FAQ for HXTT XML Packages of type 4 JDBC Driver for XML file

The most recent version of this document can be viewed at [here](#).

Table of Contents

1. [General Questions](#)
2. [Applet Questions](#)
3. [Remote Access Questions and Client/Server Mode Questions](#)
4. [SQL Questions](#)
5. [Performance Questions](#)
6. [Concurrence Questions](#)
7. [Internationalization Questions](#)
8. [Interoperability Questions](#)

General Questions

1. How to know the detailed version information of HXTT XML package?
1st way: "java com.hxtt.sql.xml.XMLDriver" will print that information.
2nd way: check that MANIFEST.MF file in jar file.

2. Can I use it in an iSeries OS/400 IBM machine that has Java 1.4 running in it?
The HXTT XML packages can run on any platform with Java VM, which includes Microsoft Windows, Novell Netware, OS2, UNIX, and LINUX. It supports Personal Java, JDK1.0.X, JDK1.1.X, JDK1.2.X, JDK1.3.X, JDK1.4.X, JDK1.5.X, JDK1.6.X, JDK1.7.X, and JDK1.8.X. It supports JDBC1.2, JDBC2.0, JDBC3.0, JDBC4.0, JDBC4.1, and JDBC4.2 now.

3. What is difference between the HXTT XML Package, Embedded Package, and Remote Access Package? Can I get some sample code to use the HXTT XML?
The HXTT XML supports Embedded and Remote Access. HXTT XML Package includes a Database GUI manager. If you're accessing the local data, you can use the HXTT XML Package or Embedded Package. If you're accessing the remote data, you can use the HXTT XML Package or Remote Access Package. There is no any difference for your code to use anyone of three packages. Please download the demo package from [here](#).

4. What causes the 'No suitable driver' SQLException?
This error usually occurs during a call to DriverManager.getConnection(). The cause can be failing to load the appropriate JDBC driver before calling getConnection(), or specifying an invalid JDBC URL that isn't recognized by your JDBC driver. If you're using a trial version, you will get "No suitable driver" SQLException, and "Evaluation period over" after using about 30 days. The HXTT XML driver's name is com.hxtt.sql.xml.XMLDriver, and its JDBC URL:

```
    Embedded:
        jdbc:xml:[//]/[DatabasePath][?prop1=value1[;prop2=value2]] (You can omit that "//" characters sometimes)
            For example:
                "jdbc:xml:/."
                "jdbc:xml:/c:/data"
                "jdbc:xml:////usr/data" for unix or linux:
                "jdbc:xml:/./data"
    Access by XML Server: Skip it if you don't use TCP, RMI or JINI.
        jdbc:xml://host:port/[DatabasePath]
            For example: "jdbc:xml://domain.com:3099/c:/data" if one XMLServer is run on the 3099 port of domain.com
```

5. How to setup XML url on the Novell Server?
XML driver can run on Novell server. You can use directly access or XMLServer to visit your data on Novell server. If your XML files is at sys:/java/yourdata, the direct URL should be:
jdbc:xml:///sys:/java/yourdata
or

jdbc:xml:////java/yourdata

6. I got "java.io.IOException: Permission denied" sometimes for my SELECT query.
Please figure out what directory Java's java.io.tmpdir system property points to, and make sure that directory is writable by the user that runs your Java applications, otherwise you should set tmpdir properity in Connection property to a writable directory. tmpdir properity indicates whether set a temp directory, Default: the value of JVM's "java.io.tmpdir" property. If that value is incorrect, uing the directory of JDBC url.

7. When I used jdbc:xml:/<DatabasePath>, the connection's schema was empty. "create catalog if not exists xmlfiles". What is Catalog?
XML's schema is always empty. You can use catalog to query subdirectory. Catalog means a directory, which contains some XML files.

8. Can HXTT XML support JDK 1.0.2?
Yeah. You need to download JDBC 1.22 from the Sun's JDBC download page and add JDBC1.22 into JDK 1.0.2. HXTT XML hasn't be tested on JDK1.0.X since we have not received such a complement request from our users. If you meet any problem, please let us know.

Applet Questions

1. I already configured the .java.policy for my applet, but I continue with problems of "acess denied".
For instance, you're using "jdbc:xml:/C:/test", and grant codeBase "file:/C:/test" in your policy file, but your applet is running from "D:\sample\CargaStatApplet.html". You should grant codeBase "file:/D:/sample", not "file:/c:/test".

2. http://localhost:8080/jdbcapplet.html, the applet started but returns a Classnotfound com.hxtt.sql.xml.XMLDriver error in the gui list.
Please add a codebase tag. For instance, "<applet code="jdbcapplet.class" codebase="XML_Remote_Access_JDBC40.jar"></applet>". The XML_Remote_Access_JDBC40.jar should be at the same directory of jdbcapplet.html.

Remote Access Questions and Client/Server Mode Questions

1. Client/Server mode question: The data directory is not in the IBM machine where the Java program should run, but instead those XML files are in another machine with Windows operating system.
com.hxtt.sql.admin.Admin provides a GUI manger for XML server. For instance, you wish to provide JDBC4.0 remote data access. Please use "java -cp yourdirectrory/XML_JDBC40.jar com.hxtt.sql.admin.Admin" to start GUI manager, and add a url setting of '"jdbc:xml://10.32.90.48:" + 8029 +"/"+databaseDirectory' on your host of 10.32.90.48(just an IP sample), then click Start button. Third, you can use 'String url = "jdbc:xml://10.32.90.48:" + 8029 +"/"+databaseDirectory;' to visit your XML database from your IBM machine. If you're running that GUI manager on "yourNT.com" host to visit "c:/database" directory, you can use "jdbc:xml://yourNT.com:8029/c:/database" on your web application. jdbc:xml://yourNT.com:8029/c:/database? user=oneuser&password=onepassword can provide a simply user/password verification for client/server mode. If you wish to write a secure XML server for some sensitive information, embedded encrypt/decrypt functions can help you.

2. Remote access through map network drive question: How to remote access XML data without XMLServer?
You can share your remote directory which contains your data files, then map it to a local driver.
For Windows: You can connect remote XML database by sharing the directory and map it to local drive. You should disable the OPLOCKS of your Samba/NT/2000 server. How to turn off opportunistic locking in windows on client and on server side? This is done by manipulating the following registry key:

For client side: oplocks.reg file
REGEDIT4
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MRXSmb\Parameters]
"OplocksDisabled"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters]
"UseOpportunisticLocking"=dword:00000000

For server side: oplocksServer.reg
REGEDIT4
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters]
"EnableOplocks"=dword:00000000

You should put additional registry entry like e.g.
The following registry entries disable opportunistic locking and caching:
HKey_Local_Machine \ System \ CurrentControlSet \ Services \ LanmanServer \ Parameters \ EnableOpLocks must be set to 0.
HKey_Local_Machine \ System \ CurrentControlSet \ Services \ LanmanServer \ Parameters \ CachedOpenLimit must be set to 0.
HKey_Local_Machine \ System \ CurrentControlSet \ Services \ LanmanWorkStation \ Parameters \ UseOpportunisticLocking must be set to 0.
HKey_Local_Machine \ System \ CurrentControlSet \ Services \ LanmanWorkStation \ Parameters \ UtilizeNtCaching must be set to 0.

For Linux: You can use mounting. One user uses Samba to maped NTFS partitions in Linux servers, and XML driver works normally like mapping any mount point in Linux.
For Novell: You can map NCP directory as driver or mount NCP directory.

3. Remote access through UNC path question: Can I setup only one datasource to access four servers for my Cold Fusion?

To access one unc path, you can use jdbc:xml:/\\PC17\c$\values or jdbc:xml:/\\PC17\val.
To access four unc pathes in the same connection, you need to use a free JDBC url, "jdbc:xml:/" or "jdbc:xml:///". Then you can use some full UNC path names in SQL to visit your four servers where your Java VM has right to access.. For instance:

```
select * from \\amd2500\e$\xmlfiles\test;
select * from "\\amd2500\d$\xmliles".test;
select * from ".".test;
```

4. Remote access through http/https/ftp protocol question: How to let my program to fetch data daily from our web host?

You need to use url database, which supports http protocol, https protocol, ftp protocol, and sftp protocol.

5. Remote access through SAMBA protocol question: How to let my servlet on Linux to access over 300 hundred shared folders that all are on Windows boxes

You need to use SAMBA table, which needn't to map or mount driver.

6. I can't get the com.hxtt.sql.admin.Admin runnig for internet --> intranet

HXTT XML supports port mapping and NAT route. Let HXTT XML listening a port on the database server, and modify your route table or NAT table to map an external port to that internal port. You can use "start java -Djava.security.policy=policy com.hxtt.sql.admin.Admin" to start GUI manager. You should add a remote url, for instance, jdbc:xml://localhost:8029/d:/xmlfiles, and click Start button to start that server. Then on your internet client side, you can use jdbc:xml://externalIP:8029/d:/xmlfiles to access your intranet host. externalIP means an external IP or domain name address of your gateway or database server.
BTW, except for TCPServer protocol, HXTT XML can use also RMIServer protocol. For instance, you have used "start rmiregistry 1099 -J-Djava.security.policy=yourPolicyFile" to startup your rmi service. Then you can use jdbc:xml://localhost:1099/d:/xmlfiles?serverType=RMIServer to let HXTT XML bind remote service in registry. The key is use "java -Djava.security.policy=policy -Djava.rmi.server.hostname=externalIP com.hxtt.sql.admin.Admin RMISERVER 8029" to start your server. RMIServer protocol is slower much than the default TCPServer protcol.

7. I would like to start a server (TCP) from our application, instead of DBAdmin. I need to be able to programmatically tell the application which profile to start.

Please read Start/Stop Server Programmatically.

8. Is there a way to specify a file path in the url that will connect to a mapped drive in Windows 2000. ie drive \\gomer\pyle\db which is mapped to f drive on the server.

XML driver can work with mapped driver, and you should use "jdbc:xml:/f:" to access your data.

Note: If you're using a database file through a UNC path or a mapped drive of Windows, there is a Windows Security restriction. If you run ColdFusion (Tomcat, or tanuki sw wrapper) as a service on Windows, it operates by default as System, and cannot access directories on a remote system or mapped drive; to resolve this issue, do not run ColdFusion (Tomcat, or tanuki sw wrapper) using the local system account.

9. When I click Start button to start a remote service, I get a security excaption: access denied (java.net.SocketPermission 127.0.0.1:8029 connect,resolve)
You have to enable java.net.SocketPermission right in your policy file if you
run a XML server. Please read
file:///yourdriver|/jdk1.6/docs/guide/security/PolicyFiles.html for more
information about policy file. It is unnecessary to know the specific content
of a policy file, since you can use policy tool to create and maintain your
policy files. Please read
file:///yourdriver|/jdk1.6/docs/tooldocs/win32/policytool.html for policy tool.

10. How to start remote service as MS Windows service and Linux(Solaris) Daemon?
Please read [Run HXTT XMLServer as Windows Service or Linux(Solaris) Daemon](#).

11. How to start remote control when XMLServer is running as Windows service or Linux(Solaris) Daemon?
You can use "java com.hxtt.sql.admin.Admin TCPCLIENT [host:]port [remoteControlPassword]" to start your remote
control.

SQL Questions

1. I need to use tables stored in a subdirectory.
table-name: [catalog.]tableName
For instance, you have many XML files on c:\data. You can use "jdbc:xml:/c:/data" as JDBC url. Then you can use
"select * from subdirectory1.table1" to visit table1 file at subdirectory1. For instance, "select tableAlias.* from
"sales/2004/04".sale as tableAlias" can access sale table at "c:\data\sales\2004\04".

2. I can't use "select RIGHT from deldob"
RIGHT is a reserved SQL keyword. "variableName", [variableName] or {v 'variableName'} is used to quote those
columns which use reserved keyword, so that you should use "RIGHT" or {v 'RIGHT'} to quote the RIGHT field, for
instance, *select {v 'RIGHT'},'other' from states where "RIGHT"=32*. HXTT XML supports using DATE, TIME,
TIMESTAMP, GROUP, ORDER, KEY, DESC, UPDATE directly in SQL, although they're reserved words too.

**3. Can I get an example on how to do a query involving a boolean value. eg. " Select * from tableName where
exported = true", where exported is a boolean column in a XML file.**
Supports. You can use "select * from tableName where exported" too. All of NOT, AND, and OR operation are
supported.

4. How to specify dates?
Please use SQL Escape Syntax, a date is specified in a JDBC SQL statement with the syntax {d `yyyy-mm-dd'} where
yyyy-mm-dd provides the year, month, and date, e.g. 1996-02-28. There are analogous escape clauses for TIME and
TIMESTAMP type: {t `hh:mm:ss'} and {ts `yyyy-mm-dd hh:mm:ss.f...'}. The fractional seconds (.f...) portion of the
TIMESTAMP can be omitted. For instance, {d '1999-11-01'} and {ts '3999-03-24 00:59:23.22222'}. You can use
PreparedStatement.setDate to set date columns too.

**5. How to handle date range selection, e.g. SELECT * FROM CALLS WHERE START >= '2001-01-01' AND
END <= '2002-01-01'**
Although the HXTT XML supports "SELECT * FROM CALLS WHERE START >= '2001-01-01' AND END <=
'2002-01-01'", but that sql syntax is unadvisable. Please use SQL Escape Syntax, {d `yyyy-mm-dd'} and {ts `yyyy-mm-
dd hh:mm:ss.f...'}, for Date and timestamp type according to JDBC standard. You can learn more about Escape Syntax
at file:///yourdriver|/jdk1.6/docs/guide/jdbc/spec/jdbc-spec.frame11.html . You should use "select * from calls where
start>={d '2001-01-01'} and end <={d '2002-01-01'}".

6. Can {d '2999-11-21'}={ts '2999-11-21 23:22:20.3335'} and {t '23:22:20'}={ts '1999-01-01 23:22:20.333'}?
 Supports.

7. I think this one is for use functions {fn abs(TEST.int1)}
 You can use abs(TEST.int1) too. HXTT XML supports more than 230 functions.

8. Update table_name set (fieldname1=X, fieldname2=X2, ...) where primary_index='blah' throws a parse exception.
 You should use "update table_name set fieldname1=X, fieldname2=X2, ... where primary_index='blah'".

9. How to delete all deleted records permanently?
 "PACK TABLE [IF EXISTS] table_name" will pack database.
 "TRUNCATE TABLE [IF EXISTS] table-name" will zap database.

Performance Questions

1. What is the most efficient method to insert records in a table, to use an updatable RecordSet or to use a PreparedStatement?
 PreparedStatement is smally quicker than updatable RecordSet. An updatable RecordSet is quicker than PreparedStatement if you insert into more than 200 columns with constant values. It can only cope with constant values. PreparedStatement can cope with complicated expressions so that you can insert timestamp, function, ResultSet, and so on.

2. "select count(*) from table" are worked a long time for large tables.
 You should use "select reccount() from table" to get the number of records. Count(*) sums always up all records except deleted row.

3. Are there any data row count, data volume, memory minimums, maximums imposed when using the HXTT XML?
 No limitation. The HXTT XML supports to join query big databases with DISTINCT, GROUP BY, and ORDER BY.

4. > Is there a scenario possible where we store frequently accessed temporary data in memory-only tables (as a cache) and use these for query's*_mixed _*with real HXTT XML tables?
 memory is a speical catalog name for memory-only database. Through _memory_ catalog, memory-only database is visible for all applications in the same JVM. For instance, in an embedded connection, you can use create table _memory_.abc (a char(10));insert into _memory_.abc values(333);select * from _memory_.abc, test where abc.a=test.b;drop table _memory_.abc; to do the same things.

Concurrence Questions

1. Does HXTT XML support multi-user access?
 The HXTT XML supports multi-user access, record lock, and table lock.

2. Is there any way to lock/unlock record programatically.
 We have provided a _LockFlag_ virtual column as row lock flag. You can know it from Set Record Lock Manually.

Internationalization Questions

1. Can the HXTT XML support Czech MS - DOS 895?

The HXTT XML supports all codepage, multilingual collation sequence, and unicode character set. Cp895(Czech MS - DOS 895), Cp620(Polish MS - DOS 620) and Mazovia are extra supported although JVM doesn't support those.

2. Do you have a solution for character translation to the right encoding?

The HXTT XML supports CharacterEncoding. Please use charSet property.

```
//Default: null
//You can find a Supported Encodings list of files:///yourdriver/jdk1.2/docs/guide/internat/encoding.doc.html
//Extra supports:
// Cp895 is supported by HXTT XML driver. //Czech MS - DOS 895
// Cp620 is supported by HXTT XML driver. //Polish MS - DOS 620
// Mazovia is supported by HXTT XML driver. //Polish
Properties properties=new Properties();
properties.setProperty("charSet","sv_SE");
Connection con = DriverManager.getConnection(url,properties);
```

3. While reading encrypted data in a XML file using u'r parser in java. The data retreived is different from the data in the XML file, certain characters are read as ? marks.(the encryption is done using ASCII values).
You can use ResultSet.getBytes(int columnIndex), not ResultSet.getString(int columnIndex) and ResultSet.getObject(int columnIndex), to get your encrypted data, since your encrypted data is binary stream.

4. When they insert accented characters, it comes out different at the Java end. There seem to be some character set conversion problems. Is there a way to solve that?
You can use ResultSet.getBytes() and ResultSet.setBytes() to avoid CharacterEncoding.

Interoperability Questions

1. How to set up HXTT XML with Tomcat4.1 as PoolableConnection?
This sample shows three PoolableConnections ways through Database Connection Pool (DBCP) Configurations and JNDI Resources(You should read JNDI Datasource HOW-TO and JNDI Resources HOW-TO also.):
In server.xml:

```
<Context path="" docBase="ROOT" debug="5" reloadable="true" crossContext="true">

    <Resource name="jdbc/testxmlPool1" auth="Container" type="javax.sql.DataSource"/>
      <ResourceParams name="jdbc/testXMLPool1">
            <parameter>
                <name>factory</name>
                <value>org.apache.commons.dbcp.BasicDataSourceFactory</value>
            </parameter>

            <parameter>
                <name>maxActive</name>
                <value>50</value>
            </parameter>

            <parameter>
                <name>maxIdle</name>
                <value>10</value>
            </parameter>

            <parameter>
                <name>maxWait</name>
                <value>10000</value>
            </parameter>

            <parameter>
                <name>username</name>
                <value></value>
            </parameter>

            <parameter>
                <name>password</name>
                <value></value>
            </parameter>

            <parameter>
                <name>driverClassName</name>
                <value>com.hxtt.sql.xml.XMLDriver</value>
            </parameter>

            <parameter>
                <name>url</name>
                <value>jdbc:xml:///d:/xmlfiles</value>
            </parameter>
    </ResourceParams>
```

```xml
<Resource name="jdbc/testXMLPool2" auth="Container" type="com.hxtt.sql.HxttConnectionPoolDataSource"/>
    <ResourceParams name="jdbc/testXMLPool2">
        <parameter>
            <name>factory</name>
            <value>org.apache.naming.factory.BeanFactory</value>
        </parameter>

        <parameter>
            <name>url</name>
            <value>jdbc:xml:///d:/xmlfiles</value>
        </parameter>

        <parameter><name>username</name><value></value></parameter>
        <parameter><name>password</name><value></value></parameter>
        <parameter><name>host</name><value></value></parameter>
        <parameter><name>port</name><value>8029</value></parameter>

    </ResourceParams>


<Resource name="jdbc/testXMLPool3" auth="Container" type="com.hxtt.sql.HxttConnectionPoolDataSource"/>
    <ResourceParams name="jdbc/testXMLPool3">
        <parameter>
            <name>factory</name>
            <value>com.hxtt.sql.HxttObjectFactory</value>
        </parameter>
        <parameter>
            <name>url</name>
            <value>jdbc:xml:///d:/xmlfiles</value>
        </parameter>

        <parameter><name>username</name><value></value></parameter>
        <parameter><name>password</name><value></value></parameter>
        <parameter><name>host</name><value></value></parameter>
        <parameter><name>port</name><value>8029</value></parameter>
    </ResourceParams>

</Context>
```

Then you can use the below code to test those PoolableConnections:

```java
Context initContext = new InitialContext();
Context envContext = (Context)initContext.lookup("java:/comp/env");

DataSource ds1 = (DataSource)envContext.lookup("jdbc/testXMLPool1");
Connection conn1 = ds1.getConnection();
out.println("testXMLPool1 OK:)<br/>");
Statement stmt1 = conn1.createStatement();
ResultSet rs1 = stmt1.executeQuery("select * from test");
if(rs1.next())
    out.println(rs1.getString(1)+":)<br/>");
rs1.close();
stmt1.close();
conn1.close();


DataSource ds2 = (DataSource)envContext.lookup("jdbc/testXMLPool2");
Connection conn2 = ds2.getConnection();
out.println("testXMLPool2 OK:)<br/>");
Statement stmt2 = conn2.createStatement();
ResultSet rs2 = stmt2.executeQuery("select * from test");
if(rs2.next())
    out.println(rs2.getString(1)+":)<br/>");
rs2.close();
stmt2.close();
conn2.close();

DataSource ds3 = (DataSource)envContext.lookup("jdbc/testXMLPool3");
Connection conn3 = ds3.getConnection();
out.println("testXMLPool3 OK:)<br/>");
Statement stmt3 = conn3.createStatement();
ResultSet rs3 = stmt3.executeQuery("select * from test");
if(rs3.next())
    out.println(rs3.getString(1)+":)<br/>");
rs3.close();
stmt3.close();
conn3.close();
```

If you use org.apache.commons.dbcp.BasicDataSource, but get "Cannot create PoolableConnectionFactory" Error, you should check your commons-pool-1.x.jar and commons-dbcp-1.*.jar file in $TOMCAT/common/lib directory to see whether two files have the same version. DBCP v1.2 requires Pool v1.2 so that you should update Pool v1.1 from the tomcat website.

If you wish to add more Connection property, you should use connectionProperties, for instance:

```
<parameter>
<name>connectionProperties</name>
<value>charSet=Cp737</value>
</parameter>
```

2. How to set up HXTT XML with vqServer 1.9.55 as web server?
 The key is to use an absolute path as Java libraries' location, and restart vqServer after modified Java libraries.
 For instance, your vqServer is installed at C:\vqServer\.
 1. Please use http://yourhost:9090/ to visit your administration server.
 2. Click on Java libraries in the vqServer control centre menu (http://yourhost:9090/admin?action=libraries&serial=14)
 3 Click New library (http://yourhost:9090/admin?lib=New_library&action=edit)
 4. Enter C:\vqServer\classes\XML_JDBC20.jar as location value, XML Driver as Description value, then click OK button.
 5. Please copy XML_JDBC20.jar into C:\vqServer\classes directory.
 6. Please copy ex01.class into C:\vqServer\servlets\servlets
 7. Stop and restart vqServer
 8. Please use http://yourhost/servlet/yourServlets to get your result.

3. How to set up HXTT XML with Coldfusion MX 6.1 Application Server?
 For instance,your Coldfusion MX is installed at C:\CFusionMX\, and wish to use XML_JDBC30.jar.
 1. Please copy XML_JDBC30.jar into C:/CFusionMX/wwwroot/WEB-INF/classes/.
 2. Use http://yourhost:8500/CFIDE/administrator/index.cfm to enter the CFMX Administrator.
 3. Go to the "Java and JVM" of Server Settings, http://yourhost:8500/CFIDE/administrator/settings/jvm.cfm page, and enter the full path, C:/CFusionMX/wwwroot/WEB-INF/classes/XML_JDBC30.jar, in the Class Path. Then, click "Submit Changes".
 4. Restart the CFMX Service.
 5. Please go back to the administrator page, and go to the "Data Sources" of Data & Services, http://yourhost:8500/CFIDE/administrator/datasources/index.cfm page, and enter the name for the new datasource, for instance "XMLTest", and select "Other" for the driver. Then Click "Add".
 6. Enter the datasource information. JDBC URL is always in the format jdbc:xml:[//[host:port]]/[DatabasePath], for instance jdbc:xml:/c:/data. You can set more connection properties in your JDBC URL, for instance: jdbc:xml:/c:data?delayedClose=15;maxCacheSize=6144;lockTimeout=2000; . Driver class is always com.hxtt.sql.xml.XMLDriver. Driver name is used to identify the driver in the datasources view, and you can use XML. Username and password are not required. They can also be specified in the cfquery tag (but datasource verification will fail if you don't enter them). Description is not required.
 7. Lastly, please press "Submit" to finalize the entered data.
 8. You can find edit.cfm and edit_action.cfm sample in demo pacakge.

4. HXTT XML with If you run ColdFusion (Tomcat, or alexandria sw and tanuki sw wrapper) on Windows 2000 and Windows XP Pro does not work on mapped drives.

Note: If you're using a database file through a UNC path or a mapped drive of Windows, there is a Windows Security restriction. If you run ColdFusion (Tomcat, or tanuki sw wrapper) as a service on Windows, it operates by default as System, and cannot access directories on a remote system or mapped drive; to resolve this issue, do not run ColdFusion (Tomcat, or tanuki sw wrapper) using the local system account.

The service(For instance, ColdFusion MX Application Server, ColdFusion MX 7 Application Server, or Apache

Tomcat) built by ColdFusion (Tomcat, or tanuki sw wrapper) can not access the share directory at other machine by default. But you can do as follows to solve this problem:
1. Right click the service built by ColdFusion (Tomcat, or tanuki sw wrapper) in service manager, and click the property menu.
2. On the open window,select the login tab, click this account radio box, and click the browse button.
3. Select the administrator account(it seems that you should select the administrator account), input the correct password in the password textbox and confirm password textbox.
4. Restart this service, you can find this service can access the share directory at other machine.

5. How to resolve 'DataSet has no unique row identifiers.' issue in JBuilder's QueryDataSet?
You can use _rowid_, a virtual column to avoid that issue, For instance:

```
//...
queryDataSet = new QueryDataSet();
    //...
    queryDataSet.setMetaDataUpdate(MetaDataUpdate.ALL-MetaDataUpdate.ROWID-MetaDataUpdate.TABLENAME);

queryDataSet.setQuery(new QueryDescriptor(database, "select _rowid_,* from test", null, true,
    Load.ALL));
queryDataSet.open();

queryDataSet.setTableName("test");
queryDataSet.setRowId("_rowid_", true);
    //...
```

6. How to set HXTT XML with WebSphere Application Server?
You can download a pdf guide from here.

7. How to set HXTT XML with Hibernate?
You should download support package and sample from here.

8. How to set HXTT XML Data Source with Oracle Application Server 10G?
You should read guide at Oracle Application Server 10G(v10.1.3) and Oracle Application Server 10G(v10.1.2.02).

9. How to set HXTT XML Data Source with JBoss AS 7.0.2?
After downloading HXTT XML driver jar file. just move the HXTT XML driver jar file into Jboss 7 installation directory under standalone/deployments (jboss-as-web-7.0.2.Final/standalone/deployments).
Restart your jboss server
Then go to the url and type http://localhost:8080/
click Administration console.
In Jboss Administration console you can press Add button of Datasource Configurations. step 1/3 Datasource Attributes: enter datasource name(testxml) and jndi name(testxml) then click Next button. step 2/3 JDBC Driver: select HXTT XML driver jar file. click Next Nutton. Step 3 Connection Settings: enter connection URL (jdbc:xml:////data) , Username(null) and Password(null). finally click Done button.

10. How to set HXTT XML Data Source with Railo 3.3.1 Express?
After downloading HXTT XML driver jar file. just move the HXTT XML driver jar file into Railo 3 installation directory under /lib/ext/railo-server/context/library (railo-3.3.1.000//lib/ext/railo-server/context/library).
Restart your Railo server
Then go to the url and type http://localhost:8888/
click Railo Server Administrator (or type http://localhost:8888/railo-context/admin/server.cfm).
In Services - Datasource of Railo Server Administrator console, you can Create new datasource: step 1/2 Name: enter datasource name(testxml) and Type(Other - JDBC driver) then click create button. step 2/2 enter Class(com.hxtt.sql.xml.XMLDriver), DSN(jdbc:xml:////data), Username(null) and Password(null). finally click create button.

11. Where's JDBC Connection Pool Templates for GlassFish?
MS Access template is at http://www.hxtt.com/test/microsoft_access_type4_datasource.xml
Cobol data file template is at http://www.hxtt.com/test/microsoft_access_type4_datasource.xml

MS Excel template is at http://www.hxtt.com/test/microsoft_excel_type4_datasource.xml
Corel Paradox template is at http://www.hxtt.com/test/paradox_type4_datasource.xml
Xbase template is at http://www.hxtt.com/test/dbf_type4_datasource.xml
CSV template is at http://www.hxtt.com/test/csv_type4_datasource.xml
Text template is at http://www.hxtt.com/test/text_type4_datasource.xml
PDF template is at http://www.hxtt.com/test/pdf_type4_datasource.xml
XML template is at http://www.hxtt.com/test/xml_type4_datasource.xml

12. How to set HXTT XML database in DBeaver?

1. Create a JDBC Data Source for HXTT XML Data
Open the DBeaver application, in the Databases menu, click the Driver Manager item. Click New to open the Create New Driver form.
In the Driver Name box, enter a user-friendly name for the driver, for isntance, testXML.
In the Class Name box, enter com.hxtt.sql.xml.XMLDriver.
In the URL Template box, enter jdbc:xml: .
In the Category box, enter XML.
In the Libraries panel, click Add File so that load the driver JAR (for instance, XML_JDBC40.jar) in DBeaver.
In the Open driver library dialog that appears, select the XML_JDBC40.jar file.

2. Create a Connection to HXTT XML Data
In the Databases menu, click New Connection item.
In the Create new connection wizard that results, select XML, then testXML.
On the next page of the wizard, click the Driver properties tab.
Add new property, Property Name is database, Value is the path of your database file, for instance, d:/test/xml .
Click Finish button.

13. How to set HXTT XML with Apache EmpireDB?

You should download support package from here.

13. Does HXTT XML driver support XA connections ?

It provides an experimental com.hxtt.sql.HxttXADataSource .

14. How to set HXTT XML into my projects maven pom.xml.?

For instance,

Once you've downloaded the JAR just add it to your computer repository with:
mvn install:install-file -DgroupId=com.hxtt.sql.xml -DartifactId=xmljdbc4 \
    -Dversion=v2.0 -Dpackaging=jar -Dfile=XML_JDBC40.jar -DgeneratePom=true
The last parameter for generating a POM will save you from pom.xml warnings.

Include the new dependency by modifying your project��s pom.xml. Add the following dependency:
```
<dependency>
    <groupid>com.hxtt.sql.xml</groupid>
    <artifactid>xmljdbc4</artifactid>
    <version>v2.0</version>
</dependency>
```

Save the pom.xml file and build the project to make sure no errors exist.

15. How to set HXTT XML Data Source with JBoss Application Server 4.0.1?

For instance,

```
<datasources>
  <local-tx-datasource>
    <jndi-name>TestData</jndi-name>
    <connection-url>jdbc:xml:////data</connection-url>
    <driver-class>com.hxtt.sql.xml.XMLDriver</driver-class>
    <connection-property name="delayedClose">-1</connection-property>
    <user-name/>
    <password/>
    <min-pool-size>5</min-pool-size>
    <max-pool-size>20</max-pool-size>
```

```
      <idle-timeout-minutes>5</idle-timeout-minutes>
    </local-tx-datasource>
  </datasources>
```

16. How to set HXTT XML database dialect with EclipseLink of Glassfish?
You should download support package and sample from [here](#).

17. How to set HXTT XML database dialect with TopLink of Glassfish?
You should download support package and sample from [here](#).

18. How to set HXTT XML database in JasperSoft Studio?
1. right-click a project in the Project Explorer and select New > Data Adapter to open the Data Adapters Wizard.
2. In the DataAdapter File window, choose the project where you want to save the data adapter file. This should be the project that contains the report(s) you want to use with your data adapter.
3. Enter a file name for your adapter and click Next.
4. Select Database JDBC Connection as the data adapter typeand click Next.
5. Enter a name for your adapter. This name is used when you select an adapter for a report.
6. Enter com.hxtt.sql.xml.XMLDriver for JDBC Driver
7. Enter jdbc:xml:/yourDatabasePath for JDBC Driver
8. Click Driver Classpah tab.
9. Click Add button to select the XML_JDBC40.jar file.
10. Click Finish to create the adapter.
The adapter is saved as an XML file in the project location you selected.

19. How to set HXTT XML database in Jasper Server?
First, copy the Driver to the JasperReports Server Classpath
Now you need to copy the driver (the XML_JDBC40.jar file) to the classpath of your application server to enable JasperServer to find it.
In general it's best to copy the driver to the application server's shared library location. Refer to your application server documentation for exact locations.
For example in a JapserServer Bundle install on windows this is located at: C:/Jaspersoft/jasperreports-server-cp-7.1.0/apache-tomcat/lib
After these modifications you will need to restart the application server for this settings to take effect. For information on restarting your server see Chapter 3 of the JasperReport Server Install Guide (is located on the /docs folder of your Jasper Server install path)
1. Log into JasperReport Server from your browser using the jasperadmin or superuser account (e.g.: http://localhost:8080/jasperserver-pro)
2. Go to View | Repository, Select a folder where you want to create your Data Source (e.g.: /Data Sources). Right click on the folder name in the repository view and from the context menu select called Add Resource and then click on Data Source. Now you are presented with a screen to enter the settings of your Data Source connection
3. Type choose JDBC Data Source. JDBC Driver option choose Other... .
4. JDBC Driver (required): enter com.hxtt.sql.xml.XMLDriver
5. URL (required): enter jdbc:xml:/yourDatabasePath .
6. Click Save button.
7. Data Source Name (required): enter a user-friendly name for the driver, for isntance, testXML.
8. Click Save button.

20. Where's the support package for MySQL Migration?
The support package for MySQL GUI Tools 5.0 is at http://www.hxtt.com/test/mysql_migration.zip.
The support package for MySQL Migration Toolkit v1.0.21 is at http://www.hxtt.com/test/mysql_migration1.0.21.zip.
The support package for MySQL Migration Toolkit v1.0.20 is at http://www.hxtt.com/test/mysql_migration1.0.20.zip.