

# New ColdFusion 8 Operators Rock!

Posted At : October 24, 2008 8:00 AM | Posted By : Steve

Related Categories: ColdFusion

Since I first started using ColdFusion (in the late 20th century), I have loved the language. Once of the few things I haven't loved about it, however, is the absence of JavaScript-like operators that so many other languages have.

With ColdFusion 8, those are added in.

I know **ColdFusion 8** has been out a while, but I generally write programs with portability in mind, so I write back a couple of versions. A program that I am working on now, however, really justified the use of ColdFusion 8 - mostly for **cfzip** and **onMissingMethod**.

One of the things that has long been tedious in ColdFusion is incrementing.

```
i = i + 1;
```

This isn't a big deal, but it isn't nearly as clean as it is in many other languages.

```
i++;
```

Fortunately, now it is. To decrement by one, just use i--. (**more on ++**)

Today, I had to do a fair bit of string work. By virtue of ColdFusion's list functions this is often quite easy. Once you get outside of what the list functions can do, however, it again gets more tedious than other languages.

```
MyString = "";
MyString = MyString & MyOtherString;
MyString = MyString & MyThirdString & crlf;
MyString = MyString & MyFourthString;
MyString = MyString & MyFifthString & crlf;
```

Now once you imagine using hard-coded values mixed in with variables for a bunch of string concatenation and the whole thing gets pretty messy.

ColdFusion 8 makes it as easy as it should be.

```
MyString = "";
MyString &= MyOtherString;
MyString &= MyThirdString & crlf;
MyString &= MyFourthString;
MyString &= MyFifthString & crlf;
```

The same principle can also be applied to math.

```
a = a + b;
a = a - b;
a = a * b;
a = a / b;
a = a % b;
```

These operations are easy enough, but if you are performing several operations on the same variable the newer options will be a bit easier to read:

```
a += b;  
a -= b;  
a *= b;  
a /= b;  
a %= b;
```

None of these changes to the language brings new power to the table, but they all improve the ease with which we can write and read code.

Thanks Adobe! Keep up the good work.