Data Types

The following table lists all possible data types used in the protocol. The types used are derived from C++ data types. Integers are transmitted in <u>network byte order</u>.

Name	Implementation	
bool	A bool is represented with an 8-bit integer. A value of 1 is "true" and a value of 0 is "false". No other values are permitted. Implementations may use either signed or unsigned representations, since only 1 and 0 are valid states.	
byte	8-bit unsigned integer	
short	16-bit unsigned integer (Endianess = network byte order = Big Endian)	
int	32-bit signed integer (Endianess = network byte order = Big Endian)	
double	IEEE 754 - double floating point (64-bit)	
string narrow	The string data types are used to transmit text. They are compound data types that consist of: 1. The text character count 2. The text data (so the text itself)	The text data encoding used in the narrow version is ASCII. Each character is encoded in one byte. The underlying C++ type is char (8-bit)
string wide	The character count is transmitted in an short (as defined above). The text data encoding depends on the string type (narrow or wide). See right column for details.	The text data encoding used in the wide version is UCS-2. Each character is encoded in one short (as defined above). The underlying C++ type is wchar_t (which is a 16-bit character in Pandoras Box)
byte buffer	Byte buffers are used to transmit binary data, such as image data or vectors. They are basically an array of bytes. The byte buffer is a compound data type that contains: 1. The length of the data encoded in a integer (as defined above) 2. The data bytes each encoded as bytes	

Pandoras Automation also uses enumerations that contain constants for frequently used values. These types are not part of the protocol since they are transmitted using the base data types defined above.

TCP Header

TCP Connections can be established to any Pandoras Box running in Master mode. The TCP port used is **6211**. After successfully connecting Pandoras Box is ready to receive commands.

All TCP Pandoras Automation packets consist of a header + data. The header is used in both directions.

Header Structure

Field	Data Type	Description
#1 Identifier	4 bytes	A constant value that denotes the beginning of a Pandoras Box Automation Message. 0x50 0x42 0x41 0x55 (or 'PBAU' ASCII encoded)
#2 Version	byte	The protocol version, constant value 0x01
#3 Domain	integer	The Domain ID Pandoras Box operates on. Default is 0x00
#4 Length	short	The message/data length (excluding header length)
#5 ConnectionID	integer	An integer of your choise. PB will return this connection id in responses. The connection ID should not change during a session. It doesn't need to be unique. Default is 0x00
#6 Protocol	byte	The protocol identifier. For TCP use 0x00
#7 Checksum	byte	To calculate the checksum take the sum of all bytes from fields 2 - 6 of the header and divide by 255. The remainder of the division is the checksum value (modulus 255)
Total	17 bytes	

Automation

Message Structure

Field	Data Type	Description
Header	17 bytes	The TCP Message header as described before. Note that the "Code" field should be included in both the checksum and length fields.
Code	short	The message code. When sending a command this identifies the requested action. Pandoras Box will (almost always) respond with the same code. Implementations should not check responses for the exact code, but rather for a positive value to determine success. Negative codes signalize failure.
Data		The message body. This contains all the arguments or return values corresponding to the message code.