

## CMPOne Building Document Expressions

---

### Basics

Document expressions in CMPOne use the C# programming language and its syntax. This document is not meant to teach C# – rather, it is meant to help users of CMPOne with little to no programming experience create simple document expressions.

### Return

To start, we will look at the return statement. The return statement is what we will use to tell CMPOne that this is what we want to show when generating a document with this expression.

#### **Example:**

```
return "Demand Letter";
```

This statement will insert "Demand Letter" (without the quotation marks) into the document.

### Using Merge Fields

The power of CMPOne document expressions comes from the ability to use and manipulate existing merge fields. You have the ability to use mathematical expressions with our merge fields, add a string of words to a merge field to give it context, evaluate if / then statements, and more.

Please note that by default, document expressions are evaluated as "strings" or text fields. In order to perform mathematical calculations, we recommend that you wrap the field in the "Zero" function to force it to evaluate to a number. All expressions must return a value of some kind to be considered valid, even if that value is an empty string.

## CMPOne Building Document Expressions

---

### Examples:

```
return [FileNumber];
```

This statement will insert the current file number into the document.

```
return (Zero([PrincipalAmount]) + 100).ToString("c");
```

This statement will take the current Principal Amount on the file and add 100 to it. This statement can be changed to subtract this amount (-), multiply (\*), or divide (/). You can also change that amount added to any numerical value.

This can also be written as:

```
return string.Format("{0:c}", Zero([PrincipalAmount]) + 100);
```

```
return (Zero([PrincipalAmount]) + Zero([FeesAmount])).ToString("c");
```

Similar to the example above, this example adds two items together. In this case, it is two merge fields being added together. This can be done with any number of merge fields, as long as they are numeric.

```
return "This is the principal amount: " + [PrincipalAmount];
```

This expression adds a string before the merge field. Strings can be added before and after by using quotations and a plus sign.

### If / Then Example:

```
if (Zero([CurrentBalance]) > 5000)
```

```
return "This balance is over 5,000!";
```

```
else
```

```
return "This is a small balance file!";
```

This expression is more complex, spanning 4 lines. In this example, a if / then statement is used directing the system to output "This balance is over 5,000!" if the balance is greater than 5000 or "This is a small balance file!" if the balance is less than or equal to 5000.

## CMPOne Building Document Expressions

---

### Testing

It is important to test all expressions before using them within documents. Testing an expression allows you to ensure there are no syntax errors and the outcome is what you expect.

- To test a document expression, click “Test” above the document expression.  
*(Note: Make sure you are on a file that has the fields you are currently testing populated in the system.)*
  - o If you receive a “Success!” result, ensure the output is correct and then save.
  - o If you receive an error, check your expression again for syntax errors.  
Some common errors are misspelled words, no brackets around merge fields, and no semicolons.
- To test your new expression within a document template, you must save the expression and then enter into a template. You are not able to test new expressions if your template is already open in your browser.

If after testing you still receive an error and you cannot identify the issue, please contact us using the contact support icon within CMPOne or through our information below and a representative will get with you shortly to assist.

### **Complex Expression Examples:**

#### **“CourtCaseNumber”**

```
if ([CaseNumber].ToString()!="") return "\n" + [VenueCounty].ToString() + " County Court Case  
No. " + [CaseNumber].ToString() ;  
  
return "";
```

#### **JudgmentLessJudgmentPayments**

```
return string.Format("{0:C}",Zero([FinalJudgmentAmount])-Zero([PostJudgmentDebtorPayments]));
```

#### **TotalDays**

```
return (DateTime.Today - Convert.ToDateTime([InterestDate])).TotalDays;
```



## CMPOne Building Document Expressions

---

### **Firm Pleading Signature**

return [CompanyStreet] + "\n" + [CompanyCityStateZip] + "\n" + [CompanyPhoneNum] + "\n" +  
[CompanyEmailAdresss] + "\n\n\n" + "By: \_\_\_\_\_" + "\n" + " " +  
[AttorneyFullName] + ", Esq." + "\n" + " Fla. Bar No. " + [AttorneyBarNumber];

### **Questions?**

**support@casemasterpro.com**

(386) 675-0177

