

# Why doesn't `String.Format` throw a `FormatException` if you pass too many parameters?

 [devblogs.microsoft.com/oldnewthing/20090810-00](http://devblogs.microsoft.com/oldnewthing/20090810-00)

August 10, 2009



Raymond Chen

Welcome to CLR Week 2009. As always, we start with a warm-up.

The `String.Format` method doesn't throw a `FormatException` if you pass too many parameters, but it does if you pass too few. Why the asymmetry?

Well, this is the type of asymmetry you see in the world a lot. You need a ticket for each person that attends a concert. If you have too few tickets, they won't let you in. If you have too many, well, that's a bit wasteful, but you can still get in; the extras are ignored. If you create an array with 10 elements and use only the first five, nobody is going to raise an `ArrayBiggerThanNecessary` exception. Similarly, the `String.Format` message doesn't mind if you pass too many parameters; it just ignores the extras. There's nothing harmful about it, just a bit wasteful.

Besides, you probably don't want this to be an error:

```
if (verbose) {
    format = "{0} is not {1} (because of {2})";
} else {
    format = "{0} not {1}";
}
String.Format(format, "Zero", "One", "Two");
```

Think of the format string as a `SELECT` clause from the dataset provided by the remaining parameters. If your table has fields `ID` and `NAME` and you select just the `ID`, there's nothing wrong with that. But if you ask for `DATE`, then you have an error.

[Raymond Chen](#)

**Follow**

