

# 1 Unformatted Plot3D grid file format in C++

Plot3D grid files are traditionally written and read in Fortran routines. Fortran is able to store files in three basic formats: Formatted, Unformatted, and Binary.

A Formatted file is a plain text file with lines breaks between Fortran `write` statements. These files are easily written, and read, in C++ as the standard C++ library provides simple streams for this purpose.

An Unformatted file is one which is stored in a system specific binary format. However, each Fortran `write` statement results in a number being written at the beginning and end of the data contained with the statement. The number is the same at both ends, and depicts the size, in bytes, of the data being written by that particular `write` statement. The purpose of these numbers is twofold: to ensure data consistency, and to allow for file transparency between big and little endian machines (for example, the first `write` command may be of a plain integer, which has four bytes. In an Unformatted file this integer will be both pre and proceeded by the number 4).

A Binary file is similar to the Unformatted format, except that the pre and proceeding integer values on each line are omitted. Hence, this format is machine specific.

value	byte-size	description
4	4	
$n_d$	↖	number of domains
4	4	
$4 \times 3 \times n_d$	4	
$N_{x_1} N_{y_1} N_{z_1} \cdots N_{x_{n_d}} N_{y_{n_d}} N_{z_{n_d}}$	↖	number of points in each domain and direction
$4 \times 3 \times n_d$	4	
$N_{x_1} \times N_{y_1} \times N_{z_1} \times (3 \times f_s + 4)$	4	( $f_s$ is the <code>float</code> size, usually 4 or 8)
$X_1 \ Y_1 \ Z_1 \ B_1$	↖	$X, Y, Z$ , and blanking vectors
$N_{x_1} \times N_{y_1} \times N_{z_1} \times (3 \times f_s + 4)$	4	
:	:	:
$N_{x_{n_d}} \times N_{y_{n_d}} \times N_{z_{n_d}} \times (3 \times f_s + 4)$	4	
$X_{n_d} \ Y_{n_d} \ Z_{n_d} \ B_{n_d}$	↖	$X, Y, Z$ , and blanking vectors
$N_{x_{n_d}} \times N_{y_{n_d}} \times N_{z_{n_d}} \times (3 \times f_s + 4)$	4	