

Table of data types

Photon defines a common set of serializable data types across all supported platforms, as listed in the table below.

Client (C++)	Server (C#)
nByte	Byte / byte
short	Int16 / short
int	Int32 / int
int64	Int64 / long
bool	Boolean / bool
float	Single / float
double	Double / double
JString	String / string
Hashtable	Hashtable
Dictionary	Dictionary

We also support arrays for **all** the above types:

Client (C++)	Server (C#)
type*	type[]

Multidimensional arrays are supported (however on the server side they will be interpreted as jagged array with all subarrays in the same dimension having the same size). Jagged arrays are not supported.

Example: int* - one dimension int** - two dimensions

Moreover we support object-arrays, which means arrays of elements of different types, as long as the element-types themselves are supported. For example the first element of the array can be an int, the second one a string.

Client (C++)	Server (C#)
Object*	Object[]

Object is not supported for non-array data.

Finally we support custom types.

This means, if your need to send some custom data, which can not be represented easily by the Photon-builtin data-types, like for example a game specific container class, then you can simply implement Photon's custom type interface for it and this way supply Photon with the needed abilities to handle your type and then you can just send and receive your custom type with Photon.

All custom types are automatically supported as (multi-dimensional) arrays, too.

Please refer to the API doc for class [CustomType](#) for details.

All of the above types are supported as values in [Hashtable](#) and [Dictionary](#) instances.

As keys both these containers accept the following types:

nByte
short
int
int64
float
double
JString