENROD2 DRAWB % draw UV box
/NHEB 24 SETFONT % select font
659 3213 MOVETO
(Hello world) SHc % print UV text
[YELLOW UV2L] SETTXC % select color and mask for 2-layer UV effect
268 3242 MOVETO
(Goodbye world) SH % print UV 2-layer text
```

Para lograr el resultado esperado, seleccione el color de la segunda capa en función del enlace UV. Para seleccionar una combinación del color UV y del color de la segunda capa de cian, magenta o amarillo, consulte las páginas de la 7 a la 9 del archivo de muestra `xgf/demo/VIS_UVSamples.nm`.



## VARDataFileName

**VARDataFileName** es una variable integrada con el nombre del archivo de envío. Cuando el nombre no está disponible, el valor predeterminado es desconocido. Solo VI Design Pro, VI Explorer y VI eCompose proporcionan el nombre del archivo de envío.

### Sintaxis

```
(Data File Name: [=VARDataFileName=]) VSUB SH
```

## VDISP

**VDISP** proporciona el desplazamiento entre la posición actual de impresión vertical y las posiciones de impresión guardadas por **SAVEPP**. Use **VDISP** con **MOVETO**, **DRAWB**, **DRAWBR**, **DRAWBM**, **DRAWBRM** y **DRAWPOL**. Para dibujar cuadros variables, **VDISP** se utiliza a menudo con **SAVEPP**, **SHPOS** y **SVPOS**.

### Sintaxis

VDISP

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

- **HDISP**
- **SAVEPP**
- **SHPOS**
- **SVPOS**

## VPOS

VPOS proporciona la posición de impresión vertical actual.

### Sintaxis

VPOS

### Modos

Esta variable dinámica integrada es aplicable en todos los modos.

### Comandos relacionados

- HPOS
- HPOS2
- SVPOS
- HDISP
- SAVEPP
- SHPOS
- VDISP

## XGFVER

**XGFVER** proporciona una cadena que muestra el número de lanzamiento de VI Compose. **XGFVER** requiere que lo utilice con un comando **SHx**.

### Sintaxis

XGFVER

### Ejemplos

Este ejemplo imprime el nivel de VI Compose que está instalado actualmente en el dispositivo.

```
XGFVER SHL
```

### Modos

Esta cadena integrada es aplicable en todos los modos.

### Comandos relacionados

Ninguno

## XMLATN

**XMLATN** proporciona el nombre del atributo XML y generalmente se combina con **XMLATV**.

### Sintaxis

Consulte **XMLATL**.

### Modos

Esta variable integrada es aplicable en el modo de XML.

### Comandos relacionados

- **BTA**
- **BTS**
- **ETA**
- **ETS**

## XMLATV

**XMLATV** proporciona el valor del atributo XML y generalmente se combina con **XMLATN**.

### Sintaxis

Consulte **XMLATL**.

### Modos

Esta variable integrada es aplicable en el modo de XML.

### Comandos relacionados

- **BTA**
- **BTS**
- **ETA**
- **ETS**

## XMLDTH

**XMLDTH** requiere que coloque el comando en un par de comandos **BTA/ETA** o **BTS/ETS**. **XMLDTH** proporciona la profundidad del nodo actual.

### Sintaxis

XMLDTH

### Modos

Esta variable integrada es aplicable en el modo de XML.

### Comandos relacionados

- **BTA**
- **BTS**
- **ETA**
- **ETS**

## XMLPAR

**XMLPAR** requiere que coloque el comando en un par de comandos **BTA/ETA** o **BTS/ETS**. **XMLPAR** proporciona el nombre del nodo primario del nodo actual.

### Sintaxis

XMLPAR

### Modos

Esta variable integrada es aplicable en el modo de XML.

### Comandos relacionados

- **BTA**
- **BTS**
- **ETA**
- **ETS**



## XMLPATH

**XMLPATH** requiere que coloque el comando en un par de comandos **BTA/ETA** o **BTS/ETS**. **XMLPATH** proporciona la VVpath del nodo actual.

### Sintaxis

XMLPATH

### Modos

Esta variable integrada es aplicable en el modo de XML.

### Comandos relacionados

- **BTA**
- **BTS**
- **ETA**
- **ETS**

## XMLTAG

Para usar **XMLTAG**, coloque el comando en un par de comandos **BTA/ETA** o **BTS/ETS**. **XMLTAG** proporciona el nombre del nodo actual.

### Sintaxis

XMLTAG

### Modos

Esta variable integrada es aplicable en el modo de XML.

### Comandos relacionados

- **BTA**
- **BTS**
- **ETA**
- **ETS**

## XMLVAL

Para usar **XMLDTH**, coloque el comando en un par de comandos **BTA/ETA** o **BTS/ETS**. **XMLDTH** proporciona la profundidad del nodo actual.

### Sintaxis

XMLVAL

### Modos

Esta variable integrada es aplicable en el modo de XML.

### Comandos relacionados

- **BTA**
- **BTS**
- **ETA**
- **ETS**

## YINIT

**YINIT** es una variable dinámica integrada que proporciona la posición vertical actual definida por **SETGRID** a una entrada **RPE** en una definición **FROMLINE**. La variable **YINIT** permite la interacción entre **RPE** y **SETGRID**. **YINIT** se utiliza con frecuencia en combinación con **LSP**.

### Sintaxis

```
YINIT
```

```
/YINIT
```

Si **SETGRID** puede variar durante el trabajo, utilice la segunda sintaxis.

### Ejemplos

El siguiente es un ejemplo de la configuración de formato, generado por **SETGRID**, que imprimirá las dos primeras líneas con fuente negrita en un fondo cebra.

```
80 60 SETGRID
LMEDIUM 1 1 SETZEBRA
100 200 200 200 SETMARGIN
/F1 /NCR 0 INDEXFONT
/F2 /NCRB 0 INDEXFONT
2 BEGINRPE
1 FROMLINE [ 0 0 0 nu]] YINIT LSP 0 80 /F2 BLACK ]
3 FROMLINE [ 0 0 0 nu]] YINIT LSP 0 80 /F1 BLACK ]
ENDRPE
```

### Modos

Esta variable es aplicable en el modo de línea.

### Comandos relacionados

**FROMLINE**, **SETGRID**, **LSP**

## ZSPAGE

**ZSPAGE** es una variable integrada usada con la base de datos **ZSORT** y solo en el **DBM**, para producir el número de página del documento, a partir del número 1. Para garantizar la compatibilidad con versiones anteriores, se puede usar la variable **BACK** en documentos de 2 caras a 2 páginas.

Consulte la carpeta `xgf/demo` para analizar el archivo `ZSORT_Example.vpc`, que contiene información acerca del uso de **ZSPAGE** con **ZSORT**.

## ZSRECNUM

**ZSRECNUM** es una variable integrada que se utiliza con solo la base de datos **ZSORT** y en la **DBM** para producir números de registros en el orden del archivo de datos. Este es el número de registro actual, aumentado desde el valor de 0.

Consulte la carpeta `xgf/demo` para analizar el archivo `ZSORT_Example.vpc`, que contiene información acerca del uso de **ZSRECNUM** con **ZSORT**.

## ZSREPCNT

**ZSREPCNT** es una variable integrada que se utiliza con solo la base de datos **ZSORT** y en la **DBM** para producir un recuento de repeticiones continuo, empezando por 0.

Consulte la carpeta `xgf/demo` para analizar el archivo `ZSORT_Example.vpc`, que contiene información acerca del uso de **ZSREPCNT** con **ZSORT**.

## ZSREPIDX

**ZSREPIDX** es una variable integrada que se utiliza con solo la base de datos **ZSORT** y en la **DBM** para producir un índice de repeticiones de registros. **ZSREPIDX** se incrementa para cada instancia pero se restablece a 0 al principio del siguiente registro.

Consulte la carpeta `xgf/demo` para analizar el archivo `ZSORT_Example.vpc`, que contiene información acerca del uso de **ZSREPIDX** con **ZSORT**.



# Parámetros

Este capítulo incluye:

Categorías del parámetro .....	626
Descripción del parámetro .....	627

Un parámetro VIPP® es una palabra clave que representa un valor interno que afecta al comportamiento de uno o varios comandos o funciones de transformación. Los parámetros solo pueden configurarse con el comando SETPARAMS

El parámetro opcional de los comandos DRAWBAR, DRAWCRV, DRAWPIE y SETPARAMS está disponible de tres maneras:

- /code
- [/param1 valor1 /param2 valor2..../paramN valorN]
- [/code /param1 valor1 /param2 valor2..../paramN valorN]

Donde:

<b>/code</b>	es un valor numérico calculado mediante la agregación de todos los códigos relacionadas con los parámetros necesarios. Los valores de los parámetros activados por el código se indican en la tabla siguiente (imprima <code>xgf /demo /samddg.ps</code> para obtener una muestra completa de códigos).
<b>/paramX valorX</b>	son parejas de teclas y valores de la tabla siguiente que se utilizan para cambiar el valor prefijado de un determinado parámetro.

Todos los parámetros configurados por DRAWBAR, DRAWCRV o DRAWPIE pueden sustituir temporalmente el valor prefijado definido por SETPARAMS y solo se aplicarán a ese comando. Se restauran los valores predeterminados para comandos posteriores.

## Ejemplos

Parámetros opcionales que producen resultados equivalentes:

```
/67  
[/66 /3D true ] [/3D true /SliceSepWidth 0 /PrintValue true ]
```

## Categorías del parámetro

Los parámetros se utilizan con varios comandos diferentes y se pueden agrupar en las siguientes categorías:

- Parámetros de gráficos, que se aplican a **DRAWPIE**, **DRAWBAR** y **DRAWCRV**
- Parámetros de formato, que se aplican al comando **FORMAT** y **Format**
- Parámetros de impresión a dos caras, que se aplican a **DUPLEX\_on** y **DUPLEX\_off**
- Parámetros de folleto, que se aplican a **STARTBOOKLET** y **ENDBOOKLET**
- Parámetros de reimpresión de trabajos, que se aplican a **PAGERANGE** y **BOOKLETRANGE**
- Parámetros de representación de la imagen, que se aplican a **ICALL** y **CACHE SCALL**
- Parámetros de selección de material de impresión, que se aplican a **SETMEDIA**
- Parámetros de migración de **LCDS**, que se aplican a **SETFORM** y **SETBFORM**
- Parámetros **PIF**, que se aplican a **SETPIF**, **INDEXPIF** y **BOOKMARK**
- Parámetros de almacenamiento en caché, que se aplican a la memoria **CACHE**, **PRECACHE** y **FSHOW**
- Parámetros de lectura de marca óptica (OMR), que se aplican a **FILLOMR**.
- Parámetros de fecha y hora, que se aplican a **GETDATE**.
- Parámetros de diseño, que se aplican a los comandos de diseño de la página lógica.
- Parámetros de composición de página, que se aplican a los comandos de formato de la página.
- Parámetros de código de barras, que se aplican a los comandos de código de barras.

Los parámetros que no encajan en otras categorías definidas se muestran como Varios.

## Descripción del parámetro

Las descripciones de parámetros de VIPP® incluyen esta información:

- Nombre del parámetro
- Categoría
- Valor predeterminado
- Unidad o formato en el que se especifica el parámetro
- Código
- Valor de código
- Descripción

Si cualquier parte de esta lista se omite de una determinada descripción de un parámetro, puede deducir que no es preciso especificar ningún valor.

### /3D

<b>Categoría:</b> Gráfico
<b>Valor predeterminado:</b> falso
<b>Unidad o formato:</b> verdadero/falso
<b>Código:</b> +1/
<b>Valor de código:</b> verdadero
<b>Descripción:</b> Efecto 3D

### /3DANGLE

#### Categoría

Gráfico

#### Valor predeterminado

0.3

#### Unidad o formato

Entero real entre 0 y 2

#### Descripción

- Ángulo de visión 3D
- **0 a 1:** Mostrar el lado derecho de las barras
- **1 a 2:** Mostrar el lado izquierdo de las barras

### /3DTHICKNESS

**Categoría:** Gráfico

**Valor predeterminado:** 0.35

**Unidad o formato:**

- % de radio
- % de ancho de barra
- % de anchura/5

**Descripción:** Grosor (profundidad) cuando /3D es verdadero Consulte [/KeepRatio](#).

### /ACROSS

**Categoría:** Diseño

**Valor predeterminado:** 1

**Unidad o formato:** entero

**Descripción:** El número de páginas lógicas a través de las hojas impresas en una aplicación de varias en 1.

### /ALIGN

**Categoría:** Tabla

**Valor predeterminado:** 0

**Unidad o formato:** entero

**Descripción:** alinear atributo; igual que SHP

### /ALIGNCHAR

**Categoría:** Composición de página

**Valor predeterminado:** valor de /DecimalPoint

**Unidad o formato:** Código ASCII

**Descripción:** Define los caracteres de alineación alternativos de SHMF align=4.

**Valores posibles:** Cualquier código ASCII, por ejemplo, 0-255

**/AMPM**

<b>Categoría:</b>	Fecha
<b>Unidad o formato:</b>	Matriz de dos cadenas.
<b>Descripción:</b>	Indicadores AM y PM

**Ejemplos**

[ (a.m.) (p.m.) ]

**/BARSPACE**

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	0.4
<b>Unidad o formato:</b>	% de ancho de barra
<b>Código:</b>	+32/
<b>Valor de código:</b>	0
<b>Descripción:</b>	Espacio entre las barras como porcentaje de la anchura de la barra.
<b>Valores posibles:</b>	<p><b>0:</b> sin espacio entre las barras.</p> <p><b>1:</b> el espacio entre las barras es igual a la anchura de las barras.</p> <p><b>10:</b> el espacio entre las barras es 10 veces la anchura de las barras.</p>

**/BARSPACE2**

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Establece el espaciado entre un grupo de barras.
<b>Valores posibles:</b>	Es igual a /BarSpace

**/BGCOLOR**

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	BLANCO

<b>Unidad o formato:</b>	Colorkey o null
<b>Código:</b>	+16
<b>Valor de código:</b>	.99 .95 .70
<b>Descripción:</b>	Color de fondo de los gráficos de barra y curva. Para transparencias, utilice nulo en lugar de un Colorkey.

#### /BGLINECOLOR

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	BLANCO
<b>Unidad o formato:</b>	Colorkey
<b>Descripción:</b>	Color para las líneas de escala de fondo horizontales. Valores posibles: Cualquier Colorkey normal.

#### /BOOKLETMISMATCH

<b>Categoría:</b>	Folleto
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Establece la acción si el número de páginas del folleto no coincide con el parámetro PagesPerBooklet.
<b>Valores posibles:</b>	<p><b>0:</b> Omitir discordancia</p> <p><b>1:</b> Agregar páginas hasta que coincidan con las páginas del folleto.</p> <p><b>2:</b> Cancelar y enviar un mensaje de error de VI Compose.</p> <p><b>3:</b> Agregar páginas con el formulario actual hasta que el número de páginas coincida.</p> <p><b>4:</b> Agregar páginas en blanco hasta PagesPerBooklet cuando el número de páginas en el folleto es inferior a PagesPerBooklet.</p> <p><b>5:</b> Agregue páginas con el formulario actual a PagesPerBooklet si el número de páginas en el folleto es inferior a PagesPerBooklet.</p>

**/BOTTOMBLEED**

<b>Categoría:</b>	Diseño
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	El valor de la cantidad de sangrado permitido en el margen inferior de la página lógica en una aplicación de varios en 1.

**/BURSTLIST**

<b>Categoría:</b>	Gráfico
<b>Unidad o formato:</b>	[(label1) (label2) ...]
<b>Descripción:</b>	Lista de las etiquetas que se pueden separar si /SliceBurst no es cero.
<b>Valores posibles:</b>	Lista de etiquetas en la lista de etiquetas/valores.

**/CACHEICALL**

<b>Categoría:</b>	Almacenamiento en caché
<b>Valor predeterminado:</b>	falso
<b>Unidad o formato:</b>	booleano
<b>Valores posibles:</b>	<b>falso:</b> No se produce el almacenamiento en caché si utiliza <b>ICALL</b> . <b>verdadero:</b> Permite el almacenamiento en caché de forma transparente si se utiliza <b>ICALL</b> .

## Ejemplos

```
[ /CacheICALL true ] SETPARAMS % Enable Caching
                                % on ICALL.
```

**/CACHING**

<b>Categoría:</b>	Almacenamiento en caché
<b>Valor predeterminado:</b>	12
<b>Unidad o formato:</b>	entero

<b>Descripción:</b>	<p>Las opciones de almacenamiento en caché son:</p> <p><b>0</b> sin almacenamiento en caché (se puede utilizar CACHE pero no se producirá ningún almacenamiento en caché).</p> <p><b>Almacenamiento en caché de Adobe (se producirá según las especificaciones de Adobe).</b></p> <p><b>1</b> almacenamiento en caché a través de CACHE y PRECACHE.</p> <p><b>2</b> almacenamiento en caché a través de CACHE, PRECACHE y FSHOW.</p> <p><b>Almacenamiento en caché de VIPP® (se producirá según las especificaciones de VIPP®).</b></p> <p><b>11</b> almacenamiento en caché a través de CACHE y PRECACHE.</p> <p><b>12</b> almacenamiento en caché a través de CACHE, PRECACHE y FSHOW (valor predeterminado).</p>
---------------------	---



Nota: Si el almacenamiento en caché de VIPP® no se implementa en el intérprete de PostScript en el que se ejecuta VIPP®, automáticamente se pasa al almacenamiento en caché de Adobe de forma predeterminada.

## /CELLIMAGE

<b>Categoría:</b>	Tabla
<b>Unidad o formato:</b>	(nombre del archivo de imagen)
<b>Descripción:</b>	imagen que se debe colocar en la celda

## /CELLFILL

<b>Categoría:</b>	Tabla
<b>Unidad o formato:</b>	Nombre del color
<b>Descripción:</b>	Define el color que se usa para rellenar la celda.

## /CELLSTROKE

<b>Categoría:</b>	Tabla
<b>Unidad o formato:</b>	[superior inferior izquierda derecha] (matriz o GEPkeys)
<b>Descripción:</b>	Define las GEPkeys para trazar los bordes de la celda.



**/CELLTEXT**

<b>Categoría:</b>	Tabla
<b>Unidad o formato:</b>	(texto)
<b>Descripción:</b>	Texto que se va a introducir en la celda.

**/CHARTDIR**

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Dirección del gráfico de <b>DRAWBAR</b> y <b>DRAWCRV</b> .
<b>Valores posibles:</b>	<b>0:</b> de abajo a arriba <b>1:</b> de izquierda a derecha <b>2:</b> de arriba a abajo <b>3:</b> de derecha a izquierda

**/CHARTORDER**

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	<b>DRAWBAR</b> y <b>DRAWCRV</b> solamente.
<b>Valores posibles:</b>	<b>0:</b> Imprimir los elementos de izquierda a derecha o de arriba a abajo. <b>1:</b> Imprimir los elementos de derecha a izquierda o de abajo a arriba.

**/CHECKLABELOVERLAP**

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	verdadero
<b>Unidad o formato:</b>	booleano

<b>Descripción:</b>	Activa y desactiva el control de superposición de etiquetas de gráficos circulares.
<b>Valores posibles:</b>	<b>verdadero:</b> Impide la superposición de etiquetas. <b>falso:</b> No impide la superposición de etiquetas.

#### /CHKRESOURCES

<b>Categoría:</b>	Comprobación de recursos
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Este parámetro admite la compatibilidad con versiones anteriores. Para obtener información de recursos, consulte la función demográfica descrita en la <i>Guía del usuario de FreeFlow VI Compose</i> .

#### /CJKUNITCOUNT

<b>Categoría:</b>	Varios
<b>Valor predeterminado:</b>	falso
<b>Unidad o formato:</b>	verdadero o falso
<b>Descripción:</b>	Establecer método de recuento de caracteres de varios bytes, como por ejemplo para chino, japonés o coreano.
<b>Valores posibles:</b>	<b>falso:</b> Cada carácter de varios bytes es una unidad. <b>verdadero:</b> Solo se aplica a SETRCD, SETPCD, GETFIELD y RPE. Un carácter de un solo byte es una unidad. Un carácter de varios bytes son dos unidades.

#### /CLEARSUBST

<b>Categoría:</b>	Interpretación de imagen
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Establece la reproducción de tinta transparente.

<b>Valores posibles:</b>	<p>Se admiten los siguientes valores:</p> <p><b>0</b> estrategia de mejor ajuste (predeterminado):</p> <p>Si hay la opción Transparente, se aplica.</p> <p>En VDP/VIE/VDEP, la impresora de escritorio sustituye el patrón de texto.</p> <p>En impresoras de producción (FFPS...) sin la opción Transparente, se omite.</p> <p>En FFPS/Normalizador, conserve o ignore esta opción (en función de si se cuenta con la opción Transparente).</p> <p>En VIEc, ignore esta opción</p> <p><b>1</b> igual que 0, pero consévelo en VIEc (PDF para la impresión)</p> <p><b>2</b> ignore la opción en todos los dispositivos excepto si se cuenta con la opción Transparente</p> <p><b>3</b> sustituya el patrón de texto en todos los dispositivos excepto si se cuenta con la opción Transparente</p> <p><b>4</b> conserve esta opción en todos los dispositivos</p> <p><b>5</b> ignore la opción en todos los dispositivos incluso si se cuenta con la opción Transparente</p> <p><b>6</b> sustituya el patrón de texto en todos los dispositivos incluso si se cuenta con la opción Transparente</p>
--------------------------	---



Nota: La compatibilidad de las tintas especiales de Xerox de VIPP® asume que el color Transparente está disponible como color directo de nombre Transparente.

## Ejemplos

```
[ /ClearSubst 3 ] SETPARAMS % substitute text pattern
% (for proof printing)
```

## /CLUSTERMODE

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Establece el modo de presentación de un grupo de barras.
<b>Valores posibles:</b>	<p><b>0:</b> barras apiladas</p> <p><b>1:</b> barras de lado a lado</p>

## /COLORCYCLE

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Establece el modo de ciclo de color.
<b>Valores posibles:</b>	<p>0: automático; generalmente el valor más apropiado.</p> <p>1: repetir un elemento, grupo</p> <p>2: repetir todo un gráfico</p>

## /COLORTABLE

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	XLGREEN LRED LGREEN RED MBLUE LMGREEN XDBLUE MRED DGREEN LBLUE
<b>Unidad o formato:</b>	[Colorkeys...]
<b>Descripción:</b>	<p>Lista de Colorkeys utilizadas de manera cíclica.</p> <p>Para transparencias, utilice nulo en lugar de un Colorkey.</p>

## /DAYLIGHTSAVING

<b>Categoría:</b>	Fecha
<b>Unidad o formato:</b>	matrices de matrices de 6 enteros
<b>Descripción:</b>	Hora de inicio y final del horario de verano.

### Ejemplos

```
[ year +-hrs startday starttime endday endtime ]
```

## /DAYS LONG

<b>Categoría:</b>	Fecha
<b>Unidad o formato:</b>	Matriz de siete cadenas.
<b>Descripción:</b>	Lista de nombres de los días en formato largo, a partir del domingo.

**/DAYSSHORT**

<b>Categoría:</b>	Fecha
<b>Unidad o formato:</b>	Matriz de siete cadenas.
<b>Descripción:</b>	Lista de nombres de los días en formato corto, a partir del domingo.

**/DECIMALPOINT**

<b>Categoría:</b>	Formato
<b>Valor predeterminado:</b>	. (EE. UU.) , (A4)
<b>Unidad o formato:</b>	Código ASCII
<b>Descripción:</b>	Delimitador decimal en datos numéricos.

**/DEFAULTDATE**

<b>Categoría:</b>	Fecha
<b>Valor predeterminado:</b>	[2003 1 1 00 00 0 0]
<b>Unidad o formato:</b>	matriz de 7 enteros
<b>Descripción:</b>	Fecha que se utiliza si no hay ningún sistema de archivos disponible.

**Ejemplos**

[ year month day hrs mns sec daylightsaving (0/1) ]

**/DEFINEDDATE**

<b>Categoría:</b>	Fecha
<b>Valor predeterminado:</b>	matriz de 6 enteros
<b>Unidad o formato:</b>	Si está presente, esta fecha sustituye la fecha del sistema
<b>Descripción:</b>	[ year month day hrs mns sec ]

### /DOTSPERMODULE

<b>Categoría:</b>	Código de barras
<b>Valor predeterminado:</b>	3
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	El número de puntos por barra o espacio en el código de barras.

### /DOWN

<b>Categoría:</b>	Diseño
<b>Valor predeterminado:</b>	1
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	El número de páginas lógicas a través de las hojas impresas en una aplicación de varias en 1.

### /DRAWMODE

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Admite el trazo o el relleno del área radial (DDG).



Nota: Para permitir una opción distinta para cada área de apilado, utilice la segunda sintaxis.

### Ejemplos

```
/DrawMode opt
```

o

```
/DrawMode [ opt1 opt2 ... optn ]
```

#### Donde:

optx es uno de:

- 0 trazar el área radial (predeterminado)
- 1 rellenar el área radial

**/EMPTYJOBREPORT**

<b>Categoría:</b>	Varios
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	<p>Este parámetro se utiliza para indicar si un trabajo que está vacío (por ejemplo un trabajo que no produce ninguna página) es válido o no.</p> <p>0 - desconocido</p> <p>1 - un trabajo que no produce ninguna página es válido</p> <p>2 - un trabajo que no produce ninguna página no es válido</p>



Nota: Este parámetro tiene como objetivo principal asistir a las aplicaciones que producen archivos PDF, como VIEC o FFCORE, para retornar los mensajes correspondientes en la interfaz de usuario.

**Ejemplos**

[ /EmptyJobReport 1 ] SETPARAMS% an empty job is valid.

**/ELEVEL**

<b>Categoría:</b>	Código de barras
<b>Valor predeterminado:</b>	23
<b>Unidad o formato:</b>	un entero entre 0 y 99
<b>Descripción:</b>	Nivel de corrección de errores

**/EXTRASPACE**

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	1
<b>Unidad o formato:</b>	% de tamaño de la fuente
<b>Descripción:</b>	Espacio adicional superior e inferior si /FitSpace > 0.

## /FDECIMALPOINT

<b>Categoría:</b>	Formato
<b>Valor predeterminado:</b>	. (EE. UU.) , (A4)
<b>Unidad o formato:</b>	Código ASCII
<b>Descripción:</b>	Delimitador decimal en formato.

## /FDIGIT

<b>Categoría:</b>	Formato
<b>Valor predeterminado:</b>	#
<b>Unidad o formato:</b>	Código ASCII
<b>Descripción:</b>	Marcador de posición de dígitos en el formato.

## /FILLORDER

<b>Categoría:</b>	Diseño
<b>Descripción:</b>	El orden de llenado de las páginas lógicas.
<b>Valores posibles:</b>	<p>/RD (derecha-abajo) de izquierda a derecha y, a continuación, de arriba a abajo</p> <p>/LD (izquierda-abajo) de derecha a izquierda y, a continuación, de arriba a abajo</p> <p>/RU (derecha-arriba) de izquierda a derecha y, a continuación, de abajo a arriba</p> <p>/LU (izquierda-arriba) de derecha a izquierda y, a continuación, de abajo a arriba</p> <p>/DR (abajo-derecha) de arriba a abajo y, a continuación, de izquierda a derecha</p> <p>/UR (arriba-derecha) de abajo a arriba y, a continuación, de izquierda a derecha</p> <p>/DL (abajo-izquierda) de arriba a abajo y, a continuación, de derecha a izquierda</p> <p>/UL (arriba-izquierda) de abajo a arriba y, a continuación, de derecha a izquierda</p>



**/FITSPLACE**

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	1
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Ajuste el espacio RPE si se usa ddg_index
<b>Valores posibles:</b>	<p>0 tamaño fijo</p> <p>+1 ajustar al espacio RPE</p> <p>+2 cancelar 3D si el radio de llenado es inferior al máximo</p> <p>+4 forzar 3D si el ajuste del radio es mayor que el máximo</p> <p>+8 ajustar el grosor de 3D para un mejor ajuste</p>

**/FLZDIGIT**

<b>Categoría:</b>	Formato
<b>Valor predeterminado:</b>	@
<b>Unidad o formato:</b>	Código ASCII
<b>Descripción:</b>	Marcador de posición de un dígito en un formato que se sustituye por un espacio si el dígito es un cero inicial.

**/FN SIGN**

<b>Categoría:</b>	Formato
<b>Valor predeterminado:</b>	-
<b>Unidad o formato:</b>	Código ASCII
<b>Descripción:</b>	Signo negativo en el formato.

## /FORMAT

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	cadena
<b>Descripción:</b>	Si /PrintValue es verdadero, use este parámetro para especificar el formato de impresión de la escala y los valores de la etiqueta. La cadena asociada con el parámetro/Format sigue las reglas que se describen en el comando de formato <b>FORMAT</b> . Los modificadores de atributos de texto, por ejemplo fuente, color, etc., se admiten en la cadena de formato.

## /FORMATV

<b>Categoría:</b>	Gráfico
<b>Unidad o formato:</b>	cadena
<b>Descripción:</b>	Si /PrintValue es verdadero, use este parámetro para especificar el formato de los valores. La cadena asociada con el parámetro /FormatV sigue las reglas que se describen en el comando <b>FORMAT</b> . La cadena puede incluir cambios de fuente y color configurados por INDEXFONT e INDEXCOLOR.

## /FORMATVI

<b>Categoría:</b>	Gráfico
<b>Unidad o formato:</b>	cadena
<b>Descripción:</b>	Si /PrintValue es verdadero, use este parámetro para especificar el formato de los valores individuales. La cadena asociada con el parámetro /FormatVI sigue las reglas que se describen en el comando <b>FORMAT</b> . La cadena puede incluir cambios de fuente y color configurados por INDEXFONT e INDEXCOLOR.

## /FORMATPC

<b>Categoría:</b>	Gráfico
<b>Unidad o formato:</b>	cadena
<b>Descripción:</b>	Si /PrintValue es verdadero (true), use este parámetro para especificar el formato de impresión de los porcentajes. La cadena asociada con el parámetro/Format sigue las reglas que se describen en el comando de formato <b>FORMAT</b> .

**/FPSIGN**

Categoría:	Formato
Valor predeterminado:	+
Unidad o formato:	Código ASCII
Descripción:	Signo positivo en el formato.

**/FPUNCTUATION**

Categoría:	Formato
Valor predeterminado:	, (EE. UU.) .(A4)
Unidad o formato:	Código ASCII
Descripción:	Delimitador de miles en el formato.

**/HALFPIE**

Categoría:	Gráfico
Valor predeterminado:	0
Unidad o formato:	entero
Código:	+512
Valor de código:	1
Código:	+1024
Valor de código:	-1
Valores posibles:	<b>0</b> Gráfico circular completo <b>1</b> Mitad superior <b>-1</b> Mitad inferior

**/HEIGHT**

Categoría:	Tabla
Valor predeterminado:	0

<b>Unidad o formato:</b>	real
<b>Descripción:</b>	Altura mínima de la celda de una tabla. El valor predeterminado son las unidades actuales.

#### /HGUTTER

<b>Categoría:</b>	Diseño
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	El valor de margen interior horizontal entre las páginas lógicas en una aplicación de varias en 1.

#### /IMAGEDEFRES

<b>Categoría:</b>	Interpretación de imagen
<b>Valor predeterminado:</b>	300
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Configure la resolución de imagen predeterminada en puntos por pulgada si no figura en los parámetros TIFF y JPEG.

#### /IMP MODE:

<b>Categoría:</b>	Imposición
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Establecer el orden de <b>BEGINIMP</b> y <b>ENDIMP</b>
<b>Valores posibles:</b>	<b>0:</b> orden de 1 a N <b>1:</b> orden de N a 1

#### /INTERPOLATE

<b>Categoría:</b>	Interpretación de imagen
<b>Valor predeterminado:</b>	falso

<b>Unidad o formato:</b>	verdadero/falso
<b>Descripción:</b>	Solicite la interpolación de la imagen y calidad mejorada en imágenes en color.



Nota: Este parámetro puede afectar al rendimiento del trabajo. Para minimizar el impacto en el trabajo, utilice el parámetro con el comando **CACHE**.

#### /KEEPRATIO

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	verdadero
<b>Unidad o formato:</b>	verdadero/falso
<b>Código:</b>	+256
<b>Valor de código:</b>	falso
<b>Valores posibles:</b>	<b>true</b> El grosor 3D se basa en la anchura de la barra <b>false</b> El grosor 3D se basa en anchura/5

#### /LABELCOLW

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	0.2
<b>Unidad o formato:</b>	Real
<b>Descripción:</b>	La anchura de la columna en el porcentaje de la anchura del gráfico del ajuste de etiquetas de gráficos horizontales (ChartDir = 1/3).

#### /LABELDASHCOLOR

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	Negro
<b>Unidad o formato:</b>	Colorkey
<b>Descripción:</b>	Color de guiones de la etiqueta (WHITE = sin guiones).

#### /LABELDASHWIDTH

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	0.5
<b>Unidad o formato:</b>	puntos
<b>Descripción:</b>	Anchura de línea de guiones de la etiqueta.

#### /LABELOFFSET

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	0.1
<b>Unidad o formato:</b>	% de radio
<b>Descripción:</b>	Desplazamiento de las etiquetas desde el gráfico circular (si SpotSize = 0).

#### /LAYOUTMARKS

<b>Categoría:</b>	Diseño
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero

<b>Descripción:</b>	Elija una de estas opciones:
<b>Valores posibles:</b>	<p><b>0</b> ninguna marca (predeterminado)</p> <p><b>1</b> marcas de recorte</p> <p><b>2</b> marcas de sangrado</p> <p>+ opción para imprimir las marcas en la portada, la contraportada o en ambas</p> <p><b>+0</b> marcas de impresión solo en la portada</p> <p><b>+10</b> marcas de impresión solo en la contraportada</p> <p><b>+20</b> marcas de impresión en la portada y la contraportada</p> <p>Para imprimir marcas de recorte solo en la contraportada, introduzca: <code>/LayoutMarks 11</code></p> <p><b>+100</b> desactivar marcas de recorte internas</p> <p><b>+200</b> desactivar marcas de sangrado internas</p> <p><b>+300</b> desactivar las marcas de sangrado y recorte internas</p> <p>Para imprimir marcas de recorte solo en la contraportada, con las marcas internas desactivadas, introduzca: <code>/LayoutMarks 111</code></p>

#### /LCDSMODE

<b>Categoría:</b>	Migración de LCDS
<b>Valor predeterminado:</b>	falso
<b>Unidad o formato:</b>	verdadero/falso
<b>Descripción:</b>	Cuando se configuran como verdadero, SETFORM y SETBFORM aplican todos los formularios a las páginas físicas en vez de a las páginas lógicas. Utilice esta opción para facilitar la migración de LCDS a VIPP®.



Nota: El comando **PROCESSDJE** configura automáticamente este valor a verdadero.

#### /LEFTBLEED

<b>Categoría:</b>	Diseño
<b>Valor predeterminado:</b>	0

<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	El valor de la cantidad de sangrado permitido en el margen izquierdo de la página lógica en una aplicación de varios en 1.

#### /LINEDASH

<b>Categoría:</b>	Gráfico
<b>Unidad o formato:</b>	[ On1 Off1 On2 Off2 ... ]
<b>Descripción:</b>	Define los guiones de la línea de <b>DRAWCRV</b> . El tamaño es en puntos de segmentos consecutivos continuos y en blanco.
<b>Valores posibles:</b>	matriz de números reales.

#### /LOCALTOUTF8

<b>Categoría:</b>	Varios
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Solo modo de base de datos. Proporciona la conversión automática de la codificación local a UTF8. Esta opción tiene como objetivo principal el envío directo de archivos de datos con la codificación local que hace referencia a un proyecto diseñado mediante VI Design Express.





Nota: El uso de este parámetro elimina la necesidad de convertir el archivo de datos antes del envío.

<b>Valores posibles:</b>	<b>0</b> sin conversión (predeterminado) <b>1</b> ISO-8859-1 o Windows-1252 (europeo occidental) <b>2</b> ISO-8859-2 (europeo central) <b>3</b> Windows-1259 (latino-2) <b>4</b> ISO-889-9 (turco) <b>5</b> Windows-1251 (cirílico) <b>6</b> Windows-1258 (vietnamita) <b>7</b> Windows-CP874/TIS-6 <b>8</b> Windows-CP866 (cirílico) <b>9</b> IDSO-8858-15 (latino-9) <b>10</b> Mac OS Roman <b>11</b> Windows-1256 (árabe) <b>12</b> Windows-1255 (hebreo) <b>13</b> UTF16
--------------------------	---

## /MARGINS

<b>Categoría:</b>	Tabla
<b>Valor predeterminado:</b>	0[0 0 0 0]
<b>Unidad o formato:</b>	[superior inferior izquierda derecha] (matriz o reales)
<b>Descripción:</b>	valor de los márgenes de celda de la tabla en unidades reales

## /MARKLENGTH

<b>Categoría:</b>	Diseño
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	La longitud de las marcas de diseño en una aplicación de varias en 1.

#### /MARKOFFSET

<b>Categoría:</b>	Diseño
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	El desplazamiento de las marcas de diseño desde la esquina de la página.

#### /MARKWIDTH

<b>Categoría:</b>	Diseño
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	La anchura de las marcas de diseño en una aplicación de varias en 1.

#### /MAXHEIGHT

<b>Categoría:</b>	Tabla
<b>Unidad o formato:</b>	real
<b>Descripción:</b>	Altura máxima de la celda de una tabla. El valor predeterminado son las unidades actuales.

#### /MAXVAL

<b>Categoría:</b>	Gráfico
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Valor máximo de la escala del gráfico

#### /MEDIASUBST

<b>Categoría</b>	Selección del material de impresión
<b>Unidad o formato</b>	lista de sustitución

**Descripción** Acciones específicas del dispositivo definidas para los requisitos de **SETMEDIA**. Utilícelo en dispositivos que no admiten la selección de material de impresión a través de **SETPAGEDEVICE**



Nota: Para obtener más información, consulte [Soporte de materiales de impresión](#).

#### /MERGEVALUE

**Categoría:** Gráfico

**Valor predeterminado:** falso

**Unidad o formato:** verdadero/falso

**Descripción:** Combinar valores con etiquetas iguales.

#### /MINDIM

**Categoría:** Código de barras

**Valor predeterminado:** 1

**Unidad o formato:** entero

**Descripción:** Valor mínimo de las dimensiones de un código de barras

#### /MINVAL

**Categoría:** Gráfico

**Unidad o formato:** entero

**Descripción:** Valor mínimo de la escala del gráfico

#### /MIXPLEXCOUNT

**Categoría:** A 2 caras

**Valor predeterminado:** 0

**Unidad o formato:** entero

**Descripción:** Número de páginas para retrasar el modo a 1 cara, consulte [DUPLEX\\_off](#)

### /MONTHSLONG

<b>Categoría:</b>	Fecha
<b>Unidad o formato:</b>	matriz de 12 cadenas
<b>Descripción:</b>	Lista de nombres de los meses (largo).

### /MONTHSSHORT

<b>Categoría:</b>	Fecha
<b>Unidad o formato:</b>	matriz de 12 cadenas
<b>Descripción:</b>	Lista de nombres de los meses (corto).

### /MUPDUPLEX

<b>Categoría:</b>	Varias en una
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Valores posibles:</b>	<p><b>0</b> las posiciones de la página lógica (valor predeterminado) son idénticas en la portada y la contraportada.</p> <p><b>1</b> las posiciones de página lógica en la contraportada se calculan para que se contrapongan físicamente a las de la portada. Las páginas lógicas pueden ajustarse totalmente a la página física, ya sea a través de la especificación explícita del tamaño de la página lógica o a través de su reducción.</p>

### /NSIGN

<b>Categoría:</b>	Formato
<b>Valor predeterminado:</b>	-
<b>Unidad o formato:</b>	Código ASCII
<b>Descripción:</b>	Inicio de sesión negativo en datos numéricos.

**/OFFSETVALUE**

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	real o [real1 real2]
<b>Descripción:</b>	<p>Solo DRAWBAR. Configura un desplazamiento positivo o negativo vertical u horizontal (en función de ChartDir) según el tamaño de la fuente (1 = tamaño de la fuente). Cuando se codifica un gráfico de barras apiladas se puede programar una matriz de dos reales. El segundo real permite imprimir valores individuales de cada barra apilada (con distintos desplazamientos) además del valor total.</p> <p>Además, pueden utilizarse dos parámetros de formato específicos para configurar los formatos del usuario en valores totales:</p> <p>/FORMATV da formato a los valores totales</p> <p>/FORMATVI da formato a los valores individuales</p> <p>Las cadenas de formato pueden incluir modificadores de fuente y color para aplicar determinados atributos de color y fuente a estos valores.</p>
<b>Valores recomendados:</b>	<p><b>Vertical:</b> 1.4</p> <p><b>Horizontal:</b> 0.8</p>

**Ejemplos**

```

/I /NHEB 11 INDEXFONT
/V /NHEB 14 INDEXFONT
/L 60 INDEXLSP
/W WHITE INDEXCOLOR
[ labx valx ....]
width height
[ /OffsetValue          [-2 1.3]
  /FORMATV              (//V//LTotal\n$@000#.00)
  /FORMATVI             (//I//w$@000#)
]DRAWBAR

```

**/OMRCONFIG**

<b>Categoría:</b>	OMR
<b>Unidad o formato:</b>	[ anchura altura espaciado (config.) ]
<b>Descripción:</b>	Define la configuración del código de OMR para OMRINIT/OMRSHOW. Para obtener más información, consulte <a href="#">OMRINIT</a> .

### /OMRDIR

<b>Categoría:</b>	OMR
<b>Valor predeterminado:</b>	/V
<b>Unidad o formato:</b>	tecla
<b>Descripción:</b>	La dirección de los caracteres de cadena OMRMap en la cuadrícula OMR.
<b>Valores posibles:</b>	<p><b>/H</b> Los caracteres de la cadena de asignación de OMR se muestran en sentido horizontal de la cuadrícula OMR, y forman cada una de sus filas</p> <p><b>/V</b> Los caracteres de la cadena de asignación de OMR se muestran en sentido vertical de la cuadrícula OMR, y forman cada una de sus filas. (valor predeterminado)</p>

### /OMRHDISP

<b>Categoría:</b>	OMR
<b>Valor predeterminado:</b>	6
<b>Unidad o formato:</b>	CPP
<b>Descripción:</b>	El número de columnas por pulgada (CPP) en la cuadrícula OMR.

### /OMRHSKIP

<b>Categoría:</b>	OMR
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Indica el número de columnas omitidas.

### /OMRMAP

<b>Categoría:</b>	OMR
<b>Valor predeterminado:</b>	(ABCDEFGHIJKLMNOPQRSTUVWXYZ)
<b>Unidad o formato:</b>	cadena de asignación

**Descripción:** Indica qué caracteres componen la cuadrícula OMR y su orden.



Nota: Todos los caracteres en la cadena de respuesta pueden ser un carácter en la cadena de asignación.

### /OMRMODE

**Categoría:** OMR

**Valor predeterminado:** 12

**Unidad o formato:** puntos

**Valores posibles:** 0 Asignación de caracteres (predeterminado)

1 Asignación decimal binaria

2 Asignación de binario litho

3 P-slugging (P, 7, 4, 2, 1)

**Donde:**

**Option 0** ajusta cada carácter de la cadena de respuesta que puede estar formado por caracteres de OMRMap.

**Option 1** ajusta cada dígito de una cadena numérica en los bits binarios correspondientes 1, 2, 4 y 8. OMRMap puede configurarse como (1248) o (8421).

**Option 2** ajusta una cadena numérica (hasta 1 073 741 823) en los bits binarios correspondientes 1, 2, 4, 8, 16, 32, etc. hasta 536 870 912. OMRMap no es relevante con esta opción.

**Option 3** ajusta cada dígito de una cadena numérica en el esquema de ajuste P, 7, 4, 2, 1.

### Ejemplos

```
(4523456) [ /OMRMode 1 /OMRMap (8421) ] FILLOMR
```

```
(234523456) [ /OMRMode 2 ] FILLOMR
```

### /OMRSLUGCHAR

**Categoría:** OMR

**Valor predeterminado:** (D)

<b>Unidad o formato:</b>	cadena
<b>Descripción:</b>	La forma de los bocadillos de la cuadrícula OMR.
<b>Valores posibles:</b>	<p>(A) cuadrado</p> <p>(B) rectángulo vertical</p> <p>(C) rectángulo horizontal</p> <p>(D) círculo (predeterminado)</p> <p>(E) óvalo vertical</p> <p>(F) óvalo vertical condensado</p> <p>(G) óvalo horizontal</p> <p>(H) óvalo horizontal condensado</p>



Nota: Las letras A-H en este ejemplo no están relacionadas con las cadenas de respuesta ni asignación. Para elegir una forma que se define en la fuente /XOMR, utilice las letras A–H.

#### /OMRSLUGFONT

<b>Categoría:</b>	OMR
<b>Valor predeterminado:</b>	/XOMR
<b>Unidad o formato:</b>	/Fontname
<b>Descripción:</b>	La fuente OMR que se usa para rellenar los bocadillos de la cuadrícula OMR.

#### /OMRSLUGSIZE

<b>Categoría:</b>	OMR
<b>Valor predeterminado:</b>	12
<b>Unidad o formato:</b>	puntos
<b>Descripción:</b>	El tamaño en puntos de la fuente OMR que se usa para llenar los bocadillos de la cuadrícula de OMR.

#### /OMRVDISP

<b>Categoría:</b>	OMR
<b>Valor predeterminado:</b>	6



<b>Unidad o formato:</b>	LPP
<b>Descripción:</b>	El número de líneas o filas por pulgada (LPP) en la cuadrícula OMR.

**/OMRVSKIP**

<b>Categoría:</b>	OMR
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Indica el número de filas omitidas.

**/OMRWITERESP**

<b>Categoría:</b>	OMR
<b>Valor predeterminado:</b>	falso
<b>Unidad o formato:</b>	verdadero/falso
<b>Descripción:</b>	Indica cómo se rellenan los cuadros de respuesta
<b>Valores posibles:</b>	<b>true</b> los cuadros de respuesta se deben rellenar con la cadena de respuesta <b>false</b> ningún cuadro respuesta (predeterminado)

**/ORILINE**

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	10
<b>Unidad o formato:</b>	entero o cadena para la compatibilidad con versiones anteriores

<b>Descripción:</b>	Establece el origen y la posición de la línea de DRAWCRV y DRAWPAR.
<b>Valores posibles:</b>	<p><b>00</b> la línea va de la izquierda del primer elemento a la izquierda de cada elemento</p> <p><b>10</b> la línea va del centro del primer elemento al centro de cada elemento</p> <p><b>20</b> la línea va de la derecha del primer elemento a la derecha de cada elemento</p> <p><b>01</b> la línea va del origen del gráfico (0,0) a la izquierda de cada elemento</p> <p><b>11</b> la línea va del origen del gráfico (0,0) al centro de cada elemento</p> <p><b>21</b> la línea va del origen del gráfico (0,0) a la derecha de cada elemento</p> <p><b>false</b> igual que 10 (para la compatibilidad con versiones anteriores)</p> <p><b>true</b> igual que 11 (para la compatibilidad con versiones anteriores)</p>

#### /PAGECLIP

<b>Categoría:</b>	Diseño
<b>Valor predeterminado:</b>	falso
<b>Unidad o formato:</b>	verdadero/falso
<b>Descripción:</b>	activar recorte de la página

#### /PAGEHEIGHT

<b>Categoría:</b>	Diseño
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	La altura de la página lógica en una aplicación de varias en 1.

#### /PAGERANGE

<b>Categoría:</b>	Reimpresión de trabajo
<b>Valor predeterminado:</b>	0 (1 en dispositivos PDF)
<b>Unidad o formato:</b>	entero

<b>Descripción:</b>	/Pagerange se utiliza para definir el comportamiento de PAGERANGE después de imprimir la última página del intervalo. Este comando se puede colocar en el archivo xgf.def, en el JDT, al principio del archivo de datos con el comando PAGERANGE.
<b>Valores posibles:</b>	<p><b>0</b> cancelar el trabajo con un mensaje de error (advertencia)</p> <p><b>1</b> vaciar el archivo de datos y finalizar sin errores</p> <p><b>2</b> procesar el archivo de datos al final, pero no imprimir páginas después de la última página del intervalo</p>

#### /PAGESPERBOOKLET

<b>Categoría:</b>	Folleto
<b>Valor predeterminado:</b>	1
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	El número de páginas que debe tener un folleto, el intervalo de valor comienza en 1, el 0 no es válido.

#### /PAGEWIDTH

<b>Categoría:</b>	Diseño
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	La anchura de la página lógica en una aplicación de varias en 1.

#### /PDFCROPPING

<b>Categoría</b>	Varios
<b>Valor predeterminado</b>	/CropBox

<b>Unidad o formato</b>	nombre
<b>Descripción</b>	<p>Este parámetro se usa para seleccionar el cuadro de recorte de un PDF cuando este sea procesado mediante SCALL o RUNPDF.</p> <p>Los siguientes valores están disponibles para su selección:</p> <p>/TrimBox</p> <p>/CropBox (predeterminado)</p> <p>/ArtBox</p> <p>/MediaBox</p> <p>/BleedBox</p> <p>Si el cuadro de recorte seleccionado no está disponible en el PDF, se utilizará /MediaBox, ya que siempre está presente en todos los PDF.</p> <p>Ejemplo:</p> <p>[ /PDFCropping /TrimBox ] SETPARAMS % esta instrucción selecciona /TrimBox como cuadro de recorte</p>

## /PDFTPAGE

<b>Categoría:</b>	Diseño
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Código:</b>	ninguno
<b>Valor de código:</b>	ninguno
<b>Descripción:</b>	Número de página del PDF de destino menos uno. Utilice esta opción para seleccionar un número de página si SCALL realiza una llamada a un PDF de varias páginas.

## Ejemplos

```
[ /PDFTPage 23 ] SETPARAMS      % select PDF page 24
(mydoc.pdf) CACHE SCALL          % print page 24
```

**/PDFXEMBED**

<b>Categoría:</b>	Interpretación de imagen
<b>Valor predeterminado:</b>	3 en VI eCompose, VI Design Pro y VI Design Express 2 en FFPS Normalizer Irrelevante en cualquier otro intérprete
<b>Unidad o formato:</b>	Entero
<b>Código:</b>	Ninguno
<b>Descripción:</b>	En VI eCompose y Normalizer, active los recursos PDF/TIFF/JPEG invocados por SCALL o RUNPDF para ser incorporados en el archivo PDF resultante.
<b>Valores admitidos:</b>	0 No incorporar recursos PDF, TIFF y JPEG 1 Incorporar solo los recursos PDF 2 Incorporar solo los recursos TIFF/JPEG 3 Incorporar los recursos PDF/TIFF/JPEG



Nota: La configuración de /PDFXembed esta diseñada solo para el flujo de trabajo FFPS/APCSDK/APPE a fin de sacar el máximo partido al uso de recursos de imágenes externas al generar archivos PDF a través de la reducción de la latencia y el tamaño del archivo, ya que el PDF puede enviarse a APPE para su impresión.

Al usar esta opción en FFPS, asegúrese de que la versión de APPE en la que se ejecuta VI Compose admite las referencias a imágenes externas. Si se selecciona una opción no incorporada, asegúrese de que se dispone de los recursos en la plataforma cuando el PDF se visualiza o se imprime.

La modificación de las opciones prefijadas en cualquier otra plataforma actualmente no es compatible.

**Ejemplos**

```
[ /PDFXembed 1 ] SETPARAMS      % embed PDF resources only
(mydoc.pdf) CACHE SCALL         % embed mydoc.pdf in the target PDF
```

**/PDFXObject**

<b>Categoría:</b>	Diseño
<b>Valor predeterminado:</b>	1 en VI eCompose y FFPS Normalizer 0 en otros intérpretes

<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	En VIECompose y Normalizer, active los recursos PDF invocados por SCALL o RUNPDF para ser procesados como XObjects en el archivo PDF resultante.
<b>Valores admitidos:</b>	<p><b>0</b> no procesar recursos PDF como XObjects. Use imágenes EPS incorporadas, si las hubiera; de lo contrario, muestre un cuadro de patrones de texto.</p> <p><b>1</b> procesar recursos PDF como XObjects</p>



Nota: Si un recurso PDF se procesa como un XObject, el PDF resultante se coloca en una carpeta de confianza para su correcta visualización. Para obtener más información, consulte Adobe Acrobat, **Editar > Preferencias > Seguridad (Mejorada)**.

## Ejemplos

```
[ /PDFXObject 1 ] SETPARAMS %enable PDF Xobjects
(mydoc.pdf) CACHE SCALL %include mydoc.pdf as an Xobject in the
target PDF
```

## /PLOTSYMBOL

<b>Categoría:</b>	Gráfico
<b>Unidad o formato:</b>	[(string) size Colorkey]
<b>Descripción:</b>	Define un símbolo de trazado que se imprime en puntos de nodo en un gráfico de líneas dibujado con DRAWCRV.
<b>Valores posibles:</b>	<p><b>(string)</b> un solo carácter de la fuente Zapf Dingbats. Puede ser un carácter entre paréntesis ( ) o un valor hexadecimal entre &lt; y &gt;. (Ejemplo: (l) = punto, (s) = triángulo hacia arriba, (t) = triángulo hacia abajo, (u) = rombo, (n) = cuadrado, &lt; AA &gt; = corazón, etc.)</p> <p><b>size</b> real</p> <p><b>ColorKey</b> cualquier Colorkey o nulo (usar color de ColorTable)</p>

## /PRINTLABEL

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	1
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Imprimir etiquetas en barras, gráficos de curvas y gráficos circulares:

<b>Valores posibles:</b>	<b>0</b> no imprimir la etiqueta <b>1</b> imprimir etiquetas debajo o junto al eje X <b>2</b> Imprimir etiquetas al final de las barras. Esta opción desactiva automáticamente PrintValue.
--------------------------	--



Nota: Los parámetros de atributos de texto (fuente, color,...) se permiten dentro del texto.

## /PRINTSCALE

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	1
<b>Unidad o formato:</b>	entero
<b>Código:</b>	+128
<b>Valor de código:</b>	0
<b>Descripción:</b>	Imprimir la escala en gráficos de curvas y barras:
<b>Valores posibles:</b>	<b>0/false</b> no imprimir la escala <b>1/true</b> imprimir la escala a la izquierda o en la parte inferior <b>2</b> imprimir la escala a la derecha o en la parte superior



Nota: Para ver las opciones de formato, consulte [/Format](#).

## /PRINTVALUE

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	falso
<b>Unidad o formato:</b>	verdadero/falso
<b>Código:</b>	+64/
<b>Valor de código:</b>	verdadero
<b>Descripción:</b>	Imprimir el valor (barra y curva) o el porcentaje (circular). Para ver las opciones de formato, consulte <a href="#">/Format</a> .

**/RESCASENSE**

<b>Categoría:</b>	Varios
<b>Valor predeterminado:</b>	verdadero
<b>Unidad o formato:</b>	verdadero/falso
<b>Descripción:</b>	<p>verdadero: active la diferenciación de mayúsculas y minúsculas del nombre del recurso (valor predeterminado).</p> <p>falso: desactive la diferenciación de mayúsculas y minúsculas del nombre del recurso.</p>



Nota: Este parámetro solo se usa en sistemas UNIX y no tiene ningún efecto en sistemas Windows. Si se configura como falso, las llamadas incorrectas al recurso xyz.ext, o con cualquier combinación de mayúsculas o minúsculas, generará los siguientes intentos de acceso adicionales:

- xyz.ext
- Xyz.ext
- XYZ.ext
- xyz.EXT
- Xyz.EXT
- XYZ.EXT

**Ejemplos**

```
[ /ResCaseSense false ] SETPARAMS      % disable resource names
                                         % case sensitivity
```

**/RESOLVEPATH**

<b>Categoría:</b> PIF
<b>Valor predeterminado:</b> 0
<b>Unidad o formato:</b> entero
<b>Descripción:</b> Establece el modo de resolución de la ruta de fileref en enlaces PIF.
<p><b>Valores posibles</b></p> <ul style="list-style-type: none"> <li>• <b>0</b> (predefinido) No resuelve la referencia al archivo. Déjelo de la manera proporcionada.</li> <li>• <b>1</b> Intenta resolver la referencia al archivo en el contexto actual de VIPP® (SETIPATH o SETMPATH). Si existe, sustitúyalo por la ruta completa. De lo contrario, déjelo de la manera proporcionada.</li> </ul> <p>:</p>



**/RIGHTBLEED**

<b>Categoría:</b>	Diseño
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	El valor de la cantidad de sangrado permitido en el margen derecho de la página lógica en una aplicación de varios en 1.

**/ROTATE**

<b>Categoría:</b>	Diseño
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	El grado de giro de las páginas lógicas.
<b>Valores posibles:</b>	0 (predefinido), 90, 180, 270 sentido horario.

**/ROWHEIGHT**

<b>Categoría:</b>	Código de barras
<b>Valor predeterminado:</b>	3 o 4
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	La altura de una fila en el código de barras. El valor prefijado es 4 cuando el nivel del error es bajo, dada la cantidad de datos codificados. El valor prefijado es 3 cuando el nivel de error es el adecuado para la cantidad de datos codificados.

**/SCALESTEP**

<b>Categoría:</b>	Gráfico
<b>Unidad o formato:</b>	entero o cadena de enteros

<b>Descripción:</b>	Establecer el incremento de la escala.
<b>Valores posibles:</b>	<p><b>El valor es un número entero:</b></p> <p><b>0</b> cálculo automático de la etapa de ajuste a escala.</p> <p><b>1-10</b> forzar un incremento de escala entre 1 y 10 (o el múltiplo más apropiado de 10).</p> <p><b>El valor es una cadena entera (con el valor entero entre paréntesis):</b></p> <p>El valor máximo se configura con el valor más alto de la recopilación de valores, y las etapas de la escala se calculan a través de la división de este valor por el entero proporcionado.</p>

#### /SHADEADJUST

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	.5
<b>Unidad o formato:</b>	real entre -1 y + 1
<b>Descripción:</b>	Permite el control del sombreado si 3D es verdadero.
<b>Valores posibles:</b>	<p><b>[-1 &lt;0]</b> sombreado más claro. Los valores más pequeños son más claros.</p> <p><b>0</b> sin sombreado</p> <p><b>[&gt;0 1]</b> sombreado más oscuro. Los valores más grandes son más oscuros.</p>

#### /SHPWRAP

<b>Categoría:</b>	composición de página
<b>Unidad o formato:</b>	<p>[ (char1) ... (charN) ]</p> <p>/ALL</p>
<b>Descripción:</b>	<p>Especifica un conjunto de caracteres de ajuste del comando SHP además del espacio. charx siempre es un único carácter.</p> <p>/ALL puede utilizarse para escribir verticalmente en una columna muy estrecha.</p>



Nota: /SHPWrap fuerza el ajuste solamente si se supera la anchura de la columna. Si la cadena de texto se ajusta a la anchura de la columna especificada, no se realiza ninguna acción.

## Ejemplos

```
[ /SHPWrap [ (.) (@) ] ] SETPARAMS
```

```
[ /SHPWrap /ALL ] SETPARAMS
```

**/SLICEBURST**

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	% de radio
<b>Código:</b>	+8
<b>Valor de código:</b>	0,1
<b>Descripción:</b>	Valor de ráfaga (0 = sin ráfagas)

**/SLICESEPCOLOR**

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	NEGRO
<b>Unidad o formato:</b>	Colorkey
<b>Descripción:</b>	Color de los bordes de la división o la barra

**/SLICESEPWIDTH**

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	0,5
<b>Unidad o formato:</b>	puntos
<b>Código:</b>	+2
<b>Valor de código:</b>	0
<b>Descripción:</b>	Anchura de línea de los bordes de la división o la barra

## /SPOTLABELS

<b>Categoría:</b>	Gráfico
<b>Unidad o formato:</b>	[ (label1) (label2) ... (labelN) ]
<b>Descripción:</b>	Proporcione las etiquetas que se asociarán con los colores directos si se suministran diversos valores de los elementos a DRAWBAR. (ejemplos de sintaxis 3 y 4 de la descripción de DRAWBAR). Los parámetros de atributos de texto (fuente, color,...) se permiten dentro del texto.

## /SPOTOFFSET

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	Horizontal: 1,4
<b>Unidad o formato:</b>	% de radio % de anchura/2
<b>Valor predeterminado:</b>	Vertical: 0
<b>Unidad o formato:</b>	% de altura/2
<b>Descripción:</b>	Hor_Offset o [Hor_offset Vert_offset ] Desplazamiento de puntos desde el centro del gráfico (positivo = desplazamiento a la derecha/arriba, negativo = desplazamiento a la izquierda/abajo) Cuando se omite, Vert_offset tiene el valor predeterminado 5.

## /SPOTSIZE

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	% de tamaño de la fuente
<b>Código:</b>	+4
<b>Valor de código:</b>	1
<b>Descripción:</b>	Tamaño de los puntos del margen (0 = sin puntos, las etiquetas se colocan alrededor de estos o bajo los mismos)

**/STACK**

<b>Categoría:</b>	Gráfico
<b>Unidad o formato:</b>	verdadero/falso
<b>Descripción:</b>	Verdadero indica que el gráfico se apilará encima de otro gráfico, y requiere un fondo transparente.

**/TABLESTROKE**

<b>Categoría:</b>	Tabla
<b>Valor predeterminado:</b>	sin trazo
<b>Unidad o formato:</b>	GEPKey
<b>Descripción:</b>	Diseñado para trazar los bordes externos de una tabla.

**Ejemplos**

```

/TableStroke GEPKey
%GEPKey = table border (stroke not fill) color
/TableStroke [GEPKey GEPKey GEPKey GEPKey ]
%Top Bottom Left Right

```

**/TEXTATT**

<b>Categoría:</b>	Tabla
<b>Unidad o formato:</b>	{ VIPP® code }
<b>Descripción:</b>	Código VIPP® utilizado para conjuntos de atributos de texto

**/TEXTFILTER**

<b>Categoría:</b>	XML
<b>Valor predeterminado:</b>	verdadero
<b>Unidad o formato:</b>	verdadero/falso

**Valores posibles:**

**true** activa el filtrado de espacios adicionales y caracteres de control de contenidos del nodo XML.

**false** desactiva el filtrado de espacios adicionales y caracteres de control de contenidos del nodo XML.



Nota: Para obtener más información, consulte Eliminación de espacios en blanco y caracteres de control en la *Guía del usuario de FreeFlow VI Compose*.

## /TEXTFIT

<b>Categoría:</b>	Composición de página
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Seleccione el método de ajuste de texto de SHMF y SHP.
<b>Valores posibles:</b>	<b>0</b> escala anamórfica (solo horizontal) <b>1</b> escala isomórfica (horizontal y vertical)

## /TIMEZONE

<b>Categoría:</b>	Fecha
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Zona horaria, +- minutos de UTC. (por ejemplo, -480 = PST)

## /TIMEZONENAME

<b>Categoría:</b>	Fecha
<b>Unidad o formato:</b>	matriz de 2 cadenas
<b>Descripción:</b>	Nombres de las zonas horarias: Hora estándar, horario de verano.

## Ejemplos

[ (PST) (PDT) ]

**/TOPBLEED**

<b>Categoría:</b>	Diseño
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	La cantidad de sangrado permitida en el margen superior de una página lógica en una aplicación de varias en 1.

**/TRANSWEIGHT**

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	1
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	Mejora los píxeles negros en los bordes suaves de las áreas transparentes.
<b>Valores posibles:</b>	<p>1 eliminar los píxeles negros (predeterminado)</p> <p>&lt; 1 reducir los píxeles negros</p> <p>0 mantener los píxeles negros (comportamiento actual)</p>

**/VALUECOLOR**

<b>Categoría:</b>	Gráfico
<b>Unidad o formato:</b>	Colorkey
<b>Descripción:</b>	DRAWBAR/DRAWCRV solamente. Configure una Colorkey distinta para los valores (el valor predeterminado es el color actual).

**/VGUTTER**

<b>Categoría:</b>	Diseño
<b>Valor predeterminado:</b>	0
<b>Unidad o formato:</b>	entero
<b>Descripción:</b>	El valor de margen interior vertical entre las páginas lógicas en una aplicación de varias en 1.

## /WIDTH

<b>Categoría:</b>	Tabla
<b>Unidad o formato:</b>	real
<b>Descripción:</b>	Ancho de la celda de una tabla. El valor predeterminado son las unidades actuales.

## /XFLOAT

<b>Categoría:</b>	Gráfico
<b>Valor predeterminado:</b>	falso
<b>Unidad o formato:</b>	booleano
<b>Descripción:</b>	Activa/desactiva la flotación del eje X en DRAWBAR y DRAWCRV.
<b>Valores posibles:</b>	<b>false</b> el eje X siempre está a cero <b>true</b> el eje X es flotante. Está cerca del valor mínimo o máximo del conjunto de valores.

## /XMLMISVAL

<b>Categoría:</b>	Varios
<b>Valor predeterminado:</b>	cadena vacía
<b>Unidad o formato:</b>	cadena
<b>Código:</b>	ninguno
<b>Descripción:</b>	Establece la cadena producida si falta un nodo de XML.

## Ejemplos

```
[ /XMLMisVal (** Missing **) ] SETPARAMS % set missing node value
```

## /ZSREPEATFIELD

<b>Categoría:</b>	Varios
<b>Unidad o formato:</b>	variable o nombre del campo





Nota: No se puede configurar ZSRepeatField en el DBM. Puede definir ZSRepeatField antes del comando **STARTDBM** en el archivo de envío o en un **JD**T que se llama antes que el comando **STARTDBM**.

**Descripción:**

Declara la repetición del campo de ZSORT



# Combinaciones compuestas

Este capítulo incluye:

Expresiones aritméticas .....	676
Operadores de prueba y expresiones condicionales.....	679

Las combinaciones compuestas utilizan expresiones y operadores de prueba con el lenguaje VIPP® para proporcionar más funcionalidades en aplicaciones VIPP®. El lenguaje VIPP® usa los operadores de prueba disponibles en el lenguaje PostScript y cuenta con varios operadores de prueba de VIPP® que se describen más abajo.

Para obtener más información sobre los operadores de prueba de PostScript, consulte la referencia del lenguaje PostScript producida por Adobe Systems Incorporated. La edición más reciente de este manual se puede descargar de este sitio: [https://partners.adobe.com/public/developer/ps/index\\_specs.html](https://partners.adobe.com/public/developer/ps/index_specs.html)

## Expresiones aritméticas

Las expresiones aritméticas permiten combinar variables numéricas, constantes numéricas y operadores aritméticos en un solo operando que se pasa a un determinado comando de VIPP®.

Una expresión aritmética VIPP® se puede definir como:

```
expression      ::= member | unop member |
                  expression (binop expression)* |
                  «expression» | unop «expression»
member          ::= numeric-variable | numeric-constant
binop           ::= '+' | '-' | '*' | ':' | 'M' | 'm' | 'q' | 'r'
unop            ::= - | + | #
numeric-variable ::= any valid VIPP variable with numeric contents
numeric-constant ::= any valid PostScript integer or real
```

Estos son los operadores aritméticos binarios disponibles:

+	suma
-	resta
*	multiplicación
:	división
M	máximo
m	mínimos
q	cociente
Q	límite de cociente redondeado al siguiente número entero
r	resto



Nota: Para evitar conflictos con los nombres de las variables, coloque entre comillas sencillas los operadores binarios, por ejemplo: '+', '-', etc.

Los siguientes operadores unarios están disponibles. Los operadores unarios no requieren comillas.

+	valor positivo
-	valor negativo
#	valor absoluto

Las limitaciones siguientes se aplican a las expresiones aritméticas:

- No puede superar los 127 caracteres de longitud
- Cabe en una línea
- No puede contener delimitadores PostScript como: , espacio, /, [, ], {, }, <, >, (, ) o %.

De forma general, las expresiones se evalúan de izquierda a derecha. Sin embargo, la multiplicación y la división se evalúan antes que la mayoría del resto de expresiones. Las expresiones secundarias, que se encapsulan entre « y », se evalúan en primer lugar. Para producir los símbolos en el teclado de un PC, asegúrese de que el botón Bloq Num esté activado, use los números en el teclado alfanumérico y, finalmente, use ALT+174 y ALT+175.

El orden de la evaluación es el siguiente:

- + (unario)
- - (unario)
- # (unario)
- 'r'
- 'q'
- 'm'
- 'M'
- ':'
- '\*\*'
- '-'
- '+'

Las expresiones aritméticas siempre producen un número entero o real (el que sea aplicable), independientemente de los tipos de miembros contenidos en la expresión: un número entero, real o una cadena numérica.

Una aplicación útil de los operadores unarios consiste en convertir una cadena numérica en un número real o entero para poder realizar comparaciones numéricas precisas en instrucciones condicionales como IF/ELSE/ENDIF.

### **No los utilice cuando necesita que los números se impriman con precisión**

No utilice expresiones aritméticas para calcular números que deben imprimirse con precisión. Las expresiones aritméticas están reservadas para la realización y colocación de cálculos, contadores de repeticiones, etc.

### **Para calcular con precisión cadenas de caracteres numéricos para la realización de tareas de impresión**

Utilice los comandos de VIPP® ADD, SUB, MUL y DIV, que proporcionan hasta 25 dígitos enteros y 15 dígitos digitales.

### **Ejemplos**

En estos ejemplos se asumen estas definiciones:

```
/VAR1 100 SETVAR
```

```
/VAR2 23 SETVAR
```

Expresiones y resultados:

VAR1 '+' VAR2	123
VAR1 '-' VAR2	77
VAR1 '*' 2	200
VAR1 '+' VAR2 '*' .35	108.05
«VAR1 '+' VAR2» '*' .35	43.05
-VAR2 '*' 10	-230
VAR1 'q' VAR2	4
VAR1 'Q' VAR2	5
/VAR1 «VAR1 ':' 2 '+' .5» 'q' 1 SETVAR	% divide by 2 and round

Comandos que usan expresiones como operandos:

«VAR1 '+' VAR2» '\*' .35 VAR3 MOVETO

VAR\_LM VAR\_TM '+' 270 VAR\_RM '-' VAR\_LM 0 S1 DRAWB

COLW '-' 420 0 440 90 XLTR\_S1 DRAWB

Usando operadores unarios y asumiendo que CopyCount es un campo de base de datos:

IF +CopyCount 50 gt ...

## Operadores de prueba y expresiones condicionales

Puede generar expresiones condicionales mediante variables, constantes y operadores de prueba. El resultado siempre es un valor booleano verdadero o falso. Se pueden utilizar expresiones condicionales como una condición en una instrucción `IF/ELSE/ENDIF` o almacenarse como referencia para su uso posterior mediante `SETVAR`.

### EJEMPLOS

```
IF CITY (Paris) eq { ... } ENDIF
IF CITY (Paris) eq COUNTRY (France) eq and { ... } ENDIF
/VAR_PAR_FR CITY (Paris) eq COUNTRY (France) eq and SETVAR
IF VAR_PAR_FR { ... } ENDIF
IF VAR_PAR_FR not { ... } ENDIF
IF VAR1 '+' VAR2 200 gt { ... } ENDIF
```

Puede usar los operadores de prueba de las instrucciones `SETRCD` y `SETPCD` si precede los operadores con una barra inclinada (/).

Entre los operadores de prueba de PostScript se incluyen:

<b>eq</b>	igual
<b>ne</b>	distinto
<b>gt</b>	mayor que
<b>ge</b>	mayor o igual que
<b>lt</b>	menos que
<b>le</b>	menos o igual que
<b>o</b>	combine dos resultados de prueba o booleanos con o
<b>y</b>	combine dos resultados de prueba o booleanos con y
<b>no</b>	niega el resultado de una prueba o una instrucción booleana

El lenguaje VIPP® se ha ampliado para incluir estos operadores de prueba:

<b>CIEQ</b>	Igual sin diferenciar entre mayúsculas y minúsculas
<b>CINE</b>	Desigual sin diferenciar entre mayúsculas y minúsculas
<b>HOLD</b>	busca una cadena

### CIEQ Y CINE

**CIEQ** (Igual sin diferenciar entre mayúsculas y minúsculas) y **CINE** (Desigual sin diferenciar entre mayúsculas y minúsculas) son operadores de prueba que permiten comparar cadenas independientemente de si los

caracteres en las cadenas están en mayúsculas o minúsculas. Puede utilizar los operadores para comparar las cadenas de la misma manera que se usan los operadores `eq` y `ne` de PostScript.

Estos operadores se basan en la codificación de la fuente activa actual, por ejemplo, la última fuente definida por **SETFONT** o **INDEXFONT**. Dos caracteres en las cadenas que se comparan se consideran iguales si sus nombres de carácter (en la tabla de codificación) son iguales independientemente de si están en mayúsculas o minúsculas: `/a=/A`, `/eacute=/Eacute`.

### Ejemplos

```
(Hello World) (hello WORLD) CIEQ
```

devuelve verdadero.

### Comandos relacionados

**IF/ELSE/ELIF/ENDIF**, **SETPCD**, **SETRCD**

### **/HOLD**

El comando **HOLD** busca la segunda cadena **IF/ELSE** o la cadena de comparación **RCD/PCD** en cualquier lugar de la primera cadena **IF/ELSE**, o la porción seleccionada del registro de los datos **RCD/PCD**.

### Ejemplos

Este ejemplo es verdadero cuando la palabra **DIVISION** aparece en cualquiera de las primeras 100 posiciones del registro. Una búsqueda de una cadena en los registros y en cada byte del registro puede afectar el rendimiento.

```
/IF_CND1 0 100 /HOLD (DIVISION) SETRCD
```

```
IF ADDRESS2 (street) HOLD
```

### Comandos relacionados

**IF/ELSE/ELIF/ENDIF**, **SETPCD**, **SETRCD**.



# Mensajes de error

Este capítulo incluye:

Gestión de errores .....	682
Lista de errores de VIPP® .....	684
Otros errores.....	689

Los mensajes de error de VI Compose son errores que se pueden producir al usar VI Compose además de los errores de PostScript estándar. Consulte la documentación correspondiente para obtener una descripción de los errores de PostScript.

## Gestión de errores

Los errores de PostScript y VI Compose se imprimen en hojas de error. Esta sección contiene información sobre el formato de los mensajes de error en las hojas de error.

### ERRORES DE POSTSCRIPT

En general, los errores de PostScript se documentan en una hoja de errores producida por la impresora cuando se produce el error. Cuando se imprime un error de PostScript en una hoja de errores, se incluye esta información:

- Comando erróneo: Indica el operador de PostScript que detectó el error
- Tipo de error: Indica el tipo de error

Además, puede que se muestren las pilas del intérprete.

Si no puede determinar la causa de un determinado error a través la información de la hoja de errores, intente aislar el origen del error mediante el uso de omisiones de partes del trabajo. Para omitir una línea en un trabajo, inserte `%` al principio de la línea. Si el origen del error no se puede determinar a través de este método, póngase en contacto con el vendedor de Xerox para obtener asistencia.

### HOJA DE ERRORES

Ocasionalmente, es posible que un trabajo no pueda imprimirse incluso si está claro que la impresora recibió el trabajo. En ese caso, es probable que se haya producido un error de PostScript pero que la opción de la impresión de la hoja de errores no se haya activado. Consulte la documentación de la impresora para obtener más información sobre cómo activar la opción de la hoja de errores. Si se activa la opción de impresión de las hojas de errores, imprima el trabajo de nuevo.

### EJECUCIÓN DE ARCHIVOS DE ENLACE

Si se produce un error durante la ejecución automática del archivo de enlace VIPP®, es difícil determinar a través de la hoja de errores qué programa generó el error. En este caso, desactive la ejecución automática del archivo de enlace VIPP® y a continuación omita la secuencia `/usr/xgf/src/xgf`. Ejecute el archivo de arranque y envíe el archivo de enlace como un archivo VIPP® normal. A continuación, omita las líneas del archivo de enlace, empezando por la parte inferior, para aislar el archivo que causa el error.

### ENCONTRAR LA CAUSA DE UN ERROR

Si utiliza una aplicación grande o tiene problemas para encontrar un determinado error, coloque una cadena no válida en la aplicación para encontrarlo. Por ejemplo, introduzca la cadena en el medio del archivo JDT o DBM para determinar si el error que desea encontrar se encuentra por encima o por debajo de la cadena especificada. La cadena no válida introducida no puede ser un comando VIPP®.

Al volver a enviar la aplicación después de especificar una cadena no válida, y si la condición del error pasa a ser un comando erróneo que hace referencia a la cadena no válida especificada, el error se encuentra después de la cadena no válida especificada. Determine el archivo que genera el error y en qué línea se encuentra el mismo mediante el desplazamiento de la cadena no válida.

Con VI Design Pro, las líneas y las secciones del código se pueden incluir en comentarios para omitirlas hasta que se consigan los resultados deseados. Esto permite aislar rápidamente áreas problemáticas del código que requieren ser modificadas. Para más información, consulte la *Guía del usuario de FreeFlow VI Design Pro*.

Para obtener más información sobre el uso correcto de la sintaxis de VIPP®, consulte [Comandos de VIPP®](#).

## Lista de errores de VIPP®

Los errores de VIPP® son un subconjunto de los errores de PostScript. Estos errores figuran en el campo Error que empieza con el texto VIPP\_ y se documentan en las secciones siguientes.

### VIPP\_ACCESS\_DENIED

Se encontró un recurso, como, por ejemplo, un formulario, un segmento, un JDT, una imagen o una lista de fuentes, pero el intérprete no tiene los derechos de acceso necesarios para acceder a él.

### VIPP\_AFM\_PARSING\_ERROR

VI Compose detectó un error al analizar un archivo de Adobe Font Metrics para obtener la información de espaciado. Compruebe la integridad y la completitud de los archivos AFM referenciados por el comando **SETENCODING**.

### VIPP\_AMBIGUOUS\_NAME IN \_NAME

El nombre indicado por este mensaje de error produce un conflicto. En el modo XML indica que VI Compose detecta un nombre de VxVpath que no es único. El VxVname está cualificado adicionalmente para eliminar la ambigüedad.

En el modo de base de datos, este nombre ya es una palabra reservada y no se puede usar como nombre de campo. Seleccione otro nombre para el campo.

Se recomienda usar un conjunto de letras en minúsculas y mayúsculas, o nombres de campo que empiecen con el prefijo '\_'. Por ejemplo, status es una palabra clave reservada y generaría la visualización del mensaje de error. Sin embargo, Status o \_status son nombres de campo válidos y no generan la visualización del mensaje.

### VIPP\_BUFFER\_OVERFLOW

El comando **ADD** o **SUB** generó el desbordamiento de la memoria intermedia de computación interna de VI Compose que tiene un límite de 40 dígitos. 25 dígitos para la parte entera y 15 dígitos para la parte decimal.

### VIPP\_CORRUPTED\_OR\_UNSUPPORTED\_IMAGE\_FILE

Indica que se ha detectado un archivo de imagen dañado o no compatible.

### VIPP\_INVALID\_ALIGN EN SHMF

El parámetro de alineación **SHMF**, **SHMf** y **SHmf** no es válido. Este parámetro no puede ser superior a tres.

### VIPP\_INVALID\_ALIGN EN SHP

El parámetro de alineación **SHP** y **SHp** no es válido. Este parámetro no puede ser mayor que cinco ni igual a tres.

**VIPP\_INVALID\_ALIGN EN SHX**

El parámetro de alineación **SHX** no es válido. Este parámetro no puede ser superior a cinco.

**VIPP\_INVALID\_BOOKLET\_LENGTH**

Un folleto no tiene la longitud necesaria especificada por `/PagePerBooklet` en **SETPARAMS**. El trabajo se ha cancelado según lo requerido por el parámetro `/BookletMismatch`.

**VIPP\_INVALID\_COLOR**

El color utilizado en combinación con un comando **MPR** no es adecuado para este comando. El color negro es el único color admitido en dispositivos monocromos.

**VIPP\_INVALID\_COMBINATION IN MULTI-UP\_COLLATE\_OFF**

Varias en una y **COLLATE\_off** no se puede combinar en el modo de base de datos

**VIPP\_INVALID\_COMBINATION IN STOREVAR\_FILE\_MUST\_EXIST**

Si **STOREVAR** se utiliza en modo de proyectos, es necesario que exista el archivo.

**VIPP\_INVALID\_COMBINATION IN STOREVAR\_VIECMULTI**

Se utiliza un comando **STOREVAR** en el modo de varias instancias VIEC.

**VIPP\_INVALID\_COMBINATION EN UV2L**

El color UV subyacente no admite una segunda capa.

**VIPP\_INVALID\_CONTENTS IN ENDPAGE**

Un comando **ENDPAGE** con la opción `/P` contiene comandos de marca en su procedimiento. Coloque los comandos de marca en un procedimiento **ENDPAGE** sin la opción `/P`.

**VIPP\_INVALID\_FONT**

La fuente que se usa en combinación con un comando **MPR** o **GLT** no es adecuada para estas funciones.

**VIPP\_INVALID\_LICENSE\_FILE**

El archivo referenciado actualmente `/usr/xf/src/xf.lic` no es un archivo de licencia o está dañado. Vuelva a cargar el archivo original.

#### VIPP\_INVALID\_PIF\_TYPE

El parámetro /PIFtype de un comando **SETPIF** o **INDEXPIF** no es válido. Asegúrese de que el parámetro es uno de los tipos indicados en las descripciones de los comandos **SETPIF** y **INDEXPIF**.

#### VIPP\_INVALID\_PN\_OPTION

El parámetro de alineación o **SETPAGENUMBER** pos o align no es válido. Asegúrese de que este parámetro sea menor que cero o mayor que siete.

#### VIPP\_INVALID\_SYNTAX EN RPE

Una entrada entre **BEGINRPE** y **ENDRPE** en una repetición RPE no es válida. Asegúrese de que la entrada sea una tabla con diez elementos o una clave de definición de criterios de registros (RCD) definida por **SETRCD**.

#### VIPP\_INVALID\_SYNTAX EN SETBAT

El número de parámetros en una instrucción **SETBAT** no es un múltiplo de 13.

#### VIPP\_INVALID\_SYNTAX IN SETMULTIUP

Una definición de **SETMULTIUP** contiene un número de elementos incorrecto. Asegúrese de que el número total de elementos de una tabla es un múltiplo de cinco.

#### VIPP\_INVALID\_SYNTAX EN SHP

Falta un parámetro de alineación en una definición de **SHP** y **SHp**.

#### VIPP\_INVALID\_VARIABLE\_NAME

Se ha detectado un nombre de variable no válido. O el nombre no empieza con VAR o ^, o su longitud supera los 127 caracteres.

#### VIPP\_INVALID\_VSUB

Durante la sustitución de un nombre de variable, **VSUB** se detectó una sintaxis incorrecta. En general, esto ocurre cuando falta un punto de cierre (.) después de los dos signos de dólar (\$\$).

#### VIPP\_LENGTH\_ERROR IN ENDPCC

Una entrada en una tabla PCC entre **BEGINPCC** y **ENDRPE** contiene un número incorrecto de elementos. Asegúrese de que la entrada tiene tres caracteres.

**VIPP\_LENGTH\_ERROR IN RPE**

Una entrada entre **BEGINRPE** y **ENDRPE** en una definición RPE contiene un número no válido de elementos. Asegúrese de que el número de elementos es diez.

**VIPP\_LICENSE\_FAILED**

La licencia de VI Compose actual no es válida para este dispositivo. Póngase en contacto con un vendedor de Xerox para actualizar la licencia.

**VIPP\_MISPLACED EN SETPAGESIZE**

Se coloca un comando **SETPAGESIZE** en la página después del primer comando de marcación. Coloque el comando **SETPAGESIZE** al principio de la página antes de todos los comandos de marcación.

**VIPP\_OMR\_INVALID\_RESPONSE\_STRING**

Uno o varios caracteres en la cadena de respuesta no se encuentra en la cadena de mapa OMR.

**VIPP\_PDF417\_DATA\_LIMIT\_EXCEEDED**

El número total de caracteres en los parámetros de la cadena de un comando **PDF417** supera el número máximo admitido.

**VIPP\_PDF417\_INVALID\_MODE**

Se genera un modo no válido durante la compactación de un parámetro de cadena de texto de un comando **PDF417**.

**VIPP\_PLANE\_NUMBER\_OUT\_OF\_RANGE**

Se ha dado a un comando **SETFORM** o **SETBFORM** un número de plano superior al máximo definido anteriormente por **SETMAXFORM** o **SETMAXBFORM**.

**VIPP\_POSTNET\_INVALID\_DIGIT**

El parámetro de la cadena de una función de transformación **POSTNET** contiene un carácter no válido. Solo se permiten caracteres numéricos, 0-9, y el guión (-) en la quinta posición.

**VIPP\_POSTNET\_INVALID\_LENGTH**

El parámetro de la cadena de una función de transformación **POSTNET** tiene una longitud no válida. Asegúrese de que el parámetro de la cadena es 5, 9 u 11.

#### **VIPP\_RPE\_INVALID\_PREFIX**

Se ha detectado un registro predeterminado con un prefijo que no se ha definido en la definición de RPE actual. Este error se produce solamente si se activa el control de prefijo RPE. **SETRPEPREFIX** establece la longitud del prefijo con un valor negativo.

#### **VIPP\_RPE\_INVALID\_PREFIX\_LENGTH**

Se detectó un registro predeterminado con un prefijo más corto que la longitud definida por **SETRPEPREFIX**.

#### **VIPP\_SETVAR\_INVALID\_NAME**

Un nombre de variable definido por **SETVAR** no empieza por VAR.

#### **VIPP\_UNABLE\_TO\_LOCATE**

Un recurso, como, por ejemplo, un formulario, un segmento, un JDT, una imagen o una lista de fuentes, no está disponible en ninguna de las bibliotecas referenciadas por el comando **SETxPATH** correspondiente.



## Otros errores

Estos errores pueden producirse como resultado de la interacción entre VI Compose y otros programas.

### ERROR DE LIMITCHECK

Este mensaje de error puede producirse al usar un formulario VIPP® creado con la utilidad mkfrm a través de un archivo EPS que se creó mediante un controlador de Windows.

```
%%[Error: limitcheck; Offending Command: flffff500defff500.....
Stack:
(eform.frm)
-mark-
[1]
0
```

Este error puede indicar que el formulario contiene imágenes de mapas de bits invocados a través de operadores de PostScript mediante la imagen del archivo actual.

Aplique una de estas soluciones para resolver este problema:

- Elimine el problema mediante el archivo PostScript original con esta sintaxis:  

```
{ (eform.ps) CACHE SETFORM
```
- Envíe el archivo EPS a los servicios de descomposición y use el resultado como un formulario. Para más información sobre los servicios de descomposición, consulte la *Guía del usuario de FreeFlow VI Compose*.

### POSTSCRIPT

Error de tiempo de ejecución en el intérprete de PostScript. Para obtener información detallada sobre este mensaje de error, consulte el manual de PostScript adecuado.

### ERROR POSTSCRIPT ERROR - DICTFULL / OFFENDING COMMAND:DEF

Este error puede producirse por la presencia de un gran número de variables, según lo definido por **SETVAR**, **GETFIELD** o los campos del archivo de base de datos, si se usa un intérprete de PostScript de nivel 1. Utilice un intérprete de PostScript de nivel 2 para resolver este problema.

### PÁGINAS SELECCIONADAS: <FIRST> <LAST>

Este error puede ser uno de los siguientes:

- **PAGERANGE** se utiliza para este trabajo. El mensaje refleja la primera y la última página impresa.
- La licencia xgf.lic que se está usando es una licencia de demostración o evaluación, que hace que el trabajo se anule después de 200 páginas en impresoras de producción, y después de 10 páginas en el resto de impresoras. En ese caso, sustituya la licencia de demostración o evaluación por una licencia de producción.

#### FOLLETOS SELECCIONADOS: <FIRST> <LAST>

Este error indica que **BOOKLETRANGE** se utiliza para este trabajo. El mensaje indica el primer y el último folleto impreso.

#### ERROR DE DESBORDAMIENTO DE LA PILA EN GHOSTSCRIPT

El error de desbordamiento de la pila se produce si se excede el número máximo de tokens, es decir, palabras de PostScript, que pueden figurar en la pila de operandos.

Sin embargo, el error no se produce en un DocuPrint NPS, porque la pila de operandos puede contener hasta 5000 tokens de manera predefinida, y se puede configurar para admitir un número superior cuando sea necesario.

Si se produce este error, Xerox recomienda que realice la actualización a una versión más reciente del software de Ghostscript. El límite de esta versión de la aplicación se ha ampliado a más de 10000.

## Sugerencias de programación

Este capítulo incluye:

Consumir frente a ejecutar.....	692
Origen 0,0 para la posición de la página lógica y objetos.....	693
Control de cyclecopy.....	696
Fecha y hora.....	697
Diseño y depuración .....	698
Fuentes y colores.....	701
Marcadores.....	711
Comandos y funciones varios .....	712
Control del dispositivo de salida .....	715
Funciones interactivas de PDF.....	731
Claves predeterminadas y palabras clave .....	733
Imprimir procesamiento de archivos .....	735
Control de carro de la impresora .....	738
Control de recursos.....	739
Elementos de RPE.....	740
Funciones de transformación.....	746

La información en Sugerencias de programación pretende dar respuesta a preguntas frecuentes sobre cómo usar varios aspectos del lenguaje VIPP®.

## Consumir frente a ejecutar

Cuando se ejecutan los trabajos VIPP®, los datos y los recursos no se procesarán de la misma manera. Los datos, todo lo que sucede después de un comando **STARTLM**, **STARTDBM** o **STARTXML**, se consumen. Los recursos VIPP®, como formularios, JDT, DBM. o XJT, se ejecutan.

Los parámetros de atributos como fuentes, color, etc son elementos de VIPP® que son interpretados por comandos de VIPP® como **SHMF**, **SHP**, etc. independientemente de su origen en cadenas consumidas o ejecutadas. Las anotaciones de caracteres octales son elementos PostScript y solo se interpretan cuando se ejecutan las cadenas. Por lo tanto, si un campo de datos (de un campo DBF o GETFIELD) contiene notaciones de caracteres octales, puede ser ejecutado para obtener la interpretación de los caracteres octales. Esto puede realizarse con VIPP® mediante una construcción VSUB2 como en este ejemplo:

```
( ($$VAR1 . ) ) VSUB2
```

Asumiendo que VAR1 contiene xxx\127yyy, la secuencia descrita más arriba sustituye VAR1 con su contenido, lo que genera una cadena intermedia:

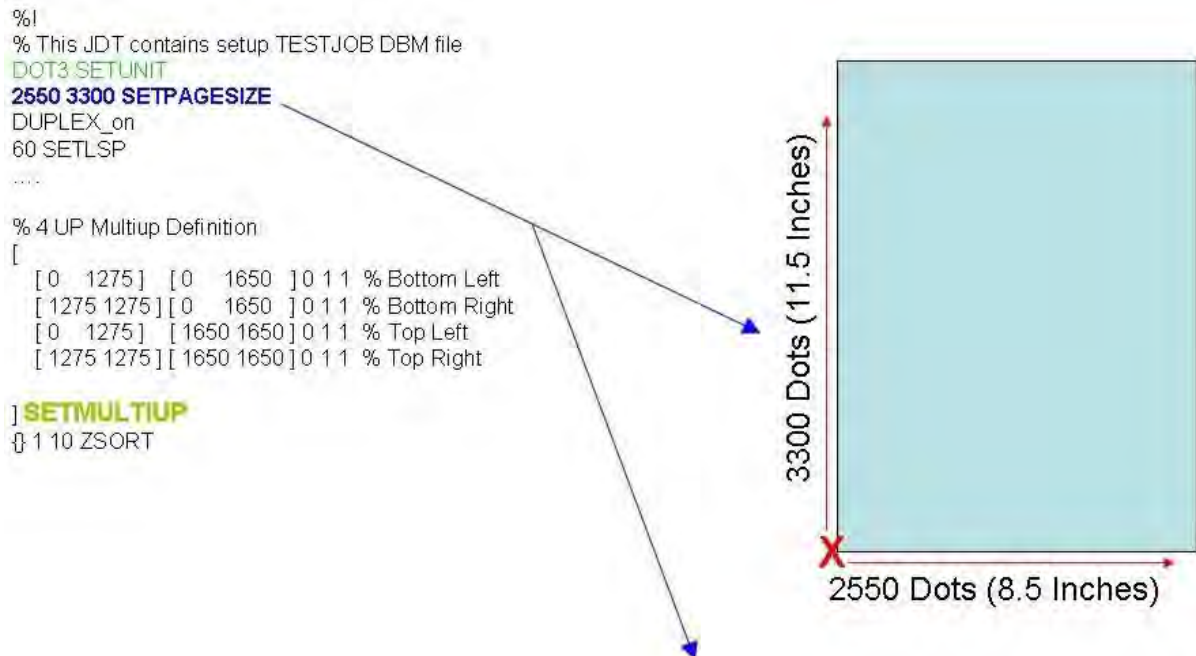
```
( (xxx\127yyy) )
```

A continuación, la cadena intermedia se ejecuta para generar el resultado final:

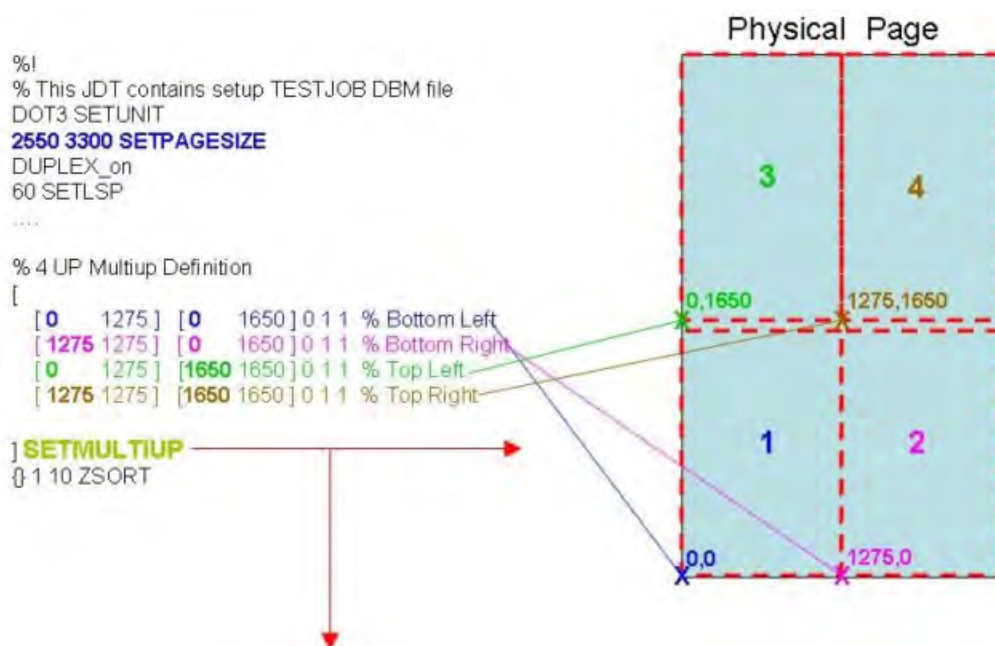
```
(xxxWyyy)
```

## Origen 0,0 para la posición de la página lógica y objetos

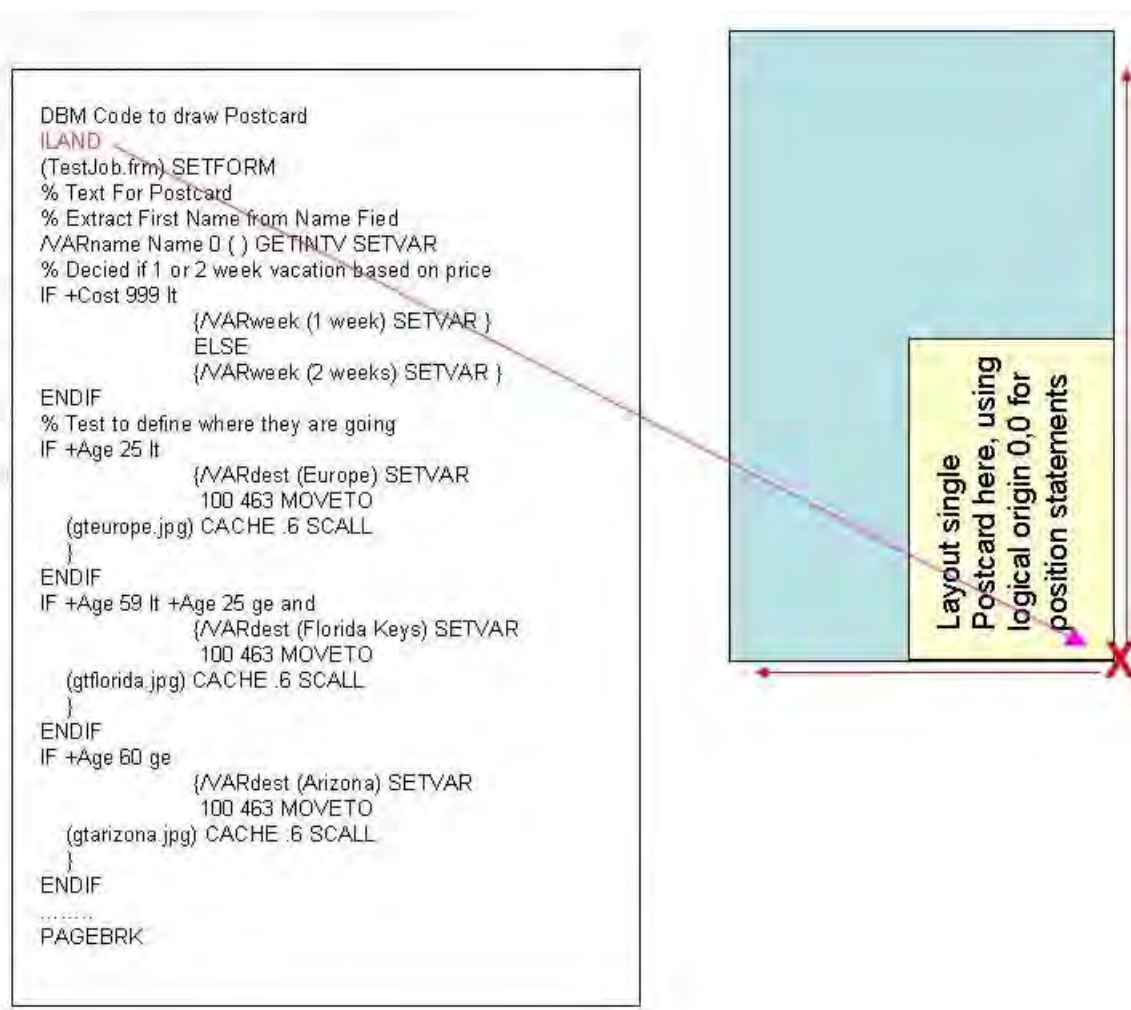
Los gráficos más abajo describen la posición de origen (X, Y) y el diseño que se usan habitualmente para crear una aplicación de tarjeta postal cuatro en 1.



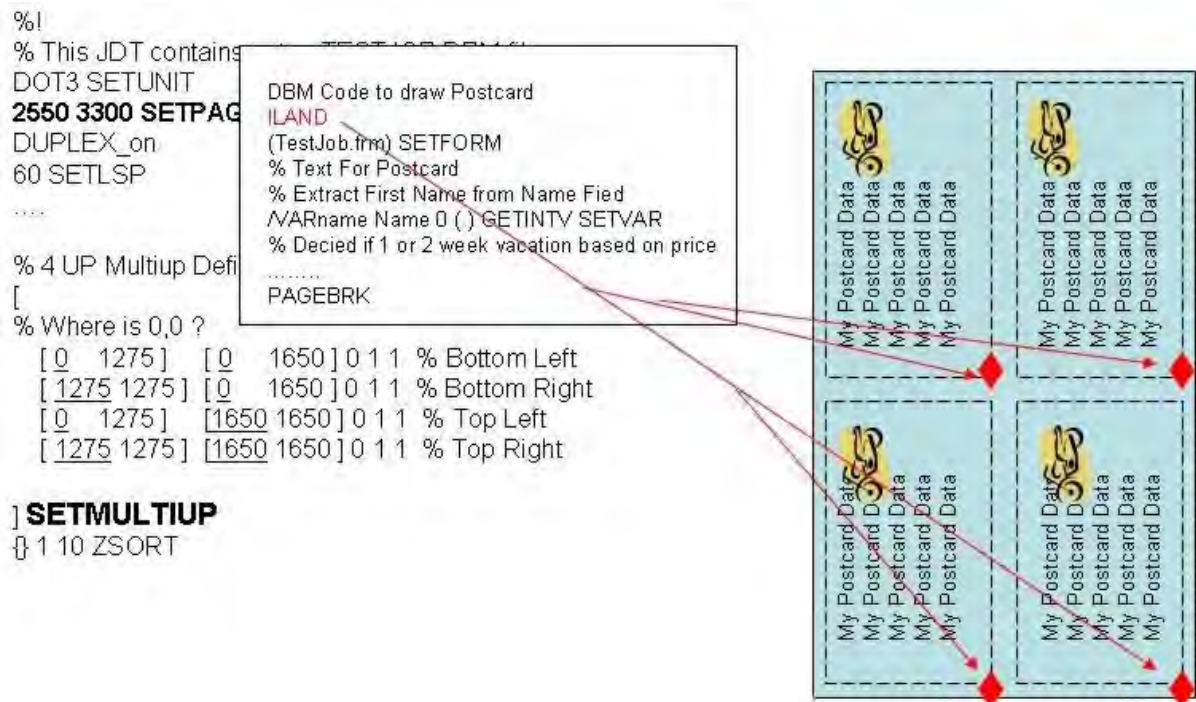
Define el tamaño de la página física. Uso de la definición de las unidades actuales, en este caso opciones de DOT3, 300 puntos por pulgada. Es la definición de una página vertical, carta EE. UU. La X roja marca el origen 0,0.



La instrucción **SETMULTIUP** define cada una de las cuatro páginas lógicas. Cada página lógica se define mediante la especificación de las coordenadas de inicio X e Y, basadas en la definición de página física para **SET-PAGESIZE** y, a continuación, la anchura y la altura desde esa coordenada de la página lógica. La X en color muestra la posición inicial de X e Y en función del origen de la página física de cada una de las cuatro páginas lógicas que se definen. Desde esa posición, cada página lógica tiene 1275 puntos de ancho y 1650 puntos de altura. El orden en el que las páginas lógicas pueden usarse depende del orden en el que se definen en el comando **SETMULTIUP**.



En el **DBM**, el comando **ILAND** configura el origen 0,0 según lo indicado por la X roja. Diseñe una sola postal mediante la X roja como origen 0,0 de la página. Si es necesario, diseñe una página frontal y una contraportada. Posteriormente, al implementar el comando **ZSORT**, puede incluirse el código de página anterior y posterior en una instrucción **IF BACK IF/ELSE/ENDIF**.



Según la combinación del comando **SETPAGESIZE**, que define el tamaño de la página física, el comando **SETMULTIUP**, que define las cuatro páginas lógicas en la página física en el archivo JDT, y el comando **ILAND** en el archivo DBM que define un formato de **ILAND** para la postal (opcional), el resultado se muestra más arriba. Las cuatro postales se colocan en la página física, cada una en las áreas de la página lógica definidas por el comando **SETMULTIUP**. La tarjeta postal está en orientación **ILAND** con el origen 0,0 que se muestra a través del diamante rojo. El orden de disposición de los registros se define según el orden de las páginas lógicas especificado en la instrucción **SETMULTIUP**.

## Control de cyclecopy

Los comandos del control del ciclo de copias son:

CHKPOINT	COLLATE_on	RPCOUNT	SETMAXCOPY
COLLATE_dbm	COPYRANGE	RPLEFT	SPOOLNAME
COLLATE_off	REPEAT	SETCYCLECOPY	

Para obtener más información sobre las funciones de control del ciclo de copias tratadas, consulte [Uso de formularios y JDT de ciclos](#).

### USO DE FORMULARIOS Y JDT DE CICLOS

Los formularios y los JDT se pueden usar de forma cíclica para realizar tareas equivalentes a las proporcionadas mediante ciclos de formularios como, por ejemplo, a través del envío de cinco formularios a cada cliente. Los tres primeros formularios son idénticos, y los formularios cuatro y cinco son diferentes. Los datos del formulario llegan en el modo Sysout. Los datos de los tres primeros formularios se envían una vez, seguidos por las páginas de datos de los formularios cuatro y cinco. Utilice este JDT para realizar esta tarea:

```
%!
PORT% or whatever
3 SETCYCLECOPY
COLLATE_off
[ (form1.frm) (form2.frm) (form3.frm) ] SETFORM
.....% layout definition (RPE or listing)
/JDT1 { 3 SETCYCLECOPY } XGFRESDEF
/JDT2 { 1 SETCYCLECOPY } XGFRESDEF
[ (JDT1) (JDT2) (JDT2) ] SETJDT
```



## Fecha y hora

Las variables de fecha y hora son:

D_DD	D_MOL	GETDATE	T_HH2
D_DOY	D_MOS	SETDATE	T_MM
D_DWL	D_YY	SHIFTDATE	T_SS
D_DWS	D_YYYY	T_AMPM	T_TZN
D_MO	DÍAS	T_HH	

## Diseño y depuración

Los comandos de diseño y depuración son:

AUTOGRID	NMP_off
BLGRID	TLGRID
FSHOW	XGFDEBUG

Utilice esta información para diseñar y depurar los trabajos de VIPP®. La información se divide en las siguientes categorías:

- **Diseño de la aplicación**
- **Herramientas de depuración y documentación**
- **Creación y mantenimiento de recursos**

### DISEÑO DE LA APLICACIÓN

FreeFlow VI Design Pro (VDP), anteriormente VIPP® IDE, es una herramienta que proporciona tres modos de edición, un editor de texto, un editor inteligente y un editor de interfaz gráfica de usuario. El conjunto completo de comandos de VIPP®, incluidos los de gráficos controlados por datos, está disponible para los usuarios de VDP. La interfaz gráfica proporciona información inmediata sobre los cambios realizados en el código, lo que reduce en gran medida el intervalo de tiempo necesario para diseñar y codificar las aplicaciones de VIPP®.

VDP es la mejor herramienta de diagnósticos de las aplicaciones VIPP®, dado que el código VIPP® se puede ejecutar en el VDP y los resultados se muestran en la ventana de la interfaz gráfica del usuario. VDP permite realizar cambios y ver los resultados de forma electrónica, sin impresiones.

VI Design Express (VDE) es un complemento de aplicación para usar con Adobe® InDesign. Está diseñado para que un artista gráfico pueda:

- Crear aplicaciones que contengan información variable.
- Exportar la aplicación de datos variables.
- Imprimir la aplicación a velocidades de producción cuando exporte el trabajo usando el formato VI Project Container (.vpc).
- Configurar la alimentación en línea y el acabado para la aplicación (dependiente del dispositivo).
- Permitir más modificaciones de diseño avanzado usando el código basado en la herramienta FreeFlow VI Design Pro o permitir el procesamiento de PDF mediante VI eCompose.

Las aplicaciones VDE pueden abrirse en VI Design Pro para fines de diagnóstico.

### HERRAMIENTAS DE DEPURACIÓN Y DOCUMENTACIÓN

Estos comandos se utilizan para la depuración de aplicaciones y la generación de documentación:

XGFDEBUG	Comandos de VIPP®
NMP_off	Comandos de VIPP®
SOF_off	Comandos de VIPP®
AUTOGRID	Variable integrada de VIPP®
BLGRID	Variable integrada de VIPP®
TLGRID	Variable integrada de VIPP®
FSHOW	Comandos de VIPP®
SHPOS	Variable integrada de VIPP®

Para más información, consulte la [Encontrar la causa de un error](#) y la *Guía del usuario de FreeFlow VI Design Pro*.

## CREACIÓN Y MANTENIMIENTO DE RECURSOS

Esta sección contiene información sobre estos temas:

- [Aplicaciones de documentos](#)
- [Herramientas de imagen](#)
- [Bibliotecas de recursos](#)
- [Editores de texto](#)

### Aplicaciones de documentos

Aplicaciones de procesamiento de documentos de alto nivel, como Word, FrameMaker, Excel y PageMaker, se pueden usar para crear recursos como los formularios o los segmentos. Una vez creado el documento, puede generar un archivo PostScript en el disco mediante un controlador de PostScript correspondiente. A continuación, coloque el archivo PostScript en un formulario VIPP® e invóquelo mediante VIPP® a través de CACHE/SETFORM o CACHE/SCALL. También se puede convertir un archivo TIFF con los servicios de descomposición.



**Nota:** Los controladores de PostScript no siempre producen códigos de PostScript adecuados para la impresión de producción. Por lo tanto, el uso de la herramienta del servicio de descomposición DocuPrint NPS puede ser más eficaz.

### Herramientas de imagen

Utilice el software de control del escáner o los editores de TIFF o JPEG para crear, editar y transformar imágenes. Para obtener más información acerca de los archivos TIFF y JPEG admitidos, consulte [ICALL](#).

### **Bibliotecas de recursos**

Por motivos de seguridad, los recursos de producción y de desarrollo normalmente se almacenan en bibliotecas independientes. El archivo de enlace de VIPP® hace referencia a las bibliotecas de producción. Para obtener más información, consulte Archivos de configuración y enlace en la *Guía del usuario de FreeFlow VI Compose*. Para acceder a las bibliotecas de desarrollo, los trabajos de prueba pueden incluir comandos SETxPATH que hacen que VI Compose acceda a los recursos en las bibliotecas para usarlos durante la duración del trabajo.

### **Editores de texto**

Todos los recursos, excepto las imágenes y las fuentes, pueden crearse mediante un editor de texto normal disponible en cualquier sistema. Los editores de texto son las herramientas más fáciles de usar a tal fin, dado que permiten el uso de texto normal y los recursos se pueden mantener directamente desde las bibliotecas en las que se almacenan.

## Fuentes y colores

Los comandos de fuentes y colores son:

GLT	INDEXLSP	SETCOL	SETTXB
HCOLOR	INDEXOTL	SETENCODING	SETTXC
INDEXALIGN	INDEXSST	SETFONT	SETTXS
INDEXBAT	MPR	SETFTSW	SETV2HCONV
INDEXCOLOR	SETBAT	SETGEP	SETV2HTABLE
INDEXFONT	SETCJKENCMAP	SETKERN	
INDEXKERN	SETCJKRULES	SETOTL	

La información de fuentes, colores y variables incluye estos temas:

- [Aplicar atributos para las fuentes](#)
- [Tonos de color](#)
- [Fuentes de códigos de barras y fuentes fijas](#)
- [Espaciado](#)
- [Fuentes de múltiples bytes](#)
- [Fuentes recodificadas e instrucciones de run/RUN](#)
- [Configuración de tamaño automático de las fuentes](#)
- [Configuración de negrita en el centro de un registro o de una línea](#)
- [Impresión especializada](#)
- [Fuentes de correlación y GlossMark \(GL/CR\)](#)
- [Uso de fuentes de varios bytes \(CJK\)](#)
- [Uso de archivos PostScript con fuentes integradas como formularios](#)

Para obtener más información acerca de las Colorkeys integradas, consulte [VIPP® Colorkeys](#).

### APLICAR ATRIBUTOS PARA LAS FUENTES

Los atributos como, por ejemplo, negrita, cursivas y negrita/cursivas no pueden aplicarse a una determinada fuente a menos que la fuente con el atributo esté instalada en el sistema.

El mecanismo usado para aplicar atributos a fuentes se basa en la declaración de familias de fuentes en el archivo de lista de fuentes, `xf/encoding/fontlist`, o cualquier otra lista de fuentes en [SETENCODING](#). Una familia de fuentes está formada por cuatro fuentes encapsuladas entre comandos `/STARTFF` y `/ENDFF` en el archivo, como máximo.

Si alguna de las fuentes constituyentes se selecciona, se puede acceder al resto mediante los nombres de fuente `/~REG`, `/~BLD`, `/~ITL` o `/~BDI`.

Si la fuente no forma parte de una familia, o si falta la entrada correspondiente, el comando **SETFONT/INDEXFONT** se ignora y la fuente actual se mantiene activa.

## TONOS DE COLOR

Un tono de color se puede aplicar a cualquier argumento CMYK o RGB Colorkey a un comando VIPP® referenciando una Colorkey para **SETTXC**, **INDEXCOLOR**.

### Sintaxis

```
/colorname~T SETTXC  
(colorname~T) SETTXC
```

### Donde:

~                      indica que hay un valor de tono vinculado al nombre del color  
T                      es un valor de tono con un rango válido entre 0 y 1

### Ejemplos

A continuación se muestran tres ejemplos de tono de color:

```
(customColorkey~.5) SETTXC    % solid color key with a tint of 50%  
(BLUE~.3) SETTXC            % RGB color key with a tint of 30%  
/PURPLE~.7 SETTXC           % RGB color key with a tint of 70%
```

El factor de tono se aplica a cada componente de color en este ejemplo:

Si:

```
/Color1 [ .3 .6 .2 .8 ] SETCOL  
,  
(Color1~.3)
```

equivale a:

```
[ .09 .18 .06 .24 ]
```

Para RGB y gris, la fórmula es más compleja porque el intervalo de intensidad va de 1 (sin color) a 0 (cuatricromía) en vez de 0 a 1.

## TRANSPARENCIA DE COLOR

La transparencia de color se puede aplicar a cualquier argumento CMYK, RGB Colorkey a un comando VIPP® referenciando una Colorkey para **SETTXC**, **INDEXCOLOR**. Solo es efectivo en VI eCompose y APPE RIP. En un color PS RIP, la transparencia se consigue mediante un tono opaco.

**Sintaxis**

```
/colorname#T SETTXC
(colorname#T) SETTXC
```

**Donde:**

**#** indica que hay un valor de transparencia vinculado al nombre del color

**T** es un valor de transparencia con un rango válido entre 0 y 1

**Ejemplos**

A continuación se muestran tres ejemplos de transparencia de color:

```
(customColorkey#.5) SETTXC    % Solid color key with a transparency of 50%
(BLUE#.3) SETTXC              % RGB color key with a transparency of 30%
/PURPLE#.7 SETTXC            % RGB color key with a transparency of 70%
```

**COLORES DIRECTOS FRENTE A COLORES DE PROCESAMIENTO**

Los colores CMYK se pueden definir como un conjunto de valores CMYK usado por el DFE para reproducir el color. Esto se denomina color de procesamiento.

Sin embargo, algunos DFE tienen la capacidad de procesar colores especiales en el espacio de color CMYK como colores directos o personalizados. Algunos colores se identifican a través de un determinado nombre de color y un conjunto de valores de aproximación de CMYK que se usan en el DFE si no tiene la capacidad de procesar colores directos o si el nombre del color no se ha definido en este contexto.

VI Compose es compatible con los dos tipos de colores según la sintaxis utilizada para hacer referencia a estos:

- Para definir el nombre del color y los valores CMYK asociados con el comando SETCOL:

```
/MyCustomColor [ c m y k ] SETCOL
```

- Para usar el color como color de procesamiento, consulte el color en cualquier comando VIPP® que requiera una Colorkey según el nombre de color normal:

```
/MyCustomColor [ c m y k ] SETCOL
```

- Para procesar el color como color directo, haga referencia al color colocando el nombre del color entre paréntesis u, opcionalmente, precedido por una barra inclinada si el nombre no contiene ningún espacio:

```
(MyCustomColor) SETTXC
/MyCustomColor SETTXC
```

## Ejemplos

```
/MyPersonnalRed [0 1 1 .8 ] SETCOL
...
(MyPersonnalRed) SETTXC      % creates a spot/custom color
/MyPersonnalRed SETTXC      % creates a spot/custom color
MyPersonnalRed SETTXC      % creates a process color
```

Para procesar un color como color directo, si el nombre contiene espacios o si se aplica un tono, los paréntesis son obligatorios. Consulte [Tonos de color](#).

```
(Another Custom Color) [ c m y k ] SETCOL
```

```
...
```

```
(Another Custom Color) SETTXC
```

VIPP® incorpora una recopilación de colores directos predefinidos. Consulte [Simulación de color sólido estucado y sin estucar](#).

## FUENTES DE CÓDIGOS DE BARRAS Y FUENTES FIJAS

Hay diversas fuentes de tipo 1 de tamaño fijo y de código de barras. Para descargarlas, consulte [Descarga de fuentes](#).

Las fuentes de Adobe se pueden utilizar en las impresoras DocuPrint NPS y en el servidor de impresión de FreeFlow. Póngase en contacto con Adobe para obtener las fuentes de Adobe, y haga el pedido de las fuentes directamente. Para obtener más información sobre la descarga de fuentes de Xerox, consulte [Descarga de fuentes](#).

## ESPACIADO

El espaciado hace referencia a la capacidad de ajustar el nivel de espacio entre los caracteres al imprimir un bloque de texto con una fuente determinada.

Hay dos tipos de espaciado, que pueden utilizarse por separado o conjuntamente:

- El espaciado de pares se aplica a determinados pares de caracteres
- El espaciado de seguimiento se aplica a todos los caracteres de manera uniforme

La información de espaciado de fuentes PostScript está disponible en un archivo llamado archivo Adobe Font Metrics (AFM). Archivos AFM por lo general, se suministran con el kit de fuentes cuando se adquiere de la fuente. Pueden obtenerse las especificaciones del archivo AFM en Adobe, Inc.

Se puede activar el espaciado para comandos **SHX** mediante:

- La sintaxis ampliada en la entrada de la lista de fuentes suministrada a los comandos **SETENCODING**
- **SETKERN** y **INDEXKERN**

Use la sintaxis para establecer el enlace entre una fuente y un archivo AFM al suministrar una lista de fuentes a SETENCODING:



```
/vippp_font_name [ /PS_font_name (AFM_filename) ]
```

#### Donde:

**AFM\_filename** es el nombre del archivo AFM que contiene la información de espaciado de la fuente asociada.

El archivo AFM se encuentra en una de las bibliotecas referenciadas por **SETMPATH** o **SETEPATH** o **SETPPATH** en el modo de proyectos.

El espaciado se desactiva de manera predeterminada.

#### Ejemplos

```
/NHE [ /Helvetica (helvetica.afm) ]
```

#### Archivos AFK

Los archivos AFK son archivos utilizados por VI Compose. Estos archivos genéricos, que son subconjuntos de archivos AFM, contienen información genérica de espaciado que se puede usar si un determinado archivo AFM de una fuente en particular no está disponible. Los archivos AFK solo existen en el directorio `xf/encoding` y son referenciados en `xf/encoding/fontlist` si un usuario activa el espaciado con una de las fuentes de VIPP® indicadas. Hay dos juegos de archivos prefijados, una para serif y otra para las sans serif.

Los archivos AFK pueden utilizarse con cualquier fuente. Seleccione una de estas en función del tipo (serif o sans serif) y los atributos de la fuente (normal, negrita, cursiva, negrita-cursiva). Sin embargo, dado que son genéricos, el resultado no puede ser totalmente preciso. Para obtener resultados precisos, debe utilizarse el archivo AFM original proporcionado por Adobe.

#### Ejemplos

Esta sintaxis hace referencia a un archivo AFK/AFM en una lista de fuentes:

```
/new_font_name [ /PS_font_name (AFK/AFM_filename) ]
```

Esta sintaxis hace referencia a un archivo AFK/AFM en el SETENCODING en línea:

```
[ /new_font_name [ /PS_font_name (AFK/AFM_filename) ]] (encoding table)
SETENCODING
```

#### FUENTES DE MÚLTIPLES BYTES

En la documentación de Variable Information Suite, el término fuentes de varios bytes se usa para describir las fuentes que requieren más caracteres de los que pueden especificarse en un byte de 8 bits. Las especificaciones de VIPP® incluyen fuentes chinas, japonesas y coreanas.

#### FUENTES RECODIFICADAS E INSTRUCCIONES DE RUN O RUN

Algunas aplicaciones utilizan los comandos run o RUN para cargar las fuentes. Esta carga de fuentes funciona si las fuentes no necesitan ser recodificadas.

La recodificación se ha optimizado y ahora solo se produce la primera vez que se realiza una llamada a la fuente. Dado que es probable que esto se produzca a mitad de la página, las fuentes pueden definirse en una VM global para evitar el almacenamiento o la restauración.

Cuando se utiliza run o RUN, las fuentes se definen en la máquina virtual local y no pueden volver a ser codificadas con SETENCODING.

Por tanto, Xerox recomienda que las instrucciones run/RUN existentes sean reemplazadas por la sintaxis SETENCODING con una referencia del archivo de fuentes.

Por ejemplo, cambie esta instrucción:

```
(myfont.pfa) RUN
```

a:

```
[ /R_myfont (myfont.pfa) ] null SETENCODING
```

Al ejecutar las fuentes antes de VIPP® en algunos archivos de inicio PS, las instrucciones de ejecución se encapsulan de la manera siguiente:

```
currentglobal true setglobal          % save VM mode and force global VM
(.../.../font1) run
(.../.../font2) run
....
setglobal                             % restore VM mode
```

## CONFIGURACIÓN DE TAMAÑO AUTOMÁTICO DE LAS FUENTES

Use **SETGRID** y **SETMARGIN** para cambiar los caracteres por línea (CPL) y las líneas por página (LPP). Esto facilita el ajuste a escala de fuentes fijas y, por lo tanto, LPP y CPL. Después de configurar estos dos parámetros, use el comando **SETFONT** con 0 como tamaño para el ajuste a escala automático.

## CONFIGURACIÓN DE NEGRITA EN EL CENTRO DE UN REGISTRO O DE UNA LÍNEA

Para especificar si desea negrita en el centro de una línea, use **SHMF**, **SHMf** y **SHmf** en la entrada RPE:

```
[ {0 SHMF} 0 Xinit Xdispl Yinit Ydispl rec_pos length /font color ]
```

Defina las fuentes con un solo carácter mediante **INDEXFONT**. El cambio ha sido producido por la secuencia de cambio de fuentes. El valor predeterminado es //. Para simular el modelo XES, redefina la secuencia y asigne la fuente en negrita con el valor b, como se muestra en este ejemplo

```
<1B> SETFTSW
/p /NHE 12 INDEXFONT % normal font
/b /NHEB 12 INDEXFONT % bold font
```

## SIMULACIÓN DE COLOR SÓLIDO ESTUCADO Y SIN ESTUCAR

La lista de Colorkeys predefinidos disponible para trabajos VIPP® incluye la simulación de colores sólidos estucados y no estucados según lo admitido por el servidor de impresión FreeFlow.

Hay más de 1000 claves de simulación de color sólido predefinidas. Se puede esperar la presencia de algunas

diferencias de color de un dispositivo a otro, ya que la salida del color se ve afectada por diversos factores como, por ejemplo, los materiales de impresión, las tintas, la configuración y el motor de color de la impresora. Para obtener la mejor representación de los colores sólidos, los nombres de Colorkey en los archivos .cck deben coincidir con los nombres de los colores usados por la impresora calibrada. Si los nombres de Colorkey coinciden, los valores de CMYK calibrados en la impresora sobrescriben los valores CMYK de los archivos .cck. Se puede producir una coincidencia incluso si el nombre de Colorkey .cck termina con un carácter s adicional. Consulte Colores sólidos personalizados estucados y no estucados en la *Guía del usuario de FreeFlow VI Compose*.

Para seleccionar un color sólido, utilice una Colorkey sólida como con cualquiera de las Colorkeys predeterminadas en `/usr/xgf/xgf.gep`.

### Sintaxis

```
(solidcolorname) SETTXC
```

### Ejemplo:

```
(customColorkey) SETTXC
```

Para ver muestras de las Colorkeys sólidas, imprima samccc.ps para la simulación del color sólido estucado o samccu.ps para la simulación de color sólido no estucado. Ambos archivos se encuentran en `xgf/demo`.

## IMPRESIÓN ESPECIALIZADA

La impresión especial se describe en profundidad en Impresión especial con VIPP® que se incluye en la *Guía del usuario de FreeFlow VI Compose*. Lea detenidamente el capítulo de la impresión especial, ya que proporciona la información necesaria para configurar y usar la impresión especial. Es importante que preste especial atención a la sección Limitaciones de este capítulo. Algunas limitaciones de impresión especial también se detallan en las descripciones de **GLT** y **MPR**.

Dado que las limitaciones de las funciones dependen de varios factores, como los materiales de impresión, el motor de impresión, la calibración, el color, el tamaño de la fuente y el tipo de la fuente, los resultados pueden variar según el sistema. No se puede garantizar que los efectos de impresión especial funcionen en todos los dispositivos y, además, pueden superar las limitaciones del dispositivo en lo referente a la memoria o las posibilidades de procesamiento. En ese caso, los efectos pueden reducir la calidad de impresión, sufrir variaciones de color o causar roturas. Xerox recomienda realizar las pruebas correspondientes y las validaciones antes del envío para su producción.

A excepción de MicroText, las fuentes de impresión especial pueden imprimirse en un fondo blanco. Esto es un requisito para efectos brillantes, de correlación y fluorescencia. Para usar estos efectos en un área de color, como el efecto en blanco, use el comando **DRAWB** de VIPP® para trazar un cuadro en blanco bajo el área del efecto.

La impresión especializada es compatible con los dispositivos del servidor de impresión FreeFlow siguientes:

- Todos los efectos:

- iGen
- DC7000, DC8000
- DC6060/5252/2060/2045
- Solo Micro Text:
  - Nuvera
  - 4110/4590 EPS
  - Dispositivos HLC (solo texto negro)

## FUENTES GLOSSMARK Y DE CORRELACIÓN (GL Y CR)

Se satisfacen automáticamente los requisitos para imprimir el efecto de la tecnología Xerox® GlossMark® o de correlación en un fondo blanco. Si un efecto GL/CR se integra en cualquier parte de un documento, VIPP® crea automáticamente un fondo blanco bajo un efecto que el usuario no puede visualizar. La acción ocurre de forma automática si se usa una fuente GlossMark o de correlación que no usa la sintaxis -L2- en el nombre de fuente. Esto permite que el efecto GL/CR imprima en cualquier parte del diseño sin necesidad de ocuparse de los colores subyacentes.

Si se usa una fuente de correlación con la sintaxis -L2 en su nombre, VIPP® no crea automáticamente un fondo blanco. Las fuentes con esta sintaxis se consideran fuentes de correlación de capa dos, normalmente utilizadas por la capa superior en un efecto de impresión especial de correlación de dos capas. En este caso, VIPP® no se puede trazar ningún cuadro blanco ya que la capa de correlación inferior debe interactuar con la capa superior CR-L2. El uso de la fuente de correlación de CR-L2 en una sola capa es válido, pero no se recomienda. Si se usa la opción, se selecciona un área en blanco en el documento, o se traza un cuadro en blanco debajo del efecto.

Al crear el efecto de correlación de dos capas, es importante usar la fuente CR de la capa inferior y la fuente CR-L2 de la capa superior, debido a la interacción de dos capas del efecto. Si no se sigue este orden, solo el efecto de la capa superior es visible.

Si se crea un efecto de correlación de capa única, se puede usar cualquier fuente CR. Sin embargo, no se recomienda usar una fuente de nivel CR-L2. Para crear un efecto de una capa, seleccione un área en blanco en la página en la que se imprimirá el efecto, o trace un cuadro en blanco detrás de él. Imprimir una cadena con una fuente CR-L2 sobre un área en color del documento provocará resultados impredecibles.

Las siguientes mejoras también se aplican si se usa una fuente de correlación o GlossMark:

- el texto se rellena automáticamente con espacios hasta la anchura de la columna definida cuando se utiliza:
  - **SHMF, SHMf y SHmf** con la opción +200
  - **SHP y SHp** con la sintaxis colwidth o ajustar al cuadro
- el cuadro de texto se rellena verticalmente con líneas en blanco al usar la opción de ajuste al cuadro SHP y un espaciado 0 como en:  
`(string) [ width height 0 ] align SHP`

## USO DE FUENTES DE VARIOS BYTES (CJK)

VI Compose se basa, principalmente, en el intérprete de PostScript para gestionar la impresión de varios bytes. Las fuentes CID u OCF compatibles con el intérprete de PostScript están disponibles para VI Compose.

Sin embargo, es importante recordar que SETENCODING no se aplica a las fuentes de varios bytes. Se cifran las fuentes de varios bytes utilizando los recursos de PostScript CMaps cuyo nombre suele ser parte del nombre de la fuente (por ejemplo: /Ryumin-Light-90ms-RKSJ-H). Por esta razón, las fuentes de varios bytes nunca deben aparecer en las lista de fuentes SETENCODING y el nombre de las fuentes PostScript debe usarse directamente con los comandos **SETFONT** e **INDEXFONT**. La codificación del flujo de datos también puede coincidir con la codificación de las fuentes seleccionadas.

Todos los comandos **SHx** pueden utilizarse con fuentes de varios bytes.

Se amplía la sintaxis de comandos de SHP y SHMF con opciones seleccionables para temas relacionadas con los idiomas asiáticos. Estos comportamientos también dependen de la codificación y de las listas de caracteres especiales definidas en el archivo `xgf/src/cjk.def` mediante nuevos comandos como **SETCJKENCMAP**, **SETCJKRULES**, **SETV2HCONV** y **SETV2HTABLE**.

Los comandos como **FROMLINE**, **RPEKEY**, **GETFIELD**, **SETRCD**, **SETPCD** y **GETINTV** se adaptan para funcionar en límites de caracteres en vez de en límites de bytes si se selecciona una fuente de varios bytes.



Nota: No se pueden utilizar caracteres de varios bytes como nombres de variables, nombres de campos de DBF ni nombres de etiquetas de archivos XML.

## Escritura vertical

Todos los comandos **SHx** pueden utilizarse con fuentes verticales de varios bytes. En este caso, todos los comportamientos horizontales se convierten en comportamientos verticales.

En particular:

- La alineación izquierda se convierte en alineación superior.
- La alineación derecha se convierte en la alineación de la parte inferior.
- Centro horizontal se convierte en centro vertical.
- Justificación horizontal se convierte en justificación vertical.
- La anchura de la columna se convierte en la altura de la columna.
- El espaciado de la línea se aplica horizontalmente en lugar de verticalmente.
- La posición horizontal secundaria se convierte en la posición vertical secundaria.

En vez de aumentar la posición de impresión vertical, se reduce la posición de impresión horizontal. El texto vertical se envía de derecha a izquierda.

## USO DE ARCHIVOS POSTSCRIPT CON FUENTES INTEGRADAS COMO FORMULARIOS

Si una fuente contiene archivos PostScript, hacer referencia al archivo como un formulario en VI Compose puede afectar al uso de la memoria virtual (VM) y al rendimiento de la impresora.

Para evitar estos problemas, instale permanentemente las fuentes en la unidad de disco de la impresora y

quítelas del formulario, o ejecute el archivo PostScript mediante los servicios de descomposición en un DocuPrint NPS. Esto puede realizarse en modo ByteCompressed. A continuación, haga referencia al formulario directamente a través del comando **SETFORM** de VIPP® en un JDT o DBM. Para obtener más información, consulte Servicios de descomposición en la *Guía del usuario de FreeFlow VI Compose*.

## Marcadores

Los marcadores utilizados en el lenguaje VIPP® son:

%	% %BoundingBox	% %PagesPerBooklet	[=name=]
% %	% %EOD_XGF	% %XGF	
%!	% %EOF	\$\$name.	

## Comandos y funciones varios

- Creación de un archivo PostScript independiente de VIPP®
- Incorporación de color de realce en formularios de descomposición
- Inclusión de un objeto no VIPP® en un trabajo VIPP®
- Texto imprimible o referencia ( )
- Impresión con idiomas especiales y caracteres acentuados
- VI Compose y EBCDIC

### CREACIÓN DE UN ARCHIVO POSTSCRIPT INDEPENDIENTE DE VIPP®

Para imprimir un archivo VIPP® en un dispositivo PostScript que no está activado con VIPP®, se crea un archivo VIPP® autónomo que es la concatenación del archivo VIPP® y todos los recursos necesarios. Esta tarea requiere dos pasos.

1. Determinar todos los recursos necesarios para el archivo VIPP®.

Se puede obtener ejecutando el trabajo a través del servicio Datos demográficos. Para más información, consulte la *Guía del usuario de FreeFlow VI Compose*.

2. Concatene todos estos recursos y los archivos principales VIPP® correspondientes con el archivo VIPP®.

Esto implica el uso de **XGFRESDEF** y posiblemente de comandos **MAKEVMFILE**. Para obtener más información, póngase en contacto con un vendedor de Xerox.

### INCORPORACIÓN DE COLOR DE REALCE EN FORMULARIOS DE DESCOMPOSICIÓN

Usando el modo de bytes comprimidos, **SaveMaskBC**, se recomienda al utilizar servicios de descomposición NPS. Este modo incluye tanto color negro como color resaltado en el mismo archivo y su rendimiento es mejor que el TIFF. El único problema es la portabilidad, puesto que esta opción es propia de las impresoras DocuPrint NPS.

### INCLUSIÓN DE UN OBJETO NO VIPP® EN UN TRABAJO VIPP®

Para incluir objetos de página, como logotipos, gráficos, tablas, etc. creadas con una aplicación externa en un trabajo VIPP®, lleve a cabo los pasos siguientes:

1. Guarde el objeto como EPS, TIFF o JPEG.
2. Coloque el objeto en un directorio con la referencia **SETPPATH** (modo de proyectos) o **SETMPATH/SETIPATH** (modo heredado)
3. Llame al objeto del trabajo VIPP® de una de estas maneras:
  - a. **CACHE/SCALL** para colocar el objeto en cualquier ubicación de la página.
  - b. **CACHE/SETFORM/SETBFORM** si el objeto es un formulario de fondo que cubre toda la página.



## TEXTO IMPRIMIBLE O REFERENCIA ( )

Los paréntesis de encapsulación ( ) identifican las referencias o el texto imprimible como operandos de cadena para el siguiente comando VIPP®.

Las referencias o el texto imprimible pueden contener nombres de variables que se pueden sustituir con un valor mediante el comando **VSUB**.

Las referencias o el texto imprimible pueden contener un carácter octal precedido del carácter \.

### Sintaxis

```
(referencia o texto imprimible)
```

### Ejemplos

Si la referencia o el texto imprimible contiene paréntesis derechos o izquierdos individuales o la barra diagonal inversa ("\"), preceden el carácter barra diagonal inversa como se muestra en estos ejemplos.

Este ejemplo imprime Delimiter is ).

```
(Delimiter is \)) SHL
```

Este ejemplo imprime Delimiter is \.

```
(Delimiter is \)) SHL
```

Este ejemplo imprime Delimiter is (\$).

```
(Delimiter is ($) ) SHL
```

Este ejemplo imprime Xerox©. \251 es el valor octal del símbolo de copyright.

```
(Xerox\251) SHL
```

## IMPRESIÓN CON CARACTERES ESPECIALES

Las tablas de codificación utilizadas por VI Compose se encuentran en `/usr/xgf/encoding`. De forma predeterminada se utiliza **pcsun**, que está definida en `xgf.def`, y soporta caracteres acentuados tanto para PC como para codificación de Sun. Para utilizar una codificación distinta, modifique la tabla de traducción actual o cree una nueva tabla de traducción. Para obtener más información, consulte *Listas estándar, tablas, claves y atributos* en la *Guía del usuario de FreeFlow VI Compose*.

## VI COMPOSE Y EBCDIC

VI Compose utiliza las tablas de traducción, lo que permite el procesamiento de flujos de datos VIPP® codificados mediante varios esquemas de codificación de caracteres. Se han proporcionado varias tablas que incluyen la tabla EBCDIC. Estas tablas pueden ser modificadas, o se pueden crear nuevas tablas. Póngase en contacto con el analista de sistema para obtener asistencia técnica para crear o modificar una tabla.

Estas tablas son válidas solo para datos imprimibles, lo que significa que los formularios, JDT, etc., no pueden codificarse en EBCDIC. Los comandos incluidos en el archivo de datos, como los comandos de arranque de VIPP®, **STARTLM**, y los comandos integrados, **% %XGFSETJDT**, pueden codificarse en ASCII incluso si los

datos de entorno se codifican en EBCDIC. Sin embargo, el envío desde el host proporciona una opción para convertir EBCDIC a ASCII para la salida a la impresora.

Al usar un dispositivo DocuPrint NPS, tiene la opción de configurar una impresora virtual que incorpora al archivo de datos de entrada un archivo de inicialización VIPP® que se utiliza para formatear los archivos de datos. Para obtener más información, consulte DocuPrint NPS XGFNub en la *Guía del usuario de FreeFlow VI Compose*. Esto permite no insertar comandos VIPP® para el formateado inicial, sino enviar un archivo de datos del cliente en formato EBCDIC o ASCII sin ninguna modificación necesaria.

SETRCD y las cadenas de referencia SETPCD se deben codificar en EBCDIC mediante la notación hexadecimal PostScript. Este ejemplo busca la cadena de EBCDIC TOTAL en la posición 10:

```
/IF_TOTAL 10 5 /eq <E3D6E3C1D3> SETRCD
```

Al imprimir texto con comandos SH, EBCDIC y ASCII vuelven a codificar las fuentes que se pueden usar simultáneamente, por ejemplo:

```
[ /FontA /Helvetica ] (sun8) SETENCODING [ /FontE /Helvetica ] (ebcdic)
SETENCODING /FontA 12 SETFONT (texto ASCII...) SHL /FontE 12 SETFONT VARdata1 SHL
% ebcdic texto de un campo dbf o GETFIELD
```

Al imprimir texto con comandos SH, las fuentes recodificadas EBCDIC y ASCII se pueden mezclar, por ejemplo:

```
/A /FontA 12 INDEXFONT /E /FontE 12 INDEXFONT (//A texto ASCII //E$VARdata1. //A
ASCII texto de nuevo) 0 SHMF
```

## Control del dispositivo de salida

### CONTROL DE PÁGINA

Los comandos de control de páginas son:

ADV PAGE	GRIDSKIP	NEWFRONT	SETPBRK
BEGINIMP	NEWBACK	NEWSIDE	SKIPPAGE
ENDIMP	NEWFRAME	PAGEBRK	SLIPSHEET
GOTOFRAME			

La información de control de páginas trata el temas [Procesamiento de PCC y VFU](#).

### Procesamiento de PCC y VFU

Use estos comandos en el JDT para definir el procesamiento **PCC/VFU**:

```
/ANSI SETPCC [/saltar-clave1 número-línea1 /saltar-clave2 número-línea2 ...] SETVFU
```

### DISEÑO DE PÁGINA

Los comandos de diseño de la página son

BEGINFRAME	ORIBL	SETGUNIT	SETUTAB
BTRIM	ORITL	SETLFI	SETZEBRA
COLW	PAGEH	SETLKF	SHEETH
FORMSHIFT	PAGEW	SETLSP	SHEETW
HPOS	PLINES	SETMARGIN	DESPLAZAMIENTO
HPOS2	PORT	SETMAXBFORM	SLENGTH
IHEIGHT	PSIZE	SETMAXFORM	SSIZE
ILAND	SETBFORM	SETMULTIUP	TIFORI_off
IPORT	SETCOLWIDTH	SETPAGENUMBER	TIFORI_on
IWIDTH	SETFORM	SETPAGESIZE	TWOUP
LAND	SETFRAME	SETTAB	VPOS
LSP	SETGRID	SETUNIT	YINIT
ONEUP			

La información de diseño de la página trata estos temas:

- Asignación de la omisión para canalizar de una a varias partes superiores de los formularios
- Capacidad de los ciclos de formularios de VIPP®
- Datos de impresión en formularios posteriores
- Impresión de dos copias a la vez, lado a lado
- Configuración de saltos de página en modo de línea
- Cambio entre varias en 1 y 1 en 1
- Uso de SETMULTIUP en el modo de base de datos

### Asignación de la omisión para canalizar de una a varias partes superiores de los formularios

La asignación de la omisión para canalizar de una a varias líneas (modo varios en 1 limitado) no se admite actualmente en VI Compose. Sin embargo, se puede lograr un resultado similar mediante el comando **SETMULTIUP**.

### Capacidad de los ciclos de formularios de VIPP®

Esta es la sintaxis de ciclo de formulario del comando **SETFORM**.

```
[ (form1) null null (form1) null null (form1) (form2) null null (form1)
(form2) ] SETFORM
```

Este ejemplo produce un ciclo de 12 páginas de formularios en los que form1 se imprime en las páginas 1, 4, 7 y 11, form2 se imprime en las páginas 8 y 12, y ninguno de los dos se imprime en las páginas 2, 3, 5, 6, 9 ni 10.

### Número de página

Es un ejemplo de dos pasos sobre cómo contar páginas de subjugos en un trabajo, que permite la numeración de páginas como 1 de 2 o 15 de 99.

Coloque el código similar a este ejemplo en el JDT al que se llama antes de STARTDBM:

```
2 SETMAXCOPY
{ IF CPCOUNT 1 eq
  { /VAR_pctot ++          % compute total pages for document
    SKIPPAGE              % skip pages on 1st pass
  } ENDIF
} BEGINPAGE
```

Agregue un código similar al del ejemplo en el DBM en el punto en el que se llega al primer registro del nuevo documento (cliente).

```

IF .. new customer ..
{ PAGEBRK                % push last page of previous customer
  2 SETCYCLECOPY          % init new copy cycle (includes implicit CHKPOINT)
  IF CPCOUNT 1 eq
    { /VAR_pctot 0 SETVAR } % reset page total
  ELSE
    { (Page #/$$VAR_pctot.) VSUB 1 1 SETPAGENUMBER }
  ENDIF
}
ENDIF

```

### Datos de impresión en formularios posteriores

Para imprimir datos variables en la parte frontal y posterior de una página con distintos formularios en cada cara, realice una de las siguientes acciones:

- Utilice los ciclos de formularios y la sintaxis de JDT en una sola JDT, como se indica en este ejemplo. Es un poco complejo. Sin embargo, contiene la configuración necesaria en un JDT y evita la duplicación de opciones comunes como la orientación o la lista de fuentes.

```

%!
%%Title: jobxy.jdt
...
DUPLEX_on
[ (front.frm) (back.frm) ] SETFORM
x BEGINRPE
..... % RPE definitions for front page
ENDRPE /RPEfront INDEXRPE
y BEGINRPE
..... % RPE definitions for back page
ENDRPE /RPEback INDEXRPE
/JDTfront { RPEfront } XGFRESDEF
/JDTback { RPEback } XGFRESDEF
[ (JDTfront) (JDTback) ] SETJDT

```

Este es el ejemplo de archivo de datos.

```

%!
(jobxy.jdt) STARTLM
1page 1, line 1
  page 1, line 2
....
1page 2, line 1
  page 2, line 2
....
%%EOF

```

- Cree dos JDT independientes, front.jdt y back.jdt, y combínelos al principio del trabajo como en este ejemplo. Si los JDT se combinan de esta manera, no use el comando **DUPLEX\_on** dado que aplica un comando **NEWFRONT**.

```

%!
DUPLEX_on
[ (front.jdt) (back.jdt) ] STARTLM
1page 1, line 1
  page 1, line 2
....
1page 2, line 1
  page 2, line 2
....
%%EOF

```

En el modo de varias en 1, puede crear los datos en este orden asumiendo una configuración 3 en 1:

- página\_frontal 1
- página\_frontal 2
- página\_frontal 3
- página\_posterior 1
- página\_posterior 2
- página\_posterior 3
- front\_page 4, etc.

### Impresión de dos copias a la vez, lado a lado

Para imprimir dos copias de cada página de un documento lado a lado, el documento se produce de esta manera:

Hoja 1 / cara 1

Página 1 / página 1

Hoja 1 / cara 2

Página 2 / página 2

Hoja 2 / cara 1

Página 3 / página 3

Hoja 2 / cara 2

Página 4 / página 4

Hoja 3 / cara 1

Página 5 / página 5

Hoja 3 / cara 2

Página 6 / página 6

Para ello, combine estos comandos:

- TWOUP
- 2 SETCYCLECOPY
- COLLATE\_off

Recorte el trabajo impreso para crear dos copias clasificadas del documento.

### Configuración de varios formularios en un trabajo

Estos ejemplos se pueden usar para configurar varios formularios en un trabajo que:

- Use la opción planenumber que permite definir varias capas de formularios, apilando formularios en la misma página

```
(formA) 0 SETFORM
```

```
(formA) 1 SETFORM
```

```
(formA) 2 SETFORM
```

- Use la opción de ciclo de formulario que permite definir una lista de formularios aplicada de manera cíclica en páginas consecutivas.

```
[ (formA1) (formA2) (formA3) ] SETFORM
```

- Combine estas opciones. Este ejemplo especifica esta información:

- La página 1 recibe los formularios formA1, formB y FormC
- La página 2 recibe los formularios formA2, formB y FormC
- La página 3 recibe los formularios formA3, formB y FormC
- La página 4 recibe los formularios formA4, formB y FormC, etc.

```
[ (formA1) (formA2) (formA3) ] 0 SETFORM
```

```
(formB) 1 SETFORM
```

```
(formC) 2 SETFORM
```

En el modo de base de datos, se pueden llamar nuevos formularios después de cada **PAGEBRK**.

### Ejemplos

```
(formA) SETFORM
```

```
o
```

```
{ x y MOVETO (tiffA) 1 0 ICALL } SETFORM
```

```
.....
```

```
PAGEBRK
(formB) SETFORM
.....
PAGEBRK
(formC) SETFORM
.....
PAGEBRK
```

### Configuración de saltos de página en modo de línea

Hay cuatro modos de configurar saltos de página en un trabajo en modo de línea:

- Número de líneas por página, **SETGRID**
- Avance de página, hexadecimal 0C
- Byte PCC
- **% %XGF** PAGEBRK

El avance de página y PCC son excluyentes entre sí.

La detección del avance de página funciona con cualquier cadena contenida en la línea. Para obtener más información, consulte **SETPBRK**.

### Cambio entre varias en 1 y 1 en 1

Para imprimir las páginas 1 y 2 en la portada en modo dos en 1, imprima la página 3 en la contraportada en el modo 1 en 1, imprima las páginas 4 y 5 en la portada en modo 2 en 1, imprima la página 6 en la contraportada en el modo 1 en 1, etc. Introduzca lo siguiente al final del JDT:

```
.....% common normal JDT code
/JDT1 { TWOUP } XGFRESDEF
/JDT2 { } XGFRESDEF
/JDT3 { ONEUP } XGFRESDEF
[ (JDT1) (JDT2) (JDT3) ] SETJDT
```

Defina todas las opciones comunes en la parte superior del JDT y, a continuación, cree los tres JDT secundarios que se aplicarán de manera cíclica.

Si necesita usar otras instrucciones de formato para cada una de las tres páginas, defina tres tablas RPE, indéxelas mediante **INDEXRPE** y, a continuación, realice la llamada a los índices alfabéticos en los JDT secundarios.



### Uso de SETMULTIUP en el modo de base de datos

Al usar **SETMULTIUP** en el modo de base de datos, hay dos situaciones que debe considerar:

Si desea que cada llamada a DBM se imprima en la página lógica siguiente, es importante especificar el comando **SETMULTIUP** en la parte superior del archivo DBF antes del comando **STARTDBM**. Al introducir este comando en DBM, se reinicializa la secuencia de varias en 1 en cada llamada de DBM.

Si está codificando un DBM de varias páginas con un diseño de varias en 1, agregue un **NEWSIDE** o **NEWFRONT** al principio del DBM o coloque **SETMULTIUP** al principio del DBM.

### Uso de tablas para almacenar datos

Hay situaciones en las que se pueden acumular datos en una tabla para utilizarlos más adelante. Esto puede realizarse con un juego de comandos VIPP® dedicado:

#### Sintaxis para inicialización de la tabla:

```
/VARtablename [ [ /VARname1 /VARname2 ... /VARnameN ] ] SETVAR
```

Donde:

**/VARnameX** son los nombres de los elementos de la tabla.

#### Sintaxis para rellenar la tabla:

```
/VARtablename [ [ name1 name2 ... nameN ] ] ADD
```

Donde:

**nameX** son los nombres de variables que contienen los valores de una entrada en la tabla.

#### Sintaxis para ordenar la tabla (opcional):

```
VARtablename /VARkeyX /opt SORT
```

Consulte el comando **SORT** para obtener una descripción completa.

Sintaxis para ordenar la tabla (opcional):

```
{ processing code UPDATE } VARtablename FOREACH
```

Consulte el comando **UPDATE** para obtener una descripción completa.

Sintaxis para procesar la tabla:

```
{ processing code } VARtablename FOREACH
```

Consulte el comando **FOREACH** para obtener una descripción completa.

## Ejemplos

**Al principio del JDT, antes de STARTDBM:**

```
/VARaccountList [ [ /VARname /VARacctNumber ] ] SETVAR
```

**En la DBM:**

```
/VARaccountList [ [ CustomerName AccountNumber ] ] ADD
```

**En ENDJOB:**

```
VARaccountList /VARname /A SORT% sort alphabetically
100 3000 MOVETO (List of accounts:) SH
100 2800 MOVETO
{ VARname SH 1000 MOVEH VARacctNumber SH NL } VARaccountList FOREACH
PAGEBRK
```

## MARCA DE PÁGINA

Los comandos de marca de página son

ABSPOS	DRAWPIE	MOVEHR	SHJ & SHj
AZTEC	DRAWPOL	MOVETO	SHL & SH
BCALL	DRAWRDR	NL	SHMF, SHMf, SHmf
CLIP	ENDCLIP	OMRINIT	SHP y SHp
CUTMARK	ETCLIP	OMRSHOW	SHPATH
DATAMATRIX	FCALL	OTCLIP y ITCLIP	SHPOS
DEFINELAYOUT	FILLOMR	PDF417	SHR y SHr
DRAWB & DRAWBR	HDISP	QRCODE	SHT y SHt
DRAWBAR	ICALL	SAVEPP	SHX
DRAWBM & DRAWBRM	INDEXOTL	SCALL	SVPOS
DRAWC	IREVERSE_off	SETINDENT	TPATH
DRAWCRV	IREVERSE_on	SETLAYOUT	USPS4CB
DRAWPAR	MAXICODE	SETOTL	VDISP
DRAWPATH	MOVEH	SHC y SHc	

Se tratan estos temas de marcado de páginas:

- [Códigos de barras](#)
- [Ubicación de archivo TIFF dinámica en modo de línea](#)
- [ICALL en un JDT](#)

- [Ubicación de imágenes TIFF en varios directorios](#)
- [Combinación de datos con archivos TIFF](#)
- [Impresión de un bloque de texto sombreado](#)
- [Supresión del desplazamiento vertical con campos vacíos en un DBM](#)
- [Consideraciones sobre el rendimiento del archivo TIFF](#)

## Códigos de barras

Los comandos descritos en [Funciones de transformación](#) con los mismos nombres que las fuentes de código de barras comunes no son fuentes de códigos de barras, sino transformaciones que pueden gestionar una cadena de entrada y producir la cadena de la fuente que esperaría una fuente de código de barras de este tipo. Utilice las funciones de transformación para evitar tener que modificar la aplicación existente.

Excepto los códigos de barras PDF417, DATAMATRIX, MAXICODE y QRCODE, creados como una imagen, el resto de códigos de barras admitidos por VI Compose, 2OF5, CODE39, EAN13/EAN8, CODE128/EAN128, UPCA y POSTNET, requieren la realización de una llamada a una fuente de código de barras para imprimir el código de barras. No se suministran las fuentes de los códigos de barras como fuentes VIPP® estándar. Para descargar fuentes, consulte [Descarga de fuentes](#).

Los códigos de barras que no requieren transformaciones de datos se pueden imprimir mediante la impresión de cadenas de texto con la fuente de código de barras apropiada. Otros códigos de barras requieren que el flujo de entrada contenga caracteres especiales. Estos caracteres especiales pueden ser suministrados en la cadena de texto por la aplicación, o una función de transformación se puede aplicar a la cadena de texto para agregar caracteres especiales. La sintaxis del comando VIPP® incluye diversos comandos VIPP® que realizan transformaciones de datos de entrada, por lo que cuando se imprimen con una fuente de código de barras se produce la salida correcta del mismo.

Si la aplicación no suministra estos caracteres especiales en la cadena, el uso de una transformación evita tener que cambiar el programa de aplicación original, ya que la transformación gestiona los requisitos del código de barras. Una fuente del código de barras todavía es necesaria para imprimir el conjunto de caracteres correcto. Todas las transformaciones de Xerox se basan en las fuentes de códigos de barras; consulte [Descarga de fuentes](#).

Por ejemplo, un código postal de los EE. UU. habitual puede ser uno de los siguientes, de 5, 9 o 11 caracteres, y puede parecerse a uno de estos ejemplos:

- 90404
- 90404-2534
- 90404-2534+DPBC

Para imprimir en el formato de código de barras Postnet se necesita:

- Una fuente Postnet PostScript. Se pueden descargar dos fuentes postnet, MB043 y MB045; consulte [Descarga de fuentes](#).
- Suministrar a la fuente de código de barras una cadena de caracteres aceptables. El suministro de la cadena 90404-2534 no funcionará dado que el código de barras impreso debe contener caracteres especiales adicionales que no figuran en la cadena original, como caracteres de trama, dígitos de

comprobación, etc. Sin estos caracteres, el lector del código de barras no podrá entender el código de barras recibido y lo rechazará como un error.

### Obtención de la cadena

Obtenga una cadena aceptable a través de una de las dos maneras siguientes:

- El programa que genera los datos puede suministrar la cadena que necesita pasarse a la fuente de código de barras, incluidos los caracteres especiales.
- Utilice los comandos de transformación del código de barras VIPP®. Estos comandos VIPP® especiales permiten pasar la cadena de texto estándar al comando de transformación. El comando de transformación modifica la cadena de entrada y agregará los caracteres especiales, dígitos de comprobación, etc. como sea necesario, asumiendo que la cadena de entrada original es válida y que la fuente del código de barras que se utiliza es una fuente de código de barras admitida. VI Compose admite las fuentes suministradas por Xerox. Consulte [Descarga de fuentes](#).

Para utilizar el comando de transformación de VIPP® POSTNET, puede:

- Realizar una llamada a la fuente Postnet
- Pasar la fuente al código postnet estándar, usar el comando VIPP® POSTNET para transformar la cadena e imprimirla mediante cualquiera de los comandos de tipo **SHx**

### Ejemplos

Estos ejemplos realizan una llamada a la fuente Postnet:

```
/MB034 12 SETFONT
```

```
(90404-2534) POSTNET SH % with the dash
```

```
/MB034 12 SETFONT
```

```
(904042534) POSTNET SH % without the dash
```

```
/MB034 12 SETFONT
```

```
($ZIP.) VSUB POSTNET SH % if calling in a variable
```

Este ejemplo utiliza un índice de fuente creado para la fuente MB034:

```
/F1 /MB034 12 INDEXFONT % Postnet font
```

```
F1 (90404-2534) POSTNET 0 SHP % with the dash
```

En este ejemplo se activa una transformación de código de barras mediante una entrada RPE. En el siguiente ejemplo, el parámetro de alineación se reemplaza con un procedimiento que contiene la transformación POSTNET y el comando SH.

```
/F1 /MB034 12 INDEXFONT % Postnet font
```

```
5 BEGINRPE
```

```
...
```

```
10 FROMLINE
```

```
[ { POSTNET SH } 0 Xinit Xdisp Yinit Ydisp 11 9 /F1 BLACK ]
```

```
..
```

```
ENDRPE
```

### Ubicación de archivo TIFF dinámica en modo de línea

Utilice uno de estos métodos para llamar de forma dinámica a los archivos TIFF en el modo de línea:

- Use el comando **ICALL** en un NMP:

```
%%XGF 1000 1500 MOVETO (name.tif) 1 90 ICALL
```

La imagen se encuentra en una posición fija, independientemente de dónde se encuentre el NMP en el archivo de datos.

- Use **ICALL** en un procedimiento de alineación de entrada RPE para lograr la colocación dinámica en función de si el nombre de la imagen es fijo o debe extraerse de los datos:

```
[ {1 90 ICALL} 0 Xinit Xdispl Yinit Ydispl 0 (xxxxx.tif) /font color ]
```

o:

```
[ {1 90 ICALL} 0 Xinit Xdispl Yinit Ydispl rec_pos length /font color ]
```

En cualquier caso, la línea que usa la entrada RPE debe identificarse con una prueba; consulte **SETRCD** o mediante un prefijo RPE.

### ICALL en un JDT

Si intenta realizar un **ICALL** independiente en un JDT, se producen errores. Utilice uno de estos para realizar un **ICALL** en un JDT:

- Un formulario de VIPP® en línea, como se muestra en este ejemplo. En este ejemplo, el archivo de formulario VIPP® se coloca como plano 0, y la imagen TIFF se coloca como plano 1.

```
2 SETMAXFORM
```

```
(myform.frm) 0 SETFORM
```

```
{ PORT ORITL 0100 0100 MOVETO (focus.tif) 1 0 ICALL } 1 SETFORM
```

Realice la parte **ICALL** de una definición de RPE, como se indica en este ejemplo. En este ejemplo, se utiliza un valor constante, image.tif. Sin embargo, el nombre del archivo de imagen se puede obtener del archivo de datos mediante los parámetros **recpos** y **length** para encontrar el nombre en el registro. Consulte **FROMLINE** y **ICALL** para obtener más información sobre cómo utilizar estos comandos en una definición RPE.

```
3 BEGINRPE
```

```
[ {1 0 ICALL} 0 1400 0 1495 0 000 (image.tif) /TEXTS BLACK]
```

```
....
```

```
ENDRPE
```

Es posible que aparezcan efectos secundarios inesperados al marcar comandos como **SHXSHxICALL**, y **SCALL** en

un JDT, cuando codifique llamadas DIRECT.

Un JDT debe incluir solo comandos generales o globales de formato del trabajo. Un comando de marca escribe directamente en la página actual. Sin embargo, cuando se ejecuta el JDT en la fase de inicialización, no se define ninguna página actual.

### Ubicación de imágenes TIFF en varios directorios

**ICALL** crea la ruta completa a una imagen mediante la concatenación de las rutas de la biblioteca especificadas en el comando **SETIPATH** con el nombre de la imagen proporcionado como argumento. Las bibliotecas predeterminadas están disponibles en el archivo `/usr/xgf/src/xgfunix.run`.



#### Nota: Gestión simplificada de recursos de VIPP®

El uso de VI Project simplifica la administración de recursos de VIPP®. Consulte *Proyectos VI* en la *Guía del usuario de FreeFlow VI Compose* para obtener más información.

#### No mezcle la gestión de recursos de VI Project y la de los recursos antiguos de VIPP®

La gestión de recursos de VI Project y recursos antiguos de VIPP® (imglib, formlib, fontlib, jdtlib, etc.) no se deben mezclar en un solo trabajo.

Realice una de estas acciones para tener acceso a las imágenes TIFF en varios directorios:

- Coloque bibliotecas adicionales en `xgfunix.run`. Asegúrese de que no haya dos imágenes con el mismo nombre en dos bibliotecas. En ese caso, solo la primera imagen podrá ser accedida.
- Incluya un comando **SETIPATH** al principio del trabajo para especificar en qué bibliotecas debe realizarse la búsqueda.

```
(/disk330/tiff1/) SETIPATH
```

```
.....
```

```
(image1) 1 0 ICALL
```

- Incluya un **SETIPATH** vacío al principio del trabajo y especifique la ruta completa con cada **ICALL**.

```
() SETIPATH
```

```
.....
```

```
(/disk330/tiff1/image1) 1 0 ICALL
```

'()' SETIPATH desactiva todas las bibliotecas definidas en `/usr/xgf/src/xgfunix.run` para el trabajo actual y proporciona una ruta vacía. Cuando se utiliza esta sintaxis, no se realiza ninguna búsqueda. Por lo tanto, el acceso a la imagen se realiza de forma directa para cada **ICALL**.

Para tener acceso a las otras bibliotecas del trabajo, introduzca "()" como la primera biblioteca en `/usr/xgf/src/xgfunix.run`. El efecto es similar, dado que la imagen con la ruta completa se encuentra en la primera biblioteca, por lo que no es necesario realizar más búsquedas. Además, utilice el nombre base de las imágenes para continuar accediendo a las imágenes ubicadas en otras bibliotecas.

```
[ ()
```

```
(/usr/xgfc/imglib/)
```

```
(/usr/xgf/imglib)
```

```
] SETIPATH
```

- Especifique una ruta raíz mediante SETIPATH e introduzca una ruta secundaria adicional en ICALL. Especifique la ruta raíz en xgfunix.run.

```
(/disk330/) SETIPATH.....(tiff1/image1) 1 0 ICALL
```

Mediante el mantenimiento de la ruta (o, como mínimo, la ruta de la raíz) fuera del trabajo se puede evitar la necesidad de realizar cambios en la aplicación posteriormente si se producen cambios en la estructura del sistema de archivos (por ejemplo, tras la adición de nuevos discos, el cambio de nombre de los discos o el montaje de un sistema de archivos DOS).

### Posiciones de impresión principal y secundaria

Las posiciones de impresión principal y secundaria (PP) hacen referencia a la ubicación de la página (X, Y) que los comandos más abajo usan para colocar texto, imágenes o segmentos.

Las posiciones de impresión secundaria y principal vertical (Y) son siempre idénticas, y son referenciadas como posición de impresión vertical. Solo las posiciones de impresión secundaria y principal horizontal (X) difieren. Las posiciones de impresión principal y secundaria se definen como se indica a continuación:

- Posición de impresión principal = posición de impresión vertical + posición de impresión principal horizontal
- Posición de impresión secundaria = posición de impresión vertical + posición de impresión secundaria horizontal

Estos comandos se usan para configurar o ajustar las posiciones de impresión principal y secundaria, o colocar texto o imágenes en un trabajo VIPP®:

- **MOVETO** establece las posiciones iniciales de impresión principal y secundaria en los mismos valores.
- **SHC y SHc, SHJ y SHj, SHL y SH, SHMF, SHMf y SHmf, SHP y SHp, SHR y SHr, SHT y SHt y SHX** usan la posición de impresión principal actual para:
  - colocar texto
  - restablecer la posición de impresión principal horizontal según el último valor de MOVETO
  - aumentar la posición de impresión vertical hacia abajo según el valor **SETLSP**
- **SHL y SH, SHC y SHc, SHJ y SHj, SHMF, SHMf y SHmf, SHP y SHp, SHR y SHr y SHT y SHt** usan la posición de impresión secundaria actual para colocar texto y configurar la posición de impresión secundaria en la posición a la que se llega después de imprimir el texto.
- **ICALL y SCALL** usan la posición de impresión secundaria actual para colocar imágenes o segmentos y dejan la posición de impresión secundaria inalterada, independientemente del tamaño de la imagen o del segmento.
- **MOVEH** configura la posición de impresión horizontal secundaria según un valor suministrado.
- **MOVEHR** configura la posición de impresión secundaria según la última posición principal horizontal.
- **NL** restablece la posición de impresión horizontal secundaria y principal al último valor de MOVETO, y aumenta la posición de impresión vertical hacia abajo según el valor suministrado o SETLSP.
- **PAGEBRK** restaura la posición de impresión principal y secundaria a 0,0.

- **SAVEPP** guarda la posición de impresión secundaria actual para un uso posterior mediante **HDISP**, **VDISP**, **SHPOS** y **SVPOS**.

### Combinación de datos con archivos TIFF

Para combinar un archivo TIFF con datos, use una de estas opciones:

- Cree un formulario (`/usr/xgfc/formlib/xyz.frm`) con este contenido:

```
%!
```

```
{ x y MOVETO (abl.tif) 1 0 ICALL } FSHOW
```

A continuación, realice la llamada al formulario en el JDT como se indica a continuación:

```
(xyz.frm) SETFORM
```

- Genere el formulario en el JDT como se indica a continuación:

```
{ x y MOVETO (abl.tif) 1 0 ICALL } 0 SETFORM
```



Nota: En el ejemplo anterior, x e y son el origen de la esquina superior izquierda de la imagen.

- Haga referencia al archivo TIFF en una entrada RPE. Consulte la opción del procedimiento de alineación en las extensiones **FROMLINE**.
- Realice la llamada al archivo TIFF en un procedimiento **ENDPAGE**.

### Impresión de un bloque de texto sombreado

El código de ejemplo más abajo muestra cómo crear un bloque de texto sombreado que crece o se reduce en función del número de líneas en el bloque de texto.

El archivo imprime un bloque de direcciones similar a este ejemplo:

```
Mr David Kirk  
Xerox Corporation  
Suite xyz  
El Segundo, CA 90245
```

En este ejemplo, el bloque de dirección se imprime en un fondo sombreado, y la anchura del fondo se define a través de un comando **SETCOLWIDTH**.



```

%%Title: batshade.dbm
%%Creator: VI Designer DataBase Mode.
%%CreationDate: 08-24-1999.
%%This sample prints a form letter from database fields.

DOT3 SETUNIT
PORT
/NTMB 20 SETFONT
90 SETLSP

% See SETBAT for SETBAT parameters
% This command sets the background attribute
/DAVEBATkey [XLTR 0 /COLW'+40 0 /LSP 0 -20 2 -.4 1 0 0 0] SETBAT

% Set the background attribute ON
/DAVEBATkey SETTXB
800 SETCOLWIDTH 175 2910 MOVETO
($$Title.$$Fname. $$Lname.) VSUB 0 SHP
Addr1 0 SHP
Addr2 0 SHP
City 0 SHP
($$State. $$Zip.) VSUB 0 SHP
% Set the background attribute OFF
null SETTXB

175 2254 MOVETO
(Start letter here ...) SH

PAGEBRK

```

### Supresión del desplazamiento vertical con campos vacíos en un DBM

Utilice uno de estos elementos para suprimir el desplazamiento vertical si un campo en un DBM está vacío:

- Use **SHP y SHP**, dado que suprime el desplazamiento vertical si la cadena está vacía (al contrario que otros comandos **SHX**):

```

1600 800 MOVETO
($$FIRST_NAME. $$LAST_NAME.) VSUB SHL
ADD1 SHL
ADD2 0 SHP
($$CITY_NAME. $$STATE. $$ZIP.-$$ZIP4.) VSUB SHL

```

En este ejemplo, si ADD2 está vacío, no hay ningún espacio en blanco después de ADD1. Sin embargo, use SHP solo cuando sea necesario, ya que requiere un mayor procesamiento con respecto a otros comandos SHx.

- Use **IF/ELSE/ELIF/ENDIF**:

```

1600 800 MOVETO
($$FIRST_NAME. $$LAST_NAME.) VSUB SHL
ADD1SHL
IF ADD2 () ne { ADD2 SHL } ENDIF
($$CITY_NAME. $$STATE. $$ZIP.-$$ZIP4.) VSUB SHL

```

### **Consideraciones sobre el rendimiento del archivo TIFF**

Si desea incluir diversas imágenes TIFF en un formulario, realice la llamada a estas imágenes TIFF desde el formulario, en vez de a través de formularios independientes en el JDT. Si los archivos TIFF y el formulario son estáticos en todo el documento o en un patrón del documento, como con un formulario de ciclo, ejecútelos como trabajos independientes a través de los servicios de descomposición y haga referencia al archivo resultante como un único formulario.

Se puede conseguir el mejor rendimiento de la impresora cuando los archivos TIFF no requieran del escalado ni el giro, y si el tamaño del archivo se mantiene al mínimo. Reduzca el tamaño del archivo de varias maneras, como a través del recorte del espacio en blanco adicional de una imagen escaneada.

Aunque VIPP® puede hacer referencia a archivos TIFF que no se han enviado junto con el archivo de envío, puede evaluar el rendimiento de la red al realizar esta tarea, dado que los archivos podrían residir en un servidor de red y diversas impresoras de la misma red podrían compartirlos.

## Funciones interactivas de PDF

Los comandos de control de PDF son:

BOOKMARK	PDFINFO
PDFDEST	PDFOPEN
PDFDEVICE	SETPIF

Las funciones interactivas VIPP® PDF (PIF) son un conjunto de comandos VIPP® que permiten al diseñador de un trabajo VIPP® crear elementos interactivos cuando el trabajo VIPP® se representa en un documento PDF:

Hay tres categorías de elementos interactivos:

- **Marcadores**
- **Enlaces**
- **Notas**

### MARCADORES

La lista de marcadores es una función opcional de un documento PDF que se muestra en la pantalla. La lista consta de una jerarquía con una estructura de árbol de elementos de texto. Al hacer clic en el texto de un marcador se abre la aplicación del visualizador para ir a su destino asociado, que es una parte específica del documento actual o de otro documento.

### ENLACES

Un enlace asocia un elemento en una página, como un fragmento de texto, un cuadro o una imagen, con un destino. Hacer clic en este elemento hace que la aplicación del visor vaya al destino.

### NOTAS

Una nota asocia un elemento en una página como un fragmento de texto, un cuadro o una imagen con una entrada de texto que no es parte del flujo de la página. Cuando se cierra la nota, el texto no es visible. Una nota cerrada puede ser invisible o visible como un icono o una marca. Al hacer clic en una nota visible o en un elemento asociado con una nota invisible, el texto se visualizará en una ventana emergente. Hay cuatro categorías de notas:

#### Icono

Una nota visible representada por un icono pequeño.

#### Sello

Una nota visible representada por un sello.

#### Texto libre

Una nota visible que muestra texto directamente en la página. Las notas de texto libre no tienen estado cerrado: en vez de mostrarse en una ventana emergente, el texto siempre se visualiza.

### **Marca**

La marca se puede visualizar como color de fondo bajo el elemento cerrado asociado si se especifica un color que no sea blanco o nulo; de lo contrario, la nota es invisible.

## Claves predeterminadas y palabras clave

Consulte Atributos, claves, tablas y listas estándar en la *Guía del usuario de FreeFlow VI Compose* para obtener una vista general de las claves predefinidas y de los atributos usados en la programación de VIPP®. Aquí se tratan estos temas adicionales:

- Agregación de GEPkeys
- Mejora de la diferenciación de sombreado mediante DDG
- VIPP® Colorkeys
- Palabras clave reservadas

### AGREGACIÓN DE GEPKEYS

La manera más efectiva de crear una nueva Colorkey o GEPkey es imprimir `/usr/xgf/demo/palrgb.ps` mediante el color de realce correcto.

Para realizar esta tarea desde la página producida, encuentre los colores que desee y apunte los tres valores para especificar cada color. Estos valores de color se encuentran en la parte superior del cuadro. Edite el archivo `/usr/xgf/src/xgf.gep` y agregue una Colorkey que especifique estos valores.

A continuación, agregue todos las GEPkeys que desea utilizar con esta Colorkey.

### MEJORA DE LA DIFERENCIACIÓN DE SOMBREADO MEDIANTE DDG

El cambio del conjunto de la Colorkey mejora la diferenciación del sombreado en impresoras en blanco y negro si se usan los comandos **DDG DRAWPIE, DRAWBAR** o **DRAWCRV**. El valor predeterminado de Colorkey está definido por el parámetro `/ColorTable`. Para cambiar las Colorkeys predeterminadas, use el comando **SETPARAMS** para especificar otro conjunto de Colorkeys. Se recomienda la mezcla de la secuencia de niveles de gris de claro a oscuro.

#### Ejemplos

Este ejemplo establece siete niveles de gris para su uso con DDG.

```
[/ColorTable [XLIGHT MEDIUM XDARK LIGHT DARK LMEDIUM DMEDIUM] ] SETPARAMS
```

Para incrementar los colores, agregue Colorkeys a `xgf.gep`. El número máximo de colores que se pueden definir en `/ColorTable` es 65 536.

### PALABRAS CLAVE RESERVADAS

Si una palabra clave reservada se usa como un nombre del campo de la base de datos, pueden producirse errores impredecibles y difíciles de aislar en los trabajos VIPP®. Para evitar este problema, VI Compose realiza una comprobación de los nombres de la base de datos, y produce un mensaje de error si se detecta un conflicto. La línea adicional de código comprueba los nombres de archivo de la base de datos que son conocidos en el contexto actual de PostScript. Entre las palabras clave reservadas figuran:

- Operadores de PostScript y palabras clave reservadas
- Comandos, funciones, variables, etc. de VIPP®

- Nombres internos de VIPP® (empezando por XGF, VIP, ^, y ")

El mensaje de error (VIPP\_ambiguous\_name en \_name) también puede producirse en un trabajo XML si el código usa un nombre de variable XML ambiguo que no ha sido cualificado correctamente por VI Compose para identificar a qué parte del árbol de XML se refiere.

Para evitar el uso de palabras clave reservadas a la asignación de nombres a los campos de la base de datos, utilice un carácter inicial en mayúsculas en el nombre del campo de la base de datos. Por ejemplo, Fieldname.

## VIPP® COLORKEYS

VI Compose usa una tabla integrada de definición de colores en el archivo `/usr/xgf/src/xgf.gep`, que contiene las definiciones de Colorkey.

Una definición de Colorkey es un conjunto de valores de color como RGB, CMYK, patrón o gradiente, vinculados a un nombre de color. Al seleccionar el nombre de una Colorkey, realmente se selecciona el valor CMYK o RGB para esa clave.

Los colores personalizados se pueden agregar a esta tabla con la sintaxis descrita en el comando **SETCOL**.

Cuando se agrega a la tabla xgf.gep, la definición puede estar disponible en todas las aplicaciones y de forma permanente. Si se agrega a DBM, JDT o XJT, la definición del color es temporal y está disponible solo para ese trabajo.

Una Colorkey puede ser referenciada en varios comandos de VIPP® **SETTXC**, **SETGEP**, **SETBAT**, **SETFRAME**, **SETZEBRA**. En estos comandos, Colorkey puede utilizarse tal cual, o precedida por un carácter de barra diagonal.

Para obtener más información acerca de las fuentes y el color, consulte **Fuentes y colores**.

## Ejemplos

En una definición de color RGB, el color RGB se define como un porcentaje de rojo, verde y azul. La definición del color más abajo define un color RGB (solo tres valores especificados en la matriz). Los tres valores representan 100 % rojo, 0 % verde y 0 % azul. La etiqueta RED se ha asignado a esta definición

```
/RED [1 0 0] SETCOL
```

Esta definición de color define un color RGB. Los tres valores representan 100 % rojo, 90 % verde y 90 % azul. La etiqueta XLRED se ha asignado a esta definición.

```
/XLRED [1 .9 .9] SETCOL
```

En una definición de color CMYK, el color CMYK se define como un porcentaje de cian, magenta, amarillo y negro. La definición del color define un color CMYK (cuatro valores especificados en la matriz). Los valores representan 0 % cian, 100 % magenta, 90 % amarillo y 0 % negro. La etiqueta CMYK\_RED se ha asignado a esta definición.

```
/CMYK_RED [0 1 .9 0] SETCOL
```

Esto es un ejemplo de uso de Colorkey en un comando VIPP®. En el ejemplo, la instrucción configura el color de todo el texto impreso posteriormente según el valor de Colorkey llamado CMYK\_RED, como se define en el ejemplo anterior.

```
CMYK_RED SETTXC
```

## Imprimir procesamiento de archivos

Los comandos de procesamiento de archivos de impresión son:

BIDIRECCIONAL	ENDARBM	PPCOUNT	STARTLM
BACKSP_off	ENDARBT	PROCESSDJE	STARTXML
BACKSPF_on	ENDBOOKLET	QSTRIP_on	TRÍO
BACKSPP_on	ENDJOB	SETBIDI	XGFEND
BEGINARBM	ENDXPD	SETBUFSIZE	XMLATL
BEGINARBT	ETA	SETDBSEP	XMLATN
BEGINXPD	ETS	SETDLFILE	XMLATV
BOOKLETRANGE	EXIT	SETJDT	XMLDTH
BSTRIP_off	FRCOUNT	SETLMFILE	XMLPAR
BTA	FRLEFT	SETNMP	XMLPATH
BTS	INDEXPIF	SETPARAMS	XMLSYN
DJDEBEGIN	LMSKIP	SOF_off	XMLTAG
DJDECMD	OVERPRINT_on	STARTBOOKLET	XMLVAL
DJDEPAR	PAGERANGE	STARTDBM	ZSORT

Estas funciones de comandos de procesamiento de archivos de impresión se abordan en:

- [Compatibilidad de folletos](#)
- [Soporte de materiales de impresión](#)
- [ZSORT y agrupación de registros](#)

### COMPATIBILIDAD DE FOLLETOS

El modo de folletos funciona a través de los comandos **STARTBOOKLET** y **ENDBOOKLET**. Inicialice el modo de folleto con esta secuencia **SETPARAMS**:

```
[ /PagesPerBooklet integer1
/BookletMismatch integer2
] SETPARAMS
```

**Donde:**

<b>integer1</b>	es el número de páginas que debe contener un folleto.
<b>integer2</b>	se usa para controlar la acción en <b>ENDBOOKLET</b> si el número de páginas no coincide: <ul style="list-style-type: none"><li><b>0</b> omite la discordancia y continúa</li><li><b>1</b> agrega páginas en blanco hasta que el número de páginas coincida, y continúa</li><li><b>2</b> cancela el trabajo con un mensaje de error de VI Compose (a través del mecanismo de error de PostScript)</li></ul>

Esta secuencia se codifica al principio de la DBF o en el JDT.

**BOOKLETRANGE** funciona en combinación con **STARTBOOKLET** y **ENDBOOKLET** para seleccionar una serie de folletos para imprimir.



Nota: No todas las DFE (controladores de la impresora) admiten los comandos **STARTBOOKLET** y **ENDBOOKLET**. Son compatibles actualmente en EFI y el servidor de impresión de FreeFlow. Sin embargo, las funciones de acabado e imposición asociadas con un folleto delimitado por estos comandos dependen del DFE utilizado. Por ejemplo, actualmente, el servidor de impresión FreeFlow puede aplicar un acabado de subconjunto como el uso de grapas, de encuadernación, etc. en combinación con un comando **SETFINISHING** agregado antes de **STARTBOOKLET**.

Compruebe el software de la impresora y su soporte antes de utilizar estos comandos.

## SOPORTE DE MATERIALES DE IMPRESIÓN

En el lenguaje VIPP®, la selección de los materiales de impresión está disponible a través de los comandos **SETMEDIA** y **SETMEDIAT**. Estos comandos asumen como valor prefijado el operador de PostScript, **setpagedevice**, para realizar la selección de los materiales de impresión apropiados. Una solución está disponible para los dispositivos en los que no se ha implementado **setpagedevice**. El parámetro **/MediaSubst** se puede configurar en estos dispositivos para sustituir una secuencia de PostScript de la selección prefijada.

### Ejemplos

Esto es un ejemplo de uso y sintaxis de la sustitución de materiales de impresión:

```
[/MediaSubst<<
(normal) { 1 definebandeja }
(con membrete) { 2 definebandeja }
>>] SETPARAMS
```

Si se implementa **setpapertray** en el dispositivo, la secuencia de VIPP®, (Letterhead) **SETMEDIA** puede hacer que el papel se alimente desde la bandeja 2.

Coloque la definición **/MediaSubst** en el archivo **/usr/xgf/src/xgf.def** en el controlador del dispositivo para que se puede modificar con facilidad al instalar nuevos materiales de impresión en el dispositivo.



## ZSORT Y AGRUPACIÓN DE REGISTROS

**ZSORT** sustituye la agrupación de registros como una manera de configurar la base de datos de trabajos de varias en 1 y a 2 caras.

La intención inicial de la agrupación de registros era dividir registros de gran tamaño en varios subregistros para omitir las limitaciones de colas que no aceptan registros largos. Sin embargo, esta funcionalidad se usa habitualmente en trabajos de base de datos, varias en 1 o a 2 caras para leer N registros a la vez, según la opción varias en 1 y para imprimir datos variables de cada registro en la cara frontal y posterior de páginas de trabajos de varias en 1.

Esta compleja configuración es obsoleta y puede sustituirse fácilmente mediante una instrucción **ZSORT** con `stacksize=1` antes de **STARTDBM** o en el JDT asociado. A continuación se elimina la agrupación de registros manteniendo solo un conjunto de nombres de campo y se simplifica el DBM manteniendo solo las líneas de código para un registro.

El parámetro `/MUPduplex` está relacionado con las colocaciones de varios en 1 de las páginas lógicas en la parte posterior. Si `/MUPduplex=1`, 0 es el valor predefinido para la compatibilidad con versiones anteriores, pero **ZSORT** lo configura automáticamente a 1, las posiciones de la página lógica en la cara posterior se calculan para que estén físicamente de cara a sus contrapartidas en su cara frontal. Por tanto, no deberá preocuparse del cálculo de distintas colocaciones en la cara frontal y la posterior, incluso si se producen giros inusuales. Las páginas lógicas encajan totalmente en la página física ya sea través de la especificación explícita de los tamaños de la página lógica o mediante la reducción de su escala, por lo que si `Multi-Up scale=1`, use la sintaxis **SETMULTIUP** que especifica el tamaño de cada página lógica para evitar la colocación incorrecta de la cara posterior.

Por ejemplo, al definir una página de tamaño carta 4 en 1 en una página de tamaño carta, 2550 × 3300, asumiendo unidades en puntos, el tamaño de la página lógica es 1275 × 1650, por lo que la instrucción de varias en 1 es:

```
[ [ 0 1275] [ 0 1650] 0 1 1 [1275 1275] [ 0 1650] 0 1 1 [ 0 1275 ] [ 1650 1650] 0 1 1 ]
```

## Control de carro de la impresora

Los comandos PCC son:

- **BEGINPCC**
- **SETSKIP**
- **ENDPCC**
- **SETVFU**
- **SETPCC**

## Control de recursos

Los comandos de control de recursos son:

- `CACHE`
- `EXIST`
- `FBIND`
- `MAKEVMFILE`
- `MAKEVMFORM`
- `PRECACHE`
- `SETEPATH`
- `SETFPATH`
- `SETIPATH`
- `SETJPATH`
- `SETMPATH`
- `SETPAT`
- `SETPAGEDEF`
- `SETPPAT`
- `SETPPATH`
- `SETPROJECT`
- `STOREVAR`
- `SUBSTFONT`
- `XGFRESDEF`

## Elementos de RPE

Los comandos de entrada de procesamiento de registros son:

BEGINRPE	FROMLINE	RPEDEF	SETRPE
CURLINE	INDEXRPE	RPEKEY	SETRPEPREFIX
ENDIFALL	NEWGROUP	RPEPOS	
ENDRPE	NEWPOS	SETRCD	

Entre los elementos RPE tratados aquí se incluyen:

- Ignorar el número de líneas al principio de cada archivo
- Sobreimpresión de una línea por lograr el efecto de fuente negrita
- Información de comando RPE
- Uso del color de realce para imprimir números negativos

### IGNORAR EL NÚMERO DE LÍNEAS AL PRINCIPIO DE CADA ARCHIVO

Si el contenido de las líneas es único en el archivo **SETLMFILE**, una condición **SETRCD** se puede definir y usar en una entrada **FROMLINE** con **recpos=0** y **length=0**. Además, use la sintaxis ampliada de **STARTLM**, que permite usar un JDT en un determinado número de páginas y, a continuación, pasar a usar otro JDT para el resto del archivo.

Cuando ese es el caso, defina **SETGRID** como el número de líneas que desea omitir en el primer JDT y use el comando **FROMLINE** que no selecciona datos.

### SOBREIMPRESIÓN DE UNA LÍNEA POR LOGRAR EL EFECTO DE FUENTE NEGRITA

Algunas aplicaciones antiguas diseñadas para impresoras de impacto utilizan la técnica de sobreimpresión para obtener un efecto de texto en negrita. Para obtener el mismo efecto con VIPP®, use el comando **SETRCD** para probar el código de sobreimpresión (normalmente, un carácter PCC + en la posición 0 del registro) y aplicar otra posición de impresión, 104 y 104 en vez de 100 y 100. Esto imprime las dos líneas, una encima de la otra, con un ligero desplazamiento y proporciona el efecto negrita.

Esto es un ejemplo:

```

%!
XGF
/ANSI SETPCC
/F1 /NCR 12 INDEXFONT
/IF_OVP 0 1 /eq (+) SETRCD

10 BEGINRPE
1 FROMLINE
  /IF_OVP [ 0 0 104 nu]] 104 50 0 80 /F1 BLACK ]
  /ELSE   [ 0 0 100 nu]] 100 50 0 80 /F1 BLACK ]
ENDRPE

() STARTLM
Aaaaaaaaaaaaaa
Bbbbbbbbbbbbbb
+Bbbbbbbbbbbbbb}
Ccccccccccccc
%%EOF

```

## INFORMACIÓN DE COMANDO RPE

La entrada de procesamiento de registros (RPE) es una opción de presentación usada en aplicaciones de modo de línea que permite a cada registro dividirse en campos que se pueden imprimir tantas veces como se desee y en cualquier ubicación de la página. Esta opción también se describe en Flujos de datos de VIPP® en la *Guía del usuario de FreeFlow VI Compose*.

Las RPE puede contener atributos de presentación específicos, como la posición en la página, el espaciado de líneas, las fuentes, el color, la alineación y el giro. Los atributos RPE se iniciaron mediante una secuencia **STARTLM**.

Los campos RPE se definen en sentido horizontal utilizando uno de los siguientes:

- posición de caracteres y longitud de campo
- número de campos

Los campos RPE se definen en sentido vertical utilizando uno de los siguientes:

- número de línea **FROMLINE**
- prefijo **SETRPEPREFIX**, **RPEKEY**

Las definiciones de campo, la posición en la página y los atributos de presentación de un grupo determinado de líneas o un prefijo específico se codifican en una definición RPE. Para obtener más información, consulte **RPEDEF** y **SETRPE**.

La delimitación de páginas se controla mediante la configuración de un número máximo de LPP o a través de un delimitador de página explícito como una alimentación de formulario (FF) o una omisión para canalizar una si se usa PCC. Para obtener más información, consulte **SETGRID**, **SETPCC** y **SETPBRK**.

Los archivos de la impresora en línea, producidos con un registro de datos por página (que se conocen como registros sin formato), también se pueden procesar, una línea por página, con los comandos RPE.

## Ampliación de los comandos FROMLINE y RPEKEY

Los comandos **FROMLINE** y **RPEKEY** se pueden ampliar mediante:

- Procesamiento condicional
- Texto fijo
- Procedimiento de alineación

### Procesamiento condicional

Las condiciones se pueden anidar en cualquier nivel mediante los subcomandos de **/ENDIF** y **/ENDIFALL** RPE. Para obtener más información, consulte las definiciones de las condiciones **SETRCDSETPCD**.

### Ejemplos

En este ejemplo, **/ENDIFALL** permite cerrar todas las instrucciones **/IF** pendientes en un comando, en vez de tener que codificar todas las instrucciones **/ENDIF** coincidentes.

```

1 FROMLINE
  /IF_CND1
    [ .... rpe entry 1 .... ]
    [ .... rpe entry 2 .... ]
  /IF_CND2
    [ .... rpe entry 3 .... ]
  /ELSE
    [ .... rpe entry 4 .... ]
  /ENDIF
  /ELSE
    [ .... rpe entry 5 .... ]
    [ .... rpe entry 6 .... ]
  /IF_CND3
    [ .... rpe entry 7 .... ]
  /ENDIF
/ENDIF

10 FROMLINE
  /IF_CND4
    [ .... rpe entry 11 .... ]
    [ .... rpe entry 12 .... ]
  /ELSE /IF_CND5
    [ .... rpe entry 13 .... ]
  /ELSE /IF_CND6
    [ .... rpe entry 14 .... ]
  /ELSE /IF_CND7
    [ .... rpe entry 15 .... ]
    [ .... rpe entry 16 .... ]
  /ELSE
    [ .... rpe entry 17 .... ]
  /ENDIFALL

```

La sintaxis anterior todavía es válida para **/IF** y **/ENDIF** coinciden con una entrada RPE respectivamente. La nueva sintaxis se aplica solo si **/ENDIF** o **/ENDIFALL** se encuentran en una definición de RPE.

Las nuevas sintaxis y las sintaxis antiguas son exclusivas en una determinada definición de RPE. Sin embargo, las definiciones de RPE con ambas sintaxis se pueden mezclar en bibliotecas de RPE compuestas de diversos

comandos **RPEKEY** o **FROMLINE** situados entre **BEGINRPE** y **ENDRPE**.

```

2 BEGINRPE           % Begin RPE library

1 FROMLINE           % RPE using old syntax
/CND1
[ .... rpe entry .... ]
/ELSE
[ .... rpe entry ... ]

2 FROMLINE           % RPE using new syntax
/IF_CND2
[ .... rpe entry .... ]
[ .... rpe entry .... ]
/ELSE /IF_CND3
[ .... rpe entry .... ]
/ELSE /IF_CND4
[ .... rpe entry .... ]
/ENDIFALL

ENDRPE              % End RPE library

```

### Texto fijo

Los parámetros **recpos** y **length** de las definiciones de RPE para los comandos **RPEKEY** o **FROMLINE** se pueden sustituir por una de estas secuencias:

- 0 (texto fijo)
- 0 /VARname

Use esta función para imprimir variables o cadenas de texto fijo definidas mediante **SETVAR** o **GETFIELD**, que no figuran en el flujo de datos.

### Ejemplos

Este ejemplo imprime Descripción: en Xpos 70 seguido de los caracteres 0 a 29 en el registro, en 500 Xpos e Ypos 910.

```

/LFA0 RPEKEY
[ 0 0 70 null 910 50 0 (Description:) /F1 BLACK]
[ 0 0 500 null 910 50 0 30 /F1 BLACK]

```

Este es un ejemplo de cómo utilizar una variable.

```

/VAR.LABEL1 (Description:) SETVAR
....
/LFA0 RPEKEY
[ 0 0 70 null 910 50 0 /VAR.LABEL1 /F1 BLACK]

```

## PROCEDIMIENTO DE ALINEACIÓN

El parámetro de alineación de una determinada entrada RPE puede ser sustituido por un procedimiento de modo nativo VIPP®. Este procedimiento llama a un comando de modo nativo mediante texto fijo o datos imprimibles como operandos.

Cuando se realiza la llamada al procedimiento, el operando se puede proporcionar automáticamente por el campo extraído especificado por los parámetros `recpos` y `length`, o a través de la función de texto fijo descrita más arriba. La posición de impresión se define según la posición de impresión RPE actual.

El procedimiento solo puede suministrar los operandos adicionales requeridos por el comando VIPP®. Estos son ejemplos de la sintaxis del procedimiento de alineación:

- { rotate GEPkey align SHX }
- { colwidth align SHP and SHp }
- { colwidth align SHMF }
- { scale rotation ICALL }
- { scale SCALL }
- { EAN13 SHX }
- { CODE39 SHX }, SHx representa SH, SHr, SHc o SHj.

### Ejemplos

```
{ DATAMATRIX }
{ /VARqrc /SWP SETVAR [ /AC VARqrc ] QRCODE }
```

Es un ejemplo que imprime una imagen TIFF llamada `signa.tif` en Xpos 70 e Ypos 910 cada vez que se selecciona /LFA0 RPEKEY. Además, se imprime del carácter 0 al carácter 29 del registro a Xpos 500 e Ypos 910.

```
/LFA0 RPEKEY
[ {1 0 ICALL} 0 70 null 910 50 0 (signa.tif) /F1 BLACK]
[ 0 0 500 null 910 50 0 30 /F1 BLACK]
```

En este ejemplo se imprime la cadena David Kirk subrayada en la posición 100 100. La N desactiva la UNDL BATkey.

```
/U /UNDL INDEXBAT
/N null INDEXBAT
1 BEGINRPE
1 FROMLINE
[ {U SH N} 0 100 0 100 30 0 (David Kirk) /F1 BLACK ]
```

## USO DEL COLOR DE REALCE PARA IMPRIMIR NÚMEROS NEGATIVOS

Los comandos `SETRCD` y `SETPCD` tienen un operador de prueba /HOLD para ubicar la cadena de comparación en las porciones del registro de datos.

Este ejemplo busca el signo menos en el campo y utiliza el color de realce actual para imprimirlo:



```
/IF_NEG 50 10 /HOLD (-) SETRCD  
.....  
x FROMLINE  
/IF_NEG [0 0 100 100 100 100 50 10 /F1 HCOLOR ]  
/ELSE [0 0 100 100 100 100 50 10 /F1 BLACK ]  
.....
```



Nota: El ejemplo asume que el número tiene 10 bytes de largo en la posición 50.

## Funciones de transformación

Los comandos de las funciones de transformación son:

2OF5	CODE39	GETINTV	REEMPLAZAR
64TO256	CS	HMS	UPCA
AZTEC	EAN128	NOHYPHEN	UPC-A
BSTRIP	EAN13/EAN8	POSTJPN	UPC-E
CASELOW	F2S	POSTNET	VSUB
CASETI	FORMAT	QSTRIP	VSUB2
CASEUP			VSUB3
CODE128			



