



# JDBC Content Access Module

Configuration Guide  
Version 2.2.0



# Copyright Notice

Copyright © 2010 by MasterObjects, Inc. All rights reserved. U.S. and international patents pending.

MasterObjects, QuestObjects, QuestField, Questlet, QOP, and the Q Arrow logo are trademarks or registered trademarks of MasterObjects, Inc. in the United States and other countries. Other trademarks used in this document are the property of their respective owners. Screen shots were used to the benefit of their respective copyright owners, for informational purposes only. Use of trademarks or screen shots is not intended to convey endorsement or other affiliation with MasterObjects.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher or copyright owner.

MasterObjects has tried to make the information contained in this publication as accurate and reliable as possible, but assumes no responsibility for errors or omissions. MasterObjects disclaims any warranty of any kind, whether express or implied, as to any matter whatsoever relating to this publication, including without limitation the merchantability or fitness for any particular purpose. In no event shall MasterObjects be liable for any indirect, special, incidental, or consequential damages arising out of purchase or use of this publication or the information contained herein.



MasterObjects, Inc.  
1156 Clement Street  
San Francisco, CA 94118

info@masterobjects.com  
<http://www.masterobjects.com>



# Table of Contents

[1]	Introduction	
1.1	About this Document	4
1.2	JDBC CAM Components Overview	4
1.3	Version History	5
[2]	CAM Instance Configuration	
2.1	Settings Reference	6
2.1.1	Authentication settings .....	7
2.1.2	Connection settings .....	7
[3]	Content Query Configuration	
3.1	Settings Reference	9
	Columns returned from queries and QuestField results .....	9
3.1.1	Content Access Module settings .....	10
3.1.2	SQL channel query settings queries.....	10
[4]	Glossary	

# [ 1 ] Introduction

## 1.1 About this Document

	This is the JDBC Content Access Module Administration Guide for QuestFields.
<b>Intended Audience</b>	This document serves as a reference for system administrators who are responsible for installing and managing the QuestFields system. Experience with SQL databases is assumed.
<b>Related Documents</b>	For information about the QuestFields Server, please refer to the <i>QuestFields Server Administration Guide</i> .  For information about the QuestFields Client, please refer to the <i>QuestFields Client Administration Guide</i> .

## 1.2 JDBC CAM Components Overview

<b>JDBC Content Access Module</b>	The JDBC Content Access Module (CAM) provides the connection between the QuestFields Server and SQL database servers. The CAM manages all physical aspects of interaction between the QuestFields system and the SQL servers, including establishing a connection to an SQL database server, executing queries and extracting results.
<b>JDBC Drivers</b>	The JDBC CAM talks to your database(s) through one or more "JDBC drivers". These drivers can be obtained from the database vendor. The QuestFields Server comes bundled with recent JDBC drivers for MySQL, PostgreSQL, and Microsoft SQL Server. Other JDBC drivers can typically be downloaded for free from your database supplier.
<b>Content queries</b>	An SQL Content Query logically defines an SQL query that is used to search a SQL server defined in the CAM. The Channels interpret QuestFields queries and triggers one or more SQL Content queries. The SQL Content query then interacts with the Content Access Module to perform the query on the SQL server and to return the results to the Channel.

## 1.3 Version History

This chapter lists the changes for every release of this document.

- 1.0.2** - Added generic sorting settings (see chapter 3.1.5).
- 2.0.0** - Updated document to describe version 2.0 of the QuestFields Server.
- 2.1.0** - Updated document to describe version 2.1 of the QuestFields Server.
- 2.2.0** - Updated document to describe version 2.2 of the QuestFields Server. Added instructions for installing JDBC drivers in directory [OO\\_HOME/drivers](#).

# [2]

## CAM Instance Configuration

### Connection Pooling

Creating a new connection to a database server for each query performed can be time consuming (often requiring multiple seconds of clock time). In order to maximize efficiency, the JDBC Content Access Module keeps open connections in a connection pool and reuses the connections for subsequent queries. The connection pool can be configured to suit the load requirements of the specific QuestFields Server and database server installation.

### Read-only Connections

All JDBC connections created to the database server are set to read-only, improving system security by ensuring that the QuestFields Server can only query the database server and can not update or change the underlying data. An additional benefit is that on most database servers, read-only connections help maximize query performance.

## 2.1 Settings Reference

Every JDBC Content Access Module instance is configured using a CAM configuration file named `{cam_id}.xml`, that is placed in the QuestFields CAM configuration directory, `QO_HOME/conf/cams`. The configuration file for a CAM is in XML format and is encoded in the UTF-8 character encoding. The settings file can contain comments in standard XML format.

The ID of a Content Access Module instance, used internally in the QuestFields system to identify the CAM instance, is the filename of the CAM's configuration file, without the `.xml` suffix.

The file starts with `<jdbcCam id="{cam_id}">` and ends with `</jdbcCam>`.

## Example CAM Configuration

```
<jdbcCam id="people-finder-sql">
  <userName>OO</userName>
  <password></password>
  <JDBCdriver>org.postgresql.Driver</JDBCdriver>

  <connectionURL>jdbc:postgresql://127.0.0.1/PeopleFinder</connectionURL>
  <connectionProperties>
    <entry>
      <key>aKey</key>
      <value>aValue</value>
    </entry>
    <entry>
      <key>anotherKey</key>
      <value>anotherValue</value>
    </entry>
  </connectionProperties>
  <poolMaxActive>10</poolMaxActive>
  <poolMaxIdle>5</poolMaxIdle>
  <poolMinIdle>5</poolMinIdle>
  <poolMaxWait>100</poolMaxWait>
  <poolTestOnBorrow>true</poolTestOnBorrow>
  <validationQuery><![CDATA[select 1;]]></validationQuery>
</jdbcCam>
```

### 2.1.1 Authentication settings

<b>userName</b>	<i>Use:</i> Username for accessing the database server. <i>Value type:</i> String.
<b>password</b>	<i>Use:</i> Password for accessing the database servers. <i>Value type:</i> String.

### 2.1.2 Connection settings

<b>JDBCdriver</b>	<i>Use:</i> JDBC driver class. <i>Value type:</i> String. This string depends on the database vendor and JDBC driver used. Here are some examples; for other databases please consult with the corresponding vendor:  <i>MySQL:</i> <code>com.mysql.jdbc.Driver</code> <i>PostgreSQL:</i> <code>org.postgresql.Driver</code> <i>Microsoft SQL Server:</i> <code>com.microsoft.sqlserver.jdbc.SQLServerDriver</code>  ⇒ <i>Note: the JDBC driver (jar file) should be installed either in the <code>OO_HOME/drivers</code> directory (available in QuestFields Server 2.2 and higher), or you can add its location to the Java class path used for the application server.</i>
-------------------	--



<b>connectionUrl</b>	<p><i>Use:</i> Database connection URL.</p> <p><i>Value type:</i> URL string in the format expected by the JDBC driver.</p>
<b>poolMaxActive</b>	<p><i>Use:</i> Maximum number of active connections that can be allocated from the pool at the same time.</p> <p><i>Value type:</i> Non-negative integer representing number of connections, or 0 for no limit.</p>
<b>poolMaxIdle</b>	<p><i>Use:</i> Maximum number of active connections that can remain idle in the pool without extra idle ones being released.</p> <p><i>Value type:</i> Non-negative integer representing number of connections, or 0 for no limit.</p>
<b>poolMinIdle</b>	<p><i>Use:</i> Minimum number of active connections that should remain idle in the pool.</p> <p><i>Value type:</i> Non-negative integer representing number of connections, or 0 to create none.</p>
<b>poolMaxWait</b>	<p><i>Use:</i> Maximum time in milliseconds that the pool will wait (when there are no available connections) for a connection.</p> <p><i>Value type:</i> Non-negative integer representing number of milliseconds, or -1 to wait indefinitely.</p>
<b>poolTestOnBorrow</b>	<p><i>Use:</i> The indication whether connections should be validated before being borrowed from the pool. If a connection fails to validate, it will be dropped from the pool.</p> <p><i>Value type:</i> String containing either <code>true</code> or <code>false</code>.</p>
<b>validationQuery</b>	<p><i>Use:</i> SQL query that will be used to validate connections from the pool if <code>testOnBorrow</code> is set to <code>true</code>. Ideally this should be a minimal SQL statement that can execute as fast as possible.</p> <p><i>Value type:</i> String or CDATA block representing an SQL select statement that returns at least one row.</p>
<b>connectionProperties</b>	<p><i>Use:</i> An optional list of JDBC connection properties.</p> <p><i>Value type:</i> List of <code>&lt;property&gt;</code> elements.</p>
<b>entry</b>	<p><i>Use:</i> One connection property.</p> <p><i>Value type:</i> Contains name and value elements.</p>
<b>key</b>	<p><i>Use:</i> Name of connection property.</p> <p><i>Value type:</i> String.</p>
<b>value</b>	<p><i>Use:</i> Value of connection property.</p> <p><i>Value type:</i> String.</p>

# [3]

## Content Query Configuration

SQL Content Query configuration files contain settings for the SQL queries used, the instance Id of the JDBC Content Access Module, and an item to tell the server whether the results are returned in a sorted order.

### 3.1 Settings Reference

Each SQL content query is configured using configuration file named `{content_query_id}.xml`, located in the QuestFields Content Queries configuration directory, `QO_HOME/conf/content-queries`. The configuration file for a CAM is in XML format and is encoded in the UTF-8 character encoding.

The ID of a Content Query, used internally in the QuestFields system to identify the Content Query, is the filename of the channel's configuration file, without the `.xml` suffix.

The file starts with `<jdbcContentQuery id="{content_query_id}">` and ends with `</jdbcContentQuery>`.

#### Example Content Query Configuration

```
<jdbcContentQuery id="sql-first">
  <camId>people-finder-sql</camId>
  <query><![CDATA[select email, first || ' ' || last, ' ', email,
phone,' ',city || ', ' || state, position from people where
(lower(first) LIKE @firstName%) order by first,last]]></query>
  <sortedResults>true</sortedResults>
</jdbcContentQuery>
```

#### Columns returned from queries and QuestField results

Queries defined in an SQL Content Query configuration file can return various numbers of columns. The QuestFields Server distinguishes between queries that return one, two or three and above columns:

- The QuestFields server assumes that a query that returns only one column contains only a value to be sent to a QuestFields client.
- The QuestFields server assumes that a query that returns two columns contains a key in the first column and a value in the second column. A QuestFields client may use the key to uniquely identify records (for example: when submitting values to another web application).



- The QuestFields server assumes that a query that returns three or more columns contains a key in the first column, a value in the second column and additional metadata about the record returned in columns three and above. A QuestFields client may use the metadata for customer specific purposes.

### 3.1.1 Content Access Module settings

**camId** *Use:* Instance ID of the JDBC Content Access Module the channel uses.  
*Value type:* String.

### 3.1.2 SQL channel query settings queries

**query** *Use:* The SQL query to be performed for the Content Query. The part of the query that needs to be filled in by the server is defined within @ characters. This can part may start or finish with the % for wildcard character. The word between the @ characters (and excluding any possible wildcard character) is seen as a key for which the Channel needs to fill with a value for the Content Query to execute. See the QuestFields Server Administration guide for more information. Multiple keys are allowed.

*Value type:* String or CDATA block representing an SQL select statement.

**sortedResults** *Use:* To tell the QuestFields Server that the returned results are sorted.

*Value type:* String that either contains `true` or `false`.



# [4]

## Glossary

<b>CAM</b>	A Content Access Module (CAM) provides a standardized mechanism to link the QuestFields system to a Data store. A CAM is the “middleware” between the QuestFields system and the data stores it uses. Different Content Access Modules are needed to communicate with JDBC-compliant databases and LDAP-compliant directory servers.
<b>Content Query</b>	A Content Query contains a specific query that can be executed by a Channel.
<b>JDBC</b>	Java Database Connectivity (JDBC) is a standard application program interface specification for connecting Java programs to the data in SQL databases.
<b>QuestField</b>	A user interface element that sends queries to, and receives results from the QuestFields system.