

# CxF2 - Schema Outline

---

Version 1.2

## Versions

<u>Version</u>	<u>Description</u>	<u>Date</u>	<u>Author</u>
1.1	Initial version	2008-11-04	tk
1.2	draft modifications	2008-11-17	r fw

## Contents

1	Preface .....	3
2	Notation .....	3
3	CxF Root Element .....	3
4	Color .....	4
5	Color Space Type .....	5
6	Color Space Specification .....	6
6.1	Value Types and Related Space Types .....	6
6.2	Spectrum Tristimulus .....	7
6.3	Spectrum Generic .....	7
7	Palette .....	8
8	Color Set.....	8
9	Collection Color Space Specification.....	9
10	Color Quality Control.....	9
10.1	Standard and Measurement .....	10
10.2	Measurement .....	10
10.3	Standard.....	11
10.4	Physical Sample .....	11
10.5	Tolerance .....	12
10.6	Function Types .....	12
10.7	Limits .....	13
10.8	Illuminants.....	14
11	Preamble.....	15
11.1	Header .....	15
11.2	Document Version.....	16
12	Common Elements .....	17
12.1	Custom Attribute .....	17

12.2	Timestamp .....	18
12.3	Owner .....	18
12.4	Semantic Association .....	18
12.5	Field of View .....	18
12.6	Measurement Device .....	19
12.7	Substrate Types .....	20
13	Common Attributes .....	20
13.1	UniqueID .....	20

## 1 Preface

This outline is based on documents available from [www.colorexchangeformat.com](http://www.colorexchangeformat.com):

- Cxv XML Schema
- Cxv White paper; Release 2.0, March 2008
- Cxv Required and Best Practice Guidelines Schema version 2.0
- Standard Colour Exchange Format 2.0

## 2 Notation

{ name }	component
{ name } ?	optional component
{ name } +	component with one or more occurrences
{ name } *	component with zero or more occurrences

## 3 Cxv Root Element

A Cxv element contains a preamble optionally followed by:

- a sequence of colors
- a sequence of palettes (containing colors)
- a sequence of color quality controls (containing Standard colors and associated sample colors)

```
<Cxv xml ns=http://colorexchangeformat.com/v2
  xml ns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  {Preamble}
  {Color} *
  {Palette} *
  {ColorQualityControl} *
</Cxv>
```

According to the 'Best Practice Guidelines':

- Cxv is the only supported root element
- The most natural Cxv sections should be used to store application data:
 

One or more unrelated colors:	sequence of colors
Group(s) of colors:	palette(s), can be subgrouped using ColorSets
Standards with tolerances, associated samples:	color quality control
- Use of more than one type of section is not recommended (use only one type of section in a Cxv file, but can have multiple instances of that type)

## 4 Color

A color has a name, at least one timestamp entry, and at least one color space type containing color values (the type of color values are determined by the type of Color Space specified).

```
<Col or Col orName="...">
  {ti mestamp} +
  {col or space type} +
</Col or>
```

The {color space type} element(s) define the type of color value data they contain (spectral or colorimetric) . Multiple color space elements may be included to provide different types of colorspace data (for instance: Spectral, CIELab, and XYZ data ).

The color element may contain two additional string attributes: unique ID and color number.

The color element may contain additional child elements: custom attribute, owner, semantic association, and substrate.

```
<Col or Col orName="..." [Uni queI D="..." ] [Col orNumber="..." ]>
  {custom attri bute} *
  {ti mestamp} +
  {owner} ?
  {semanti c associ ati on} *
  { col or space type} +
  {substrate} ?
</Col or>
```

## 5 Color Space Type

The {color space type} defines the type of color space and the parameters it can contain. A color space type can be either spectral (ColorSpaceSpectral) or one of many types of colorimetric color spaces (example: ColorSpaceCIELab, or ColorSpaceCIEXYZ).

```
<Col orSpacexxxx [Name=". . ." ] [Comments=". . ." ]>
  {custom attribute} *
  {timestamp} +
  {col or space speci fi cati on}
  {col or space parameter} +
</Col orSpacexxxx>
```

Examples:

```
<Col orSpaceCI EXYZ>
  <ChangeHi story>2000-04-06T09: 15: 27. 000-04: 00</ChangeHi story>
  <Col orSpaceSpeci fi cati onSpectrumTri sti mul us xsi : ni l =" true" />
  <X>0. 1</X>
  <Y>0. 7</Y>
  <Z>0. 3</Z>
</Col orSpaceCI EXYZ>
```

```
<Col orSpacePANTONE>
  <ChangeHi story>2000-04-06T09: 15: 27. 000-04: 00</ChangeHi story>
  <Fami l yName>spri ng</Fami l yName>
  <Col orName>
    <Prefi xName>PANTONE</Prefi xName>
    <BaseName>100</BaseName>
    <Suffi xName>C</Suffi xName>
  </Col orName>
  <Page>67-A</Page>
  <Gamut>Gamut_CMYK</Gamut>
</Col orSpacePANTONE>
```

### 5.1 Color Space Specification

The {color space specification} element starts with the prefix ColorSpaceSpecification and is required for all types except ColorSpacePANTONE. It defines the specific type of measurement and/or calculation conditions for the parameters given in the color space type. Two types are available, ColorSpaceSpecificationSpectrumSpectral (for defining the measurement conditions) or ColorSpaceSpecificationSpectrumTristimulus (for defining the measurement and calculation conditions). The {color space specification} may be nilled. A nilled value indicates that the color is using the default ColorSpaceSpecification defined in the encompassing palette or colorset.

Example:

```
<Col orSpaceSpeci fi cati onSpectrumSpectral >
  <ChangeHi story>2000-04-06T09: 15: 27. 000-04: 00</ChangeHi story>
  <CxF: Devi ceFi l ter>Fi l ter_UVExcl uded</CxF: Devi ceFi l ter>
  <CxF: Devi cePol ari zati on>fal se</CxF: Devi cePol ari zati on>
  <CxF: Spectrum>Spectrum_Refl ectance</CxF: Spectrum>
  <CxF: GeometryChoi ce>
    <CxF: Angl e>45</CxF: Angl e>
  </CxF: GeometryChoi ce>
</CxF: Col orSpaceSpeci fi cati onSpectrumSpectral >
```

## 6 Color Space Specification

### 6.1 Value Types and Related Space Types

<u>{color space} element name</u>	<u>{color space specification} element name</u>
ColorSpaceRGB	ColorSpaceSpecificationSpectrumTristimulus
ColorSpaceSRGB	“
ColorSpaceAdobeRGB	“
ColorSpaceAdobeWideGamutRGB	“
ColorSpaceCIELab	“
ColorSpaceCIELCh	“
ColorSpaceCIEXYZ	“
ColorSpaceCIExyY	“
ColorSpaceCIELuv	“
ColorSpaceRecipe	ColorSpaceSpecificationSpectrumGeneric
ColorSpaceGeneric	“
ColorSpaceDensity	“
ColorSpaceHSL	“
ColorSpaceHSV	“
ColorSpaceCMYK	“
ColorSpacePANTONEHexachrome	“
ColorSpaceYUV	“
ColorSpaceYIQ	“
ColorSpaceYBR	“
ColorSpaceNCS	“
ColorSpaceMunsell	“
ColorSpaceSpectral	ColorSpaceSpecificationSpectrumSpectral
ColorSpaceEmissiveGeneric	ColorSpaceSpecificationEmissiveGeneric
ColorSpaceEmissiveSpectral	ColorSpaceSpecificationEmissiveSpectral
ColorSpaceEmissiveCIEXYZ	ColorSpaceSpecificationEmissiveTristimulus
ColorSpaceEmissiveCIExyY	“
ColorSpacePANTONE	n/a

## 6.2 Spectrum Tristimulus

```
<ColorSpaceSpecificationSpectrumTristimulus>
  {custom attribute} *
  {profile - internal base64 or external uri} ?
  {device} ?
  {device filter}
  {device polarization}
  {timestamp} +
  {spectrum type}
  {illumination options}
  {field of view}
  {geometry choice} ?
  {ASTM table} ?
  {gamma} ?
</ColorSpaceSpecificationSpectrumTristimulus>
```

## 6.3 Spectrum Generic

```
<ColorSpaceSpecificationSpectrumGeneric>
  {custom attribute} *
  {profile - internal base64 or external uri} ?
  {device} ?
  {device filter}
  {device polarization}
  {timestamp} +
  {spectrum type}
  {geometry choice} ?
  {illumination options} ?
  {field of view} ?
  {ASTM table} ?
  {gamma} ?
  {density status} ?
</ColorSpaceSpecificationSpectrumGeneric>
```

The “spectrum generic” element has the same components as [spectrum tristimulus](#) plus an additional {density status} component. Some of the components are required for tristimulus but optional for the generic variant.

## 7 Palette

A {palette} element has a name attribute, at least one timestamp entry, an optional collection color space specification that may be inherited by colors within the palette, and a possibly empty sequence of color sets.

```
<Pal ette Pal etteName="...">
  {ti mestamp} +
  {col lecti on col or space speci fi cati on} ?
  {col or set} *
</Pal ette>
```

The {palette} element may contain additional attributes and elements:

```
<Pal ette Pal etteName="..." [Pal ettePartNumber="..."] [Pal etteVersi on="..."]
  [Uni queID="..." [Comments="..."]>
  {custom attribute} *
  {ti mestamp} +
  {owner} ?
  {semanti c associ ati on} *
  {col lecti on col or space speci fi cati on} ?
  {col or set} *
</Pal ette>
```

## 8 Color Set

A {color set} element has a name attribute, at least one timestamp entry, an optional collection color space specification that may be inherited by colors in the set, and a possibly empty sequence of colors.

```
<Col orSet Col orSetName="...">
  {ti mestamp} +
  {col lecti on col or space speci fi cati on} ?
  {col or} *
</Col orSet>
```

The {color set} element may contain additional attributes and elements:

```
<Col orSet Col orSetName="..." [Uni queID="..."] [PartNumber="..."] [Comments="..."]>
  {custom attribute} *
  {ti mestamp} +
  {owner} ?
  {semanti c associ ati on} *
  {col lecti on col or space speci fi cati on} ?
  {col or} *
  {substrate} ?
</Col orSet>
```



## 9 Collection Color Space Specification

A {collection color space specification} provides specifications that may be inherited by {color space} elements within a {color set}.

```
<Col I ecti onCol orSpaceSpeci fi cati on>
  {Col orSpaceSpeci fi cati onSpectrumGeneri c} ?
  {Col orSpaceSpeci fi cati onSpectrumSpectral } ?
  {Col orSpaceSpeci fi cati onSpectrumTri sti mul us} ?
  {Col orSpaceSpeci fi cati onEmi ssi veGeneri c} ?
  {Col orSpaceSpeci fi cati onEmi ssi veSpectral } ?
  {Col orSpaceSpeci fi cati onEmi ssi veTri sti mul us} ?
</Col I ecti onCol orSpaceSpeci fi cati on>
```

A {color space type} inherits a specification from the {collection color space specification} of its {color set} or {palette} if it nulls its {color space specification} element.

A specification within a {collection color space specification} of a {color set} overrides a specification of the same type contained in the {collection color space specification} of the enclosing {palette}.

Example:

```
<Col I ecti onCol orSpaceSpeci fi cati on>
  <Col orSpaceSpeci fi cati onSpectrumSpectral >
    <Devi ceFi lter CustomOrParti al ="NA">Fi lter_None</Devi ceFi lter>
    <Devi cePol ari zati on>fal se</Devi cePol ari zati on>
    <ChangeHi story>2008-06-05T09: 07: 44. 293-04: 00</ChangeHi story>
    <Spectrum>Spectrum_Refl ectance</Spectrum>
    <GeometryChoi ce>
      <Angl e>45. 0</Angl e>
    </GeometryChoi ce>
  </Col orSpaceSpeci fi cati onSpectrumSpectral >
</Col I ecti onCol orSpaceSpeci fi cati on>

<!-- A color that inherits the spectral specification for its spectral value -->
<Col or Col orName="pi nk">
  <ChangeHi story>2000-04-06T09: 15: 27. 000-04: 00</ChangeHi story>
  <Col orSpaceSpectral >
    <ChangeHi story>2000-04-06T09: 15: 27. 000-04: 00</ChangeHi story>
    <Col orSpaceSpeci fi cati onSpectrumSpectral xsi : ni l ="true"/>
    <Refl ectancePoi nt Wavel ength=" 400. 0" > 0. 05175</Refl ectancePoi nt>
    <Refl ectancePoi nt Wavel ength=" 410. 0" > 0. 04265</Refl ectancePoi nt>
    <Refl ectancePoi nt Wavel ength=" 700. 0" > 0. 74884</Refl ectancePoi nt>
  </Col orSpaceSpectral >
</Col or>
```

## 10 Color Quality Control

A {color quality control} element represents one physical sample type and associated standard, measurement(s) and tolerance(s). Each standard and measurement element may contain one standard and multiple samples associated with that standard. A single ColorQualityControl element may contain multiple “standard and measurement” elements to contain multiple standards (but should consist of the same physical sample type (example: knit fabric)).

```

<ColorQualityControl >
  {timestamp} +
  {standard and measurement} +
  {physical sample}
</ColorQualityControl >

```

The {color quality control} element may contain additional attributes and elements:

```

<ColorQualityControl [Name="..." [Description="..."] [UniqueID="..."]
  [Comments="..."]>
  {custom attribute} *
  {timestamp} +
  {standard and measurement} +
  {physical sample}
  {tolerance} *
</ColorQualityControl >

```

## 10.1 Standard and Measurement

```

<StandardAndMeasurement>
  {custom attribute} *
  {measurement} *
  {standard}
</StandardAndMeasurement>

```

The {standard} element may be nilled: <Standard xsi:nil="true"/>

A nilled standard indicates that the standard associated with the samples is not specified or included in the CxF file, but all samples are QC measurements that should be associated together for QC purposes.

## 10.2 Measurement

A {measurement} element contains one or more color values (within its {color} element) for a single sample. For instance, the measurement device may return reflectance data for multiple angles. The specification of the measurement device is required.

```

<Measurement>
  {custom attribute} *
  {measurement device}
  {color}
  {function} *
</Measurement>

```

### 10.3 Standard

A {standard} element specifies the color of the item to be measured.

```
<Standard>
  {custom attribute} *
  {measurement device}
  {color}
  {tolerance} *
</Standard>
```

### 10.4 Physical Sample

The {physical sample} may consist of a single patch, or it may consist of a named target (array of patches) used to make profiles, for instance.

```
<Physical Sample>
  {custom attribute} *
  {coating} ?
  {process} ?
  {sample}
</Physical Sample>
```

The {sample} element is either a {SampleSpot} or {SampleTarget} element.

```
<SampleSpot | SampleTarget>
  {custom attribute} *
  {substrate}
  {backing} ?
  {manufacturer} ?
  {production date} ?
  {serial number} ?
  {comment} ?
  {specification or target}
</SampleSpot | SampleTarget>
```

The {specification or target} element depends on the parent type.

```
<SampleSpot>
  ...
  <Specification>
    <SampleWidth>2.0</SampleWidth>
    <SampleHeight>3.8</SampleHeight>
  </Specification>
</SampleSpot>

<SampleTarget>
  ...
  <Target>Target_IT8.7/1</Target>
</SampleTarget>
```

The Target element within {SampleTarget} contains a string from the following enumeration:

```
Target_I T8. 7/1
Target_I T8. 7/2
Target_I T8. 7/3
Target_I T8. 7/4
Target_ECI 2002
Target_Custom
```

## 10.5 Tolerance

```
<Tolerance>
  {custom attribute} *
  {function}
  {limits} ?
  {primary illuminant} ?
  {secondary illuminant} ?
  {tertiary illuminant} ?
  {field of view} ?
</Tolerance>
```

## 10.6 Function Types

A {function} element starts with the prefix Function.

```
<Functionxxxx>
  {custom attribute} *
  {result} ?
  {type specific parameter} *
</Functionxxxx>
```

The {result} element is named Result and contains a double value.

Function types that do not have any {type specific parameter}:

```
FunctionAverageStdDevA
FunctionAverageStdDevB
FunctionAverageStdDevL
FunctionDE
FunctionMeanDE
FunctionStdDevA
FunctionStdDevB
FunctionStdDevC
FunctionStdDevH
FunctionStdDevL
FunctionStdDevX
FunctionStdDevY
FunctionStdDevZ
```

Function type with additional parameters *Param\_l* , *Param\_c*:

```
FunctionDECMC
```

Note that the *l* and *c* parameters are required although the schema specifies default values 2.0, 1.0.

Function types with additional elements *Param\_l* , *Param\_c* , *Param\_h*:

```
FunctionDE2000
```

```
FunctionDE94
```

Note that the *l* , *c* , *h* parameters are required although the schema specifies default values 2.0, 1.0, 1.0.

Generic function type with a required {Name} element and an optional sequence of {Parameters} elements:

```
FunctionGeneric
```

Some examples:

```
<FunctionDE>
  <Result>0.37</Result>
</FunctionDE>
```

```
<FunctionDE2000>
  <Result>0.37</Result>
  <Param_l>2.0</Param_l>
  <Param_c>1.0</Param_c>
  <Param_h>1.0</Param_h>
</FunctionDE2000>
```

```
<FunctionGeneric>
  <Result>0.37</Result>
  <Name>Foo</Name>
  <Parameters>
    <Name>alpha</Name><ValueChoice><Double>1.7</Double></ValueChoice>
  </Parameters>
  <Parameters>
    <Name>beta</Name><ValueChoice><Double>0.05</Double></ValueChoice>
  </Parameters>
</FunctionGeneric>
```

## 10.7 Limits

```
<Limits>
  {custom attribute} *
  {high tolerance}
  {low tolerance} ?
</Limits>
```

The {high tolerance} element is named *HighTolerance* and contains a double value.

The {low tolerance} element is named *LowTolerance* and contains a double value.

## 10.8 Illuminants

The {primary illuminant}, {secondary illuminant}, and {tertiary illuminant} elements are named PrimaryIlluminant, SecondaryIlluminant, and TertiaryIlluminant.

They contain a string from the following enumeration:

- Illuminant\_A
- Illuminant\_B
- Illuminant\_C
- Illuminant\_D50
- Illuminant\_D55
- Illuminant\_D65
- Illuminant\_D75
- Illuminant\_E
- Illuminant\_F2
- Illuminant\_F3
- Illuminant\_F7
- Illuminant\_F11
- Illuminant\_F12
- Illuminant\_9300
- Illuminant\_Custom

## 11 Preamble

Each CxF document has a preamble. The {header} element is required, while the {owner} and {document version} are optional.

```
<Preamble>
  {header}
  {owner} ?
  {document version} ?
</Preamble>
```

### 11.1 Header

A minimal {header} consists of a creator string, a creator version string, a timestamp entry, and a purpose string.

```
<Header>
  {creator}
  {creator version}
  {timestamp} +
  {purpose}
</Header>
```

A {header} may contain additional string elements, and a sequence of {revision} elements.

```
<Header>
  {custom attribute} *
  {creator}
  {creator version}
  {timestamp} +
  {purpose}
  {header name} ?
  {header comment} ?
  {read-SDK information} ?
  {write-SDK information} ?
  {read-platform information} ?
  {write-platform information} ?
</Header>
```

## 11.2 Document Version

The {document version} element contains a version string, and an optional sequence of URIs locating previous versions of this document or locations of documents from which this data was derived.

```
<Version>  
  {custom attribute} *  
  {version}  
  {predecessor URI} *  
</Version>
```



## 12 Common Elements

### 12.1 Custom Attribute

A {custom attribute} is a named value with a specified data type. It may be used to express properties related to the parent element.

```
<CustomAttribute>
  {name}
  {attribute value}
</CustomAttribute>
```

The {attribute value} element is named {ValueChoice} and contains a typed value.

```
<ValueChoice>
  {typed value}
</ValueChoice>
```

The following {typed value} elements are available: DoubleValue, IntegerValue, StringValue.

Examples:

```
<CustomAttribute>
  <Name>Z position</Name>
  <ValueChoice><DoubleValue>4.16</DoubleValue></ValueChoice>
</CustomAttribute>
```

```
<CxF: CustomAttribute>
  <CxF: Name>iQC_Tag06</CxF: Name>
  <CxF: ValueChoice>
    <CxF: StringValue>SEASON=SUMMER 2009</CxF: StringValue>
  </CxF: ValueChoice>
</CxF: CustomAttribute>
```

## 12.2 Timestamp

A {timestamp} element contains a timestamp with a time-zone value. It is named {ChangeHistory}.

Example:

```
<ChangeHistory>2008-06-05T09:07:44-04:00</ChangeHistory>
```

The timestamp syntax is a restriction of xs:dateTime.

The first {timestamp} element in a sequence represents the **creation** time. Additional {timestamp} elements represent **modification** times.

## 12.3 Owner

An {owner} element specifies ownership information related to the parent element.

```
<Owner>
  {custom attribute} *
  {person} *
  {copyright} *
  {company} *
</Owner>
```

## 12.4 Semantic Association

A {semantic association} is an element that may be used to specify terms associated with the parent element, typically related to alternate color names, generic color descriptors, or classifications.

```
<SemanticAssociation>Text...</SemanticAssociation>
```

## 12.5 Field of View

A {field of view} element contains a string value from the following enumeration:

```
FieldOfView_2_Degree
FieldOfView_10_Degree
```

## 12.6 Measurement Device

A {measurement device} specifies a manufacturer, a model, and a serial number:

```
<Device>
  {manufacturer}
  {model}
  {serial number}
</Device>
```

Additional elements may be provided:

```
<Device>
  {custom attribute} *
  {manufacturer}
  {model}
  {serial number}
  {device class} ?
  {firmware version} ?
  {boot firmware version} ?
  {calibration state} ?
  {device settings} ?
</Device>
```

The {calibration state} element may specify a **CalibrationDateTime** attribute, and contains one of the following strings:

CalibrationState\_NotCalibrated, CalibrationState\_Calibrated, CalibrationState\_NA

The {device settings} element contains a variety of optional components:

```
<DeviceSettings>
  {custom attribute} *
  {aperture} ?
  {lens position} ?
  {glass compensation} ?
  {gloss compensation} ?
  {averaging} ?
  {NetProfiler} ?
</DeviceSettings>
```

{aperture (in mm)}, {lens position (in mm)} and {averaging} specify numerical values. These values are restricted to integer.

{glass compensation}, {gloss compensation} and {NetProfiler} specify boolean values.

## 12.7 Substrate Types

A {substrate} element name starts with “Substrate” followed by a type qualifier. A {substrate} may contain custom attributes, a part number, a color, and substrate type dependent parameters.

```
<Substrate-type>
  {custom attribute} *
  {part number} ?
  {color} ?
  {substrate parameter} *
</Substrate-type>
```

List of substrate types with related parameters:

<b><u>Substrate type</u></b>	<b><u>Parameters</u></b>
SubstrateGeneric	Type
SubstratePaper	PaperThickness, PaperType
SubstrateMetal	MetalWeight, MetalType
SubstratePlastic	PlasticWeight, PlasticType
SubstrateWood	WoodWeight, WoodType
SubstrateVinyl	VinylWeight, VinylType
SubstrateTile	TileWeight, TileType
SubstrateFilm	FilmWeight, FilmType
SubstrateTextile	TextileWeight, TextileType

All parameter elements except the {SubstrateGeneric} Type parameter are optional.

## 13 Common Attributes

### 13.1 UniqueID

UniqueID is an optional attribute of various elements. It is typically a UUID (such as Microsoft GUID), but the schema allows any string content. It is primarily designed as a method of indicating a globally unique ID which can be used to identify this specific color object even when names are not unique. If a unique ID is assigned to an object when it is created, then that unique ID identifies that object and remains constant throughout its life. Modifying the object does not change its unique ID unless the intent is to create a new object from this object. A unique ID allows software to import color objects from foreign sources and determine whether the incoming objects already exist in the local database (in which case the software may need to check ChangeHistory to determine which copy of the object should be kept or whether a merge of the two objects should take place.