The **OpenVX™** XML Schema Extension

Version 1.0 (Provisional)

Document Revision: r29754
Generated on Wed Feb 4 2015 10:42:03

Khronos Vision Working Group

Editor: Erik Rainey
Editor: Jesse Villarreal

Contents

Chapter 1

XML Schema Extension

1.1 Purpose

The purpose of this extension is to create and standardize a description of an OpenVX Context (a set of graphs and their related data objects) in XML format.

1.2 Motivation

The intent is to standardize a representation of the OpenVX Context with an XML Schema and to standardize on an API. Having a standardized Schema means that:

- Some amount of validation for OpenVX Graph Verification may happen at *Import* time.
- Graphs and data may now be platform *portable*.
- The XML may be parsed or created by external tools for a variety of uses:
  - Documentation
  - Standards Compliance
  - Language Portability

1.3 Schema

The XML Schema is included below, and is available at the following url:

- https://www.khronos.org/registry/vx/schema/openvx-1-0.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- xmlns = The default location to find unknown elem/attr -->
<!-- xmlns:vx = A specific name for our namespace (same as default) -->
<!-- xmlns:xs = The location of the "schema" (xs) namespace -->
<!-- targetNamespace = When we define elem/attr here they go into this
      namespace -->
<!-- Elements have to be fully qualified in their namespaces -->
<!-- Attributes do not have to be fully qualified in their namespaces -->
<xs:schema xmlns="https://www.khronos.org/registry/vx/schema"
       targetNamespace="https://www.khronos.org/registry/vx/schema"
       xmlns:vx="https://www.khronos.org/registry/vx/schema" xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
    <!-- TYPEDEFS -->
    <xs:attributeGroup name="referrable">
        <xs:annotation>
            <xs:documentation> All referrable OpenVX objects have a reference
      attribute.
            </xs:documentation>
```

```
    </xs:annotation>
    <xs:attribute name="reference" type="xs:nonNegativeInteger" use="requir
  ed" />
    <xs:attribute name="name" type="xs:string" use="optional" />
</xs:attributeGroup>
<xs:attributeGroup name="indexable">
    <xs:attribute name="index" type="xs:nonNegativeInteger" use="required"
  />
</xs:attributeGroup>
<xs:complexType name="openvx.type">
    <xs:sequence>
        <xs:sequence>
            <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element ref="library" minOccurs="0" maxOccurs="unbounde
  d" />
                <xs:element ref="struct" minOccurs="0" maxOccurs="unbounded
  " />
            </xs:choice>
        </xs:sequence>
        <xs:sequence>
            <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element ref="graph" minOccurs="1" maxOccurs="unbounded"
   />
                <xs:element ref="scalar" minOccurs="1" maxOccurs="unbounded
  " />
                <xs:element ref="array" minOccurs="0" maxOccurs="unbounded"
   />
                <xs:element ref="image" minOccurs="0" maxOccurs="unbounded"
   />
                <xs:element ref="lut" minOccurs="0" maxOccurs="unbounded" /
  >
                <xs:element ref="matrix" minOccurs="0" maxOccurs="unbounded
  " />
                <xs:element ref="delay" minOccurs="0" maxOccurs="unbounded"
   />
                <xs:element ref="distribution" minOccurs="0" maxOccurs="unb
  ounded" />
                <xs:element ref="convolution" minOccurs="0" maxOccurs="unbo
  unded" />
                <xs:element ref="remap" minOccurs="0" maxOccurs="unbounded"
   />
                <xs:element ref="pyramid" minOccurs="0" maxOccurs="unbounde
  d" />
                <xs:element ref="threshold" minOccurs="0" maxOccurs="unboun
  ded" />
            </xs:choice>
        </xs:sequence>
    </xs:sequence>
    <xs:attribute name="references" type="xs:positiveInteger" use="required
  ">
        <xs:annotation>
            <xs:documentation> The Overall OpenVX tag indicates how many
  references are contained inside. This counts all data types plus graphs plus nodes
  .
                           All reference values in this document shall be
  bound from 0 to "references"-1 (any order).
            </xs:documentation>
        </xs:annotation>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="graph.parameter.type">
    <xs:attribute name="node" type="xs:nonNegativeInteger" use="required">
        <xs:annotation>
            <xs:documentation> This is the reference of the node which
  contains the parameter.
            </xs:documentation>
        </xs:annotation>
    </xs:attribute>
    <xs:attribute name="parameter" type="xs:nonNegativeInteger" use="requir
  ed">
        <xs:annotation>
            <xs:documentation> This is the index of the parameter on the
  node.
            </xs:documentation>
        </xs:annotation>
    </xs:attribute>
    <xs:attribute name="index" type="xs:nonNegativeInteger" use="required">
        <xs:annotation>
            <xs:documentation> This is the index of the graph parameter as
  it relates to the graph, not the node parameter.
            </xs:documentation>
        </xs:annotation>
    </xs:attribute>
</xs:complexType>
<xs:complexType name="graph.type">
    <xs:sequence>
        <xs:element ref="node" minOccurs="1" maxOccurs="unbounded" />
```

```
        <xs:element name="parameter" type="graph.parameter.type" minOccurs=
  "0" maxOccurs="unbounded" />
        <!-- These are the virtual objects which are tied to the graph -->
        <xs:sequence>
            <xs:choice maxOccurs="unbounded">
                <xs:element name="image" type="virtual.image.type" minOccur
  s="0" maxOccurs="unbounded" />
                <xs:element name="array" type="virtual.array.type" minOccur
  s="0" maxOccurs="unbounded" />
                <xs:element name="pyramid" type="virtual.pyramid.type" minO
  ccurs="0" maxOccurs="unbounded" />
            </xs:choice>
        </xs:sequence>
    </xs:sequence>
    <xs:attribute name="nodes" type="xs:positiveInteger" use="optional">
        <xs:annotation>
            <xs:documentation> Graphs may optional hint about how many
  nodes are listed inside.
            </xs:documentation>
        </xs:annotation>
    </xs:attribute>
    <xs:attributeGroup ref="referrable" />
</xs:complexType>
<xs:complexType name="node.type">
    <xs:sequence>
        <xs:element name="kernel" type="kernel.type" minOccurs="1" maxOccur
  s="1" />
        <xs:element name="parameter" type="parameter.type" minOccurs="1" ma
  xOccurs="10" />
    </xs:sequence>
    <xs:attribute name="bordermode" type="bordermode.type" use="optional" d
  efault="UNDEFINED" />
    <xs:attributeGroup ref="referrable" />
</xs:complexType>
<xs:complexType name="parameter.type">
    <xs:attribute name="index" type="index.type" use="required" />
    <xs:attribute name="reference" type="xs:nonNegativeInteger" use="requir
  ed" />
</xs:complexType>
<xs:simpleType name="kernel.type">
    <xs:annotation>
        <xs:documentation> Kernel Names must match a "dotted hierarchy"
  pattern of at least 1 level. Example: org.khronos.vision.sobel3x3 - pass Example:
            my.kernelname - pass Example: kernelname - fail
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:pattern value="[0-9A-Za-z_-]+(\.[0-9A-Za-z_-]+)+" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="bordermode.type">
    <xs:restriction base="xs:string">
        <xs:enumeration value="UNDEFINED" />
        <xs:enumeration value="CONSTANT" />
        <xs:enumeration value="REPLICATE" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="df_image.type">
    <xs:restriction base="xs:string">
        <xs:enumeration value="VIRT" />
        <xs:enumeration value="RGB2" />
        <xs:enumeration value="RGBA" />
        <xs:enumeration value="NV12" />
        <xs:enumeration value="NV21" />
        <xs:enumeration value="UYVY" />
        <xs:enumeration value="YUYV" />
        <xs:enumeration value="IYUV" />
        <xs:enumeration value="YUV4" />
        <xs:enumeration value="U008" />
        <xs:enumeration value="U016" />
        <xs:enumeration value="S016" />
        <xs:enumeration value="U032" />
        <xs:enumeration value="S032" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="df_image.list.type">
    <xs:list itemType="df_image.type" />
</xs:simpleType>
<xs:simpleType name="colorspace.type">
    <xs:restriction base="xs:string">
        <xs:enumeration value="BT601_525" />
        <xs:enumeration value="BT601_625" />
        <xs:enumeration value="BT709" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="colorrange.type">
    <xs:restriction base="xs:string">
```

```
        <xs:enumeration value="FULL" />
        <xs:enumeration value="RESTRICTED" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="invalid.type">
    <xs:restriction base="xs:string">
        <xs:enumeration value="VX_TYPE_INVALID" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="atomic.type">
    <xs:restriction base="xs:string">
        <xs:enumeration value="VX_TYPE_CHAR" />
        <xs:enumeration value="VX_TYPE_UINT8" />
        <xs:enumeration value="VX_TYPE_INT8" />
        <xs:enumeration value="VX_TYPE_UINT16" />
        <xs:enumeration value="VX_TYPE_INT16" />
        <xs:enumeration value="VX_TYPE_UINT32" />
        <xs:enumeration value="VX_TYPE_INT32" />
        <xs:enumeration value="VX_TYPE_UINT64" />
        <xs:enumeration value="VX_TYPE_INT64" />
        <xs:enumeration value="VX_TYPE_FLOAT32" />
        <xs:enumeration value="VX_TYPE_FLOAT64" />
        <xs:enumeration value="VX_TYPE_ENUM" />
        <xs:enumeration value="VX_TYPE_SIZE" />
        <xs:enumeration value="VX_TYPE_DF_IMAGE" />
        <xs:enumeration value="VX_TYPE_BOOL" />
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="scalar.type">
    <xs:choice>
        <xs:element name="char" type="char.type" minOccurs="0" maxOccurs="1
  " />
        <xs:element name="uint8" type="uint8.type" minOccurs="0" maxOccurs=
  "1" />
        <xs:element name="uint16" type="uint16.type" minOccurs="0" maxOccur
  s="1" />
        <xs:element name="uint32" type="uint32.type" minOccurs="0" maxOccur
  s="1" />
        <xs:element name="uint64" type="uint64.type" minOccurs="0" maxOccur
  s="1" />
        <xs:element name="int8" type="int8.type" minOccurs="0" maxOccurs="1
  " />
        <xs:element name="int16" type="int16.type" minOccurs="0" maxOccurs=
  "1" />
        <xs:element name="int32" type="int32.type" minOccurs="0" maxOccurs=
  "1" />
        <xs:element name="int64" type="int64.type" minOccurs="0" maxOccurs=
  "1" />
        <xs:element name="float32" type="float32.type" minOccurs="0" maxOcc
  urs="1" />
        <xs:element name="float64" type="float64.type" minOccurs="0" maxOcc
  urs="1" />
        <xs:element name="enum" type="int32.type" minOccurs="0" maxOccurs="
  1" />
        <xs:element name="df_image" type="df_image.type" minOccurs="0" maxO
  ccurs="1" />
        <xs:element name="size" type="uint64.type" minOccurs="0" maxOccurs=
  "1" />
        <xs:element name="bool" type="xs:boolean" minOccurs="0" maxOccurs="
  1" />
    </xs:choice>
    <xs:attribute name="elemType" type="atomic.type" />
    <xs:attributeGroup ref="referrable" />
</xs:complexType>
<xs:simpleType name="struct.type">
    <xs:restriction base="xs:string">
        <xs:enumeration value="VX_TYPE_KEYPOINT" />
        <xs:enumeration value="VX_TYPE_RECTANGLE" />
        <xs:enumeration value="VX_TYPE_COORDINATES2D" />
        <xs:enumeration value="VX_TYPE_COORDINATES3D" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="object.type">
    <xs:restriction base="xs:string">
        <xs:enumeration value="VX_TYPE_ARRAY" />
        <xs:enumeration value="VX_TYPE_IMAGE" />
        <xs:enumeration value="VX_TYPE_LUT" />
        <xs:enumeration value="VX_TYPE_MATRIX" />
        <xs:enumeration value="VX_TYPE_DELAY" />
        <xs:enumeration value="VX_TYPE_DISTRIBUTION" />
        <xs:enumeration value="VX_TYPE_CONVOLUTION" />
        <xs:enumeration value="VX_TYPE_SCALAR" />
        <xs:enumeration value="VX_TYPE_THRESHOLD" />
        <xs:enumeration value="VX_TYPE_PYRAMID" />
        <xs:enumeration value="VX_TYPE_REMAP" />
    </xs:restriction>
</xs:simpleType>
```

```
<xs:simpleType name="index.type">
    <xs:restriction base="xs:nonNegativeInteger">
        <xs:maxExclusive value="10" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="hex.type">
    <xs:restriction base="xs:string">
        <xs:pattern value="([a-fA-F0-9][a-fA-F0-9])+" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="char.type">
    <xs:restriction base="xs:string">
        <xs:length value="1" />
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="int8.type">
    <xs:simpleContent>
        <xs:extension base="xs:byte" />
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="uint8.type">
    <xs:simpleContent>
        <xs:extension base="xs:unsignedByte" />
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="int16.type">
    <xs:simpleContent>
        <xs:extension base="xs:short" />
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="uint16.type">
    <xs:simpleContent>
        <xs:extension base="xs:unsignedShort" />
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="int32.type">
    <xs:simpleContent>
        <xs:extension base="xs:int" />
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="uint32.type">
    <xs:simpleContent>
        <xs:extension base="xs:unsignedInt" />
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="int64.type">
    <xs:simpleContent>
        <xs:extension base="xs:long" />
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="uint64.type">
    <xs:simpleContent>
        <xs:extension base="xs:unsignedLong" />
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="float32.type">
    <xs:simpleContent>
        <xs:extension base="xs:float" />
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="float64.type">
    <xs:simpleContent>
        <xs:extension base="xs:double" />
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="enum.type">
    <xs:simpleContent>
        <xs:extension base="xs:int" />
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="size.type">
    <xs:simpleContent>
        <xs:extension base="xs:unsignedLong" />
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="bool.type">
    <xs:simpleContent>
        <xs:extension base="xs:boolean" />
    </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="user.struct.type">
    <xs:restriction base="xs:string">
        <xs:pattern value="USER_STRUCT_[0-9]+" />
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="user.struct.type.ext">
    <xs:simpleContent>
```

```
        <xs:extension base="user.struct.type">
            <xs:attribute name="size" type="xs:positiveInteger" use="requir
  ed" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="array.type.type">
    <xs:union memberTypes="atomic.type struct.type user.struct.type" />
</xs:simpleType>
<xs:simpleType name="virtual.array.type.type">
    <xs:union memberTypes="invalid.type array.type.type" />
</xs:simpleType>
<xs:complexType name="array.type">
    <xs:choice>
        <xs:element name="char" type="xs:string" minOccurs="0" maxOccurs="1
  " />
        <xs:element name="uint8" type="uint8.list.type" minOccurs="0" maxOc
  curs="1" />
        <xs:element name="uint16" type="uint16.list.type" minOccurs="0" max
  Occurs="1" />
        <xs:element name="uint32" type="uint32.list.type" minOccurs="0" max
  Occurs="1" />
        <xs:element name="uint64" type="uint64.list.type" minOccurs="0" max
  Occurs="1" />
        <xs:element name="int8" type="int8.list.type" minOccurs="0" maxOccu
  rs="1" />
        <xs:element name="int16" type="int16.list.type" minOccurs="0" maxOc
  curs="1" />
        <xs:element name="int32" type="int32.list.type" minOccurs="0" maxOc
  curs="1" />
        <xs:element name="int64" type="int64.list.type" minOccurs="0" maxOc
  curs="1" />
        <xs:element name="float32" type="float32.list.type" minOccurs="0" m
  axOccurs="1" />
        <xs:element name="float64" type="float64.list.type" minOccurs="0" m
  axOccurs="1" />
        <xs:element name="enum" type="int32.list.type" minOccurs="0" maxOcc
  urs="1" />
        <xs:element name="df_image" type="df_image.list.type" minOccurs="0"
   maxOccurs="1" />
        <xs:element name="bool" type="bool.list.type" minOccurs="0" maxOccu
  rs="1" />
        <xs:element name="size" type="uint64.list.type" minOccurs="0" maxOc
  curs="1" />
        <xs:element name="keypoint" type="keypoint.type" minOccurs="0" maxO
  ccurs="unbounded" />
        <xs:element name="coordinates2d" type="coordinates2d.type" minOccur
  s="0" maxOccurs="unbounded" />
        <xs:element name="coordinates3d" type="coordinates3d.type" minOccur
  s="0" maxOccurs="unbounded" />
        <xs:element name="rectangle" type="rectangle.type" minOccurs="0" ma
  xOccurs="unbounded" />
        <xs:element name="user" type="uint8.list.type" minOccurs="0" maxOcc
  urs="unbounded" />
    </xs:choice>
    <xs:attribute name="capacity" type="xs:positiveInteger" use="required"
  />
    <xs:attribute name="elemType" type="array.type.type" use="required" />
    <xs:attributeGroup ref="referrable" />
</xs:complexType>
<xs:complexType name="virtual.array.type">
    <xs:attribute name="capacity" type="xs:nonNegativeInteger" default="0"
  />
    <xs:attribute name="elemType" type="virtual.array.type.type" default="V
  X_TYPE_INVALID" />
    <xs:attributeGroup ref="referrable" />
</xs:complexType>
<xs:simpleType name="planeindex.type">
    <xs:restriction base="xs:unsignedInt">
        <xs:maxExclusive value="4" />
    </xs:restriction>
</xs:simpleType>
<xs:attributeGroup name="image_addressable">
    <xs:attribute name="x" type="xs:unsignedInt" use="required" />
    <xs:attribute name="y" type="xs:unsignedInt" use="required" />
</xs:attributeGroup>
<xs:complexType name="image.uint8.type">
    <xs:simpleContent>
        <xs:extension base="xs:unsignedByte">
            <xs:attributeGroup ref="image_addressable" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="image.uint16.type">
    <xs:simpleContent>
        <xs:extension base="xs:unsignedShort">
            <xs:attributeGroup ref="image_addressable" />
```

```
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="image.uint32.type">
    <xs:simpleContent>
        <xs:extension base="xs:unsignedInt">
            <xs:attributeGroup ref="image_addressable" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="image.int16.type">
    <xs:simpleContent>
        <xs:extension base="xs:short">
            <xs:attributeGroup ref="image_addressable" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="image.int32.type">
    <xs:simpleContent>
        <xs:extension base="xs:int">
            <xs:attributeGroup ref="image_addressable" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="bool.list.type">
    <xs:list itemType="xs:boolean" />
</xs:simpleType>
<xs:simpleType name="uint8.list.type">
    <xs:list itemType="xs:unsignedByte" />
</xs:simpleType>
<xs:simpleType name="uint16.list.type">
    <xs:list itemType="xs:unsignedShort" />
</xs:simpleType>
<xs:simpleType name="uint32.list.type">
    <xs:list itemType="xs:unsignedInt" />
</xs:simpleType>
<xs:simpleType name="uint64.list.type">
    <xs:list itemType="xs:unsignedLong" />
</xs:simpleType>
<xs:simpleType name="int8.list.type">
    <xs:list itemType="xs:byte" />
</xs:simpleType>
<xs:simpleType name="int16.list.type">
    <xs:list itemType="xs:short" />
</xs:simpleType>
<xs:simpleType name="int32.list.type">
    <xs:list itemType="xs:int" />
</xs:simpleType>
<xs:simpleType name="int64.list.type">
    <xs:list itemType="xs:long" />
</xs:simpleType>
<xs:simpleType name="float32.list.type">
    <xs:list itemType="xs:float" />
</xs:simpleType>
<xs:simpleType name="float64.list.type">
    <xs:list itemType="xs:double" />
</xs:simpleType>
<xs:simpleType name="tuple.2.uint8.type">
    <xs:restriction base="uint8.list.type">
        <xs:length value="2" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tuple.3.uint8.type">
    <xs:restriction base="uint8.list.type">
        <xs:length value="3" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tuple.4.uint8.type">
    <xs:restriction base="uint8.list.type">
        <xs:length value="4" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tuple.4.uint32.type">
    <xs:restriction base="uint32.list.type">
        <xs:length value="4" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tuple.2.int32.type">
    <xs:restriction base="int32.list.type">
        <xs:length value="2" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tuple.3.int32.type">
    <xs:restriction base="int32.list.type">
        <xs:length value="3" />
    </xs:restriction>
</xs:simpleType>
```

```
<xs:simpleType name="tuple.2.uint32.type">
    <xs:restriction base="uint32.list.type">
        <xs:length value="3" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tuple.3.uint32.type">
    <xs:restriction base="uint32.list.type">
        <xs:length value="3" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="hex.rgb.string.type">
    <xs:restriction base="xs:string">
        <xs:pattern value="\#[0-9A-Fa-f]{6}" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="rgb.string.type">
    <xs:union memberTypes="hex.rgb.string.type tuple.3.uint8.type" />
</xs:simpleType>
<xs:simpleType name="hex.rgba.string.type">
    <xs:restriction base="xs:string">
        <xs:pattern value="\#[0-9A-Fa-f]{8}" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="rgba.string.type">
    <xs:union memberTypes="hex.rgba.string.type tuple.4.uint8.type" />
</xs:simpleType>
<xs:complexType name="image.rgb.type">
    <xs:simpleContent>
        <xs:extension base="rgb.string.type">
            <xs:attributeGroup ref="image_addressable" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="image.rgba.type">
    <xs:simpleContent>
        <xs:extension base="rgba.string.type">
            <xs:attributeGroup ref="image_addressable" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="image.422.type">
    <xs:simpleContent>
        <xs:extension base="tuple.2.uint8.type">
            <xs:attributeGroup ref="image_addressable" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="rectangle.type">
    <xs:sequence>
        <xs:element name="start_x" type="xs:unsignedInt" minOccurs="0" maxO
  ccurs="1" />
        <xs:element name="start_y" type="xs:unsignedInt" minOccurs="0" maxO
  ccurs="1" />
        <xs:element name="end_x" type="xs:unsignedInt" minOccurs="0" maxOcc
  urs="1" />
        <xs:element name="end_y" type="xs:unsignedInt" minOccurs="0" maxOcc
  urs="1" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="coordinates2d.type">
    <xs:sequence>
        <xs:element name="x" type="xs:unsignedInt" minOccurs="1" maxOccurs=
  "1" />
        <xs:element name="y" type="xs:unsignedInt" minOccurs="1" maxOccurs=
  "1" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="coordinates3d.type">
    <xs:sequence>
        <xs:element name="x" type="xs:unsignedInt" minOccurs="1" maxOccurs=
  "1" />
        <xs:element name="y" type="xs:unsignedInt" minOccurs="1" maxOccurs=
  "1" />
        <xs:element name="z" type="xs:unsignedInt" minOccurs="1" maxOccurs=
  "1" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="image.pixel.type">
    <xs:choice>
        <xs:element name="uint8" type="image.uint8.type" minOccurs="0" maxO
  ccurs="unbounded" />
        <xs:element name="uint16" type="image.uint16.type" minOccurs="0" ma
  xOccurs="unbounded" />
        <xs:element name="uint32" type="image.uint32.type" minOccurs="0" ma
  xOccurs="unbounded" />
        <xs:element name="int16" type="image.int16.type" minOccurs="0" maxO
  ccurs="unbounded" />
```

```
        <xs:element name="int32" type="image.int32.type" minOccurs="0" maxO
  ccurs="unbounded" />
        <xs:element name="rgb" type="image.rgb.type" minOccurs="0" maxOccur
  s="unbounded" />
        <xs:element name="rgba" type="image.rgba.type" minOccurs="0" maxOcc
  urs="unbounded" />
        <xs:element name="yuv" type="image.422.type" minOccurs="0" maxOccur
  s="unbounded" />
      </xs:choice>
  </xs:complexType>
  <xs:complexType name="image.rectangle.type">
      <xs:complexContent>
          <xs:extension base="rectangle.type">
              <xs:sequence>
                  <xs:element name="pixels" type="image.pixel.type" minOccurs
  ="1" maxOccurs="unbounded" >
                      <xs:unique name="image_pixels_key">
                          <xs:selector xpath=".//
  vx:uint8|.//vx:uint16|.//vx:uint32|.//vx:int16|.//vx:int32|.//vx:rgb|.//vx:rgba|.//vx:yuv" />
                          <xs:field xpath="@x" />
                          <xs:field xpath="@y" />
                      </xs:unique>
                  </xs:element>
              </xs:sequence>
              <xs:attribute name="plane" type="planeindex.type" use="required
  " />
          </xs:extension>
      </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="image.type">
      <xs:sequence>
          <xs:element name="rectangle" type="image.rectangle.type" minOccurs=
  "0" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute name="width" type="xs:positiveInteger" default="320" />
      <xs:attribute name="height" type="xs:positiveInteger" default="240" />
      <xs:attribute name="format" type="df_image.type" default="U008" />
      <xs:attribute name="colorspace" type="colorspace.type" use="optional" d
  efault="BT709" />
      <xs:attribute name="colorrange" type="colorrange.type" use="optional" d
  efault="FULL" />
      <xs:attributeGroup ref="referrable" />
  </xs:complexType>
  <xs:complexType name="virtual.image.type">
      <xs:attribute name="width" type="xs:nonNegativeInteger" default="0" />
      <xs:attribute name="height" type="xs:nonNegativeInteger" default="0" />
      <xs:attribute name="format" type="df_image.type" default="VIRT" />
      <xs:attributeGroup ref="referrable" />
  </xs:complexType>
  <xs:complexType name="pyramid.image.type">
      <xs:complexContent>
          <xs:extension base="image.type">
              <xs:attribute name="level" type="xs:nonNegativeInteger" />
          </xs:extension>
      </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="pyramid.type">
      <xs:complexContent>
          <xs:extension base="image.type">
              <xs:sequence>
                  <xs:element name="image" type="pyramid.image.type" minOccur
  s="0" maxOccurs="unbounded" />
              </xs:sequence>
              <xs:attribute name="levels" type="xs:positiveInteger" default="
  4" />
              <xs:attribute name="scale" type="xs:float" default="0.5" />
          </xs:extension>
      </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="virtual.pyramid.type">
      <xs:complexContent>
          <xs:extension base="virtual.image.type">
              <xs:attribute name="levels" type="xs:positiveInteger" default="
  4" />
              <xs:attribute name="scale" type="xs:float" default="0.5" />
          </xs:extension>
      </xs:complexContent>
  </xs:complexType>
  <xs:simpleType name="lut.count.type">
      <xs:restriction base="xs:positiveInteger">
          <xs:minInclusive value="1" />
          <xs:maxInclusive value="256" />
      </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="indexed.uint8.type">
      <xs:complexContent>
          <xs:extension base="uint8.type">
```

```
            <xs:attributeGroup ref="indexable" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="lut.type">
    <xs:sequence>
        <xs:element name="uint8" type="indexed.uint8.type" minOccurs="0" ma
  xOccurs="256" />
    </xs:sequence>
    <xs:attribute name="count" type="lut.count.type" default="256" />
    <xs:attribute name="elemType" type="atomic.type" default="VX_TYPE_UINT8
  " />
    <xs:attributeGroup ref="referrable" />
</xs:complexType>
<xs:complexType name="mat.int32.type">
    <xs:simpleContent>
        <xs:extension base="xs:int">
            <xs:attribute name="row" type="xs:nonNegativeInteger" use="requ
  ired" />
            <xs:attribute name="column" type="xs:nonNegativeInteger" use="r
  equired" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="mat.f32.type">
    <xs:simpleContent>
        <xs:extension base="xs:float">
            <xs:attribute name="row" type="xs:nonNegativeInteger" use="requ
  ired" />
            <xs:attribute name="column" type="xs:nonNegativeInteger" use="r
  equired" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="matrix.type">
    <xs:choice>
        <xs:element name="int32" type="mat.int32.type" minOccurs="0" maxOcc
  urs="unbounded" />
        <xs:element name="float32" type="mat.f32.type" minOccurs="0" maxOcc
  urs="unbounded" />
    </xs:choice>
    <xs:attribute name="rows" type="xs:positiveInteger" default="1" />
    <xs:attribute name="columns" type="xs:positiveInteger" default="1" />
    <xs:attribute name="elemType" default="VX_TYPE_FLOAT32">
        <xs:simpleType>
            <xs:restriction base="atomic.type">
                <xs:enumeration value="VX_TYPE_INT32" />
                <xs:enumeration value="VX_TYPE_FLOAT32" />
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attributeGroup ref="referrable" />
</xs:complexType>
<xs:complexType name="delay.type">
    <xs:choice>
        <xs:element ref="image" maxOccurs="unbounded" />
        <xs:element ref="array" maxOccurs="unbounded" />
        <xs:element ref="lut" maxOccurs="unbounded" />
        <xs:element ref="matrix" maxOccurs="unbounded" />
        <xs:element ref="distribution" maxOccurs="unbounded" />
        <xs:element ref="convolution" maxOccurs="unbounded" />
        <xs:element ref="pyramid" maxOccurs="unbounded" />
        <xs:element ref="threshold" maxOccurs="unbounded" />
        <xs:element ref="remap" maxOccurs="unbounded" />
        <xs:element ref="scalar" maxOccurs="unbounded" />
    </xs:choice>
    <xs:attribute name="objType">
        <xs:simpleType>
            <xs:restriction base="object.type">
                <!-- delays can not contain delays -->
                <xs:enumeration value="VX_TYPE_ARRAY" />
                <xs:enumeration value="VX_TYPE_IMAGE" />
                <xs:enumeration value="VX_TYPE_LUT" />
                <xs:enumeration value="VX_TYPE_MATRIX" />
                <xs:enumeration value="VX_TYPE_DISTRIBUTION" />
                <xs:enumeration value="VX_TYPE_CONVOLUTION" />
                <xs:enumeration value="VX_TYPE_PYRAMID" />
                <xs:enumeration value="VX_TYPE_THRESHOLD" />
                <xs:enumeration value="VX_TYPE_SCALAR" />
                <xs:enumeration value="VX_TYPE_REMAP" />
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="count" type="xs:positiveInteger" use="required" />
    <xs:attributeGroup ref="referrable" />
</xs:complexType>
<xs:complexType name="freq.type">
```

```
    <xs:simpleContent>
        <xs:extension base="xs:unsignedInt">
            <xs:attribute name="bin" type="xs:nonNegativeInteger" use="requ
  ired" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="distribution.type">
    <xs:sequence>
        <xs:element name="frequency" type="freq.type" minOccurs="0" maxOccu
  rs="unbounded" />
    </xs:sequence>
    <xs:attribute name="bins" type="xs:positiveInteger" default="16" />
    <xs:attribute name="offset" type="xs:nonNegativeInteger" default="0" />
    <xs:attribute name="range" type="xs:positiveInteger" default="256" />
    <xs:attributeGroup ref="referrable" />
</xs:complexType>
<xs:complexType name="coeff16.type">
    <xs:simpleContent>
        <xs:extension base="xs:short">
            <xs:attribute name="row" type="xs:nonNegativeInteger" use="requ
  ired" />
            <xs:attribute name="column" type="xs:nonNegativeInteger" use="r
  equired" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="convolution.type">
    <xs:sequence>
        <xs:element name="int16" type="coeff16.type" minOccurs="0" maxOccur
  s="unbounded" />
    </xs:sequence>
    <xs:attribute name="rows" type="xs:positiveInteger" use="required" />
    <xs:attribute name="columns" type="xs:positiveInteger" use="required" /
  >
    <xs:attribute name="scale" type="xs:unsignedInt" default="1" />
    <xs:attributeGroup ref="referrable" />
</xs:complexType>
<xs:complexType name="remap_point.type">
    <xs:attribute name="src_x" type="xs:float" use="required" />
    <xs:attribute name="src_y" type="xs:float" use="required" />
    <xs:attribute name="dst_x" type="xs:unsignedInt" use="required" />
    <xs:attribute name="dst_y" type="xs:unsignedInt" use="required" />
</xs:complexType>
<xs:complexType name="remap.type">
    <xs:sequence>
        <xs:element name="point" type="remap_point.type" minOccurs="0" maxO
  ccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="src_width" type="xs:positiveInteger" use="required"
   />
    <xs:attribute name="src_height" type="xs:positiveInteger" use="required
  " />
    <xs:attribute name="dst_width" type="xs:positiveInteger" use="required"
   />
    <xs:attribute name="dst_height" type="xs:positiveInteger" use="required
  " />
    <xs:attributeGroup ref="referrable" />
</xs:complexType>
<xs:complexType name="threshold.type">
    <xs:choice>
        <xs:element name="binary" type="xs:unsignedByte" maxOccurs="1" />
        <xs:element name="range" maxOccurs="1">
            <xs:complexType>
                <xs:attribute name="lower" type="xs:unsignedByte" />
                <xs:attribute name="upper" type="xs:unsignedByte" />
            </xs:complexType>
        </xs:element>
    </xs:choice>
    <xs:attribute name="elemType" default="VX_TYPE_UINT8">
        <xs:simpleType>
            <xs:restriction base="atomic.type">
                <xs:enumeration value="VX_TYPE_UINT8" />
                <!-- further versions may have other types -->
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="true_value"  type="xs:unsignedByte" />
    <xs:attribute name="false_value" type="xs:unsignedByte" />
    <xs:attributeGroup ref="referrable" />
</xs:complexType>
<xs:complexType name="keypoint.type">
    <xs:sequence>
        <xs:element name="x" type="xs:nonNegativeInteger" minOccurs="1" max
  Occurs="1" />
        <xs:element name="y" type="xs:nonNegativeInteger" minOccurs="1" max
  Occurs="1" />
```

```
        <xs:element name="strength" type="xs:float" minOccurs="1" maxOccurs
  ="1" />
        <xs:element name="scale" type="xs:float" default="1.0" minOccurs="0
  " maxOccurs="1" />
        <xs:element name="orientation" type="xs:float" default="0.0" minOcc
  urs="0" maxOccurs="1" />
        <xs:element name="tracking_status" type="xs:nonNegativeInteger" def
  ault="1" minOccurs="0" maxOccurs="1" />
        <xs:element name="error" type="xs:float" default="0.0" minOccurs="0
  " maxOccurs="1" />
    </xs:sequence>
</xs:complexType>
<!-- ELEMENT DEFS AND KEYS -->
<xs:element name="openvx" type="openvx.type">
    <xs:key name="reference_key">
        <xs:selector
            xpath="./vx:graph|.//
  vx:node|./vx:scalar|.//vx:array|.//vx:image
  |./vx:distribution|./vx:delay|./vx:lut|.//vx:pyramid|./vx:threshold|./vx:matrix|./vx:convolution|./vx:remap" />
        <xs:field xpath="@reference" />
    </xs:key>
    <!-- Parameters must reference declared objects -->
    <xs:keyref name="parameter_keyref" refer="reference_key">
        <xs:selector xpath="./vx:graph/vx:node/vx:parameter" />
        <xs:field xpath="@reference" />
    </xs:keyref>
    <!-- Graph parameter node references must refer to nodes (ideally withi
  n the same graph ) -->
    <xs:keyref name="gp_node_keyref" refer="reference_key">
        <xs:selector xpath="./vx:graph/vx:parameter" />
        <xs:field xpath="@node" />
    </xs:keyref>
</xs:element>
<xs:element name="graph" type="graph.type">
    <!-- Graph parameter indexes must be unique, but only within a Graph --
  >
    <xs:key name="gp_index_key">
        <xs:selector xpath="./vx:parameter" />
        <xs:field xpath="@index" />
    </xs:key>
    <!-- Graph parameters must refer to node parameter indexes <xs:keyref n
  ame="gp_param_keyref" refer="parameter_id_key"> <xs:selector xpath="./vx:paramet
  er" /> <xs:field
        xpath="@parameter" /> </xs:keyref> -->
</xs:element>
<xs:element name="node" type="node.type">
    <xs:key name="parameter_id_key">
        <xs:selector xpath="./vx:parameter" />
        <xs:field xpath="@index" />
    </xs:key>
</xs:element>
<xs:element name="library" type="xs:string" />
<xs:element name="struct" type="user.struct.type.ext" />
<xs:element name="scalar" type="scalar.type" />
<xs:element name="array" type="array.type" />
<xs:element name="image" type="image.type" />
<xs:element name="lut" type="lut.type" />
<xs:element name="matrix" type="matrix.type">
    <xs:key name="matrix_key">
        <xs:selector xpath="./vx:int32|./vx:float32" />
        <xs:field xpath="@row" />
        <xs:field xpath="@column" />
    </xs:key>
</xs:element>
<xs:element name="delay" type="delay.type" />
<xs:element name="distribution" type="distribution.type">
    <xs:key name="freq_key">
        <xs:selector xpath="./vx:frequency" />
        <xs:field xpath="@bin" />
    </xs:key>
</xs:element>
<xs:element name="convolution" type="convolution.type">
    <xs:key name="convolution_key">
        <xs:selector xpath="./vx:coeff16" />
        <xs:field xpath="@row" />
        <xs:field xpath="@column" />
    </xs:key>
</xs:element>
<xs:element name="remap" type="remap.type">
    <xs:key name="remap_key">
        <xs:selector xpath="./vx:point" />
        <!-- Destintation Points must be unique -->
        <xs:field xpath="@dst_x" />
        <xs:field xpath="@dst_y" />
    </xs:key>
</xs:element>
<xs:element name="pyramid" type="pyramid.type" />
```

```
    <xs:element name="threshold" type="threshold.type" />
</xs:schema>
```

Chapter 2

XML Description

The XML Schema provides the necessary specification required to validate a OpenVX conformant XML file, but it alone is not sufficient to ensure full compatibility with OpenVX compliant implementations. Therefore, this section is intended supplement the schema by providing more details and examples of a conformant xml document.

The rest of this section will discuss each xml element in more detail. The examples contained in this section are a sample of a larger example xml file which is included in section XML Example

Each element described in this section will also reference the schema type name in order to help cross-reference the schema. For all elements which can be references, an optional "name" field can be used so that the import program can reference a reference by name.

2.1 OpenVX Element

Schema type: openvx.type

The 'openvx' tag encapsulates the entire OpenVX element. The 'references' attribute informs the parser as to the the number of OpenVX references that are contained in the xml file. Each data and graph object that corresponds to an OpenVX object will have a unique reference number in the xml file. The range of valid reference numbers should be between 0 and ["references"-1]. These references may appear in any order. Below is an example of the openvx element which indicates that the xml file has 124 objects enumerated.

```
<openvx xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns="https://www.khronos.org/registry/vx/schema"
        xsi:schemaLocation="https://www.khronos.org/registry/vx/schema
       openvx-1-0.xsd"
        references="124">
```

2.2 OpenVX Graph Objects

Schema type: graph.type

Graph elements each have their own reference number (and optional name), and may have child elements consisting of nodes, graph parameters, and virtual objects associated with the graph. The following example can be used as a reference for the discussion on graph objects.

```
<graph reference="40" name="GRAPH1" >
    <node reference="52">
        <kernel>org.khronos.openvx.sobel3x3</kernel>
        <parameter index="0" reference="0" />
        <parameter index="1" reference="41" />
        <parameter index="2" reference="42" />
    </node>
    <node reference="54">
        <kernel>org.khronos.openvx.convert_depth</kernel>
        <parameter index="0" reference="41" />
        <parameter index="1" reference="1" />
```

```
        <parameter index="2" reference="53" />
        <parameter index="3" reference="51" />
    </node>
    <parameter index="0" node="52" parameter="0" />
    <parameter index="1" node="54" parameter="1" />
    <image reference="41" width="0" height="0" format="VIRT" />
    <image reference="42" width="0" height="0" format="S016" />
    <image reference="43" width="320" height="240" format="VIRT" />
    <image reference="44" width="640" height="480" format="U008" />
    <pyramid reference="45" width="0" height="0" format="VIRT" scale="0.500000"
       levels="4" />
    <pyramid reference="46" width="640" height="480" format="VIRT" scale="
      0.500000" levels="4" />
    <pyramid reference="47" width="640" height="480" format="U008" scale="
      0.500000" levels="4" />
    <array reference="48" capacity="0" elemType="VX_TYPE_INVALID" />
    <array reference="49" capacity="0" elemType="VX_TYPE_KEYPOINT" />
    <array reference="50" capacity="1000" elemType="VX_TYPE_KEYPOINT" />
</graph>
```

2.2.1 Nodes

Schema type: node.type

Each node element of the graph has its own reference number (and optional name) and one child element specifying the kernel name. Each node element contains "n" number of parameter child elements, where "n" is equal to the number of parameters corresponding to the kernel specified. The index attribute of each parameter should monotonically increment. The reference attribute associated with each parameter should correspond to the reference number of the data object which is enumerated either as a virtual object reference within the graph, or elsewhere in the xml document.

2.2.2 Graph Parameters

Schema type: graph.parameter.type

Graph parameters do not have a reference number. The index attribute of each parameter should monotonically increment. The graph parameter identifies which parameter of which node is being exported as a graph parameter at the corresponding index.

2.2.3 Virtual Data Objects

Schema type: virtual.image.type, virtual.array.type, virtual.pyramid.type

Typically, data objects are enumerated outside of the graph element. However, virtual data objects (such as virtual arrays, virtual images, and virtual pyramids) are always listed as child elements of the graph element that they are associated with. Since virtual objects may not have accessible memory to the user, the xml does not specify the data of virtual objects. The format of the virtual data objects will be discussed further in OpenVX Data Objects.

2.3 OpenVX Data Objects

Each of the data objects specified in OpenVX can be represented in xml representation. For each data object, the minimum information required to create the data object in a context is required. Optionally, the actual data included in the data object may also be represented in child elements of the object element in the xml. This way, during xml import into a context, the import utility should create all listed data objects, and if data is also specified for any data objects, then this data should be initialized within the imported data object. During xml export, if a data object has been created but not yet written to, then it may not be necessary to export the uninitialized data into the xml file for the associated data object.

Since virtual objects (virtual arrays, virtual images, or virtual pyramids) may not have accessible memory to the user, the xml does not specify the data of virtual objects.

2.3.1 Array

Schema type: array.type

Each array element below is a valid example of a virtual arrays within a graph element:

```
<array reference="48" capacity="0" elemType="VX_TYPE_INVALID" />
<array reference="49" capacity="0" elemType="VX_TYPE_KEYPOINT" />
<array reference="50" capacity="1000" elemType="VX_TYPE_KEYPOINT" />
```

Each array element is a valid example of array objects. Please refer to the schema under array.type for all supported data types and formats of arrays:

```
<array reference="96" capacity="10" elemType="VX_TYPE_UINT8">
    <uint8>2 3 4 5 6 7 8 9 10 11 </uint8>
</array>
<array reference="97" capacity="20" elemType="VX_TYPE_CHAR">
    <char>a 13,.;^-</char>
</array>
<array reference="98" capacity="4" elemType="VX_TYPE_ENUM">
    <enum>-1 0 45057 </enum>
</array>
<array reference="99" capacity="4" elemType="VX_TYPE_DF_IMAGE">
    <df_image>RGB2 U008 VIRT </df_image>
</array>
<array reference="100" capacity="3" elemType="VX_TYPE_KEYPOINT">
    <keypoint>
        <x>0</x>
        <y>0</y>
        <strength>2.300000</strength>
        <scale>6.555550</scale>
        <orientation>0.905900</orientation>
        <tracking_status>5</tracking_status>
        <error>3.545500</error>
    </keypoint>
    <keypoint>
        <x>400</x>
        <y>235</y>
        <strength>5.222200</strength>
        <scale>1.221000</scale>
        <orientation>0.569500</orientation>
        <tracking_status>8</tracking_status>
        <error>462.500000</error>
    </keypoint>
</array>
<array reference="101" capacity="5" elemType="VX_TYPE_RECTANGLE">
    <rectangle>
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>640</end_x>
        <end_y>320</end_y>
    </rectangle>
    <rectangle>
        <start_x>65</start_x>
        <start_y>32</start_y>
        <end_x>128</end_x>
        <end_y>362</end_y>
    </rectangle>
</array>
<array reference="102" capacity="6" elemType="VX_TYPE_COORDINATES2D">
    <coordinates2d>
        <x>1</x>
        <y>2</y>
    </coordinates2d>
    <coordinates2d>
        <x>0</x>
        <y>55</y>
    </coordinates2d>
</array>
<array reference="103" capacity="6" elemType="VX_TYPE_COORDINATES3D">
    <coordinates3d>
        <x>1</x>
        <y>2</y>
        <z>3</z>
    </coordinates3d>
    <coordinates3d>
        <x>55</x>
        <y>66</y>
        <z>77</z>
    </coordinates3d>
</array>
<array reference="104" capacity="8" elemType="VX_TYPE_INT8">
    <int8>5 0 -3 -8 </int8>
```

```
</array>
<array reference="105" capacity="6" elemType="VX_TYPE_INT16">
    <int16>200 100 0 -100 -200 </int16>
</array>
<array reference="106" capacity="6" elemType="VX_TYPE_INT32">
    <int32>200000 100000 0 -100000 -200000 </int32>
</array>
<array reference="107" capacity="3" elemType="VX_TYPE_BOOL">
    <bool>true false true </bool>
</array>
<array reference="108" capacity="4" elemType="VX_TYPE_SIZE">
    <size>8000 24000 </size>
</array>
<array reference="109" capacity="2" elemType="VX_TYPE_FLOAT64">
    <float64>1235.255660 -563.256700 </float64>
</array>
<array reference="110" capacity="8" elemType="VX_TYPE_UINT64">
    <uint64>9000000000 8000000000 7000000000 6000000000 </uint64>
</array>
<array reference="111" capacity="6" elemType="VX_TYPE_UINT16">
    <uint16>290 100 0 100 260 </uint16>
</array>
<array reference="112" capacity="6" elemType="VX_TYPE_UINT32">
    <uint32>200000 100000 0 100000 200000 </uint32>
</array>
<array reference="113" capacity="2" elemType="VX_TYPE_FLOAT32">
    <float32>1235.255615 -563.256714 </float32>
</array>
<array reference="114" capacity="8" elemType="VX_TYPE_INT64">
    <int64>9000000000 8000000000 -7000000000 -6000000000 </int64>
</array>
<array reference="120" capacity="4" elemType="USER_STRUCT_0">
    <user>32 0 0 0 158 127 0 0 154 153 153 153 153 153 9 64 </user>
    <user>64 0 0 0 158 127 0 0 154 153 153 153 153 153 25 64 </user>
</array>
```

2.3.2 Convolution

Schema type: convolution.type

Below is a valid example of a convolution object:

```
<convolution reference="31" rows="3" columns="3" scale="16">
    <int16 row="0" column="0">-1</int16>
    <int16 row="0" column="1">0</int16>
    <int16 row="0" column="2">1</int16>
    <int16 row="1" column="0">-2</int16>
    <int16 row="1" column="1">0</int16>
    <int16 row="1" column="2">1</int16>
    <int16 row="2" column="0">-1</int16>
    <int16 row="2" column="1">0</int16>
    <int16 row="2" column="2">1</int16>
</convolution>
```

2.3.3 Delay

Schema type: delay.type

Delay onject types can be one of many of the other data types. For list of supported delay types, please refer to the schema under delay.type. Below is a valid example of a delay object:

```
<delay reference="55" count="3">
    <image reference="56" width="6" height="4" format="U008">
    </image>
    <image reference="57" width="6" height="4" format="U008">
    </image>
    <image reference="58" width="6" height="4" format="U008">
    </image>
</delay>
```

2.3.4 Distribution

Schema type: distribution.type

Below is a valid example of a distribution object:

```
<distribution reference="34" bins="16" offset="0" range="256">
    <frequency bin="0">0</frequency>
    <frequency bin="1">1</frequency>
    <frequency bin="2">2</frequency>
    <frequency bin="3">3</frequency>
    <frequency bin="4">4</frequency>
    <frequency bin="5">5</frequency>
    <frequency bin="6">6</frequency>
    <frequency bin="7">7</frequency>
    <frequency bin="8">8</frequency>
    <frequency bin="9">7</frequency>
    <frequency bin="10">6</frequency>
    <frequency bin="11">5</frequency>
    <frequency bin="12">4</frequency>
    <frequency bin="13">3</frequency>
    <frequency bin="14">2</frequency>
    <frequency bin="15">1</frequency>
</distribution>
```

2.3.5 Image

Schema type: image.type

Below are valid examples of image objects:

```
<image reference="0" width="6" height="4" format="U008">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <uint8 x="0" y="0">0</uint8>
            <uint8 x="1" y="0">1</uint8>
            <uint8 x="2" y="0">2</uint8>
            <uint8 x="3" y="0">3</uint8>
            <uint8 x="4" y="0">4</uint8>
            <uint8 x="5" y="0">5</uint8>
            <uint8 x="0" y="1">6</uint8>
            <uint8 x="1" y="1">7</uint8>
            <uint8 x="2" y="1">8</uint8>
            <uint8 x="3" y="1">9</uint8>
            <uint8 x="4" y="1">10</uint8>
            <uint8 x="5" y="1">11</uint8>
            <uint8 x="0" y="2">12</uint8>
            <uint8 x="1" y="2">13</uint8>
            <uint8 x="2" y="2">14</uint8>
            <uint8 x="3" y="2">15</uint8>
            <uint8 x="4" y="2">16</uint8>
            <uint8 x="5" y="2">17</uint8>
            <uint8 x="0" y="3">18</uint8>
            <uint8 x="1" y="3">19</uint8>
            <uint8 x="2" y="3">20</uint8>
            <uint8 x="3" y="3">21</uint8>
            <uint8 x="4" y="3">22</uint8>
            <uint8 x="5" y="3">23</uint8>
        </pixels>
    </rectangle>
</image>
<image reference="1" width="6" height="4" format="U008">
</image>
```

2.3.6 LUT

Schema type: lut.type

Below is a valid example of a lut object:

```
<lut reference="32" count="256" elemType="VX_TYPE_UINT8">
    <uint8 index="0">255</uint8>
    <uint8 index="1">0</uint8>
    <uint8 index="2">1</uint8>
    <uint8 index="3">2</uint8>
    <uint8 index="4">3</uint8>
    <uint8 index="5">4</uint8>
    <uint8 index="6">5</uint8>
    <uint8 index="7">6</uint8>
    <uint8 index="8">7</uint8>
```

```
    // skipping indicies 9-253 to save space

    <uint8 index="252">251</uint8>
    <uint8 index="253">252</uint8>
    <uint8 index="254">253</uint8>
    <uint8 index="255">254</uint8>
</lut>
```

2.3.7 Matrix

Schema type: matrix.type

Below is a valid example of a matrix object:

```
<matrix reference="14" elemType="VX_TYPE_FLOAT32" rows="3" columns="3">
    <float32 row="0" column="0">-3.141593</float32>
    <float32 row="0" column="1">0.000000</float32>
    <float32 row="0" column="2">3.141593</float32>
    <float32 row="1" column="0">-6.283185</float32>
    <float32 row="1" column="1">0.000000</float32>
    <float32 row="1" column="2">6.283185</float32>
    <float32 row="2" column="0">-3.141593</float32>
    <float32 row="2" column="1">0.000000</float32>
    <float32 row="2" column="2">3.141593</float32>
</matrix>
```

2.3.8 Pyramid

Schema type: pyramid.type

Below is a valid example of a pyramid object:

```
<pyramid reference="35" width="24" height="16" format="U008" scale="0.500000"
      levels="4">
    <image reference="36" width="24" height="16" format="U008">
    </image>
    <image reference="37" width="12" height="8" format="U008">
    </image>
    <image reference="38" width="6" height="4" format="U008">
    </image>
    <image reference="39" width="3" height="2" format="U008">
    </image>
</pyramid>
```

2.3.9 Remap

Schema type: remap.type

Below is a valid example of a remap object:

```
<remap reference="33" src_width="6" src_height="4" dst_width="6" dst_height="4"
      >
    <point src_x="0.500000" src_y="0.500000" dst_x="0" dst_y="0" />
    <point src_x="1.000000" src_y="0.500000" dst_x="1" dst_y="0" />
    <point src_x="1.500000" src_y="0.500000" dst_x="2" dst_y="0" />
    <point src_x="2.000000" src_y="0.500000" dst_x="3" dst_y="0" />
    <point src_x="2.500000" src_y="0.500000" dst_x="4" dst_y="0" />
    <point src_x="3.000000" src_y="0.500000" dst_x="5" dst_y="0" />
    <point src_x="0.500000" src_y="1.000000" dst_x="0" dst_y="1" />
    <point src_x="1.000000" src_y="1.000000" dst_x="1" dst_y="1" />
    <point src_x="1.500000" src_y="1.000000" dst_x="2" dst_y="1" />
    <point src_x="2.000000" src_y="1.000000" dst_x="3" dst_y="1" />
    <point src_x="2.500000" src_y="1.000000" dst_x="4" dst_y="1" />
    <point src_x="3.000000" src_y="1.000000" dst_x="5" dst_y="1" />
    <point src_x="0.500000" src_y="1.500000" dst_x="0" dst_y="2" />
    <point src_x="1.000000" src_y="1.500000" dst_x="1" dst_y="2" />
    <point src_x="1.500000" src_y="1.500000" dst_x="2" dst_y="2" />
    <point src_x="2.000000" src_y="1.500000" dst_x="3" dst_y="2" />
    <point src_x="2.500000" src_y="1.500000" dst_x="4" dst_y="2" />
    <point src_x="3.000000" src_y="1.500000" dst_x="5" dst_y="2" />
    <point src_x="0.500000" src_y="2.000000" dst_x="0" dst_y="3" />
    <point src_x="1.000000" src_y="2.000000" dst_x="1" dst_y="3" />
    <point src_x="1.500000" src_y="2.000000" dst_x="2" dst_y="3" />
```

```
    <point src_x="2.000000" src_y="2.000000" dst_x="3" dst_y="3" />
    <point src_x="2.500000" src_y="2.000000" dst_x="4" dst_y="3" />
    <point src_x="3.000000" src_y="2.000000" dst_x="5" dst_y="3" />
</remap>
```

2.3.10 Scalar

Schema type: scalar.type

Below are valid examples of scalar objects:

```
<scalar reference="16" elemType="VX_TYPE_CHAR">
    <char>z</char>
</scalar>
<scalar reference="17" elemType="VX_TYPE_UINT8">
    <uint8>255</uint8>
</scalar>
<scalar reference="18" elemType="VX_TYPE_INT8">
    <int8>-128</int8>
</scalar>
<scalar reference="19" elemType="VX_TYPE_FLOAT32">
    <float32>3.141593</float32>
</scalar>
<scalar reference="20" elemType="VX_TYPE_FLOAT64">
    <float64>6.283185</float64>
</scalar>
<scalar reference="21" elemType="VX_TYPE_DF_IMAGE">
    <df_image>NV12</df_image>
</scalar>
<scalar reference="22" elemType="VX_TYPE_UINT16">
    <uint16>65535</uint16>
</scalar>
<scalar reference="23" elemType="VX_TYPE_INT16">
    <int16>-32768</int16>
</scalar>
<scalar reference="24" elemType="VX_TYPE_UINT64">
    <uint64>18446744073709551615</uint64>
</scalar>
<scalar reference="25" elemType="VX_TYPE_INT64">
    <int64>-9223372036854775808</int64>
</scalar>
<scalar reference="26" elemType="VX_TYPE_UINT32">
    <uint32>4294967295</uint32>
</scalar>
<scalar reference="27" elemType="VX_TYPE_SIZE">
    <size>100</size>
</scalar>
<scalar reference="28" elemType="VX_TYPE_BOOL">
    <bool>true</bool>
</scalar>
```

2.3.11 Threshold

Schema type: threshold.type

Below is are valid examples of a threshold object:

```
<threshold reference="29" elemType="VX_TYPE_UINT8" true_value="0" false_value="
      0">
    <binary>240</binary>
</threshold>
<threshold reference="30" elemType="VX_TYPE_UINT8" true_value="0" false_value="
      0">
    <range lower="10" upper="240" />
</threshold>
```

2.4 User Defined Elements

The OpenVX XML Schema extension contains two additional optional elements which are intended to aid in portability of an OpenVX graph which is dependent on additional libraries or user structs.

2.4.1 Library

The 'library' element is simply a string type. If the OpenVX context is dependent on user kernels which may be defined in a separate library, then the name of this library can be listed here. This way, if the OpenVX implementation importing this XML file also has the library in the system, then it may know that it needs to load or link against the named library before parsing and loading the rest of the xml file. Because of this dependency, all library names should be listed in the xml file before any objects, as per the sequence constraint in the openvx.type of the schema. See an example below:

```
<library>openvx-debug</library>
```

2.4.2 Struct

The 'struct' element is intended to communicate to the parser that there is a user struct defined in the context. If there are array objects which have a user struct as an element type, then the array object of this type can not be created without knowing the size of the user struct. Therefore, this size needs to be communicated as part of the xml file as the "size" attribute in the struct element before any other objects are defined, as per the sequence constraint in the openvx.type of the schema. See an example below:

```
<struct size="16">USER_STRUCT_0</struct>
```

The schema expects the label used for user structs to always be "USER_STRUCT_$*$", where the wildcard can be any number. This way, multiple user structs can be exported/imported using unique suffix numbers. Now that this is defined at the top of the xml file, an array can reference the same label in the implementation:

```
<array reference="120" capacity="4" elemType="USER_STRUCT_0">
    <user>32 0 0 0 158 127 0 0 154 153 153 153 153 153 153 9 64 </user>
    <user>64 0 0 0 158 127 0 0 154 153 153 153 153 153 153 25 64 </user>
</array>
```

Chapter 3

XML Example

3.1 example.xml

The following is an OpenVX XML Schema compatible example xml file. It contains at least one example of each object type that the schema supports, and can be used as a reference to for programming compatible import and export utilities.

```
<?xml version="1.0" encoding="utf-8"?>
<openvx xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns="https://www.khronos.org/registry/vx/schema"
        xsi:schemaLocation="https://www.khronos.org/registry/vx/schema
       openvx-1-0.xsd"
        references="124">
    <library>openvx-debug</library>
    <struct size="16">USER_STRUCT_0</struct>
    <image reference="0" width="6" height="4" format="U008" name="INPUT_IMG" >
        <rectangle plane="0">
            <start_x>0</start_x>
            <start_y>0</start_y>
            <end_x>6</end_x>
            <end_y>4</end_y>
            <pixels>
                <uint8 x="0" y="0">0</uint8>
                <uint8 x="1" y="0">1</uint8>
                <uint8 x="2" y="0">2</uint8>
                <uint8 x="3" y="0">3</uint8>
                <uint8 x="4" y="0">4</uint8>
                <uint8 x="5" y="0">5</uint8>
                <uint8 x="0" y="1">6</uint8>
                <uint8 x="1" y="1">7</uint8>
                <uint8 x="2" y="1">8</uint8>
                <uint8 x="3" y="1">9</uint8>
                <uint8 x="4" y="1">10</uint8>
                <uint8 x="5" y="1">11</uint8>
                <uint8 x="0" y="2">12</uint8>
                <uint8 x="1" y="2">13</uint8>
                <uint8 x="2" y="2">14</uint8>
                <uint8 x="3" y="2">15</uint8>
                <uint8 x="4" y="2">16</uint8>
                <uint8 x="5" y="2">17</uint8>
                <uint8 x="0" y="3">18</uint8>
                <uint8 x="1" y="3">19</uint8>
                <uint8 x="2" y="3">20</uint8>
                <uint8 x="3" y="3">21</uint8>
                <uint8 x="4" y="3">22</uint8>
                <uint8 x="5" y="3">23</uint8>
            </pixels>
        </rectangle>
    </image>
    <image reference="1" width="6" height="4" format="U008" name="OUTPUT_IMG" >
    </image>
    <image reference="2" width="6" height="4" format="S016">
        <rectangle plane="0">
            <start_x>0</start_x>
            <start_y>0</start_y>
            <end_x>6</end_x>
            <end_y>4</end_y>
            <pixels>
                <int16 x="0" y="0">-32768</int16>
                <int16 x="1" y="0">-32767</int16>
```

```
            <int16 x="2" y="0">-32766</int16>
            <int16 x="3" y="0">-32765</int16>
            <int16 x="4" y="0">-32764</int16>
            <int16 x="5" y="0">-32763</int16>
            <int16 x="0" y="1">-32762</int16>
            <int16 x="1" y="1">-32761</int16>
            <int16 x="2" y="1">-32760</int16>
            <int16 x="3" y="1">-32759</int16>
            <int16 x="4" y="1">-32758</int16>
            <int16 x="5" y="1">-32757</int16>
            <int16 x="0" y="2">-32756</int16>
            <int16 x="1" y="2">-32755</int16>
            <int16 x="2" y="2">-32754</int16>
            <int16 x="3" y="2">-32753</int16>
            <int16 x="4" y="2">-32752</int16>
            <int16 x="5" y="2">-32751</int16>
            <int16 x="0" y="3">-32750</int16>
            <int16 x="1" y="3">-32749</int16>
            <int16 x="2" y="3">-32748</int16>
            <int16 x="3" y="3">-32747</int16>
            <int16 x="4" y="3">-32746</int16>
            <int16 x="5" y="3">-32745</int16>
        </pixels>
    </rectangle>
</image>
<image reference="3" width="6" height="4" format="U016">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <uint16 x="0" y="0">65535</uint16>
            <uint16 x="1" y="0">65534</uint16>
            <uint16 x="2" y="0">65533</uint16>
            <uint16 x="3" y="0">65532</uint16>
            <uint16 x="4" y="0">65531</uint16>
            <uint16 x="5" y="0">65530</uint16>
            <uint16 x="0" y="1">65529</uint16>
            <uint16 x="1" y="1">65528</uint16>
            <uint16 x="2" y="1">65527</uint16>
            <uint16 x="3" y="1">65526</uint16>
            <uint16 x="4" y="1">65525</uint16>
            <uint16 x="5" y="1">65524</uint16>
            <uint16 x="0" y="2">65523</uint16>
            <uint16 x="1" y="2">65522</uint16>
            <uint16 x="2" y="2">65521</uint16>
            <uint16 x="3" y="2">65520</uint16>
            <uint16 x="4" y="2">65519</uint16>
            <uint16 x="5" y="2">65518</uint16>
            <uint16 x="0" y="3">65517</uint16>
            <uint16 x="1" y="3">65516</uint16>
            <uint16 x="2" y="3">65515</uint16>
            <uint16 x="3" y="3">65514</uint16>
            <uint16 x="4" y="3">65513</uint16>
            <uint16 x="5" y="3">65512</uint16>
        </pixels>
    </rectangle>
</image>
<image reference="4" width="6" height="4" format="S032">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <int32 x="0" y="0">-2147483648</int32>
            <int32 x="1" y="0">-2147483647</int32>
            <int32 x="2" y="0">-2147483646</int32>
            <int32 x="3" y="0">-2147483645</int32>
            <int32 x="4" y="0">-2147483644</int32>
            <int32 x="5" y="0">-2147483643</int32>
            <int32 x="0" y="1">-2147483642</int32>
            <int32 x="1" y="1">-2147483641</int32>
            <int32 x="2" y="1">-2147483640</int32>
            <int32 x="3" y="1">-2147483639</int32>
            <int32 x="4" y="1">-2147483638</int32>
            <int32 x="5" y="1">-2147483637</int32>
            <int32 x="0" y="2">-2147483636</int32>
            <int32 x="1" y="2">-2147483635</int32>
            <int32 x="2" y="2">-2147483634</int32>
            <int32 x="3" y="2">-2147483633</int32>
            <int32 x="4" y="2">-2147483632</int32>
            <int32 x="5" y="2">-2147483631</int32>
            <int32 x="0" y="3">-2147483630</int32>
            <int32 x="1" y="3">-2147483629</int32>
            <int32 x="2" y="3">-2147483628</int32>
```

```
            <int32 x="3" y="3">-2147483627</int32>
            <int32 x="4" y="3">-2147483626</int32>
            <int32 x="5" y="3">-2147483625</int32>
        </pixels>
    </rectangle>
</image>
<image reference="5" width="6" height="4" format="U032">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <uint32 x="0" y="0">2147483647</uint32>
            <uint32 x="1" y="0">2147483646</uint32>
            <uint32 x="2" y="0">2147483645</uint32>
            <uint32 x="3" y="0">2147483644</uint32>
            <uint32 x="4" y="0">2147483643</uint32>
            <uint32 x="5" y="0">2147483642</uint32>
            <uint32 x="0" y="1">2147483641</uint32>
            <uint32 x="1" y="1">2147483640</uint32>
            <uint32 x="2" y="1">2147483639</uint32>
            <uint32 x="3" y="1">2147483638</uint32>
            <uint32 x="4" y="1">2147483637</uint32>
            <uint32 x="5" y="1">2147483636</uint32>
            <uint32 x="0" y="2">2147483635</uint32>
            <uint32 x="1" y="2">2147483634</uint32>
            <uint32 x="2" y="2">2147483633</uint32>
            <uint32 x="3" y="2">2147483632</uint32>
            <uint32 x="4" y="2">2147483631</uint32>
            <uint32 x="5" y="2">2147483630</uint32>
            <uint32 x="0" y="3">2147483629</uint32>
            <uint32 x="1" y="3">2147483628</uint32>
            <uint32 x="2" y="3">2147483627</uint32>
            <uint32 x="3" y="3">2147483626</uint32>
            <uint32 x="4" y="3">2147483625</uint32>
            <uint32 x="5" y="3">2147483624</uint32>
        </pixels>
    </rectangle>
</image>
<image reference="6" width="6" height="4" format="RGB2">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <rgb x="0" y="0">#000102</rgb>
            <rgb x="1" y="0">#010203</rgb>
            <rgb x="2" y="0">#020304</rgb>
            <rgb x="3" y="0">#030405</rgb>
            <rgb x="4" y="0">#040506</rgb>
            <rgb x="5" y="0">#050607</rgb>
            <rgb x="0" y="1">#060708</rgb>
            <rgb x="1" y="1">#070809</rgb>
            <rgb x="2" y="1">#08090a</rgb>
            <rgb x="3" y="1">#090a0b</rgb>
            <rgb x="4" y="1">#0a0b0c</rgb>
            <rgb x="5" y="1">#0b0c0d</rgb>
            <rgb x="0" y="2">#0c0d0e</rgb>
            <rgb x="1" y="2">#0d0e0f</rgb>
            <rgb x="2" y="2">#0e0f10</rgb>
            <rgb x="3" y="2">#0f1011</rgb>
            <rgb x="4" y="2">#101112</rgb>
            <rgb x="5" y="2">#111213</rgb>
            <rgb x="0" y="3">#121314</rgb>
            <rgb x="1" y="3">#131415</rgb>
            <rgb x="2" y="3">#141516</rgb>
            <rgb x="3" y="3">#151617</rgb>
            <rgb x="4" y="3">#161718</rgb>
            <rgb x="5" y="3">#171819</rgb>
        </pixels>
    </rectangle>
</image>
<image reference="7" width="6" height="4" format="RGBA">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <rgba x="0" y="0">#00010203</rgba>
            <rgba x="1" y="0">#01020304</rgba>
            <rgba x="2" y="0">#02030405</rgba>
            <rgba x="3" y="0">#03040506</rgba>
            <rgba x="4" y="0">#04050607</rgba>
            <rgba x="5" y="0">#05060708</rgba>
```

```
            <rgba x="0" y="1">#06070809</rgba>
            <rgba x="1" y="1">#0708090a</rgba>
            <rgba x="2" y="1">#08090a0b</rgba>
            <rgba x="3" y="1">#090a0b0c</rgba>
            <rgba x="4" y="1">#0a0b0c0d</rgba>
            <rgba x="5" y="1">#0b0c0d0e</rgba>
            <rgba x="0" y="2">#0c0d0e0f</rgba>
            <rgba x="1" y="2">#0d0e0f10</rgba>
            <rgba x="2" y="2">#0e0f1011</rgba>
            <rgba x="3" y="2">#0f101112</rgba>
            <rgba x="4" y="2">#10111213</rgba>
            <rgba x="5" y="2">#11121314</rgba>
            <rgba x="0" y="3">#12131415</rgba>
            <rgba x="1" y="3">#13141516</rgba>
            <rgba x="2" y="3">#14151617</rgba>
            <rgba x="3" y="3">#15161718</rgba>
            <rgba x="4" y="3">#16171819</rgba>
            <rgba x="5" y="3">#1718191a</rgba>
        </pixels>
    </rectangle>
</image>
<image reference="8" width="6" height="4" format="UYVY">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <yuv x="0" y="0">0 1</yuv>
            <yuv x="1" y="0">1 2</yuv>
            <yuv x="2" y="0">2 3</yuv>
            <yuv x="3" y="0">3 4</yuv>
            <yuv x="4" y="0">4 5</yuv>
            <yuv x="5" y="0">5 6</yuv>
            <yuv x="0" y="1">6 7</yuv>
            <yuv x="1" y="1">7 8</yuv>
            <yuv x="2" y="1">8 9</yuv>
            <yuv x="3" y="1">9 10</yuv>
            <yuv x="4" y="1">10 11</yuv>
            <yuv x="5" y="1">11 12</yuv>
            <yuv x="0" y="2">12 13</yuv>
            <yuv x="1" y="2">13 14</yuv>
            <yuv x="2" y="2">14 15</yuv>
            <yuv x="3" y="2">15 16</yuv>
            <yuv x="4" y="2">16 17</yuv>
            <yuv x="5" y="2">17 18</yuv>
            <yuv x="0" y="3">18 19</yuv>
            <yuv x="1" y="3">19 20</yuv>
            <yuv x="2" y="3">20 21</yuv>
            <yuv x="3" y="3">21 22</yuv>
            <yuv x="4" y="3">22 23</yuv>
            <yuv x="5" y="3">23 24</yuv>
        </pixels>
    </rectangle>
</image>
<image reference="9" width="6" height="4" format="YUYV">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <yuv x="0" y="0">0 1</yuv>
            <yuv x="1" y="0">1 2</yuv>
            <yuv x="2" y="0">2 3</yuv>
            <yuv x="3" y="0">3 4</yuv>
            <yuv x="4" y="0">4 5</yuv>
            <yuv x="5" y="0">5 6</yuv>
            <yuv x="0" y="1">6 7</yuv>
            <yuv x="1" y="1">7 8</yuv>
            <yuv x="2" y="1">8 9</yuv>
            <yuv x="3" y="1">9 10</yuv>
            <yuv x="4" y="1">10 11</yuv>
            <yuv x="5" y="1">11 12</yuv>
            <yuv x="0" y="2">12 13</yuv>
            <yuv x="1" y="2">13 14</yuv>
            <yuv x="2" y="2">14 15</yuv>
            <yuv x="3" y="2">15 16</yuv>
            <yuv x="4" y="2">16 17</yuv>
            <yuv x="5" y="2">17 18</yuv>
            <yuv x="0" y="3">18 19</yuv>
            <yuv x="1" y="3">19 20</yuv>
            <yuv x="2" y="3">20 21</yuv>
            <yuv x="3" y="3">21 22</yuv>
            <yuv x="4" y="3">22 23</yuv>
            <yuv x="5" y="3">23 24</yuv>
        </pixels>
```

```
    </rectangle>
</image>
<image reference="10" width="6" height="4" format="IYUV">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <uint8 x="0" y="0">0</uint8>
            <uint8 x="1" y="0">1</uint8>
            <uint8 x="2" y="0">2</uint8>
            <uint8 x="3" y="0">3</uint8>
            <uint8 x="4" y="0">4</uint8>
            <uint8 x="5" y="0">5</uint8>
            <uint8 x="0" y="1">6</uint8>
            <uint8 x="1" y="1">7</uint8>
            <uint8 x="2" y="1">8</uint8>
            <uint8 x="3" y="1">9</uint8>
            <uint8 x="4" y="1">10</uint8>
            <uint8 x="5" y="1">11</uint8>
            <uint8 x="0" y="2">12</uint8>
            <uint8 x="1" y="2">13</uint8>
            <uint8 x="2" y="2">14</uint8>
            <uint8 x="3" y="2">15</uint8>
            <uint8 x="4" y="2">16</uint8>
            <uint8 x="5" y="2">17</uint8>
            <uint8 x="0" y="3">18</uint8>
            <uint8 x="1" y="3">19</uint8>
            <uint8 x="2" y="3">20</uint8>
            <uint8 x="3" y="3">21</uint8>
            <uint8 x="4" y="3">22</uint8>
            <uint8 x="5" y="3">23</uint8>
        </pixels>
    </rectangle>
    <rectangle plane="1">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <uint8 x="0" y="0">0</uint8>
            <uint8 x="2" y="0">2</uint8>
            <uint8 x="4" y="0">4</uint8>
            <uint8 x="0" y="2">12</uint8>
            <uint8 x="2" y="2">14</uint8>
            <uint8 x="4" y="2">16</uint8>
        </pixels>
    </rectangle>
    <rectangle plane="2">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <uint8 x="0" y="0">0</uint8>
            <uint8 x="2" y="0">2</uint8>
            <uint8 x="4" y="0">4</uint8>
            <uint8 x="0" y="2">12</uint8>
            <uint8 x="2" y="2">14</uint8>
            <uint8 x="4" y="2">16</uint8>
        </pixels>
    </rectangle>
</image>
<image reference="11" width="6" height="4" format="YUV4">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <uint8 x="0" y="0">0</uint8>
            <uint8 x="1" y="0">1</uint8>
            <uint8 x="2" y="0">2</uint8>
            <uint8 x="3" y="0">3</uint8>
            <uint8 x="4" y="0">4</uint8>
            <uint8 x="5" y="0">5</uint8>
            <uint8 x="0" y="1">6</uint8>
            <uint8 x="1" y="1">7</uint8>
            <uint8 x="2" y="1">8</uint8>
            <uint8 x="3" y="1">9</uint8>
            <uint8 x="4" y="1">10</uint8>
            <uint8 x="5" y="1">11</uint8>
            <uint8 x="0" y="2">12</uint8>
            <uint8 x="1" y="2">13</uint8>
            <uint8 x="2" y="2">14</uint8>
            <uint8 x="3" y="2">15</uint8>
```

```
            <uint8 x="4" y="2">16</uint8>
            <uint8 x="5" y="2">17</uint8>
            <uint8 x="0" y="3">18</uint8>
            <uint8 x="1" y="3">19</uint8>
            <uint8 x="2" y="3">20</uint8>
            <uint8 x="3" y="3">21</uint8>
            <uint8 x="4" y="3">22</uint8>
            <uint8 x="5" y="3">23</uint8>
        </pixels>
    </rectangle>
    <rectangle plane="1">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <uint8 x="0" y="0">0</uint8>
            <uint8 x="1" y="0">1</uint8>
            <uint8 x="2" y="0">2</uint8>
            <uint8 x="3" y="0">3</uint8>
            <uint8 x="4" y="0">4</uint8>
            <uint8 x="5" y="0">5</uint8>
            <uint8 x="0" y="1">6</uint8>
            <uint8 x="1" y="1">7</uint8>
            <uint8 x="2" y="1">8</uint8>
            <uint8 x="3" y="1">9</uint8>
            <uint8 x="4" y="1">10</uint8>
            <uint8 x="5" y="1">11</uint8>
            <uint8 x="0" y="2">12</uint8>
            <uint8 x="1" y="2">13</uint8>
            <uint8 x="2" y="2">14</uint8>
            <uint8 x="3" y="2">15</uint8>
            <uint8 x="4" y="2">16</uint8>
            <uint8 x="5" y="2">17</uint8>
            <uint8 x="0" y="3">18</uint8>
            <uint8 x="1" y="3">19</uint8>
            <uint8 x="2" y="3">20</uint8>
            <uint8 x="3" y="3">21</uint8>
            <uint8 x="4" y="3">22</uint8>
            <uint8 x="5" y="3">23</uint8>
        </pixels>
    </rectangle>
    <rectangle plane="2">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <uint8 x="0" y="0">0</uint8>
            <uint8 x="1" y="0">1</uint8>
            <uint8 x="2" y="0">2</uint8>
            <uint8 x="3" y="0">3</uint8>
            <uint8 x="4" y="0">4</uint8>
            <uint8 x="5" y="0">5</uint8>
            <uint8 x="0" y="1">6</uint8>
            <uint8 x="1" y="1">7</uint8>
            <uint8 x="2" y="1">8</uint8>
            <uint8 x="3" y="1">9</uint8>
            <uint8 x="4" y="1">10</uint8>
            <uint8 x="5" y="1">11</uint8>
            <uint8 x="0" y="2">12</uint8>
            <uint8 x="1" y="2">13</uint8>
            <uint8 x="2" y="2">14</uint8>
            <uint8 x="3" y="2">15</uint8>
            <uint8 x="4" y="2">16</uint8>
            <uint8 x="5" y="2">17</uint8>
            <uint8 x="0" y="3">18</uint8>
            <uint8 x="1" y="3">19</uint8>
            <uint8 x="2" y="3">20</uint8>
            <uint8 x="3" y="3">21</uint8>
            <uint8 x="4" y="3">22</uint8>
            <uint8 x="5" y="3">23</uint8>
        </pixels>
    </rectangle>
</image>
<image reference="12" width="6" height="4" format="NV12">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <uint8 x="0" y="0">0</uint8>
            <uint8 x="1" y="0">1</uint8>
            <uint8 x="2" y="0">2</uint8>
            <uint8 x="3" y="0">3</uint8>
            <uint8 x="4" y="0">4</uint8>
```

```
            <uint8 x="5" y="0">5</uint8>
            <uint8 x="0" y="1">6</uint8>
            <uint8 x="1" y="1">7</uint8>
            <uint8 x="2" y="1">8</uint8>
            <uint8 x="3" y="1">9</uint8>
            <uint8 x="4" y="1">10</uint8>
            <uint8 x="5" y="1">11</uint8>
            <uint8 x="0" y="2">12</uint8>
            <uint8 x="1" y="2">13</uint8>
            <uint8 x="2" y="2">14</uint8>
            <uint8 x="3" y="2">15</uint8>
            <uint8 x="4" y="2">16</uint8>
            <uint8 x="5" y="2">17</uint8>
            <uint8 x="0" y="3">18</uint8>
            <uint8 x="1" y="3">19</uint8>
            <uint8 x="2" y="3">20</uint8>
            <uint8 x="3" y="3">21</uint8>
            <uint8 x="4" y="3">22</uint8>
            <uint8 x="5" y="3">23</uint8>
        </pixels>
    </rectangle>
    <rectangle plane="1">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <yuv x="0" y="0">0 1</yuv>
            <yuv x="2" y="0">2 3</yuv>
            <yuv x="4" y="0">4 5</yuv>
            <yuv x="0" y="2">12 13</yuv>
            <yuv x="2" y="2">14 15</yuv>
            <yuv x="4" y="2">16 17</yuv>
        </pixels>
    </rectangle>
</image>
<image reference="13" width="6" height="4" format="NV21">
    <rectangle plane="0">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <uint8 x="0" y="0">0</uint8>
            <uint8 x="1" y="0">1</uint8>
            <uint8 x="2" y="0">2</uint8>
            <uint8 x="3" y="0">3</uint8>
            <uint8 x="4" y="0">4</uint8>
            <uint8 x="5" y="0">5</uint8>
            <uint8 x="0" y="1">6</uint8>
            <uint8 x="1" y="1">7</uint8>
            <uint8 x="2" y="1">8</uint8>
            <uint8 x="3" y="1">9</uint8>
            <uint8 x="4" y="1">10</uint8>
            <uint8 x="5" y="1">11</uint8>
            <uint8 x="0" y="2">12</uint8>
            <uint8 x="1" y="2">13</uint8>
            <uint8 x="2" y="2">14</uint8>
            <uint8 x="3" y="2">15</uint8>
            <uint8 x="4" y="2">16</uint8>
            <uint8 x="5" y="2">17</uint8>
            <uint8 x="0" y="3">18</uint8>
            <uint8 x="1" y="3">19</uint8>
            <uint8 x="2" y="3">20</uint8>
            <uint8 x="3" y="3">21</uint8>
            <uint8 x="4" y="3">22</uint8>
            <uint8 x="5" y="3">23</uint8>
        </pixels>
    </rectangle>
    <rectangle plane="1">
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>6</end_x>
        <end_y>4</end_y>
        <pixels>
            <yuv x="0" y="0">0 1</yuv>
            <yuv x="2" y="0">2 3</yuv>
            <yuv x="4" y="0">4 5</yuv>
            <yuv x="0" y="2">12 13</yuv>
            <yuv x="2" y="2">14 15</yuv>
            <yuv x="4" y="2">16 17</yuv>
        </pixels>
    </rectangle>
</image>
<matrix reference="14" elemType="VX_TYPE_FLOAT32" rows="3" columns="3">
    <float32 row="0" column="0">-3.141593</float32>
    <float32 row="0" column="1">0.000000</float32>
```

```
        <float32 row="0" column="2">3.141593</float32>
        <float32 row="1" column="0">-6.283185</float32>
        <float32 row="1" column="1">0.000000</float32>
        <float32 row="1" column="2">6.283185</float32>
        <float32 row="2" column="0">-3.141593</float32>
        <float32 row="2" column="1">0.000000</float32>
        <float32 row="2" column="2">3.141593</float32>
    </matrix>
    <matrix reference="15" elemType="VX_TYPE_INT32" rows="3" columns="3">
        <int32 row="0" column="0">-23</int32>
        <int32 row="0" column="1">904</int32>
        <int32 row="0" column="2">29</int32>
        <int32 row="1" column="0">782</int32>
        <int32 row="1" column="1">823</int32>
        <int32 row="1" column="2">288</int32>
        <int32 row="2" column="0">848</int32>
        <int32 row="2" column="1">828</int32>
        <int32 row="2" column="2">994</int32>
    </matrix>
    <scalar reference="16" elemType="VX_TYPE_CHAR">
        <char>z</char>
    </scalar>
    <scalar reference="17" elemType="VX_TYPE_UINT8">
        <uint8>255</uint8>
    </scalar>
    <scalar reference="18" elemType="VX_TYPE_INT8">
        <int8>-128</int8>
    </scalar>
    <scalar reference="19" elemType="VX_TYPE_FLOAT32">
        <float32>3.141593</float32>
    </scalar>
    <scalar reference="20" elemType="VX_TYPE_FLOAT64">
        <float64>6.283185</float64>
    </scalar>
    <scalar reference="21" elemType="VX_TYPE_DF_IMAGE">
        <df_image>NV12</df_image>
    </scalar>
    <scalar reference="22" elemType="VX_TYPE_UINT16">
        <uint16>65535</uint16>
    </scalar>
    <scalar reference="23" elemType="VX_TYPE_INT16">
        <int16>-32768</int16>
    </scalar>
    <scalar reference="24" elemType="VX_TYPE_UINT64">
        <uint64>18446744073709551615</uint64>
    </scalar>
    <scalar reference="25" elemType="VX_TYPE_INT64">
        <int64>-9223372036854775808</int64>
    </scalar>
    <scalar reference="26" elemType="VX_TYPE_UINT32">
        <uint32>4294967295</uint32>
    </scalar>
    <scalar reference="27" elemType="VX_TYPE_SIZE">
        <size>100</size>
    </scalar>
    <scalar reference="28" elemType="VX_TYPE_BOOL">
        <bool>true</bool>
    </scalar>
    <threshold reference="29" elemType="VX_TYPE_UINT8" true_value="0"
      false_value="0">
        <binary>240</binary>
    </threshold>
    <threshold reference="30" elemType="VX_TYPE_UINT8" true_value="0"
      false_value="0">
        <range lower="10" upper="240" />
    </threshold>
    <convolution reference="31" rows="3" columns="3" scale="16">
        <int16 row="0" column="0">-1</int16>
        <int16 row="0" column="1">0</int16>
        <int16 row="0" column="2">1</int16>
        <int16 row="1" column="0">-2</int16>
        <int16 row="1" column="1">0</int16>
        <int16 row="1" column="2">1</int16>
        <int16 row="2" column="0">-1</int16>
        <int16 row="2" column="1">0</int16>
        <int16 row="2" column="2">1</int16>
    </convolution>
    <lut reference="32" count="256" elemType="VX_TYPE_UINT8">
        <uint8 index="0">255</uint8>
        <uint8 index="1">0</uint8>
        <uint8 index="2">1</uint8>
        <uint8 index="3">2</uint8>
        <uint8 index="4">3</uint8>
        <uint8 index="5">4</uint8>
        <uint8 index="6">5</uint8>
        <uint8 index="7">6</uint8>
        <uint8 index="8">7</uint8>
```

```
        <uint8 index="9">8</uint8>
        <uint8 index="10">9</uint8>
        <uint8 index="11">10</uint8>
        <uint8 index="12">11</uint8>
        <uint8 index="13">12</uint8>
        <uint8 index="14">13</uint8>
        <uint8 index="15">14</uint8>
        <uint8 index="16">15</uint8>
        <uint8 index="17">16</uint8>
        <uint8 index="18">17</uint8>
        <uint8 index="19">18</uint8>
        <uint8 index="20">19</uint8>
        <uint8 index="21">20</uint8>
        <uint8 index="22">21</uint8>
        <uint8 index="23">22</uint8>
        <uint8 index="24">23</uint8>
        <uint8 index="25">24</uint8>
        <uint8 index="26">25</uint8>
        <uint8 index="27">26</uint8>
        <uint8 index="28">27</uint8>
        <uint8 index="29">28</uint8>
        <uint8 index="30">29</uint8>
        <uint8 index="31">30</uint8>
        <uint8 index="32">31</uint8>
        <uint8 index="33">32</uint8>
        <uint8 index="34">33</uint8>
        <uint8 index="35">34</uint8>
        <uint8 index="36">35</uint8>
        <uint8 index="37">36</uint8>
        <uint8 index="38">37</uint8>
        <uint8 index="39">38</uint8>
        <uint8 index="40">39</uint8>
        <uint8 index="41">40</uint8>
        <uint8 index="42">41</uint8>
        <uint8 index="43">42</uint8>
        <uint8 index="44">43</uint8>
        <uint8 index="45">44</uint8>
        <uint8 index="46">45</uint8>
        <uint8 index="47">46</uint8>
        <uint8 index="48">47</uint8>
        <uint8 index="49">48</uint8>
        <uint8 index="50">49</uint8>
        <uint8 index="51">50</uint8>
        <uint8 index="52">51</uint8>
        <uint8 index="53">52</uint8>
        <uint8 index="54">53</uint8>
        <uint8 index="55">54</uint8>
        <uint8 index="56">55</uint8>
        <uint8 index="57">56</uint8>
        <uint8 index="58">57</uint8>
        <uint8 index="59">58</uint8>
        <uint8 index="60">59</uint8>
        <uint8 index="61">60</uint8>
        <uint8 index="62">61</uint8>
        <uint8 index="63">62</uint8>
        <uint8 index="64">63</uint8>
        <uint8 index="65">64</uint8>
        <uint8 index="66">65</uint8>
        <uint8 index="67">66</uint8>
        <uint8 index="68">67</uint8>
        <uint8 index="69">68</uint8>
        <uint8 index="70">69</uint8>
        <uint8 index="71">70</uint8>
        <uint8 index="72">71</uint8>
        <uint8 index="73">72</uint8>
        <uint8 index="74">73</uint8>
        <uint8 index="75">74</uint8>
        <uint8 index="76">75</uint8>
        <uint8 index="77">76</uint8>
        <uint8 index="78">77</uint8>
        <uint8 index="79">78</uint8>
        <uint8 index="80">79</uint8>
        <uint8 index="81">80</uint8>
        <uint8 index="82">81</uint8>
        <uint8 index="83">82</uint8>
        <uint8 index="84">83</uint8>
        <uint8 index="85">84</uint8>
        <uint8 index="86">85</uint8>
        <uint8 index="87">86</uint8>
        <uint8 index="88">87</uint8>
        <uint8 index="89">88</uint8>
        <uint8 index="90">89</uint8>
        <uint8 index="91">90</uint8>
        <uint8 index="92">91</uint8>
        <uint8 index="93">92</uint8>
        <uint8 index="94">93</uint8>
        <uint8 index="95">94</uint8>
```

```
        <uint8 index="96">95</uint8>
        <uint8 index="97">96</uint8>
        <uint8 index="98">97</uint8>
        <uint8 index="99">98</uint8>
        <uint8 index="100">99</uint8>
        <uint8 index="101">100</uint8>
        <uint8 index="102">101</uint8>
        <uint8 index="103">102</uint8>
        <uint8 index="104">103</uint8>
        <uint8 index="105">104</uint8>
        <uint8 index="106">105</uint8>
        <uint8 index="107">106</uint8>
        <uint8 index="108">107</uint8>
        <uint8 index="109">108</uint8>
        <uint8 index="110">109</uint8>
        <uint8 index="111">110</uint8>
        <uint8 index="112">111</uint8>
        <uint8 index="113">112</uint8>
        <uint8 index="114">113</uint8>
        <uint8 index="115">114</uint8>
        <uint8 index="116">115</uint8>
        <uint8 index="117">116</uint8>
        <uint8 index="118">117</uint8>
        <uint8 index="119">118</uint8>
        <uint8 index="120">119</uint8>
        <uint8 index="121">120</uint8>
        <uint8 index="122">121</uint8>
        <uint8 index="123">122</uint8>
        <uint8 index="124">123</uint8>
        <uint8 index="125">124</uint8>
        <uint8 index="126">125</uint8>
        <uint8 index="127">126</uint8>
        <uint8 index="128">127</uint8>
        <uint8 index="129">128</uint8>
        <uint8 index="130">129</uint8>
        <uint8 index="131">130</uint8>
        <uint8 index="132">131</uint8>
        <uint8 index="133">132</uint8>
        <uint8 index="134">133</uint8>
        <uint8 index="135">134</uint8>
        <uint8 index="136">135</uint8>
        <uint8 index="137">136</uint8>
        <uint8 index="138">137</uint8>
        <uint8 index="139">138</uint8>
        <uint8 index="140">139</uint8>
        <uint8 index="141">140</uint8>
        <uint8 index="142">141</uint8>
        <uint8 index="143">142</uint8>
        <uint8 index="144">143</uint8>
        <uint8 index="145">144</uint8>
        <uint8 index="146">145</uint8>
        <uint8 index="147">146</uint8>
        <uint8 index="148">147</uint8>
        <uint8 index="149">148</uint8>
        <uint8 index="150">149</uint8>
        <uint8 index="151">150</uint8>
        <uint8 index="152">151</uint8>
        <uint8 index="153">152</uint8>
        <uint8 index="154">153</uint8>
        <uint8 index="155">154</uint8>
        <uint8 index="156">155</uint8>
        <uint8 index="157">156</uint8>
        <uint8 index="158">157</uint8>
        <uint8 index="159">158</uint8>
        <uint8 index="160">159</uint8>
        <uint8 index="161">160</uint8>
        <uint8 index="162">161</uint8>
        <uint8 index="163">162</uint8>
        <uint8 index="164">163</uint8>
        <uint8 index="165">164</uint8>
        <uint8 index="166">165</uint8>
        <uint8 index="167">166</uint8>
        <uint8 index="168">167</uint8>
        <uint8 index="169">168</uint8>
        <uint8 index="170">169</uint8>
        <uint8 index="171">170</uint8>
        <uint8 index="172">171</uint8>
        <uint8 index="173">172</uint8>
        <uint8 index="174">173</uint8>
        <uint8 index="175">174</uint8>
        <uint8 index="176">175</uint8>
        <uint8 index="177">176</uint8>
        <uint8 index="178">177</uint8>
        <uint8 index="179">178</uint8>
        <uint8 index="180">179</uint8>
        <uint8 index="181">180</uint8>
        <uint8 index="182">181</uint8>
```

```
      <uint8 index="183">182</uint8>
      <uint8 index="184">183</uint8>
      <uint8 index="185">184</uint8>
      <uint8 index="186">185</uint8>
      <uint8 index="187">186</uint8>
      <uint8 index="188">187</uint8>
      <uint8 index="189">188</uint8>
      <uint8 index="190">189</uint8>
      <uint8 index="191">190</uint8>
      <uint8 index="192">191</uint8>
      <uint8 index="193">192</uint8>
      <uint8 index="194">193</uint8>
      <uint8 index="195">194</uint8>
      <uint8 index="196">195</uint8>
      <uint8 index="197">196</uint8>
      <uint8 index="198">197</uint8>
      <uint8 index="199">198</uint8>
      <uint8 index="200">199</uint8>
      <uint8 index="201">200</uint8>
      <uint8 index="202">201</uint8>
      <uint8 index="203">202</uint8>
      <uint8 index="204">203</uint8>
      <uint8 index="205">204</uint8>
      <uint8 index="206">205</uint8>
      <uint8 index="207">206</uint8>
      <uint8 index="208">207</uint8>
      <uint8 index="209">208</uint8>
      <uint8 index="210">209</uint8>
      <uint8 index="211">210</uint8>
      <uint8 index="212">211</uint8>
      <uint8 index="213">212</uint8>
      <uint8 index="214">213</uint8>
      <uint8 index="215">214</uint8>
      <uint8 index="216">215</uint8>
      <uint8 index="217">216</uint8>
      <uint8 index="218">217</uint8>
      <uint8 index="219">218</uint8>
      <uint8 index="220">219</uint8>
      <uint8 index="221">220</uint8>
      <uint8 index="222">221</uint8>
      <uint8 index="223">222</uint8>
      <uint8 index="224">223</uint8>
      <uint8 index="225">224</uint8>
      <uint8 index="226">225</uint8>
      <uint8 index="227">226</uint8>
      <uint8 index="228">227</uint8>
      <uint8 index="229">228</uint8>
      <uint8 index="230">229</uint8>
      <uint8 index="231">230</uint8>
      <uint8 index="232">231</uint8>
      <uint8 index="233">232</uint8>
      <uint8 index="234">233</uint8>
      <uint8 index="235">234</uint8>
      <uint8 index="236">235</uint8>
      <uint8 index="237">236</uint8>
      <uint8 index="238">237</uint8>
      <uint8 index="239">238</uint8>
      <uint8 index="240">239</uint8>
      <uint8 index="241">240</uint8>
      <uint8 index="242">241</uint8>
      <uint8 index="243">242</uint8>
      <uint8 index="244">243</uint8>
      <uint8 index="245">244</uint8>
      <uint8 index="246">245</uint8>
      <uint8 index="247">246</uint8>
      <uint8 index="248">247</uint8>
      <uint8 index="249">248</uint8>
      <uint8 index="250">249</uint8>
      <uint8 index="251">250</uint8>
      <uint8 index="252">251</uint8>
      <uint8 index="253">252</uint8>
      <uint8 index="254">253</uint8>
      <uint8 index="255">254</uint8>
  </lut>
  <remap reference="33" src_width="6" src_height="4" dst_width="6" dst_height
    ="4">
      <point src_x="0.500000" src_y="0.500000" dst_x="0" dst_y="0" />
      <point src_x="1.000000" src_y="0.500000" dst_x="1" dst_y="0" />
      <point src_x="1.500000" src_y="0.500000" dst_x="2" dst_y="0" />
      <point src_x="2.000000" src_y="0.500000" dst_x="3" dst_y="0" />
      <point src_x="2.500000" src_y="0.500000" dst_x="4" dst_y="0" />
      <point src_x="3.000000" src_y="0.500000" dst_x="5" dst_y="0" />
      <point src_x="0.500000" src_y="1.000000" dst_x="0" dst_y="1" />
      <point src_x="1.000000" src_y="1.000000" dst_x="1" dst_y="1" />
      <point src_x="1.500000" src_y="1.000000" dst_x="2" dst_y="1" />
      <point src_x="2.000000" src_y="1.000000" dst_x="3" dst_y="1" />
      <point src_x="2.500000" src_y="1.000000" dst_x="4" dst_y="1" />
```

```
        <point src_x="3.000000" src_y="1.000000" dst_x="5" dst_y="1" />
        <point src_x="0.500000" src_y="1.500000" dst_x="0" dst_y="2" />
        <point src_x="1.000000" src_y="1.500000" dst_x="1" dst_y="2" />
        <point src_x="1.500000" src_y="1.500000" dst_x="2" dst_y="2" />
        <point src_x="2.000000" src_y="1.500000" dst_x="3" dst_y="2" />
        <point src_x="2.500000" src_y="1.500000" dst_x="4" dst_y="2" />
        <point src_x="3.000000" src_y="1.500000" dst_x="5" dst_y="2" />
        <point src_x="0.500000" src_y="2.000000" dst_x="0" dst_y="3" />
        <point src_x="1.000000" src_y="2.000000" dst_x="1" dst_y="3" />
        <point src_x="1.500000" src_y="2.000000" dst_x="2" dst_y="3" />
        <point src_x="2.000000" src_y="2.000000" dst_x="3" dst_y="3" />
        <point src_x="2.500000" src_y="2.000000" dst_x="4" dst_y="3" />
        <point src_x="3.000000" src_y="2.000000" dst_x="5" dst_y="3" />
    </remap>
    <distribution reference="34" bins="16" offset="0" range="256">
        <frequency bin="0">0</frequency>
        <frequency bin="1">1</frequency>
        <frequency bin="2">2</frequency>
        <frequency bin="3">3</frequency>
        <frequency bin="4">4</frequency>
        <frequency bin="5">5</frequency>
        <frequency bin="6">6</frequency>
        <frequency bin="7">7</frequency>
        <frequency bin="8">8</frequency>
        <frequency bin="9">7</frequency>
        <frequency bin="10">6</frequency>
        <frequency bin="11">5</frequency>
        <frequency bin="12">4</frequency>
        <frequency bin="13">3</frequency>
        <frequency bin="14">2</frequency>
        <frequency bin="15">1</frequency>
    </distribution>
    <pyramid reference="35" width="24" height="16" format="U008" scale="
      0.500000" levels="4">
        <image reference="36" width="24" height="16" format="U008">
        </image>
        <image reference="37" width="12" height="8" format="U008">
        </image>
        <image reference="38" width="6" height="4" format="U008">
        </image>
        <image reference="39" width="3" height="2" format="U008">
        </image>
    </pyramid>
    <graph reference="40" name="GRAPH1" >
        <node reference="52">
            <kernel>org.khronos.openvx.sobel3x3</kernel>
            <parameter index="0" reference="0" />
            <parameter index="1" reference="41" />
            <parameter index="2" reference="42" />
        </node>
        <node reference="54">
            <kernel>org.khronos.openvx.convert_depth</kernel>
            <parameter index="0" reference="41" />
            <parameter index="1" reference="1" />
            <parameter index="2" reference="53" />
            <parameter index="3" reference="51" />
        </node>
        <parameter index="0" node="52" parameter="0" />
        <parameter index="1" node="54" parameter="1" />
        <image reference="41" width="0" height="0" format="VIRT" />
        <image reference="42" width="0" height="0" format="S016" />
        <image reference="43" width="320" height="240" format="VIRT" />
        <image reference="44" width="640" height="480" format="U008" />
        <pyramid reference="45" width="0" height="0" format="VIRT" scale="
      0.500000" levels="4" />
        <pyramid reference="46" width="640" height="480" format="VIRT" scale="
      0.500000" levels="4" />
        <pyramid reference="47" width="640" height="480" format="U008" scale="
      0.500000" levels="4" />
        <array reference="48" capacity="0" elemType="VX_TYPE_INVALID" />
        <array reference="49" capacity="0" elemType="VX_TYPE_KEYPOINT" />
        <array reference="50" capacity="1000" elemType="VX_TYPE_KEYPOINT" />
    </graph>
    <scalar reference="51" elemType="VX_TYPE_INT32">
        <int32>7</int32>
    </scalar>
    <scalar reference="53" elemType="VX_TYPE_ENUM">
        <enum>40961</enum>
    </scalar>
    <delay reference="55" count="3">
        <image reference="56" width="6" height="4" format="U008">
        </image>
        <image reference="57" width="6" height="4" format="U008">
        </image>
        <image reference="58" width="6" height="4" format="U008">
        </image>
    </delay>
```

```
<delay reference="59" count="2">
    <pyramid reference="60" width="24" height="16" format="U008" scale="
  0.500000" levels="4">
        <image reference="61" width="24" height="16" format="U008">
        </image>
        <image reference="62" width="12" height="8" format="U008">
        </image>
        <image reference="63" width="6" height="4" format="U008">
        </image>
        <image reference="64" width="3" height="2" format="U008">
        </image>
    </pyramid>
    <pyramid reference="65" width="24" height="16" format="U008" scale="
  0.500000" levels="4">
        <image reference="66" width="24" height="16" format="U008">
        </image>
        <image reference="67" width="12" height="8" format="U008">
        </image>
        <image reference="68" width="6" height="4" format="U008">
        </image>
        <image reference="69" width="3" height="2" format="U008">
        </image>
    </pyramid>
</delay>
<delay reference="70" count="2">
    <lut reference="71" count="256" elemType="VX_TYPE_UINT8">
    </lut>
    <lut reference="72" count="256" elemType="VX_TYPE_UINT8">
    </lut>
</delay>
<delay reference="73" count="2">
    <matrix reference="74" elemType="VX_TYPE_FLOAT32" rows="3" columns="3">
    </matrix>
    <matrix reference="75" elemType="VX_TYPE_FLOAT32" rows="3" columns="3">
    </matrix>
</delay>
<delay reference="76" count="2">
    <convolution reference="77" rows="3" columns="3" scale="1">
    </convolution>
    <convolution reference="78" rows="3" columns="3" scale="1">
    </convolution>
</delay>
<delay reference="79" count="2">
    <distribution reference="80" bins="16" offset="0" range="256">
    </distribution>
    <distribution reference="81" bins="16" offset="0" range="256">
    </distribution>
</delay>
<delay reference="82" count="2">
    <threshold reference="83" elemType="VX_TYPE_UINT8" true_value="0"
  false_value="0">
        <binary>0</binary>
    </threshold>
    <threshold reference="84" elemType="VX_TYPE_UINT8" true_value="0"
  false_value="0">
        <binary>0</binary>
    </threshold>
</delay>
<delay reference="85" count="2">
    <threshold reference="86" elemType="VX_TYPE_UINT8" true_value="0"
  false_value="0">
        <range lower="0" upper="0" />
    </threshold>
    <threshold reference="87" elemType="VX_TYPE_UINT8" true_value="0"
  false_value="0">
        <range lower="0" upper="0" />
    </threshold>
</delay>
<delay reference="88" count="4">
    <scalar reference="89" elemType="VX_TYPE_UINT8">
        <uint8>0</uint8>
    </scalar>
    <scalar reference="90" elemType="VX_TYPE_UINT8">
        <uint8>0</uint8>
    </scalar>
    <scalar reference="91" elemType="VX_TYPE_UINT8">
        <uint8>0</uint8>
    </scalar>
    <scalar reference="92" elemType="VX_TYPE_UINT8">
        <uint8>0</uint8>
    </scalar>
</delay>
<delay reference="93" count="2">
    <remap reference="94" src_width="6" src_height="4" dst_width="6"
  dst_height="4">
    </remap>
    <remap reference="95" src_width="6" src_height="4" dst_width="6"
```

```
  dst_height="4">
    </remap>
</delay>
<array reference="96" capacity="10" elemType="VX_TYPE_UINT8">
    <uint8>2 3 4 5 6 7 8 9 10 11 </uint8>
</array>
<array reference="97" capacity="20" elemType="VX_TYPE_CHAR">
    <char>a 13,.;^-</char>
</array>
<array reference="98" capacity="4" elemType="VX_TYPE_ENUM">
    <enum>-1 0 45057 </enum>
</array>
<array reference="99" capacity="4" elemType="VX_TYPE_DF_IMAGE">
    <df_image>RGB2 U008 VIRT </df_image>
</array>
<array reference="100" capacity="3" elemType="VX_TYPE_KEYPOINT">
    <keypoint>
        <x>0</x>
        <y>0</y>
        <strength>2.300000</strength>
        <scale>6.555550</scale>
        <orientation>0.905900</orientation>
        <tracking_status>5</tracking_status>
        <error>3.545500</error>
    </keypoint>
    <keypoint>
        <x>400</x>
        <y>235</y>
        <strength>5.222200</strength>
        <scale>1.221000</scale>
        <orientation>0.569500</orientation>
        <tracking_status>8</tracking_status>
        <error>462.500000</error>
    </keypoint>
</array>
<array reference="101" capacity="5" elemType="VX_TYPE_RECTANGLE">
    <rectangle>
        <start_x>0</start_x>
        <start_y>0</start_y>
        <end_x>640</end_x>
        <end_y>320</end_y>
    </rectangle>
    <rectangle>
        <start_x>65</start_x>
        <start_y>32</start_y>
        <end_x>128</end_x>
        <end_y>362</end_y>
    </rectangle>
</array>
<array reference="102" capacity="6" elemType="VX_TYPE_COORDINATES2D">
    <coordinates2d>
        <x>1</x>
        <y>2</y>
    </coordinates2d>
    <coordinates2d>
        <x>0</x>
        <y>55</y>
    </coordinates2d>
</array>
<array reference="103" capacity="6" elemType="VX_TYPE_COORDINATES3D">
    <coordinates3d>
        <x>1</x>
        <y>2</y>
        <z>3</z>
    </coordinates3d>
    <coordinates3d>
        <x>55</x>
        <y>66</y>
        <z>77</z>
    </coordinates3d>
</array>
<array reference="104" capacity="8" elemType="VX_TYPE_INT8">
    <int8>5 0 -3 -8 </int8>
</array>
<array reference="105" capacity="6" elemType="VX_TYPE_INT16">
    <int16>200 100 0 -100 -200 </int16>
</array>
<array reference="106" capacity="6" elemType="VX_TYPE_INT32">
    <int32>200000 100000 0 -100000 -200000 </int32>
</array>
<array reference="107" capacity="3" elemType="VX_TYPE_BOOL">
    <bool>true false true </bool>
</array>
<array reference="108" capacity="4" elemType="VX_TYPE_SIZE">
    <size>8000 24000 </size>
</array>
<array reference="109" capacity="2" elemType="VX_TYPE_FLOAT64">
```

```
        <float64>1235.255660 -563.256700 </float64>
    </array>
    <array reference="110" capacity="8" elemType="VX_TYPE_UINT64">
        <uint64>9000000000 8000000000 7000000000 6000000000 </uint64>
    </array>
    <array reference="111" capacity="6" elemType="VX_TYPE_UINT16">
        <uint16>290 100 0 100 260 </uint16>
    </array>
    <array reference="112" capacity="6" elemType="VX_TYPE_UINT32">
        <uint32>200000 100000 0 100000 200000 </uint32>
    </array>
    <array reference="113" capacity="2" elemType="VX_TYPE_FLOAT32">
        <float32>1235.255615 -563.256714 </float32>
    </array>
    <array reference="114" capacity="8" elemType="VX_TYPE_INT64">
        <int64>9000000000 8000000000 -7000000000 -6000000000 </int64>
    </array>
    <delay reference="115" count="4">
        <array reference="116" capacity="6" elemType="VX_TYPE_COORDINATES2D">
        </array>
        <array reference="117" capacity="6" elemType="VX_TYPE_COORDINATES2D">
        </array>
        <array reference="118" capacity="6" elemType="VX_TYPE_COORDINATES2D">
        </array>
        <array reference="119" capacity="6" elemType="VX_TYPE_COORDINATES2D">
            <coordinates2d>
                <x>1</x>
                <y>2</y>
            </coordinates2d>
            <coordinates2d>
                <x>0</x>
                <y>55</y>
            </coordinates2d>
        </array>
    </delay>
    <array reference="120" capacity="4" elemType="USER_STRUCT_0">
        <user>32 0 0 0 0 2 0 0 154 153 153 153 153 153 9 64 </user>
        <user>64 0 0 0 0 0 0 0 154 153 153 153 153 153 25 64 </user>
    </array>
    <delay reference="121" count="2">
        <array reference="122" capacity="4" elemType="USER_STRUCT_0">
        </array>
        <array reference="123" capacity="4" elemType="USER_STRUCT_0">
        </array>
    </delay>
</openvx>
```

Chapter 4

Module Documentation

4.1 Extension: XML API

The Khronos Extension for OpenVX XML Import and Export Support.

Macros

- #define VX_MAX_REFERENCE_NAME (64)

  Defines the maximum number of characters in a reference name string.

Typedefs

- typedef struct _vx_import * vx_import

  An abstract handle to an import object.

Enumerations

- enum vx_ext_import_reference_attribute_e { VX_REF_ATTRIBUTE_NAME = (((VX_ID_KHRONOS) << 20) | (VX_TYPE_REFERENCE << 8)) + 0x2 }

  Extended reference attribute list. This extension adds new attributes that can be queried by the `vxQueryReference` function.
- enum vx_ext_import_type_e { VX_TYPE_IMPORT = 0x814 }

  The Object Type Enumeration for Imports.
- enum vx_ext_import_types_e { VX_IMPORT_TYPE_XML = 0 }

  The import type enumeration.
- enum vx_import_attribute_e { VX_IMPORT_ATTRIBUTE_COUNT = (((VX_ID_KHRONOS) << 20) | (VX_TYPE_IMPORT << 8)) + 0x0, VX_IMPORT_ATTRIBUTE_TYPE = (((VX_ID_KHRONOS) << 20) | (VX_TYPE_IMPORT << 8)) + 0x1 }

  The import attributes list.

Functions

- vx_status vxExportToXML (vx_context context, vx_char xmlfile[])

  Exports all objects in the context to an XML file which uses the OpenVX XML Schema.
- vx_import vxImportFromXML (vx_context context, vx_char xmlfile[])

  Imports all framework and data objects from an XML file into the given context.

- vx_status vxSetReferenceName (vx_reference ref, const vx_char *name)

  Name a reference
  This function is used to associate a name to a reference. This name can be used by the OpenVX implementation in log messages and any other reporting mechanisms. It is also intended to be used by `vxGetImportReferenceByName` to retrieve a named reference from a vx_import object.
- vx_reference vxGetImportReferenceByName (vx_import import, const vx_char *name)

  Used to retrieve a reference by name from the import when the name is known beforehand. If multiple references have the same name, then any one of them may be returned.
- vx_reference vxGetImportReferenceByIndex (vx_import import, vx_uint32 index)

  Used to retrieve a reference by the index from the import.
- vx_status vxQueryImport (vx_import import, vx_enum attribute, void *ptr, vx_size size)

  Used to query the import about its properties.
- vx_status vxReleaseImport (vx_import *import)

  Releases a reference to an import object. Also internally releases its references to its imported objects. These imported objects may not be garbage collected until their total reference counts are zero.
- vx_status vxReleaseReference (vx_reference *ref)

  Releases a reference. The object may not be garbage collected until its total reference count is zero.

4.1.1 Detailed Description

The Khronos Extension for OpenVX XML Import and Export Support.

4.1.2 Macro Definition Documentation

4.1.2.1 #define VX_MAX_REFERENCE_NAME (64)

Defines the maximum number of characters in a reference name string.

See Also

vxSetReferenceName
vxGetImportReferenceByName

Definition at line 43 of file vx_khr_xml.h.

4.1.3 Enumeration Type Documentation

4.1.3.1 enum vx_ext_import_reference_attribute_e

Extended reference attribute list. This extension adds new attributes that can be queried by the `vxQueryReference` function.

Enumerator:

VX_REF_ATTRIBUTE_NAME Used to query the reference for its name. Use a `*vx_char` parameter.

Definition at line 50 of file vx_khr_xml.h.

4.1.3.2 enum vx_ext_import_type_e

The Object Type Enumeration for Imports.

Enumerator:

VX_TYPE_IMPORT A `vx_import`

Definition at line 58 of file vx_khr_xml.h.

4.1.3.3 enum vx_ext_import_types_e

The import type enumeration.

See Also

VX_IMPORT_ATTRIBUTE_TYPE

Enumerator:

VX_IMPORT_TYPE_XML The XML import type.

Definition at line 66 of file vx_khr_xml.h.

4.1.3.4 enum vx_import_attribute_e

The import attributes list.

See Also

vxQueryImport

Enumerator:

VX_IMPORT_ATTRIBUTE_COUNT Returns the number of references in the import object. Use a `vx_uint32` parameter.

VX_IMPORT_ATTRIBUTE_TYPE Returns the type of import. Use a `vx_ext_import_types_e` parameter.

Definition at line 74 of file vx_khr_xml.h.

4.1.4 Function Documentation

4.1.4.1 vx_status vxExportToXML (vx_context *context*, vx_char *xmlfile[]*)

Exports all objects in the context to an XML file which uses the OpenVX XML Schema.

Parameters

in	*context*	The context to export.
in	*xmlfile*	The file name to write the XML into.

Note

The reference numbers contained in the xml file can appear in any order but should be inclusive from index number 0 to [number of references - 1]. For example, if there are 20 references in the xml file, none of the reference indices should be >= 20.

Returns

A `vx_status_e` enumeration.

See Also

https://www.khronos.org/registry/vx/schema/openvx-1-0.xsd

4.1.4.2 vx_import vxImportFromXML (vx_context *context*, vx_char *xmlfile[]*)

Imports all framework and data objects from an XML file into the given context.

Parameters

in	*context*	The context to import into.
in	*xmlfile*	The XML file to read.

Note

The reference indices in the import object corresponds with the reference numbers in the XML file. It is assumed that the program has some means to know which references to use from imported list (either by name: `vxGetImportReferenceByName`, or by index from looking at the XML file (debug use case): `vxGetImportReferenceByIndex`). Alternativly, the program can use `vxGetImportReferenceByIndex` in a loop and query each one to understand what was imported. After all references of interest have been retrieved, this import obects should be released using `vxReleaseImport`.

Returns

vx_import object containing references to the imported objects in the context

See Also

`https://www.khronos.org/registry/vx/schema/openvx-1-0.xsd`

4.1.4.3 vx_status vxSetReferenceName (vx_reference *ref*, const vx_char * *name*)

Name a reference

This function is used to associate a name to a reference. This name can be used by the OpenVX implementation in log messages and any other reporting mechanisms. It is also intended to be used by `vxGetImportReferenceByName` to retrieve a named reference from a vx_import object.

The OpenVX implementation will not check if the name is unique in the reference scope (context or graph). Several references can then have the same name.

Parameters

in	*ref*	The reference to name.
in	*name*	Pointer to the '\0' terminated string that identifies the reference. The string is copied by the function so that it stays the property of the caller. NULL means that the reference is not named.

Returns

A vx_status_e enumeration.

Return values

VX_SUCCESS	No errors.
VX_ERROR_INVALID_REFERENCE	if reference is not valid.

4.1.4.4 vx_reference vxGetImportReferenceByName (vx_import *import*, const vx_char * *name*)

Used to retrieve a reference by name from the import when the name is known beforehand. If multiple references have the same name, then *any* one of them may be returned.

Parameters

in	*import*	The reference to the import object.
in	*name*	The reference string name.

Returns

vx_reference

Return values

0	Invalid import object or name does not match a reference in the import object.
*	The reference matching the requested name.

Note

Use vxReleaseReference to release the reference before releasing the context.

Precondition

vxImportFromXML

4.1.4.5 vx_reference vxGetImportReferenceByIndex (vx_import *import*, vx_uint32 *index*)

Used to retrieve a reference by the index from the import.

Parameters

in	*import*	The reference to the import object.
in	*index*	The index of the reference in the import object to return.

Returns

vx_reference

Return values

0	Invalid import object or index.
*	The reference at the requested index number.

Note

Use vxQueryImport with VX_IMPORT_ATTRIBUTE_COUNT to retrieve the upper limit of references in the import.
Use vxReleaseReference to release the reference before releasing the context.

Precondition

vxImportFromXML

4.1.4.6 vx_status vxQueryImport (vx_import *import*, vx_enum *attribute*, void * *ptr*, vx_size *size*)

Used to query the import about its properties.

Parameters

in	*import*	The reference to the import object.
in	*attribute*	The vx_import_attribute_e value to query for.
out	*ptr*	The location at which the resulting value will be stored.
in	*size*	The size of the container to which ptr points.

Returns

A vx_status_e enumeration.

Precondition

vxImportFromXML

4.1.4.7 vx_status vxReleaseImport (vx_import * *import*)

Releases a reference to an import object. Also internally releases its references to its imported objects. These imported objects may not be garbage collected until their total reference counts are zero.

Parameters

in	*import*	The pointer to the import object to release.

Returns

A vx_status_e enumeration.

Return values

VX_SUCCESS	No errors.
VX_ERROR_INVALID_REFERENCE	If import is not a vx_import.

Note

After returning from this function the reference will be zeroed.

Precondition

vxImportFromXML

4.1.4.8 vx_status vxReleaseReference (vx_reference * *ref*)

Releases a reference. The object may not be garbage collected until its total reference count is zero.

Parameters

in	*ref*	The pointer to the reference to release.

Returns

A `vx_status_e` enumeration.

Return values

VX_SUCCESS	No errors.
VX_ERROR_INVALID_REFERENCE	If reference is not a `vx_reference`.

Note

After returning from this function the reference will be zeroed.

Precondition

`vxGetImportReferenceByName` or `vxGetImportReferenceByIndex`

Index