

Adding Formulas to a Document

Beginning in WordWriter version 4.1, WordTemplate now has the ability to evaluate a certain number of [formulas](#) on data fields that are brought into a template.



WordWriter formulas are only supported with the Word 2007 OOXML file format (.docx/.docm). This is an Enterprise Edition only feature.

Placing Formulas in Merge Fields

Formulas in a Merge Field in your template can be used to calculate average, products, sums, and other formulas based on values in the data fields being bound to the template.

To place a formula into your template document, insert a merge field with a value as following:

```
<<=FORMULA(DataSource.Field)>>
```

The following formulas are valid with WordTemplate and will be evaluated when the data are bound to the document: AVERAGE, COUNT, COUNTA, MAX, MIN, PRODUCT, STDEV, STDEVP, SUM, VAR, and VARP. Additional information is available on the [Creating Merge Fields](#) page.

The WordTemplate Formula Demo uses formulas to display the average, standard deviation, maximum, and minimum college entrance exam scores of a group of fictional students as part of a table including their individual scores. A repeat block on the first row allows WordTemplate to bring in all of the values of the referenced fields.

Test date: «TestDate»
Test location: «Town»
Number of students: «=COUNTA(Items.StudentID)»

Student ID	Reading	Writing	Math	Total
«StudentID»	«ReadingScore»	«WritingScore»	«MathScore»	«TotalScore»
Avg	«=AVERAGE(Items.ReadingScore)»	«=AVERAGE(Items.WritingScore)»	«=AVERAGE(Items.MathScore)»	«=AVERAGE(Items.TotalScore)»
Stdev	«=STDEV(Items.ReadingScore)»	«=STDEV(Items.WritingScore)»	«=STDEV(Items.MathScore)»	«=STDEV(Items.TotalScore)»
High	«=MAX(Items.ReadingScore)»	«=MAX(Items.WritingScore)»	«=MAX(Items.MathScore)»	«=MAX(Items.TotalScore)»
Low	«=MIN(Items.ReadingScore)»	«=MIN(Items.WritingScore)»	«=MIN(Items.MathScore)»	«=MIN(Items.TotalScore)»

The following code opens the template and binds data to it using the SetDataSource and SetRepeatBlock methods, then streams the resulting document to the user:

```

protected void OnDownload(object sender, EventArgs e)
{
    WordTemplate wt = new WordTemplate();
    wt.Open(Page.MapPath("templates/TestScoreReport.docx"));

    string[] fields = new string[] { "TestDate", "School", "Town", "Version" };
    object[] values = new object[] {
        new DateTime(2009, 11, 16).ToShortDateString(),
        "Samuel Adams HS",
        "Milbury",
        wt.Version
    };
    wt.SetDataSource(values, fields, "");

    DataTable dt = GetScores();
    wt.SetRepeatBlock(dt, "Items");

    wt.Process();
    wt.Save(Response, "ScoreReport.docx", false);
}

private DataTable GetScores()
{
    DataTable dt = new DataTable();

    dt.Columns.Add("StudentID", typeof(string));
    dt.Columns.Add("ReadingScore", typeof(int));
    dt.Columns.Add("WritingScore", typeof(int));
    dt.Columns.Add("MathScore", typeof(int));
    dt.Columns.Add("TotalScore", typeof(int));

    Random rand = new Random();

    int rScore, wScore, mScore;
    for (int i = 0; i < 30; ++i)
    {
        rScore = rand.Next(400, 800);
        wScore = rand.Next(400, 800);
        mScore = rand.Next(400, 800);
        dt.Rows.Add(
            string.Format("S{0:000000}", rand.NextDouble() * 1000000),
            rScore, wScore, mScore, rScore + wScore + mScore
        );
    }

    DataTable dtSorted = dt.Clone();
    foreach (DataRow row in dt.Select("", "StudentID asc"))
        dtSorted.ImportRow(row);

    return dtSorted;
}

```

The formulas are evaluated by WordTemplate, as you can see in the output file:

Test date: 11/16/2009

Test location: Milbury

Number of students: 30



Student ID	Reading	Writing	Math	Total
S000972	517	609	690	1816
S012286	690	687	561	1938
S019944	482	516	463	1461
S038044	400	677	725	1802
S144528	510	700	515	1725
S152202	776	749	621	2146
S268508	735	520	785	2040
S268629	454	631	499	1584
S268696	787	548	501	1836
S289371	428	762	417	1607
S291744	637	446	652	1735
S294071	520	421	599	1540
S322061	659	606	653	1918
S419318	561	533	564	1658
S469662	499	561	688	1748
S613649	793	447	697	1937
S619757	622	617	461	1700
S627363	595	770	452	1817
S658488	456	456	427	1339
S676564	488	712	457	1657
S694787	484	493	684	1661
S767279	788	781	431	2000
S786398	741	499	662	1902
S799806	757	578	694	2029
S812340	481	693	499	1673
S838011	416	756	517	1689
S879081	581	427	718	1726
S957026	644	776	730	2150
S975064	495	578	456	1529
S988330	620	636	479	1735
<u>Avg</u>	587.2	606.2	576.6	1769.9
<u>Stdev</u>	125.4	115.4	113.0	194.9
<u>High</u>	793	781	785	2150
<u>Low</u>	400	421	417	1339

