
SMPTE ST 2042-1 (VC-2) Data Tables

Release 2.0.0

BBC

Jul 21, 2021

CONTENTS:

1	Introduction	1
2	The <code>vc2_data_tables</code> Python module	3
2.1	(10.5) Parse Info Block	4
2.2	(11) Picture coding mode identifiers	4
2.3	(11.4.6) Preset frame rates	4
2.4	(11.4.7) Preset pixel aspect ratios	4
2.5	(11.4.9) Signal ranges	5
2.6	(11.4.10.2) Color Primaries	5
2.7	(11.4.10.3) Color Matrices	5
2.8	(11.4.10.4) Transfer functions	6
2.9	(11.4.10.1) Colour specifications	6
2.10	(11.3) Base Video Formats	6
2.11	(15.4.4) Lifting filters	7
2.12	(C.2) Profiles	8
2.13	(D) Quantisation matrices	8
2.14	(ST 2042-2) Levels	8
3	CSV Files and Conventions	9
3.1	CSV Dialect	9
3.2	Headings	9
3.3	Comments	10
3.4	Ditto	10
3.5	Lists	10
	Index	11

INTRODUCTION

The `vc2_data_tables` (page 3) Python package contains machine-readable values for the constants and data tables in the SMPTE ST 2042-series of standards relating to the VC-2 professional video codec¹.

Specifically, the values in this package are those defined in:

- SMPTE ST 2042-1:2017² (VC-2)
- SMPTE ST 2042-2:2017³ (VC-2 Level Definitions)

These values may be used either via the `vc2_data_tables` (page 3) Python module or, in most cases, values may be read from CSV formatted data files located in `vc2_data_tables/csv/* .csv`.

To read about the Python module, see *the module API documentation* (page 3).

For an overview of the CSV formatting conventions see the *CSV Files and Conventions* (page 9) documentation.

Finally, you can find the source code (and CSV data) for `vc2_data_tables` (page 3) on GitHub⁴.

Note: This documentation is also available to browse online in HTML format⁵.

¹ <https://www.bbc.co.uk/rd/projects/vc-2>

² <https://ieeexplore.ieee.org/document/7967896>

³ <https://ieeexplore.ieee.org/document/8187792>

⁴ https://github.com/bbc/vc2_data_tables/

⁵ https://bbc.github.io/vc2_data_tables/

THE VC2_DATA_TABLES PYTHON MODULE

The `vc2_data_tables` (page 3) module defines a number of constants and tables based on values published in the SMPTE ST 2042-series of standards documents. References to the standard are shown in brackets and, unless otherwise stated, refer to SMPTE ST 2042-1:2017, the main VC-2 specification.

Enumerated indices defined by the VC-2 specification (for example the parse codes which appear in parse info blocks (10.5.2)) are defined as `IntEnum`⁶ types (e.g. `ParseCodes` (page 4)) with informative names assigned to each index. These may be used interchangeably with plain integer values if required or preferred.

Tables of values (for example the table of preset frame rates in (Table 11.1)) are typically represented as dictionaries (e.g. `PRESET_FRAME_RATES` (page 4)). In these dictionaries the key is an index (e.g. `PresetFrameRates` (page 4)) and the value a `namedtuple`⁷ (e.g. `FrameRate` (page 4)) giving the values for that row in the table.

As an example, the snippet below looks up the frame size of the Digital Cinema 2K base video format:

```
>>> from vc2_data_tables import BaseVideoFormats, BASE_VIDEO_FORMAT_PARAMETERS

>>> index = BaseVideoFormats.dc2k
>>> index
<BaseVideoFormats.dc2k: 15>

>>> # The named enumerated types are optional and completely interchangeable
>>> # with normal integer values, e.g.
>>> index == 15
True

>>> # Data tables are usually looked up by index (either integers or named
>>> # values may be used)
>>> params = BASE_VIDEO_FORMAT_PARAMETERS[index]
>>> params.frame_width
2048
>>> params.frame_height
1080
```

⁶ <https://docs.python.org/3/library/enum.html#enum.IntEnum>

⁷ <https://docs.python.org/3/library/collections.html#collections.namedtuple>

2.1 (10.5) Parse Info Block

PARSE_INFO_PREFIX = int

(10.5.1) The ‘magic bytes’ used to identify the start of a parse info header.

PARSE_INFO_HEADER_BYTES = int

(10.5.1) The number of bytes in the parse_info header.

class ParseCodes(IntEnum)

(10.5.2) Valid parse_code values from (Table 10.1). Names are not normative.

2.2 (11) Picture coding mode identifiers

class PictureCodingModes(IntEnum)

(11.5) Indices defined in the text. Names are not normative.

class ColorDifferenceSamplingFormats(IntEnum)

(11.4.4) Indices from (Table 11.2)

class SourceSamplingModes(IntEnum)

(11.4.5) Indices defined in the text. Names are not normative.

2.3 (11.4.6) Preset frame rates

PRESET_FRAME_RATES = {PresetFrameRates: FrameRate, ...}

(11.4.6) Frame-rate presets from (Table 11.3). Lookup from *PresetFrameRates* (page 4) to *FrameRate* (page 4) tuples.

class PresetFrameRates(IntEnum)

(11.4.6) Preset framerate indices from (Table 11.1).

class FrameRate(enumerator, denominator)

(11.4.6) A frame rate numerator and denominator value from (Table 11.1).

Parameters

numerator [int]

denominator [int]

2.4 (11.4.7) Preset pixel aspect ratios

PRESET_PIXEL_ASPECT_RATIOS = {PresetPixelAspectRatios: PixelAspectRatio, ...}

(11.4.7) Pixel aspect ratio presets from (Table 11.4). Lookup from *PresetPixelAspectRatios* (page 4) to *PixelAspectRatio* (page 4)

class PresetPixelAspectRatios(IntEnum)

(11.4.7) Pixel aspect ratio preset indices from (Table 11.4).

class PixelAspectRatio(enumerator, denominator)

(11.4.7) Pixel aspect ratio preset indices from (Table 11.4).

2.5 (11.4.9) Signal ranges

PRESET_SIGNAL_RANGES = {**PresetSignalRanges**: **SignalRangeParameters**, ...}

(11.4.9) Signal offsets/ranges presets from (Table 11.5). Lookup from [PresetSignalRanges](#) (page 5) to [SignalRangeParameters](#) (page 5).

class PresetSignalRanges(*IntEnum*)

(11.4.9) Signal offsets/ranges preset indices from (Table 11.5).

class SignalRangeParameters(*luma_offset, luma_excursion, color_diff_offset, color_diff_excursion*)

An entry in (Table 11.5).

Parameters

luma_offset The luma value corresponding with 0.

luma_excursion The maximum value of an offset luma value.

color_diff_offset The color difference value corresponding with 0.

color_diff_excursion The maximum value of an offset color difference value.

2.6 (11.4.10.2) Color Primaries

PRESET_COLOR_PRIMARIES = {**PresetColorPrimaries**: **ColorPrimariesParameters**, ...}

(11.4.10.2) Normative specification names for color primaries from (Table 11.7). Lookup from [PresetColorPrimaries](#) (page 5) to [ColorPrimariesParameters](#) (page 5).

class PresetColorPrimaries(*IntEnum*)

(11.4.10.2) Color primaries from (Table 11.7).

class ColorPrimariesParameters(*name, specification*)

(11.4.10.2) A color primaries description.

Parameters

name Informative name.

specification [str] The name of the specification defining the primaries in use.

2.7 (11.4.10.3) Color Matrices

PRESET_COLOR_MATRICES = {**PresetColorMatrices**: **PresetColorMatrices**, ...}

(11.4.10.3) Color matrices from (Table 11.8). Lookup from [PresetColorMatrices](#) (page 5) to [ColorMatrixParameters](#) (page 5).

class PresetColorMatrices(*IntEnum*)

(11.4.10.3) Color matrices from (Table 11.8).

class ColorMatrixParameters(*name, specification, color_matrix*)

An entry in (Table 11.8)

Parameters

name Informative name.

specification Normative specification name.

color_matrix Normative color matrix description.

2.8 (11.4.10.4) Transfer functions

PRESET_TRANSFER_FUNCTIONS = {**PresetTransferFunctions**: **TransferFunctionParameters**, ...}

(11.4.10.3) Color matrices from (Table 11.8). Lookup from *PresetTransferFunctions* (page 6) to *TransferFunctionParameters* (page 6).

class PresetTransferFunctions(*IntEnum*)

(11.4.10.4) Transfer functions from (Table 11.9).

class TransferFunctionParameters(*name, specification*)

An entry in (Table 11.9)

Parameters

name Informative name.

specification Normative specification name.

2.9 (11.4.10.1) Colour specifications

PRESET_COLOR_SPECS = {**PresetColorSpecs**: **ColorSpecficiation**, ...}

(11.4.10.3) Color matrices from (Table 11.8). Lookup from *PresetColorSpecs* (page 6) to *ColorSpecficiation* (page 6).

class PresetColorSpecs(*IntEnum*)

(11.4.10.1) Preset color specification collections from (Table 11.6).

class ColorSpecficiation(*color_primaries_index, color_matrix_index, transfer_function_index*)

An entry in (Table 11.6)

Parameters

color_primaries A *PresetColorPrimaries* (page 5) index.

color_matrix A *PresetColorMatrices* (page 5) index.

transfer_function A *PresetTransferFunctions* (page 6) index.

2.10 (11.3) Base Video Formats

BASE_VIDEO_FORMAT_PARAMETERS = {**BaseVideoFormats**: **BaseVideoFormatParameters**, ...}

(B) Base video format specifications from (Table B.1a, B.1b, B.1c). Lookup from *BaseVideoFormats* (page 6) to *BaseVideoFormatParameters* (page 6).

class BaseVideoFormats(*IntEnum*)

(11.3) Base video format indices from (Table 11.1).

class BaseVideoFormatParameters(*frame_width, frame_height, color_diff_format_index, source_sampling, top_field_first, frame_rate_index, pixel_aspect_ratio_index, clean_width, clean_height, left_offset, top_offset, signal_range_index, color_spec_index*)

(B) An entry in (Table B.1a, B.1b or B.1c)

Parameters

frame_width

frame_height

color_diff_format_index An entry from the enum *ColorDifferenceSamplingFormats* (page 4). Listed as ‘color difference sampling format’ in (Table B.1).

source_sampling An entry from the enum *SourceSamplingModes* (page 4). Specifies progressive or interlaced.

top_field_first If True, the top-line of the frame is in the first field.

frame_rate_index The frame rate, one of the indices of PRESET_FRAME_RATES.

pixel_aspect_ratio_index The pixel aspect ratio, an entry from the enum *PresetPixelAspectRatios* (page 4).

clean_width

clean_height

left_offset

top_offset The clean area of the pictures. See (11.4.8) and (E.4.2).

signal_range_index The signal ranges, an entry from the enum *PresetSignalRanges* (page 5).

color_spec_index The color specification, an entry from the enum *PresetColorSpecs* (page 6).

2.11 (15.4.4) Lifting filters

LIFTING_FILTERS = {WaveletFilters: LiftingFilterParameters, ...}

(15.4.4.3) Filter definitions taken from (Table 15.1 to 15.6). Lookup from *WaveletFilters* (page 7) to *LiftingFilterParameters* (page 7).

class WaveletFilters(IntEnum)

(12.4.2) Wavelet filter type indices from (Table 12.1). Names are based on the informative names in the table.

See also: *LIFTING_FILTERS* (page 7).

class LiftingFilterParameters(filter_bit_shift, stages)

(15.4.4.3) The generic container for the details described by (Table 15.1 to 15.6).

Parameters

filter_bit_shift Right-shift to apply after synthesis (or before analysis).

stages A list of LiftingStage objects to be used in sequence to perform synthesis with this filter.

class LiftingStage(lift_type, S, L, D, taps)

(15.4.4.1) Definition of a lifting stage/operation in a lifting filter.

Parameters

lift_type Specifies which lifting filtering operation is taking place. One of the indices from the LiftingFilterTypes enumeration.

S Scale factor (right-shift applied to weighted sum)

L Length of filter.

D Offset of filter.

taps An array of integers defining the filter coefficients.

class LiftingFilterTypes(IntEnum)

(15.4.4.1) Indices of lifting filter step types. Names are informative and based on an interpretation of the pseudo-code in the specification.

2.12 (C.2) Profiles

PROFILES = {Profiles: ProfileParameters, ...}

The list of supported profiles from (C.2). Lookup from *Profiles* (page 8) to *ProfileParameters* (page 8).

class Profiles(IntEnum)

(C.2) VC-2 profile identifiers.

class ProfileParameters(allowed_parse_codes)

(C.2) Parameters describing a profile specification.

Parameters

allowed_parse_codes A list of supported data units. A list of values from the ParseCodes enum.

2.13 (D) Quantisation matrices

QUANTISATION_MATRICES = {(wavelet_index, wavelet_index_ho, dwt_depth, dwt_depth_ho): quantisation_matrix, ...}

The preset quantisation matrices from (Table D.1) to (Table D.8)

The loaded matrices are stored in a nested dictionary with the following layout:

```
QUANTISATION_MATRICES[(wavelet_index, wavelet_index_ho, dwt_depth, dwt_depth_
↪ho)][level][orientation]
```

Where:

- wavelet_index and wavelet_index_ho are *WaveletFilters* (page 7) values
- dwt_depth and dwt_depth_ho are transform depths (integers)
- level is the transform level (integer)
- orientation is one of “L”, “H”, “LL”, “HL”, “LH” or “HH”

Warning: The values in these tables correspond to those published in SMPTE ST 2042-1:2017. Consequently, errors in the quantisation matrices for the ‘Fidelity’ filter are repeated here.

2.14 (ST 2042-2) Levels

LEVELS = {Levels: LevelParameters, ...}

The list of supported levels from (ST 2042-2:2017: 5.2). A lookup from *Levels* (page 8) to *LevelParameters* (page 8).

class Levels(IntEnum)

(ST 2042-2:2017: 5.2) VC-2 level identifiers.

class LevelParameters(standard)

(ST 2042-2) Parameters describing a level.

Parameters

standard [str] Name of the standards document which defines the level.

CSV FILES AND CONVENTIONS

Many of the data tables exposed by this Python module are read from CSV files contained in `vc2_data_tables/` `csv/*`. `csv`. You are free to use these files directly rather than via the `vc2_data_tables` (page 3) module.

In general, the CSV files mimic the format which appears in the VC-2 standards documents and can be opened in common spreadsheet packages and are intended to be relatively human-readable.

Tip: After opening the CSV files included with this module with a spreadsheet package it is often helpful to reset the width of all columns to a fixed size. This is because many files will contain comments which cause certain columns to be assigned very large sizes (see section about comments below).

Warning: Take care when modifying these CSV files some spreadsheet packages contain unhelpful ‘smart’ features which can mangle cells containing values matching certain formats (e.g. anything which looks like a date). Also watch out for trailing spaces in cells.

The sections below define the specific conventions used by the CSV files in this package.

3.1 CSV Dialect

The CSV provided use the Microsoft Excel CSV dialect and are encoded using UTF-8. See Python’s `csv`⁸ module for more background on CSV formatting dialects.

3.2 Headings

In all CSV files, the first non-empty/comment row contains headings for the values beneath. The meanings of each column should be self-explanatory, but further hints may be given in the `vc2_data_tables` (page 3) API documentation associated with each table.

Note: The `vc2_data_tables` (page 3) software expects certain column names to be used but is insensitive to column order and will ignore any unexpected extra columns.

⁸ <https://docs.python.org/3/library/csv.html#module-csv>

3.3 Comments

To aid human consumption of the CSV data files, human-readable comments are used. All rows in a CSV file containing only empty cells or cells whose values start with a hash (#) character should be considered to be comments and omitted.

The following illustrates how comments may be used in a CSV.

Not	a	comment!
	# A comment	# Another comment
Not	a	comment!
123	abc	# <i>Not</i> a comment

Warning: A cell starting with a hash (#) on a row which contains data is *not* a comment (see final row of the example above). Comments may only occur on rows which otherwise do not contain data.

3.4 Ditto

In some tables, values from previous rows may be repeated using ‘ditto’, written as " or ' '. For example:

Number	Above 3?
1	No
2	“
3	“
4	Yes
5	“
6	“

Warning: Excel and other spreadsheet packages often replace ‘straight’ quote characters with their ‘curly’ unicode variants. Take care when manually parsing CSV files containing ditto to also handle these variants.

3.5 Lists

Occasionally table cells may contain a comma separated list of values. These commas are escaped according to the Excel CSV convention of enclosing the whole cell value in double quotes.

B

BASE_VIDEO_FORMAT_PARAMETERS (in module *vc2_data_tables*), 6
 BaseVideoFormatParameters (class in *vc2_data_tables*), 6
 BaseVideoFormats (class in *vc2_data_tables*), 6

C

ColorDifferenceSamplingFormats (class in *vc2_data_tables*), 4
 ColorMatrixParameters (class in *vc2_data_tables*), 5
 ColorPrimariesParameters (class in *vc2_data_tables*), 5
 ColorSpecification (class in *vc2_data_tables*), 6

F

FrameRate (class in *vc2_data_tables*), 4

L

LevelParameters (class in *vc2_data_tables*), 8
 Levels (class in *vc2_data_tables*), 8
 LEVELS (in module *vc2_data_tables*), 8
 LIFTING_FILTERS (in module *vc2_data_tables*), 7
 LiftingFilterParameters (class in *vc2_data_tables*), 7
 LiftingFilterTypes (class in *vc2_data_tables*), 7
 LiftingStage (class in *vc2_data_tables*), 7

M

module
 vc2_data_tables, 3

P

PARSE_INFO_HEADER_BYTES (in module *vc2_data_tables*), 4
 PARSE_INFO_PREFIX (in module *vc2_data_tables*), 4
 ParseCodes (class in *vc2_data_tables*), 4
 PictureCodingModes (class in *vc2_data_tables*), 4
 PixelAspectRatio (class in *vc2_data_tables*), 4
 PRESET_COLOR_MATRICES (in module *vc2_data_tables*), 5
 PRESET_COLOR_PRIMARIES (in module *vc2_data_tables*), 5
 PRESET_COLOR_SPECS (in module *vc2_data_tables*), 6
 PRESET_FRAME_RATES (in module *vc2_data_tables*), 4
 PRESET_PIXEL_ASPECT_RATIOS (in module *vc2_data_tables*), 4
 PRESET_SIGNAL_RANGES (in module *vc2_data_tables*), 5
 PRESET_TRANSFER_FUNCTIONS (in module *vc2_data_tables*), 6
 PresetColorMatrices (class in *vc2_data_tables*), 5
 PresetColorPrimaries (class in *vc2_data_tables*), 5
 PresetColorSpecs (class in *vc2_data_tables*), 6
 PresetFrameRates (class in *vc2_data_tables*), 4
 PresetPixelAspectRatios (class in *vc2_data_tables*), 4
 PresetSignalRanges (class in *vc2_data_tables*), 5
 PresetTransferFunctions (class in *vc2_data_tables*), 6
 ProfileParameters (class in *vc2_data_tables*), 8
 Profiles (class in *vc2_data_tables*), 8
 PROFILES (in module *vc2_data_tables*), 8

Q

QUANTISATION_MATRICES (in module *vc2_data_tables*), 8

S

SignalRangeParameters (class in *vc2_data_tables*), 5
 SourceSamplingModes (class in *vc2_data_tables*), 4

T

TransferFunctionParameters (class in *vc2_data_tables*), 6

V

vc2_data_tables
 module, 3

W

WaveletFilters (class in *vc2_data_tables*), 7