

# Table of Contents

- Special usage of rules** ..... 2
- Rule tables** ..... 2
- Relational expression ..... 2
- String Expression ..... 2
- List Expression ..... 3
- Range Expression ..... 3
- WAS CHANGED expression ..... 3
- IS UNDEFINED expression ..... 3
- Assignment action ..... 4
- Incremental action ..... 4
- Usage of rules inside tags** ..... 4
- Arithmetic Operations ..... 5
- Conditional Expressions ..... 5
- Expressions marking section boundaries ..... 5
- Report Designer** ..... 6
- Queries** ..... 6

[Manuals, Rule Language](#)

# Special usage of rules

There are several places in the software where the special format of the rule expressions is used.

## Rule tables

Inside rule tables some rule expressions can be represented in the simplified format. This can happen only if an attribute identifier is specified in the header of the rule table column or row. For example, if `Account.Balance` attribute is in the header of the table column and the value of the cell is 1000, the value is the short form of the following expression:

```
Account.Balance = 1000
```

The following expressions can be shortened in cells of a rule table:

## Relational expression

Compared to the full form of the relational expression (see section [Relational Expression](#)) the shortened version used in rule tables has attribute identifier omitted from the expression. Also when "=" is used it is also omitted from the expression. Examples of shortened relational expressions and the corresponding full versions follow (these assume that the header of the rule table column is `Account.Balance`):

1. 1000  
the corresponding expression in the full form is:  
`Account.Balance = 1000`
2. <1000  
the corresponding expression in the full form is:  
`Account.Balance < 1000`
3. >= Transaction.Amount - 500  
the corresponding expression in the full form is:  
`Account.Balance >= Transaction.Amount - 500`

## String Expression

The short form of the `String Expression` (see section [String Expression](#)) has attribute identifier omitted from the expression. The examples, which assume that `Account.Name` is the header of the column, follow:

1. CONTAINS 'ith'  
The corresponding expression in the full form is:  
`Account.Name CONTAINS 'ith'`
2. STARTSWITH 'John'

The corresponding expression in the full form is:  
`Account.Name STARTSWITH 'John'`

## List Expression

The short form of the List Expression (see section [List Expression](#)) has the attribute identifier omitted from the expression. Also OR is used instead of comma to separate members of the list. The examples, which assume that `Account.State` is the header of the column, follow:

1. `'OPEN' OR 'CLOSED'`  
the corresponding expression in the full form is:  
`Account.State IN 'OPEN', 'CLOSED'`
2. `'OPEN' OR 'CLOSED' OR 'SUSPENDED'`  
the corresponding expression in the full form is: `Account.State IN 'OPEN', 'CLOSED', 'SUSPENDED'`

## Range Expression

The short form of the Range Expression (see section [Range Expression](#)) has the attribute identifier omitted from the expression. Also AND is always used to separate boundaries of the range. There is also an alternative form, which has two boundaries separated by “-” sign. The examples, which assume that `Account.Balance` is the header of the column, follow:

1. `BETWEEN 1000 AND 5000`  
The corresponding expression in the full form is:  
`Account.Balance BETWEEN 1000 AND 5000`
2. `1000-5000`  
the corresponding expression in the full form is:  
`Account.Balance BETWEEN 1000 AND 5000`

## WAS CHANGED expression

The short form of the WAS CHANGED expression (see section [WAS CHANGED](#)) has the attribute identifier omitted from the expression. The examples, which assume that `Account.State` is the header of the column, follow:

1. `WAS CHANGED`  
The corresponding expression in the full form is:  
`Account.State WAS CHANGED`
2. `WAS CHANGED TO 'CLOSED'`  
The corresponding expression in the full form is:  
`Account.State WAS CHANGED TO 'CLOSED'`

## IS UNDEFINED expression

The short form of the IS UNDEFINED expression (see section [IS UNDEFINED](#)) has the attribute identifier omitted from the expression. The Examples, which assume that `Account.Name` is the header

of the column, follow:

1. IS UNDEFINED  
The corresponding expression in the full form is:  
`Account.Name IS UNDEFINED`
2. IS DEFINED  
The corresponding expression in the full form is:  
`Account.Name IS DEFINED`

## Assignment action

The short form of the [assignment action](#) has the attribute identifier omitted from the action. The examples, which assume that `Account.Balance` is the header of the column, follow:

1. 1000  
the corresponding action in the full form is:  
`Account.Balance = 1000`
2. Transaction.Amount\*100  
The corresponding action in the full form is:  
`Account.Balance = Transaction.Amount * 100`

## Incremental action

The short form of the [incremental action](#)) has the attribute identifier omitted from the action. "+" and "-" signs can also be used to indicate whether the INCREASE or REDUCE operation is used. The examples, which assume that `Account.Balance` is the header of the column, follow:

1. INCREASE BY 1000  
The corresponding action in the full form is:  
`INCREASE Account.Balance BY 1000`
2. +1000  
the corresponding action in the full form is:  
`INCREASE Account.Balance BY 1000`
3. -20%  
the corresponding action in the full form is:  
`REDUCE Account.Balance BY 20%`

## Usage of rules inside tags

Rule expressions can be used inside special tags in documents. The start of the tag is denoted by the symbol "<<" and the end of the tag - by the symbol ">>". The following expressions can be used inside tags:

1. Arithmetic Operations (see section [Arithmetic Operation](#))
2. Conditional expressions
3. Special expressions identifying boundaries of document sections (MS Word documents only).

## Arithmetic Operations

Any Arithmetic Operation can be used inside a tag, for example:

```
<<Account.Balance>> or <<Account.Balance + 1000>>
```

Arithmetic Operation inside a tag can be optionally followed by the Formatting Expression, which indicates how the value of the tag will be presented to the user when the tag contents is calculated at run time. The Formatting Expression is separated from the Arithmetic Operation by comma or @ symbol. If no formatting expression is specified the value will be formatted according to its default format, which is obtained from the configured presentation information of the attribute(s) participating in the Arithmetic Operation inside the tag.

There are four types of the formatting expression:

1. The Date/Time format
2. The Number format
3. The Duration format.
4. HTML format

The examples of rule expressions using formatting expressions inside tags follow:

1. <<Account.Balance, 00000>>
2. <<Account.OpeningDate, dd/MM/yyyy>>
3. <<ActivityRecord.TimeSpent@WWW DDd HH:mm>>
4. <<Account.Description,html>> (Description attribute is assumed to store text in the HTML format)

## Conditional Expressions

It is possible to indicate that attribute values or static text should be included in a document conditionally. This can be done using conditional expression. The format of this expression is that of any rule that uses condition, except that this expression must use special action called SHOW. For example,

1. <<IF Account.Balance > 1000 Then SHOW Account.Holder.Name>>
2. <<IF Account.Balance > 1000 Then SHOW 'Account balance is greater than 1000'>>

## Expressions marking section boundaries

It is possible to conditionally include entire sections of a document template in the final document. To do this you need to specify the condition of inclusion, mark the start of the section to be included in the document and mark the end of the section in the document. Condition is specified using the conditional expression with the special action SHOW SECTION\_START. This also marks the start of the section. To mark the end of the section include, the following tag: <<SECTION\_END>>. For example,

```
<<IF Account.Balance > 1000 THEN SHOW SECTION_START>>  
The text of the section follows ...
```

```
<<SECTION_END>  
The rest of the text follows
```

**Note:** conditional inclusion of document sections is only available for MS Word document templates. MS Word document templates also support inclusion of sub-documents.

## Report Designer

When rule expressions are used inside tags in the Report Designer the following additional construct can be used:

```
{ Id ( ) }
```

where Id() indicates the name of the report parameter. Before the report using some parameter(s) is executed the user is asked to provide the specific values of the report parameter(s). During the report calculation report parameters are replaced with values provided by the user. For example:

```
<<{REPORT_NAME}>>
```

**Note:** the report parameter expression can only be used inside the Report Designer.

## Queries

In **Aware IM** queries can be defined using the Rule Language. The only Rule Language construct that can be used when constructing queries is the FIND action (see section [FIND Action](#)).

The action can use ? symbol to indicate that the value of the attribute used in the query should be prompted to the user when the query is executed. For example,

```
FIND Account WHERE Account.Balance > ?Balance
```

When such a query is executed, the user is asked to enter the specific value of the account balance. The "Balance" string after the question mark indicates the name of the prompt that will be displayed to the user. See also the "[Configuring Queries](#)" section".

From:  
<http://www.awareim.com/dokuwiki/> - **Documentation**

Permanent link:  
[http://www.awareim.com/dokuwiki/docs/3000\\_rule\\_language/0600\\_rule\\_language?rev=1661385631](http://www.awareim.com/dokuwiki/docs/3000_rule_language/0600_rule_language?rev=1661385631)

Last update: **2022/09/13 18:09**

