

## Contents

- [Retrieving Query Lists](#)
  - [Transfer.list\(class, \[orderProperty\], \[orderASC\], \[useAliases\]\)](#)
  - [Transfer.listByProperty\(class, property, value, \[orderProperty\], \[orderASC\], \[useAliases\]\)](#)
  - [Transfer.listByPropertyMap\(class, propertyMap, \[orderProperty\], \[orderASC\], \[useAliases\]\)](#)
  - [Transfer.listByQuery\(query\)](#)

---

## Retrieving Query Lists

There are several ways in which to retrieve Queries from Transfer, where sometimes getting an object is not appropriate. This can be particularly true when needing to retrieve a large number of values, and performance is a concern.

### Transfer.list(class, [orderProperty], [orderASC], [useAliases])

[\(API\)](#)

Queries the database for all the records of a given class in the database and returns them

Column names are aliased to the property names set in the configuration file, unless otherwise specified by setting 'useAliases=false'.

```
qCategories = transfer.list("system.Category", "orderIndex");
<--- The variable qCategories will be a query of all categories. The records will be in ascending order based on the orderIndex property. ---!>
```

### Transfer.listByProperty(class, property, value, [orderProperty], [orderASC], [useAliases])

[\(API\)](#)

Queries the database for all the records of a given class in the database, filtered by a property of the class, and returns them.

Column names are aliased to the property names set in the configuration file, unless otherwise specified by setting 'useAliases=false'.

```
qAddresses = transfer.listByProperty("user.address", "state", "NY");
<--- The variable qAddresses will be a query of addresses that have the value of the state property equal to NY. ---!>
```

### Transfer.listByPropertyMap(class, propertyMap, [orderProperty], [orderASC], [useAliases])

[\(API\)](#)

Queries the database for all the records of a given class in the database, filtered by a struct of values, where the struct key is the object property, and the value is the value to filter by.

Column names are aliased to the property names set in the configuration file, unless otherwise specified by setting 'useAliases=false'.

```
map = StructNew();
StructInsert(map, "state", "NY");
StructInsert(map, "city", "Albany");
qAddresses = transfer.listByPropertyMap("user.address", map, "lastName", false);
<--- The variable qAddresses will be a query of addresses that have the value of the state property equal to NY and a value of the city property equal to Albany. The records will be in descending order based on the lastName property. ---!>
```

### Transfer.listByQuery(query)

[\(API\)](#)

Queries the database and returns the results based on the TQL query that is passed to it.

It should be noted that TQL Query Objects are reused after they are executed, so they will lose their state after the listByQuery() operation.

```
query = transfer.createQuery("from post.Post as Post join user.User orderBy Post.DateTime");
qPosts = transfer.listByQuery(query);
<--- The variable qPosts will be a query of posts with their author information. The records will be in ascending order based on the post's dateTime property. ---!>
```

For information on TQL and the TQL Query object, see [Transfer Query Language](#)

Categories: • [TQL](#)