Table of Contents

Notes on rule execution 2
Type conversions 2
Comparison 2
References 3

Manuals, Rule Language

Notes on rule execution

Type conversions

When performing arithmetic calculations with attribute values and numeric constants ***Aware IM*** automatically performs the necessary type conversions if arguments are of different types. The following conversion rules apply:

- Addition of two strings causes the strings to be concatenated, for example `'John' + ' ' + 'Smith'` produces 'John Smith'
- Addition of a string and a number produces a string, for example `'Number' + Account.Balance` produces `'Number1000'` if the value of `Account.Balance` is 1000
- If a value is undefined it is omitted from calculations, for example if the value of `Account.Balance` is 1000 and the value of `Transaction.Amount` is undefined, the value of `Account.Balance − Transaction.Amount` will be 1000.
- Strings may not participate in any arithmetic calculations other than addition.
- References may not participate in arithmetic operations.
- A number added (subtracted) to (from) a date produces the date which is “value” days greater or less than the original date, where “value” is the value of the number. For example, `Account.OpeningDate + 7` produces a date that is a week after the account opening date.
- A number added (subtracted) to (from) a timestamp produces the timestamp which is “value” hours greater or less than the original timestamp, where “value” is the value of the number. For example, `Transaction.Timestamp − 3` produces the timestamp that is 3 hours before the transaction timestamp.
- Duration added or subtracted from a date produces date.
- No other arithmetic operations with date/timestamp are allowed.

> **note**
>
> no arithmetic operations with date/timestamp may be used inside the `FIND` action.

- Duration added to or subtracted from another Duration produces Duration
- Attributes of the `Yes/No` type may not participate in arithmetic calculations.
- Attributes of the `Document` type may not participate in arithmetic calculations, although it is possible to assign a value to the attribute of the Document type. For example, `Statement.Doc = 'StatementTemplate'` – assigns the specified template to the attribute of the business object. This will also resolve contents of tags if the template has them. Another example, `Statement.Doc = OtherObject.Doc` – assigns the document of Statement business object to be the exact copy of the document stored in OtherObject.

Comparison

The following rules apply when comparing values in the Relational Expression:

All attribute types can be compared for equality or non-equality (even references), provided that the types on both sides are compatible. If the types are the same on both sides, they are always compatible. If they are different, the following rules apply:

- If strings are compared with numbers, numbers are converted to strings, for example, the following will only be true if the value of `Account.Name` is '1000': `Account.Name = 1000`
- If a value of an attribute is `UNDEFINED` it is considered to be 0 for comparison purposes.

Only numbers, dates and durations can be compared using other operators (<, >, ⇐, >=).

References

It is possible to use references and reference lists inside rule conditions and assign reference and reference lists in actions. For example, the following rule:

```
IF Customer.Policies.Cartype='Wagon' THEN
Customer.Policies.Premium=100
```

will check every policy in the list of policies of a customer and if a car type of the policy is wagon will set the premium of such policy to 100.

The following rule:

```
IF Policy.CarType='Wagon' THEN
   Customer.CurrentPolicy=Policy
```

will set the policy, the car type of which is "wagon", as the policy of the customer.

If multiple references are used in a rule condition the corresponding action will be triggered if the condition holds for ANY value of the reference in the reference list. For example,

```
IF Customer.Policies.CarType='Wagon' THEN ...
```

The action in this rule will be executed if there is any `Policy` that the `Customer` owns that has the `Wagon` car type.