CxF3 Schema Overview

Version 1.0

Contents

1 Preface

CxF3 is a color exchange format that grew out of the CxF2 proposal, but greatly modified to address the concerns and issues with creating, using, and exchanging color information in the CxF2 format. The major issues with CxF2 were:

- Terminology – many terms were confusing, did not fit standard terminology, and names used for elements or enumerations were often too long and difficult to understand.
- Size – Due to the organization, limited ability to use a reference mechanism for common or shared information, and use of long element names most CxF2 files were very large (many megabytes in size), requiring the use of compression to make the size reasonable. While compression is provided in the SDK – not all applications supported it, and using CXFZ compressed files required first uncompressing the file before using the CxF file – an operation which complicates using XML files with standard tools like stylesheets and XML parsers.
- Organization – CxF2 required putting the color data into either Palettes or QualityControl sections – imposing an organizational structure on the information which was not always meaningful to the user or reading application. CxF should be able to easily exchange color information between applications of very different purposes without those applications being aware of the other . In addition, the structure and organization of CxF2 was generally confusing to most users and difficult to parse with standard XML tools and apply simple generic stylesheets.
- Customization – Most applications "customized" their use of CxF2 by putting lots of "CustomAttributes" into each Color element, and imposing special "meaning" to having information in specific locations or particular order. This "custom information" tended to be meaningful ONLY to the creating application and made parsing by other applications difficult and subject to misreading the critical information.

2 Division into Resources and CustomResources

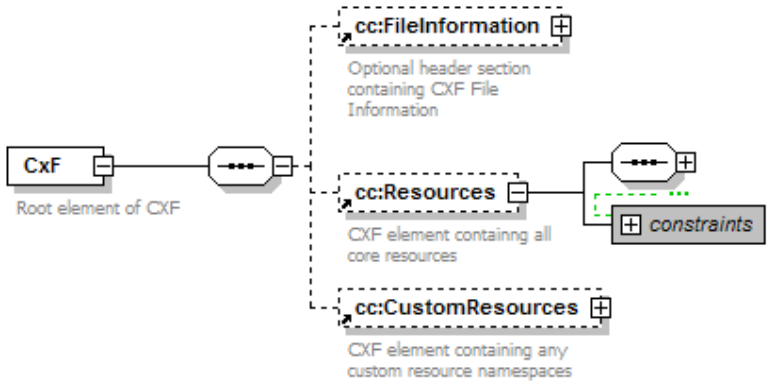
2.1 Document Division

The information contained in a CxF3 document is divided into **File Information**, **Resources** and **CustomResources**. A CxF3 file may contain one optional File Information header, contains at most ONE resource, and MAY contain zero to many CustomResources. The File_Information and Resources are described by the "CxF3_Core" schema (CxF3_Core.xsd) and namespace, while the CustomResources are each defined by their own custom schema and namespace as developed for that specific CustomResource.

File Information header (optional) contains information regarding this CXF3 file including creation date, creator, description, and any custom tag information specific to the file itself.

Resources are well defined "core" color data that are generally of interest to ALL applications. It contains a list of "color Objects" that each contain the spectral and/or other color values, names, dates, specifications, etc.

CustomResources reference the Resources and can add semantics, structures, and custom data generally targeted towards a specific usage or type of application. An example might be the "layout" of the color data contained in the Resources by arranging them in a specific way into pages, columns, and rows – and could include additional information like page names, cross reference names, etc. In this manner, application(s) which need to use this CxF file to convey an arrangement of color page information can do so without affecting the ability of other types of applications to read the same "core" color objects and use them in whatever manner is most fitting for that reading application. Applications which need to arrange the color information into a color page arrangement would need to understand the Custom "colorpage.xsd" schema – but other applications which have no interest in using the color information in this manner need only know about and support the standard CxF3_Core schema and namespace.



Note 1: Objects within the Resource may reference other objects within the Resource, but cannot reference CustomResources or data within CustomResources.

Note 2: CustomResources are typically independent from one another. A CxF3 file may include more than one CustomResource.

2.2 Document Modification

When making modifications to an existing CXF3 document – certain rules apply. Resources may be added but not removed unless one can assert that a resource is neither used by some CustomResource nor by some other resource object.

Extending or modifying an existing resource may be critical: the meaning of the resource could be affected. This is especially true for making changes to a resource that has a GUID – since that GUID identifies this object in a global nature and other applications may use this information to update existing objects within their own system.

CustomResources may be added or removed provided that the CustomResource is not referenced by some other CustomResource. CustomResources may be modified.

2.2 Document Rules for Creation

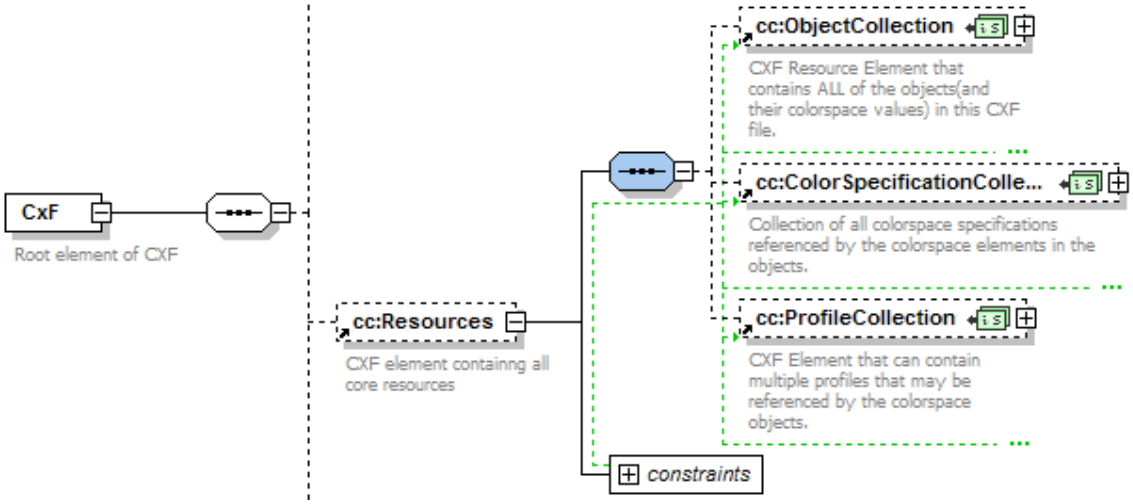
CxF is meant to be a file format for exchanging color information between applications of different and unknown type … as such you cannot depend on a target application having specific knowledge about your applications behavior when it reads the contents of a CXF file. To assure the ability for unknown target applications to properly read the information contained in a CXF file – the following rules should be followed when creating CXF documents:

- Use GUIDs only when the preservation of identity is critical within the application or workflow – otherwise leave them blank and let the receiving application treat them as new objects.
- Do not duplicate objects unless your intent is that they represent different unique objects within your system – duplicated objects should never have the same GUID or unique Id within the same CXF3 file.
- Information that is defined by the Core schema and required by a CustomResource should be referenced by that CustomResource and not duplicated within it (or defined within the CustomResource and omitted from the core resources).

3 Core Resources Data Model

The Resources section of a CxF3 file contains 3 collections –

- **ObjectCollection** – all color "objects". These are each identified by a unique ID within the CXF3 file. This ID is used by CustomResources to reference the color object. An "object" is the CXF3 representation of a physical or virtual "color object" for which color information (name, measurement or creation date, spectral values, color values, tags, physical attributes, etc.) may exist.
- **ColorSpecificationCollection** – a list of all color specifications (illuminant, observer, measurement specifications, etc.) that are used by the color objects. They each have an ID that is unique within the CXF3 file and are referenced by the color values within color objects. In this way multiple color objects (and color values) can share a single color specification – reducing the size of the CXF3 file.
- **ProfileCollection** – a list of all profiles (input/output) that are referenced by the color objects and provide the data that is used (or was used) to modify/profile the color values to/from specific devices or display spaces.



- ### 3.1 Object

Color "objects" are the focus of CxF3 – and contain (or reference) the entire core related color data (name, date, color values, specifications, etc.) pertaining to that object. All color objects are contained in a single ObjectCollection – so all applications that read a CxF3 file need ONLY to look in a single location for all the color data in the CxF3 file. Note: There is no particular importance placed in the order of the objects listed – any such "organizational" information should be contained in a "customresource".

Objects contain attributes:

- **Name** (required) – String containing formal name identifying this object.
- **Id** (required) – xs:NCName string containing a unique reference key ID used to reference this object within this CxF file. This name can be a simple number or a more compound string construction – but must follow the rules for XML NCName type element and must be unique within this file.
- **ObjectType** (required) – String identifying what type of object (in a general context) this is (Standard, Trial, Target, Substrate, Colorant, etc.). **May be blank** to indicate there is no implied generic context (each application is free to treat the object in whatever manner it chooses).
- **GUID** – An optional element that is used to identify this color object in a global manner (unique across all instances of this object worldwide).

Objects contain elements:

- **CreationDate** (required) – The creation date/time of the object (and default for measurement date/time). Given as an XS:DateTime format with or without Time Zone information included.
- **ColorValues** (optional) – list of one or more types of device independent color value elements (spectral, CIELab values, CIEXYZ values, sRGB, etc.)
- **ColorDifferenceValues** (optional) – list of one or more sets of DeltaCIELab or DeltaCustom color difference elements for CIELab or Custom color differences.

- **DeviceColorValues** (optional) – list of one or more types of device dependent color value elements (RGB, CMYK, HTML, ColorNotation, PantoneHexachrome, etc.)
- **TagCollection** (optional) – contains a group of "tags" which contain custom name/value data for this object. An object can contain multiple "named" TagCollections" in order to separate tag groups into distinct uses or types of information.
- **PhysicalAttributes** (optional) – contains information which describes the non-colorimetric qualities of the object (dimensions, quantity, weight, thickness, gloss, finish, substrate, image, opacity, custom attributes).

3.1.1 Color Values Element

Each element (of the following types) within this group contains a name (to help distinguish this element instance from others of the same type within the same ColorValues element), a ColorSpecification reference, a measurement date (optional), and a StartingWL (optional):

- ReflectanceSpectrum – list of reflectance values in equal increments of 1,2,5,10, or 20 nm
- TransmittanceSpectrum - list of transmittance values in equal increments of 1,2,5,10, or 20 nm
- EmissiveSpectrum– list of emissive spectral values in equal increments of 1,2,5,10, or 20 nm
- CustomSpectrum – collection of spectral values given at specific wavelengths
- ColorSRGB – device independent sRGB data (*by definition sRGB is always D65-2*).
- ColorAdobeRGB – device independent RGB data in Adobe format.
- ColorCIELab
- ColorCIELch
- ColorCIEXYZ
- ColorEmissiveCIEXYZ
- ColorCIExyY
- ColorEmissiveCIExyY
- ColorCIELuv
- ColorDensity – a Density value and status/filter/base offset used.
- PrivateSpectrum – an encrypted set of spectral values for private exchange.
- PrivateColorValues – an encrypted set of color space values (CIELab, etc.)

3.1.2 Color Difference Values Element

Each element (of the following Color Difference types) within this group contains a name (to help distinguish this element instance from others within the same ColorDifferenceValues element), a ColorSpecification reference, and a StandardRef reference to a color object that was used as the standard.

- DeltaCIELab – multiple color differences and colorspecification reference for CIELab color differences. Holds values for DL,Da,Db, DC, DH, DE, DEcmc, DE94, and DE2000.
- DeltaCustom – multiple custom color difference values for non-CIELab color values.

3.1.3 Device Color Values Element

Each element (of the following types) within this group contains a name, a colorspecification reference, a profile reference, and one or more device dependent color value types for this object:

- ColorHTML – web page syntax for displaying RGB values in hexadecimal form.
- ColorNotation – name and string notation (ex: Munsell code) for this color.
- ColorRGB – device specific RGB values.
- ColorHSL – device specific hue, saturation, lightness values.
- ColorCMYK – device specific CMYK values.
- ColorCMYKPlusN – device specific CMYK (plus additional colors) values.
- ColorPantoneHexachrome – device specific PantoneHexachrome values.
- ColorRecipe – device specific color recipe information (colorants and percentages).
- ColorCustom – device specific custom color values.
- PrivateColorValues – encrypted device specific color values for private exchange.

3.1.4 Tag Collection (may have more than one per color object)

A named collection of one or more Tags, each Tag holding a name/value pair. Tag Collections are used for storing groups of related meta-data for this color object. The data held within these tags is meant to be exchanged with the color object – but is not data which is "colorimetric" in nature, nor data which is directly related to

physical attributes of the object. Examples might be “customer name”, “season”, or “contact email address”.

3.1.5 PhysicalAttributes

An element that contains optional data specific to the physical nature of this color object, taken from the following list:

- TargetType – enumerated list of IT8 target types.
- FinishType – enumerated list of finishes (coated, matte, glossy, etc.)
- SubstrateType – enumerated list of substrate types (wood, paper, textile, etc.)
- Quantity
- Width
- Length
- Height
- Thickness
- Gloss
- Opacity
- CustomAttributeString
- CustomAttributeValue
- Image (binary or URI reference)

3.2 ColorSpecification

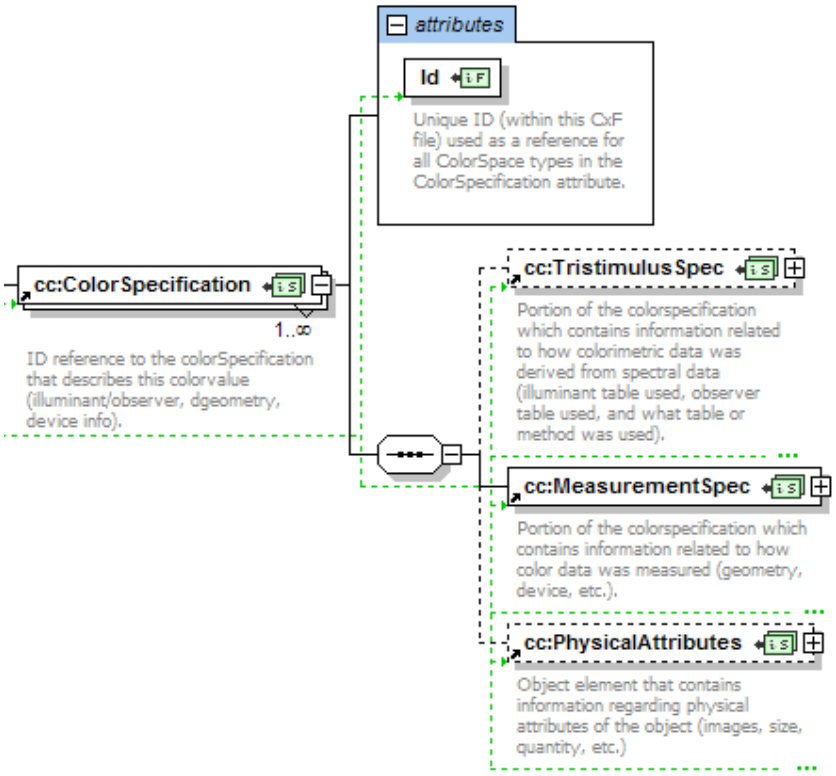
Color Specifications are contained within the ColorSpecificationCollection and contain the color related specifications that apply to the color values within the color objects. Each color value type contains an attribute named “ColorSpecification” that references the ColorSpecification ID that applies to it. The ColorSpecification contains such information as illuminant/observer, measurement geometry, measurement device information, and other data which pertains to the manner in which the measurement was made or the calculations performed.

ColorSpecifications contain attributes:

- Id (required) – the unique string used to reference this ColorSpecification object.

ColorSpecifications contain elements:

- TristimulusSpec (optional) – contains information regarding colorimetric/tristimulus calculations such as illuminant, observer, method/table used. Can also contain custom illuminant data and SPD.
- MeasurementSpec (required) – contains information regarding measurement type (emissive, reflectance, transmittance, etc), geometry, wavelength range, device characteristics, and calibration standard used.
- PhysicalAttributes (optional) – can contain physical attribute data that is common to all referencing color objects.



•

3.2.1 TristimulusSpec

Contains the information used to process spectral values into other color value types. Contains name and optional X,Y,Z values as attributes, and the following elements:

- Illuminant – enumerated choice of CIE illuminant designations (ex D65) or “Custom”.
- CustomIlluminant – optional element containing the custom illuminant SPD
- Observer – enumerated choice of observer database used (2_degree or 10_degree) or optional attributes to further describe the name, angle, age used to produce the custom CIE 170 database.
- Method – enumerated choice (Table 5, Table 6, 1nm Table, custom) of table/method used.

3.2.2 MeasurementSpec

Contains the information used to describe how the original data was measured or synthesized:

- MeasurementType – required enumerated choice of spectral type or Colorimetric values (Spectrum_Reflectance, Spectrum_Transmittance, Colorimetric_Emissive, etc.)
- GeometryChoice–required enumerated choice of measurement geometry (emissive, sphere, single-angle, multi-angle) and additional element types used to describe appropriate parameters. Also includes optional attributes to describe cell pathlength, aperture, bandwidth, and bandpass corrected.
- WavelengthRange – contains attributes for StartWL and Increment used to describe the starting wavelength and increment used for the data list within spectral color values.
- LuminanceUnitsType – optional element used to describe the units used for emissive color value types (ColorEmissiveCIEXYZ and ColorEmissiveCIExyY).
- CalibrationStandard – optional element that holds a string describing the calibration standard that the data is referenced to or the material used (NIST, XRGA, NBS, Ceramic, Spectralon, etc.).
- Aperture – optional element describing the size of measurement aperture used as a string value – either the label such as “LAV”, or the actual aperture size in mm (“6.0mm”).
- BandpassCorrected – optional True/False element designating whether the spectral data (or the colorimetric data derived from it) was bandpass corrected.
- Device – optional element containing parameters that describe the measuring apparatus including manufacturer, model, serial number, filters used, device class, device illumination, and polarization.

3.1.3 PhysicalAttributes

An element that contains optional data common to the physical nature of all of the referencing color objects, taken from the following list:

- TargetType – enumerated list of IT8 target types.
- FinishType – enumerated list of finishes (coated, matte, glossy, etc.)
- SubstrateType – enumerated list of substrate types (wood, paper, textile, etc.)
- Quantity
- Width
- Length
- Height
- Thickness
- Gloss
- Opacity
- CustomAttributeString
- CustomAttributeValue
- Image (binary or URI reference)

3.3 Profiles

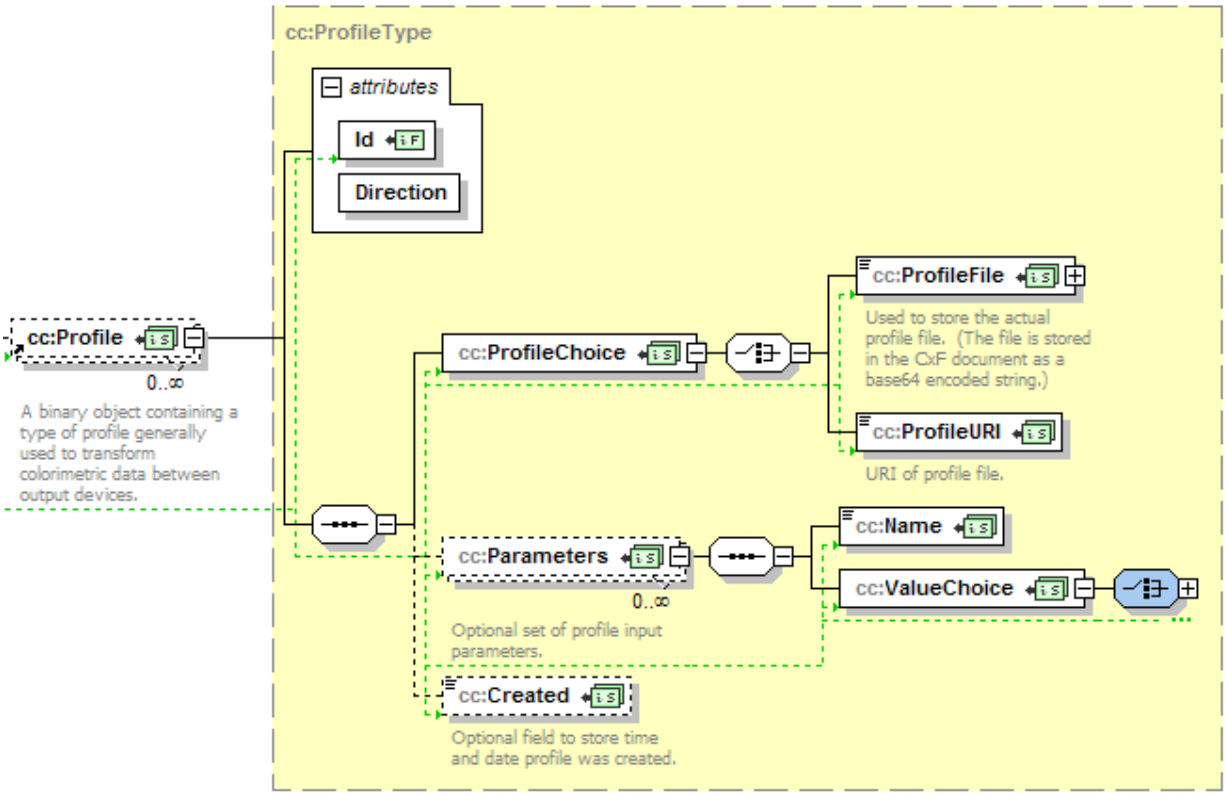
Some device color value types contain an attribute named "ProfileSpecification" that references the Profile ID that applies to it. Profiles contain information that other applications can use to transform color value information between device types or other color spaces. A profile can be binary, a set of parameter data, or it can be a URI reference to an externally supplied profile.

Profile contain attributes:

- Id (required) – the unique string used to reference this Profile object.
- Direction – enumerated choice of Input, Output, Both describing whether this profile is used as an output modifier, an input modifier, or can be used for both.

Profiles contain elements:

- ProfileChoice (required) – contains either a ProfileFile (as a binary object) or a ProfileURI that contains a reference to an external file.
- Parameters (optional) – multiple parameters which each contain a name/value pair. Used to store a profile matrix array of data used for implementing a transform equation.
- Created (optional) – an optional date/time filed to describe when this profile was created.


cc:ProfileType
attributes
Id
Direction
cc:Profile
0..∞
A binary object containing a type of profile generally used to transform colorimetric data between output devices.
cc:ProfileChoice
cc:ProfileFile
Used to store the actual profile file. (The file is stored in the CxF document as a base64 encoded string.)
cc:ProfileURI
URI of profile file.
cc:Parameters
0..∞
Optional set of profile input parameters.
cc:Name
cc:ValueChoice
cc:Created
Optional field to store time and date profile was created.

4 CustomResources Data Model

A CustomResource implements a specific application use or extension of the core Resources. They define the extension characteristics and parameters within the CustomResource but reference the color objects stored in the core resources (using the objects reference Id).

Each CxF3 file can contain 0 to many CustomResources.

4.1 CustomResource usage

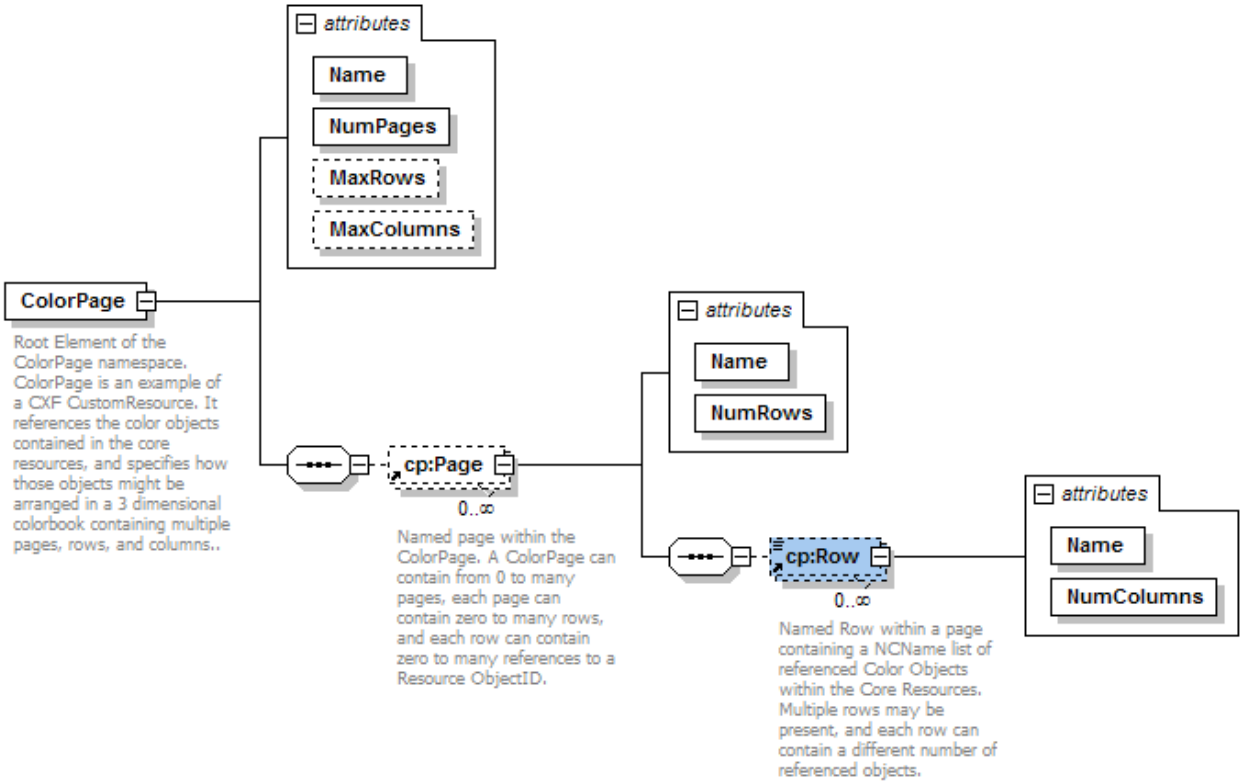
CustomResources are used whenever the core "Resource" information needs to be augmented with procedural, organizational, or standardized information not already contained in the Resources. An example of this is shown in the example CxF3 file "ColorChecker.cxf" (image shown below), where the Resources contain the color information for 24 color objects (collapsed in the image below), and the "ColorPage" CustomResource contains the information necessary to arrange, identify, and display that information in the form of a single page of colors arranged in 6 columns and 4 rows. Note the use of multiple namespaces within the same CxF file. The validation of a CxF3 file is generally performed strictly on the Core Schema – with "lax" validation being enforced on the CustomResource element.

```
<?xml version="1.0" encoding="UTF-8"?>
<CxF xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://colorexchangeformat.com/CxF3-core" xsi:schemaLocation="
http://colorexchangeformat.com/CxF3-core CxF3_Core.xsd">
  <Resources>
    <ObjectCollection>
    <ColorSpecificationCollection>
    <ProfileCollection>
  </Resources>
  <CustomResources>
    <ColorPage xmlns="http://colorexchangeformat.com/CxF3-colorpage" Name="x-rite Color Checker" xsi:SchemaLocation="
http://colorexchangeformat.com/CxF3-colorpage CxF3_colorpage.xsd" NumPages="1" MaxRows="4" MaxColumns="6">
      <Page Name="ColorChecker" NumRows="4">
        <Row Name="row1" NumColumns="6">R1 R2 R3 R4 R5 R6</Row>
        <Row Name="row2" NumColumns="6">R7 R8 R9 R10 R11 R12</Row>
        <Row Name="row3" NumColumns="6">R13 R14 R15 R16 R17 R18</Row>
        <Row Name="row4" NumColumns="6">R19 R20 R21 R22 R23 R24</Row>
      </Page>
    </ColorPage>
  </CustomResources>
</CxF>
```

4.2 CustomResource schema

Each CustomResource type specifies its own data model via a custom namespace and xml schema, embedded within the CxF file. The elements within the CustomResource use referencing to refer to the actual color objects contained within the Resources "ObjectCollection" .

An example schema for a CustomResource might look like the following (schema for example ColorPage shown above), where the element "Row" is defined as a xs:NCNameList and contains a list of object references for each row.



4.3 Possible CustomResources

Some possible CustomResource types include:

- Palette/Collections (adds membership/ multiple levels of hierarchies to color objects)
- Fandecks (adds specific arrangements of colored chips in multiple strips with possible blank spaces)
- Quality Control (adds standard/sample relationship and tolerancing information)
- Color Pages (arrangement of colors in 3 dimensional pages)
- Colorant characterization sets (colorant files)
- Lingual/Lookup (cross reference of color objects by alternate names, languages, notations)
- StandardMethod (specific ISO method for layout, organization, and/or additional required data)

Appendix A: Example CxF3 file (ColorChecker.cxf)

The following example CxF3 file contains 24 color objects, and includes spectral data, CIELab data, HTML data, and Munsell notation for each object. In addition the first color object (dark skin) contains examples of the use of CMYK ,Color Recipe, Tag Collection, and PhysicalAttributes. Also included in this example is the use of a CustomResource (colorpage). The listing shown below collapses the nodes for the remaining 23 objects to save space – the complete example file "ColorChecker.cxf" is available.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <CxF xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://colorexchangeformat.com/CxF3-core"
xsi:schemaLocation="http://colorexchangeformat.com/CxF3-core CxF3_Core.xsd">
- <Resources>
  - <ObjectCollection>
        - <Object ObjectType="Standard" Id="R1" Name="dark skin" GUID="30453E30-3068-524A-2E4D-917F2E4D917F">
            <CreationDate>2003-09-28T12:15:33-05:00</CreationDate>
            - <ColorValues>
                <ReflectanceSpectrum MeasureDate="2003-09-28T12:15:33-05:00" ColorSpecification="CSD65-2"
            StartWL="400">0.0580 0.0594 0.0594 0.0584 0.0581 0.0591 0.0599 0.0601 0.0603 0.0610 0.0634 0.0695 0.0760 0.0786
            0.0798 0.0826 0.0897 0.1024 0.1197 0.1350 0.1434 0.1455 0.1499 0.1594 0.1721 0.1842 0.1913 0.1928 0.1878 0.1734
            0.1704</ReflectanceSpectrum>
            - <ColorSRGB ColorSpecification="CSD65-2">
                <MaxRange>254</MaxRange>
                <R>115</R>
                <G>82</G>
                <B>68</B>
            </ColorSRGB>
            - <ColorCIELab ColorSpecification="CSD50-2">
                <L>37.986</L>
                <A>13.555</A>
                <B>14.059</B>
            </ColorCIELab>
            </ColorValues>
            - <DeviceColorValues>
                <ColorHTML ColorSpecification="CSD65-2" Name="PREVIEW" HTML="735244" />
                <ColorNotation ColorSpecification="CSD65-2" Name="Munsell" Notation="3YR 3.7/3.2" />
                - <ColorCMYK ColorSpecification="CSD50-2" ProfileSpecification="Profile1">
                    <Cyan>42</Cyan>
                    <Magenta>39</Magenta>
                    <Yellow>24</Yellow>
                     <Black>38</Black>
                </ColorCMYK>
                - <ColorRecipe ColorSpecification="CSD50-2" Name="" Comments="" Units="Percent">
                    <CreationDate>2009-02-02T09:10:12-05:00</CreationDate>
                    <Tag Name="Customer" Value="Paints-R-Us" />
                     <Substrate>Paper</Substrate>
                    <Process>Paint.ifs</Process>
                    - <Colorant Name="Clear Neutral Base" ID="1">
                    <PartNumber />
                    <Density xsi:nil="true" />
                    <Value>0.98020835876465</Value>
                    </Colorant>
                    - <Colorant Name="Raw Umber" ID="5">
                      <PartNumber />
                     <Density xsi:nil="true" />
                     <Value>0.0044791665673256</Value>
                     </Colorant>
                    - <Colorant Name="Exterior Red" ID="7">
                      <PartNumber />
                      <Density xsi:nil="true" />
                      <Value>0.0008854166418314</Value>
                      </Colorant>
                    - <Colorant Name="Medium Yellow" ID="12">
                      <PartNumber />
                      <Density xsi:nil="true" />
                      <Value>0.014427083730698</Value>
                      </Colorant>
                  </ColorRecipe>
            </DeviceColorValues>
            - <TagCollection Name="iQC_Tags">
                <Tag Name="Munsell Notation" Value="3YR 3.7/3.2" />
            </TagCollection>
            - <PhysicalAttributes>
                <Quantity>1</Quantity>
                <Height Units="cm">4.0</Height>
                <Width Units="cm">4.0</Width>
             </PhysicalAttributes>
    </Object>
```

```
+ <Object ObjectType="Standard" Id="R2" Name="light skin" GUID="30453E30-3068-524A-3E4D-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R3" Name="blue sky" GUID="30453E30-3068-524A-4E4D-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R4" Name="foliage" GUID="30453E30-3068-524A-5E4D-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R5" Name="blue flower" GUID="30453E30-3068-524A-7E4D-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R6" Name="bluish green" GUID="30453E30-3068-524A-2E4D-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R7" Name="orange" GUID="30453E30-3068-524A-3A4D-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R8" Name="purplish blue" GUID="30453E30-3068-524A-2E9M-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R9" Name="moderate red" GUID="304A3E30-3068-524A-2E4D-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R10" Name="purple" GUID="30453E30-3068-524A-2E4D-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R11" Name="yellow green" GUID="30453E30-3068-524A-2E4D-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R12" Name="orange yellow" GUID="30453E30-3068-524A-2E4D-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R13" Name="blue" GUID="20453E30-3068-524A-234D-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R14" Name="green" GUID="34453C30-3068-524A-5B4D-317F2E4D917D">
+ <Object ObjectType="Standard" Id="R15" Name="red" GUID="30453E30-3068-524A-2E12-917C8E4D9132">
+ <Object ObjectType="Standard" Id="R16" Name="yellow" GUID="30253E30-3068-524A-2E4D-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R17" Name="magenta" GUID="30453E30-3068-524A-2E4D-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R18" Name="cyan" GUID="30453E30-3068-124A-2E4D-917F2D4D917F">
+ <Object ObjectType="Standard" Id="R19" Name="white" GUID="30453E30-3068-581A-2E4D-917F2C4D917F">
+ <Object ObjectType="Standard" Id="R20" Name="neutral 8" GUID="30453E30-3068-524A-2E4D-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R21" Name="neutral 6.5" GUID="30453E30-3068-524A-2E4D-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R22" Name="neutral 5" GUID="30453E30-3068-524A-2E4D-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R23" Name="neutral 3.5" GUID="30453E30-3068-524A-2E4D-917F2E4D917F">
+ <Object ObjectType="Standard" Id="R24" Name="black" GUID="30453E30-3068-524A-2E4D-917F2E4D917F">
 </ObjectCollection>
- <ColorSpecificationCollection>
        - <ColorSpecification Id="CSD50-2">
                - <TristimulusSpec>
                        <Illuminant>D50</Illuminant>
                         <Observer>2_Degree</Observer>
                        <Method>E308_Table5</Method>
                </TristimulusSpec>
                - <MeasurementSpec>
                        <MeasurementType>Colorimetric_Reflectance</MeasurementType>
                        - <GeometryChoice>
                                - <SingleAngle>
                                        <SingleAngleConfiguration>Annular</SingleAngleConfiguration>
                                        <IlluminationAngle>45.0</IlluminationAngle>
                                        <MeasurementAngle>0.0</MeasurementAngle>
                                </SingleAngle>
                        </GeometryChoice>
                </MeasurementSpec>
 </ColorSpecification>
        - <ColorSpecification Id="CSD65-2">
                - <TristimulusSpec>
                         <Illuminant>D65</Illuminant>
                         <Observer>10_Degree</Observer>
                         <Method>E308_Table6</Method>
                  </TristimulusSpec>
                - <MeasurementSpec>
                        <MeasurementType>Spectrum_Reflectance</MeasurementType>
                        - <GeometryChoice>
                                - <SingleAngle>
                                         <SingleAngleConfiguration>Annular</SingleAngleConfiguration>
                                        <IlluminationAngle>45.0</IlluminationAngle>
                                        <MeasurementAngle>0.0</MeasurementAngle>
                                 </SingleAngle>
                        </GeometryChoice>
                 <WavelengthRange StartWL="400" Increment="10" />
                 <CalibrationStandard>Ceramic</CalibrationStandard>
                - <Device>
                        <Manufacturer>XRite</Manufacturer>
                        <Model>XR962</Model>
                         <SerialNumber />
                         <DeviceClass>DeviceClass_Spot</DeviceClass>
                        <DeviceFilter>Filter_None</DeviceFilter>
                        <DeviceIllumination>M1_Daylight</DeviceIllumination>
                        <DevicePolarization>false</DevicePolarization>
                </Device>
                </MeasurementSpec>
        </ColorSpecification>
 </ColorSpecificationCollection>
- <ProfileCollection>
        - <Profile Id="Profile1" Direction="Profile_Output">
                - <ProfileChoice>
                         <ProfileFile ProfileName="Dummy profiledata" />
                 </ProfileChoice>
          </Profile>
 </ProfileCollection>
 </Resources>
- <CustomResources>
```

```
- <ColorPage xmlns="http://colorexchangeformat.com/CxF3-colorpage" Name="x-rite Color Checker"
xsi:SchemaLocation="http://colorexchangeformat.com/CxF3-colorpage CxF3_colorpage.xsd" NumPages="1" MaxRows="4"
MaxColumns="6">
        - <Page Name="ColorChecker" NumRows="4">
            <Row Name="row1" NumColumns="6">R1 R2 R3 R4 R5 R6</Row>
            <Row Name="row2" NumColumns="6">R7 R8 R9 R10 R11 R12</Row>
            <Row Name="row3" NumColumns="6">R13 R14 R15 R16 R17 R18</Row>
            <Row Name="row4" NumColumns="6">R19 R20 R21 R22 R23 R24</Row>
        </Page>
    </ColorPage>
</CustomResources>
</CxF>
```

Appendix B: ColorChecker.xsl Stylesheet Example 1

The ColorCheckerTable.xsl stylesheet uses xpath parsing to read the above example colorchecker.cxf file and render it in an html format that can display as a “color checker” table in your browser:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:cc="http://colorexchangeformat.com/CxF3-core">
<xsl:output method="html" version="1.0" encoding="UTF-8" indent="yes" />
- <xsl:template match="/">
- <html>
- <body>
<h2>x-rite color checker</h2>
<xsl:apply-templates select="cc:CxF/cc:Resources" />
</body>
</html>
</xsl:template>
- <xsl:template match="cc:Resources/cc:ObjectCollection">
- <table border="1">
- <tr bgcolor="#b4b4b4">
<th align="center" />
<th align="center" />
<th align="center" />
<th colspan="3" align="center">sRGB</th>
<th colspan="3" align="center">CIE L*a*b*</th>
<th align="center" cellpadding="3">Munsell</th>
</tr>
- <tr bgcolor="#b4b4b4">
<th align="center">No.</th>
<th align="center" cellpadding="3">Name</th>
<th align="center" cellpadding="3">patch</th>
<th align="center" cellpadding="3">R</th>
<th align="center" cellpadding="3">G</th>
<th align="center" cellpadding="3">B</th>
<th align="center" cellpadding="3">L*</th>
<th align="center" cellpadding="3">a*</th>
<th align="center" cellpadding="3">b*</th>
<th align="center" cellpadding="3">HVC Notation</th>
</tr>
- <xsl:for-each select="cc:Object">
- <tr>
- <xsl:variable name="xnumber">
<xsl:value-of select="@Id" />
</xsl:variable>
- <td> <xsl:value-of select="substring-after($xnumber,'R')" /> </td>
- <td> <xsl:value-of select="@Name" /> </td>
- <xsl:for-each select="cc:DeviceColorValues">
- <xsl:for-each select="cc:ColorHTML/@HTML">
<td bgcolor="{.}" />
</xsl:for-each>
</xsl:for-each>
- <xsl:for-each select="cc:ColorValues">
- <xsl:for-each select="cc:ColorSRGB">
- <td> <xsl:value-of select="./cc:R" /> </td>
- <td> <xsl:value-of select="./cc:G" /> </td>
- <td> <xsl:value-of select="./cc:B" /> </td>
</xsl:for-each>
- <xsl:for-each select="cc:ColorCIELab">
- <td> <xsl:value-of select="./cc:L" /> </td>
- <td> <xsl:value-of select="./cc:B" /> </td>
- <td> <xsl:value-of select="./cc:B" /> </td>
</xsl:for-each>
</xsl:for-each>
- <xsl:for-each select="cc:DeviceColorValues">
- <xsl:for-each select="cc:ColorNotation">
- <td> <xsl:value-of select="@Notation" /> </td>
</xsl:for-each>
</xsl:for-each>
</tr>
</xsl:for-each>
</table>
- <xsl:for-each select="cc:Object[1]/cc:DeviceColorValues/cc:ColorHTML">
- <xsl:for-each select="id(@ColorSpecification)">
<b>Color Patches rendered for illuminant</b>
- <b> <xsl:value-of select="./cc:TristimulusSpec/cc:Illuminant" /> </b>
</xsl:for-each>
</xsl:for-each>
</xsl:template>
<xsl:template match="cc:Resources/cc:ColorSpecificationCollection" />
```

```
<xsl:template match="cc:Resources/cc:ProfileCollection" />
</xsl:stylesheet>
```

This stylesheet will produce the following output:

x-rite color checker

			sRGB			CIE L*a*b*			Munsell
No.	Name	patch	R	G	B	L*	a*	b*	HVC Notation
1	dark skin		115	82	68	37.986	14.059	14.059	3YR 3.7/3.2
2	light skin		194	150	130	65.711	17.81	17.81	2.2YR 6.47/4.1
3	blue sky		98	122	157	49.927	21.925	21.925	4.3PB 4.95/5.5
4	foliage		87	108	67	43.139	21.905	21.905	3YR 3.7/3.2
5	blue flower		133	128	177	55.112	-25.399	-25.399	9.7PB 5.47/6.7
6	bluish green		103	189	170	70.719	-0.199	-0.199	2.5BG 7/6
7	orange		214	126	44	62.661	57.096	57.096	3YR 3.7/3.2
8	purplish blue		80	91	166	40.02	-45.964	-45.964	7.5PB 4/10.7
9	moderate red		193	90	99	51.124	16.248	16.248	2.5R 5/10
10	purple		94	60	108	30.325	-21.587	-21.587	5P 3/7
11	yellow green		157	188	64	72.532	57.255	57.255	3YR 3.7/3.2
12	orange yellow		224	163	46	71.941	67.857	67.857	10YR 7/10.5
13	blue		56	61	150	28.778	-50.297	-50.297	7.5PB 2.9/12.7
14	green		70	148	73	55.261	31.37	31.37	0.25G 5.4/8.65
15	red		175	54	60	42.101	28.19	28.19	5R 4/12
16	yellow		231	199	31	81.733	79.819	79.819	5Y 8/11.1
17	magenta		187	86	149	51.935	-14.574	-14.574	2.5RP 5/12
18	cyan		8	133	161	51.038	-28.638	-28.638	5B 5/8
19	white		243	243	242	96.539	1.186	1.186	N 9.5/
20	neutral 8		200	200	200	81.257	-0.339	-0.339	N 8/
21	neutral 6.5		160	160	160	66.766	-0.506	-0.506	N 6.5/
22	neutral 5		122	122	121	50.867	-0.27	-0.27	N 5/
23	neutral 3.5		85	85	85	35.656	-1.231	-1.231	N 3.5/
24	black		52	52	52	20.461	-0.973	-0.973	N 2/

Color Patches rendered for illuminant D65

Appendix C: CustomResource Stylesheet Example 2

The following stylesheet uses xpath parsing to read the above example colorchecker.cxf file and render it in a html format that can display as a “color checker” page in your browser (see example image below).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="2.0"
            xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
            xmlns:cc="http://colorexchangeformat.com/CxF3-core" xmlns:cp="http://colorexchangeformat.com/CxF3-colorpage">
<xsl:output method='html' version='2.0' encoding='UTF-8' indent='yes'/>
<xsl:template match="/">
  <html>
  <body>
  <h2>x-rite color checker</h2>
            <xsl:apply-templates select="cc:CxF/cc:CustomResources"/>
  </body>
  </html>
</xsl:template>
  <xsl:template match="cc:CustomResources/cp:ColorPage">
   <table border="1">
            <xsl:for-each select="cp:Page">
                        <xsl:for-each select="cp:Row">
                                    <tr>
                                       <xsl:for-each select="//cc:Object[@Id=tokenize(current(), '\s+')]">
                                                <xsl:variable name="color" select=".//cc:ColorHTML/@HTML"/>
                                                            <td style="text-align:center; background-color:{$color}; height:80px; width:80px">
                                                                        <xsl:value-of select="./@Name"/>
                                                            </td>
                                       </xsl:for-each>
                                    </tr>
                        </xsl:for-each>
      </xsl:for-each>
   </table>
 </xsl:template>
</xsl:stylesheet>
```

This stylesheet will produce the following output:

x-rite color checker

dark skin	light skin	blue sky	foliage	blue flower	bluish green
orange	purplish blue	moderate red	purple	yellow green	orange yellow
blue	green	red	yellow	magenta	cyan
white	neutral 8	neutral 6.5	neutral 5	neutral 3.5	black