

# Data representation on external devices



**ASSIGN  
BUSTER**

The IOstream is used for input and output of data. This is the transfer of data between an external device and a program; this makes it convenient for a program to do data processing efficiently. There are two main types of external representation using the IOstream; portable data exchange format, and the human-readable form of display. These are traditionally used for text processing. It involves formatting and code conversion.

Formatting is the translation of a byte sequence of data (binary digits) into a human readable character sequence (characters). Code conversion involves translation, where one character representation is translated to another. The IOstream has two layers which communicate through a buffer. One handles formatting, which entails transforming internal data representation of a program, while the other transport and code conversion from one device to the other, basically the transport layer consumes and produces characters.

Data representation on an XDR standard works like this, when some information is to be send from one computer to the other, a device in the sending computer converts the information to a XDR representation with data types, then XDR encoding is used by the two computers to transmit it across the network. When the receiving computer receives it, it then uses its device to translate it back to its own internal representation. This happens quickly that the user will access it as if it was on the local machine.

This work is mostly for the presentation layer, where each machine's device only needs to know how to translate from its own internal representation to the XDR representation and vice-versa. With this kind of a situation the two devices don't need to know the internal details of the other. The idea driven by external data representation (XDR) is basically to enable a user in one

computer to either write or read files on another computer as easily as they do on their local computer. It can also be seen as a way of describing and encoding data.

This makes it possible for two machines to communicate through transferring data from one computer to the other, it can also be used by machines which are diverse using the ISO presentation layer. There is a language similar to C language, which XDR has to use to describe data formats, it's not a programming language and it is only used to describe data from an intricate format to a concise manner. There is an alternative of using graphic representations but this language is mostly faced with some complexity.

XDR language makes assumptions that bytes are portable; there is a device that should encode the bytes onto a media such that other devices can decode them. Representation of data to send must have 32 bits of data, numbered 0 to n-1 (bytes), these are written to a byte stream, they are numbered such that m+1 is preceded by m. XDR standard indicates how its data types are to be declared with a graphical illustration of the encoding. Data sequences of variable length are denoted by angle brackets while fixed-length is denoted by square brackets.

Integers are denoted by the letters n, i, and m. Among others which are not discussed in this paper. There are two types of XDR integers called 32-bit datum that encodes signed and unsigned integers. The signed one's range between -2147483648 and 2147483647, are encoded by an XDR signed integer, and they are represented in a two complement notation. They have the same representation as Enumerations The unsigned ones range between

0 to 4294967295, and are encoded by an XDR unsigned integer. This standard also uses Boolean and unsigned hyper integers as well as hyper integers.