3

Parameter Types

XIE and Core X Types

X Image Extension *types* are defined in this section. All XIE parameters are defined as being either one of the extension types, or one of the core protocol types. XIE makes use of the following core protocol types (defined in the X11 core protocol specification):

BITMAP	BOOL	CARD8	CARD16
CARD32	CHAR2B	COLORMAP	DRAWABLE
EVENT	GCONTEXT	INT8	INT16
INT32	PIXMAP	STRING8	TIMESTAMP
VISUAL	WINDOW	XID*	

* XID is used to refer to the core X11 type defined as the identifier for a resource.

Techniques

Several *standard* techniques are defined in this document. Each is assigned a technique number and a descriptive name string. XIE can also be extended with *private* techniques that are implementation dependent. Private technique numbers are generated dynamically. Private technique name strings include the name of the organization that defined the technique. The organization name is encompassed by "_" (underscore) characters. Technique naming and numbering conventions are summarized in Table 3-1.

	standard technique	**private technique**
name	<standard-technique-name> example: "ERROR-DIFFUSION"	_<organization-name>_<private-technique-name> example: "_PHOTOCO_SQUASH-BITS"
number	MS bit is zero (range: 0 - 32767)	MS bit is one (range: 32768 - 65535)

Table 3-1 Technique naming and numbering conventions

The technique number is supplied to pipeline elements to specify the desired technique. Numbers for standard techniques can be hard-coded, whereas numbers for private techniques must be obtained using the **QueryTechniques** protocol request.

For more information on techniques and a description of the techniques defined in this document, refer to Appendix A. A list of techniques itemized by ServiceClass can be found in Appendix B. Numbers assigned to standard techniques are encoded in Appendix C.

Definitions

XieTypAlignment

Alignment defines the valid pixel and scanline alignments allowed for image data transported through the protocol stream. The server's **Alignment** attribute is returned by the **QueryImageExtension** protocol request.

XieTypAlignment { XieValAlignable, XieValArbitrary }

- ***Alignable*** data units must fit evenly within a byte, or they must fill a byte, or fill a multiple of bytes (i.e. pixels may be 1, 2, 4, 8, 16, 24*, or 32* bits in the protocol stream).
- ***Arbitrary*** data units may fall at any bit address (i.e. 10 bit packed pixels are acceptable in the protocol stream).

* XIE supports pixels up to sixteen (16) bits per band for both ***SingleBand*** and ***TripleBand*** data. In addition, ***SingleBand*** data are supported up to the depth of the deepest `DRAWABLE` supported by the server.

XieTypArithmeticOp

ArithmeticOp defines the valid operations for the **Arithmetic** process element.

XieTypArithmeticOp { XieValAdd,
XieValSub, XieValSubRev,
XieValMul,
XieValDiv, XieValDivRev,
XieValMin, XieValMax,
XieValGamma }

- ***Add*** src1 + src2 or src1 + constant
- ***Sub*** src1 - src2 or src1 - constant
- ***SubRev*** src2 - src1 or constant - src1
- ***Mul*** src1 * constant
- ***Div*** src1 / constant
- ***DivRev*** constant / src1
- ***Min*** minimum(src1, src2) or minimum(src1, constant)
- ***Max*** maximum(src1, src2) or maximum(src1, constant)
- ***Gamma*** if src1 is ***Constrained*** $(\text{levels} - 1) * ((\text{src1} / (\text{levels} - 1))^{\text{constant}})$
  if src1 is ***Unconstrained*** $\text{src1}^{\text{constant}}$

XieTypColorAllocTechnique

ColorAllocTechnique defines the recognized color allocation techniques used by the **ConvertToIndex** element.

XieTypColorAllocTechnique	{ XieValDefault, XieValColorAlloc_AllocAll, XieValColorAlloc_Match, XieValColorAlloc_Requantize }

* ***Default***	
ColorAlloc_AllocAll	"ALLOC-ALL"
ColorAlloc_Match	"MATCH"
ColorAlloc_Requantize	"REQUANTIZE"
private-technique-number	private-name-string

* The server is required to support the ***Default*** technique which is bound to one of the standard techniques defined above or a private technique.

XieTypColorList

ColorList is the type for the XIE resource used to store `COLORMAP` allocations made by the **ConvertToIndex** element. A **ColorList** is a permanent resource and as such must be created and destroyed by the client. A **ColorList** contains a counted list of the pixel indices that were allocated, and the resource-id of the `COLORMAP` from which they were allocated.

XieTypColorList `XID`

XieTypCompareOp

CompareOp defines the operators for the **Compare** element.

XieTypCompareOp	{ XieValLT,	XieValLE,
	XieValEQ,	XieValNE,
	XieValGT,	XieValGE }

- ***LT*** src1 $<$ src2 or src1 $<$ constant
- ***LE*** src1 $\leq$ src2 or src1 $\leq$ constant
- ***EQ*** src1 $=$ src2 or src1 $=$ constant
- ***NE*** src1 $\neq$ src2 or src1 $\neq$ constant
- ***GT*** src1 $>$ src2 or src1 $>$ constant
- ***GE*** src1 $\geq$ src2 or src1 $\geq$ constant

XieTypConstant

Constant is typically used to supply per-band constant values. All constants are passed as floats. When a constant is to be used as a substitute for a ***Constrained*** image pixel, the constant is rounded to the nearest integer, and then hard-clipped to the range of *levels* that applies to the pixel for which it is a substitute. In most other cases the constant is used as is (i.e. as a float).

XieTypConstant XieTypTripletofXieTypFloat

Element	Op/*arg*/***Tech***	Type	Usage	If *Constrained* …
Arithmetic	Add	**Constant**	pixel value when monadic	round/hard-clip
	Sub	**Constant**	pixel value when monadic	round/hard-clip
	SubRev	**Constant**	pixel value when monadic	round/hard-clip
	Min	**Constant**	pixel value when monadic	round/hard-clip
	Max	**Constant**	pixel value when monadic	round/hard-clip
	Mul	**Constant**	multiplicand	use as is
	Div	**Constant**	divisor	use as is
	DivRev	**Constant**	dividend	use as is
	Gamma	**Constant**	exponent	use as is
BandExtract	*coefficients*	**Constant**	multipliers	use as is
	bias	**Float**	final offset	use as is
Blend	*constant*	**Constant**	pixel value when monadic	round/hard-clip
	alpha-constant	**Float**	constant alpha value	use as is
Compare	*constant*	**Constant**	pixel value when monadic	round/hard-clip
Constrain	***ClipScale***	**Constant**	scale src levels to dst levels	use as is
Convert From/To RGB	***CIElab***	**Matrix**	conversion matrix	use as is
	CIElabShift	**Constant**	white-point	use as is
	CIEXYZ	**Matrix**	conversion matrix	use as is
	YCbCr	**Constant**	luma values and bias	use as is
	YCC	**Constant**	luma values	use as is
	YCC	**Float**	scale	use as is
ConvertToIndex	***Match***	**Float**	match-limit, gray-limit	use as is
Convolve	*kernel*	LISTof**Float**	convolution kernel	use as is
	Constant	**Constant**	edge pixel value	round/hard-clip
Geometry	a,b,c,d,t_x,t_y	**Float**	coefficients [6]	use as is
	constant	**Constant**	fill pixel value	round/hard-clip
	Gaussian	**Float**	sigma, normalize	use as is
Logical	*constant*	**Constant**	pixel value when monadic	round/hard-clip
MatchHistogram	***Gaussian***	**Float**	mean, sigma	use as is
	Hyperbolic	**Float**	constant	use as is
PasteUp	*constant*	**Constant**	fill pixel value	round/hard-clip

Table 3-2 Treatment of floating point parameters passed through the protocol

XieTypConstrainTechnique

ConstrainTechnique defines various methods of constraining data. These techniques are applied by the **Constrain** element.

XieTypConstrainTechnique	{ XieValConstrain_ClipScale, XieValConstrain_HardClip }
Constrain_ClipScale	"CLIP-SCALE"
Constrain_HardClip	"HARD-CLIP"
private-technique-number	private-name-string

XieTypConvertFromRGBTechnique

ConvertFromRGBTechnique defines the trichromatic colorspaces known to the **ConvertFromRGB** element.

XieTypConvertFromRGBTechnique	{ XieValConvertFromRGB_CIELab, XieValConvertFromRGB_CIEXYZ, XieValConvertFromRGB_YCbCr, XieValConvertFromRGB_YCC }
ConvertFromRGB_CIELab	"CIELAB"
ConvertFromRGB_CIEXYZ	"CIEXYZ"
ConvertFromRGB_YCbCr	"YCbCr"
ConvertFromRGB_YCC	"YCC"
private-technique-number	private-name-string

XieTypConvertToRGBTechnique

ConvertToRGBTechnque defines the trichromatic colorspaces known to the **ConvertToRGB** element.

XieTypConvertToRGBTechnique	{ XieValConvertToRGB_CIELab, XieValConvertToRGB_CIEXYZ, XieValConvertToRGB_YCbCr, XieValConvertToRGB_YCC }
ConvertToRGB_CIELab	"CIELAB"
ConvertToRGB_CIEXYZ	"CIEXYZ"
ConvertToRGB_YCbCr	"YCbCr"
ConvertToRGB_YCC	"YCC"
private-technique-number	private-name-string

XieTypConvolveTechnique

ConvolveTechnique provides various methods of handling edge conditions in the **Convolve** element. This technique determines what pixel values are used when **Convolve** requires data beyond the image bounds.

XieTypConvolveTechnique	{ XieValDefault, XieValConvolve_Constant, XieValConvolve_Replicate }

* ***Default***	
Convolve_Constant	"CONSTANT"
Convolve_Replicate	"REPLICATE"
private-technique-number	private-name-string

* The server is required to support the ***Default*** technique which is bound to one of the standard techniques defined above or a private technique.

XieTypDataClass

DataClass defines the class of data being transported over the wire.

XieTypDataClass	{ XieValSingleBand, XieValTripleBand }

- ***SingleBand*** index data (e.g. `ZPixmap` format `COLORMAP` indices), or achromatic data (bitonal or gray-scale)
- ***TripleBand*** non-index trichromatic data (RGB or other supported colorspaces)

XieTypDataStream

DataStream is a segmented stream of data units that are transported in either direction through the protocol stream. Segments of image data are transported as an arbitrary number of unsigned bytes, whereas all other types of data (e.g. lookup table entries, rectangles, histogram counts, etc.) must be transported as one or more complete aggregates. The status indicating the eventual end of data, is supplied in the associated protocol request or reply. The interpretation of the data stream is determined by parameters specified to the **PhotoElement** accepting or generating the data.

XieTypDataStream `LISTofCARD8`

XieTypDataType

DataType defines the limits and processing model for the data.

XieTypDataType	{ XieValConstrained, XieValUnconstrained }

- ***Constrained*** pixels are integer values and are limited to the range [0, *levels*-1]
- ***Unconstrained*** pixels may be any arbitrary value

XieTypDecodeTechnique

DecodeTechnique defines the techniques that can be used to interpret uncompressed image data or decode compressed images.

XieTypDecodeTechnique	{ XieValDecode_UncompressedSingle, XieValDecode_UncompressedTriple, XieValDecode_CCITT-G31D, XieValDecode_CCITT-G32D, XieValDecode_CCITT-G42D, XieValDecode_JPEG-Baseline, XieValDecode_JPEG-Lossless, XieValDecode_TIFF-2, XieValDecode_TIFF-PackBits }

Decode_UncompressedSingle	"UNCOMPRESSED-SINGLE"
Decode_UncompressedTriple	"UNCOMPRESSED-TRIPLE"
Decode_CCITT-G31D	"CCITT-G31D"
Decode_CCITT-G32D	"CCITT-G32D"
Decode_CCITT-G42D	"CCITT-G42D"
Decode_JPEG-Baseline	"JPEG-BASELINE"
Decode_JPEG-Lossless	"JPEG-LOSSLESS"
Decode_TIFF-2	"TIFF-2"
Decode_TIFF-PackBits	"TIFF-PACKBITS"
private-technique-number	private-name-string

XieTypDitherTechnique

DitherTechnique defines the techniques that can be used to dither an image.

XieTypDitherTechnique	{ XieValDefault, XieValDither_ErrorDiffusion, XieValDither_Ordered }

* ***Default***	
Dither_ErrorDiffusion	"ERROR-DIFFUSION"
Dither_Ordered	"ORDERED"
private-technique-number	private-name-string

* The server is required to support the ***Default*** technique which is bound to one of the standard techniques defined above or a private technique.

XieTypEncodeTechnique

EncodeTechnique defines the techniques that can be used to compress an image or format it as uncompressed data.

XieTypEncodeTechnique	{ XieValEncode_ServerChoice, XieValEncode_UncompressedSingle, XieValEncode_UncompressedTriple, XieValEncode_CCITT-G31D, XieValEncode_CCITT-G32D, XieValEncode_CCITT-G42D, XieValEncode_JPEG-Baseline, XieValEncode_JPEG-Lossless, XieValEncode_TIFF-2, XieValEncode_TIFF-PackBits }

Encode_ServerChoice	(not returned by QueryTechniques)
Encode_UncompressedSingle	"UNCOMPRESSED-SINGLE"
Encode_UncompressedTriple	"UNCOMPRESSED-TRIPLE"
Encode_CCITT-G31D	"CCITT-G31D"
Encode_CCITT-G32D	"CCITT-G32D"
Encode_CCITT-G42D	"CCITT-G42D"
Encode_JPEG-Baseline	"JPEG-BASELINE"
Encode_JPEG-Lossless	"JPEG-LOSSLESS"
Encode_TIFF-2	"TIFF-2"
Encode_TIFF-PackBits	"TIFF-PACKBITS"
private-technique-number	private-name-string

XieTypExecutable

Executable is the type used to identify a specific **Photoflo** instance.

XieTypExecutable

[	*name-space*	`ServerIDSpace`	or	XieTypPhotospace
	flo-id	XieTypPhotoflo	or	`CARD32`
]				

Name-space identifies the execution domain used for a **Photoflo**. *Flo-id* identifies a particular instance of a **Photoflo**.

For `stored` **Photoflo**s *name-space* is always *`ServerIDSpace`* (the value zero) and *flo-id* is the **Photoflo**'s resource-id.

For `immediate` **Photoflo**s *name-space* is a **Photospace** resource-id and *flo-id* is a thirty-two (32) bit value that uniquely identifies the instance of the **Photoflo** within *name-space*.

XieTypExportNotify

ExportNotify is used by `ExportClient` elements to control how **ExportAvailable** events are sent to the client.

XieTypExportNotify	{ XieValDisable XieValFirstData XieValNewData }

- ***Disable*** no events are sent
- ***FirstData*** a single event is sent when the first data is available
- ***NewData*** an event is sent each time the amount of available data changes from zero to non-zero

The end of data is signaled by a *final* flag in the **GetClientData** reply that is returned when the last segment of data is retrieved.

XieTypExportState

ExportState is a status value returned from an `ExportClient` element in a reply to the corresponding **GetClientData** protocol request.

XieTypExportState	{ XieValExportDone, XieValExportMore, XieValExportEmpty, XieValExportError }

- ***ExportDone*** all data have been retrieved
- ***ExportMore*** more data currently available
- ***ExportEmpty*** no more data available at this time
- ***ExportError*** an error condition prevents data availability

XieTypFloat

Float specifies data is in the IEEE single-precision format as described in ANSI/IEEE Standard 754-1985, "IEEE Standard for Binary Floating-Point Arithmetic". **Float** is used only for constant parameter values, matrix coefficients, etc., but never for transported image data. The byte order of each **Float** is dealt with in the same manner as other numeric values; it is determined at core protocol connection setup time.

XieTypGamutTechnique

GamutTechnique defines the gamut compression techniques used by **ConvertToRGB** to deal with converted colors which lie outside the gamut of the RGB space.

XieTypGamutTechnique	{ XieValDefault, XieValGamut_None, XieValGamut_ClipRGB }
* ***Default***	
Gamut_None	"NONE"
Gamut_ClipRGB	"CLIP-RGB"
private-technique-number	private-name-string

* The server is required to support the ***Default*** technique which is bound to one of the standard techniques defined above or a private technique.

XieTypGeometryTechnique

GeometryTechnique defines the retrospective image re-sampling techniques used by the **Geometry** element.

XieTypGeometryTechnique	{ XieValDefault, XieValGeometry_Antialias, XieValGeometry_AntialiasByArea, XieValGeometry_AntialiasByLowpass, XieValGeometry_BilinearInterpolation, XieValGeometry_Gaussian, XieValGeometry_NearestNeighbor }
* ***Default***	
Geometry_Antialias	"ANTIALIAS"
Geometry_AntialiasByArea	"ANTIALIAS-BY-AREA"
Geometry_AntialiasByLowpass	"ANTIALIAS-BY-LOWPASS"
Geometry_BilinearInterpolation	"BILINEAR-INTERPOLATION"
Geometry_Gaussian	"GAUSSIAN"
Geometry_NearestNeighbor	"NEAREST-NEIGHBOR"
private-technique-number	private-name-string

* The server is required to support the ***Default*** technique which is bound to one of the standard techniques defined above or a private technique.

XieTypHistogramData

HistogramData defines the organization of histogram entries created by the **ExportClientHistogram** element.

XieTypHistogramData

```
[   value        CARD32
    count        CARD32
]
```

XieTypHistogramShape

HistogramShape defines the various match-histogram shape techniques that can be requested in the **MatchHistogram** element.

XieTypHistogramShape { XieValHistogram_Flat, XieValHistogram_Gaussian, XieValHistogram_Hyperbolic }

Histogram_Flat	"FLAT"
Histogram_Gaussian	"GAUSSIAN"
Histogram_Hyperbolic	"HYPERBOLIC"
private-technique-number	private-name-string

XieTypInterleave

Interleave defines the way in which bands of ***TripleBand*** data may be interleaved.

XieTypInterleave { XieValBandByPixel, XieValBandByPlane }

- ***BandByPixel*** the data for all bands are interleaved on a per-pixel basis and transmitted as a single data plane (e.g. for RGB data red, green, and blue values for a given pixel are immediately followed by the red, green, and blue values for the next pixel). ***BandByPixel*** is similar to the core protocol *ZPixmap* format.
- ***BandByPlane*** the data for each band are transmitted as separate data planes (e.g. for RGB data all red data are transmitted in one plane, all green data in a separate plane, and all blue data in another plane).

Interleave is generally used in conjunction with **Orientation**, which would define the order of the interleaved bands.

XieTypLevels

Levels describes the potential dynamic range for the quantization levels associated with each band of an image.

XieTypLevels	XieTypTripletofCARD32

Note: for ***Constrained*** image data a value of zero (0) represents 2^{32} (4,294,967,296) levels.

XieTypLUT

LUT is the type for the XIE server resource used to hold arrays which map one set of values to another. The arrays are one-dimensional, and the resource holds either one or three arrays. The **ImportClientLUT** and **ImportLUT** elements are used to import **LUT** data into a **Photoflo**, the **Point** element uses the data, the **ExportLUT** element is used to (re)populate the **LUT** resource, and the **ExportClientLUT** element is provided to facilitate sending the **LUT** data back to the client.

XieTypLUT	XID

XieTypMathOp

MathOp defines the valid mathematical operations that can be invoked through the **Math** element. The **MathOp** is applied to each input pixel value to determine the output pixel value.

XieTypMathOp	{ XieValExp, XieValLn, XieValLog2, XieValLog10, XieValSquare, XieValSqrt }

- ***Exp*** exponential
- ***Ln*** natural logarithm
- ***Log2*** logarithm base 2
- ***Log10*** logarithm base 10
- ***Square*** square
- ***Sqrt*** square root

XieTypMatrix

Matrix is a 3 x 3 matrix of floats that is used by some of the colorspace conversion techniques (e.g. conversion between RGB and CIE colorspaces).

XieTypMatrix	XieTypTripletofXieTypConstant

XieTypOrientation

Orientation defines the transmission order of image data units through the protocol stream.

XieTypOrientation { XieValLSFirst, XieValMSFirst }

Note: in the examples that follow, the order of bytes in the **DataStream** should be read from left to right.

encoded-order

Specifies the order of bits within bytes of encoded (compressed) data.

Example: showing 2 bytes (within each byte the first encoded bit is "0", last is bit "7")

LSFirst	***MSFirst***
76543210 76543210	01234567 01234567

fill-order

When multiple pixels are put in the same byte, or a pixel spans multiple bytes, *fill-order* specifies whether the pixels (or parts of a pixel) are packed into the most or least significant bits of a byte first.

Example: showing 8 2-bit pixels ("aa", "bb", "cc", etc.), and 2 10-bit pixels: "aaaaaaaaaa" and "bbbbbbbbbb")

LSFirst	***MSFirst***
ddccbbaa hhggffee	aabbccdd eeffgghh
aaaaaaaa bbbbbbaa xxxxbbbb	aaaaaaaa aabbbbbb bbbbxxxx

pixel-order

For pixels that span a byte boundary, *pixel-order* specifies whether the most or least significant bits of the pixel are put into the **DataStream** first. Any pad bits that are present between pixels always follow the pixel data (i.e. pad bits are not considered to be either LS-bits or MS-bits of the pixel data).

Example: showing 2 10-bit pixels, each with 2 bits of pad (within each pixel the LS-bits are "0" and "a", the MS-bits are "9" and "j", the pad bits are "p")

fill-order	***LSFirst*** *(pixel-order)*	***MSFirst*** *(pixel-order)*
LSFirst	76543210 dcbapp98 ppjihgfe	98765432 jihgpp10 ppfedcba
MSFirst	76543210 98ppdcba jihgfepp	98765432 10ppjihg fedcbapp

band-order

Definition: at the protocol level, the least significant band of trichromatic data is the first band mentioned in the common name of the colorspace (e.g. red is the least significant band of `RGB` data).

For ***BandByPixel*** data, *band-order* specifies whether this band is put in the least or most significant bits of a pixel.

LSFirst	***MSFirst***
$B_1B_0G_2G_1G_0R_2R_1R_0$	$R_2R_1R_0G_2G_1G_0B_1B_0$

For ***BandByPlane*** data, *band-order* specifies whether this band corresponds with the least significant or most significant image plane. Each plane is transported as a separate **DataStream**.

band	***LSFirst***	***MSFirst***
0	$R_7R_6R_5R_4R_3R_2R_1R_0$	$B_7B_6B_5B_4B_3B_2B_1B_0$
1	$G_7G_6G_5G_4G_3G_2G_1G_0$	$G_7G_6G_5G_4G_3G_2G_1G_0$
2	$B_7B_6B_5B_4B_3B_2B_1B_0$	$R_7R_6R_5R_4R_3R_2R_1R_0$

For all other instances where a triplet of per-band information is conveyed through the protocol (levels, constants, etc.), the first element of the triplet corresponds with the least significant band as defined above.

XieTypPhotoElement

PhotoElement, or just “element”, is a generic type used for *import, process,* or *export* elements in a **Photoflo**. The syntax of a generic element was described in Chapter 2. The actual element definitions are found in Chapters 6, 7, and 8.

XieTypPhotoflo

Photoflo is the type for the XIE server resource which is used to represent a sequence of image processing operations. `Stored` **Photoflo**s are permanent resources. As such they must be created and destroyed by the client.

XieTypPhotoflo	`XID`

For `immediate` **Photoflo**s, see type **Photospace** and the **ExecuteImmediate** protocol request.

XieTypPhotofloOutcome

PhotofloOutcome is returned by the **PhotofloDone** event and indicates the reason why the **Photoflo** left the ***Active*** state.

XieTypPhotofloOutcome	{ XieValFloSuccess, XieValFloError, XieValFloAbort }

- ***FloSuccess*** the **Photoflo** completed normally
- ***FloError*** the **Photoflo** terminated due to an error condition
- ***FloAbort*** the **Photoflo** was aborted by the client

XieTypPhotofloState

PhotofloState identifies the current execution state of a **Photoflo**.

XieTypPhotofloState { XieValInactive, XieValActive, XieValNonexistent }

- ***Inactive*** — the state of a `stored` **Photoflo** prior to execution, after an error or abort, or after execution completes. ***Inactive*** `stored` **Photoflo**s can be modified and redefined.
- ***Active*** — the state of a `stored` or `immediate` **Photoflo** during execution.

  A **Photoflo** remains ***Active*** until:

  – all export elements have finished their task, and

  – all data exported for the client have been retrieved, or

  – an error prevents further progress, or

  – the client issues an abort request.
- ***Nonexistent*** — the state of a **Photoflo** that does not exist (never existed or has been destroyed).

XieTypPhotomap

Photomap is the type for the XIE resource used to store image data in the server. A **Photomap** is a permanent resource that can be created and destroyed by the client. **ImportPhotomap** is used to import permanent image data into a **Photoflo**. **ExportPhotomap** is used to (re)populate the resource.

XieTypPhotomap `XID`

XieTypPhotospace

Photospace is the type for the XIE resource used to define a XIE name space within which `immediate` **Photoflo**s may be executed. A **Photospace** is a permanent resource and as such must be created and destroyed by the client.

XieTypPhotospace `XID`

XieTypPhototag

Phototag is the position or index of a **PhotoElement** within a list of elements used to specify a **Photoflo**. The first element in the list has a **Phototag** value of one (1). A **Phototag** value of zero (0) is reserved for special purposes, such as indicating an optional element input that is not connected, or indicating **Photoflo** errors that are not specific to an element.

XieTypPhototag `CARD16`

XieTypProcessDomain

ProcessDomain is a sub-structure inserted in many **PhotoElement** definitions. It is used to restrict the element's processing to a subset of the source data pixels. A **ProcessDomain** can be either a *list-of-rectangles* or a *control-plane*.

XieTypProcessDomain

[	*offset-x*	INT32
	offset-y	INT32
	domain	XieTypPhototag
]		

Offset-x and *offset-y* specifies the spatial registration between the element's data source(s) and the **ProcessDomain** (i.e. offset relative to the origin of the source(s)). *Domain* is the **Phototag** of the element providing the **ProcessDomain** data.

If the value zero (0) is specified for *domain*, processing is not restricted by the processing *domain*, otherwise, *domain* references an element outputting a *list-of-rectangles* or image data which is used as a *control-plane*.

If *domain* is a *list-of-rectangles*, processing is restricted to the intersection of the data source(s) and any rectangle within the list (*offset-x* and *offset-y* are added to the *x* and *y* of each rectangle).

If *domain* is a source of ***Constrained***, ***SingleBand***, bi-level image data (i.e. a *control-plane*), processing is restricted to the intersection of the data source(s) and non-zero locations within *domain*.

ProcessDomain is not supported by ***DIS*** but is included in the subset because the **Point** element takes a **ProcessDomain** as a parameter (in this case *domain* must be zero).

XieTypRectangle

Rectangle describes a rectangle used in **XieTypROI**.

XieTypRectangle

[	*x*	INT32
	y	INT32
	width	CARD32
	height	CARD32
]		

X and *y* are the coordinates of the upper left hand corner of the rectangle within the **ProcessDomain** (the domain's *x,y* offset is also added to these coordinates to position the rectangle relative to the image being processed). *Width* and *height* are the extent of the rectangle.

XieTypROI

ROI *(Rectangles-Of-Interest)* is the XIE resource used to contain a *list-of-rectangles* (input for a **ProcessDomain**). A **ROI** resource is a permanent resource and as such must be created and destroyed by the client. **ImportClientROI** and **ImportROI** elements are used to import a *list-of-rectangles* into a **Photoflo**, the **ExportROI** element is used to (re)populate an **ROI** resource, and the **ExportClientROI** element is provided to facilitate sending rectangles back to the client *.

XieTypROI XID

* The *list-of-rectangles* that can be retrieved by the client need not be identical to the original list that was received from the client, but it must be logically equivalent.

XieTypServiceClass

ServiceClass defines the recognized image processing service sets supported by this X Image Extension standard.

XieTypServiceClass	{ XieValFull, XieValDIS }

- ***Full*** supports the entire XIE protocol as defined in this document
- ***DIS*** supports the `Document Image Subset,` a proper subset of ***Full*** XIE

An itemized list of XIE types, techniques, protocol requests, elements, events, and errors that are included in each **ServiceClass** can be found in Appendix B.

XieTypTechniqueGroup

TechniqueGroup defines the standard technique group names that can be queried using the **QueryTechniques** protocol request.

XieTypTechniqueGroup	{ XieValDefault, XieValAll, XieValColorAlloc, XieValConstrain, XieValConvertFromRGB, XieValConvertToRGB, XieValConvolve, XieValDecode, XieValDither, XieValEncode, XieValGamut, XieValGeometry, XieValHistogram, XieValWhiteAdjust }

- ***Default*** select all default techniques
- ***All*** select all supported techniques
- ***ColorAlloc*** select color allocation techniques
- ***Constrain*** select techniques for constraining data
- ***CovertFromRGB*** select colorspace conversion techniques (for conversion from the RGB colorspace)
- ***CovertToRGB*** select colorspace conversion techniques (for conversion to the RGB colorspace)
- ***Convolve*** select techniques for handling convolution edge conditions
- ***Decode*** select image decoding (decompression) techniques
- ***Dither*** select dithering techniques
- ***Encode*** select image encoding (compression) techniques
- ***Gamut*** select colorspace conversion gamut compression techniques
- ***Geometry*** select geometric sampling techniques
- ***Histogram*** select match-histogram shapes
- ***WhiteAdjust*** select colorspace conversion white point adjustment techniques

If a vendor defined an additional private technique group, it could be discovered by querying for ***All*** groups.

XieTypTechniqueRec

TechniqueRec defines the information which is returned for each technique in the reply to a **Query-Techniques** protocol request.

XieTypTechniqueRec

[	*needs-parameters*	BOOL
	group	XieTypTechniqueGroup
	number	CARD16
	speed	CARD8
	name	STRING8
]		

Needs-parameters indicates that a technique requires additional parameters; or if false, the technique either takes no parameters, or its parameters are optional. *Group* identifies which group the technique belongs to. *Number* is the numeric identifier assigned to the technique *. *Speed* represents the server's view of the speed of this technique relative to the other techniques in the same group (0 is the slowest, and 255 is the fastest). *Name* is the XIE compliant technique name string.

* Numbers assigned to standard techniques are encoded in Appendix C.

XieTypTile

Tile defines a source data tile for the **PasteUp** element.

XieTypTile

[	*src*	XieTypPhototag
	dst-x	INT32
	dst-y	INT32
]		

Src is the **Phototag** of the element supplying source data. *Dst_x, dst_y* specify the coordinates where the *tile* belongs in the output data.

XieTypTripletof*type*

Tripletof is a generic type used to define a fixed sized array of 3 items* of the specified *type*.

XieTypTripletof [v0, v1, v2 : (of *type*)]

* **Tripletof** is usually used to describe per-band attributes and parameters. When dealing with ***SingleBand*** images, the value zero must be used for the nonexistent bands.

XieTypWhiteAdjustTechnique

WhiteAdjustTechnique defines the white point adjustment techniques which can be used when converting to or from the RGB colorspace.

XieTypWhiteAdjustTechnique	{ XieValDefault, XieValWhiteAdjust_None, XieValWhiteAdjust_CIELabShift }

* ***Default***	
WhiteAdjust_None	"NONE"
WhiteAdjust_CIELabShift	"CIELAB-SHIFT"
private-technique-number	private-name-string

* The server is required to support the ***Default*** technique which is bound to one of the standard techniques defined above or a private technique.

almost blank