

INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11 MPEG/m53680
April 2020, Online

<i>Source:</i>	Apple Inc.	
<i>Status:</i>	Input document	
<i>Title:</i>	G-PCC: Extensible signalling of attribute descriptions	
<i>Author(s):</i>	David Flynn Khaled Mammou	<i>davidflynn@apple.com</i> <i>kmammou@apple.com</i>

Abstract

In the current G-PCC design [1, N19088], attribute descriptions are signalled in the SPS. The type of attributes is signalled by either a known identifier or by an unmanaged fourcc code. Each attribute description describes a set of fixed parameters without any means to convey specific parameters or to avoid signalling irrelevant parameters. This document proposes to replace the fourcc signalling with registered identifiers, along with a mechanism to generalise attribute description signalling.

Current design

Each attribute description may signal the following metadata:

- `attribute_label_four_bytes` / `known_attribute_label` – A means to identify the sort of attribute (colour (bgr), reflectance, temperature, etc.)
- `attr_instance_id` – A means to distinguish multiple attributes of the same type.
- `cicp_colour_primaries_idx` – Standardised colour information.
- `cicp_transfer_characteristics_idx` – *idem*.
- `cicp_matrix_coefficients_idx` – *idem*.
- `cicp_video_full_range_flag` – *idem*.

Drawbacks

While simple, there are two issues with this approach.

Firstly, unregistered identifiers are unpleasant to manage. It is highly likely that the fourcc code “rgb” will be used with various different connotations. Fortunately for colour information this might be lessened by the presence of CICP parameters. However, the fundamental issue still stands.

Secondly, not all attribute data needs CICP information, other attribute data will find that the CICP information is insufficient to describe their nature (imagine a more complicated transfer function for reflectance data).

Proposal — Registration identifiers

The solution to the first problem is to use registered identifiers. Even fourcc codes from AVI are supposed to be registered with an authority (Microsoft).

Rather than creating a new registration authority, a simple solution is to reuse generic identifiers issued by another authority. Two notable examples are:

- Manufacturer specific (terminal provider oriented) codes (Rec. ITU-T T.35)
- International object identifiers (Rec. ITU-T X.660 | ISO/IEC 9834-1)

Terminal provider oriented codes

Recommendation ITU-T T.35 provides the means for registering terminal provider codes with a national registration authority. Identifiers consist of a single byte country identifier, a conditional extension byte, followed by an identifier encoded according to a scheme provided by the issuing registration authority. Terminal provider oriented codes permit the terminal provider to further extend the issued code according to their own definition.

There is no requirement for a T.35 code to have a fixed length.

For example, the identifier B4h 00h XXh YYh identifies a terminal provider registered by the relevant assignment authority in the United Kingdom. The terminal provider with this code may themselves issue terminal provider oriented codes according to their own semantics.

Example semantics adapted from HEVC:

```
...
itu_t_t35_provider_oriented_code_length = u(8)
itu_t_t35_country_code = u(8)
if (itu_t_t35_country_code = 255)
    itu_t_t35_country_code_extension_byte = u(8)
for (i = 0; i < itu_t_t35_provider_oriented_code_length; i++)
    itu_t_t35_provider_oriented_code_byte[i] = u(8)
```

itu_t_t35_country_code shall be a byte having a value specified as a country code by Rec. ITU-T T.35 Annex A. **itu_t_t35_country_code_extension_byte** shall be a byte having a value specified as a country code by Rec. ITU-T T.35 Annex B.

itu_t_t35_payload_byte shall be a byte containing data registered as specified in Rec. ITU-T T.35.

The ITU-T T.35 terminal provider code and terminal provider oriented code shall be contained in the first one or more bytes of the **itu_t_t35_payload_byte**, in the format specified by the Administration that issued the terminal provider code.

A more concise syntax description could be used to provide such codes in place of the **attribute_label_four_bytes** syntax element.

International object identifiers

The ISO/IEC and ITU-T define Abstract Syntax Notation One (ASN.1) which includes the notion of object identifiers (OID). Encoding rules such as the basic encoding rules (BER) or distinguished encoding rules (DER) describe how to represent such abstract identifiers in a binary format.

Recommendation ITU-T X.660 | ISO/IEC 9834-1 defines a tree structure that may be represented using object identifiers. The tree structure provides means to delegate sub-trees for management by registration authorities.

For example, it is possible to register with ANSI an organisation name to obtain an OID under the sub-tree 2.16.840.1. An organisation may then create additional OIDs under the sub-tree 2.16.840.1.XXX (where XXX is the organisation identifier).

The BER and DER encoding of an OID are naturally variable length.

Proposal — Replacing `attribute_label_four_bytes`

We propose to replace the `attribute_label_four_bytes` syntax element with one of the above methods. While standards such as HEVC have successfully employed ITU-T T.35 terminal provider codes, it should be noted that G-PCC is not an ITU standard. That said, T.35 codes are likely to be shorter than OIDs.

Therefore, until such time as it can be determined that T.35 codes are suitable for use, it is proposed to use OIDs with the following syntax. To avoid the use of any ASN.1 object type being conveyed, the DER tag of the encoding rule is implied.

```
sequence_parameter_set() {  
    ...  
    if (!known_attribute_label_flag) {  
        attribute_label_oid = oid()  
    }  
    ...  
}
```

`attribute_label_oid` identifies the type of attribute according to an international object identifier as specified in Recommendation ITU-T X.660 | ISO/IEC 9834-1.

Generalised attribute descriptions

Some attributes may require additional metadata in order to be correctly interpreted. To facilitate this, it is proposed to allow custom description blocks to be signalled.

Description blocks use a type-length-value format. This design permits decoders that do not understand a particular attribute to skip over the opaque data.

The type value of each parameter block is divided into three categories:

- Known types (values 2–127) convey parameter blocks defined in the G-PCC specification.
- Implicit label specific types (values 128–255) convey parameter blocks defined by the specification for the relevant `attribute_label_oid`.
- Explicit label specific types (values 1 and 2) are identical to implicit label specific types, except that they include a relevant OID which may differ from the current attribute label.

Illustrative specification text:

Sequence parameter set syntax

```
sequence_parameter_set() {  
    ...  
    for (attrIdx = 0; attrIdx < num_attributes; attrIdx++) {  
        ...  
    }
```

```

        num_attribute_parameters = u(8)
        for (i = 0; i < num_attribute_parameters; i++)
            attribute_parameter()
    }
    ...
}

```

Attribute parameter syntax

```

attribute_parameter() {
    attr_param_type = u(8)
    attr_param_len = u(8)
    if (attr_param_type == 0) {
        itu_t_t35_country_code = u(8)
        if (itu_t_t35_country_code < 255)
            attribute_parameter_data(attr_param_len - 1);
        else {
            itu_t_t35_country_code_extension_byte = u(8)
            attribute_parameter_data(attr_param_len - 2);
        }
    } else if (attr_param_type == 1) {
        attribute_parameter_oid = oid()
        attribute_parameter_data(attr_param_len - Len(attribute_parameter_oid))
    } else if (attr_param_type == 2) {
        cicp_colour_primaries_idx = ue(v)
        cicp_transfer_characteristics_idx = ue(v)
        cicp_matrix_coefficients_idx = ue(v)
        cicp_video_full_range_flag = ue(v)
    } else if (attr_param_type < 128)
        /* Reserved for future use */
    else
        attribute_parameter_data(attr_param_len)

    byte_align()
}

```

Attribute parameter data syntax

```

attribute_parameter_data(len) {
    for (i = 0; i < len; i++)
        attr_param_byte[i] = u(8)
}

```

Attribute parameter semantics

attr_param_type identifies the type of parameter data conveyed in the attribute parameter block.

attr_param_type equal to 0 indicates that attr_param_byte shall be a byte containing data registered as specified in Rec. ITU-T T.35. The ITU-T T.35 terminal provider code and terminal provider oriented code shall be contained in the first one or more bytes of the attribute_parameter_data syntax structure, in the format specified by the Administration that issued the terminal

provider code.

`attr_param_type` equal to 2 indicates that the attribute parameter consists of codec independent code point data.

`attr_param_len` indicates the length in bytes of the attribute parameter.

References

- [1] 3DG, “Text of ISO/IEC 23090-9 DIS Geometry-based PCC,” ISO/IEC JTC1/SC29/WG11, 129th meeting, Brussels, Tech. Rep. w19088, Jan. 2020.