

[Documentation Home](#) > [Solaris Dynamic Tracing Guide](#) > [Chapter 2 Types, Operators, and Expressions](#) > Data Types and Sizes

Solaris Dynamic Tracing Guide

- [Previous: Identifier Names and Keywords](#)
- [Next: Constants](#)

# Data Types and Sizes

D provides fundamental data types for integers and floating-point constants. Arithmetic may only be performed on integers in D programs. Floating-point constants may be used to initialize data structures, but floating-point arithmetic is not permitted in D. D provides a 32-bit and 64-bit data model for use in writing programs. The data model used when executing your program is the native data model associated with the active operating system kernel. You can determine the native data model for your system using `isainfo -b`.

The names of the integer types and their sizes in each of the two data models are shown in the following table. Integers are always represented in twos-complement form in the native byte-encoding order of your system.

Table 2–2 D Integer Data Types

Type Name	32–bit Size	64–bit Size
char	1 byte	1 byte
short	2 bytes	2 bytes
int	4 bytes	4 bytes
long	4 bytes	8 bytes
long long	8 bytes	8 bytes

Integer types may be prefixed with the signed or unsigned qualifier. If no sign qualifier is present, the type is assumed to be signed. The D compiler also provides the type aliases listed in the following table:

Table 2–3 D Integer Type Aliases

Type Name	Description
<code>int8_t</code>	1 byte signed integer
<code>int16_t</code>	2 byte signed integer
<code>int32_t</code>	4 byte signed integer
<code>int64_t</code>	8 byte signed integer
<code>intptr_t</code>	Signed integer of size equal to a pointer
<code>uint8_t</code>	1 byte unsigned integer
<code>uint16_t</code>	2 byte unsigned integer
<code>uint32_t</code>	4 byte unsigned integer
<code>uint64_t</code>	8 byte unsigned integer
<code>uintptr_t</code>	Unsigned integer of size equal to a pointer

These type aliases are equivalent to using the name of the corresponding base type in the previous table and are appropriately defined for each data model. For example, the type name `uint8_t` is an alias for the type unsigned char. See [Chapter 8, Type and Constant Definitions](#) for information on how to define your own type aliases for use in your D programs.

D provides floating-point types for compatibility with ANSI-C declarations and types. Floating-point operators are not supported in D, but floating-point data objects can be traced and formatted using the `printf()` function. The floating-point types listed in the following table may be used:

Table 2–4 D Floating-Point Data Types

--

Type Name	32-bit Size	64-bit Size
float	4 bytes	4 bytes
double	8 bytes	8 bytes
long double	16 bytes	16 bytes

D also provides the special type `string` to represent ASCII strings. Strings are discussed in more detail in [Chapter 6, Strings](#).

- [Previous: Identifier Names and Keywords](#)
- [Next: Constants](#)
- © 2010, Oracle Corporation and/or its affiliates