PCL XL Feature Reference Protocol Class 3.0 Supplement

Revision: 0.90	
Revision Date: July 8, 2002	
Original Author:	
Word for Windows File: xl_ref30r089.doc	

Document Revision History

Rev	Revision Description	Date	Author
0.1	First Draft	29Nov2000	
0.2	Added in new XL operators & language switching	30Nov2000	
0.3	Incorporated suggested changes	07Dec2000	
0.31	Incorporated Color PQ changes	12Apr2001	
0.32	Incorporated suggested PQ changes	16Apr2001	
0.4	Updated XL PassThrough operator	29May2001	
0.5	Added new feature to SetFont for PCL5 fonts	05June2001	
0.6	Corrected some PQ default settings	26June2001	
0.7	Fixed some PQ, removed canceled features	03July2001	
0.8	Added PQ Defaults to DefaultGS table	12July2001	
0.81	Fixed PassThrough definition	19July2001	
0.82	Corrected a label	24July2001	
0.83	Updates to PassThrough technical description	03Aug2001	
0.84	PassThrough Pixel Placement Grid Centered	07Aug2001	
0.85	PCL Font Selection Cautions Added	28Aug2001	
0.89	Updated passthrough and dither matrix info	08July2002	

***** NOTICE *****

HEWLETT-PACKARD COMPANY MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THE SOFTWARE OR TECHNICAL INFORMATION. HEWLETT-PACKARD COMPANY DOES NOT WARRANT, GUARANTEE OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE SOFTWARE OR TECHNICAL INFORMATION IN TERMS OF ITS CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS, OR OTHERWISE. YOU ASSUME THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE OR TECHNICAL INFORMATION. The exclusion of implied warranties is not permitted by some jurisdictions. The above exclusion may not apply to you.

IN NO EVENT WILL HEWLETT-PACKARD COMPANY BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION AND THE LIKE) ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE OR TECHNICAL INFORMATION EVEN IF HEWLETT-PACKARD HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Because some jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitations may not apply to you. Hewlett-Packard liability to you for actual damages from any cause whatsoever, and regardless of the form of the action (whether in contract, tort including negligence, product liability or otherwise), will be limited to US $50.

Table Of Contents

1.0 INTRODUCTION 4

2.0 PCL XL OPERATORS, ATTRIBUTES AND STREAMS 5

2.1 GRAPHICS STATE SETTINGS 5
Default Graphics State 5
Operator: SetHalftoneMethod 7
Operator: SetAdaptiveHalftoning 9
Operator: SetColorTrapping 10
Operator: SetNeutralAxis 11
2.2 PASSTHROUGH SUPPORT 12
Operator: PassThrough 15
2.3 PCL FONT ACCESS 16
Operator: SetFont 17

APPENDICES 19

APPENDIX B. BINARY STREAM TAG VALUES 19
APPENDIX E. ATTRIBUTE ID NUMBER TO ATTRIBUTE NAME TABLE 23
APPENDIX F. ATTRIBUTE NAME TO DATA TYPES TABLE 27
APPENDIX G. ATTRIBUTE VALUE ENUMERATIONS TABLE 31
APPENDIX K. PCL XL ERROR AND WARNING CODES 35

1.0 Introduction

This document is designed to provide a supplement to the ***PCL-XL Protocol 2.1 Reference Manual***. Information in this supplement is on operators that have changes for protocol class 3.0 as well as information about new operators. Information that is different from the 2.1 reference manual is printed in red for clarity.

2.0 PCL XL Operators, Attributes and Streams

2.1 Graphics State Settings

Graphics state attributes are set by the user to obtain a specific result during painting. For example, the graphics state contains the current paint source (color) associated with the brush. When an object is painted, the device retrieves the brush's current paint source from the graphics state. If the brush is associated with a valid paint source PCL XL fills the object with the corresponding color or pattern specified.

DEFAULT GRAPHICS STATE

All graphics state attributes are set by commands preceded by "**Set**..." with the exception of the current path which is set by path operators (i.e. Begin/EndPath, Arc, Line, etc.). Graphics state attributes and defaults at each **BeginPage**, or when the **SetDefaultGS** operator is executed are listed in the table below:

GS Attribute	Description	Default
AdaptiveHalftone	Text, Vector and Raster Adaptive Halftoning	Enabled
BrushSource	Paint source currently associated with the brush	RGB black
ColorTrapping		Normal
ColorTreatment	The color treatment to be applied to the current RGB colors.	eScreenMatch
CharAngle	The angle at which to draw characters (additive to page CTM)	0
CharScale	The scaling factor for characters (additive to page CTM)	x=1, y=1
CharShear	The shearing factor for characters (additive to page CTM)	x=0, y=0
ClipMode	The mode determining even-odd or non-zero winding construction of the clip path	eNonZeroWinding
ColorSpace	Current color space	RGB
CurrentClipPath	The region defining the current clip path	imagable area of the page
CurrentFont	The font that will be used for painting characters	no font defined
CurrentPath	The region defining the current path	no path defined
DefaultCTM	The default page coordinate transformation matrix	Device dependent
DitherAnchor	The x,y point in which the dither is anchored	0,0
DitherMatrixID	A user definable identifier used to select a specific dither matrix	none
HalftoneMethod	The method used for halftone operations Text Objects	HighLPI
HalftoneMethod	Vector Objects	MediumLPI
HalftoneMethod	Raster Objects	LowLPI
FillMode	The mode in which closed paths should be filled	eNonZeroWinding
LineCap	The shape to draw on the end of lines (open subpaths)	eButtCap
LineDash	The dash style to use when stroking lines with a pen	solid line
LineJoin	The shape to draw where two lines meet at an angle	eMiterJoin
MiterLimit	The length limit on miter join shapes	10.0
NeutralAxis	Text Objects	Toner Black
NeutralAxis	Vector Objects	Toner Black
NeutralAxis	Raster Objects	Process Black

PageCTM	Current page coordinate transformation matrix	session defaults at the current page orientation
PaintTxMode	The current transparency mode for patterns	eOpaque
PalletteID	A user definable identifier used to select a specific palettes	none
PatternAnchor	The x,y point in which the brush and pen patterning is started	0,0
PenSource	Paint source currently associated with the pen	RGB black
PenWidth	The current width (in user units) for pen stroking operations	1 user unit
ROP	The current raster operation in effect	ROP3=252
SourceTxMode	The current transparency mode for source objects	eOpaque

OPERATOR: SETHALFTONEMETHOD

Purpose

Set the halftone method by specifying the current dither matrix to be used in the halftoning process for text, scanned (bitmap) images, raster patterns, and colors by reading the matrix data from the currently open data source. This is a device resolution-dependent operator.

Precondition

Immediate execution of this operator occurs in a legal PCL XL operator sequence.

The length of each dither matrix row must be a multiple of four bytes. If row data is not a multiple of four bytes, row data must be padded with the appropriate number of bytes.

The depth of the dither matrix data must be eight bits.

Attribute List Specification

multiAttributeList ::= {DitherOrigin}opt &
{{ DeviceMatrix | {TextObjects | VectorObjects | RasterObjects} } |
{ DitherMatrixDataType & DitherMatrixSize & DitherMatrixDepth } }

Attribute ID	Description and {value}
DitherOrigin	The origin for the dithering process in user units. If this attribute is missing the origin is x=0, y=0 in user units. { xyValue }
DeviceMatrix	For XL 2.1 and below: An enumeration to select internal device dither matrices. This attribute has no effect on Color devices. { eDeviceBest }
TextObjects	An enumeration to select internal device dither matrices for text object types. { eHighLPI (default) | eMediumLPI | eLowLPI }
VectorObjects	An enumeration to select internal device dither matrices for vector object types. { eHighLPI | eMediumLPI (default) | eLowLPI }
RasterObjects	An enumeration to select internal device dither matrices for raster (bitmap) object types. { eHighLPI | eMediumLPI | eLowLPI (default) }
DitherMatrixDataType	The data type of data to be read from the data source for the new dither matrix setting. Each data element read from the data source represents one dither matrix cell value. The first byte of the data is the left-most cell of a row (no byte swapping is performed on dither matrix data). { eUByte }

DitherMatrixSize	The width (x value) and height (y value) of the dither matrix in device pixels, the values of which may each be one or greater. If the actual width is not a multiple of four bytes, the data for each row must be padded to four bytes. { xyValue }
DitherMatrixDepth	The depth of each dither matrix entry in bits. { e8Bit }

Postcondition

The current halftone method has been set in the graphics state.

Example

Setting the printer's default dither matrix

```
ubyte eDeviceBest DeviceMatrix
SetHalftoneMethod
```

Setting the different values for objects.

```
ubyte eMediumLPI TextObjects
ubyte eHighLPI VectorObjects
ubyte eLowLPI RasterObjects
SetHalftoneMethod
```

Setting a downloaded dither pattern

```
uint16_xy 16 2 DitherMatrixSize
ubyte 0 DitherMatrixDataType
ubyte 2 DitherMatrixDepth
SetHalftoneMethod
```

This operator must be immediatly followed by dither matrix data.

```
dataLength 32
hex_raw* [
80 00 80 00 80 00 80 00 80 00 80 00 80 00 80 00
00 80 00 80 00 80 00 80 00 80 00 80 00 80 00 80
]
```

Comments

Note. Some PCL XL implementations may choose to read and ignore the dither matrix data without causing an error condition. In this case, the user-defined dither matrix will have no effect.

Note: For XL 3.0 and above, the DeviceMatrix attribute will set TextObjects, VectorObjects and RasterObjects with whatever value is defined.

See **Appendix F** and **Appendix G** for valid attribute data types and ranges.

See **Appendix K** for related error codes.

OPERATOR: SETADAPTIVEHALFTONING

Protocol Class

3.0

Purpose

TBD

Precondition

TBD

Attribute List Specification

multiAttributeList ::= { AllObjectTypes} |
{ TextObjects | VectorObjects | RasterObjects}

Attribute ID	Description and {value}
TextObjects	Enable or Disable Adaptive Halftoning for Text objects. { eDisable | eEnable (default) }
VectorObjects	Enable or Disable Adaptive Halftoning for Vector objects. { eDisable | eEnable (default) }
RasterObjects	Enable or Disable Adaptive Halftoning for Raster objects. { eDisable | eEnable (default) }
AllObjectTypes	Enable or Disable Adaptive Halftoning for Text, Vector and raster objects. { eDisable | eEnable (default) }

Postcondition

The color adaptive halftoning is defined for the current object.

Example

```
Ubyte eDisable TextObjects
SetAdaptiveHalftoning
```

Comments

See **Appendix F** and **Appendix G** for valid attribute data types and ranges.

See **Appendix K** for related error codes.

OPERATOR: SETCOLORTRAPPING

Protocol Class

3.0

Purpose

TBD

Precondition

TBD

Attribute List Specification

multiAttributeList ::= { AllObjectTypes}

Attribute ID	Description and {value}
AllObjectTypes	Disable or set the type of color trapping to be used for Text, Vector and raster objects. { eDisable | eMax | eNormal (default) | eLight }}

Postcondition

The color treatment for the current color space has been set.

Example

```
Ubyte eLight AllObjectTypes
SetColorTrapping
```

Comments

See **Appendix F** and **Appendix G** for valid attribute data types and ranges.

See **Appendix K** for related error codes.

OPERATOR: SETNEUTRALAXIS

Protocol Class

3.0

Purpose

TBD

Precondition

TBD

Attribute List Specification

multiAttributeList ::= { AllObjectTypes} |
{ TextObjects | VectorObjects | RasterObjects}

Attribute ID	Description and {value}
TextObjects	Enumerate the neutral axis setting to be used for Text objects. { eTonerBlack (default) | eProcessBlack }
VectorObjects	Enumerate the neutral axis setting to be used for Vector objects. { eTonerBlack (default) | eProcessBlack }
RasterObjects	Enumerate the neutral axis setting to be used for Raster objects. { eTonerBlack | eProcessBlack (default) }
AllObjectTypes	Enumerate the neutral axis setting to be used for Text, Vector and Raster objects. { eTonerBlack | eProcessBlack }

Postcondition

The color treatment for the current color space has been set.

Example

```
ubyte eProcessBlack AllObjectTypes
SetNeutralAxis
```

Comments

See **Appendix F** and **Appendix G** for valid attribute data types and ranges.

See **Appendix K** for related error codes.

2.2 Passthrough Support

XL Protocol Class 3.0 will support the concept of Passthrough data from drivers. In the past, when an application needed to invoke an operation that the printer supported and the driver did not, it asked the driver to "Pass Through" a block of data to the device. The driver did not parse or otherwise examine the Passthrough data.

In most cases, the passed through data consisted of PCL5 based on the application's assumption that Windows non-Postscript printers only understood PCL5 when the application was created.

In order for XL to support a non-Postscript Passthrough request XL 3.0 has been redesigned to allow it access to PCL5c and PCL5c has had some modifications made to accept input from XL. In the case where a PassThrough arrives, control will be passed to PCL5c so it may execute the contents of the data block.

Passthrough data is limited to 64k "chunks" by the some versions of the Windows OS so consideration for "stitching" together passthrough data by XL will be detailed.

2.2.1 PassThrough Definitions

2.2.1.1 PCL Passthrough - Passthrough data will be considered PCL and will be passed to the PCL5c language for processing. Again, if there are no intermediate operators between the end of the first passthrough chunk and the next, the next block will automatically be considered the same language as the previous block.

2.2.1.2.1 Whole-Job PCL Passthrough - *If a block of PCL5c passthrough data is processed after XL BeginPage but before XL marks on that page, then PCL5c can reset the page state (Media Size, Type, Orientation, Duplex Mode, Source, Destination, etc.) when it needs to*. This will be refered to as the "Whole-Job" passthrough as an entire PCL5c multi page job can be processed when wrapped in an XL session. FormFeed and <esc>E reset commands will be fully honored.

2.2.1.2.2 Snippet PCL Passthrough - *If a block of PCL5c passthrough data is processed after XL has opened a page and marked on it, then PCL5c will not be allowed to alter the page state*. A "Snippet" is a block of PCL passthrough that is intent on marking on the page that a driver has opened. PCL Page control commands in the passthrough sugh as Media Size, Media Type, Orientation, Duplex Mode, etc. will be ignored. FormFeed will not eject the page, but will move the CAP back to PCL5c's page origin. <esc>E reset will reset PCL as far as possible without altering the page state, which will be kept compatible with the PCL-XL setttings. The <esc>E reset will not eject the current page or start a new one because this page is considered to be "owned" by XL.

It is recommended therefore that wherever possible PCL5 snippets avoid attempting to change the current page state, eject the current page or otherwise make changes to the printer's state that the Windows driver does not know about.

2.2.2 PCL Passthrough – XL will take the following steps before passing control to PCL5c.

XL will store away the current font selection and current font attributes.

XL will execute a “PushGS” to save the current graphics state.

XL will execute a “NewPath” to empty the current path.

XL will execute a “SetClipToPage” to undo any currently set clip path.

NOTE: An XL user defined dither matrix will affect the printing and appearance of PCL passthrough data.

XL will then pass control to PCL5c and allow it to process the passed through block of data. When the block has been consumed, if the next operator is not PassThrough, control will return to XL. Otherwise, PCL will be allowed to consume the next block of passed through data.

PCL will adopt the current XL page state (page size, orientation, duplexing, etc.). PCL will NOT adopt XL’s coordinate transformation matrix changes which were made by the SetOrigin, SetRotation, or SetScale operators. Instead, PCL will reset to the default coordinate transformation matrix. Other than the page settings, PCL resets to normal default values, with the following exceptions:

-Top and Bottom Margins are set to 1/6 inch instead of ½ inch. (A subsequent Escape E Reset will restore the PCL defaults of ½ inch.)
-The PCL Unit of measure is set to the PCL-XL UnitsPerMeasure value. (A subsequent Escape E reset will restore the PCL default of 300 dots per inch.)
-If XL has a valid cursor location, the PCL CAP will be set to the XL location. If the cursor location is not valid, then PCL CAP will be set to the top left corner of the text area. (A subsequent Escape E Reset will restore the PCL default of top left corner of the text area.)
-Pixel Placement is set to Grid-Centered Mode, instead of Grid-Intersection mode.

If this is the first Passthrough operator of the current page but a passthrough has previously occurred in the current session, then PCL will adopt the new page state, reset margins to 1/6 inch, and adopt XL’s CAP but will not reset any other values to default PCL settings.

Once PCL has returned control to XL, re-entry to PCL on the same page will cause PCL to adopt XL’s cursor location but otherwise continue on with the PCL state set by the previous Passthrough. (Contiguous passthrough operators do not return control to XL. Instead, control remains with PCL until an operator other than Passthrough is detected.)

If the operator immediately after a passthrough block is not another passthrough, then XL will gain control back from PCL5c and perform the following steps before continuing to execute XL operators.

XL will execute a “PopGS” to restore the graphics state, path, clip path and color space.

XL will execute a “SetFont” and any other font settings that will restore the current font selection to setting that the Windows driver expects.

OPERATOR: PASSTHROUGH

Purpose

Informs XL that the block of data following the operator is being passed by the driver for processing by the PCL subsystem.

PreCondition

Immediate execution of this operator occurs in a legal PCL XL operator sequence. XL is in the page open state, which means that both the BeginSession and BeginPage operators have successfully executed without corresponding EndPage and EndSession operators.

Attribute List Specification

NullAttributeList

The data block does need to be prepended with a dataLength or dataLengthByte value that indicates how much data is contained in the passthrough block.

Post Condition

The PassThrough data stream was executed as PCL.

Example

```
PassThrough

dataLengthByte 12
hex* [1b 2d 2d 31 32 33 34 35 72 0d 1B 43]
```

Comments

NOTE: An XL user defined dither matrix will affect the printing and appearance of any PCL passthrough data.

Passthrough data will be treated as PCL.

See **Appendix K** for related error codes.

2.3 PCL Font Access

There are times that an application will want to access a PCL5 font for use by XL.

Each font that the driver is aware of has two strings used for selection, the PCL string and the XL string. The PCL string consists of the PCL ESC sequences needed to select the font. The XL string consists of the 16 character name of the font. If there is a font that is normally not accessable to XL, then the XL string is NULL.

Current XL drivers ignore any font in the table that has a NULL for the XL selection string. Future XL drivers will be able to send the PCL string for font selection by using the new PCLSelectFont attribute. XL will send the PCL string to PCL for processing and use whatever font is selected via that string for printing.

When selecting an XL font, the driver must send Symbol Set and Character Size information along with the font name. PCL font select strings normally contain this information so when the PCLSelectFont attribute is used, the SymbolSet abd CharSize operators are not required.

Because XL and PCL are sharing the selected font, after XL calls PCL to set up the font, any subsequent PassThrough calls to PCL that contain text will also use that font unless the PassThrough data contains font selection operators.

OPERATOR: SETFONT

Purpose

Set the current font from which characters will be selected by character codes in text operators.

Precondition

Immediate execution of this operator occurs in a legal PCL XL operator sequence.

The font is already defined in the device.

Note: If the requested font does not exist, PCL XL will substitute a font from those existing in the device at the time of the request. A warning will be generated due to the font substitution that may optionally be reported to the user. The exact font substitution algorithm is device- and language-region dependent. See the PCL XL addendum for the target device for more details concerning the font substitution algorithms.

Attribute List Specification

multiAttributeList ::= { SymbolSet & CharSize & FontName } | PCLSelectFont

Attribute ID	Description and **(value)**
FontName	The name of the font to be selected (i.e. "Times New Roman Bold"). (ubyte_array)
CharSize	The size of the characters to be rendered for the font in user units. This value is ignored for characters downloaded as bitmaps. The size of bitmap characters are the size of their original download definition. (+number greater than zero)
SymbolSet	The identifier of the symbol set to use for the current font (See Appendix O for values.) (+integer)
PCLSelectFont	A string containing the PCL operators (ESC sequences) that will select a PCL font. (ubyte_array)

Postcondition

The current font has been set in the graphics state. If the font selected is a PCL font, subsequent PassThrough's that contain text will also print using that font.

Example

Selecting an XL standard font

```
ubyte_array (TimesNewRmn) FontName
real32 100 CharSize
uint16 629 SymbolSet
SetFont
```

Selecting the PCL font Line Printer

```
ubyte_array “\x1B(10U\x1Bs0p16.6h8.5v0s0b0T” PCLSelectFont
SetFont
```

Comments

NOTE: When using the PCLSelectFont attribute, the symbols set and character size are to be included in the PCL5 font selection string so the XL SymbolSet and CharSize attributes are not needed.

NOTE: When a PCL font is selected, that font is then shared between PCL and XL. If a PassThrough operator is executed after an XL SetFont operator (where a PCL font was selected) than any text in that passthrough will be printed in the font that XL requested.

See **Appendix F** and **Appendix G** for valid attribute data types and ranges.

See **Appendix K** for related error codes.

Appendices

Appendix B. Binary Stream Tag Values

The following table outlines the specific values assigned for attribute list tags, operator tags, and data type tags.

Tag Value	Tag Name	Tag Type	Description
0x00		White Space	Null
0x01 – 0x08			Not Used
0x09 – 0x0d		White Space	HT, LF, VT, FF, CR
0x0e – 0x1f			Not Used
0x20		White Space	Space
0x21 – 0x26			Not Used
0x27			Reserved for beginning of ASCII binding.
0x28			Binary binding – high byte first.
0x29			Binary binding – low byte first.
0x2a – 0x40			Not Used
0x41	BeginSession	Operator	
0x42	EndSession	Operator	
0x43	BeginPage	Operator	
0x44	EndPage	Operator	
0x45		Operator	Reserved for future use.
0x46		Operator	Reserved for future use.
0x47	Comment	Operator	
0x48	OpenDataSource	Operator	
0x49	CloseDataSource	Operator	
0x4a		Operator	Reserved for future use.
0x4b		Operator	Reserved for future use.
0x4c		Operator	Reserved for future use.
0x4d		Operator	Reserved for future use.
0x4e		Operator	Reserved for future use.
0x4f	BeginFontHeader	Operator	
0x50	ReadFontHeader	Operator	
0x51	EndFontHeader	Operator	
0x52	BeginChar	Operator	
0x53	ReadChar	Operator	
0x54	EndChar	Operator	
0x55	RemoveFont	Operator	
0x56	SetCharAttributes	Operator	
0x57	SetDefaultGS	Operator	

0x58	SetColorTreatment	Operator	
0x59		Operator	
0x5a		Operator	
0x5b	BeginStream	Operator	
0x5c	ReadStream	Operator	
0x5d	EndStream	Operator	
0x5e	ExecStream	Operator	
0x5f	RemoveStream	Operator	
0x60	PopGS	Operator	
0x61	PushGS	Operator	
0x62	SetClipReplace	Operator	
0x63	SetBrushSource	Operator	
0x64	SetCharAngle	Operator	
0x65	SetCharScale	Operator	
0x66	SetCharShear	Operator	
0x67	SetClipIntersect	Operator	
0x68	SetClipRectangle	Operator	
0x69	SetClipToPage	Operator	
0x6a	SetColorSpace	Operator	
0x6b	SetCursor	Operator	
0x6c	SetCursorRel	Operator	
0x6d	SetHalftoneMethod	Operator	
0x6e	SetFillMode	Operator	
0x6f	SetFont	Operator	
0x70	SetLineDash	Operator	
0x71	SetLineCap	Operator	
0x72	SetLineJoin	Operator	
0x73	SetMiterLimit	Operator	
0x74	SetPageDefaultCTM	Operator	
0x75	SetPageOrigin	Operator	
0x76	SetPageRotation	Operator	
0x77	SetPageScale	Operator	
0x78	SetPatternTxMode	Operator	
0x79	SetPenSource	Operator	
0x7a	SetPenWidth	Operator	
0x7b	SetROP	Operator	
0x7c	SetSourceTxMode	Operator	
0x7d	SetCharBoldValue	Operator	
0x7e	SetNeutralAxis	Operator	
0x7f	SetClipMode	Operator	
0x80	SetPathToClip	Operator	
0x81	SetCharSubMode	Operator	
0x82	BeginUserDefinedLineCaps	Operator	
0x83	EndUserDefinedLineCaps	Operator	
0x84	CloseSubPath	Operator	
0x85	NewPath	Operator	
0x86	PaintPath	Operator	
0x87	Reserved	Operator	
0x88	Reserved	Operator	
0x89	Reserved	Operator	
0x8a	Reserved	Operator	
0x8b		Operator	

0x8c		Operator	
0x8d		Operator	
0x8e		Operator	
0x8f		Operator	
0x90		Operator	
0x91	ArcPath	Operator	
0x92	SetColorTrapping	Operator	Reserved for future use.
0x93	BezierPath	Operator	
0x94	SetAdaptiveHalftoning	Operator	Reserved for future use.
0x95	BezierRelPath	Operator	
0x96	Chord	Operator	
0x97	ChordPath	Operator	
0x98	Ellipse	Operator	
0x99	EllipsePath	Operator	
0x9a		Operator	Reserved for future use.
0x9b	LinePath	Operator	
0x9c		Operator	Reserved for future use.
0x9d	LineRelPath	Operator	
0x9e	Pie	Operator	
0x9f	PiePath	Operator	
0xa0	Rectangle	Operator	
0xa1	RectanglePath	Operator	
0xa2	RoundRectangle	Operator	
0xa3	RoundRectanglePath	Operator	
0xa4		Operator	Reserved for future use.
0xa5		Operator	Reserved for future use.
0xa6		Operator	Reserved for future use.
0xa7		Operator	Reserved for future use.
0xa8	Text	Operator	
0xa9	TextPath	Operator	
0xaa		Operator	Reserved for future use.
0xab		Operator	Reserved for future use.
0xac		Operator	Reserved for future use.
0xad		Operator	Reserved for future use.
0xae		Operator	Reserved for future use.
0xaf		Operator	Reserved for future use.
0xb0	BeginImage	Operator	
0xb1	ReadImage	Operator	
0xb2	EndImage	Operator	
0xb3	BeginRastPattern	Operator	
0xb4	ReadRastPattern	Operator	
0xb5	EndRastPattern	Operator	
0xb6	BeginScan	Operator	
0xb7		Operator	Reserved for future use.
0xb8	EndScan	Operator	
0xb9	ScanLineRel	Operator	
0xba – 0xbe			Reserved for future use.
0xbf	Passthrough	Operator	XL 3.0
0xc0	ubyte	Data Type	Unsigned 8-bit value
0xc1	uint16	Data Type	Unsigned 16-bit value
0xc2	uint32	Data Type	Unsigned 32-bit value
0xc3	sint16	Data Type	Signed 16-bit value
0xc4	sint32	Data Type	Signed 32-bit value

0xc5	real32	Data Type	Real number value
0xc6		Data Type	Reserved for future use.
0xc7		Data Type	Reserved for future use.
0xc8	ubyte_array	Data Type	Array of Unsigned 8-bit values
0xc9	uint16_array	Data Type	Array of Unsigned 16-bit values
0xca	uint32_array	Data Type	Array of Unsigned 32-bit values
0xcb	sint16_array	Data Type	Array of Signed 16-bit values
0xcc	sint32_array	Data Type	Array of Signed 32-bit values
0xcd	real32_array	Data Type	Array of Real number values
0xce		Data Type	Reserved for future use.
0xcf		Data Type	Reserved for future use.
0xd0	ubyte_xy	Data Type	Two Unsigned 8-bit values
0xd1	uint16_xy	Data Type	Two Unsigned 16-bit values
0xd2	uint32_xy	Data Type	Two Unsigned 32-bit values
0xd3	sint16_xy	Data Type	Two Signed 16-bit values
0xd4	sint32_xy	Data Type	Two Signed 32-bit values
0xd5	real32_xy	Data Type	Two Real number values
0xd6-0xdf		Data Type	Reserved for future use.
0xe0 – 0xf7			Reserved for future use.
0xf8	attr_ubyte	Attribute	Unsigned, 8-bit Attribute
0xf9	attr_uint16	Attribute	Unsigned, 16-bit Attribute
0xfa	dataLength	Embed Data	Embedded Data Follows
0xfb	dataLengthByte	Embed Data	Embedded Data Follows (0-255 bytes)
0xfc – 0xff			Reserved for future use.

Appendix E. Attribute ID Number to Attribute Name Table

The following table outlines the attribute ID numbers for PCL XL attributes in binary streams.

Attribute ID	Attribute Name
1.	
2.	PaletteDepth
3.	ColorSpace
4.	NullBrush
5.	NullPen
6.	PaletteData
7.	
8.	PatternSelectID
9.	GrayLevel
10.	
11.	RGBColor
12.	PatternOrigin
13.	NewDestinationSize
14.	PrimaryArray
15.	PrimaryDepth
16.	
17.	
18.	
19.	
20.	
21.	
22.	
23.	
24.	
25.	
26.	
27.	
28.	
29.	AllObjectTypes
30.	TextObjects
31.	VectorObjects
32.	RasterObjects
33.	DeviceMatrix
34.	DitherMatrixDataType
35.	DitherOrigin
36.	MediaDestination
37.	MediaSize
38.	MediaSource
39.	MediaType
40.	Orientation
41.	PageAngle
42.	PageOrigin
43.	PageScale
44.	ROP3
45.	TxMode
46.	

47.	CustomMediaSize
48.	CustomMediaSizeUnits
49.	PageCopies
50.	DitherMatrixSize
51.	DitherMatrixDepth
52.	SimplexPageMode
53.	DuplexPageMode
54.	DuplexPageSide
55.	
56.	
57.	
58.	
59.	
60.	
61.	
62.	
63.	
64.	
65.	ArcDirection
66.	BoundingBox
67.	DashOffset
68.	EllipseDimension
69.	EndPoint
70.	FillMode
71.	LineCapStyle
72.	LineJoinStyle
73.	MiterLength
74.	LineDashStyle
75.	PenWidth
76.	Point
77.	NumberOfPoints
78.	SolidLine
79.	StartPoint
80.	PointType
81.	ControlPoint1
82.	ControlPoint2
83.	ClipRegion
84.	ClipMode
85.	
86.	
87.	
88.	
89.	
90.	
91.	
92.	
93.	
94.	
95.	
96.	
97.	
98.	ColorDepth
99.	BlockHeight

100.	ColorMapping
101.	CompressMode
102.	DestinationBox
103.	DestinationSize
104.	PatternPersistence
105.	PatternDefineID
106.	
107.	SourceHeight
108.	SourceWidth
109.	StartLine
110.	PadBytesMultiple
111.	BlockByteLength
112.	
113.	
114.	
115.	NumberOfScanLines
116.	
117.	
118.	
119.	
120.	ColorTreatment
121.	
122.	Reserved
123.	
124.	
125.	
126.	
127.	
128.	
129.	CommentData
130.	DataOrg
131.	
132.	
133.	
134.	Measure
135.	
136.	SourceType
137.	UnitsPerMeasure
138.	
139.	StreamName
140.	StreamDataLength
141.	
142.	
143.	ErrorReport
144.	
145.	Reserved
146.	Reserved
147.	Reserved
148.	Reserved
149.	Reserved
150.	Reserved
151.	Reserved
152.	Reserved

153.	Reserved
154.	Reserved
155.	Reserved
156.	Reserved
157.	Reserved
158.	Reserved
159.	Reserved
160.	
161.	CharAngle
162.	CharCode
163.	CharDataSize
164.	CharScale
165.	CharShear
166.	CharSize
167.	FontHeaderLength
168.	FontName
169.	FontFormat
170.	SymbolSet
171.	TextData
172.	CharSubModeArray
173.	PCLSelectFont
174.	
175.	XSpacingData
176.	YSpacingData
177.	CharBoldValue
178.	
179.	

Appendix F. Attribute Name to Data Types Table

The following table outlines the attribute ID numbers for PCL XL attributes in binary streams.

Attribute Name	Valid Value(s)	Data Types	Attribute #	Class.rev #
AllrObjectTypes	HalftoneMethod Enums AdaptiveHalftone Enums ColorTreatment Enums NeutralAxis Enums ColorTrapping Enums	ubyte	29	3.0
ArcDirection	ArcDirection Enumeration	ubyte	65	1.1
BlockByteLength	0 – (2^{32} – 1)	uint32	111	2.0
BlockHeight	0 – 65535	uint16	99	1.1
BoldValue	0.0 – 1.0	real32	177	1.1
BoundingBox	Range of data types	ubyte_box uint16_box sint16_box	66	1.1
CharAngle	-360 <= CharAngle <= 360	uint16 sint16 real32	161	1.1
CharCode	Any value within Range of data types	ubyte uint16	162	1.1
CharDataSize	0 – 65535	uint16	163	1.1
CharScale	Two numeric values less than or equal to 32767.0 and greater than or equal to –32768.0, excluding a value of zero.	ubyte_xy uint16_xy real32_xy	164	1.1
CharShear	Two numeric values within Range of data types and less than 32767.0 and greater than –32768.0	ubyte_xy uint16_xy sint16_xy real32_xy	165	1.1
CharSize	Any numeric value within range of data types but less than 32767.0 and greater than zero.	ubyte uint16 real32	166	1.1
CharSubModeArray	CharSubModeArray Enumeration	ubyte_array	172	1.1
ClipMode	ClipMode Enumeration	ubyte	84	1.1
ClipRegion	ClipRegion Enumeration	ubyte	83	1.1
ColorDepth	ColorDepth Enumeration	ubyte	98	1.1
ColorMapping	ColorMapping Enumeration	ubyte	100	1.1
ColorSpace	ColorSpace Enumeration	ubyte	3	1.1
ColorTreatment	Enumeration	ubyte	120	2.1
CommentData	Any data	ubyte_array uint16_array	129	1.1
CompressMode	Compress Mode Enumeration	ubyte	101	1.1

ControlPoint1	Range of data types	ubyte_xy uint16_xy sint16_xy	81	1.1
ControlPoint2	Range of data types	ubyte_xy uint16_xy sint16_xy	82	1.1
CustomMediaSize	dimensions of physical custom media size	uint16_xy real32_xy	47	1.1
CustomMediaSizeUnits	Measure Enumeration	ubyte	48	1.1
DashOffset	Range of data types	ubyte uint16 sint16	67	1.1
DataOrg	DataOrg Enumeration	ubyte	130	1.1
DestinationBox	Four numeric values:	uint16_box	102	1.1
DestinationSize	Two numeric values; x nor y equal to zero	uint16_xy	103	1.1
DeviceMatrix	DitherMatrix Enumeration	ubyte	33	1.1
DitherMatrixDataType	eUByte	ubyte	34	1.1
DitherMatrixDepth	e8Bit	ubyte	51	1.1
DitherMatrixSize	1-256	uint16_xy	50	1.1
DitherOrigin	Range of data types	ubyte_xy uint16_xy sint16_xy	35	1.1
DuplexPageMode	DuplexPageMode enumeration	ubyte	53	1.1
DuplexPageSide	DuplexPageSide enumeration	ubyte	54	1.1
EllipseDimension	Range of data types	ubyte_xy uint16_xy	68	1.1
EndPoint	Range of data types	ubyte_xy uint16_xy sint16_xy	69	1.1
ErrorReport	ErrorReport Enumeration	ubyte	143	1.1
FillMode	FillMode Enumeration	ubyte	70	1.1
FontFormat	0	ubyte	169	1.1
FontHeaderLength	0 – 65535	uint16	167	1.1
FontName	Valid Font Name	ubyte_array	168	1.1
GrayLevel	0 – 1.0 if real or range of integer data type	real32 ubyte	9	1.1
LineCapStyle	LineCap Enumeration	ubyte	71	1.1
LineDashStyle	Range of data types Maximum array size = MAXDASHES (device-dependent)	ubyte_array uint16_array sint16_array	74	1.1
LineJoinStyle	LineJoin Enumeration	ubyte	72	1.1
Measure	Measure Enumeration	ubyte	134	1.1
MediaSize	MediaSize Enumeration	ubyte	37	1.1
MediaSource	MediaSource Enumeration	ubyte	38	1.1
MiterLength	range of data types	ubyte uint16	73	1.1

NewDestinationSize	range of data types; neither x nor y equal to zero	uint16_xy	13	1.1
NullBrush	0	ubyte	4	1.1
NullPen	0	ubyte	5	1.1
NumberOfPoints	Range of data types	ubyte uint16	77	1.1
NumberOfScanLines	Range of data types	uint16	115	1.1
Orientation	Orientation Enumeration	ubyte	40	1.1
PadBytesMultiple	1-255	ubyte	110	2.0
PageAngle	Positive or Negative multiples of 90: -360, -270, -180, -90, 0, 90, 180, 270, 360	uint16 sint16	41	1.1
PageCopies	range of data types; zero causes the page not to be imaged	uint16	49	1.1
PageOrigin	Range of data types	ubyte_xy uint16_xy sint16_xy	42	1.1
PageScale	Two numeric values less than or equal to 32767.0 and greater than or equal to 0.	ubyte_xy uint16_xy real32_xy	43	1.1
PaletteData	Any data in the range of the array elements. If eGray, array lengths of 2, 16, and 256 are allowed. If eRGB, array lengths of 6, 48, 768 are allowed.	ubyte_array	6	1.1
PaletteDepth	ColorDepth Enumeration	ubyte	2	1.1
PatternDefineID	Range of data types	sint16	105	1.1
PatternOrigin	Range of data types	sint16_xy	12	1.1
PatternPersistence	PatternPersistence Enumeration	ubyte	104	1.1
PatternSelectID	Range of data types	sint16	8	1.1
PenWidth	Range of data types: zero or more	ubyte uint16	75	1.1
PCLSelectFont	Valid PCL Font Selector String	ubyte_array	173	3.0
Point	Two numeric values within Range of data types	ubyte_xy uint16_xy sint16_xy	76	1.1
PointType	DataType Enumeration	ubyte	80	1.1
PrimaryArray	Color data for colorspace	real32_array ubyte_array	14	2.0
PrimaryDepth	Primary Depth Enumeration	ubyte	15	2.0
RasterObjects	HalftoneMethod Enums AdaptiveHalftone Enums ColorTreatment Enums NeutralAxis Enums ColorTrapping Enums	ubyte	32	3.0

RGBColor	Three values: 0 – 1.0 if real or range of integer data type	real32_array ubyte_array	11	1.1
ROP3	range of data types	ubyte	44	1.1
SimplexPageMode	SimplexPageMode Enumeration	ubyte	52	1.1
SolidLine	0	ubyte	78	1.1
SourceHeight	1 – 65535	uint16	107	1.1
SourceType	DataSource Enumeration	ubyte	136	1.1
SourceWidth	1 – 65535	uint16	108	1.1
StartLine	0 – 65535	uint16	109	1.1
StartPoint	Range of data types	ubyte_xy uint16_xy sint16_xy	79	1.1
StreamDataLength	Range of data types	uint32	140	1.1
StreamName	ASCII character string	ubyte_array uint16_array	139	1.1
SymbolSet	SymbolSet Enumeration	uint16	170	1.1
TextData	Any character codes in range of data types	ubyte_array uint16_array	171	1.1
TextObjects	HalftoneMethod Enums AdaptiveHalftone Enums ColorTreatment Enums NeutralAxis Enums ColorTrapping Enums	ubyte	30	3.0
TxMode	TxMode enumeration	ubyte	45	1.1
UnitsPerMeasure: xUnits, yUnits	two positive numeric values maximum of 65535.0	uint16_xy real32_xy	114	1.1
VectorObjects	HalftoneMethod Enums AdaptiveHalftone Enums ColorTreatment Enums NeutralAxis Enums ColorTrapping Enums	ubyte	31	3.0
WritingMode	WritingMode enumeration	Ubyte	173	2.0
XSpacingData	Range of data types—must be same size as TextData array	ubyte_array uint16_array sint16_array	175	1.1
YSpacingData	Range of data types—must be same size as TextData array	ubyte_array uint16_array sint16_array	176	1.1

Appendix G. Attribute Value Enumerations Table

The following are values used for enumerated data types through out this document. These enumerations are standardized across protocol classes. All values are listed in decimal form.

Attribute Name / Enumeration	Value	Class
AllObjectTypes **TextObjects** **RasterObjects** **VectorObjects**		
(AdaptiveHalftone Enums)		
eDisable	0	3.0
eEnable	1	3.0
(HalftoneMethod Enums)		
eHighLPI	0	1.1
eMediumLPI	1	3.0
eLowLPI	2	3.0
(ColorTrapping Enums)		
eDisable	0	3.0
eMax	1	3.0
eNormal	2	3.0
eLight	3	3.0
(NeutralAxis Enums)		
eTonerBlack	0	3.0
eProcessBlack	1	3.0
(ColorTreatment Enums)		
eNoTreatment	0	2.1
eScreenMatch	1	2.1
eVivid	2	2.1
ArcDirection		
eClockWise	0	1.1
eCounterClockWise	1	1.1
CharSubModeArray		
eNoSubstitution	0	1.1
eVerticalSubstitution	1	1.1
ClipMode		
see FillMode Enumeration		
ClipRegion		
eInterior	0	1.1
eExterior	1	1.1
ColorDepth		
e1Bit	0	1.1
e4Bit	1	1.1
e8Bit	2	1.1
ColorMapping		
eDirectPixel	0	1.1

eIndexedPixel	1	1.1
ColorSpace		
eGray	1	1.1
eRGB	2	1.1
ColorTreatment		
eNoTreatment	0	2.1
eScreenMatch	1	2.1
eVivid	2	2.1
CompressMode		
eNoCompression	0	1.1
eRLECompression	1	1.1
eJPEGCompression	2	2.0
eDeltaRowCompression	3	2.1
DataOrg		
eBinaryHighByteFirst	0	1.1
eBinaryLowByteFirst	1	1.1
DataSource		
eDefault	0	1.1
DataType		
eUByte	0	1.1
eSByte	1	1.1
eUint16	2	1.1
eSint16	3	1.1
DitherMatrix		
eDeviceBest	0	1.1
DuplexPageMode		
eDuplexHorizontalBinding	0	1.1
eDuplexVerticalBinding	1	1.1
DuplexPageSide		
eFrontMediaSide	0	1.1
eBackMediaSide	1	1.1
ErrorReport		
eNoReporting	0	1.1
eBackChannel(BackCh)	1	1.1
eErrorPage(ErrPage)	2	1.1
eBackChAndErrPage	3	1.1
eNWBackChannel	4	2.0
eNWErrorPage	5	2.0
eNWBackChAndErrPage	6	2.0
FillMode		
eNonZeroWinding	0	1.1
eEvenOdd	1	1.1

LineCapStyle		
eButtCap	0	1.1
eRoundCap	1	1.1
eSquareCap	2	1.1
eTriangleCap	3	1.1
LineJoin		
eMiterJoin	0	1.1
eRoundJoin	1	1.1
eBevelJoin	2	1.1
eNoJoin	3	1.1
Measure		
eInch	0	1.1
eMillimeter	1	1.1
eTenthsOfAMillimeter	2	1.1
MediaSize Enumerations		
eDefaultPaperSize	96	2.1
eLetterPaper	0	1.1
eLegalPaper	1	1.1
eA4Paper	2	1.1
eExecPaper	3	1.1
eLedgerPaper	4	1.1
eA3Paper	5	1.1
eCOM10Envelope	6	1.1
eMonarchEnvelope	7	1.1
eC5Envelope	8	1.1
eDLEnvelope	9	1.1
eJB4Paper	10	1.1
eJB5Paper	11	1.1
eB5Paper	13	2.1
eB5Envelope	12	1.1
eJPostcard	14	1.1
eJDoublePostcard	15	1.1
eA5Paper	16	1.1
eA6Paper	17	2.0
eJB6Paper	18	2.0
JIS8K	19	2.1
JIS16K	20	2.1
JISExec	21	2.1
MediaSource		
eDefaultSource	0	1.1
eAutoSelect	1	1.1
eManualFeed	2	1.1
eMultiPurposeTray	3	1.1
eUpperCassette	4	1.1
eLowerCassette	5	1.1
eEnvelopeTray	6	1.1
eThirdCassette	7	2.0
External Trays[*] 1-248	8-255	2.0

MediaDestination		
eDefaultDestination	0	2.0
eFaceDownBin	1	2.0
eFaceUpBin	2	2.0
eJobOffsetBin	3	2.0
External Bins[**] 1-251	5-255	
Orientation		
ePortraitOrientation	0	1.1
eLandscapeOrientation	1	1.1
eReversePortrait	2	1.1
eReverseLandscape	3	1.1
eDefaultOrientation	4	2.1
PatternPersistence		
eTempPattern	0	1.1
ePagePattern	1	1.1
eSessionPattern	2	1.1
RasterObjects (See AllObjectTypes)		
SymbolSet		
See Appendix O.		
SimplexPageMode		
eSimplexFrontSide	0	1.1
TextObjects (See AllObjectTypes)		
TxMode		
eOpaque	0	1.1
eTransparent	1	1.1
VectorObjects (See AllObjectTypes)		
WritingMode		
eHorizontal	0	2.0
eVertical	1	2.0

[*] External input trays 1 through 248 are selected by substituting the values 8 through 255 for the enumerated values. Example, 8 = first external input tray, 9 = second external input tray, etc.

[**] External output bins 1 through 251 are selected by substituting the values 5 through 255 for the enumerated values. Example, 5 = first external output bin, 6 = second external output bin, etc.

Appendix K. PCL XL Error and Warning Codes

Generic Operator Errors

PCL XL reports several error codes that apply to more than one operator in a general way. Generic operator error codes are described in this section.

Generic Errors

IllegalOperatorSequence	PCL XL read an operator that is out-of-sequence according to the legal sequencing of operators defined by the protocol. For example, a *ReadImage* occurring prior to a *BeginImage* yields this error.
IllegalTag	PCL XL had expected to read a data tag or operator tag and instead read something that is undefined for the current version of PCL XL.
InsufficientMemory	The amount of memory required to complete the current operation is unavailable.
InternalOverflow	Completing the current operation requires an amount of one or more internal resources that exceeds the maximum allowed.

Generic Attribute Errors

IllegalArraySize	PCL XL may yield this error for any array that cannot be zero through infinity in length (see the Attribute ID table in the appendix for valid array sizes).
IllegalAttribute	An attribute provided to an operator is not valid for that operator. For example, the *BeginPage* operator given a *Point* attribute yields this error.
IllegalAttributeCombination	An operator has two or more attributes that either conflict or do not make sense when presented together. For example, providing *BeginPage* with both the *MediaSize* and *CustomMediaSize* attribute yields this error.
IllegalAttributeDataType	The data type of the data value provided to an attribute is not valid for that attribute. For example, the single-value "*sint16"* data type is not valid for the Point attribute which requires two values: one each for the x and y coordinates of the point. The "*sint16_xy*" data type is valid for the Point attribute.
IllegalAttributeValue	PCL XL will yield this error if an attribute value given is out-of-range for the attribute value expected.
MissingAttribute	An operator with required attributes is missing one or more of the required attributes. For example, *SetCursor* without the *Point* attribute yields this error.

Generic Cursor Errors

CurrentCursorUndefined	An operator needing the current cursor position in the current path found that there was no current path and therefore no current cursor position. For example, *SetCursorRel* yields this error if performed immediately following a *NewPath* operator.

Generic Font Errors

NoCurrentFont	An operator needing a valid current font found that no font is set in the current graphics state. For example, attempting to perform the *Text or TextPath* operators without a successful *SetFont* operation for the current graphics state yields this error.
BadFontData	The font and/or character description data for the current font operation is incompatible with the current font operation. For example, attempting to perform character scaling on a bitmap characters by setting CharScale to something other than 1.0 will yield this error.

Generic Data Source Errors

DataSourceNotOpen	An operator that requires data from the data source yields this error when no data source is open.
ExtraData	An operator reading the *default* data source and finding that there is more data than required for the operation yields this error.
IllegalDataLength	An operator reading the *default* data source and finding that the length of data is not either a required constant length nor a required multiple of a constant length yields this error. For example, the BezierPath operator yields this error if the length of data in the default data source is not a multiple of 3.
IllegalDataValue	An operator that reads the data source and finds that a field in the data source contains an illegal value for that operator yields this error. For example, an illegal value in the x-pair type field for the *ScanLineRel* operator yields this error.
MissingData	An operator reading the data source and finding that there is less data than is required to complete the operation yields this error.

Operator-Specific Errors

PCL XL reports several error codes that are specific to an operator. Specific operator errors are described in this section.

BeginChar Errors

CannotReplaceCharacter	An attempt was made to download a character to an internal font or to a font on a mass-storage device.
FontUndefined	The user is attempting to start a character definition with *BeginChar*, but the font name specified is undefined.

BeginFontHeader Errors

FontNameAlreadyExists	The font name given already exists. The font must be removed before re-defining the header.

BeginImage/BeginRastPattern Errors

ImagePaletteMismatch	The palette is too small or too large for the ColorDepth specified for the indexed image or pattern.
MissingPalette	There is no palette for the indexed image/pattern.

BeginPage Warnings

IllegalMediaSize	The values for MediaSize or CustomMediaSize or CustomMediaSizeUnits are out of range for the current device.
IllegalMediaSource	The value for MediaSource is out of range for the current device or device configuration.
IllegalMediaDestination	The value for MediaDestination is out of range for the current device or device configuration.
IllegalOrientation	The value for page orientation is out of range for the current device.

OpenDataSource Errors

DataSourceNotClosed	A data source is already open (not closed).

PushGS Errors

MaxGSLevelsExceeded	The maximum number of graphics state save levels has been exceeded. The maximum level is determined by the available memory.

ReadChar Errors

FSTMismatch	The value of the character format field is inconsistent with the value of the Font Scaling Technology field in the corresponding font header.
UnsupportedCharacterClass	The value of the character class field represents a character class that is not supported by the current version of PCL XL.
UnsupportedCharacterFormat	The value of the character format field represents a character format that is not supported by the current version of PCL XL.
IllegalCharacterData	The character data is in a format unrecognized by PCL XL.

ReadFontHeader Errors

IllegalFontData	The font header data is in a format unrecognized by PCL XL.
IllegalFontHeaderFields	The font header Font Format, Font Scaling Technology, Variety, and/or Orientation fields contain invalid values.
IllegalNullSegmentSize	The size of the Null segment in the font header is invalid.
IllegalFontSegment	A segment included in the font header is not defined for the corresponding font scaling technology.
MissingRequiredSegment	A segment required by the corresponding font scaling technology is missing.
IllegalGlobalTrueTypeSegment	The global TrueType segment is in a format unrecognized by PCL XL.
IllegalGalleyCharacterSegment	The galley character segment is in a format unrecognized by PCL XL.
IllegalVerticalTxSegment	The vertical transformation segment is in a format unrecognized by PCL XL.
IllegalBitmapResolutionSegment	The bitmap resolution segment is in a format unrecognized by PCL XL.

RemoveFont Warnings

UndefinedFontNotRemoved	An attempt was made to remove an undefined font.
InternalFontNotRemoved	An attempt was made to remove an internal font.
MassStorageFontNot Removed	An attempt was made to remove a font on a mass storage device.

ExecStream Errors

StreamUndefined	An attempt was made to execute an undefined stream.
IllegalOpSequence	An attempt was made to execute a stream within another stream (Stream Nesting) when the protocol class is set to less than 2.1.
StreamNestingError	An attempt was made by a stream to start another copy of itself.
StreamAlreadyRunning	An attempt was made to start another copy of an already executing stream.
StreamStackFull	An attempt was made to start more that 32 nested streams.
InternalStreamError	Normally XL cannot exit and clean up while streams are executing. If this does occur and XL cannot safely clean up the stream stack entries, this error will result.

RemoveStream Warnings

UndefinedStreamNot Removed	An attempt was made to remove an undefined stream.
InternalStreamNotRemoved	An attempt was made to remove an internal stream.
MassStorageStreamNotRemoved	An attempt was made to remove a stream on a mass storage device.
StreamAlreadyRunning	An attempt was made to remove a copy of an already executing stream.

SetBrushSource/SetPenSource Errors

ColorSpaceMismatch	The color attribute for SetPen/SetBrush (i.e. RGBColor or GrayLevel) does not match the ColorSpace at the current graphics state level. The ColorSpace was either set by default (at BeginPage) or by the most recent SetColorSpace operator.
RasterPatternUndefined	The value given for PatternSelectID does not identify a raster pattern.

SetClipReplace/SetClipIntersect/SetClipRectangle Errors

ClipModeMismatch	The ClipRegion attribute is "eExterior" and the ClipMode is "eNonZeroWinding." When the ClipRegion is "eExterior" the ClipMode must be "eEvenOdd."

SetFont Errors

FontUndefinedNoSubstituteFound	An attempt was made to set a font that was not defined in the device. A font substitution could not be found by PCL XL.
SymbolSetRemapUndefined	The font is defined, but unusable with the requested symbol set.

SetFont Warnings

"font name" substituted for "font name"	An attempt was made to set a font that was undefined in the device. A substitution was found and made by PCL XL.

Stream Header Errors

UnsupportedBinding	PCL XL does not support the binding requested in the stream header.
UnsupportedClassName	The class name is unknown to PCL XL (e.g. "PCL-XY").
UnsupportedProtocol	The protocol class number and/or revision is not supported by current version of PCL XL.
IllegalStreamHeader	The stream header is in a format unrecognized by PCL XL.