

ICC Votable Proposal Submission Dictionary Type and Metadata TAG Definition

Proposers: **Ann McCarthy (Lexmark International, Inc.),
Max Derhak (Onyx Graphics),
Lars Borg (Adobe Systems, Inc.)**

Submission date: **January 26, 2010**

1. Introduction

Scanners and printers, for example, can provide adjusted color behavior options for media, quality modes, document types, ink characteristics, etc. Similarly, in general, devices that handle digitally encoded color content can have various modes and use various materials such that a specialized profile can be used to optimize the device color behavior for those conditions. In the most general sense, a profile may have been built for a particular use that cannot be described or identified using the currently defined ICC profile tags.

The current gap in the profile identification information contained in ICC profiles is addressed in a variety of ways in the interfaces between devices and operating systems, between devices and applications, between devices and users, between operating systems and users, between operating systems and applications, between applications and users, etc. The current methods do not provide consistent, complete, and flexible coverage of profile selection options in general multi-vendor workflow environments. Most significantly, broadly implemented automation of correct profile selection is made difficult if not completely unworkable by the lack of consistent digitally encoded information describing each device profile's correct use case.

The use of ICC profiles and the reliance on color management is increasing with equipment that previously did not rely on color management, e.g., office equipment. With the use of color management in environments such as a business office, the user expectation is that color management is fully automated, with no requirement for the user to select profiles or even understand their function. This is true in home environments as well. Even in commercial digital print shops, workflows with non-expert users can benefit from improved automation in profile selection.

This proposal provides a new tag type for future tags to be used in ICC profiles. Tags based on this tag type may be used by a profile builder to identify the usage conditions for which a profile is intended. This tag type definition provides a consistent structure on which to base future work defining descriptive profile parameter names and values. Tags based on this tag type may be defined incorporating profile information parameters and parameter values registered at <http://www.color.org/dictionary>.

This proposal is to create a tag type for tags that can be incorporated into any ICC profile of any version. Tags based on the tag type of this proposal are not intended for use in the computational operation of any CMM and so are not restricted to a particular ICC profile version. This proposal provides a structure for use in consistently and openly encoding profile description and usage details in a profile, to enable improved manual and/or automated profile search and selection, so that a selected profile corresponds to the current color management use case.

2. The acceptance of this proposal will result in:

A tag type and tag to be added to the next minor revisions of the ICC.1 specifications, both ICC.1:2001-04 (version 2) and ICC.1:2010 (version 4). The establishment of an ICC metadata registry on the ICC web site.

3. Nature of the proposal

This proposal defines a new tag type for use in optional tags. This proposal includes establishing an ICC metadata registry. This proposal defines an optional new tag using this new tag type.

4. Votable Proposal

The ICC technical secretary shall establish a registry for metadata items.

The ICC technical secretary shall manage the metadata registry. The metadata registry shall be open to non-ICC members.

For each metadata item, the registry shall include the following mandatory and may include the following optional entries:

- Owner: Mandatory entry. Identifier for the organization or standard providing the entry specification.
- Change date: Mandatory entry. Date of latest change of this entry (date of creation if never changed).
- Category: Mandatory entry. A metadata subgroup name selected from a list of subgroup names provided by the ICC metadata registry, for example, “printing conditions”, or defined by the owner if none of the previously defined categories is suitable.
- Name: Mandatory entry. Name string permitted in the name field in the dictType tag structure. The name shall be unique within the registry.
- Meaning: Mandatory entry. The meaning field shall include a textual description of what this metadata item defines. This entry should also include a link to a reference document providing the technical detail for the metadata item if correct usage of the item requires further explanation.
- Usage Restrictions: Optional, conditionally Mandatory, entry. Optional for non-deprecated metadata items. Textual description for when this metadata item is required to be used and/or when the metadata item shall not be used. May be used to identify conflicts when using this metadata item with other tags. If the owner has deprecated use of the metadata item, the Usage Restriction field entry shall be Mandatory and shall contain ‘Deprecated’.
- Display name element: Any number of Optional entries per Name. Textual representations of the Name entry in various languages suitable as Display Name Element data. When Display Name Element is not entered (it is empty), then no Display Name Element is included in the tag. (This indicates to use the Name entry value for display.) When no Display Name Element display text is provided and the Name entry value is not intended for display, the Display Name Element entry shall be present and shall be set to “”, a zero length string.
- One or more value entries for the Name, defining the closed set of permitted values, each including:
 - o Value: Mandatory entry (at least one). Specification of a value string permitted in the value field in the dictType tag structure. Each value entry may be a single predefined string value (such as ‘Off’ or “”), or a variable string value defined by a format

description (such as ‘one integer in range 1...9999’, ‘list of integers’ or ‘any five character value which contains no (‘_’)', or special case “offset==0” which indicates no Value shall be stored.

- Meaning of value: Optional entry. Textual description of what this value implies.
- Display Value Element: Any number of Optional entries per Value. Textual representations of a Value entry in various languages suitable as Display Value Element data. When Display Value Element is not entered (it is empty), then no Display Value Element is included in the tag. (This indicates to use the Value entry value for display.) When no Display Value Element display text is provided and the Value entry value is not intended for display, the Display Value Element entry shall be present and shall be set to “”, a zero length string.

In general, non-localized text entries, Name and Value, shall be in English. Exceptions include proper names, and non-English standards. Any registry entry may be modified or extended. Names and their corresponding Values shall not be deleted from the registry, but Name entries may be marked deprecated. An entry's change date indicates the latest change. See Appendix A to this proposal for an example of registry entries.

ICC shall add this tag to the tag listing in the ICC specification:

x.y metadataTag

Tag signature: ‘meta’ (6D657461h)

Allowed tag types: dictType

This tag contains a set of metadata items for the profile.

The names and values in the set shall be taken from the ICC metadata registry, available on the ICC web site <http://www.color.org/>. Display elements should be taken from the metadata registry, as this provides common localizations.

ICC shall add this tag type to the tag type listing in the ICC specification:

x.y dictType *[add a new type in the tag type definition section in each specification (different section numbers in version 2 and version 4 specs)]*

The dictType structure contains a dictionary array of name-value pairs with each name being uniquely associated with a single value. Each name and value can optionally be associated with localized text strings for display purposes.

The byte assignment and encoding shall be as given in Table A and Table B.

Table A – dictType encoding

Byte position	Field length (bytes)	Content	Encoded as
0 to 3	4	‘dict’ (27646963h) type signature	27646963h

4 to 7	4	Reserved, shall be set to 0	0
8 to 11	4	Number of name-value records (<i>m</i>)	uint32Number
12 to 15	4	The length of each name-value record, in bytes. (<i>n</i>) The value shall be 16, 24, or 32.	uint32Number
16 to 15+n	n	The first name-value record	Name-Value record structure
16+n to 15+m*n	n*(m - 1)	Additional name-value records as needed	Name-Value record structure [m-1]
16+m*n to end	Variable	Storage area of strings of Unicode characters and mluc tags	

Table B – Name-Value record structure

Byte position	Field length (bytes)	Content	Encoded as
0 to 3	4	name string offset: the offset from the start of the tag to the start of the name string, in bytes	uint32Number
4 to 7	4	name string size: the number of bytes used by name string	uint32Number
8 to 11	4	value string offset: the offset from the start of the tag to the start of the value string, in bytes	uint32Number
12 to 15	4	value string size: the number of bytes used by value string	uint32Number
16 to 19	4	display name element offset: the offset from the start of the tag to the start of the display name element, in bytes	uint32Number
20 to 23	4	display name element size: the number in bytes used by the display name element	uint32Number
24 to 27	4	display value element offset: the offset from the start of the tag to the start of the display value element, in bytes	uint32Number
28 to 31	4	display value element size: the number of bytes used by the display value element	uint32Number

The value in the length of each name-value record field shall determine how many entries shall be present in each name-value record.

- When the length value is 16, each name-value record shall be 16 bytes long and only the length and offset fields for name and value items shall be present.
- When the length value is 24, each name-value record shall be 24 bytes long and only the length and offset fields for name, value and display name items shall be present.
- When the length value is 32, each name-value record shall be 32 bytes long and the length and offset fields for name, value, display name and display value items shall be present.

In the general use of dictType, there may be no localized values, so 16 would be appropriate. In other use cases, localized display values are needed, and 32 would be used. When using localization for value fields and not localizing names, use 32 bit name-value records with the display name element offsets set to zero.

A name string shall be present for each name-value record and name string size shall be greater than zero. Other data items referenced by the name-value record are optional according to dictType, although particular dictType tag definitions may impose restrictions.

Both the name string and value string shall be Unicode strings, encoded as UTF-16BE, and shall not be NULL terminated. For the specification of Unicode, see The Unicode Standard published by The Unicode Consortium or visit their website at <http://www.unicode.org>. UTF-16BE is defined as 16-bit Unicode characters in big-endian encoding.

Name strings shall contain at least one Unicode character, and the string contents of each name string shall be unique within a dictType tag. In general, a zero-length string (NUL) is valid for value strings, and shall be indicated by a non-zero value string offset and a value string size equal to zero.

NOTE: Value string = NUL may be restricted in particular dictType tags.

A value string offset, display name element offset or display value element offset of zero shall indicate that the corresponding data item is not present and is undefined. When an offset is zero, the meaning of the corresponding size field is undefined, and that size field should be zero.

When a localized display name element or display value element is undefined (offset equal zero), no translation is provided for the corresponding name string or value string, and the name string or value string may be displayed. This is equivalent to the behavior for all name strings and value strings when record size is 16.

Alternatively, a defined display name element offset (nonzero) with a display name element size equal to zero indicates that the name string is not intended for display. Similarly a defined display value element offset (nonzero) with a display value element size equal to zero indicates that the value string, if provided, is not intended for display. A localized display value may be provided without a localized display name.

NOTE: It is permitted to share data between the name-value records of a dictType tag. For example, the offsets for the value strings can be identical, as well as the offsets for display value elements can be identical.

Example for value string offset, display name element offset or display value element offset:

If Offset == 0

Then item is undefined (Length can be ignored, when offset is 0)

Else

*"If ((Offset >= 20+n*m) && ((Length >= minSizeOfItemType) || (Length = zero))
&& (Offset + Length <= end + 1))*

Then item is defined.

If ((item == value string) && (Length == 0))

Then value string is NUL string

Else if ((item == display name element) && (Length == 0))

Then name string is not for display use and no display name is provided

Else if ((item == display value element) && (Length == 0))

*Then value string is not for display use and no display value is provided
Else ERROR (offset is not zero and offset or length are invalid)*

where

minSizeOfItemType == 0 for value strings

minSizeOfItemType == 28 for mluc items, i.e., minimum mluc length = 28 bytes, or is zero as allowed with display name element size and display value element size.

-end example.

Unless otherwise stated, numbers shall be encoded in the value string as follows:

- A number shall be encoded as zero or more blanks and/or tabs, an optional '+' or '-' sign, a string of decimal digits that may contain one decimal point '.', and an optional exponent part. The exponent part shall consist of 'e' or 'E', an optional '+' or '-' sign, and one or two decimal digits. The exponent shall indicate a power of 10.
- Multiple numbers stored in a single value string shall be separated by one comma ',' between adjacent numbers.

The optional display name element shall be encoded as a multiLocalizedUnicodeType tag.

The optional display value element shall be encoded as a multiLocalizedUnicodeType tag.

The name strings, value strings, display name elements, and display value elements shall be stored in the storage area after the last name-value record. Thus, the offsets to stored data items shall be in the range $20+m*n$ to the end of the tag. These data items may be stored in any order in this area.

All offsets shall be integer multiples of 4, as each data item shall start on a 4-byte boundary. To achieve this, each data item shall be followed with zero to three 00h pad bytes as needed.

NOTE: The definition of a tag using this tag type may impose restrictions, including permitted and required names and values.

5. Applications and Workflows

There are four different software roles [usage scenarios] that interact with ICC profiles. Each of these is affected and interacts differently with respect to profile conditions and potentially with respect to tags based on dictType. A particular software application may address one or more of these roles.

Build/edit profile tool software:

Profile building software may populate tags based on dictType when a profile is built. Such tags may also be added to a profile after it is built. Standalone software can be devised to edit a profile to add such informational tags. For example – imagine a user who manually edits a printer profile built for one paper so that it works well on another paper [in a particular print system, etc.]. The user can choose to add an informational dictType tag to the profile at that time so that their print system can automatically select the profile matched to the user's paper selection. Software tools that validate profiles fall into this profile tool category.

CMM software:

No indication to use tags based on dictType.

Install profile software:

Optionally, software used to install profiles at the time of device installation, workflow configuration, etc., may be updated to include the function of updating or perhaps automatically creating PPDs or database entries with the content from dictType tags — if necessary re-encoding that information into a system-specific form.

Runtime profile selection tool/OS/application/device interface software:

Such tools can utilize dictType tags to refine algorithms that automate the selection of correct profiles and also minimize or eliminate user profile selection effort. In some cases this will be carried out using dictType tag information content reformatted into a proprietary database or PPD that is populated at profile install time. In other cases the profile selection software will use such tag content directly from profiles. This is a system and implementation dependent choice.

Appendix A. Sample Registry Entries

Owner	Change Date	Category	Name	Meaning of name	Usage Restriction	Display Name Element:	Values	Meaning of value	Display Value Element: en
Req	Req	Req	Req	Req	Opt/Req	Opt	Req	Opt	Opt
AFPC	5-Nov-09	Printing/Display	MediaColor	The color of the media, URL: xxx	Optional		noc (X'006E 006F 0063)	No-color	clear
							wht (X'0077 0068 0074)	white	white
							pnk (X'0070 006E 006B')	pink	pink
							ylw (X'0079 006C 0077')	yellow	yellow
							blu (X'0062 006C 0075')	blue	blue
							grn (X'0067 0072 006E')	green	green
							buf (X'0062 0075 0066')	buff	buff
							gdr (X'0067 0064 0072')	goldenrod	goldenrod
							red (X'0072 0065 0064')	red	red
							gry (X'0067 0072 0079')	gray	gray
							ivy (X'0069 0076 0079')	ivory	ivory
							org (X'006F 0072 0067')	orange	orange
							Any three character value which contains no ('_')	custom	mycolor
AFPC	5-Nov-09	Printing/Display	ManufacturerName	Name of device manufacturer, URL: xxx	Optional		Any five character value which contains no ('_')	For IPDS printers, the ManufacturerName, the DeviceType, and the DeviceModel must be provided in accordance with the IPDS definitions. This information can be obtained from the XOH Obtain Printer Characteristics (OPC) under Product Identifier Self-Defining Field in the Intelligent Printer Data Stream Reference, S544-3417.	

AFPC	5-Nov-09	Printing	MediaWeight	weight of the media rounded to the nearest whole number of grams per square meter, URL: xxx.	Optional	en:Media Weight; ge:Media-Gewicht; it:Peso di media; fr:Poids de medias;	Number in range 1-999		
------	----------	----------	-------------	--	----------	--	-----------------------	--	--