Aximmetry Pin Data Types

Compound

🟨 Video

A rendered video frame or image.
A frame can be any size up to 8K. Video source modules usually enable to set an arbitrary output size, but if it's unspecified then the default system frame size is used that can be set in Preferences / Rendering.
Image and video player modules, of couse, always output a frame size corresponding to the source file.
A video frame also can have various pixel formats. When frame is coming from an image file, its format is corresponding to the file format. Other video source modules usually enable choosing between Low and High Dynamics Range (HDR) output, which are 8 bits per channel RGBA (LDR) and 16/32 bits per channel floating point RGBA formats (HDR).
HDR formats are 16 bits per channel by default. That can be changed in Preferences / Rendering.
Video also can be a special texture: texture array, cube map or volume texture that can be fed into specific shaders.
When a video postprocessing modules combines two or more inputs into a single output (blending, crossfading, masking etc.) it usually sets the output size to the largest frame size and the widest pixel format of the inputs.
A video frame can be combined with a depth map, usually when outputted from a **Camera**. There are specific modules that use depth map information for various purposes. In general, all postprocessing modules that don't distort or resize the image simply passes thru the depth map without modification.

🟪 Audio

Audio data, represented by a per-frame set of audio samples.
Audio can have an arbitrary number of channels. Audio input modules usually outputs two stereo channels, but for e.g. a video input from an SDI gives 16 channels.
Audio postprocessing modules usually deals with a single channel only, but a series of modules deals with splitting/merging/downmixing channels.
Audio sampling frequency is constant 48kHz according to the frequency used in video systems.

⬛ Scalar

A 32 bit floating point numeric value.

🟩 Double Precision Scalar

A 64 bit floating point numeric value.

🟩 Vector

An arbitrary number of 32 bit floating point numeric values.

It either can represent a scalar (1D vector), a 2D, 3D, 4D etc. vector, or an arbitrary length data array.
In the case it is connected to a scalar or integer pin, only the first element is used. In the opposite direction, when a scalar or integer value is connected to a vector pin, it treated as a one-element vector.

⬜ Integer

A 32 bit integer numeric value.

🟥 Logical

An ON/OFF switch value.
In the case it is connected to a numeric pin, it appears as 0 or 1 value. In the opposite direction, when a numeric value is connected to a logical pin, 0 treated as OFF, all other values as ON.

🟪 Trigger

A signal pulse value outputted by various modules to signal a specific event (for e.g. reaching the end of something) that can be used to start/stop various animations or timing functions.
Essentially it’s a Logical value where OFF to ON transitions represents the moment of signaling. The ON state lasts for a single frame, then drops back to OFF.
Therefore logical operators like And, Or can be used for triggers as well. For example And-ing two triggers means waiting for a simultaneous occuring of two different signals.

🟦 Color

A 32 bits per channel floating point RGBA value.
Normal interval for the channels is 0 - 1. Values above 1 are used for HDR imaging.
In the case it is connected to a vector pin, the value is converted into a four-element vector.
In the opposite direction, when a vector value is connected to a color pin, the first four element is used. If the vector is three-element only, the alpha is set to 1. If the vector is two-element only, blue becomes 0. When a scalar value is connected, it becomes a gray level.
When a color value is edited in the Pin Values editor, it appears as two components: a usual 8 bits (0-255) per channel RGBA value (the LDR part), and a 32 bit floating point HDR factor acting as an intensity multiplier.

🟦 Text

A unicode text string.

🟩 Transformation

Represents a spatial position-rotation-scaling which can be applied to various scene objects.
In special cases it can represent homogeneous space transformations described by a 4x4 matrix, for e.g. a camera’s perspective transformation.

🟧 Collection

Stores an arbitrary number of named data elements. Each data element can be a scalar, a double, an integer, a logical, or a text, or it can be an entire subcollection thus creating a

hierarchical data structure. Elements can be accessed by their name, or a period-separated path in the case of a hierarchy.
Collections cannot be edited directly, instead they can be imported from an XML or a JSON file/text.

Configurator

Only used when creating Performer patches to enable a special configuration interfaces for a module which supports it. If a configurator pin is exposed on the root level of the compound, a corresponding configurator panel appears in Performer when the patch is selected.

Reference Connections

These pin/connection types does not represent a flow of any kind of data, instead designate specific relationships between modules.

Sequence Track Reference

Connects a **Sequencer** to its various track modules thus creating tracks in the Sequence Editor.

Video Coplayer Reference

Video Player module can co-operate with special coplayer modules that extract extra per-frame informations from special video files, for e.g. depth map.

Scene Object Child Reference

Designates a parent-child relationship between two scene object modules. Child's transformation is relative to its parent's transformation.

Scene Object Binding Reference

Ties a scene object's transformation of part of its transformation to another scene object. Target object's **Binding Type** value selects the nature of the binding. Most common application is a look-at relationship between a **Camera** and an object.

Affecting Light Reference

Designates which light sources have effect on a scene node and all of its children.

Light Volume Reference

Connects a **Light Volume** module to the light sources for which the volumetric effect have to be rendered.

Graph Node Route Reference

Connects **Graph Node** modules to each other representing an edge of the graph.

Transmitter Reference

Transmitter modules can be connected to a **Scene Node** designating that only that node and its children can see that specific tunnel.

⬜ Paint Object Child Reference

Designates a parent-child relationship between two paint object modules. Child’s transformation is relative to its parent’s transformation.

⬜ Path Element Reference

Used to chain the graphic elements of a **Paint Path** module.

⬜ Touch Reference

Connects a **Touch Receiver** to the client modules that use its touch inputs.

⬜ Recorder Reference

Connects an **Input Recorder** to the module from which data is to be recorded.

Shader

🟨 Texture

🟨 Texture Array

🟧 Cube Texture

🟥 Volume Texture

Video data connected to a shader appears as a texture inside the shader.
On the compound level there are no deferrentiated pin types for these texture kinds, but shader design must be very specific on that matter.

⬛ Scalar

A 32 bit floating point numeric value.

🟩 Vector

A four-element XYZW vector consisting of 32 bit floating point numeric values.
It either can represent a scalar (all components are equal), or a 4D vector. Certain modules only use the 2D or 3D slice of the vector.
In the case it is connected to a scalar or integer pin, only the first element is used. In the opposite direction, when a scalar or integer value is connected to a vector pin, it treated as an all-equal vector.

⬜ Integer

A 32 bit signed integer numeric value.

⬜ Unsigned Integer

A 32 bit unsigned integer numeric value.

🟥 Logical

An ON/OFF switch value.

🟦 Color

A 32 bits per channel floating point RGB value. (Alpha is not included, it is handled separately in shaders.)
Normal interval for the channels is 0 - 1. Values above 1 are used for HDR imaging.
When a color value is edited in the Pin Values editor, it appears as two components: a usual 8 bits (0-255) per channel RGBA value (the LDR part), and a 32 bit floating point HDR factor acting as an intensity multiplier.

🟩 Transformation

Represents a spatial position-rotation-scaling described by a 4x3 matrix. It can be used to transform spatial position or direction vectors.

🟩 Homogeneous Space Transformation

Represents a homogeneous space transformation described by a 4x4 matrix, for e.g. a camera’s perspective transformation.

🟫 Sampler State

Assigns a sampler state definition to a sampler module.

🟫 Rasterizer State

Assigns a rasterizer state definition to a pixel shader output module.
Be careful, because it overrides the system’s automatic rasterizer state selection based on shader parameters.

🟫 Depth Stencil State

Assigns a depth-stencil state definition to a pixel shader output module.
Be careful, because it overrides the system’s automatic depth-stencil state selection based on shader parameters.

🟫 Blend State

Assigns a blend state definition to a pixel shader output module.
Be careful, because it overrides the system’s automatic blend state selection based on shader parameters.