

Table of Contents

Configuring Queries 2

[data](#), [data retrieval](#), [query](#), [user interface](#), [UI](#), [concepts](#)

Configuring Queries

Queries can be defined at the configuration stage - a user can run such pre-configured queries in the Operation Mode. Pre-configured queries can also be referenced by rules (see [FIND](#) action).

When defining a query the configurator has to indicate the business object, (or [business object group](#)) the instances of which the query will be searching. Then the configurator has to indicate the conditions of search - only those instances of the business object that have attribute values matching the specified conditions will be retrieved by the query (if conditions are not specified all instances of the business object will be retrieved). For example, if a query is to retrieve accounts of John Smith, the query has to include a condition that compares the value of the OwnerName attribute with 'John Smith':

```
FIND Account WHERE Account.OwnerName = 'John Smith'
```

One can define multiple conditions for a query - the conditions may be linked with the AND or OR keywords.

Note that query conditions allow specifying not only attributes of the business object that the query will be searching, but also attributes of the related objects. For example, if account owner is a Customer object related to the Account object via the Owner reference attribute, the above query could be specified as follows:

```
FIND Account WHERE Account.Owner.Name = 'John Smith'
```

It is also possible to indicate attributes of objects related to the related objects, for example, Account.Owner.Company.Name. There is no limit on the level of nesting of the related objects.

The configurator may indicate that instances of a business object found by a query should be sorted by a particular attribute(s) in a particular order and also indicate how these instances should be displayed in the Operation Mode - see [Data Presentation](#).

Configuration of queries is described in more detail in the [Adding/Editing Queries](#) section.

note

If a query searches for instances of business objects belonging to a business object group, the conditions of the query may only use attributes common to all members of the group (see [Business Object Groups](#)). When the results of such a query are presented to the user the presentation is only allowed to show common attributes of the group (see [Data Presentation](#)).

note

It is possible to define a query with conditions that use attributes of a business object that the query is not searching for (they are also called *dynamically resolved* attributes or simply *dynamic* attributes). Consider the following query as an example:

```
FIND Account WHERE Account.OwnerName = Customer.Name
```

The business object Customer used in a query condition is not the object that the query is searching for. **AwareIM** replaces references to such attributes with the actual values at run time – just before the search is performed. For example, if the value of the Name attribute of the Customer object is 'John Smith' the actual query performed by **AwareIM** is:

```
FIND Account WHERE Account.OwnerName = 'John Smith'
```

Where does **AwareIM** find the instance of the Customer object and what happens if there are no instances of this object or if there are more than one instance? The answer to the first part of the question is that the instance is taken from the Context (this is explained in the [Other Usages of Context](#) section); the answer to the second part is that if **AwareIM** cannot find the instance of the object in the Context it logs a run-time error and the query is not run. The queries using dynamic attributes should only be used in situations where such attributes can be unambiguously resolved.

See also [Queries that Require User Input](#)

From:

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

Permanent link:

http://www.awareim.com/dokuwiki/docs/2000_concepts/0600_data_retrieval/0100_config_queries

Last update: **2025/06/12 02:09**

