

IBM i World 2022

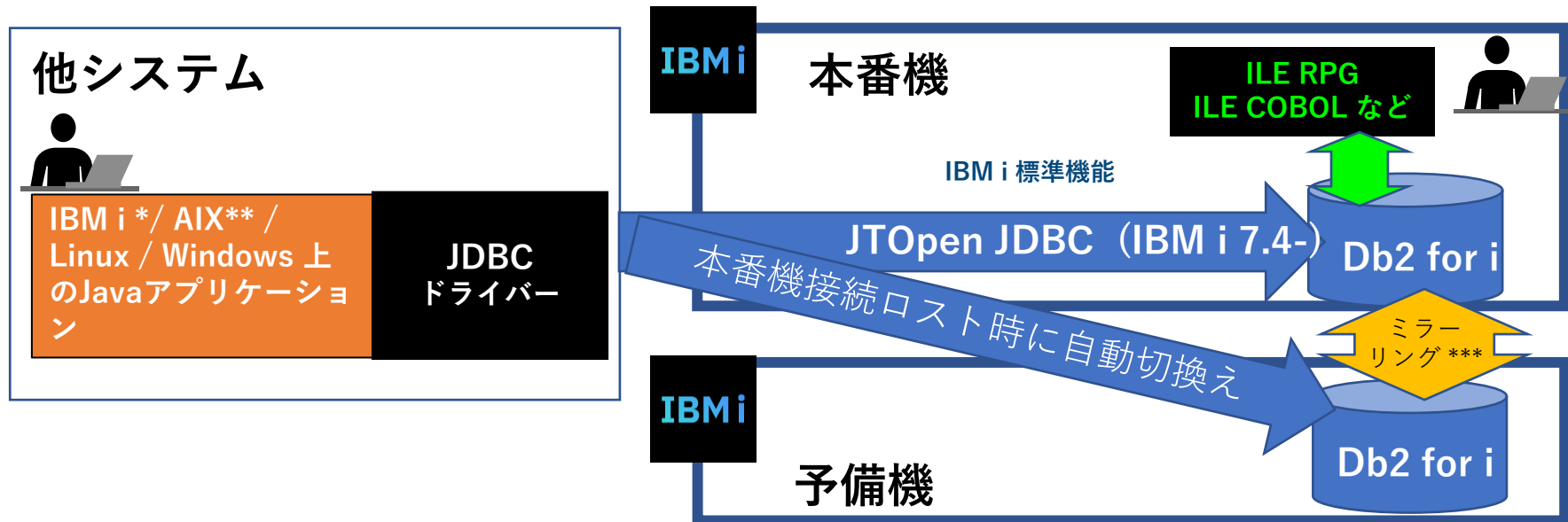
IBM i コンテンツ (2022年10月版)

**基幹DBにJavaアプリからリアルタイム・アクセスしよう
- JDBCによるIBM i DB (Db2 for i) アクセス方法のご紹介 -**

日本アイ・ビー・エム株式会社
テクノロジー事業本部
IBM Powerテクニカルセールス
澤田英寿

アプリ開発を より簡単に

- JDBCを利用して、IBM i のデータベース (Db2 for i) にアクセス
- 他システムから IBM i 本番機への接続ロスト時には IBM i 予備機へ自動切換え



*: IBM i のJavaアプリは右側「本番機」上でも稼働できます。

** : AIXのJavaアプリは右側「本番機」HW上の別LPARでも稼働できます。

***: IBMのDb2 Mirror for iの他、HAのISVソリューション等がミラーリングに利用できます。

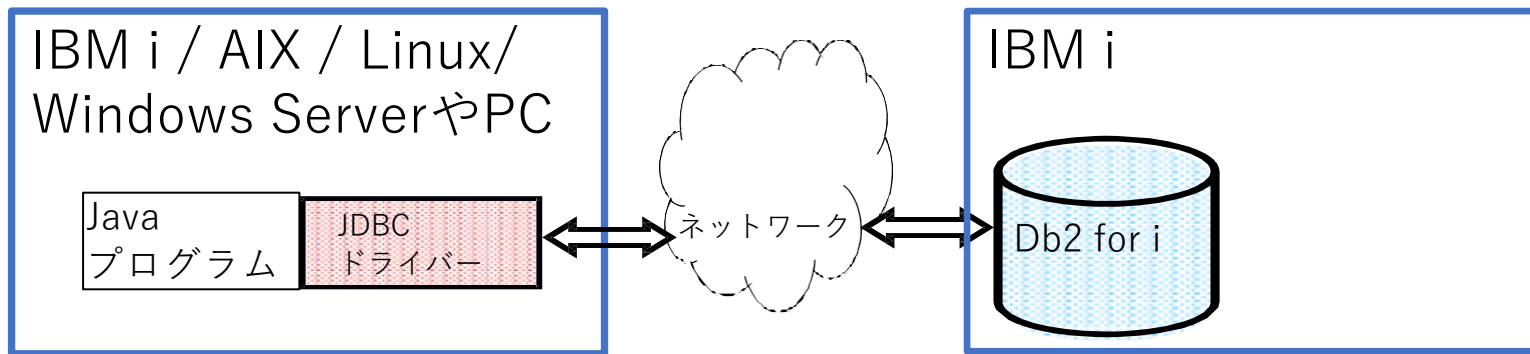
目次

1. JDBC概要と導入手順
2. JDBCでアクセスするJavaプログラムを作ってみよう
3. JT400機能を使ってJavaプログラムからIBM i コマンドを実行
4. 本番機接続ロスト時の自動切換えサンプル
5. 補足情報

1. JDBC概要と導入手順 (1)

JDBC機能の概要

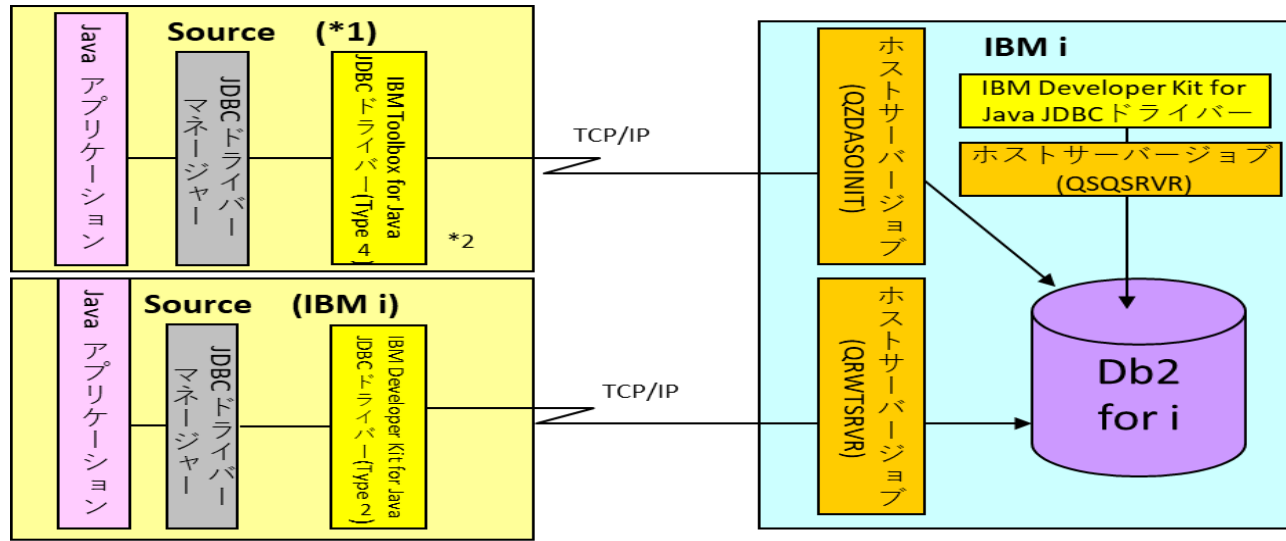
- 他社サーバーや、PC上で、IBMが提供するJDBCドライバーを経由してIBM iサーバー上のデータベースにアクセスすることが可能です。
- この資料では、PC上のJavaプログラムから、JDBCを経由してIBM iデータベースにアクセスする方法をご紹介します



1. JDBC概要と導入手順 (2)

JDBC接続構成

IBM 提供のJDBCドライバーによる、Db2 for iへの接続構成は、以下ようになります。ソースシステム上のJavaアプリケーションからJDBCドライバーマネージャーおよびIBM Toolbox for Java JDBCドライバーを経由します。IBM i上でホストサーバージョブ QZDASOINITを介して、Db2 for iへ接続します。リモートから接続するのであれば、IBM Toolbox for Java JDBCドライバーを使用することが推奨されています。

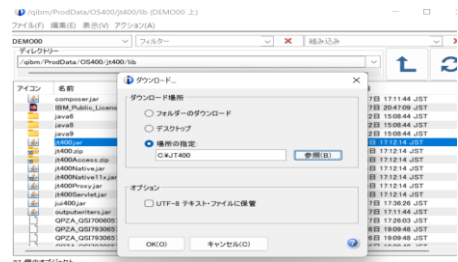


*1: AIX, Windows, Linux, IBM i
*2: JTOpen JDBC ドライバーも可能

1. JDBC概要と導入手順 (3)

JDBCドライバーの導入

- IBM® Toolbox for Java™ は、IBM i のデータとリソースへのアクセスを最適化した Java クラスの ライブラリーです。IBM Toolbox for Javaの JDBC driver を使用すると、Java をサポートしている任意のプラットフォーム上で実行されるサーバー・サイドおよびクライアントの Java アプリケーションから、ローカルまたはリモートの Db2 for i データベースにアクセスできます。
- 最新版のJDBCドライバーのダウンロード方法は、下記のURLにある、PTFをIBMiに適用して、IBM i上からPCへダウンロードします。
<https://sourceforge.net/projects/jt400/> (左記には、下記のPTFを適用するように指示があります)
 V7R5M0 SI79326
 V7R4M0 SI79306
 V7R3M0 SI79308
- PTFを適用すると、IBM i上 (下記フォルダー) に、jt400.jarの最新版が作成されるので、これをPCへダウンロードします。 **/qibm/ProdData/OS400/jt400/lib/jt400.jar**
- 例えば、PC上に、C:¥JT400という名称のフォルダーを作成し、ACSの統合ファイルシステムの機能を利用して、jt400.Jarをダウンロードします (右図イメージ)。

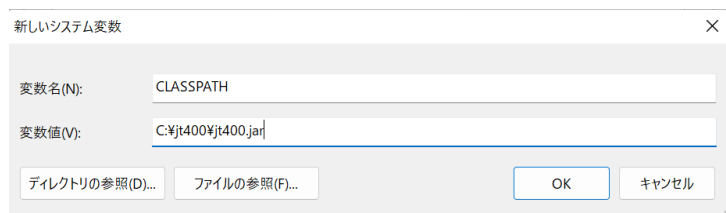


1. JDBC概要と導入手順 (4)

JDBCドライバーと環境設定

- PC上の CLASSPATH 環境変数を更新します。たとえば、Windows を使用していて、jt400.jar を C:¥jt400にコピーした場合、CLASSPATH の終わりに次のストリングを追加します。

c:¥jt400¥jt400.jar



- また、開発の前提として、Java(JDK)がPCにインストールされているか確認してください。PC上では下記のjavaコマンドで確認できます。

```
C:¥JAVASAMPLE>java -version
openjdk version "18.0.2.1" 2022-08-18
OpenJDK Runtime Environment (build 18.0.2.1+1-1)
OpenJDK 64-Bit Server VM (build 18.0.2.1+1-1, mixed mode, sharing)
```

- 設定の詳細は、下記のマニュアルを参照してください。
IBM Toolbox for Java をワークステーションにインストールする
<https://www.ibm.com/docs/ja/i/7.5?topic=java-installing-toolbox-your-workstation>

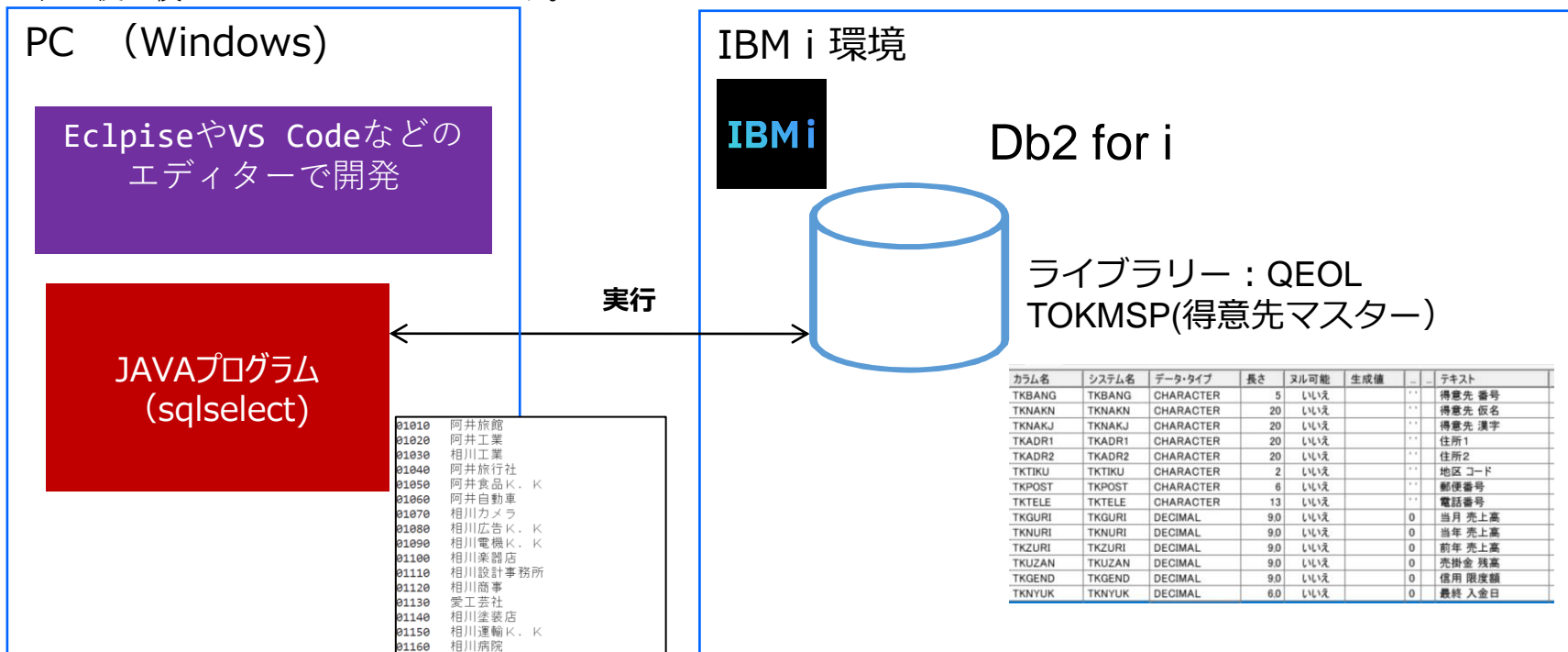
2. JDBCでアクセスするJavaプログラムを作ってみよう

データベースにアクセスするプログラムを作ってみよう（実行環境イメージ）

IBM i上の既存のライブラリーにある、データベースにアクセスしてデータを表示するプログラムを作ってみましょう。

Javaのソース編集を、この章ではVS Codeを使用して、PCのソースファイルを編集します。

他の使い慣れたエディターでもOKです。



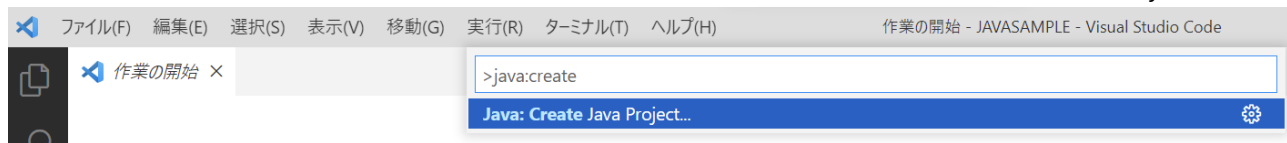
2. JDBCでアクセスするJavaプログラムを作ってみよう

データベースにアクセスするプログラムを作ってみよう (VS Code設定1)

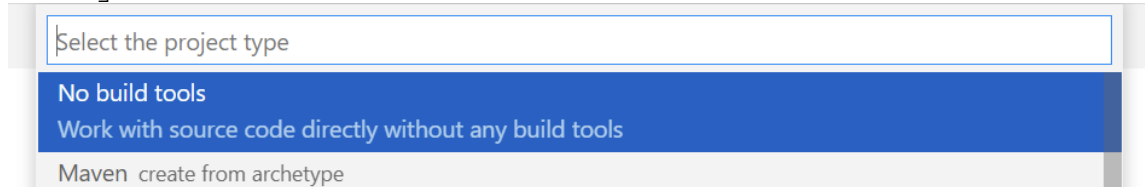
Visual Studio Code (VS Code) で新規プロジェクトを作成します。

VS CodeのExtension Pack for Javaが導入されている前提で、下記のようにしてください。

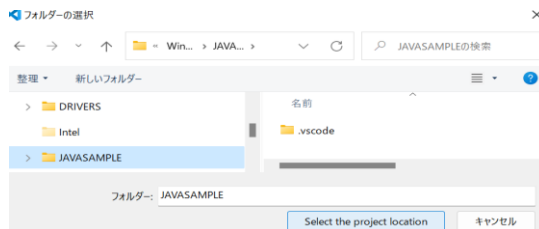
1. 「表示」メニューのコマンドパレットを選択して、下記のように「Java : Create Java Project」を選択。



2. 「No build tools」をクリック



3. プロジェクトを格納するフォルダーの親フォルダー (ここではを選択javasample というフォルダーを作成しました)



2. JDBCでアクセスするJavaプログラムを作ってみよう

データベースにアクセスするプログラムを作ってみよう (VS Code設定2)

4. プロジェクトを格納するフォルダー名を指定 (ここではJDBCSAMPLE)

Input a Java project name ('Enter' を押して確認するか 'Escape' を押して取り消します)

5. 下記のようになります。JAVAプログラムのソースは、srcフォルダーに作成します。

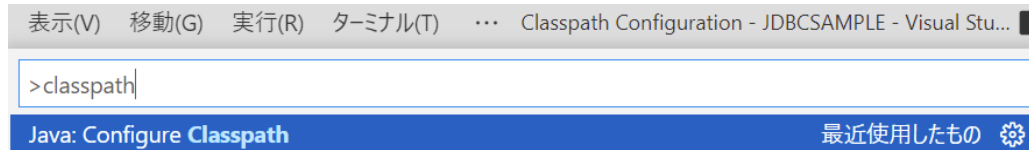
```
▼ JDBCSAMPLE
  > .vscode
  > lib
  > src
  ⓘ README.md
```

2. JDBCでアクセスするJavaプログラムを作ってみよう

データベースにアクセスするプログラムを作ってみよう (VS Code設定3)

JDBC利用のJavaプログラムに必要な、CLASSPATHの設定は下記のようにします。

1. 「表示」メニューのコマンドパレットを選択して、下記のように「classpath」を選択。



2. 下記の画面で path設定で、「C:\JT400\jt400.jar」を追加

The screenshot shows the 'Configure Classpath' dialog in VS Code. The 'Path' field under the 'Sources' section is highlighted with a red box. A blue arrow points from this box to a callout box on the right. The callout box shows the path 'c:\JT400\jt400.jar' and an 'Add' button. Below the main dialog, the 'Referenced Libraries' section is also highlighted with a red box, showing 'No referenced libraries are configured.' and an 'Add' button.

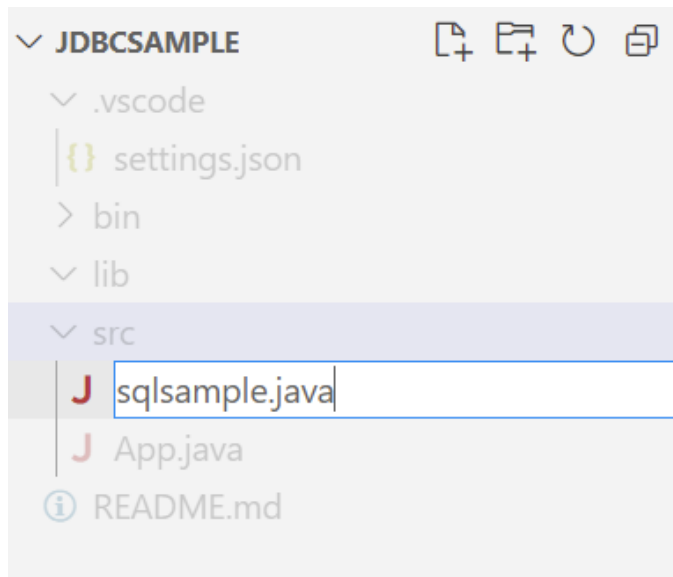
2. JDBCでアクセスするJavaプログラムを作ってみよう

データベースにアクセスするプログラムを作ってみよう（ソースコード入力）

JDBCを使ったサンプルプログラム（sqlsample）を作ってみましょう。

VS Codeで新規プログラム（sqlselect）を作成します。

JDBCSAMPLEプロジェクトの[src]の下に下記のように作成します。



自動的に下記のように入力されますが、次ページのソース（sqlsample）に置き換えてください。

```
src > J sqlsample.java > ...  
1 public class sqlsample {  
2  
3 }  
4
```

プログラムソース (sqlsample)

```
import java.sql.*;

public class sqlsample {

    /**
     * @param args
     */
    public static void main(String[] args) {

        String url = "jdbc:as400://";
        String system = "demo00";// IPアドレス
        String libl = "JDBCLIB";// スキーマ
        String user = "SAWADA";//ユーザーID
        String password = "SAWADA";//パスワード
        Connection con = null;

        try {
            // JDBCドライバのロード
            Class.forName("com.ibm.as400.access.AS400JDBCDriver");

            // DB接続
            con = DriverManager.getConnection(
                url +
                system + ";" + "libraries=" + libl,
                user, password);
        }
    }
}
```

前半

```
// SQLコンテナの作成
Statement stmt = con.createStatement();
// SQLステートメント作成
String sql = "SELECT TKBANG, TKNKJ FROM JDBCLIB.TOKMSP";
// SQL実行
ResultSet rs = stmt.executeQuery(sql);

// 検索結果取り出し
int i = 0;
while(rs.next()) {
    System.out.println(rs.getString(1) + "¥t" + rs.getString(2));
    i++;
}
stmt.close();
} catch(SQLException e) {
    while(e != null) {
        System.out.println("エラー：SQL例外");
        System.err.println(e.getMessage());
        System.out.println("");
        e = e.getNextException();
    }
} catch(Exception e) {
    System.out.println("エラー：例外");
    e.printStackTrace();
} finally {
    try {
        con.close();
    } catch(SQLException e) {
        e.printStackTrace();
    }
}
}
```

後半

2. JDBCでアクセスするJavaプログラムを作ってみよう

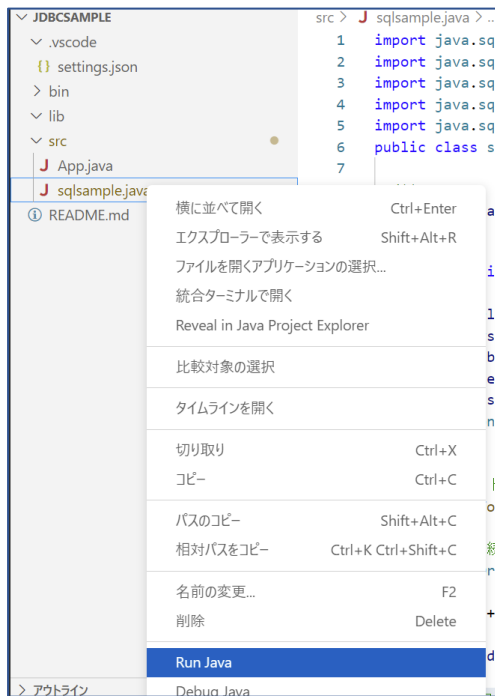
データベースにアクセスするプログラムを作ってみよう（プログラム実行）

JDBCのサンプルプログラム（sqlsample）を実行してみよう。

VS Codeでプログラム（sqlsample）を実行します。
プログラムを選択してRun Javaを選択すると、
コンパイル&実行されます。

結果が、VS Codeのターミナルに表示されます。

・コンパイルが実行され、その後、プログラムが実行され
ます。得意先番号と得意先名が取得されています。



2. JDBCでアクセスするJavaプログラムを作ってみよう

データベースにアクセスするプログラムを作ってみよう（プログラム解説1）

```
import java.sql.*;
public class sqlsample {
    /** @param args*/
    public static void main(String[] args) {
        String url = "jdbc:as400://";
        String system = "demo00"; // IPアドレス
        String libl = "JDBCLIB"; // スキーマ
        String user = "SAWADA"; // ユーザーID
        String password = "SAWADA"; // パスワード
        Connection con = null;
        try {
```

前半

簡単にプログラムを解説します。

Toolbox for JavaのJDBCドライバー (jt400.jar)によるIBM iデータベースアクセスのためのモジュールを使うのに必要な設定

```
    // JDBCドライバのロード
    Class.forName("com.ibm.as400.access.AS400JDBCdriver");
    // DB接続
    con = DriverManager.getConnection(url + system + ";" + "libraries=" +
libl,user,password);
    // SQLコンテナの作成
    Statement stmt = con.createStatement();
    // SQLステートメント作成
    String sql = "SELECT TKBANG, TKNAKJ FROM JDBCLIB.TOKMSP";
    // SQL実行
    ResultSet rs = stmt.executeQuery(sql);
```

executeQuery(sql) でselect文を実行します。

2. JDBCでアクセスするJavaプログラムを作ってみよう

データベースにアクセスするプログラムを作ってみよう（プログラム解説2）

// 検索結果取り出し

```
int i = 0;
while(rs.next()) {
    System.out.println(rs.getString(1) + "¥t" + rs.getString(2));
    i++;
}
```

後半

```
stmt.close(); //ステートメントのCLOSE
```

```
} catch(SQLException e) {
while(e != null) {
    System.out.println("エラー：SQL例外");
    System.err.println(e.getMessage());
    System.out.println("");
    e = e.getNextException();
}
} catch(Exception e) {
    System.out.println("エラー：例外");
    e.printStackTrace();
} finally {
try {
    con.close(); //接続のCLOSE
} catch(SQLException e) {
    e.printStackTrace();
}
}
}
```

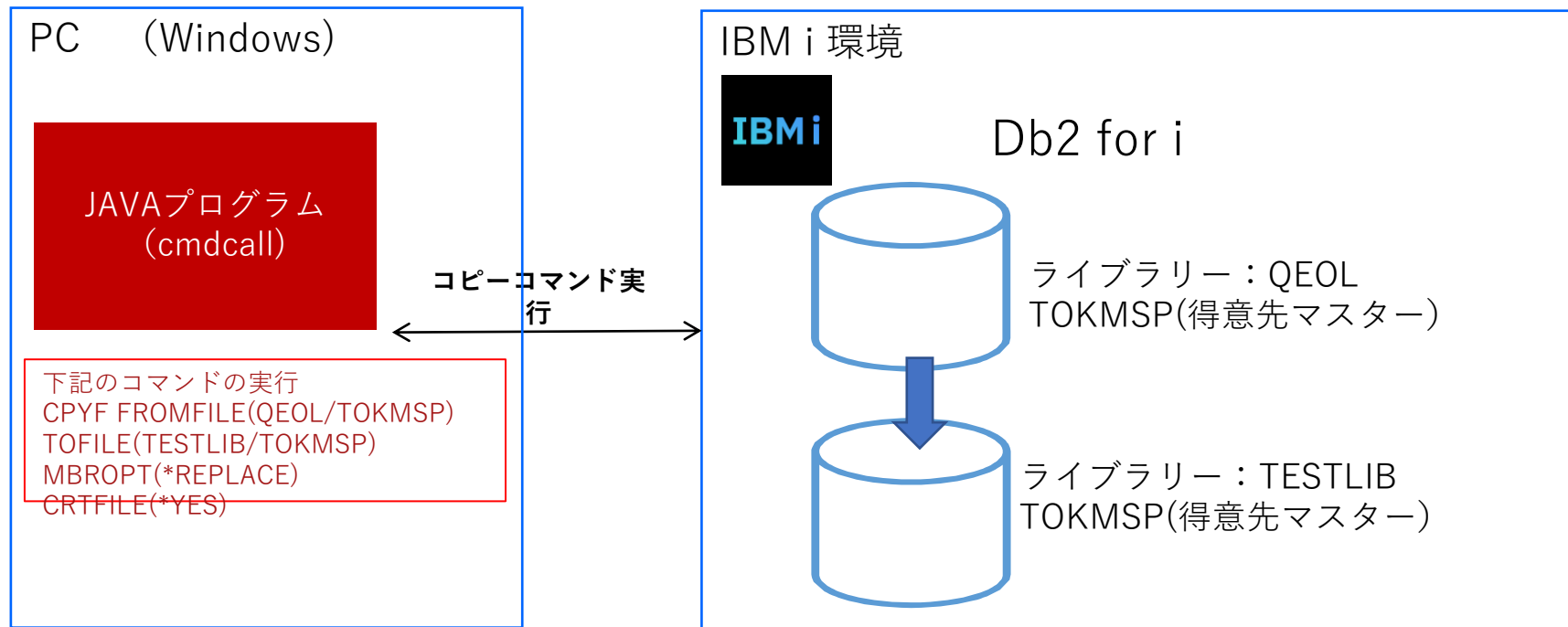
簡単にプログラムを解説します。
検索結果は、処理対象の行を表すために、カーソルを用います。
Next () メソッドにより1行ずつ処理します。
行データは、getxxx()メソッドで取り出します。getxxx(1)を指定すると1列目のフィールドのデータが取得できます。

エラー時の処理は重要です。
SQL Exceptionは、SQL実行時のエラーの問題判別になります。
Exceptionは、プログラム中に発生するエラーを処理します。

3. JT400機能を使ってJavaプログラムからIBM iコマンドを実行

Toolbox for Javaのjt400のクラスを使ってIBM iにコマンド発行する（実行環境イメージ）

ToolboxにあるJt400.jarのクラスライブラリーには、IBM i上の様々なオブジェクトにアクセスする機能が含まれています。その1つである、IBM iのコマンドを発行する機能を試してみましょう。

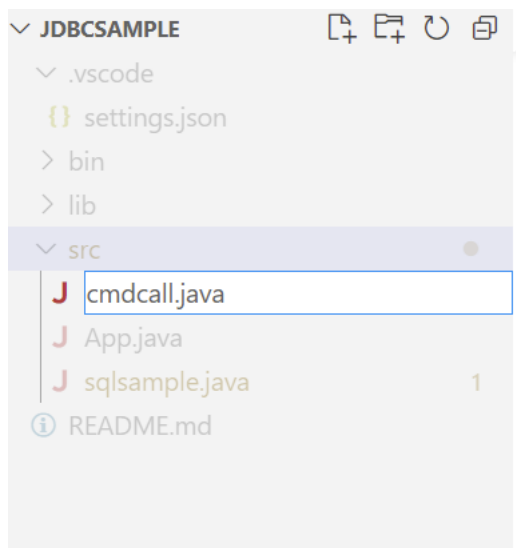


3. JT400機能を使ってJavaプログラムからIBM iコマンドを実行

JT400を使ったJavaプログラム (cmdcall) を作ってみましょう (ソースコード入力)。

VS Codeで新規プログラム (cmdcall) を作成します。

JDBCSAMPLEプロジェクトのsrcフォルダーの下に下記のように作成します。



自動的に下記のように入力されますが、次ページのソース (cmdcall.java)に置き換えてください。

```
src > J cmdcall.java > ...  
1 public class cmdcall {  
2  
3 }  
4
```

プログラムソース (cmdcall)

```
import com.ibm.as400.access.*;
public class cmdcall {
    public static void main(String[] args)
    {
        try
        {
            AS400 system = new AS400("demo00","SAWADA","SAWADA");

            CommandCall cmd = new CommandCall(system);

            cmd.run(" CPYF FROMFILE(QEOL/TOKMSP) TOFILE(TESTLIB/TOKMSP) MBROPT(*REPLACE) CRTFILE(*YES) ");

            AS400Message[] ml = cmd.getMessageList();
            for (int i=0; i<ml.length; i++)
            {
                System.out.println(ml[i].getText());
            }
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
        System.exit(0);
    }
}
```

3. JT400機能を使ってJavaプログラムからIBM iコマンドを実行

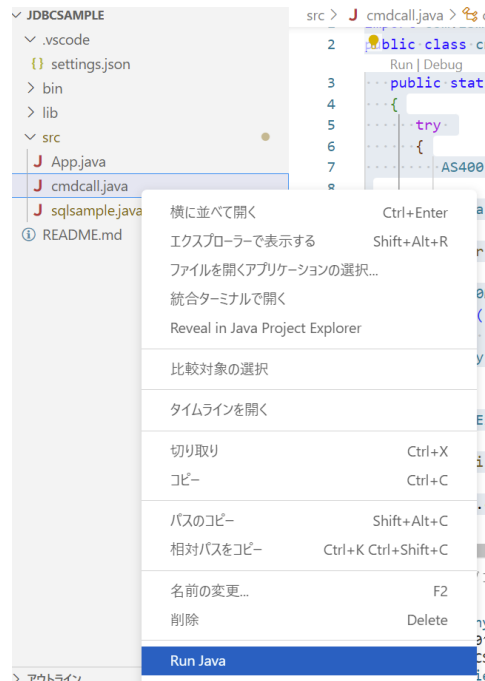
JT400を使ったJavaプログラム (cmdcall) を作ってみましょう (プログラム実行)

サンプルプログラム (cmdcall) を実行してみよう。

VS Codeでプログラム (cmdcall)を実行します。

プログラムを選択して Run Javaを選択すると、
コンパイル&実行されます。

結果が、VS Codeのターミナルに表示されます。
・コンパイルが実行され、その後、プログラムが
実行されます。TOKMSPファイルが、TESTLIBに
コピーされました。



```
PS C:\JAVASAMPLE\JDBCSAMPLE> c:; cd 'c:\JAVASAMPLE\JDBCSAMPLE'; & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java
.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '@C:\Users\204685~1\AppData\Local\Temp\cp_1h
x2b7u11nzyxubsuc5feonyp.argfile' 'cmdcall'
メンバーTOKMSPから210個のレコードがコピーされた。
```

3. JT400機能を使ってJavaプログラムからIBM iコマンドを実行

JT400を使ったJavaプログラム (cmdcall) を作ってみましょう (プログラム解説)

```
import com.ibm.as400.access.*;
```

```
public class cmdcall {  
    public static void main(String[] args)  
    {
```

```
        try
```

```
        {  
            AS400 system = new AS400("demo00", "SAWADA", "SAWADA");  
            CommandCall cmd = new CommandCall(system);
```

```
            cmd.run(" CPYF FROMFILE(QEOL/TOKMSP) TOFILE(TESTLIB/TOKMSP) MBROPT(*REPLACE) CRTFILE(*YES) ");
```

```
            AS400Message[] ml = cmd.getMessageList();  
            for (int i=0; i<ml.length; i++)  
            {  
                System.out.println(ml[i].getText());  
            }  
        }
```

```
        catch (Exception e)  
        {  
            e.printStackTrace();  
        }  
        System.exit(0);  
    }
```

```
}
```

簡単にプログラムを解説します。

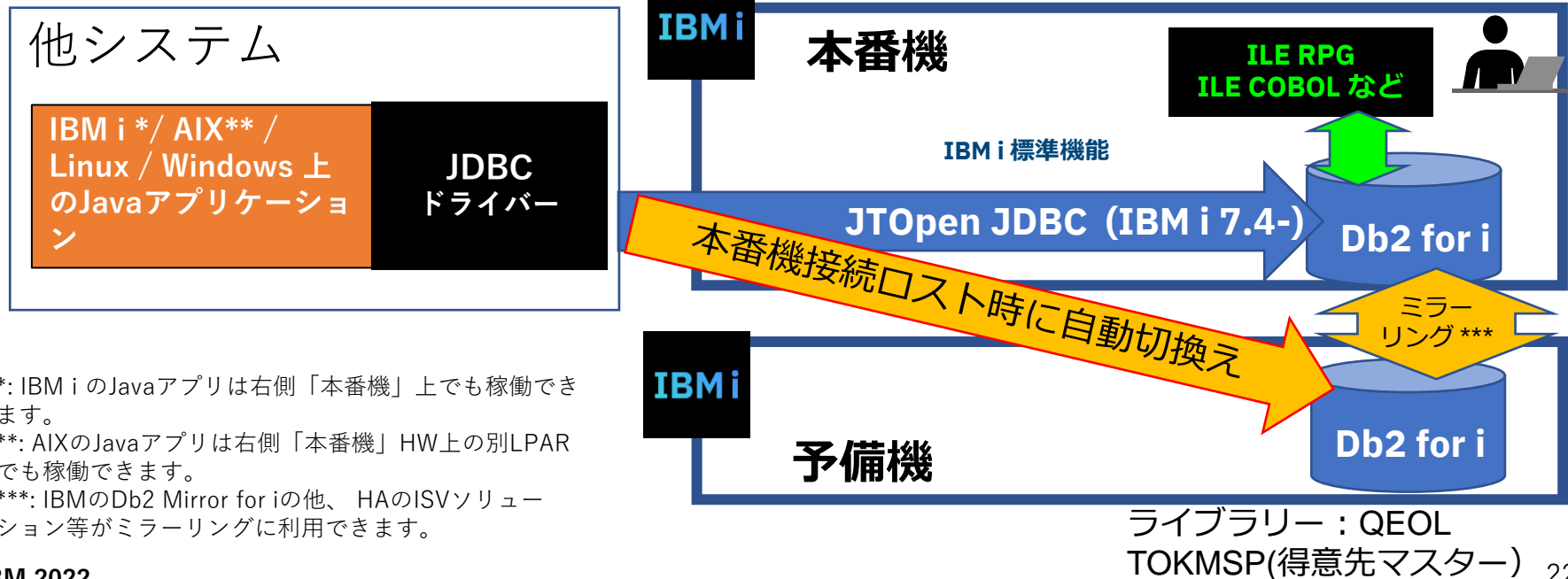
Toolbox for Javaのjt400.jarによる
IBM iコマンド発行のためのモジュール
を使うのに必要な設定

IBM Toolbox for Java CommandCall オブジェクトを使用して、コマンドをサーバーに送信します。CommandCall オブジェクトを作成するときは、IBMiオブジェクトを渡して、どのサーバーがコマンドのターゲットであるかを認識できるようにします。Run()メソッドでコマンド実行

コマンドを実行した結果は、IBM iメッセージのリストになります。IBM Toolbox for Java は、これらのメッセージを AS400Message オブジェクトとして表します。コマンドが完了すると、CommandCall オブジェクトから結果のメッセージが表示されます。

4.本番機接続ロスト時の自動切換えサンプル

- 他システムから IBM i 本番機への接続ロスト時には IBM i 予備機へ自動切換え
- この「可用性のための JTOpen JDBC ドライバーのサポート」は、JT/Open 9.6 以降でサポートされています。



*: IBM i のJavaアプリは右側「本番機」上でも稼働できます。

** : AIXのJavaアプリは右側「本番機」HW上の別LPARでも稼働できます。

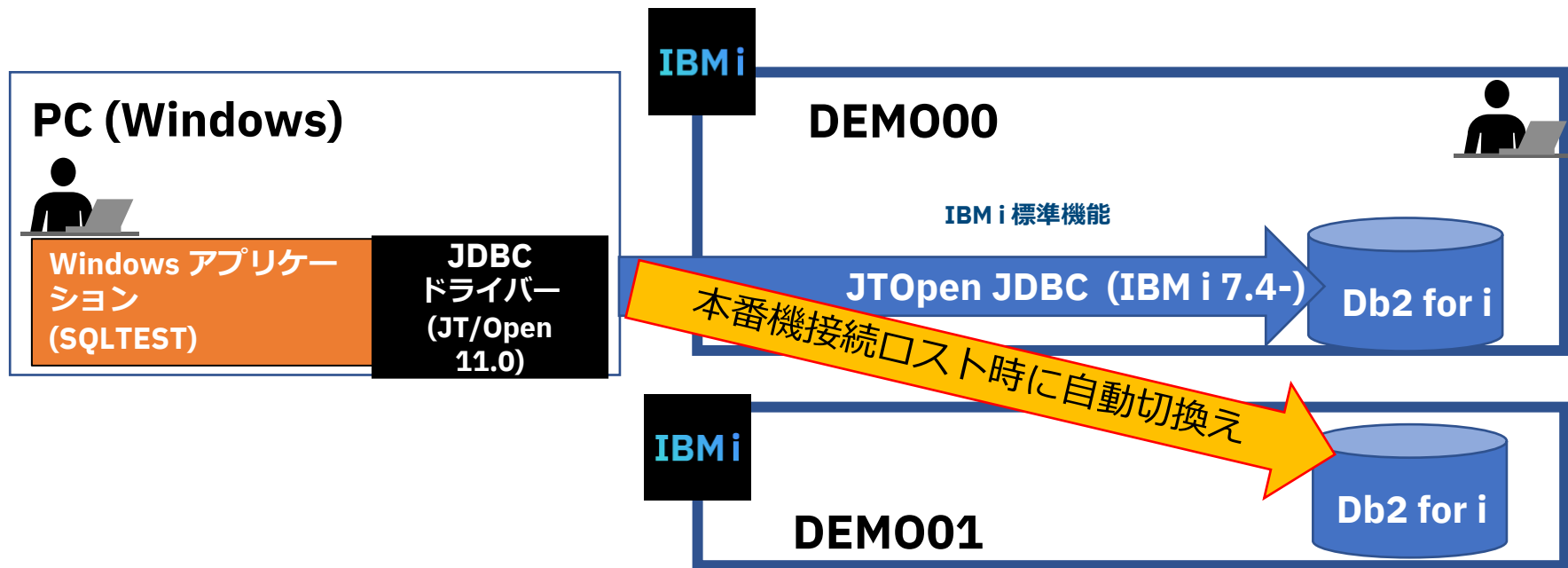
*** : IBMのDb2 Mirror for iの他、HAのISVソリューション等がミラーリングに利用できます。

4.本番機接続ロスト時の自動切換えサンプル

簡単なテストプログラムで、実証してみましよう。

下記のような環境を想定しています。(実行イメージ)

DEMO00が本番サーバーで、障害時には、予備サーバーのDEMO01に切り替わります。



下記のようなプログラムソースを作成します。(SQLTEST)

```
import java.sql.*;
import java.util.Properties;
public class SQLTEST {
    /** @param args */
    public static void main(String[] args) {
        String url = "jdbc:as400://DEMO00/QEOL";
        java.util.Properties properties = new java.util.Properties ();
        properties.setProperty ("password", "SAWADA");
        properties.setProperty ("user", "SAWADA");
        properties.setProperty ("enableClientAffinitiesList", "1");
        properties.setProperty ("clientRerouteAlternateServerName", "DEMO01");
        properties.setProperty ("password", "SAWADA");
        Connection con = null;
        try {
            // JDBCドライバのロード
            Class.forName("com.ibm.as400.access.AS400JDBCDriver");
            // DB接続
            con = DriverManager.getConnection(url,properties);
            // SQLコンテナの作成
            Statement stmt = con.createStatement();
            String sql = "SELECT TKBANG, TKNKJ FROM QEOL.TOKMSP";
            ResultSet rs = stmt.executeQuery(sql);
```

前半

```
        // 検索結果取り出し
        int i = 0;
        while(rs.next()) {
            System.out.println(rs.getString(1) +
                "%t" + rs.getString(2));
            i++;
        }
        stmt.close();
    } catch(SQLException e) {
        while(e != null) {
            System.out.println("エラー : SQL例外");
            System.err.println(e.getMessage());
            System.out.println("");
            e = e.getNextException();
        }
    } catch(Exception e) {
        System.out.println("エラー : 例外");
        e.printStackTrace();
    } finally {
        try {
            con.close();
        } catch(SQLException e) {
            e.printStackTrace();
        }
    }
}
}
```

後半

4. 本番機接続ロスト時の自動切換えサンプル

サンプルプログラム(SQLTEST)を実行してみましょう。(プログラム実行)

VS Codeでプログラム (SQLTEST)を実行します。

プログラムを選択して Run Javaを選択すると、
コンパイル&実行されます。



結果が、VS Codeのターミナルに表示されます。

・コンパイルが実行され、その後、プログラムが実行されます。本番機 (DEMO00)を止めておくと本番機サーバーが障害であることを検知して、予備機サーバー (DEMO01) からデータを照会しています。

```
PS C:\JAVASAMPLE\JDBCSAMPLE> & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '@C:\Users\204685~1\AppData\Local\Temp\cp_1hx2b7u1lnzyxubsuc5feonyp.angfile' 'SQLTEST'
01010 阿井旅館
01020 阿井工業
01030 相川工業
01040 阿井旅行社
01050 阿井食品K、K
01060 阿井自動車
01070 相川カメラ
01080 相川広告K、K
01090 相川電機K、K
01100 相川楽器店
01110 相川設計事務所
```

4. 本番機接続ロスト時の自動切換えサンプル

自動切換えを使ったJavaプログラム（SQLTEST）の解説です。後半は2章と同じです。ここでは前半のみの解説です。

```
import java.sql.*;
import java.util.Properties;
public class SQLTEST {
    /** @param args */
    public static void main(String[] args) {
        String url = "jdbc:as400://DEMO00/QEOL";

        java.util.Properties properties = new java.util.Properties ();
        properties.setProperty ("password", "SAWADA");
        properties.setProperty ("user", "SAWADA");
        properties.setProperty ("enableClientAffinitiesList", "1");
        properties.setProperty ("clientRerouteAlternateServerName", "DEMO01");
        properties.setProperty ("password", "SAWADA");
        Connection con = null;
        try {
            // JDBCドライバのロード
            Class.forName("com.ibm.as400.access.AS400JDBCDrive");
            // DB接続
            con = DriverManager.getConnection(url,properties);
```

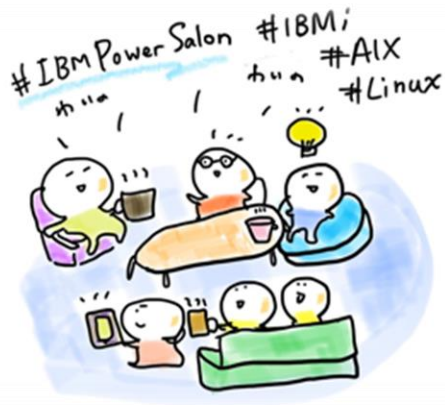
“enableClientAffinitiesList”というパラメータを1にすると、フェールオーバーが使用可能になります。“clientRerouteAlternateServerName”というパラメータは、本番サーバー(ここではDEMO00)にアクセスできない場合に切り替えるサーバー名(DEMO01)を指定します。

上記で指定したpropertiesを、getConnection()メソッドに指定します。

5. 補足情報

- (1) IBM i 7.5製品マニュアル「Java JDBC ドライバーを使用した IBM i データベースへのアクセス」
<https://www.ibm.com/docs/ja/i/7.5?topic=programs-accessing-your-i-database-java-jdbc-driver>
- (2) IBM i 7.5製品マニュアル「Java プログラムからのデータベース・アクセス」
<https://www.ibm.com/docs/ja/i/7.5?topic=java-database-access-from-programs>
- (3) JDBCアクセスでのパフォーマンス調整方法のガイド
「第9回 「Db2 for iのODBC/JDBCサーバージョブ QZDASOINITのパフォーマンス調整」
https://www.i-cafe.info/column/serials/dekiruibmi_no9
- (4) JDBCプログラミングの基本
<http://www63.tok2.com/home2/jd4/JDBCbasics.html>
- (5) JAVAプログラムでのコマンド呼び出し
<https://www.ibm.com/docs/ja/i/7.4?topic=classes-commandcall-class>
- (6) IBM Toolbox for Java JDBC プロパティ
<https://www.javadoc.io/static/net.sf.jt400/jt400/9.7/com/ibm/as400/access/docfiles/JDBCProperties.html>
- (7) JDBCの代替サーバーについての設定ガイド
<https://www.ibm.com/support/pages/jtopen-jdbc-driver-%E2%80%93-alternate-server-support>

IBM Power Salon のご案内



IBM Powerユーザーのための自由なオンラインサロンで、
お客様同士、IBMスペシャリストと繋がりませんか？

是非、お気軽にご参加ください。

内容：お客様によるDX事例、クラウド活用事例
IBM i/AIX/Linuxの技術情報、サポート情報
IBMスペシャリストによるQ&A 他

主催：日本アイ・ビー・エム（株）IBM Power 事業部

参加方法：オンライン開催
どなたでも参加可能、無料
事前申し込み不要

ご参加URL：ibm.biz/Powersalon-webex

お問い合わせ：NO1POWER@jp.ibm.com



	日程	テーマ
第11回	10月 12日(水)	株式会社 保健科学西日本様 ご講演
第12回	11月 9日(水)	立命館大学様 ご講演
第13回	12月 14日(水)	お客様講演予定

IBM Power Salonの詳細はこちら
→<https://ibm.biz/power-salon>

IBM Community Japan

IBM i Club ご案内

■ IBM i Clubとは

- 自社システムとしてIBM iをご利用いただいている皆様同士で、各社の工夫や事例を紹介し合ったり、ディスカッションをしていただく場です。
- IBMの技術者も参加し、IBM iの最新情報のご提供や、ディスカッションに入らせていただくこともあります。
- 年5回(予定)、それぞれテーマを変えて開催します。

■ 2021年のテーマ例 (ご参加の皆様からいただいた課題をもとにテーマを選出しました)

- IBM i ユーザーハイブリッドクラウドへの道
- IBM i 人材育成の勘所
- 新技術の活用
- IBM i ユーザーのDX
- IBM i の優位性と今後

■ 開催期間

- 2022年2月24日から2022年12月末(予定)

■ 開催内容

- オンライン(Webex)セッションの開催: 2022年2月24日(木)を第1回とし年間5回(予定)
- 情報共有・ディスカッションの場: コミュニケーションツール(Slack)をご利用いただき、セッション以外の時も情報交換など可能です。

■ ご参加にあたってのお願い・ご注意点

- 守秘義務をお守りください。
- 営業活動を目的としたご参加はお断りいたします。
- IBM i Clubお申し込みには、事前にIBM Community Japanのメンバー登録が必要です。
- 開催期間の途中からのご参加も可能です。



■ 2022年開催予定

*日時・内容が変更になる可能性があります

	日時	実施内容	
1	2/24 (終了)	最新情報	「数字で見るIBM i小辞典」IBM 佐々木
		事例紹介	「利用部門からの要望」にどう対応しているか? トクラス(株)様
		ディスカッション	テーマ: これからのIT部門の役割
2	5月	最新情報	「IBM i 新リリース発表」
		事例紹介	(調整中)
		ディスカッション	テーマ: 人材確保・人材育成
3	7月	最新情報	
		事例紹介	
		ディスカッション	テーマ: 新技術の活用(仮)
4	9月	最新情報	
		事例紹介	
		ディスカッション	テーマ: DX(仮)
5	11月	最新情報	
		事例紹介	
		ディスカッション	

■ コース詳細・お申込み

<https://www.ibm.com/ibm/jp/ja/ibmcommunityjapan-product-community.html>

■ ご参考 昨年 (2021年) 開催内容

https://higherlogicdownload.s3.amazonaws.com/IMWUC/2fde9da6-6e7d-43b4-bae3-7f25168bbbd0/UploadedImages/japan/2022/2021_IBM_i_Club.pdf

IBM i 関連情報

IBM i ポータル・サイト
<https://ibm.biz/ibmijapan>

i Magazine (IBM i 専門誌。春夏秋冬の年4回発刊)
<https://www.imagazine.co.jp/IBMi/>

月イチIBM Power情報セミナー「IBM Power Salon」
<https://ibm.biz/power-salon>

IBM i 関連セミナー・イベント
<https://ibm.biz/powerevents-j>

IBM i Club (日本のIBM i ユーザー様のコミュニティー)
<https://ibm.biz/ibmiclubjapan>

IBM i 研修サービス (i-ラーニング社提供)
<https://www.i-learning.jp/service/it/iseriess.html>

IBM Power Systems Virtual Server 情報
<https://ibm.biz/pvsjapan>

IBM i 情報サイト iWorld
<https://www.i-cafe.info/>

IBM i 7.5 技術資料
<https://www.ibm.com/docs/ja/i/7.5>

IBM Power ソフトウェアのダウンロードサイト (ESS)
<https://ibm.biz/powerdownload>

Fix Central (HW・SWのFix情報提供)
<https://www.ibm.com/support/fixcentral/>
IBM My Notifications (IBM IDの登録 [無償] が必要)
「IBM i」「9009-41G」などPTF情報の必要な製品を
選択して登録できます。

<https://www.ibm.com/support/mynotifications>
IBM i 各バージョンのライフサイクル
<https://www.ibm.com/support/pages/release-life-cycle>

IBM i 以外のSWのライフサイクル (個別検索)
<https://www.ibm.com/support/pages/lifecycle/>



ワークショップ、セッション、および資料は、IBMによって準備され、IBM独自の見解を反映したものです。それらは情報提供の目的のみで提供されており、いかなる読者に対しても法律的またはその他の指導や助言を意図したのではなく、またそのような結果を生むものでもありません。本資料に含まれている情報については、完全性と正確性を期するよう努力しましたが、「現状のまま」提供され、明示または暗示にかかわらずいかなる保証も伴わないものとします。本資料またはその他の資料の使用によって、あるいはその他の関連によって、いかなる損害が生じた場合も、IBMは責任を負わないものとします。本資料に含まれている内容は、IBMまたはそのサプライヤーやライセンス交付者からいかなる保証または表明を引き出すことを意図したもので、IBMソフトウェアの使用を規定する適用ライセンス契約の条項を変更することを意図したものでなく、またそのような結果を生むものでもありません。

本資料でIBM製品、プログラム、またはサービスに言及していても、IBMが営業活動を行っているすべての国でそれらが使用可能であることを暗示するものではありません。本資料で言及している製品リリース日付や製品機能は、市場機会またはその他の要因に基づいてIBM独自の決定権をもっていつでも変更できるものとし、いかなる方法においても将来の製品または機能が使用可能になると確約することを意図したものではありません。本資料に含まれている内容は、読者が開始する活動によって特定の販売、売上高の向上、またはその他の結果が生じると述べる、または暗示することを意図したもので、またそのような結果を生むものでもありません。パフォーマンスは、管理された環境において標準的なIBMベンチマークを使用した測定と予測に基づいています。ユーザーが経験する実際のスループットやパフォーマンスは、ユーザーのジョブ・ストリームにおけるマルチプログラミングの量、入出力構成、ストレージ構成、および処理されるワークロードなどの考慮事項を含む、数多くの要因に応じて変化します。したがって、個々のユーザーがここで述べられているものと同様の結果を得られると確約するものではありません。

記述されているすべてのお客様事例は、それらのお客様がどのようにIBM製品を使用したか、またそれらのお客様が達成した結果の実例として示されたものです。実際の環境コストおよびパフォーマンス特性は、お客様ごとに異なる場合があります。

IBM、IBM ロゴ、ibm.com、Db2、Rational、Power、POWER8、POWER9、AIXは、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。

他の製品名およびサービス名等は、それぞれIBMまたは各社の商標である場合があります。

現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtml をご覧ください。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Centrino、Intel Centrino ロゴ、Celeron、Xeon、Intel SpeedStep、Itanium、およびPentium は Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linuxは、Linus Torvaldsの米国およびその他の国における登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは Microsoft Corporationの米国およびその他の国における商標です。

ITILはAXELOS Limitedの登録商標です。

UNIXはThe Open Groupの米国およびその他の国における登録商標です。

JavaおよびすべてのJava関連の商標およびロゴは Oracleやその関連会社の米国およびその他の国における商標または登録商標です。