

ICC Votable Proposal Submission

Profile Sequence Identifier Tag

Proposer: Manish Kulkarni, Adobe Systems Inc.

Date: November 27, 2006

Proposal Version: 1.0

1. Introduction

In certain workflows using ICC Device Link Profiles, it is necessary to identify the original profiles that were combined to create the Device Link Profile (see the Applications and Workflows section for an example of such a workflow).

The profileSequenceDescTag in current ICC Device Link Profiles defines an array of structures containing the following information about the original profiles that were combined to create the final profile:

- Device Manufacturer Signature
- Device Model Signature
- Device Attributes
- Device Technology
- Device Manufacturer Description
- Device Model Description

These attributes are not sufficient to identify the original profiles that were used to create the Device Link Profile.

This proposal would add a new optional tag containing the following information about the original profiles that were combined to create the final profile:

- Profile ID
- Profile Description

This information would enable the identification of the original profiles that were combined to create the final profile.

2. The acceptance of this proposal will result in a:

minor revision of Specification ICC.1:2004-10

3. Nature of the proposal:

This is a **new technical addition** to the specification.

A new optional tag profileSequenceIdentifierTag using a new type profileSequenceIdentifierType is added to the Profile Specification.

4. Votable Proposal

- **Add new Clause:**

9.2.X profileSequenceIdentifierTag

Tag signature: 'psid' (70736964h)

Allowed tag type: profileSequenceIdentifierType

Structure containing information for identification of the profiles used in a sequence. This tag is typically used in Device Link Profiles to identify the original profiles that were combined to create the final profile.

- **Add new Clause:**

10.Y profileSequenceIdentifierType

This type is an array of structures, each of which contains information for identification of a profile used in a sequence.

When used, the byte assignment and encoding shall be as given in Table A.

Table A – profileSequenceIdentifierType structure

Byte Position	Field Length (bytes)	Content	Encoded as...
0..3	4	'psid' (70736964h) type signature	
4..7	4	reserved, must be set to 0	
8..11	4	Count (N), specifying number of structures in the array	uint32Number
12..11+8N	8 x N	Positions Table for Profile Identifiers	positionNumber[]
12+8N..end		Profile Identifier structures – see Table B	

The offsets stored in the Positions Table shall be relative to the start of the tag.

NOTE: It is permitted for Profile Identifier structures to be shared. So it is possible that a positionNumber in the Positions Table is identical to another positionNumber in the Positions Table.

Each Profile Identifier structure shall start on a 4-byte boundary. To achieve this, each structure shall be followed by up to three 00h pad bytes as needed.

Each Profile Identifier structure has the format shown in Table B:

Table B – Profile Identifier structure

Byte Position	Field Length (bytes)	Content	Encoded as...
0..15	16	Profile ID (see below)	See 7.2.18
16..end	variable	Profile Description (see below)	multiLocalizedUnicodeType

Profile ID: If a profile contains a Profile ID in the Profile Header, it shall be used in the Profile Identifier structure. If a profile does not contain a Profile ID in the Profile Header, either an all-zero Profile ID or a computed Profile ID shall be used in the Profile Identifier structure.

Profile Description: For profiles conforming to ICC Profile Specification ICC.1:2001-12 (ICC V4.0.0) and later, the entire multiLocalizedUnicodeType contents of the Profile Description Tag shall be included in the Profile Identifier structure. For profiles conforming to ICC Profile Specification ICC.1:2001-04 (ICC V2.4.0) and earlier, the contents of the textDescriptionType Profile Description Tag shall be converted to multiLocalizedUnicodeType and used in the Profile Identifier structure.

NOTE: One way of creating a multiLocalizedUnicodeType from a textDescriptionType is by converting the 7-bit ASCII part of the textDescriptionType to a 'enUS' Unicode string by mapping the 7-bit ASCII characters to 16-bit Unicode characters, and storing the 'enUS' Unicode string in the multiLocalizedUnicodeType.

5. Applications and Workflows

Consider a CMYK Image with an embedded “U.S. Web Coated (SWOP)” profile. A user wishes to convert the image to “Euroscale Coated” using a Device Link Profile, using an application that supports such a conversion, and save a new CMYK Image with an embedded “Euroscale Coated” profile. Currently, the user would need to do the following:

- Open the Image in the application, using the embedded “U.S. Web Coated (SWOP)” profile
- Select a Device Link Profile to Convert To. The user may have many Device Link Profiles on the system, so the user must be careful to select the correct profile.
- Convert the image using the selected Device Link Profile
- After conversion, the user must explicitly find the “Euroscale Coated” profile, and assign it to the converted image.

This simple operation involves a number of steps, which are error-prone.

With the addition of the Profile Sequence identifier tag, the user would do the following:

- Open the Image in the application, using the embedded “U.S. Web Coated (SWOP)” profile
- Select a Device Link Profile to Convert To. Although the user may have many Device Link Profiles on the system, the application can choose to display only those profiles where the First Profile in the Profile Sequence Identifier tag corresponds to the Image Profile “U.S. Web Coated (SWOP)”.
- Convert the image using the selected Device Link Profile
- After conversion, the application can automatically find and assign the Last Profile in the Profile Sequence Identifier tag (e.g. “Euroscale Coated” profile) to the converted image.

In this scenario, the application uses the Profile ID and Profile Description from the Profile Sequence Identifier Tag to assist the user and simplify the workflow.