

Christoph Albrecht  
BW End User Tools



**Formula Operators in BW 2.0B**  
Version 1.0

10. January 2002

Distribution: BW DEV

# 1 Percent Functions

## 1.1 Percentage Deviation (%)

### Use

<Operand1> % <Operand2>

### Description

Gives the percentage deviation between operand 1 and operand 2. Identical to the formula  $100 * (<Operand1> - <Operand2>) / \text{abs}(<Operand2>)$ , if <Operand2> is not equal to zero, and DIV0 if <Operand2> is equal to zero.

### Example

*Planned Sales % Actual Sales*, expresses, as a percentage, the difference between the planned sales and the actual sales.

## 1.2 Percentage Share (%A)

### Use

<Operand1> %A <Operand2>

### Description

Gives the percentage share of operand1 in operand2. Identical to the formula  $100 * <Operand1> / \text{abs}(<Operand2>)$ , if <Operand2> is not equal to 0, and Div0, if <Operand2> is equal to 0.

### Example

*Fixed costs %A Costs*, shows how big a part of the total costs is taken up by fixed costs.

## 1.3 Percentage Share of Result (%CT)

### Use

%CT <Operand>

### Description

Shows how high the percentage share is, with regard to the result. The result being the outcome of aggregating the second highest level (interim result).

If a characteristic has been drilled down upon in both the rows and the columns, this relationship is not unique, causing the system to display the warning, and relevant symbol for, *Data does not exist* (see customizing table RSADMINC).

### Example

Year	Region	Sales	%CT Sales
1999	North	30	50
	South	30	50

	(Interim) Result1999	60	33,3333
2000	North	60	50
	South	60	50
	(Interim) Result 2000	120	66.6667
Overall Result		180	100

## 1.4 Percentage Share of the Overall Result (%GT)

### Use

%GT <Operand>

### Description

Shows the percentage share, with regard to the overall result. The overall result being the outcome of aggregating the highest level in the list. The dynamic filters (filters that have not been determined already in the query definition) also play a role in calculating the overall result.

### Example

Year	Region	Sales	%GT Sales
1999	North	30	16,6667
	South	30	16,6667
	(Interim) Result 1999	60	33,3333
2000	North	60	33,3333
	South	60	33,3333
	(Interim) Result 2000	120	66.6667
Overall Result		180	100

## 1.5 Percentage Share of the Report Result (%RT)

Much the same as &GT. Unlike the process of calculating the overall result, dynamic filters are **not** used in the calculation of the report result. This means that, independently of the filter status and the navigation status, the same value is always normalized.

## 2 Data Functions

### 2.1 COUNT()

#### Use

COUNT(<Expression>)

#### Description

Delivers the value 1, if <expression> <> 0, otherwise 0.

## 2.2 DELTA () [obsolete!]

### Use

DELTA(<expression>)

### Description

Delivers the value 1, if <expression> is 0, otherwise 1.

The function is identical to the NOT operator, and should no longer be used.

## 2.3 NDIV0 ()

### Use

NDIV(<expression>)

### Description

Is 0, if the calculation of <expression> causes a division by 0. Otherwise, the result is the value of the expression. Used to avoid the system displaying an error message, or to enable you to continue to work with a defined result.

## 2.4 NODIM ()

### Use

NODIM(<expression>)

### Description

Delivers from <expression> the basic numeric value, and suppresses the unit and the currency. Using this function means that you avoid the 'mixed currency' character (see Customizing table RSADMINC).

## 2.5 NOERR ()

### Use

NOERR(<expression>)

### Description

Is 0, if <expression> leads to a calculation error. Otherwise, the result is the value of the expression. Used to avoid the system returning an error message, or to enable you to continue working with a defined result.

### Example

NOERR(SQRT(-1)) = 0

## 2.6 SUMCT

### Use

SUMCT <operand>

### Description

Delivers the (interim) result of the operand in all the rows or columns (compare %CT).

**Example**

Year	Region	Sales	SUMCT Sales
1999	North	30	60
	South	30	60
	(Interim) Result 1999	60	180
2000	North	60	120
	South	60	120
	(Interim) Result 2000	120	180
Overall Result		180	180

**2.7 SUMGT****Use**

SUMGT &lt;operand&gt;

**Description**

Delivers the overall result of the operands (compare %GT).

**Example**

Year	Region	Sales	SUMGT Sales
1999	North	30	180
	South	30	180
	(Interim) Result 1999	60	180
2000	North	60	180
	South	60	180
	(Interim) Result 2000	120	180
Overall result		180	180

**2.8 SUMRT****Use**

SUMRT &lt;operand&gt;

**Description**

Delivers the report result of the operand (see %GT and %RT for the difference between the overall result and the report result).

## 3 Boolean Operators

### 3.1 Relational Operators ==, <>, <, >, <=, >=

#### Use

<expression1> <operator> <expression2>

#### Description

Is 1, if <expression1> is in the <operator> relation with <expression2> (meaning <expression1> <operator> <expression2> is true), otherwise 0. Only the numeric values from <expression1> and <expression2> are compared. The unit is not referred to.

#### Example

*Costs < Sales* delivers 1, if the sales value is greater than the costs value, and 0, if the costs value is greater than or equal to the sales value.

### 3.2 Logical And (AND)

#### Use

<expression1> AND <expression2>

#### Description

Is 1, if <expression1> and <expression2> are both not equal to 0, otherwise 0. Only the numeric values from <expression1> and <expression2> are taken into account. The unit is not referred to.

### 3.3 Logical Or (OR)

#### Use

<expression1> OR <expression2>

#### Description

Is 1, if <expression1> or <expression2> is not equal to 0, otherwise 0. Only the numeric values from <expression1> and <expression2> are taken into account. The unit is not referred to.

### 3.4 Logical Exclusive Or (XOR)

#### Use

<expression1> XOR <expression2>

#### Description

Is 1, if either <expression1> or <expression2> (but not both) is not equal to 0, otherwise 0. Only the numeric values from <expression1> and <expression2> are taken into account. The unit is not referred to.

### 3.5 Logical Not (NOT)

#### Use

NOT <expression>

#### Description

Is 1, if <Expression> is 0, otherwise 0. Only the numeric value from <expression> is taken into account. The unit is not referred to.

### 3.6 LEAF()

#### Use

This operator allows you to carry out various calculations on results rows and elementary rows.

#### Description

Is 0 for results rows or real (inner) nodes of a hierarchy, and the value 1 for elementary rows or the leaves of a hierarchy.

### 3.7 Conditional Calculations (IF-THEN-ELSE)

#### Description

Conditional calculations are made using Boolean operators. An expression like, IF <logic expression> THEN <expression1> ELSE <expression2> can also be written as the formula <logic expression> \* <expression1> + NOT <logic expression> \* <expression2>

#### Example

IF *Actual Costs* > *Planned Costs* THEN *Planned Costs* + *Plan Deviation* ELSE *Actual Costs*

can be written as the following formula:

$(Actual\ Costs > Planned\ Costs) * (Planned\ Costs + Plan\ Deviation) + (Actual\ Costs \leq Planned\ Costs) * Actual\ Costs$

or

$(Actual\ Costs > Planned\ Costs) * (Planned\ Costs + Plan\ Deviation) + (NOT (Actual\ Costs > Planned\ Costs)) * Actual\ Costs$

## 4 Technical Notes

### 4.1 Testing <expression> = 0

The test to see whether a mathematical expression will produce 0, has been designed to avoid problems caused by rounding operations that use  $abs(<expression>) < 10^{*-6}$ . In other words, all values in the interval  $[-10^{*-6}; 10^{*-6}] = [-0,000001; 0,000001]$  are regarded as 0.

### 4.2 Result for Non-defined Mathematical Operations

If you carry out a mathematical operation that is not permitted or not defined (for example, division by 0, square root of negative values) the system displays the symbol for *Division by 0* or *Data does not exist* (see Customizing table RSADMIN), and the calculation is terminated.

You can suppress this function, if the expression that causes the error is compounded by the operator NDIV0() or NOERR() (see above). Note: If you use these operators, check carefully that the user is still able to interpret the result meaningfully, even if an error should occur.

## 4.3 ABAP Coding for Calculating Operators

If you are in anyway unsure, take a look at the operator functions in the ABAP coding (Include LRRK0F10, Form LOC\_RECHNEN).