

Resolving fn:collection() Errors

DataDirect XQuery™

Introduction

No standard way exists to access a relational table in XQuery, so each XQuery implementation defines its own way to do this. DataDirect XQuery uses fn:collection() to access a relational table. For example, the following query accesses the HOLDINGS database table:

```
collection('HOLDINGS')
```

Generally, DataDirect XQuery processes fn:collection() arguments in XQuery queries without problems if the fn:collection() argument is correct and the connection information specified in the connection associated with the XQuery query is correct. When DataDirect XQuery cannot resolve the fn:collection() argument to a specific database object, it raises an error.

This paper provides guidelines to help you troubleshoot and correct common causes of errors that occur when using fn:collection() in an XQuery query.

Guidelines for Resolving Errors

When DataDirect XQuery cannot resolve the fn:collection() argument to a specific database object, it raises an error, such as `Collection not found` or `Multiple collections found`. If you encounter these types of errors when using DataDirect XQuery, the following guidelines will help you troubleshoot and correct the cause of the error:

- Qualify table names in fn:collection() arguments if you have multiple database tables with the same name or the default catalog and schema associated with the connection do not provide access to the database table.
- Escape special characters in catalog, schema, and table names.
- Verify connections associated with the XQuery query.

Qualifying Table Names

If you have multiple database tables with the same name or the default catalog and schema associated with the connection do not provide access to the database table, you can qualify the database table name in the fn:collection() argument to target the specific table. For example, if the

default catalog (database) and schema (user) associated with the connection is FINANCIAL and JOSEPH, respectively, and the target table is owned by the schema MARY, qualify the table name in the fn:collection() argument:

```
collection('FINANCIAL.MARY.HOLDINGS')
```

Using Catalog and Schema Names

To verify that you know the correct catalog name, schema name, and table name, start the SQL tool shipped with your database and connect to the database server. Once connected, execute the following SQL statement against the database:

For DB2, Microsoft SQL Server, and Sybase:

```
SELECT * FROM "catalog". "schema". "table" WHERE 1=0
```

For Oracle:

```
SELECT * FROM "schema". "table" WHERE 1=0
```

where *catalog*, *schema*, and *table* are the catalog name, schema name, and table name of the database object you are trying to access.

NOTE: Oracle databases do not have catalogs.

If the SQL statement returns an empty result, the values you entered correspond to the correct catalog name, schema name, and table name. Use these values in the fn:collection() argument to qualify the table name. If the SQL statement returns an error, the values you entered are incorrect.

IMPORTANT: The case of the values specified in the fn:collection() argument must match the case of the database.

On Microsoft SQL Server and Sybase, a user can have the special status of database owner. For example, if the SQL name of the target table is "FINANCIAL"."dbo"."HOLDINGS," qualify the table name in the fn:collection() argument with the schema name dbo:

```
collection('FINANCIAL.dbo.HOLDINGS')
```

Using JDBC Connection Names

If the table name in the fn:collection() argument is qualified with a catalog name or schema name (or both) and DataDirect XQuery returns an error indicating that multiple tables with the same name exist, you need to qualify the table name using a JDBC connection name. A JDBC connection name identifies a specific connection associated with the database table.

Suppose two tables of the same name, HOLDINGS, exist on different database servers with the same schema name, JOSEPH. In this case, the following query does not provide enough information for DataDirect XQuery to locate the target table:

```
collection('JOSEPH.HOLDINGS')
```

To identify the correct table, you can qualify the table name in the `fn:collection()` argument with a JDBC connection name. Here's an example that shows a connection made explicitly in the application to two different databases; each connection is assigned a unique JDBC connection name, `stocks1` and `stocks2`, respectively:

```
DDXQJDBCConnection jc1 = new DDXQJDBCConnection();
jc1.setUrl("jdbc:xquery:sqlserver://server1:1433;databaseName=FINANCIAL");
jc1.setName("stocks1");
DDXQJDBCConnection jc2 = new DDXQJDBCConnection();
jc2.setUrl("jdbc:xquery:oracle://server2:1433;SID=ORCL");
jc2.setName("stocks2");
DDXQDataSource ds = new DDXQDataSource();
ds.setDDXQJDBCConnection(new DDXQJDBCConnection[] {jc1,jc2});
XQConnection conn = ds.getConnection("myuserid","mypasswd");
```

Here's an example showing the same connection information configured in a DataDirect XQuery source configuration file:

```
...
<JDBCConnection name="stocks1">
  <description>connection to stocks1 data</description>
  <url>jdbc:xquery:sqlserver://localhost:1433;DatabaseName=FINANCIAL</url>
  <user>myuserid</user>
  <password>mypasswd</password>
  ...
  <catalog name="FINANCIAL">
    <schema name="JOSEPH">
      <table name="HOLDINGS">
<JDBCConnection name="stocks2">
  <description>connection to stocks2 data</description>
  <url>jdbc:xquery:oracle://localhost:1433;SID=ORCL</url>
  <user>myuserid</user>
  <password>mypasswd</password>
  ...
  <schema name="JOSEPH">
    <table name="HOLDINGS">
...

```

To target the `HOLDINGS` table on `server1`, qualify the table name with the JDBC connection name `stocks1` in addition to the catalog name and schema name:

```
collection('stocks1:JOSEPH.HOLDINGS')
```

Refer to the [DataDirect XQuery User's Guide and Reference](#) for more information about connecting with DataDirect XQuery, including specifying JDBC connection names.

Escaping Special Characters

If the catalog name, schema name, or table name in the `fn:collection()` argument contains a period (.) or a colon (:), escape the character with a backslash (\) so that DataDirect XQuery can parse the argument into its different parts. For example, if the target table is named `A.HOLDINGS` and you specify the following query, DataDirect XQuery parses 'A' as the schema name, not as part of the table name:

```
collection('A.HOLDINGS')
```

Escaping the period (.) in the fn:collection() argument using the backslash character allows DataDirect XQuery to parse the argument correctly:

```
collection('A\ .HOLDINGS')
```

In addition, XQuery string literal syntax applies to the fn:collection() argument. If a table name contains double quotation marks, for example, A"HOLDINGS, and the fn:collection() argument uses double quotation mark delimiters, you must repeat the double quotation marks:

```
collection("A" "HOLDINGS")
```

Verifying Connections

Using DataDirect XQuery, an application establishes a connection to the database to execute an XQuery query. The application can establish a connection to the database in multiple ways: explicitly specifying connection information in the application, using a data source registered with JNDI, or using a DataDirect XQuery source configuration file. If DataDirect XQuery cannot access the database because connection information is specified incorrectly or because the structure of the configuration file is incorrect, it raises an error.

Verifying Connection URLs

Verify that the following information in your connection URL is correct:

- Type of database to which the application is connecting.
- TCP/IP address or host name of the database server to which the application is connecting.
- Number of the TCP/IP port.
- User name used to connect to the database.
- Password used to connect to the database.
- Database-specific connection properties that provide additional connection information. DatabaseName (for DB2 and Microsoft SQL Server) and SID (for Oracle) are commonly used properties.

Verifying DataDirect XQuery Source Configuration Files

Minor differences in the structure of the configuration file are required depending on the database accessed. If you are using a configuration file, make sure that it validates against the source_config.xsd schema shipped with DataDirect XQuery. This schema is located in the examples/config subdirectory of your DataDirect XQuery installation directory. You can validate the configuration file using a tool such as <oxygen>® XML Editor for Eclipse (DataDirect XQuery Edition) or Stylus Studio®.

In addition, make sure that the values of the following elements, if specified in your configuration file, are correct:

- catalog
- schema
- table

Checklist

If you encounter an error when using `fn:collection()` with DataDirect XQuery, examine the following checklist to resolve the problem:

- ✓ Qualify table names in `fn:collection()` arguments if you have multiple database tables with the same name or the default catalog and schema associated with the connection do not provide access to the database table.
 - Make sure that you know the correct catalog name, schema name, and table name (including case).
 - If you are accessing Microsoft SQL Server or Sybase and the database table is owned by `dbo`, make sure that you qualify the table name with the schema name `dbo`. For example:

```
collection('FINANCIAL.dbo.HOLDINGS')
```
 - If you qualify the table name with a catalog name or schema name (or both) and DataDirect XQuery returns an error indicating that multiple collections are found, you may want to qualify the table name with a JDBC connection name.
- ✓ Escape special characters in catalog, schema, and table names.
- ✓ Verify connections associated with the XQuery query.
 - Make sure that the information specified in your connection URL is correct including: database type, server name, port, user, password, and any database-specific connection properties.
 - If using a DataDirect XQuery source configuration file, make sure that it validates against the `source_config.xsd` schema shipped with DataDirect XQuery. This schema is located in the `examples/config` subdirectory of your DataDirect XQuery installation directory. You can validate the configuration file using a tool such as <oxyen> XML Editor for Eclipse (DataDirect XQuery Edition) or Stylus Studio.
 - If using a DataDirect XQuery source configuration file, make sure that the values of the catalog, schema, and table elements are correct

We welcome your feedback! Please send any comments concerning documentation, including suggestions for other topics that you would like to see, to:

docgroup@datadirect.com

FOR MORE INFORMATION

800-876-3101**Worldwide Sales**

Belgium (French).....	0800 12 045
Belgium (Dutch).....	0800 12 046
France	0800 911 454
Germany	0800 181 78 76
Japan	0120.20.9613
Netherlands	0800 022 0524
United Kingdom	0800 169 19 07
United States	800 876 3101

Copyright © 2005 DataDirect Technologies Corp. All rights reserved. DataDirect Connect is a registered trademark of DataDirect Technologies Corp. in the United States and other countries. DataDirect XQuery is a trademark of DataDirect Technologies Corp. in the U.S. and other countries. Java and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Other company or product names mentioned herein may be trademarks or registered trademarks of their respective companies.



DataDirect Technologies is focused on standards-based data connectivity, enabling software developers to quickly develop and deploy business applications across all major databases and platforms. DataDirect Technologies offers the most comprehensive, proven line of data connectivity components available anywhere. Developers worldwide at more than 250 leading independent software vendors and thousands of corporate IT departments rely on DataDirect® products to connect their applications to an unparalleled range of data sources using standards-based interfaces such as ODBC, JDBC™ and ADO.NET. Developers also depend on DataDirect to radically simplify complex data integration projects using XML products based on the emerging XQuery and XQJ standards. DataDirect Technologies is an operating company of Progress Software Corporation (Nasdaq: PRGS), a US\$300+ million global software industry leader. Headquartered in Bedford, Mass., DataDirect Technologies can be reached on the Web at <http://www.datadirect.com> or by phone at +1-800-876-3101.