Technical Note 04-23
JDBC コネクティビティ

By Yvan Ayaay, Technical Support, 4D, Inc.
Technical Note 04-23

（原題: JDBC Connectivity for 4D Server）

概要

最新の JDBC(Java Database Connectivity) driver for 4D Server を使用すれば、Java プログラムから 4D Server に接続し、クエリを実行することが可能になります。同ドライバは Java Virtual Machine をサポートするすべての OS に対応したマルチプラットフォーム API です。

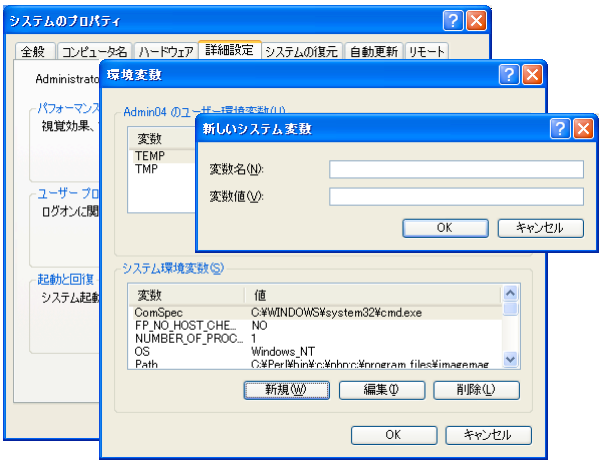
Java プログラムで 4D JDBC を使用する

はじめに 4D Server との接続を確立し、次いで SQL ステートメントを通して 4D データベースにアクセスすることになります。JDBC ドライバは、クエリを 4D が理解できるネットワークコールに変換する働きをします。

接続を試みる前に、JDBX ドライバ(jdbc4d.jar)ファイルを CLASSPATH に追加します。たとえば Windows でドライバが 4D フォルダに置かれている場合、コンソールで次のようにタイプすることができます。

```
C:\>set classpath=.;C:\4D\jdbc4d.jar
```

XP か 2000 を使用しているなら、コントロールパネル/システムで詳細設定タブを選択し、新しい環境変数 classpath を作成して値を.;C:\4D\jdbc4d.jar に設定することができます。



JDBC ドライバをロードする

ドライバは Java メソッド Class.forName にフルクラス名を渡すことによってロードできます。

Class.forName(“com.fourd.jdbc.DriverImpl”);

4D Server に接続する

ドライバがロードされたら Java SQL DriverManager クラスの getConnection メソッドに 4D の URL、ユーザ名、パスワードをアーギュメントとして渡し、接続のインスタンスを受け取ることができます。

```
import java.sql.*;
import java.util.*;

public class Connect4D
{
	public Connect4D()
	{
	super();
	}
	public static Connection Connect4DServer()
	{
		String driver = " com.fourd.jdbc.DriverImpl";
		String url = "jdbc:4d:127.0.0.1";
		String username = "designer";
		String password = "test";
		Connection c=null;
		t ry
		{
			Class.forName("com.fourd.jdbc.DriverImpl"); // load JDBC driver for 4D Server
			c = DriverManager.getConnection(url, username, password); //establish connection
		}
		catch (SQLException e)
		{
			e.printStackTrace();
		}
		catch (java.lang.ClassNotFoundException e)
		{
			System.err.print("ClassNotFoundException: " ) ;
			System.err.println(e.getMessage());
		}
		return c;
	}
}
```

このクラスをインスタンス化すれば Connect4DServer メソッドをコールすることによって 4D Server との接続を確立することができます。

データを読み取る

接続が確立したなら、そのインスタンスを使用して 4D Server のアクセスすることができます。データを読み取るには、接続インスタンスを使用してステートメントインスンタンスを作成し、SQL を収めてから実行します。クエリの結果は getString、getInt、getBytes などの getType メソッドを使って取り出すことができます。4D JDBC は Java と 4D のデータタイプを次のようにマッピングします。

4D	Java
文字	String
テキスト	String
実数	Double
整数	Short
倍長整数	Integer
日付	java.sql.Date
時間	java.sql.Time
ブール	Boolean
BLOB	Byte[ ]

```
import java.sql.*;
import java.util.*;

public class DatabaseRead
{
	public static void main(String[] args)
	{
	Connection conn= Connect4D.Connect4DServer(); //establish connection to 4D Server
	String select= "SELECT FName,LName FROM Student where Major='Biology'"; //declare select command
	try{
		Statement s = conn.createStatement(); //create statement
		ResultSet rs = s.executeQuery(select); //execute select command
		while (rs.next())
		{
			fname =rs.getString(1); //get result
			lname =rs.getString(2);
			System.out.println(fname+" "+lname);
		}
		s.close();
		conn.close();
	  }
	catch(SQLException ex)
		{
		ex.printStackTrace();
		}
	}
}
```

4D JDBC がサポートする SELECT の機能およびシンタックスは次のとおりです。

```
SELECT_Command :=    SELECT    * | Column_Comma_List
                     FROM TableName [AS TableAlias]
                     [LEFT JOIN TableName[AS TableAlias]ON Join_Condition_List]
                     [Where Boolean_Expression]
                     [ORDER BY Sorting_Comma_List]

Column_Reference :=  [TableName_or_TableAlias.]ColumnName[AS “ColumnAlias”]

Join_Condition_List  :=  Column_Reference   |
                         Column_Comma_List , Column_Reference

Boolean_Expression  :=   Column_Reference <|<=|=|>=|>|!= Scalar_Value   |
                         Boolean_Expression AND | OR
                         Column_Reference <|<=|=|>=|>|!= Scalar_Value

Sorting_Comma_List :=    Column_Reference[ASC|DESC]     |
                         Column_Comma_List , Column_Reference [ASC | DESC]
```

データを書き込む

読み取る場合と同じように接続インスタンスを使用してステートメントインスタンスを作成し、UPDATE、INSERT、DELETE などの SQL を実行します。

4D JDBC がサポートする UPDATE、INSERT、DELETE の機能およびシンタックスは次のとおりです。

```
UPDATE_Command :=    UPDATE TableName
                     SET Assign_Comma_List
                     [ WHERE Boolean_Expression ]

Assign_Comma_List :=     ColumnName = NewScalarValue   |
                         Assign_Comma_List , ColumnName = NewScalarValue

INSERT_Command :=    INSERT INTO TableName
                     [ ( Column_Name_List) ]
                     VALUES ( Scalar_Comma_List)

Column_Name_List :=      ColumnName     |
                         Column_Name_List , ColumnName

Scalar_Comma_List :=     ScalarValue      |
                         Scalar_Comma_List , ScalarValue

DELETE_Command :=    DELETE FROM TableName
                     [ WHERE Boolean_Expression]
```

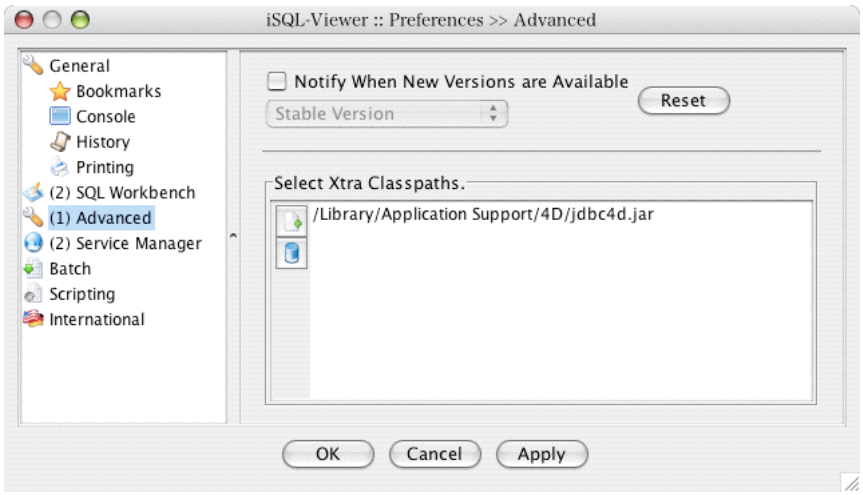
iSQL-Viewer を 4D Server に接続する

4D JDBC driver を使用すれば JDBC をサポートする外部アプリケーションから 4D に接続することが可能です。そのアプリケーションの CLASSPATH にドライバを追加し、JDBC 接続を設定するだけで 4D Server に接続することができるからです。

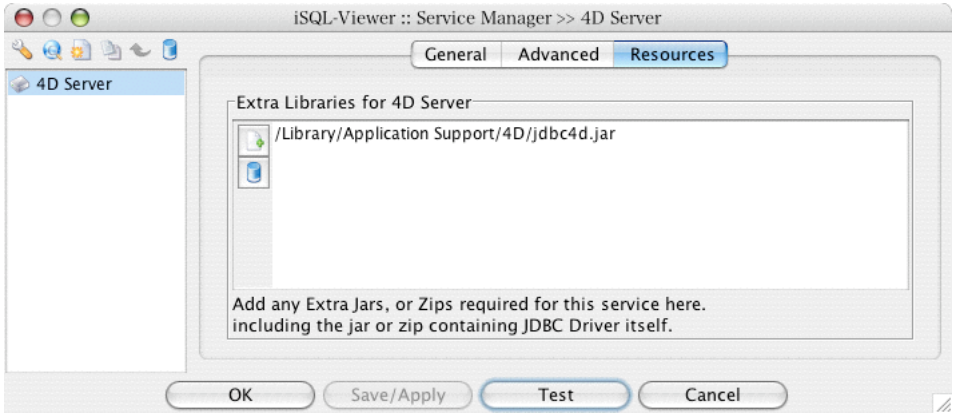
iSQL-Viewer はオープンソースのデータベースフロントエンドアプリケーションで、クエリの実行に JDBC を利用しています。JDBC によるデータのインポート/エクスポートの方法は、他の JDBC アプリケーションの場合もほとんど同じです。

4D Server に接続する

接続を試みる前に、JDBX ドライバ(jdbc4d.jar)ファイルを CLASSPATH に追加します。環境設定で Advanced を選択し、Select Xtra Classpaths にパスを入力して Apply をクリックします。



Service Manager を開き、接続名、JDBC クラス名(com.fourd.jdbc.DriverImpl)、JDBC URL(jdbc:4d: 4D Server の IP アドレス)、ユーザ名、パスワードを入力して接続を定義します。



Test ボタンをクリックして接続のテストしてみます。接続に成功すると 4D Server のプロセスビューアに JDBC 接続が緑色の文字で表示されます。

接続が確立したなら、iSQL-Viewer の SQL エディタを使用してクエリを実行することができます。SQL の結果は中央のウィンドウに表示され、他のデータソース、あるいは HTML、XML、ASCII 区切りフォーマットをサポートするアプリケーションにエクスポートすることができます。4D Server に対して INSERT、UPDATE、DELETE などの SQL を実行することもできます。