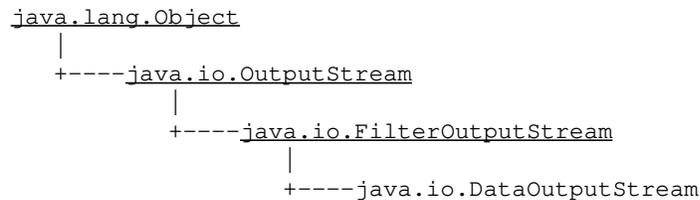


# Class `java.io.DataOutputStream`



public class **DataOutputStream**  
extends [FilterOutputStream](#)  
implements [DataOutput](#)

This class lets you write primitive Java data types to a stream in a portable way. Primitive data types are well understood types with associated operations. For example, an Integer is considered to be a good primitive data type. The data can be converted back using a `DataInputStream`.

---

## Variable Index

- o **written**  
The number of bytes written so far.

## Constructor Index

- o **DataOutputStream(OutputStream)**  
Creates a new `DataOutputStream`.

## Method Index

- o **flush()**  
Flushes the stream.
- o **size()**  
Returns the number of bytes written.
- o **write(int)**  
Writes a byte.
- o **write(byte[], int, int)**  
Writes a sub array of bytes.

- o **writeBoolean**(boolean)  
Writes a boolean.
- o **writeByte**(int)  
Writes an 8 bit byte.
- o **writeBytes**(String)  
Writes a String as a sequence of bytes.
- o **writeChar**(int)  
Writes a 16 bit char.
- o **writeChars**(String)  
Writes a String as a sequence of chars.
- o **writeDouble**(double)  
Writes a 64 bit double.
- o **writeFloat**(float)  
Writes a 32 bit float.
- o **writeInt**(int)  
Writes a 32 bit int.
- o **writeLong**(long)  
Writes a 64 bit long.
- o **writeShort**(int)  
Writes a 16 bit short.
- o **writeUTF**(String)  
Writes a String in UTF format.

## Variables

### o **written**

```
protected int written
```

The number of bytes written so far.

## Constructors

### o **DataOutputStream**

```
public DataOutputStream(OutputStream out)
```

Creates a new `DataOutputStream`.

**Parameters:**

out – the output stream

## Methods

### o **write**

```
public synchronized void write(int b) throws IOException
```

Writes a byte. Will block until the byte is actually written.

**Parameters:**

b – the byte to be written

**Throws:** IOException

If an I/O error has occurred.

**Overrides:**

write in class FilterOutputStream

**o write**

```
public synchronized void write(byte b[],
                               int off,
                               int len) throws IOException
```

Writes a sub array of bytes.

**Parameters:**

b – the data to be written

off – the start offset in the data

len – the number of bytes that are written

**Throws:** IOException

If an I/O error has occurred.

**Overrides:**

write in class FilterOutputStream

**o flush**

```
public void flush() throws IOException
```

Flushes the stream. This will write any buffered output bytes.

**Throws:** IOException

If an I/O error has occurred.

**Overrides:**

flush in class FilterOutputStream

**o writeBoolean**

```
public final void writeBoolean(boolean v) throws IOException
```

Writes a boolean.

**Parameters:**

v – the boolean to be written

**o writeByte**

```
public final void writeByte(int v) throws IOException
```

Writes an 8 bit byte.

**Parameters:**

v – the byte value to be written

## o writeShort

```
public final void writeShort(int v) throws IOException
```

Writes a 16 bit short.

**Parameters:**

v – the short value to be written

## o writeChar

```
public final void writeChar(int v) throws IOException
```

Writes a 16 bit char.

**Parameters:**

v – the char value to be written

## o writeInt

```
public final void writeInt(int v) throws IOException
```

Writes a 32 bit int.

**Parameters:**

v – the integer value to be written

## o writeLong

```
public final void writeLong(long v) throws IOException
```

Writes a 64 bit long.

**Parameters:**

v – the long value to be written

## o writeFloat

```
public final void writeFloat(float v) throws IOException
```

Writes a 32 bit float.

**Parameters:**

v – the float value to be written

## o writeDouble

```
public final void writeDouble(double v) throws IOException
```

Writes a 64 bit double.

**Parameters:**

v – the double value to be written

## o writeBytes

```
public final void writeBytes(String s) throws IOException
```

Writes a String as a sequence of bytes.

**Parameters:**

s – the String of bytes to be written

**o writeChars**

```
public final void writeChars(String s) throws IOException
```

Writes a String as a sequence of chars.

**Parameters:**

s – the String of chars to be written

**o writeUTF**

```
public final void writeUTF(String str) throws IOException
```

Writes a String in UTF format.

**Parameters:**

str – the String in UTF format

**o size**

```
public final int size()
```

Returns the number of bytes written.

**Returns:**

the number of bytes written thus far.

---

[All Packages](#)

[This Package](#)

[Previous](#)

[Next](#)