2012 ICC DevCon

The Premiere Learning and Networking Event for Users and Product Developers Working with ICC Based Color Management

RefIccLabs – A Peek Behind the Curtain

Max Derhak

Onyx Graphics Inc.

Agenda

- Challenges for color management and ICC.1
- IccLabs and ReflccLabs
- Overview of Proposals
- Demo
- Conclusions

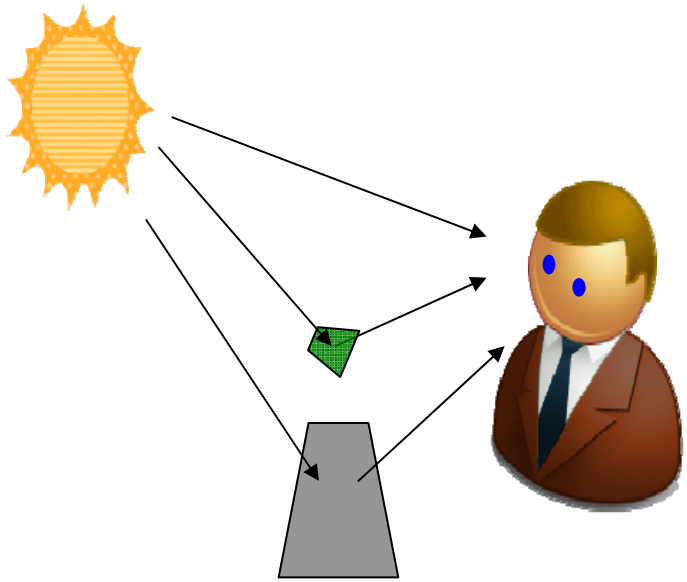
ICC Color Management

- The purpose of the ICC is to promote the use and adoption of open, vendor-neutral, cross-platform color management systems.

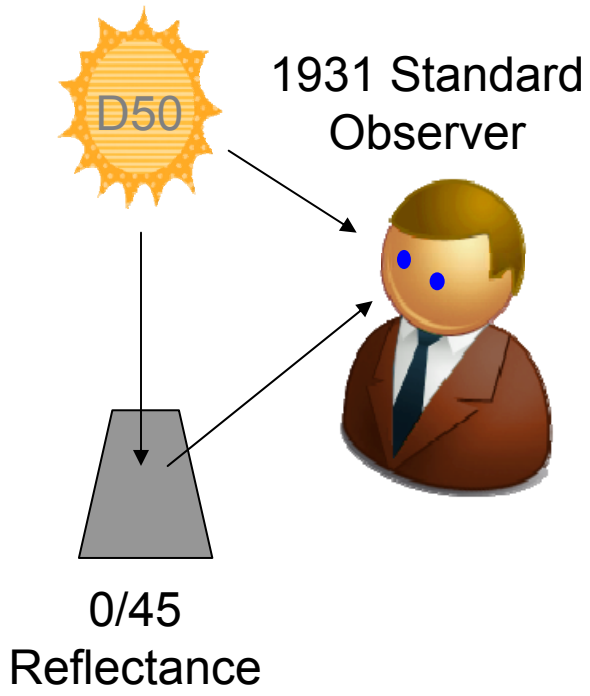
- With “Color Management” being defined as the “communication of the associated data required for unambiguous interpretation of color content data, and application of color data conversions, as required, to produce the intended reproductions”.

Challenges for Color Management

- Different Light Sources
- Characteristics of Surfaces
- Variations in Observer
- Modeling Everything
- Variations in Reproduction Intent



ICC.1 Color Management Simplifications



Note: Other Illuminants can be indirectly represented. However, color data in profile MUST always be converted to these viewing conditions for processing by the CMM.

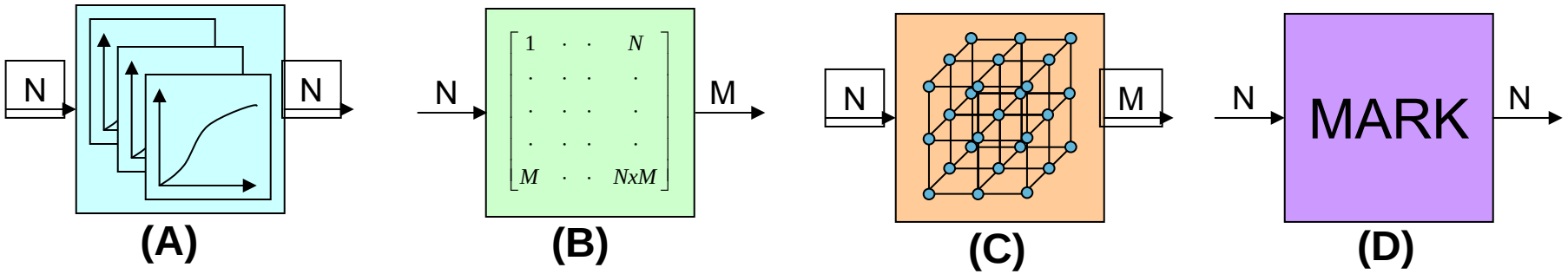
- **ICC.1 color management simplifications:**
  - Fixed PCS Viewing Conditions
    - D50 Illuminant
    - 500 lx
  - Simple Reflectance Model
    - Flat surface
    - 0/45 geometry
    - No gloss
    - No Fluorescence
  - Standard 1931 Observer
  - Explicit Transforms…

ICC.1 General Profile Contents

- **Display Profiles**
  - Header
  - Metadata Tags
  - Transform Tags
    - Matrix / TRC : XYZ Chromaticities / Sampled Curves, Parametric Curves)
- **Device / Color Space Profiles**
  - Header
  - Metadata Tags
  - Transform Tags
    - AtoB0 / BtoA0 (*Optionally additional AtoBx / BtoAx) :* lut8, lut16, lutAtoB, lutBtoA
    - *Optional - Matrix / TRC* : *XYZ Chromaticities / Sampled Curves, Parametric Curves)*
    - *v4.3 Optional - DtoBx / BtoDx : multiProcessElementType*
    - Gamut Tag
- **Device Link Profiles**
  - Header
  - Metadata Tags
  - Transform Tags
    - AtoB0 : lut8, lut16, lutAtoB
    - *V4.3 Optional – DtoB0 : multiProcessElementType*
- ***Named Color Profiles***
  - Header
  - Metadata Tags
  - Transform Tag
    - Named Color Table : namedColorTagType

Review - multiProcessElementsType

- **Completely defines transformation from tag input to tag output**
- **32-bit Floating Point Encoding**
- **Arbitrary order of processing elements**
- **Repertoire of processing elements:**
  A. Sets of 1-dimensional functions that are made up of parametric and sampled curve segments
  B. N x M linear matrix transform
  C. N x M multi-dimensional lookup table
  D. Marker elements that perform no operation on values



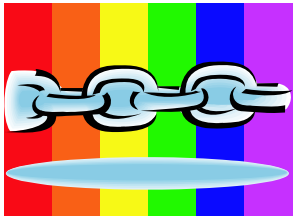
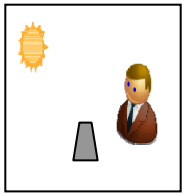
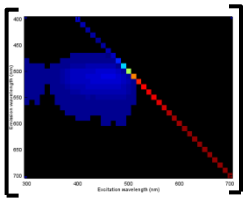
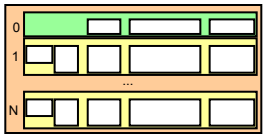
The IccLabs Project and RefIccLabs

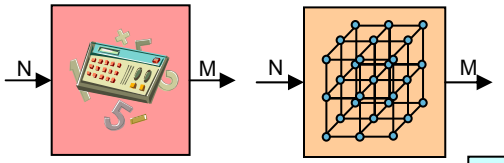
Going Forward with IccLabs

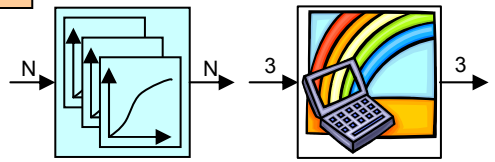
- **A main goal of IccLabs was to better address color management challenges**
  - Overcomming limitations of current transforms with D50 colorimetry
  - Resulting in (a) new file specification(s) and profiles
  - New CMM to be backwards compatible with V2 and V4 profiles
  - New profiles not expected to be compatible with older CMMs
- **ICC will provide a reference implementation of an IccLabs based parser and CMM - RefIccLabs**

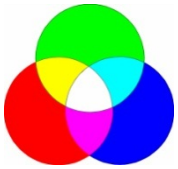
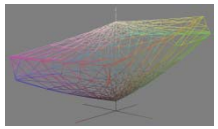
IccLabs – A Collection of Proposals

- **PCS Extensions**
  - Spectral profile header proposal
  - Spectral viewing conditions tag proposal
  - Sparse matrix encoding proposal
- **Tag Array and Tag Structure tag types**
  - Named Color Tag Array proposal
  - Support for BRDF proposal
- **multiProcessingElement Extensions**
  - Calculator element proposal
  - CLUT element extensions proposal
  - Segmented curve extensions proposal
  - ICC Color Appearance Model element proposal
- **Other Extensions**
  - Three component color encoding profiles proposal
  - Gamut Boundary Description encoding proposal
  - CxF tag encoding proposal



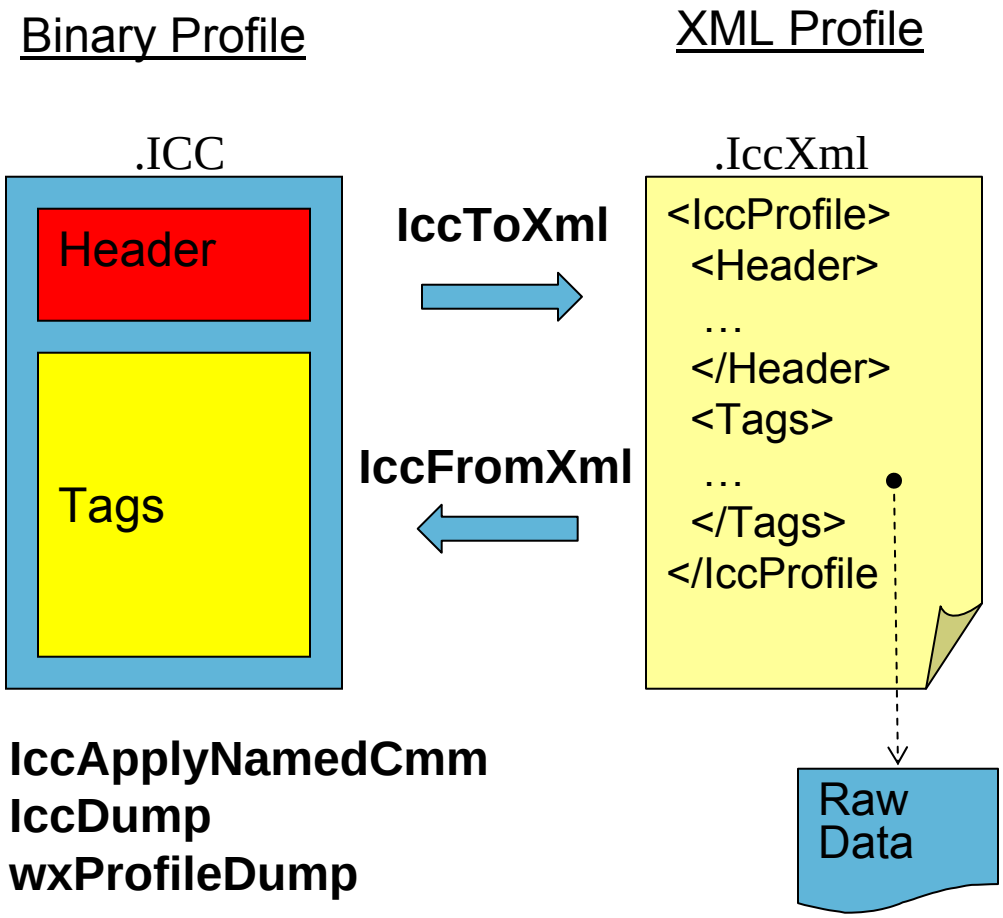


CxF

RefIccLabs

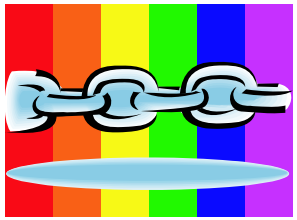
- **Provides a C++ reference implementation of profile library and CMM changes defined by IccLabs proposals**
  - RefIccLabs was intially based on SampleICC and IccXml projects
- **Simultaneously defines two types of color profile**
  - Binary
    - Extensions of ICC profile specification
  - Xml
    - XML representation of ICC profile contents
    - File may refer to outside (raw) data files
- **Package contains Libraries and tools**
  - IccProfLib (.ICC)
    - IccApplyNamedCMM
    - IccDump
    - wxProfileDump
  - IccLibXml (.IccXml)
    - IccFromXml
    - IccToXml

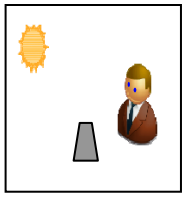


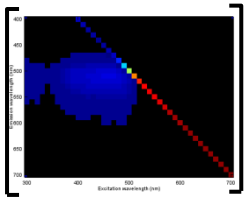
- **IccApplyNamedCmm**
- **IccDump**
- **wxProfileDump**

PCS Extensions

- *Tag Array and Tag Structure tag types*
- *multiProcessingElement Extensions*
- *Other Extensions*







Header changes for PCS extensions

Byte position	Field length (bytes)	Field contents	Encoded as
0 to 3	4	Profile size	UInt32Number
4 to 7	4	Preferred CMM type	See 6.2.3
8 to 11	4	Profile version number	See 6.2.4
12 to 15	4	Profile/Device class	See 6.2.5
16 to 19	4	Colour space of data (possibly a derived space)	See 6.2.6
20 to 23	4	PCS	See 6.2.7
24 to 35	12	Date and time this profile was first created	dateTimeNumber
36 to 39	4	'acsp' (61637370h) profile file signature	See 6.2.9
40 to 43	4	Primary platform signature	See 6.2.10
44 to 47	4	Profile flags to indicate various options for the CMM such as distributed processing and caching options	See 6.2.11
48 to 51	4	Device manufacturer of the device for which this profile is created	See 6.2.12
52 to 55	4	Device model of the device for which this profile is created	See 6.2.13
56 to 63	8	Device attributes unique to the particular device setup such as media type	See 6.2.14
64 to 67	4	Rendering Intent	See 6.2.15
68 to 79	12	The nCIEXYZ values of the illuminant of the PCS.	XYZNumber
80 to 83	4	Profile creator signature	See 6.2.17
84 to 99	16	Profile ID	See 6.2.18
100 to 103	4	Spectral colour space signature	See table X
104 to 109	6	Spectral PCS wavelength range	spectralRange
110 to 115	6	Bi-directional PCS wavelength range	spectralRange
116 to 127	12	Bytes reserved for future expansion and shall be set to zero (00h)	

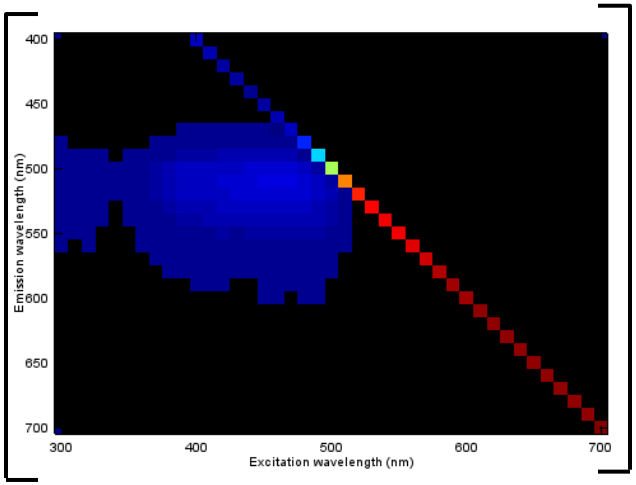
- **PCS field defines PCS color space for A2Bx/B2Ax tags**
  - If zero then no colorimetric PCS is used and no A2BX/B2AX tags should exist
  - A2Bx/B2Ax tags can also be of type multiProcessElementType
- **Colorimetric PCS can be other than D50 with 2 degree observer**
  - XYZ values of illuminant must match illuminant and observer used
- **Spectral PCS fields added**
  - Spectral PCS field defines color space for D2Bx/B2Dx tags
    - If zero and D2Bx/B2Dx tags exist then PCS field is used to define PCS
  - Spectral PCS range defines wavelength range
  - Bi-spectral PCS range defines excitation range when bi-spectral color space is used

New Color Space Signatures

- **Color spaces extended to support new PCS and Device encoding with up to 65535 channels**
- **New 32 bit color space signature encoding**
  - First two text digits (most significant 16 bits) represent color space type
    - Digits MUST NOT conflict with existing color space signatures
  - Least significant 16 bits contain binary representation of channel count
- **Textual representation of signature represented using six characters**
  - First two characters represent color space type
  - Last four characters provide hexadecimal channel count
  - Example “rs0024” represents 36 channel spectral reflectance
- **New color space signatures**
  - “ncXXXX” N-color device channels
  - “rsXXXX” spectral reflectance color
  - “tsXXXX” transmissive spectral color
  - “esXXXX” radiant (emissive) spectral color
  - “brXXXX” sparse matrix bi-spectral reflectance color (fluorescence)

Fluorescence and Sparse Matrices

- **Fluorescence can be modeled using a "Donaldson Matrix"**
  - "Diagonal" terms specify reflectance factor
  - Off-diagonal terms specify emission due to fluorescence

$$\mathbf{v}_{emission} = \begin{bmatrix} \text{  } \end{bmatrix} \times \mathbf{v}_{illuminant}$$

- **Most of matrix is zero filled *(black)***
  - Storing all entries is wasteful
  - Processing all entries is wasteful

- **Sparse Matrix Proposal**
  - Defines Encoding Types, Tags, and Processing Elements
    - Matrix entries can be encoded as 8-bit, 16-bit, 16-bit float, and 32-bit float numbers
  - Uses compressed row order encoding
    - Only non-zero data is stored and processed
    - Padding used to ensure fixed and consistent data size
    - Allows multiple rows to be simultaneously processed
  - Defines methods for matrix application and interpolation between sparse matrices in lookup tables
  - PCS values can be sparse matrices

New Data Types

- **icFloat16Number**
  - IEEE 16 bit floating point
- **icFloat64Number**
  - IEEE 64 bit floating point encoding
- **spectralRange**
  - Starting Wavelength (icFloat16Number)
  - Ending Wavelength (icFloat16Number)
  - Total number of even spaced steps including start and end

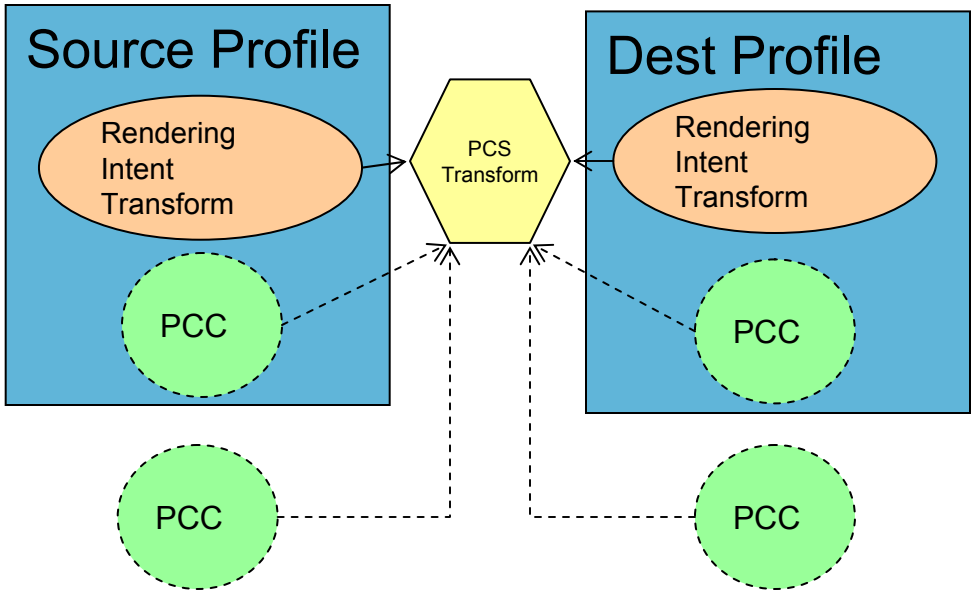
Tag Precedence Changes

1. **Tag associated with rendering intent**
2. **Perceptual intent tag**
3. ***Relative intent tag (with absolute PCS processing as needed)***
   - *Relative only profiles are valid*
4. ***Absolute intent tag using direct encoding of absolute data (with relative PCS processing as needed)***
   - *Absolute only profiles are valid*

Changes in Required Tags

- **Eliminate requirement that perceptual tags are present**
- **Eliminate requirement for gamutTag to be present**
  - gamutTag can still be optionally included
  - Note: New gamut boundary tags can also be included to define colorimetric gamuts
- **spectralMediaWhiteTag required for relative/absolute processing of spectral data**

Profile Connection Conditions



Allows PCS data in profiles to use actual viewing conditions

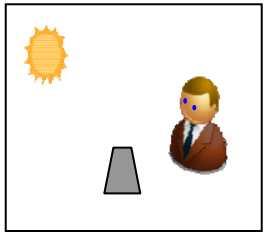
No need for chromaticAdaptationTag!

- **Spectral and custom colorimetric PCS processing is performed using Profile Connection Conditions (PCC) information**
- **PCC information can come from either the profile or externally provided to the CMM**
- **Profile Connection Conditions are NOT required for legacy PCS processing**

Profile Connection Conditions (PCC)

- **PCC information is required whenever PCS is NOT standard D50 using 2-degree observer**
  - Provides for interoperability to connect different viewing conditions
  - CMM becomes responsible for converting PCS data – not profile maker
  - Provides flexibility for when and how conversions are made

- **Profile Connection Conditions comprise of:**
  - Header color space and spectral PCS metadata
  - Three new tags:



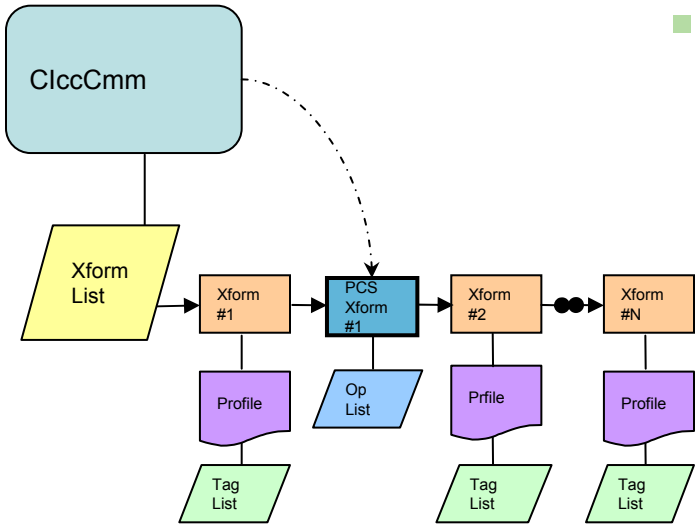
  - **spectralViewingConditionsTag** defines a *spectralViewingConditionsType* (a new tag type) based tag that spectrally specifies observer and illuminant
    - Illuminant and observer signatures used to determine PCS pass through connection
  - **customToStandardPcsTag** defines an MPE based tag to convert from custom PCS colorimetry to standard D50 / 2-degree observer colorimetry
  - **standardToCustomPcsTag** defines an MPE based tag to convert from standard D50 / 2-dgree observer colorimetry to custom PCS colorimetry

RefIccLabs PCS Connection Support

	From Lab	*From XYZ*	*From Reflectance*	*From Transmittance/ Transmissive*	*From Radiant/ Emission*	*From Fluorescence*
To Lab	Yes	Yes	Using PCC	Using PCC	Using PCC	Using PCC
To XYZ	Yes	Yes	Using PCC	Using PCC	Using PCC	Using PCC
To Reflectance	No	No	Yes	Yes	Extract PCC illuminant	Apply then extract PCC illuminant
To Transmittance/ Transmissive	No	No	Yes	Yes	Use PCC illuminant	Apply then extract PCC illuminant
To Radiant / Emission	No	No	Apply PCC Illuminant	Apply PCC illuminant	Yes	Apply PCC illuminant
To Fluorescence	No	No	No	No	No	Exact match required

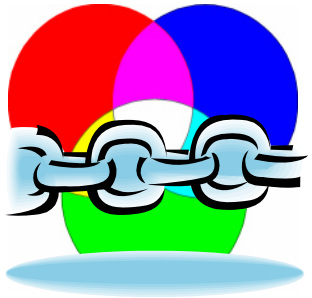
Extended PCS Processing in RefIccLabs

- **Profile Connection Conditions are defined using the interface class IIccProfileConnectionConditions**
  - Inherited by CIccProfile class



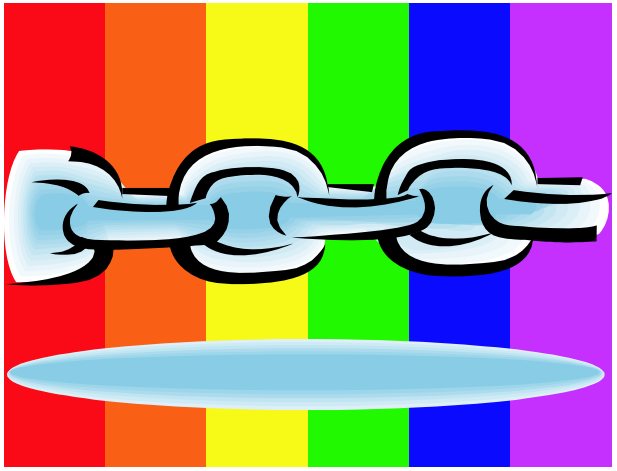
- **CIccCmm implementation**
  - Maintains list of CIccXform based objects associated with profiles
  - Inserts CIccXformPCS objects into Xform list to perform PCS processing (non Device connections)
    - Provided information about source illuminant/observer, and destination illuminatnt/observer from source and destination PCC
    - Establishes and maintains list of operations needed to get from source PCS to destination PCS
    - Note: Device connections pass data directly between Xform objects without need of an additional Xform

General Colorimetric PCS Processing



- **Convert to XYZ colorimetry**
  - Convert Spectral to XYZ colorimetry (using PCC)
  - Convert from Lab to XYZ (using PCC)
- **XYZ scaling for relative/absolute/BPC**
- **Conversion of PCS colorimetry (as needed using PCC)**
  - Use source PCC customToStandardPcsTag
  - Use destination PCC standardToCustomPcsTag
- **Conversion to Lab (as needed)**

Spectral PCS processing

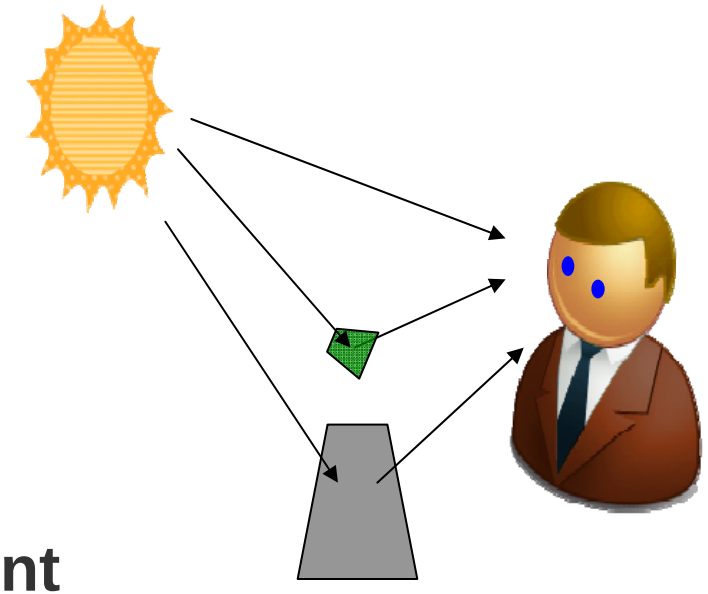


- **Same PCS to Same PCS**
  - Perform spectral range resampling (as needed)
- **Reflectance PCS to Emissive PCS**
  - Range resample reflectance (as needed)
  - Multiply reflectance by PCC illuminant
- **Emissive PCS to Reflectance PCS**
  - Range resample emission (as needed)
  - Divide emissive spectra by PCC illuminant
- **Fluorescence PCS to Emissive PCS**
  - Range resample light source (as needed)
  - Multiply Donaldson matrix by PCC illuminant (cannot easily resample matrix)
- **Fluorescence PCS to Reflectance PCS**
  - Range resample light source (as needed)
  - Multiply Donaldson matrix by source PCC illuminant (cannot easily resample matrix)
  - Range resample resulting emission (as needed)
  - Divide results by destination PCC illuminant

- **Note: If possible it is best to avoid range resampling**

Color Management Challenges Addressed by PCS Extensions

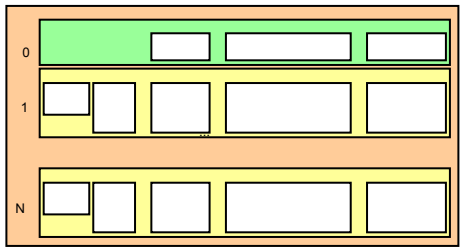
- ✓ Different Light Sources
- ✓ Characteristics of Surfaces
- ✓ Variations in Observer
- ✓ Variations in Reproduction Intent



- *PCS extensions*

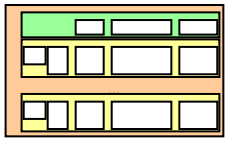
Tag Array and Tag Structure tag types

- *multiProcessingElement extensions*
- *Other Proposals*



Named Profile Changes

- **Use of Spectral PCS by named color profiles requires changes to named color storage**
- **What about tints of named colors?**

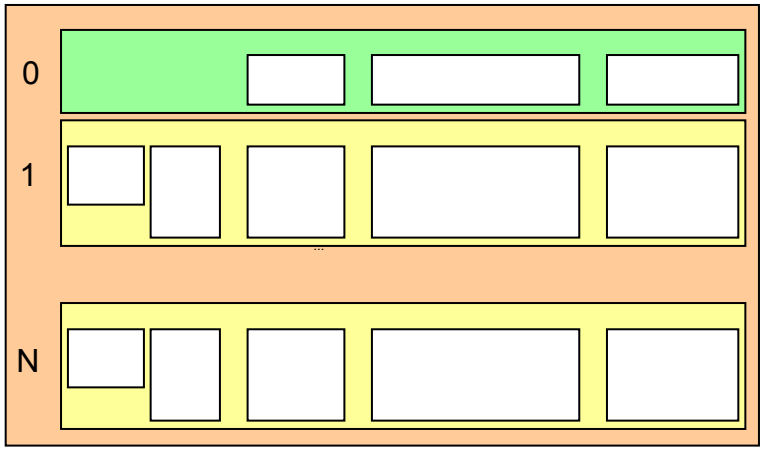


- **New general tag types added to implement named color encoding in profiles:**
  - ***tagArrayType*** : Indexed array of sub-tags
  - ***tagStructType*** : Structure of sub-tags identified by signature keys

- **Both tagArrayType and tagStructType tag types include a type signature to define the type of array or structure**
  - Specialized handlers can be implemented based upon array type or structure type
  - Enables other complex hierarchical data types to be defined

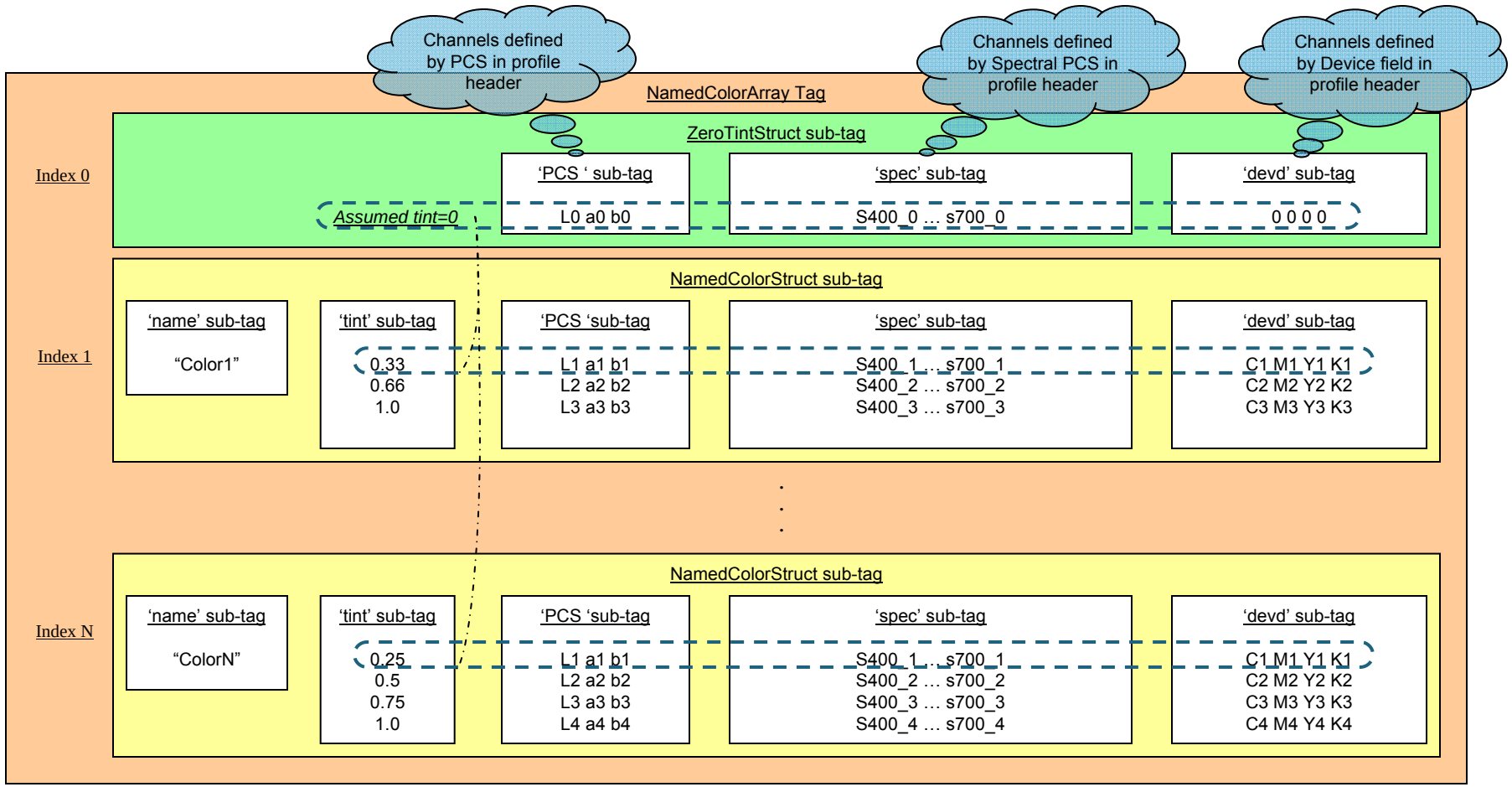
Named colors as a namedColorArray

- **A namedColorArray is an array of tag structures (orange)**
  - The first tag element (index 0) defines zero tint data as a ***zeroTintStruct*** (structure type 'tnt0') structure (green)
  - The following tag elements (index 1 and above) define data for individual named colors as ***namedColorStruct*** (structure type ‘nmcl’) structures (yellow)



- **Tints are defined using successive data elements in zeroTintStruct and namedColorStruct sub-tag number arrays…**

Named Color Tag Overview



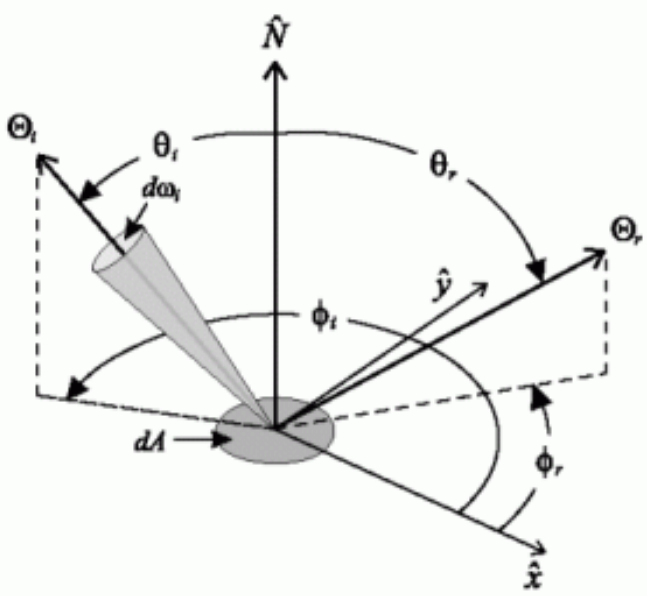
- Inclusion of 'PCS ', 'spec', and 'devd' sub-tags is *optional**
- Additional data can be associated with individual named colors as private sub-tags

Some Useful Tag Types

- **For both inside and outside of tagArrayType and tagStructType based tags**
- **utf8TextType and utf16TextType define single string storage for utf8 and utf16 strings**
  - Useful for storing color names
  - Can be used to store XML data
- **zipUtf8TextType defines single string storage for utf8 text using deflate (zlib) compression**
  - Provides a more efficient way to store XML data
- **float16ArrayType, float32ArrayType, float64ArrayType for storing arrays of floating point data**
  - Note: Direct interpolation between values in all number array tags has been implemented in RefIccLabs
- **segmentedCurveType**
  - Identical encoding as segmented curve in MPE ***curveSetElementType***

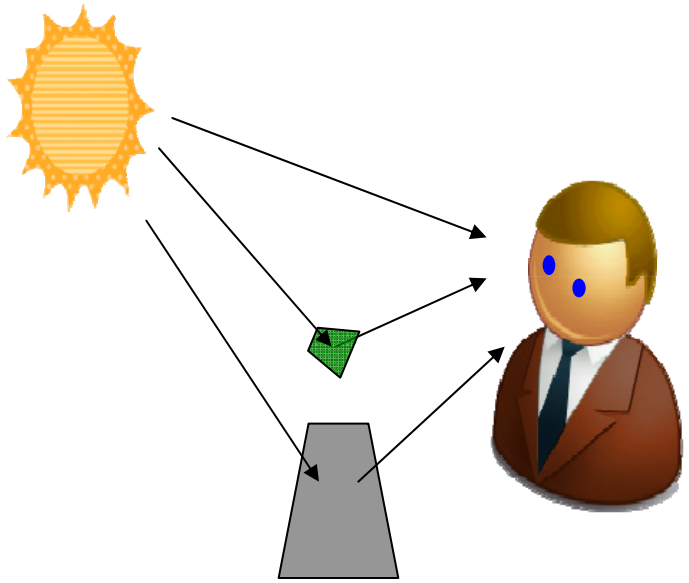
BRDF Tags

- Many surfaces change appearance as viewing angle or illumination angle changes
- A Bidirectional Reflectance Distribution Function (BRDF) can be used to characterize these properties
- Proposed BRDF tags in a profile help communicate this information about color
  - These tags make use of the tagStructType tag type
- BRDF Tags can possibly also be included in a namedColorStruct to define geometrical aspects of a named color
- More information will be presented by James Vogh



Color Management Challenges Addressed by Tag Array and Tag Structure tag types

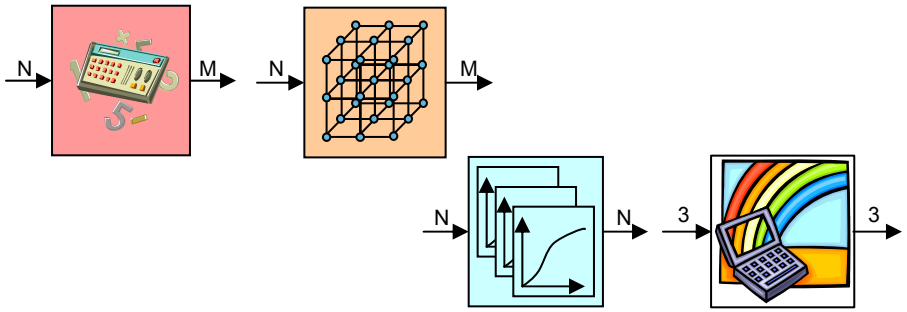
- ✓ Different Light Sources
- ✓ Characteristics of Surfaces
- ✓ Modeling Everything
- ✓ Variations in Reproduction Intent



- *PCS extensions*
- *Tag Array and Tag Structure tag types*

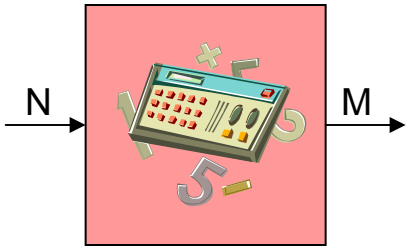
multiProcessingElement Extensions

- *Other extensions*



MPE extension - Calculator Element (Part 1)

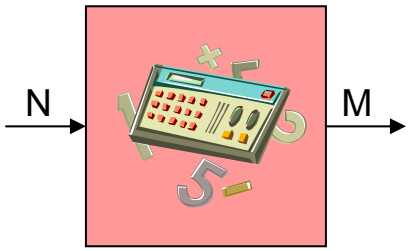
- **Provides mechanism for encoding more complex (non-linear) models**
  - Avoids limitations of CLUT input channel dimensionality
  - Direct encoding of Polynomial Regression
  - Allows for addressing LUTs using polar notation
  - Results in smaller potentially more accurate profiles



- **Defines a script based expression calculator to determine output channels based upon input channels**
  - Uses a sequence of operations that apply to an RPN argument stack
  - Finite memory storage for temporary results
  - Nearly all operations are vector based (operating on multiple channels at same time)

MPE extension - Calculator Element (Part 2)

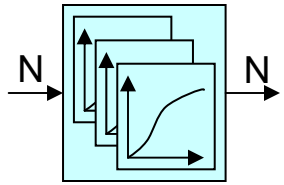
- **Operations**
  - Stack operations
  - Input / Output channel access
  - Temporary memory access
  - Arithmetic and Math functions
  - Comparisons (<, ≤, = ≥, >)
  - Conditional operator sequence evaluation
  - Polar / Cartesian conversion
  - Lab / XYZ conversion
  - Calls to embedded sub-multiProcessElements



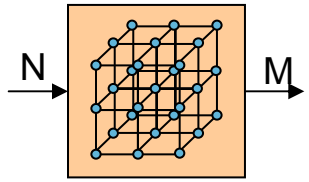
- **Secure deterministic behavior**
  - Script validation MUST be performed before processing begins
  - All possible code paths are known for any arbitrary input (NO LOOPS!)
  - Extent of all possible memory usage is known for any arbitrary input
    - Stack underflow/overflow
    - Input/Output/Temporary channel access

Other multiProcessElement changes

- ***To reduce file size and complexity…***



- **singleCurveSegment type for segmented curves**
  - V4.3 CurveSegments currently require at least 3 curve segments to define a sampled curve
  - A **singleCurveSegment** defines a sampled curve segment as a single curve segment that has simple endpoint extension
  - The sampled curve can be defined using UInt8, UInt16, float16Number, and float32Number arrays



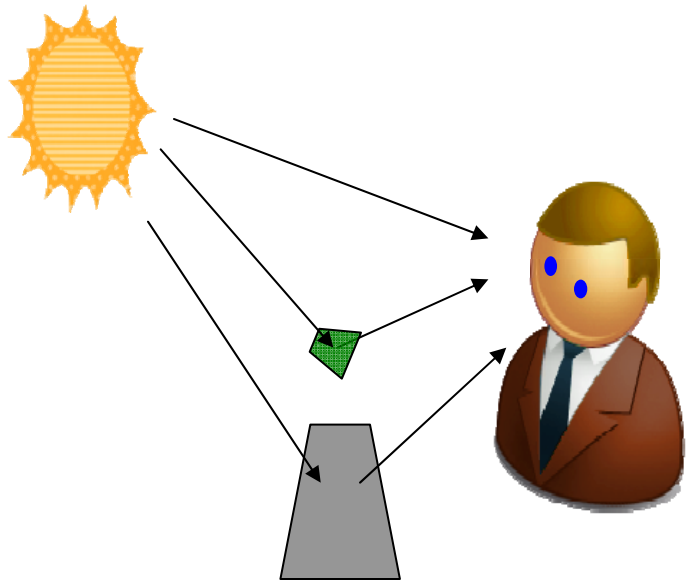
- **New ExtendedCLUT element type**
  - Current CLUT element defines all values as float32Number
  - ExtendedCLUT allows uInt8Number, uInt16Number, float16Number, and float32Number types to be used in the LUT

Color Appearance Model in an MPE tag

- **ICC defines a Color Appearance Model (IccCAM) with a modification of CIECAM02 to avoid problems found with CIECAM02**
- **Color appearance parameters to apply are stored as part of each IccCAM based processing element**
  - XYZtoJab element
    - Utilizes IccCAM methodology to convert from colorimetry to cartesian based color appearance correlates
  - JabtoXYZ element
    - Utilizes IccCAM methodology to convert from cartesian based color appearance correlates to colorimetry
- **Uses:**
  - XYZtoJab and JabtoXYZ processing elements can be utilized to define standardToCustomPCS and customToStandardPCS tags
  - CLUT's can be addressed using Jab

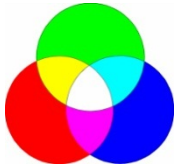
Color Management Challenges Addressed by multiProcessElementType Extensions

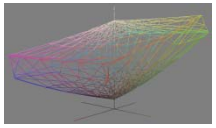
- ✓ Different Light Sources
- ✓ Modeling Everything



- *PCS extensions*
- *Tag Array and Tag Structure tag types*
- *multiProcessingElement extensions*

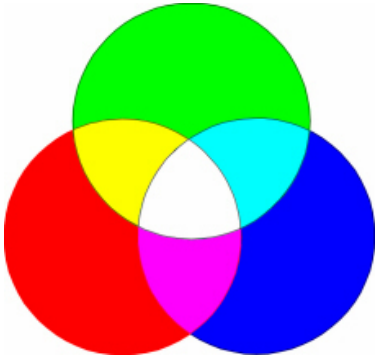
Other Extensions





CxF

New Profile Class



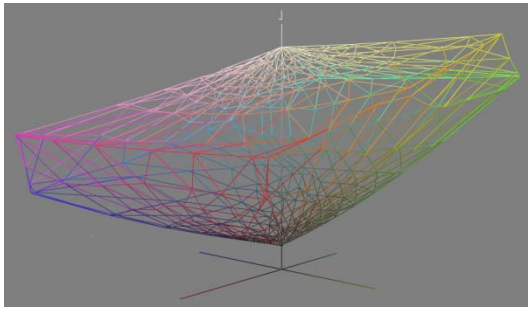
- **The “three component color encoding” profile class**
  - Class signature is ‘enc3’ (656e6333h)
  - Allows a profile to specify **what color data is**
  - Does **NOT specify how color data should be transformed**
  - Prescribes minimally filled ICC header
  - A single required tag *referenceNameTag* (based on utf8TextType)
- **Two modes of operation:**
  1. When *referenceNameTag* specifies the name of a standardized encoding space (Example: “sRGB”)
     - An optional *colorEncodingParamsTag* (based on tagStructType) allows overrides of the named encoding space specification (defined by ISO 22028-1) to be defined

sRGB

  2. When *referenceNameTag* is “ISO 22028-1”
     - A required colorSpaceNameTag (based on utf8TextType) defines the name to associate with the color space encoding
     - A required *colorEncodingParamsTag* (based tagStructType) defines the relevant parameters of the color space encoding according to ISO 22028-1
- **The CMS provides an “appropriate” transform**

Optional Gamut Boundary Description Tags

- **Optional gamut boundary description tags can be included to define colorimetric gamut boundary for each rendering intent**
- **Encodes:**
  - Array of vertices on surface of gamut
  - Array of triangular faces using vertex indices
  - Colorimetric PCS values for each vertex
  - Optional Device values for each vertex



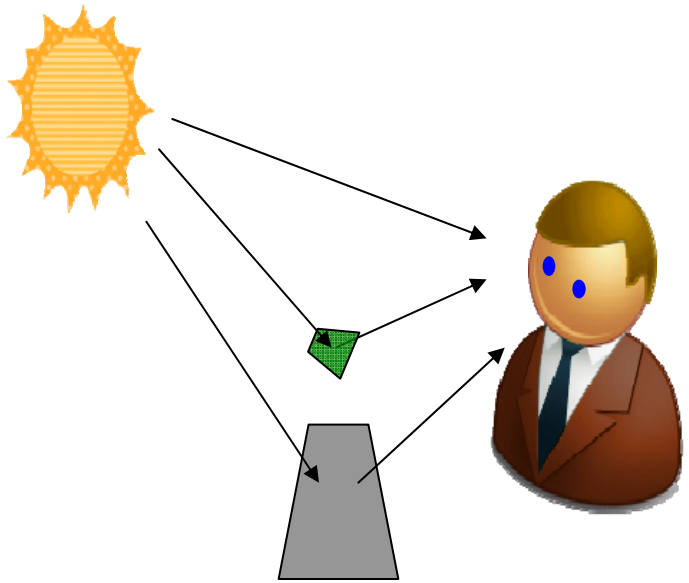
- **Allows for 3-dimensional aspects of gamut to be identified and used by applications and dynamic CMM's**

Encoding Measurement data using CxF™

- **New Optional CxFTag *(based upon either utf8TextType or zipUtf8TextType)***
  - The CxF™ data in this tag is an UTF-8 encoded XML document
  - Contains characterization set used to create the profile
  - Contains measurement data used to create profile
  - May contain any other data that conforms to the CxF™ specification
  - CxF™ is specified at http://www.colorexchangeformat.com
  - Compression using a zipUtf8TextType is a much more efficient way of storing XML data
- **Note: CxF based tags can also potentially be placed inside of named color structures as a sub-tag to associate measurement data with each named color**

Color Management Challenges Addressed by Other Extensions

- ✓ Characteristics of Surfaces
- ✓ Modeling Everything



Example Profiles

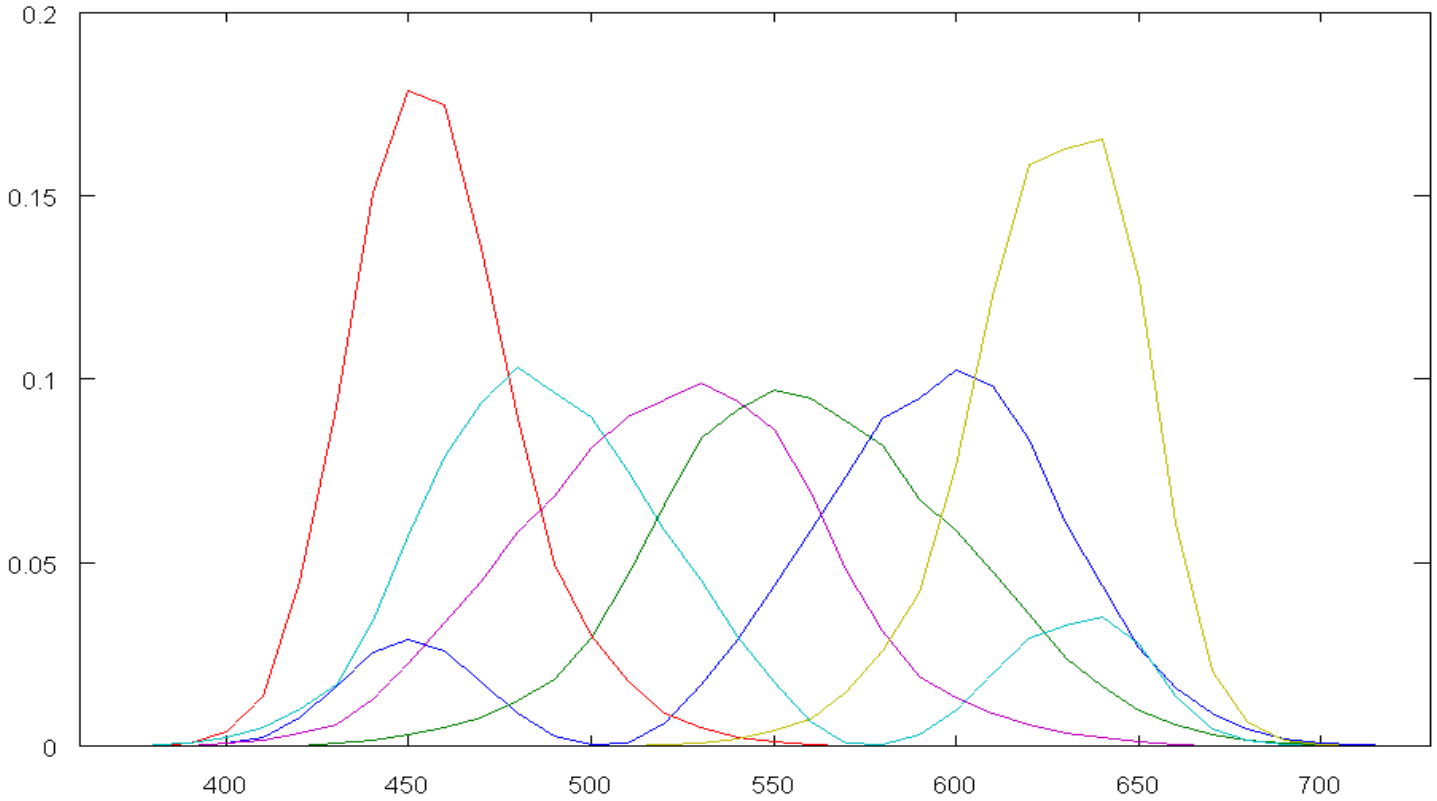
IccLabs General Profile Contents

- **Display / Device / Color Space Profiles**
  - Header (with spectral PCS)
  - Metadata Tags
  - *Profile Connection Conditions Tags*
  - *Colorimetric Transform Tags*
    - AtoBx / BtoAx : *lut8, lut16, lutAtoB, lutBtoA, multiProcessElementType*
  - *Spectral Transform Tags*
    - DtoBx / BtoDx : *multiProcessElementType*
  - *Note 1: PCS and Spectral PCS entries in header determine whether colorimetric and/or spectral transform tags are needed*
  - *Note 2: Profiles are valid when only relative or absolute transforms are present*
- **Device Link Profiles**
  - Header
  - Metadata Tags
  - Transform Tags
    - AtoB0 : *lut8, lut16, lutAtoB, multiProcessElementType*
- ***Named Color Profiles***
  - Header (with spectral PCS)
  - Metadata Tags
  - Profile Connection Conditions Tags
  - Transform Tag
    - Named Color Table : *namedColorTagType, tagArrayType(namedColorArray)*
- **3 Color Space Encoding Profiles**
  - Minimal Header
  - Encoding Space Type (and Name)
  - *Optionally override color space encoding parameters : tagStructType*

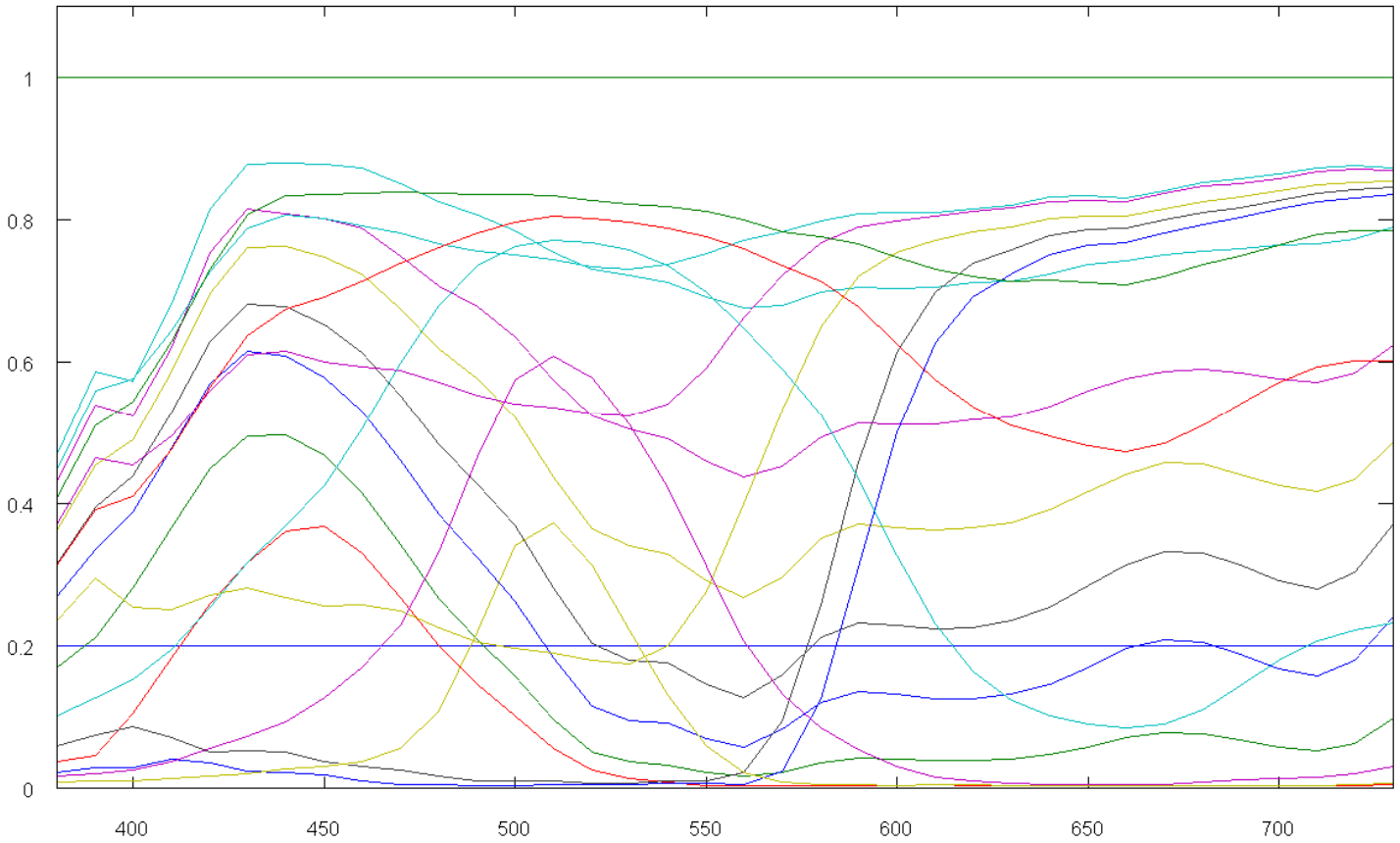
Example Testing Profiles

- **Basic V4 sRGB profile**
  - sRGB_v4_preference.icc
- **Six channel to/from spectral reflectance profile**
  - SixChannelCameraRef.iccxml / .icc
- **Named colors with spectral information**
  - NamedColor.iccxml / .icc
- **Test calculator multiProcessElement**
  - srgbCalcTest.iccxml / .icc
- **Spectral Abstract Profiles (Can be used to define PCC)**
  - Spec400_10_700-D50_2deg.iccxml / .icc
  - Spec400_10_700-D50_10deg.iccxml / .icc
  - Spec400_10_700-D65_2deg.iccxml / .icc
  - Spec400_10_700-D65_10deg.iccxml / .icc
  - Spec400_10_700-D93_2deg.iccxml / .icc
  - Spec400_10_700-D93_10deg.iccxml / .icc
  - Spec400_10_700-IllumA_2deg.iccxml / .icc
  - Spec400_10_700-IllumA_10deg.iccxml / .icc

SixChannelCameraRef.icc Sensitivities



NamedColor.icc Reflectances



Examples Using ReflccLabs

Using iccApplyNamedColorCmm

iccApplyNamedColorCmm is a command line tool in RefIccLabs that allows color values in a data_file to be applied against IccLabs based profiles

Usage parameters can be determined by invoking the tool with no arguments

Note: The output from iccApplyNamedColorCmm can be redirected and later used as an input data_file

data_file format

- First line contains color space signature of data
- Second line contains encoding type of data
- Following lines contain space delimited channel data
  - Special format for named colors

Demo

What's Next?

- **IccLabs Proposals and RefIccLabs Implementation Need Validation!!!**
  - Make sure proposals and implementation correctly reflect each other
  - Make sure that color management challenges are met
  - Eliminate possible "lurking" problems
- **Proposals need to be organized into (a) working specification document(s)**
- **Ratify specifications**
- **Other implementations**

How to get involved?

- **Documentation and Implementation are available to all members of the ICC**
  - Universities that are Honorary members of ICC have access
- **Become an individual member of the ICC Architecture Working Group**
  - Contact Kip Smythe for more information
- **Help validate, test, refine, and document**
- **Implement and incorporate new possibilities**

Sumary

- **The work of IccLabs provides significant developments over existing ICC V4 capabilities by meeting color management challenges and thus fulfilling the ICC mission:**
  - “To create, promote and encourage the standardization and evolution of open, vendor-neutral, cross-platform colour management system architectures and components”
- **RefIccLabs provides a reference implementation that allows new color management workflows to be developed and tested**
- **Get involved!!!!!**

Thank You!

Questions?