

## String

BSON strings are UTF-8. In general, drivers for each programming language convert from the language's string format to UTF-8 when serializing and deserializing BSON. This makes it possible to store most international characters in BSON strings with ease.<sup>29</sup> In addition, MongoDB `$regex` queries support UTF-8 in the regex string.

## Timestamps

BSON has a special timestamp type for *internal* MongoDB use and is **not** associated with the regular *Date* (page 180) type. Timestamp values are a 64 bit value where:

- the first 32 bits are a `time_t` value (seconds since the Unix epoch)
- the second 32 bits are an incrementing `ordinal` for operations within a given second.

Within a single `mongod` instance, timestamp values are always unique.

In replication, the *oplog* has a `ts` field. The values in this field reflect the operation time, which uses a BSON timestamp value.

---

**Note:** The BSON timestamp type is for *internal* MongoDB use. For most cases, in application development, you will want to use the BSON date type. See *Date* (page 180) for more information.

---

If you insert a document containing an empty BSON timestamp in a top-level field, the MongoDB server will replace that empty timestamp with the current timestamp value. For example, if you create an insert a document with a timestamp value, as in the following operation:

```
var a = new Timestamp();  
  
db.test.insert( { ts: a } );
```

Then, the `db.test.find()` operation will return a document that resembles the following:

```
{ "_id" : ObjectId("542c2b97bac0595474108b48"), "ts" : Timestamp(1412180887, 1) }
```

If `ts` were a field in an embedded document, the server would have left it as an empty timestamp value.

Changed in version 2.6: Previously, the server would only replace empty timestamp values in the first two fields, including `_id`, of an inserted document. Now MongoDB will replace any top-level field.

## Date

BSON Date is a 64-bit integer that represents the number of milliseconds since the Unix epoch (Jan 1, 1970). This results in a representable date range of about 290 million years into the past and future.

The *official BSON specification*<sup>30</sup> refers to the BSON Date type as the *UTC datetime*.

Changed in version 2.0: BSON Date type is signed.<sup>31</sup> Negative values represent dates before 1970.

---

## Example

Construct a *Date* using the `new Date()` constructor in the `mongo` shell:

---

<sup>29</sup> Given strings using UTF-8 character sets, using `sort()` on strings will be reasonably correct. However, because internally `sort()` uses the C++ `strcmp` api, the sort order may handle some characters incorrectly.

<sup>30</sup><http://bsonspec.org/#/specification>

<sup>31</sup> Prior to version 2.0, *Date* values were incorrectly interpreted as *unsigned* integers, which affected sorts, range queries, and indexes on *Date* fields. Because indexes are not recreated when upgrading, please re-index if you created an index on *Date* values with an earlier version, and dates before 1970 are relevant to your application.