

# IBM i World 2022

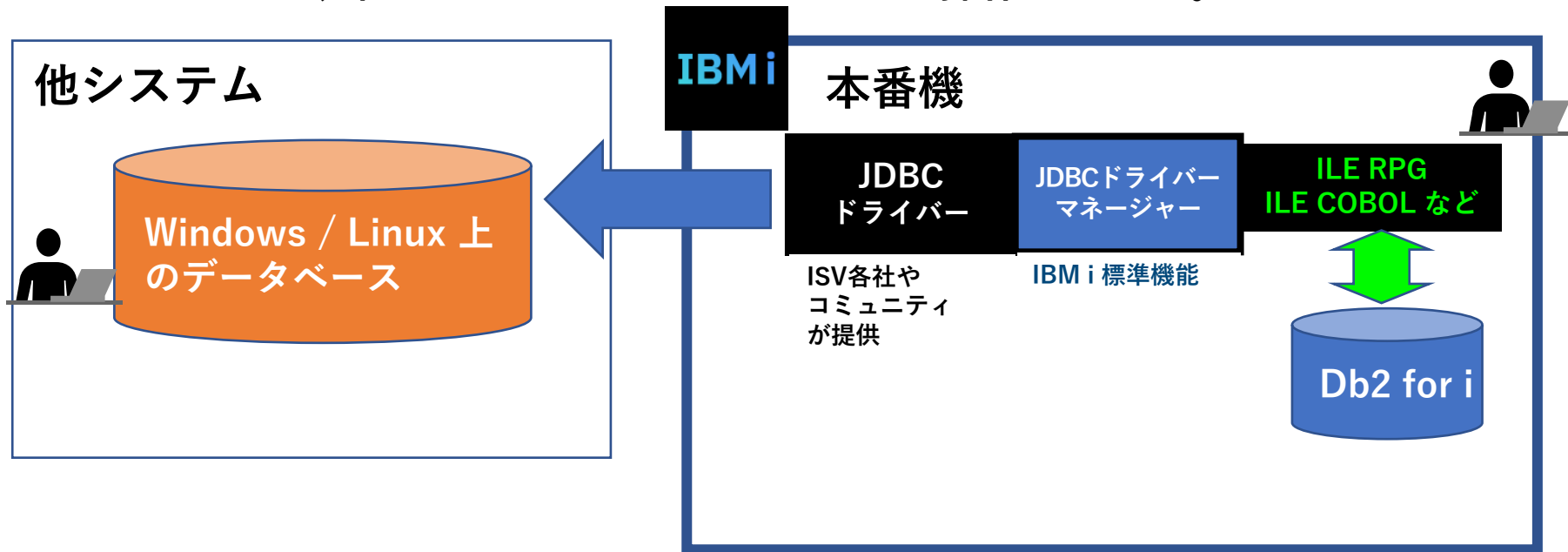
IBM i コンテンツ (2022年9月版)

## RPGからSQL ServerにアクセスOK! JDBCで他サーバー上のDBへアクセスする機能ご紹介

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 と他システムのデータベースに相互アクセスが可能
- ILE RPGから、他システム上のデータベースを操作できます。



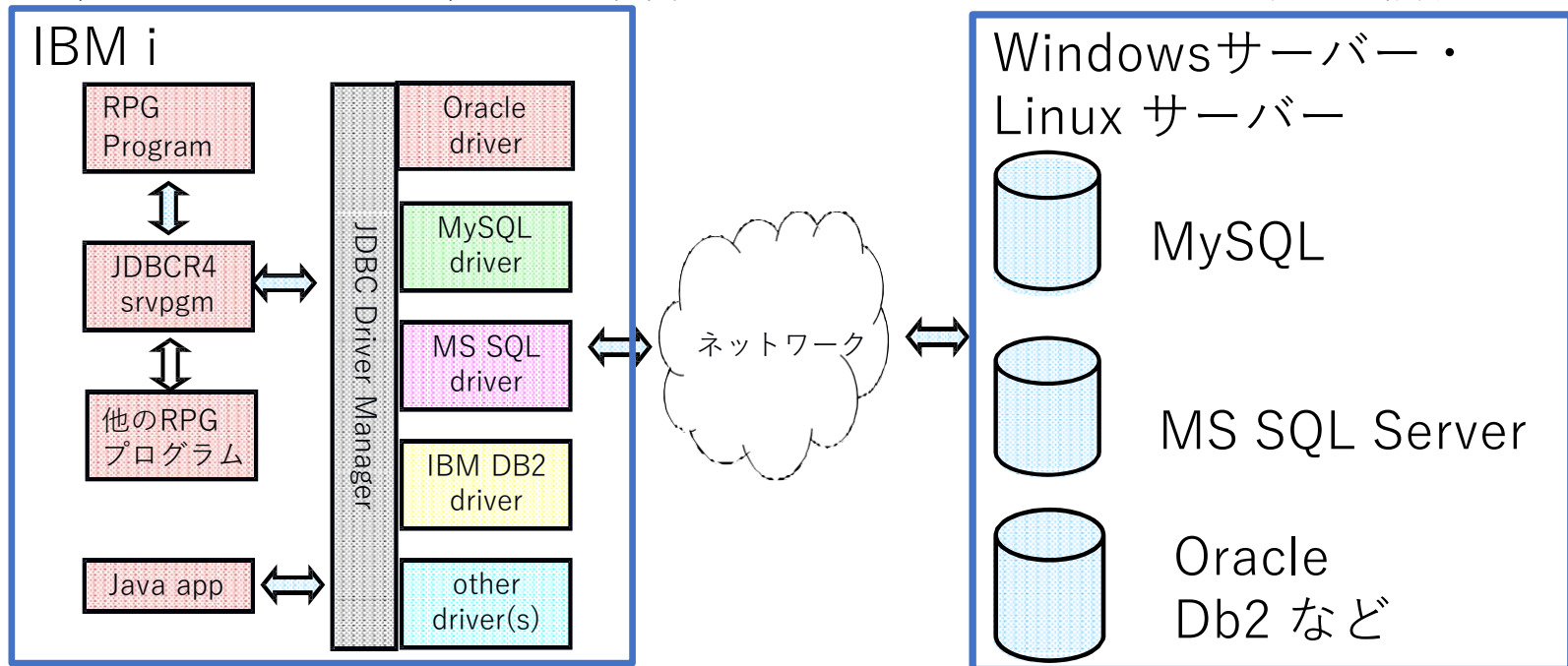
## 目次

1. JDBC概要と導入手順
2. 他サーバー上のデータベースへのアクセス手順
3. MySQLデータベースへの接続手順
4. MS SQL Serverへの接続手順
5. 補足情報

# 1. JDBC概要と導入手順 (1)

## JDBC機能の概要

- IBM iサーバー上で、各社やコミュニティが提供するJDBCドライバーを他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。
- ここでは、RPGプログラムから、JDBCを経由して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)

## 各社のJDBCドライバーについて

- 各社のドライバーについては、代表的なデータベースについては下記を使用してください。

➤ MySQL: mysql-connector-java-8.0.30.jar (MySQL 8.x用)

➤ Oracle: ojdbc14.jar

➤ SQL Server: jtds-1.3.1.jar JTDSドライバーは、2014年以降更新されていないので、最新のSQLサーバーの場合は、Microsoft製の専用ドライバーを使用してください (例: mssql-jdbc-11.2.0.jre8.jar)

➤ IBM Db2 for i: jt400.jar

➤ IBM Db2: db2jcc.jar

- 各社のドライバーは下記のリンクからダウンロードできます。

➤ Oracle Driver <http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html>

➤ MS SQL Server:

➤ J-TDS Open Source driver for MS SQL Server <http://jtds.sourceforge.net>

➤ Microsoft JDBC Driver 11.2 for SQL Server <https://docs.microsoft.com/ja-jp/sql/connect/jdbc/using-the-jdbc-driver?view=sql-server-ver16>

➤ MySQL Connector/J Driver 8.0.30: <https://search.maven.org/search?q=g:mysql%20AND%20a:mysql-connector-java>

➤ IBM Db2 UDB Driver (Windows or Linux DB2) <https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=swg-idsdjs>

➤ IBM Db2 Driver for IBM i (JTOpen) → IBM i PTFで提供 <https://sourceforge.net/projects/jt400/>

# 1. JDBC概要と導入手順 (3)

## JDBCドライバーのご紹介と導入手順

- JDBCドライバーは、純粋なJavaのType4 JDBCドライバーを利用します。
- Type4 JDBCドライバーは、Javaをサポートするどんなプラットフォームでも稼働します
- JDBC Type4 ドライバーは Java クラスです。ほとんどの場合、JAR ファイルにパッケージ化されています。
- 多くの場合、ドキュメント、ライセンス契約などの他のものと一緒に、JAR ファイルを ZIP または EXE ファイル内に配置します。

### 手順

1. ベンダーのパッケージを PC にダウンロード/解凍/インストールします。
2. JAR ファイル (複数可) を IBM i 上の IFS にアップロードします。
3. JAR ファイルを (完全な IFS パス名を使用して) CLASSPATH に追加します。
4. RPG が Java を呼び出すと、Java データベース マネージャーは CLASSPATH を使用して  
ドライバーを見つけます。

# 1. JDBC概要と導入手順 (4)

## 【例】MySQLのJDBCドライバー導入手順

### 手順

1. JDBC ドライバーを格納するための IFS フォルダを作成します。

```
> CRTDIR DIR('/java') DTAAUT(*RX) OBJAUT(*NONE)
ディレクトリが作成された。
> CRTDIR DIR('/java/jdbc') DTAAUT(*RX) OBJAUT(*NONE)
ディレクトリが作成された。
```

2. MySQL用の JDBCドライバーをダウンロードします。(URLはP5参照)
3. Windows PCで.ZIP をダブルクリックして解凍します。  
C:¥QTEMP フォルダに解凍するように指示します。
4. BINARY モードで FTP を使用して、C:¥QTEMP から mysql-connector-java-8.0.30.jar ファイルを  
コピーします。ファイルを IFS の /java/jdbc フォルダに移動します。
5. CLASSPATH を次のように設定します。  
ADDENVVAR ENVVAR(CLASSPATH) VALUE('/java/jdbc/mysql-connector-java-

8.0.30.jar')

\* IVM をロードする前に CLASSPATH を設定する必要があります。

## 2. 他サーバー上のデータベースへのアクセス手順(1)

各社のJDBCドライバーのプログラムからの識別について

- 各社のドライバーのロード後に、接続したいJDBCドライバーを識別するクラス名と、接続文字列が必要です。代表的なデータベースについては下記を使用してください。

(ドライバーの完全に資格のあるJavaクラス名)

```
SQL Server:      com.microsoft.sqlserver.jdbc.SQLServerDriver
Oracle:          oracle.jdbc.OracleDriver
MySQL:          com.mysql.cj.jdbc.Driver
Db2 for i:      com.ibm.as400.access.AS400JDBCdriver
Other Db2:      com.ibm.db2.jcc.DB2Driver
```

- 接続文字列は下記を使用してください。

```
SQL Server:      jdbc:sqlserver://myserver.example.com:1433
Oracle:          jdbc:oracle:thin:@myserver.example.com:1521:myDataBase
MySQL:          jdbc:mysql://myserver.example.com/myDataBase
DB2 for i:      jdbc:as400://myserver.example.com
Other DB2:      jdbc:db2://myserver.example.com:50000/myDataBase
```

- その他必要なプロパティ  
通常、ユーザー名とパスワード。  
他の属性 (\*SYS vs \*SQL、エラーの報告方法など)



## 2. 他サーバー上のデータベースへのアクセス手順(2)

この章のサンプルプログラムは、Mr. Scottの記述したJDBC4を参考に解説します。

- Scottのサンプルプログラムは下記のurlに詳細が掲載されています。  
<https://www.scottklement.com/jdbc/>
- JDBC4 は、Scott が RPG から JDBC を呼び出すタスクを単純化するために作成した RPG サービス プログラムです。
- これはもともと、System iNetwork Programming Tips ニュースレターの JDBC に関する記事のために書かれたものです。
- コードを (無料で) ダウンロードできる記事へのリンクが含まれます。  
この資料の RPG サンプル コードは、このサービス プログラムを使用します。
- このサービス プログラムを使用せずに Java メソッドを呼び出すこともできます  
独自のプロトタイプを作成するJava と RPG のデータ型の間で変換する独自のルーチンを作成できます。

## 2. 他サーバー上のデータベースへのアクセス手順(3)

### JDBCで接続するRPGのコードについて

- JDBC\_Connect() は、ユーザー ID とパスワードを使用してデータベースに接続します。
- 最初のパラメーターは、ロードする JDBC ドライバーの Java クラス名です。
- 2 番目のパラメーターは接続文字列です。
- 最後の 2 つのパラメーターは、ユーザー名とパスワードです。

```
/copy JDBC_H
D userid      s          50a
D passwd      s          50a
D conn        s          like (Connection)
.
.
userid = 'klemscot';
passwd = 'bigboy';

conn = JDBC_Connect( 'com.mysql.jdbc.Driver'
                    : 'jdbc:mysql://myserver.example.com/myDataBase'
                    : %trim(userid)
                    : %trim(passwd) );

if (conn = *NULL);
  errorMsg = 'Unable to connect to MYSQL database!';
  // show message to user.
endif;
```

別の種類のデータベースに接続するには、正しいクラスと接続文字列を指定します。

## 2. 他サーバー上のデータベースへのアクセス手順(4)

### Java SQL ステートメントのタイプ

- 即時ステートメント  
SQL 文字列はデータベースによって解釈され、すぐに実行されます。
- 準備されたステートメント  
SQL 文字列はデータベースによって「コンパイル」されます。その後、ステートメントは再コンパイルせずに複数回実行できます。ステートメントを実行する前に、プレースホルダーに値を入力できます。
- 呼び出し可能なステートメント  
ストアド プロシージャを呼び出す点を除いて、準備済みステートメントと非常によく似ています。

---- 上記のメソッドで使用されるステートメントの種類 ----

- “Query” ステートメント  
「結果セット」を返すステートメント  
(結果セット内の列に関するメタ情報が含まれることを除いて、カーソルに非常に似ています)。
- “Update” ステートメント  
結果セットを返さないステートメント。名前は正確ではありません。  
これは、DDL、INSERT、UPDATE など、結果セットを返さないものすべてに使用できます。
- “Call” ステートメント

## 2. 他サーバー上のデータベースへのアクセス手順(5)

### 即時ステートメントのルーチン

- JDBC\_ExecUpd( connection : sql statement string )  
Updateステートメント (結果セットを返さないステートメント) を実行します。  
Java executeUpdate() メソッドのラッパー。

影響を受けた行数を返します

行に影響を与えないステートメント (「テーブルの作成」など) の場合は 0、エラーが発生した場合は -1。

- JDBC\_ExecQry( connection : sql statement string )  
Queryステートメント (結果セットを返すステートメント) を実行します。  
Java executeQuery() メソッドのラッパー。

ResultSet オブジェクトを返します。

(カーソルのように - ステートメントの結果を取得するために使用) または、エラーが発生した場合は \*NULL。



## 2. 他サーバー上のデータベースへのアクセス手順(6)

### 即時 Update ステートメントのサンプル

- SQL ステートメントは、ターゲット データベースの構文である必要があります。
- これにより、SQL 標準に対する拡張機能を利用できます。
- JDBC には、ステートメントをデータベース中立にするのに役立つ「エスケープ」と呼ばれるツールがあります

```
... Make Connection Here ...

rc = JDBC_ExecUpd( conn : 'Create Table Item_Information'
                      + '('
                      + '  ItemNo      Dec(5,0) Not Null, '
                      + '  Count       Int Not Null, '
                      + '  LastChg     Timestamp '
                      + '             Default CURRENT_TIMESTAMP, '
                      + '  LastSold   Date Default Null, '
                      + '  TimeTest   Time Default Null, '
                      + '  Price       Dec(7,2) Not Null, '
                      + '  Description VarChar(25) not Null '
                      + ')' );

if (rc < 0);
  ErrMsg = 'Unable to CREATE table';
  // show error message to user
endif;
```

## 2. 他サーバー上のデータベースへのアクセス手順(7)

### 結果セットの操作

- **JDBC\_ExecQry( connection : sql statement string )**  
Queryステートメント (結果セットを返すステートメント) を実行します。  
Java executeQuery() メソッドのラッパー。  
結果セット、またはエラーの場合は \*NULL を返します。
- 結果セットを操作する場合は、次を使用します ---
- **JDBC\_nextRow( ResultSet )**  
結果セットで次に使用可能な行に進みます。  
成功した場合は \*ON を返し、結果セットの最後に到達した場合は \*OFF を返します。
- **JDBC\_getCol(ResultSet : ColNo)**  
成功した場合は \*ON を返し、結果セットの最後に到達した場合は \*OFF を返します。  
結果セット内の列の値を返します。  
列は序数で識別されます。(返される最初の列は番号 1、2 番目は番号 2 など)
- **JDBC\_getColByName( ResultSet : ColumnName )**  
列名を使用して、結果セット内の列の値を返します。
- **JDBC\_freeResult( ResultSet )**  
ResultSet を閉じます (埋め込み SQL でカーソルを閉じるのと同様)。  
結果セットが使用するメモリを解放します。

## 2. 他サーバー上のデータベースへのアクセス手順(8)

### 即時 Query ステートメントのサンプル

- SQLステートメントは、データベースサーバーのSQL構文を使う必要があります。

```
D ResSet          s          like(ResultSet)

... Connect as before ...
ResSet = JDBC_ExecQry( conn : 'Select Department, +
                               Employee_No, +
                               Employee_Name +
                               from Employee_Master +
                               order by Department' );

if (ResSet = *null);
  ErrMsg = 'Error running SELECT statement';
  // show error message to user
endif;

dow JDBC_nextRow(ResSet);
  Dept  = JDBC_getCol(ResSet: 1);
  EmpNo = %int(JDBC_getCol(ResSet: 2));
  Name  = JDBC_getCol(ResSet: 3);

  ... Print Dept, EmpNo, Name or whatever here ...

enddo;

JDBC_freeResult(ResSet);
```

## 2. 他サーバー上のデータベースへのアクセス手順(9)

### 準備されたステートメントのルーチン

- JDBC\_PrepStmt( connection : SQL statement string )
  - ー指定された SQL ステートメントの PreparedStatement Java オブジェクトを返します。
  - ーSQL ステートメントを「準備」します。
  - ーこれをステートメントの「コンパイル」と考えて、ステートメント内のコードを何度もすばやく実行できるようにします。
  - ープレースホルダー(「パラメーター マーカー」と呼ばれる)は変数値を表すことができるため、新しいステートメントを準備しなくてもステートメントを再実行できます。
  - ー「パラメーター マーカー」は、「SQL インジェクション攻撃」の回避にも役立ちます。
- JDBC\_ExecPrepUpd( PreparedStatement )
  - 結果セットを返さない準備済み ステートメントを実行します。
- JDBC\_ExecPrepQry( PreparedStatement )
  - 結果セットを返す準備済みステートメントを実行します
- JDBC\_FreePrepStmt( PreparedStatement )
  - 準備済み ステートメントによって使用されるメモリを解放します。



## 2. 他サーバー上のデータベースへのアクセス手順(10)

### 準備された「Query ステートメント」のサンプル

- SQLステートメントは、データベースサーバーのSQL構文を使う必要があります。

```
D Stmt          s          like (PreparedStatement)
D ResSet        s          like (ResultSet)
. . . Connect as before . . .

Stmt = JDBC_PrepStmt( conn : 'Select Department, +
                             Employee_No, +
                             Employee_Name +
                             from Employee_Master +
                             order by Department' );

if ( stmt = *null );
    // error occurred
endif

ResSet = JDBC_ExecPrepQry( Stmt );
if (ResSet = *null);
    // error occurred.
endif;

. . . Read the Result Set The Same Way You Did with
      an immediate statement . . .

JDBC_freeResult( ResSet );
JDBC_freePrepStmt( stmt );
```

## 2. 他サーバー上のデータベースへのアクセス手順(11)

### パラメーター・マーカー

- データを挿入する場所に「？」を置きます。次に、左から右にマーカーに番号を付け、番号で、次のルーチンを呼び出す設定します。
- 下記のコマンドが使用できます。
  - JDBC\_setString( stmt : parameter number : 'String Value' );
  - JDBC\_setInt( stmt : parameter number : integer value );
  - JDBC\_setDouble( stmt : parameter number : floating point value );
  - JDBC\_setDecimal( stmt : parameter number : decimal number );
  - JDBC\_setDate( stmt : parameter number : date field );
  - JDBC\_setTime( stmt : parameter number : time field );
  - JDBC\_setTimestamp( stmt : parameter number : timestamp field );

## 2. 他サーバー上のデータベースへのアクセス手順(12)

### パラメーター・マーカのサンプル

- SQLステートメントは、データベースサーバーのSQL構文を使う必要があります。

```
Stmt = JDBC_PrepStmt( conn : 'Select Department, +
                               Employee_Name +
                               from Employee_Master +
                               where Employee_No=?' );

if ( stmt = *null );
    // error occurred
endif

EmpNo = 1234;
JDBC_SetInt( stmt: 1: EmpNo );

ResSet = JDBC_ExecPrepQry( Stmt );
if (ResSet = *null);
    // error occurred.
endif;

. . . Read the Result Set The Same Way You Did with
      an immediate statement . . .

JDBC_freeResult( ResSet );
JDBC_freePrepStmt( stmt );
```



## 2. 他サーバー上のデータベースへのアクセス手順(13)

### 呼び出し可能なステートメントの実行

- 呼び出し可能ステートメントは、結果セットを返す準備済みステートメントによく似ています。
- 唯一の実際の違いは、データベースにアクセスする代わりにルーチン呼び出すことです。
- JDBC を介して実行されるすべての SQL と同様に、SQL ステートメントの構文はプラットフォームごとに異なります。
  
- JDBC\_PrepCall( Connection : Call Statement String )  
呼び出し可能なステートメントを準備します。
- JDBC\_RegisterOutParameter( CallableStatement: Parm No: DataType )  
パラメータの 1 つを使用してストアド プロシージャからデータを返すことを JDBC に通知します。  
デフォルトでは、すべてのパラメータは入力のみです。
- JDBC\_ExecCall( CallableStatement )  
呼び出し可能なステートメントを実行する
- JDBC\_FreeCallStmt( CallableStatement )  
呼び出し可能なステートメントによって使用されるメモリを解放します。

## 2. 他サーバー上のデータベースへのアクセス手順(14)

### 呼び出し可能なステートメントの結果

- ストアド プロシージャは、(「update」ステートメントのように) 更新カウントを返すか、1つ以上の結果セットを返すことができます。
- JDBC\_ExecCall() からの戻り値は、結果セットが返された場合は \*ON になり、それ以外の場合は \*OFF になります。
- JDBC\_getUpdateCount( CallableStatement )  
結果セットが返されない場合、影響を受ける行の数が表示されます (executeUpdate)。
- JDBC\_getResultSet( CallableStatement )  
1つ (または複数) の結果セットが返されると、結果セットが得られます。  
ResultSet オブジェクト (executeQuery など) を返します。
- JDBC\_getMoreResults( CallableStatement )  
複数の結果セットが返された場合は、次の結果セットに進みます。  
別の結果セットが見つかった場合は \*ON を返し、それ以外の場合は \*OFF を返し、ResultSet も閉じます。
- JDBC\_getString(), JDBC\_getInt(), JDBC\_getShort(), JDBC\_getBoolean()  
ストアド プロシージャから渡された出力パラメーターの値を取得します。

## 2. 他サーバー上のデータベースへのアクセス手順(15)

### ストアドプロシージャのサンプル

- SQLステートメントは、データベースサーバーのSQL構文を使う必要があります。

```
D stmt          s          like(CallableStatement)
D rs            s          like(ResultSet)
D IsResultSet   s          1n

      stmt = JDBC_PrepCall( conn
                          : 'call order_new(012001)' );

      IsResultSet = JDBC_execCall( stmt );

      dow IsResultSet;
        rs = JDBC_getResultSet( stmt );

        dow JDBC_nextRow(rs);
          field1 = JDBC_getCol(rs: 1);
          field2 = JDBC_getColByName(rs: 'SecondField');
          . . . etc . . .
        enddo;

      IsResultSet = JDBC_getMoreResults( stmt );
    enddo;

      JDBC_FreeCallStmt(stmt);
      JDBC_Close(conn);
```

### 3. MySQLデータベースへの接続手順(1)

この章では、下記の図のWindowsサーバー2019上のMySQLデータベースへのアクセスの手順をご紹介します。

#### Windows Server 2019



#### IBM i v7r4



データベース名:testsck 表名: rpgtest

```
mysql> show columns from rpgtest from test;
```

| Field       | Type           | Null | Key | Default           | Extra             |
|-------------|----------------|------|-----|-------------------|-------------------|
| ItemNo      | decimal (5, 0) | NO   | PRI | NULL              |                   |
| Count       | int            | NO   |     | NULL              |                   |
| LastChg     | timestamp      | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| LastSold    | date           | YES  |     | NULL              |                   |
| TimeTest    | time           | YES  |     | NULL              |                   |
| Price       | decimal (7, 2) | NO   |     | NULL              |                   |
| Description | varchar (25)   | NO   |     | NULL              |                   |

JDBC4のサンプルプログラム名: JDBCTEST  
QRPGLESRC/JDBCCLIBにあります。

1. MySQLデータベースに接続
2. データベース作成(testsck)
3. 表作成(rpgtest)
4. データの挿入
5. データ抽出、印刷

### 3. MySQLデータベースへの接続手順(2)

#### 前提

- サンプルプログラムをJDBCLIB（ライブラリー）に展開する。  
ー下記のコマンドで、JDBCR4のサービスの作成しておく  
CRTRPGMOD JDBCR4 SRCFILE(QRPGLESRC) DBGVIEW(\*LIST)  
CRTSRVPGM SRVPGM(JDBCR4) EXPORT(\*SRCFILE) SRCFILE(QSRVSRC)  
CRTBNDDIR BNDDIR(JDBCLIB/JDBC)  
ADDBNDDIRE BNDDIR(JDBCLIB/JDBC) OBJ((JDBCR4 \*SRVPGM))
- P7のMySQL用のJDBCドライバーをダウンロードして、環境変数（CLASSPATH）を設定する。



### 3. MySQLデータベースへの接続手順(3)

#### プログラム解説

#### 1. MySQLデータベースに接続

- JDBCドライバーがMySQLバージョンに合わせて、8.0.30用を使っているため、下記のように変更しました

```
オリジナル(MySQL 5.X用)
conn = JDBC_Connect( 'com.mysql.jdbc.Driver'
                    : 'jdbc:mysql://test.example.com/mysql'
                    : %trim(userid)
                    : %trim(passwrđ) );
```



```
変更後 (MySQL 8.X用)
conn = JDBC_Connect( 'com.mysql.cj.jdbc.Driver'
                    : 'jdbc:mysql://IPアドレス/mysql?'
                    + 'characterEncoding=utf8'
                    + '&useSSL=false'
                    : %trim(userid)
                    : %trim(passwrđ) );
```

## 3. MySQLデータベースへの接続手順(4)

### プログラム解説

#### 2. MySQLデータベース作成

- データベースは、下記のコマンドで作成されます。

```
rc = JDBC_ExecUpd( conn : 'Create Database testsck'  
                  + ' Default Character Set utf8'  
                  + ' Collate utf8_bin');
```

#### 3.表作成

- 下記のコマンドで、表は作成されます。

```
rc = JDBC_ExecUpd( conn : 'Create Table RpgTest'  
                  + '('  
                  + ' ItemNo Dec(5,0) Not Null, '  
                  + ' Count Int Not Null, '  
                  + ' LastChg Timestamp '  
                  + '   Default CURRENT_TIMESTAMP, '  
                  + ' LastSold Date Default Null, '  
                  + ' TimeTest Time Default Null, '  
                  + ' Price Dec(7,2) Not Null, '  
                  + ' Description VarChar(25) not Null, '  
                  + ' Primary Key( ItemNo )'  
                  + ')');
```

### 3. MySQLデータベースへの接続手順(5)

#### プログラム解説

#### 4.データの挿入

- データは、下記のコマンドで挿入されます。P19でご紹介したパラメーターマーカーを使用しています。

```
InsertRec( 10001: 887: %timestamp(): d'2005-01-02': t'12.34.56'  
          : 1003.05: 'Testing Product 10001');  
InsertRec( 10002: 888: %timestamp(): d'2005-03-04': t'18.34.43'  
          : 1003.05: 'Testing Product 10002');  
InsertRec( 10003: 999: %timestamp(): d'2001-12-29': t'18.34.43'  
          : 12.58: 'Testing Product 10003');
```

```
stmt = JDBC_PrepStmt(conn : 'Insert Into RpgTest '  
                    + '(ItemNo, Count, LastChg, LastSold, '  
                    + 'TimeTest, Price, Description)'  
                    + ' values (?, ?, ?, ?, ?, ?, ?)');
```

```
JDBC_setDecimal (stmt: 1: ItemNo);  
JDBC_setInt     (stmt: 2: Count);  
JDBC_setTimeStmp(stmt: 3: LastChg );  
JDBC_setDate   (stmt: 4: LastSold );  
JDBC_setTime   (stmt: 5: TimeTest );  
JDBC_setDecimal (stmt: 6: Price );  
JDBC_setString (stmt: 7: Description );  
rc = JDBC_ExecPrepUpd( stmt );  
JDBC_FreePrepStmt( stmt );
```

### 3. MySQLデータベースへの接続手順(6)

#### プログラム解説

#### 5.データの抽出と印刷

- データは、下記のコマンドで抽出・印刷されます。

##### データ抽出 と印刷

```
rs = jdbc_ExecQry( conn : 'Select ItemNo,Description'
                    + ' from RpgTest'
                    + ' where LastSold < DATE("2003-01-01")'
                    );
dow (jdbc_nextRow(rs));
  ItemNo = %int(jdbc_getCol(rs: 1));
  Desc   = jdbc_getCol(rs: 2);
except;
enddo;
```

```
OQSYSPRT  E
O          ItemNo      Z    5
O          Desc        32
```

### 3. MySQLデータベースへの接続手順(6)

#### 稼働手順

- ・ 前提

ILE-RPGのJDBCTESTプログラムをJDBCCLIBにコンパイル

- ・ プログラム起動


```
CALL JDBCTEST PARM('ユーザーID' 'パスワード')
```

スプール表示で、下記のように表示されればOKです。

```
File . . . . . : QSYSPRT
Control . . . . . _____
Find . . . . . _____
*...+...1...+...2...+...3...+...4...+...5...+..
10003 Testing Product 10003
```

- ・ MySQLの表は下記のようになってます。

testsck.rpgtest: 3 行合計 (約) >> 次 ◆ すべて表示

| ItemNo  | Count | LastChg             | LastSold   | TimeTest | Price    | Description           |
|--|-------|---------------------|------------|----------|----------|-----------------------|
| 10,001   | 887   | 2022-09-04 14:11:34 | 2005-01-02 | 12:34:56 | 1,003.05 | Testing Product 10001 |
| 10,002   | 888   | 2022-09-04 14:11:34 | 2005-03-04 | 18:34:43 | 1,003.05 | Testing Product 10002 |
| 10,003   | 999   | 2022-09-04 14:11:34 | 2001-12-29 | 18:34:43 | 12.58    | Testing Product 10003 |

## 4. MS SQL Serverへの接続手順 (1)

この章では、下記の図のWindowsサーバー2019上のMS SQL Serverデータベースへのアクセスの手順をご紹介します。

### Windows Server 2019



### IBM i 7.4



データベース名:TESTDB 表名: dbo.person

カラム: + 追加 × 削除 ▲ 上へ ▼ 下へ

| # | 名前         | データ型     | 長さ/セット | 符号なし                     | NULLを...                            |
|---|------------|----------|--------|--------------------------|-------------------------------------|
| 1 | ssn_ein_id | INT      |        | <input type="checkbox"/> | <input type="checkbox"/>            |
| 2 | last_name  | NVARCHAR | 50     | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 3 | first_name | NVARCHAR | 50     | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 4 | birth_date | DATE     |        | <input type="checkbox"/> | <input type="checkbox"/>            |

JDBC R4のサンプルプログラム名: JDBCTEST3  
QRPGLESRC/JDBCCLIBにあります。

1. MS SQL Serverデータベースに接続
2. データベース作成(TESTDB)
3. 表作成(dbo.person)
4. データの挿入
5. データ抽出、印刷

## 4. MS SQL Serverへの接続手順 (2)

### 前提

- サンプルプログラムをJDBCLIB（ライブラリー）に展開する。  
ー下記のコマンドで、JDBCR4のサービスの作成しておく  
CRTRPGMOD JDBCR4 SRCFILE(QRPGLESRC) DBGVIEW(\*LIST)  
CRTSRVPGM SRVPGM(JDBCR4) EXPORT(\*SRCFILE) SRCFILE(QSRVSRG)  
CRTBNDDIR BNDDIR(JDBCLIB/JDBC)  
ADDBNDDIRE BNDDIR(JDBCLIB/JDBC) OBJ((JDBCR4 \*SRVPGM))
- MS SQL Server 用の最新JDBCドライバーをダウンロードする。(下記のurl)から  
Microsoft JDBC Driver 11.2.0 for SQL Server (zip)をダウンロードする。  
<https://docs.microsoft.com/ja-jp/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server?view=sql-server-ver16>
- 解凍して、Java8用の、「mssql-jdbc-11.2.0.jre8.jar」を、/JAVA/JDBCへアップロードする
- 環境変数（CLASSPATH）を設定する。  
ADDENVVAR ENVVAR(CLASSPATH) VALUE('/java/jdbc/mssql-jdbc-11.2.0.jre8.jar')

## 4. MS SQL Serverへの接続手順 (3)

### プログラム解説

#### 1. MS SQL Server データベースに接続

- JDBCドライバーは、SQL Serverバージョンに合わせて、MicroSoft製を使っているため、下記のように変更しました。また、Jdbc class、url、ユーザーIDなどのパラメータ渡しではなく、プログラムに組みこみました。

オリジナル

```
JDBC_setProp(prop: 'User' : %trim(userid));  
JDBC_setProp(prop: 'Password': %trim(passwrd));  
JDBC_setProp(prop: 'DatabaseName': %trim(database));  
conn = JDBC_ConnProp( %trim(driver)  
: %trim(url)  
: prop );
```

変更後

```
JDBC_setProp(prop: 'userName' : 'ユーザーID');  
JDBC_setProp(prop: 'password' : 'パスワード');  
JDBC_setProp(prop: 'databaseName' : 'TESTDB');  
conn =JDBC_ConnProp('com.microsoft.sqlserver.jdbc.SQLServerDriver'  
: 'jdbc:sqlserver://IPアドレス:1433;'  
+ 'encrypt=false;trustServerCertificate=false'  
: prop );
```



## 4. MS SQL Serverへの接続手順 (4)

### 事前準備

#### 2. MS SQL Serverデータベース作成

- データベースは、MS SQL Server側で、先に作成しておきます。

```
CREATE DATABASE TESTDB;
```

#### 3.表作成とサンプルデータの入力

- 表についても、MS SQL Server側で、作成しておきます。

```
CREATE TABLE "person" (  
    "ssn_ein_id" INT NOT NULL,  
    "last_name" NVARCHAR(50) NULL DEFAULT NULL COLLATE 'Japanese_CI_AS',  
    "first_name" NVARCHAR(50) NULL DEFAULT NULL COLLATE 'Japanese_CI_AS',  
    "birth_date" DATE NOT NULL  
);
```

## 4. MS SQL Serverへの接続手順 (5)

### 事前準備

#### 4.データの挿入

- データは、表(person)に先に入力しておきます。

```
INSERT INTO "person" ("ssn_ein_id", "last_name", "first_name", "birth_date") VALUES  
  (1, 'Taro', 'Yamada', '1980-01-01'),  
  (2, 'Hanako', 'Tanaka', '1979-12-30'),  
  (3, 'yukiko', 'Takahashi', '2000-01-01'),  
  (4, 'Takao', 'Saito', '1980-01-01');
```

- 表の内容は下記のようになります。

TESTDB.person: 4 行合計

| ssn_ein_id | last_name | first_name | birth_date |
|------------|-----------|------------|------------|
| 1          | Taro      | Yamada     | 1980-01-01 |
| 2          | Hanako    | Tanaka     | 1979-12-30 |
| 3          | yukiko    | Takahashi  | 2000-01-01 |
| 4          | Takao     | Saito      | 1980-01-01 |

## 4. MS SQL Serverへの接続手順 (6)

### プログラム解説

#### 5.データの抽出と印刷(1/3)

- データは、下記のコマンドで抽出・印刷されます。

```
// Query the database 1
//      sql文は実行時のパラメータです
rs = jdbc_ExecQry( conn
                  : %trim(sql)
                  );
rsmd = jdbc_GetMetaData(rs);
colcnt = jdbc_GetColCount(rsmd);
// Get column names
idx = 1;
Row1 = jdbc_getColName(rsmd: idx)
      + ' '
      + jdbc_getColTypeName(rsmd: idx)
      + '('
      + %trim(%editc(jdbc_getColDspSize(rsmd: idx):'Z'))
      + ')';
```

```
colcnt = jdbc_GetColCount(rsmd);
// Get column names  2
idx = 1;
Row1 = jdbc_getColName(rsmd: idx)
      + ' '
      + jdbc_getColTypeName(rsmd: idx)
      + '('
      + %trim(%editc(jdbc_getColDspSize(rsmd: idx):'Z'))
      + ')';
dow (idx < colcnt);
  idx = idx + 1;
  Row1 = %trim(Row1)
        + ', '
        + jdbc_getColName(rsmd: idx)
        + ' '
        + jdbc_getColTypeName(rsmd: idx)
        + '('
        + %trim(%editc(jdbc_getColDspSize(rsmd:
idx):'Z'))
        + ')';
```

## 4. MS SQL Serverへの接続手順 (7)

### プログラム解説

#### 5.データの抽出と印刷(2/3)

- データは、下記のコマンドで抽出・印刷されます。

```
dow (idx < colcnt);
  idx = idx + 1;
  Row1 = %trim(Row1)
    + ','
    + jdbc_getColName(rsmd: idx)
    + ''
    + jdbc_getColTypName(rsmd: idx)
    + '('
    + %trim(%editc(jdbc_getColDspSize(rsmd: idx):'Z'))
    + ')';
enddo;
except;
// Get rows from table
dow (jdbc_nextRow(rs));
  idx = 1;
  Row1 = jdbc_getCol(rs: idx);
  dow (idx < colcnt);
    idx = idx + 1;
    Row1 = %trim(Row1)
```

```
enddo;
except;

// Get rows from table
dow (jdbc_nextRow(rs));
  idx = 1;
  Row1 = jdbc_getCol(rs: idx);
  dow (idx < colcnt);
    idx = idx + 1;
    Row1 = %trim(Row1)
      + ','
      + jdbc_getCol(rs: idx);
  enddo;
except;
enddo;

jdbc_freeResult(rs);

jdbc_close(conn);
return;
```

## 4. MS SQL Serverへの接続手順 (8)

### プログラム解説

#### 5.データの抽出と印刷(3/3)

- データは、下記のコマンドで抽出・印刷されます。

```
+ ', '  
      + jdbc_getCol(rs: idx);  
      enddo;  
      except;  
      enddo;  
  
      jdbc_freeResult(rs);  
  
      jdbc_close(conn);  
      return;  
/end-free
```

```
OQSYSPRT  E  
O          Row1          132
```

## 4. MS SQL Serverへの接続手順 (9)

### 稼働手順

- ・ 前提  
ILE-RPGのJDBCTEST3プログラムをJDBCCLIBにコンパイル
- ・ プログラム起動  
CALL JDBCTEST3 PARM('select ssn\_ein\_id,last\_name, first\_name, birth\_date  
from dbo.person')
- ・ スプール表示で、下記のように表示されればOKです。

```
File      : QSYSPRT
Control   : _____
Find      : _____
*  + 1  + 2  + 3  + 4  + 5  + 6  + 7  + 8  + 9  +
ssn_ein_id int(11), last_name nvarchar(50), first_name nvarchar(50), birth_date date(10)
1, Taro, Yamada, 1980-01-01
2, Hanako, Tanaka, 1979-12-30
3, yukiko, Takahashi, 2000-01-01
4, Takao, Saito, 1980-01-01
```

## 5. 補足情報

### (1) Scott Klement 著 [Accessing External Database]資料

当資料は、基本的にはこの資料を日本語翻訳して、最新情報を追加したものです。

<https://www.scottklement.com/jdbc/>

- ・ MySQLとMS SQL Serverについては最新バージョンでテストして、更新しました。
- ・ OracleとDb2に関しては、本文の内容をそのまま使用しています。

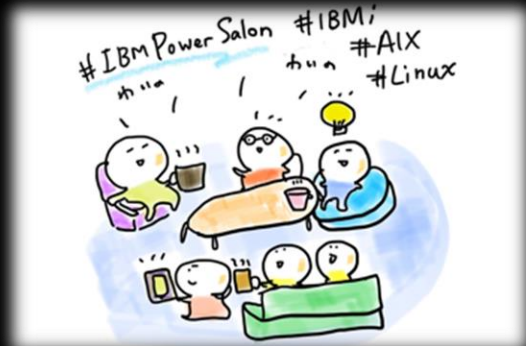
### (2) MySQL Connector/J 8.0 Developer Guide

<https://dev.mysql.com/doc/connector-j/8.0/en/>

### (3) MS SQL Server jdbcドライバーの考慮点

<https://docs.microsoft.com/ja-jp/sql/connect/jdbc/using-the-jdbc-driver?view=sql-server-ver16>

# IBM Power Salon ご案内



IBM Powerユーザーのための自由な語り場。  
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](https://ibm.biz/Powersalon-webex)

お問い合わせ：[NO1POWER@jp.ibm.com](mailto:NO1POWER@jp.ibm.com)



|      |                          |                            |
|------|--------------------------|----------------------------|
| 第7回  | 5月11日(水)<br>9:00-10:00   | IBM i<br>エム・シーシー食品株式会社 石川様 |
| 第8回  | 6月 8日(水)<br>9:00-10:00   | IBM i World 2022 Q&A       |
| 第9回  | 7月 13日(水)<br>9:00-10:00  | IBM i<br>日本サニバック様 ご講演      |
| -    | 8月 休会                    |                            |
| 第10回 | 9月 14日(水)<br>9:00-10:00  | お客様講演予定                    |
| 第11回 | 10月 8日(水)<br>9:00-10:00  |                            |
| 第12回 | 11月 9日(水)<br>9:00-10:00  |                            |
| 第13回 | 12月 14日(水)<br>9:00-10:00 |                            |



## 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<br>(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](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](http://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。