



XML UTILITY FORMAT

SP-08698-001_v01 | August 2017

Specification



DOCUMENT CHANGE HISTORY

SP-08698-001_v01

Version	Date	Description of Change
01	8/04/2017	Initial release

TABLE OF CONTENTS

Chapter 1. Introduction..... 1

Chapter 2. Rig Descriptor XML 1

Chapter 3. Audio Rig Descriptor XML..... 1

Chapter 4. Input Media Feed XML 1

Chapter 5. Input Audio Feed XML 1

Chapter 6. Stitcher Properties XML 4

Chapter 1. INTRODUCTION

The document specifies the format for XMLs used by the XML utility. The XML utility provides reading and writing of XMLs from this XML format and populates commonly needed structs in the stitching API calls for ease of use.

The following XMLs are defined:

- ▶ Rig Descriptor XML
- ▶ Audio Rig Descriptor XML
- ▶ Input Feed XML
- ▶ Audio Input Feed XML
- ▶ Stitcher Properties XML

A combination of the following are needed based on the application and options:

- ▶ **Calibration:** Inputs Rig Descriptor XML and Outputs Calibrated Rig Descriptor XML
- ▶ **Video Stitching:** Input Calibrated Rig Descriptor XML, Input Feed XML, and Input Stitcher Properties XML
- ▶ **Video + Audio Stitching:** All the XMLs

Chapter 2. RIG DESCRIPTOR XML

The Rig Descriptor XML contains the camera rig properties and all individual camera properties. This XML is used to provide input estimates for camera calibration, to save the calibrated properties from camera calibration, and to input rig properties for video stitching. It populates the `nvstitchVideoRigProperties_t` struct of `nvstitch` API.

Here is the layout with various options to specify:

```

<!-- RIG XML SPEC -->

<?xml version="1.0" encoding="utf-8"?>

<!-- CAMERA RIG DESCRIPTOR -->
<camera_rig coord_axes="y-up|z-up|y-down" rig_diameter_cm="" >

  <camera width="" height="" layout="equatorial|general" >
    <pose>
      <rotation yaw_deg="" pitch_deg="" roll_deg=""/>
      or <rotation yaw_rad="" pitch_rad="" roll_rad=""/>
      or <rotation m0="" m1="" m2="" m3="" m4="" m5="" m6="" m7="" m8="" />

      <translation x_cm="" y_cm="" z_cm="" >
    </pose>

    <optics>
      <focal_length focal_pixels="" />
      or <focal_length focal_x_pixels="" focal_y_pixels=""/>
      or <focal_length hfov_deg="" vfov_deg=""/>

      <principal_point center_offset_x="" center_offset_y=""/>
      or <principal_point top_left_offset_x="" top_left_offset_y=""/>

      <lens type="brown" r1="" r2="" t1="" t2="" r3=""/>
      or <lens type="brown" k1="" k2="" k3="" k4="" k5=""/>
      or <lens type="fisheye" k1="" k2="" k3="" k4=""/>
      or <lens type="fisheye" r1="" r2="" r3="" r4=""/>

      <fisheye_radius radius_pixels="" />
    </optics>

    <input_calib_file name=".jpg|.bmp|.png"/> <!-- optional -->
  </camera>

  <camera width="" height="" layout="equatorial|general">
    ...
  </camera>

</camera_rig>

```

Note: Blank attributes imply numbers.

<camera_rig>
coord_axes

Axes in which the camera pose are specified. Default is y-up.

Refer to the API guide for the description of the axes types.

Specify either y-up, or z-up, or y-down

rig_diameter_cm

Rough estimate of the rig diameter in centimeters. Required for stereo-stitching for equatorial cameras.

<camera>**width**

Camera image width in pixels.

Type: Integer

height

Camera image height in pixels.

Type: Integer

layout

Layout of camera, specifies the position of camera in rig, specify equatorial

For cameras in ring along the equator, else specify general. Stereo-stitching uses only equatorial camera.

<pose>

Specify the pose of the camera as rotation and translation with respect to the rig.

<rotation>

Rotation pose of the camera.

yaw_deg, pitch_deg, roll_deg

The Yaw, pitch and roll in degrees of the camera.

Type: float

yaw_rad, pitch_rad, roll_rad

The Yaw, pitch and roll in radians of the camera.

Type: float

m0, m1, m2 .. m8

The pose matrix of the camera in the coordinate axes specified.

Type: float

<translation>**x_cm, y_cm, z_cm**

The X,Y and Z translation offset of the camera from the rig center in the cord_axes specified. Used for camera calibration with translation option, specify 0,0,0 as an input estimate for calibration if unavailable.

<optics>**<focal_length>**

Specify either focal_pixels or focal_x_pixels, focal_y_pixels

Or hfov_deg="" vfov_deg=""

focal_pixels

Focal length of camera in pixels.

Type: float

focal_x_pixels, focal_y_pixels

Focal length and X and Y direction of camera in pixels

Type: float

hfov_deg, vfov_deg

Horizontal and vertical Field of View of the Camera specified in degrees.

Type: Float

<principal_point>

center_offset_x, center_offset_y

The principal point specified as an offset from the center in pixels from the image center.

Offset X is positive in the right direction.

Offset Y is positive in the down direction.

If performing calibration this can be set to 0,0 as in input estimate.

Type: Float

top_left_offset_x, top_left_offset_y

The principal point specified as an offset from the top left corner of the image in pixels,

Offset X is positive in the right direction.

Offset Y is positive in the down direction.

If performing calibration this can be set to (width-1)/2, (height-1)/2 as an input estimate.

Type: Float

<lens>

type="brown"

Specifies brown lens type of the camera, use this for normal angle lenses, with perspective distortion.

r_1, r_2, t_1, t_2, r_3

r_1, r_2, r_3 - Radial distortion coefficients.

t_1, t_2 – Tangential distortion coefficients.

Specify 0,0,0,0,0 as estimates to calibration.

Type: Float

OR

k_1, k_2, k_3, k_4, k_5

3 Radial coefficients k_1, k_2, k_3 , followed by k_4, k_5 tangential distortion coefficients.

type="fisheye"

Specifies fisheye lens type of the camera, use this for wide angle fisheye lens.

k_1, k_2, k_3, k_4, k_5 OR

r_1, r_2, r_3, r_4, r_5

Radial distortion coefficients for fisheye.

If un-available specify 0,0,0,0,0 as input estimates to calibration.

Type: Float

<fisheye_radius>

radius_pixels

Specifies the radius of valid pixels in the image for fisheye lens cameras.

Optional parameter, specify hypo/2 for normal angle lens or fisheye lens with image with all valid pixels, where hypo is diagonal pixel length of camera image.

This parameter is used for calibration quality evaluation and for stereo-stitching.

Type: Float

<input_calib_file>

name

Specifies the input images to be used for calibration.

Optional if calibration is not performed.

Specify name of input image file from camera with extension.

For providing multiple input image frames from camera specify all filenames separated by space.

Supported image type: .jpg, .png and .bmp

Type: string

Example:

```
<input_calib_file name="frame1.png, frame2.png, frame3.png" />
```

Chapter 3. AUDIO RIG DESCRIPTOR XML

The Audio Rig Descriptor XML is used for specifying the properties of the audio sources in the rig and is used for stitching with audio.

Here is the layout with various options to specify:

```
<!-- AUDIO RIG DESCRIPTOR -->

<?xml version="1.0" encoding="utf-8"?>

<audio_rig coord_axes="y-up|z-up|y-down" rig_diameter_cm="" >
  <audio_source>
    <pose>
      <rotation yaw_deg="" pitch_deg="" roll_deg=""/>
      or <rotation yaw_rad="" pitch_rad="" roll_rad=""/>
      or <rotation m0="" m1="" m2="" m3="" m4="" m5="" m6="" m7="" m8="" />

      <translation x_cm="" y_cm="" z_cm="" >
    </pose>

    <input_type type="omni|shotgun|cardioid|supercardioid"/>
  </audio_source>

  <audio_source>
    .
    .
    .
  </audio_source>
</audioRig>
```

Note: Blank attributes imply numbers.

<input_type>

type

Specify the input type of the audio source.

Options are: **omni** | **shotgun** | **cardioid** | **supercardioid**

Type: String

All attributes and nodes are similar to the Rig Descriptor and describe the audio sources instead of cameras.

Chapter 4. INPUT MEDIA FEED XML

The Input Media Feed XML specifies the video input payload form and properties for video stitching. This XML is needed while performing video stitching.

Here is the layout with various options to specify:

```
<!-- INPUT PAYLOAD XML SPEC -->

<?xml version="1.0" encoding="utf-8"?>

<!-- LIST OF ALL MEDIA SOURCE PAYLOAD -->
<camera_rig>

  <camera>
    <input_media_feed>

      <input_media_form form="file|host_buffer|device_buffer" />

      <!-- Use only if form is file -->
      <input_media_file name=".mp4"
        fps="" or fps_num="" fps_den=""
        width="" height="" />

    </input_media_feed>
  </camera>

  <camera>
    ...
  </camera>
</camera_rig>
```

<input_media_feed>

Specifies the media feed for each media source

<input_media_form>**form**

Specifies the form of input.

Options: file|host_buffer|device_buffer

Type: string

<input_media_file>

Specifies properties of input files, only used if form type is file.

name

Name of media file with extension

Example: media_file.mp4

Type: String

fps, fps_num or fps_den

Frame per second of the input media.

For fps, specify either fps as a floating point number, fraction of two number separated by /

Else specify fps_num or fps_den as two integers.

Example:

fps="30.5"

fps="100/3"

fps_num="100" fps_den="3"

width, height

Width and height of the media file.

Type: Integer

Chapter 5. INPUT AUDIO FEED XML

The Input Audio Feed XML specifies the form and properties of the input payload for each audio source. This XML is need for stitching with audio.

Here is the layout with various options to specify:

```
<!-- INPUT AUDIO FEED XML -->

<?xml version="1.0" encoding="utf-8"?>

<audio_rig>

  <audio_source>
    <input_audio_feed>
      <input_audio_form form="file|buffer" />
      <input_audio_file name=".wav"/> <!-- Used only for file -->
      <input_audio_feed_config format="auto|pcm16lsb|pc32float|aac"
                              channels="" sample_rate="" bit_rate=""
                              samples_per_frame="" />
    </input_audio_feed>
  </audio_source>

  <audio_source>
    ...
  </audio_source>
</audio_rig>
```

<input_audio_feed>

Node for the input feed for audio stitching.

<input_audio_form>**form**

The form of the input audio.

Options: file|buffer

<input_audio_file>

Used only if audio form is file.

name

Name of the input audio file with extension.

Example: audio_file.wav

Type: String

<input_audio_feed_config>

Specifies the configuration data of the input payload for audio.

format

Format/ arrangement of the payload data of audio

Options: auto|pcm16lsb|pc32float|aac

channels

Channels in the audio payload source.

Type: Integer

Example: channels="2"

sample_rate

The sample rate of the audio input payload.

Type: Float

bit_rate

the bit rate of the input audio payload.

Type: Float

samples_per_frame

Sample per frame of the input audio payload.

Type: float

Chapter 6. STITCHER PROPERTIES XML

The Stitcher Properties XML is used for specifying the options and properties for video and audio stitching.

```
<!-- STITCHER PROPERTIES XML SPEC -->
<?xml version="1.0" encoding="utf-8"?>

<stitcher_properties>

    <!-- INPUT MEDIA FEED PROPERTIES -->
    <input_media_form form="file|host_buffer|device_buffer"/>
    <input_media_format format="rgb8ui|yuv420|h264|mp4"/>

    <!-- OUTPUT VIDEO PROPERTIES -->
    <output_video_properties>
        <output_video_projection type="equirectangular|cubemap" />
        <output_video_options stereo_ipd="" quality="low|medium|high"
                               pipeline="mono|stereo"/>
        <output_video_format format="rgb8ui|yuv420|h264|mp4" />
        <output_video_payloads count="1|2"/>
        <output_video_form form="file|host_buffer|device_buffer" />
        <output_video_file name=""
                           fps = "" or fps_num="" fps_den=""
                           width="" height="" /> <!-- Only for files-->
    </output_video_properties>

    <!-- OUTPUT AUDIO PROPERTIES -->
    <output_audio_properties>
        <output_audio_gain value="" />
        <output_audio_blend type="stereo_mixdown" />
        <output_audio_form form="file|host_buffer" />
        <output_audio_feed_config format="pcm16lsb|pc32float|aac"
                                  channels="" sample_rate=""
                                  bit_rate="" samples_per_frame="" />
    </output_audio_properties>
</stitcher_properties>
```

Properties populate the `nvstitchStitcherProperties_t` , all options match to options specified in API enums.

Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication of otherwise under any patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all other information previously supplied. NVIDIA Corporation products are not authorized as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

Trademarks

NVIDIA and the NVIDIA logo are trademarks or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2017 NVIDIA Corporation. All rights reserved.