

<p>DATE : Sept 09, 2019</p> <p>Doc : PosiStageNet Protocol description v2.03</p>			<p>VYV Corporation 5550 Fullum suite 342 Montréal, Québec, Canada H2G 2H4 514.373.2364</p> <p>MA Lighting Technology GmbH Dachdeckerstr. 16 97297 Waldbüttelbrunn Germany Phone: +49 (0) 931 49794 -0</p>
---	---	--	---

PosiStageNet Protocol description v2.0

© Copyright by VYV Corporation and MA Lighting Technology GmbH 2014-2019

This document is released into the public domain without warranty of any kind including but not limited to the implied warranties of fitness for a particular purpose. VYV Corporation and MA Lighting International GmbH shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, accuracy or use of this material.

The PosiStageNet protocol and associated documentation are copyright of VYV Corporation and MA Lighting Technology GmbH. **Any third parties are welcome to use this communication protocol without royalty.** Any party is welcome to use this protocol as a basis for their own work, as long as they refrain from using the name PosiStageNet to describe it.

VYV Corporation politely requests that any manufacturer who implements this protocol send details to info@posistage.net so that protocol updates may be communicated. Furthermore, if changes to the protocol are needed, VYV Corporation politely requests that such changes be communicated so that they may be included in the official protocol, rather than create unnecessary redundant standards.

This document defines the PosiStageNet communication specifications. This protocol was developed by VYV Corporation with input from MA Lighting International GmbH, as a means for VYV's Photon server to communicate the position of uniquely identified points in space to MA Lighting systems. Its intent is to provide a means for tracking systems to share tracking data with other show systems through an open, common and royalty-free protocol.

Communication is achieved using **UDP** for efficient and simultaneous access from multiple systems.

The PosiStageNet sends tracking information as calculated by a positioning system; it defines 2 types of packets to transmit tracking information, the **PSN_DATA** packet and the **PSN_INFO** packet. **PSN_DATA** packets are transmitted when the tracking server is active. The default transmission rate is **60Hz**, however this parameter can be modified on the tracking server and transmission could go as fast as the tracking hardware supports it (**250Hz** at the time of this writing). The **PSN_INFO** packet is transmitted at a slower rate as it is not supposed to change as often as positioning information. The default rate of **1Hz** is used for **PSN_INFO** packets.

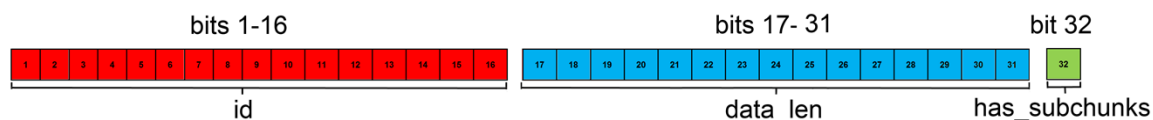
DATE : Sept 09, 2019 Doc : PosiStageNet Protocol description v2.03			VYV Corporation 5550 Fullum suite 342 Montréal, Québec, Canada H2G 2H4 514.373.2364 MA Lighting Technology GmbH Dachdeckerstr. 16 97297 Waldbüttelbrunn Germany Phone: +49 (0) 931 49794 -0
--	---	--	--

The PosiStageNet protocol is transmitted as **UDP multicast** at address **236.10.10.10** over the port **56565**. These are the defaults parameters and can be modified on the tracking server. The protocol defines a maximum size of 1500 bytes for a single UDP packet. This limitation can cause a single frame of data to be split into multiple packets. A client must take care of reconstructing the data using the information provided in the packet header.

The PosiStageNet protocol uses a concept of CHUNKS to organize the data in a packet.

- Chunks recursively define all data sections within one data packet.
- A chunk can either contain other chunks, or binary data that needs interpretation.
- An application that is decoding a data packet and comes across unknown chunks must simply ignore them.

A PSN_CHUNK_HEADER is defined as a 32-bit unsigned integer and is structured like this:



id an enum, defining the meaning of the chunk. This meaning is only valid within the current decoding context which is defined by the parent chunk.

data_len the length of the data that follows the chunk header.

has_subchunks the bit that defines if this chunk contains child chunks.

To obtain maximum compatibility across protocol versions, care must be taken, when existing binary chunk data structures are changed:

- If a structure is expanded, new members shall be added in the end, and default values for these members must be 0. In this way, a new client can receive data from old servers by only partially reading in the data. Old clients can also receive data from newer servers by skipping the part that is too big for them.
- If such a trivial conversion is not possible or not wanted, a new id must be introduced, so old clients can safely skip the new data.

Care must be taken when introducing new ids, as different manufacturers might collide when expanding the protocol. Please write at info@posistage.net for any protocol change request.

DATE : Sept 09, 2019 Doc : PosiStageNet Protocol description v2.03			VYV Corporation 5550 Fullum suite 342 Montréal, Québec, Canada H2G 2H4 514.373.2364 MA Lighting Technology GmbH Dachdeckerstr. 16 97297 Waldbüttelbrunn Germany Phone: +49 (0) 931 49794 -0
--	---	--	--

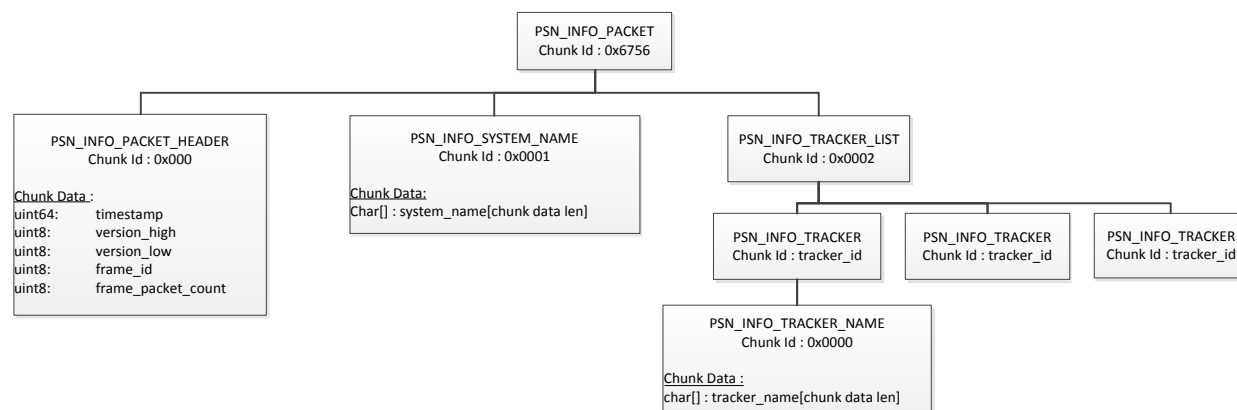
Data Type Definition

Type	Symbol
8-bit unsigned integer	uint8
16-bit unsigned integer	uint16
32-bit unsigned integer	uint32
64-bit unsigned integer *	uint64
32-bit float	float
32-bit float array	float[]
8-bit character	char
8-bit character array	char[]
PSN_CHUNK_HEADER	pch32

* The PosiStageNet protocol is encoded following the Little Endian convention. For integration on architectures that does not support 64-bit types, simply replace all 64-bit unsigned integers by two 32-bit unsigned integers. The first integer should hold the least significant bytes and the second integer the most significant bytes.

PSN_INFO packet scheme

Chunk Header	Chunk Data	Type
{0x6756} PSN_INFO_PACKET		pch32
{0x0000} PSN_INFO_PACKET_HEADER		pch32
	packet_timestamp	uint64
	version_high	uint8
	version_low	uint8
	frame_id	uint8
	frame_packet_count	uint8
{0x0001} PSN_INFO_SYSTEM_NAME		pch32
	system_name[chunk data len]	char[]
{0x0002} PSN_INFO_TRACKER_LIST		pch32
{tracker_id} PSN_INFO_TRACKER		pch32
{0x0000} PSN_INFO_TRACKER_NAME		pch32
	tracker_name[chunk data len]	char[]



DATE : Sept 09, 2019 Doc : PosiStageNet Protocol description v2.03			VYV Corporation 5550 Fullum suite 342 Montréal, Québec, Canada H2G 2H4 514.373.2364 MA Lighting Technology GmbH Dachdeckerstr. 16 97297 Waldbüttelbrunn Germany Phone: +49 (0) 931 49794 -0
--	---	--	--

PSN_INFO_PACKET

This is the root chunk of a PSN_INFO packet.

PSN_INFO_PACKET_HEADER

<i>timestamp</i>	This is the number of microseconds elapsed since the PSN server was started to the moment the packet was sent by the server.
<i>version_high</i>	An 8-bit unsigned integer defining the high version number of the Protocol. This parameter insures that all systems using the same high version number are compatible with the sent binary packet.
<i>version_low</i>	An 8-bit unsigned integer defining the low version number of the Protocol.
<i>frame_id</i>	An 8-bit unsigned integer defining the current frame ID. A frame is a short period of time in which we analyze active trackers and give them a position and a velocity. As the UDP packet size is limited, it may happen that we need to split the information of one frame into multiple packets. The frame ID is then used to merge back the information on the client side.
<i>frame_packet_count</i>	An 8-bit unsigned integer defining how many packets are to be expected for the current frame ID.

PSN_INFO_SYSTEM_NAME

<i>name</i>	An array of characters containing the name of the PSN server.
-------------	---

PSN_INFO_TRACKER_LIST

This chunk is the root for all the PSN_INFO_TRACKER chunks.

PSN_INFO_TRACKER

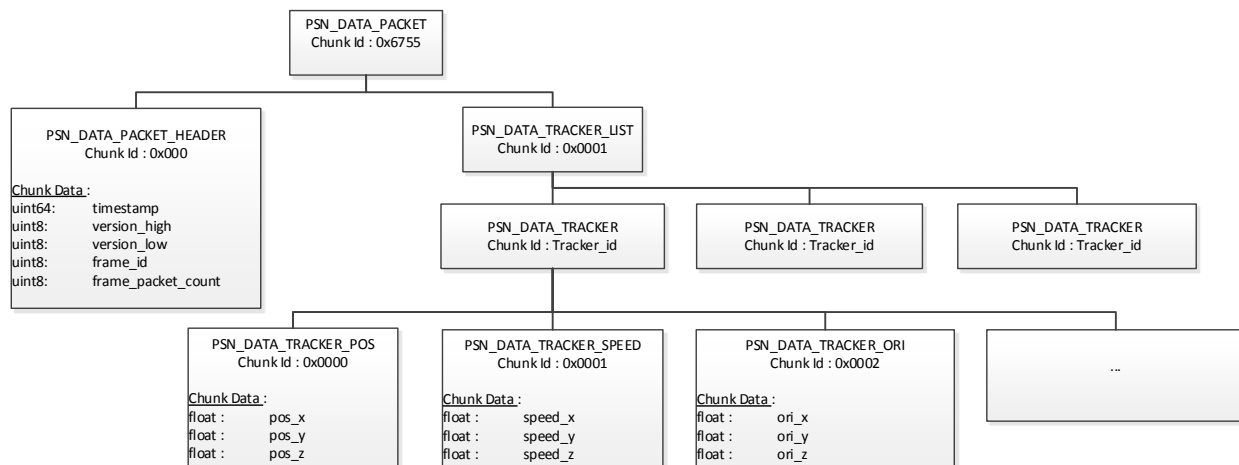
This contains all the information of a specific tracker. The tracker Id is also the chunk Id.

PSN_INFO_TRACKER_NAME

<i>name</i>	An array of characters containing the name of the tracker.
-------------	--

PSN_DATA packet scheme

Chunk Header	Chunk Data	Type
{0x6755} PSN_DATA_PACKET		pch32
{0x0000} PSN_DATA_PACKET_HEADER		pch32
	packet_timestamp	uint64
	version_high	uint8
	version_low	uint8
	frame_id	uint8
	frame_packet_count	uint8
{0x0001} PSN_DATA_TRACKER_LIST		pch32
{tracker_id} PSN_DATA_TRACKER		pch32
{0x0000} PSN_DATA_TRACKER_POS		pch32
	pos_x	float
	pos_y	float
	pos_z	float
{0x0001} PSN_DATA_TRACKER_SPEED		pch32
	speed_x	float
	speed_y	float
	speed_z	float
{0x0002} PSN_DATA_TRACKER_ORI		pch32
	ori_x	float
	ori_y	float
	ori_z	float
{0x0003} PSN_DATA_TRACKER_STATUS		pch32
	validity	float
{0x0004} PSN_DATA_TRACKER_ACCEL		pch32
	accel_x	float
	accel_y	float
	accel_z	float
{0x0005} PSN_DATA_TRACKER_TRGTPOS		pch32
	trgtpos_x	float
	trgtpos_y	float
	trgtpos_z	float
{0x0006} PSN_DATA_TRACKER_TIMESTAMP		pch32
	tracker_timestamp	uint64



PSN_DATA PACKET

This is the root chunk of a PSN_DATA packet.

PSN_DATA PACKET HEADER

timestamp

This is the number of microseconds elapsed since the PSN server was started to the moment the packet was sent by the server.

version_high

An 8-bit unsigned integer defining the high version number of the Protocol. This parameter insures that all systems using the same high version number are compatible with the sent binary packet.

version_low

An 8-bit unsigned integer defining the low version number of the Protocol.

frame_id

An 8-bit unsigned integer defining the current frame ID. A frame is a short period of time in which we analyze active trackers and give them a position and a velocity. As the UDP packet size is limited, it may happen that we need to split the information of one frame into multiple packets. The frame ID is then used to merge back the information on the client side.

frame_packet_count

An 8-bit unsigned integer defining how many packets are to be expected for the current frame ID.

DATE : Sept 09, 2019 Doc : PosiStageNet Protocol description v2.03			VYV Corporation 5550 Fullum suite 342 Montréal, Québec, Canada H2G 2H4 514.373.2364 MA Lighting Technology GmbH Dachdeckerstr. 16 97297 Waldbüttelbrunn Germany Phone: +49 (0) 931 49794 -0
--	---	--	--

PSN_DATA_TRACKER_LIST

This chunk is the root for all the PSN_DATA_TRACKER chunks.

PSN_DATA_TRACKER

This contains all the data for a specific tracker. The tracker Id is also the chunk Id.

PSN_DATA_TRACKER_POS

<i>pos_x</i>	A 32-bit float representing the X Euclidean Coordinates of the tracker current position. Positive x is right, positive y is up and Positive z is depth. Position is expressed in meters (m).
<i>pos_y</i>	A 32-bit float representing the Y Euclidean Coordinates of the tracker current position.
<i>pos_z</i>	A 32-bit float representing the Z Euclidean Coordinates of the tracker current position.

PSN_DATA_TRACKER_SPEED

<i>speed_x</i>	A 32-bit float representing the tracker current X velocity. Velocity is expressed in meters per second (m/s).
<i>speed_y</i>	A 32-bit float representing the tracker current Y velocity. Velocity is expressed in meters per second (m/s).
<i>speed_z</i>	A 32-bit float representing the tracker current Z velocity. Velocity is expressed in meters per second (m/s).

DATE : Sept 09, 2019 Doc : PosiStageNet Protocol description v2.03			VYV Corporation 5550 Fullum suite 342 Montréal, Québec, Canada H2G 2H4 514.373.2364 MA Lighting Technology GmbH Dachdeckerstr. 16 97297 Waldbüttelbrunn Germany Phone: +49 (0) 931 49794 -0
--	---	--	--

PSN_DATA_TRACKER_ORI

<i>ori_x</i>	A 32-bit float representing the X component of a vector indicating an axis around which the tracker is rotated. The vector's length is the amount of rotation in radians. The orientation is absolute and not cumulated from packet to packet.
<i>ori_y</i>	A 32-bit float representing the Y component of a vector indicating an axis around which the tracker is rotated. The vector's length is the amount of rotation in radians. The orientation is absolute and not cumulated from packet to packet.
<i>ori_z</i>	A 32-bit float representing the Z component of a vector indicating an axis around which the tracker is rotated. The vector's length is the amount of rotation in radians. The orientation is absolute and not cumulated from packet to packet.

PSN_DATA_TRACKER_STATUS

<i>validity</i>	A 32-bit float representing the tracker's validity.
-----------------	---

PSN_DATA_TRACKER_ACCEL

<i>accel_x</i>	A 32-bit float representing the tracker current X acceleration. Acceleration is expressed in meters per second squared.
<i>accel_y</i>	A 32-bit float representing the tracker current Y acceleration. Acceleration is expressed in meters per second squared.
<i>accel_z</i>	A 32-bit float representing the tracker current Z acceleration. Acceleration is expressed in meters per second squared.

PSN_DATA_TRACKER_TRGTPOS

<i>trgtpos_x</i>	A 32-bit float representing the X position of the target that the tracker is trying to reach. Position is expressed in meters.
<i>trgtpos_y</i>	A 32-bit float representing the Y position of the target that the tracker is trying to reach. Position is expressed in meters.
<i>trgtpos_z</i>	A 32-bit float representing the Z position of the target that the tracker is trying to reach. Position is expressed in meters.

<p>DATE : Sept 09, 2019</p> <p>Doc : PosiStageNet Protocol description v2.03</p>			<p>VYV Corporation 5550 Fullum suite 342 Montréal, Québec, Canada H2G 2H4 514.373.2364</p> <p>MA Lighting Technology GmbH Dachdeckerstr. 16 97297 Waldbüttelbrunn Germany Phone: +49 (0) 931 49794 -0</p>
---	---	--	---

PSN_DATA_TRACKER_TIMESTAMP

tracker_timestamp

This is the number of microseconds elapsed since the PSN server was started to the moment the tracker position was computed. Since some trackers can be computed at different times or even repeated across different frames, this timestamp is usually more accurate. If this field is not present, you can simply use the packet timestamp as a fallback.