



# サードパーティ・ソフトウェア用 インターシステムズ実装リファ レンス

Version 2023.1  
2024-01-02

サードパーティ・ソフトウェア用インターシステムズ実装リファレンス  
InterSystems IRIS Data Platform Version 2023.1 2024-01-02  
Copyright © 2024 InterSystems Corporation  
All rights reserved.

InterSystems®, HealthShare Care Community®, HealthShare Unified Care Record®, IntegratedML®, InterSystems Caché®, InterSystems Ensemble®, InterSystems HealthShare®, InterSystems IRIS®, および TrakCare は、InterSystems Corporation の登録商標です。HealthShare® CMS Solution Pack™ HealthShare® Health Connect Cloud™, InterSystems IRIS for Health™, InterSystems Supply Chain Orchestrator™, および InterSystems TotalView™ For Asset Management は、InterSystems Corporation の商標です。TrakCare は、オーストラリアおよび EU における登録商標です。

ここで使われている他の全てのブランドまたは製品名は、各社および各組織の商標または登録商標です。

このドキュメントは、インターシステムズ社(住所: One Memorial Drive, Cambridge, MA 02142)あるいはその子会社が所有する企業秘密および秘密情報を含んでおり、インターシステムズ社の製品を稼働および維持するためにのみ提供される。この発行物のいかなる部分も他の目的のために使用してはならない。また、インターシステムズ社の書面による事前の同意がない限り、本発行物を、いかなる形式、いかなる手段で、その全てまたは一部を、再発行、複製、開示、送付、検索可能なシステムへの保存、あるいは人またはコンピュータ言語への翻訳はしてはならない。

かかるプログラムと関連ドキュメントについて書かれているインターシステムズ社の標準ライセンス契約に記載されている範囲を除き、ここに記載された本ドキュメントとソフトウェアプログラムの複製、使用、廃棄は禁じられている。インターシステムズ社は、ソフトウェアライセンス契約に記載されている事項以外にかかるソフトウェアプログラムに関する説明と保証をするものではない。さらに、かかるソフトウェアに関する、あるいはかかるソフトウェアの使用から起こるいかなる損失、損害に対するインターシステムズ社の責任は、ソフトウェアライセンス契約にある事項に制限される。

前述は、そのコンピュータソフトウェアの使用およびそれによって起こるインターシステムズ社の責任の範囲、制限に関する一般的な概略である。完全な参照情報は、インターシステムズ社の標準ライセンス契約に記載され、そのコピーは要望によって入手することができる。

インターシステムズ社は、本ドキュメントにある誤りに対する責任を放棄する。また、インターシステムズ社は、独自の裁量にて事前通知なしに、本ドキュメントに記載された製品および実行に対する代替と変更を行う権利を有する。

インターシステムズ社の製品に関するサポートやご質問は、以下にお問い合わせください:

InterSystems Worldwide Response Center (WRC)  
Tel: +1-617-621-0700  
Tel: +44 (0) 844 854 2917  
Email: support@InterSystems.com

# 目次

1 サードパーティ・ソフトウェアの概要 .....	1
2 JDBC ドライバのサポート .....	3
2.1 JDBC と InterSystems JDBC ドライバ .....	3
2.1.1 インストールと構成 .....	4
2.2 JDBC ドライバの準拠 .....	4
2.2.1 必須の java.sql インタフェース .....	4
2.2.2 オプションの java.sql インタフェース .....	5
2.2.3 java.sql の例外 .....	5
2.2.4 必須の javax.sql インタフェース .....	6
2.2.5 オプションの javax.sql インタフェース .....	6
2.3 バリエーションおよびサポートされないオプションのメソッド .....	7
2.3.1 CallableStatement : サポートされないメソッド .....	7
2.3.2 Connection : サポートされないまたは制限のあるメソッド .....	8
2.3.3 DatabaseMetaData : バリエーション・メソッド .....	9
2.3.4 Driver : サポートされないメソッド .....	10
2.3.5 PreparedStatement : サポートされないメソッド .....	10
2.3.6 ResultSet : サポートされないまたは制限のあるメソッド .....	10
2.3.7 Statement : サポートされないまたは制限のあるメソッド .....	11
2.4 インターシステムズの拡張機能と拡張メソッド .....	12
2.4.1 CallableStatement getBinaryStream() 拡張メソッド .....	13
2.4.2 ConnectionPoolDataSource の拡張メソッドと拡張機能 .....	13
2.4.3 DataSource の拡張メソッドと拡張機能 .....	14
3 Python DB-API のサポート .....	19
4 Hibernate のサポート .....	21
4.1 Hibernate と InterSystems Hibernate 言語 .....	21
4.1.1 Hibernate を使用する状況 .....	21
4.2 インストールと構成 .....	22
4.2.1 要件 .....	22
4.2.2 ディレクトリ .....	22
4.2.3 システムの設定 .....	22
4.2.4 Hibernate 構成 .....	23
4.3 言語ファイルの場所 .....	24



# 1

## サードパーティ・ソフトウェアの概要

このドキュメントで取り上げる内容の詳細なリストは、“[目次](#)”を参照してください。

インターシステムズでは、以下のサードパーティのドライバおよびインタフェースのネイティブの実装を提供します。

### InterSystems JDBC ドライバ

InterSystems JDBC ドライバは、JDBC 標準に完全に準拠したタイプ 4 の実装です。

JDBC ドライバの準拠および拡張機能の詳細は、“[JDBC ドライバのサポート](#)”を参照してください。ここには、オプションの機能すべてのサポート・レベル、InterSystems IRIS 固有の追加機能すべてのリストも記載されています。

以下の関連ドキュメントも参照してください。

- ・ 機能および使用法の詳細は、“[InterSystems ソフトウェアでの Java の使用法](#)”を参照してください。

### InterSystems Hibernate 言語

InterSystems Hibernate 言語は、Hibernate 言語インタフェースの実装です。SQL の実装はベンダごとに若干異なるので、ベンダは Hibernate 言語インタフェースを使用することで、特定のデータベースに対して Hibernate のカスタム・マッピングを作成できるようになります。

詳細は、“[Hibernate のサポート](#)”を参照してください。



# 2

## JDBC ドライバのサポート

InterSystems IRIS® JDBC ドライバは、JDBC 4.2 標準に完全に準拠したタイプ 4 の実装です。ここでは、JDBC 4.2 API のすべてのクラスとインタフェースをリストして、それぞれのサポート・レベルを示し、インターシステムズ固有の機能すべてについて説明します。以下の項目について説明します。

- ・ [JDBC と InterSystems JDBC ドライバ](#) – JDBC ドライバの概要とリソースへのリンクを提供します。
- ・ [JDBC ドライバの準拠](#) – JDBC 標準で指定されているすべてのクラスとインタフェースをリストし、現在のサポート・レベルを示します。
- ・ [バリエーションおよびサポートされていないオプションのメソッド](#) – 標準で許可されているバリエーションを含め、クラスに関する詳細を示します。
- ・ [インターシステムズの拡張機能と拡張メソッド](#) – 標準の JDBC API に対するインターシステムズの拡張機能をリストして説明します。

“[Connecting Your Application to InterSystems IRIS](#)” には、サンプル・コードを含め、JDBC を使用して Java アプリケーションから InterSystems IRIS サーバに接続する手順が示されています。

### 2.1 JDBC と InterSystems JDBC ドライバ

Java JDBC API は、ベンダに依存しないデータベース接続の業界標準です。これは、Java アプリケーションが、サポートされているプラットフォーム上のデータ・ソースに接続したり、それらのデータ・ソースに対して SQL を使用して操作を実行したりするための信頼性の高い方法を提供します。

InterSystems JDBC は、可能な限り最高のパフォーマンスを実現できるように、タイプ 4 のドライバに実装されています。タイプ 4 とは、データベースに直接アクセスできる純正 Java ドライバであり、クライアント JVM 内にインストールされ、外部ソフトウェアのサポートを必要としないことを意味します。これは、JDBC 4.2 API の仕様に完全に準拠しています。また、必要なインタフェースをすべてサポートし、JDBC 4.2 のすべてのガイドラインおよび要件に従っています。InterSystems IRIS は、SQL の例外処理の強化、各国語文字セット変換、および XML データ型以外のすべての機能をサポートしています。

API の機能および使用法の詳細は、“[InterSystems ソフトウェアでの Java の使用法](#)” を参照してください。そのドキュメントでは、JDBC ドライバによって有効になるすべての InterSystems IRIS Java テクノロジーの概要も示しています (“[InterSystems Java 接続オプション](#)” を参照してください)。

## 2.1.1 インストールと構成

InterSystems JDBC ドライバは、標準の InterSystems IRIS インストール・パッケージに含まれています。追加のインストールや設定の手順は不要です。クライアントの要件および使用法については、“[InterSystems ソフトウェアでの Java の使用法](#)” の “[クライアントとサーバ間の構成](#)” を参照してください。

JDBC ドライバを使用して InterSystems IRIS とアプリケーション間の接続を作成する方法の詳細は、“[Connecting Your Application to InterSystems IRIS](#)” を参照してください。

InterSystems IRIS がインストールされていないシステムで JDBC ドライバを使用する場合は、GitHub の [InterSystems IRIS ドライバのページ](#) からドライバ・パッケージをダウンロードできます。

## 2.2 JDBC ドライバの準拠

ここでは、各 JDBC インタフェースのサポート・レベルに関する情報を提供します。

### 2.2.1 必須の java.sql インタフェース

以下のインタフェースは実装する必要があります。クラスによっては、データベースでサポートされない機能に実装が依存する場合はオプションになるメソッドが含まれているものがあります。標準の実装という注釈は、クラスの一般的な実装が変更なしに使用されていることを示します。

- ・ `java.sql.CallableStatement` — 許可されるバリエントと共に実装されます (“[CallableStatement](#) : サポートされないメソッド” および “[CallableStatement getBinaryStream\(\) 拡張メソッド](#)” を参照)。
- ・ `java.sql.ClientInfoStatus` — 標準の実装。
- ・ `java.sql.Connection` — 許可されるバリエントと共に実装されます (“[Connection](#) : サポートされないまたは制限のあるメソッド” を参照)。
- ・ `java.sql.DatabaseMetaData` — 許可されるバリエントと共に実装されます (“[DatabaseMetaData](#) : バリエント・メソッド” を参照)。
- ・ `java.sql.Date` — 標準の実装。
- ・ `java.sql.Driver` — 許可されるバリエントと共に実装されます (“[Driver](#) : サポートされないメソッド” を参照)。
- ・ `java.sql.DriverManager` — 標準の実装。
- ・ `java.sql.DriverPropertyInfo` — 標準の実装。
- ・ `java.sql.ParameterMetaData` — すべてのメソッドが完全にサポートされます。
- ・ `java.sql.PreparedStatement` — 許可されるバリエントと共に実装されます (“[PreparedStatement](#) : サポートされないメソッド” を参照)。
- ・ `java.sql.ResultSet` — 許可されるバリエントと共に実装されます (“[ResultSet](#) : サポートされないまたは制限のあるメソッド” を参照)。
- ・ `java.sql.ResultSetMetaData` — すべてのメソッドが完全にサポートされます。
- ・ `java.sql.RowIdLifetime` — 標準の実装。
- ・ `java.sql.SQLPermission` — 標準の実装。
- ・ `java.sql.Statement` — 許可されるバリエントと共に実装されます (“[Statement](#) : サポートされないまたは制限のあるメソッド” を参照)。



- ・ `java.sql.Time` — 標準の実装。
- ・ `java.sql.Timestamp` — 標準の実装。
- ・ `java.sql.Types` — 標準の実装。
- ・ `java.sql.Wrapper` — すべてのメソッドが完全にサポートされます。

## 2.2.2 オプションの `java.sql` インタフェース

オプションの `java.sql` インタフェースをすべて以下に示します。イタリック体になっている項目は実装されません。

- ・ `java.sql.Array`
- ・ `java.sql.Blob` — すべてのメソッドが完全にサポートされます。
- ・ `java.sql.Clob` — すべてのメソッドが完全にサポートされます。
- ・ `java.sql.NClob` — すべてのメソッドが完全にサポートされます。
- ・ `java.sql.Ref`
- ・ `java.sql.RowId` — すべてのメソッドが完全にサポートされます。
- ・ `java.sql.Savepoint` — すべてのメソッドが完全にサポートされます。
- ・ `java.sql.SQLData`
- ・ `java.sql.SQLInput`
- ・ `java.sql.SQLOutput`
- ・ `java.sql.SQLXML`
- ・ `java.sql.Struct`

## 2.2.3 `java.sql` の例外

InterSystems JDBC ドライバは、以下の例外のみをスローします。

- ・ `java.sql.BatchUpdateException`
- ・ `java.sql.SQLException`
- ・ `java.sql.SQLWarning`

以下の例外は、完全を期すためにここにリストしていますが、必要ではなく、使用することはありません。

- ・ `DataTruncation`
- ・ `SQLClientInfoException`
- ・ `SQLDataException`
- ・ `SQLFeatureNotSupportedException`
- ・ `SQLIntegrityConstraintViolationException`
- ・ `SQLInvalidAuthorizationSpecException`
- ・ `SQLNonTransientConnectionException`
- ・ `SQLNonTransientException`
- ・ `SQLRecoverableException`

- ・ `SQLException`
- ・ `TimeoutException`
- ・ `TransactionRollbackException`
- ・ `TransientConnectionException`
- ・ `TransientException`

## 2.2.4 必須の `javax.sql` インタフェース

以下の必須インタフェースがサポートされています。標準の実装という注釈は、クラスの一般的な実装が変更なしに使用されていることを示します。

- ・ `javax.sql.ConnectionEvent` — 標準の実装。
- ・ `javax.sql.DataSource` — 拡張機能や追加メソッドと共に実装されます（詳細は“[DataSource の拡張メソッドと拡張機能](#)”を参照）。
- ・ `javax.sql.RowSetEvent` — 標準の実装。
- ・ `javax.sql.StatementEvent` — 標準の実装。

## 2.2.5 オプションの `javax.sql` インタフェース

オプションの `javax.sql` インタフェースをすべて以下に示します。イタリック体になっている項目は実装されません。

- ・ `javax.sql.CommonDataSource` — 実装されません。代わりに `javax.sql.DataSource` を使用してください（関連情報は、[DataSource の拡張メソッドと拡張機能](#)を参照）。
- ・ `javax.sql.ConnectionEventListener` — すべてのメソッドが完全にサポートされます。
- ・ `javax.sql.ConnectionPoolDataSource` — バリエントや追加メソッドと共に実装されます（詳細は“[ConnectionPoolDataSource の拡張メソッドと拡張機能](#)”を参照）。
- ・ `javax.sql.PooledConnection` — すべてのメソッドが完全にサポートされます。
- ・ `javax.sql.Rowset`
- ・ `javax.sql.RowSetInternal`
- ・ `javax.sql.RowSetListener`
- ・ `javax.sql.RowSetMetaData`
- ・ `javax.sql.RowSetReader`
- ・ `javax.sql.RowSetWriter`
- ・ `javax.sql.StatementEventListener`
- ・ `javax.sql.XAConnection`
- ・ `javax.sql.XADataSource`

## 2.3 バリエントおよびサポートされないオプションのメソッド

以下のインタフェースは、InterSystems JDBC ドライバがサポートしないオプションのメソッドまたは標準以外の方法で実装されているメソッドを含みます。

- `CallableStatement` : サポートされないメソッド
- `Connection` : サポートされないまたは制限のあるメソッド
- `DatabaseMetaData` : バリエント・メソッド
- `Driver` : サポートされないメソッド
- `PreparedStatement` : サポートされないメソッド
- `ResultSet` : サポートされないまたは制限のあるメソッド
- `Statement` : サポートされないまたは制限のあるメソッド

### 2.3.1 `CallableStatement` : サポートされないメソッド

#### サポートされないオプションのメソッド

`java.sql.CallableStatement` は、以下のオプションのメソッドをサポートしません。

- `getArray()`  
  

```
Array getArray(int i)
Array getArray(String parameterName)
```
- `getObject()`  
  

```
Object getObject(int i, java.util.Map map)
Object getObject(String parameterName, java.util.Map map)
```
- `getRef()`  
  

```
Ref getRef(int i)
Ref getRef(String parameterName)
```
- `getRowId()` および `setRowId()`  
  

```
java.sql.RowId getRowId(int i)
java.sql.RowId getRowId(String parameterName)

void setRowId(String parameterName, java.sql.RowId x)
```
- `getURL()` および `setURL()`  
  

```
java.net.URL getURL(int i)
java.net.URL getURL(String parameterName)

void setURL(String parameterName, java.net.URL val)
```
- `getSQLXML()` および `setSQLXML()`  
  

```
java.sql.SQLXML getSQLXML(int parameterIndex)
java.sql.SQLXML getSQLXML(String parameterName)

void setSQLXML(String parameterName, java.sql.SQLXML xmlObject)
```

注釈 `java.sql.CallableStatement` クラスはインターシステムズの拡張メソッドも 1 つサポートしますが、これについては別の場所で説明しています (“[CallableStatement getBinaryStream\(\) 拡張メソッド](#)” を参照)。

## 2.3.2 Connection : サポートされないまたは制限のあるメソッド

### サポートされないオプションのメソッド

`java.sql.Connection` は、以下のオプションのメソッドをサポートしません。

- `abort()`  
`void abort(Executor executor)`
- `createArrayOf()`  
`java.sql.Array createArrayOf(String typeName, Object[] elements)`
- `createBlob()`  
`Blob createBlob()`
- `createClob()`  
`Clob createClob()`
- `createNClob()`  
`java.sql.NClob createNClob()`
- `createSQLXML()`  
`java.sql.SQLXML createSQLXML()`
- `createStruct()`  
`java.sql.Struct createStruct(String typeName, Object[] attributes)`
- `getTypeMap()`  
`java.util.Map getTypeMap()`
- `setTypeMap()`  
`void setTypeMap(java.util.Map map)`

### 制限のあるオプションのメソッド

以下のオプションの `java.sql.Connection` メソッドは、制限事項または制約付きで実装されています。

- `prepareCall()`  
`TYPE_FORWARD_ONLY` のみが `resultSetType` でサポートされます。`CONCUR_READ_ONLY` のみが `resultSetConcurrency` でサポートされます。  
`java.sql.CallableStatement prepareCall(String sql, int resultSetType, int resultSetConcurrency)`
- `setReadOnly()`

空命令 (InterSystems IRIS ドライバは READ\_ONLY モードをサポートしない)

```
void setReadOnly(Boolean readOnly)
```

- setCatalog()

空命令 (InterSystems IRIS ドライバはカタログをサポートしない)

```
void setCatalog(String catalog)
```

- setTransactionIsolation()

TRANSACTION\_READ\_COMMITTED および TRANSACTION\_READ\_UNCOMMITTED のみが level でサポートされます。

```
void setTransactionIsolation(int level)
```

以下の **java.sql.Connection** メソッドは resultSetHoldability に対して CLOSE\_CURSORS\_AT\_COMMIT をサポートしません。

- createStatement()

```
java.sql.Statement createStatement(int resultSetType, int result, int resultSetHoldability)
```

- prepareCall()

```
java.sql.CallableStatement prepareCall(String sql,
                                         int resultSetType,
                                         int resultSetConcurrency,
                                         int resultSetHoldability)
```

- prepareStatement()

```
java.sql.PreparedStatement prepareStatement(String sql,
                                             int resultSetType,
                                             int resultSetConcurrency,
                                             int resultSetHoldability)
```

InterSystems IRIS では現在、0 個または 1 個の自動生成キーのみがサポートされています。以下の **java.sql.Connection** メソッドに、配列の長さが 1 以外の columnIndexes 配列または columnNames 配列を指定すると、例外がスローされます。

- prepareStatement()

```
java.sql.PreparedStatement prepareStatement(String sql, int[] columnIndexes)
java.sql.PreparedStatement prepareStatement(String sql, String[] columnNames)
```

## 2.3.3 DatabaseMetaData : バリエント・メソッド

### バリエント・メソッド

**java.sql.DatabaseMetaData** は完全にサポートされますが、戻り値の処理をインターシステムズ固有の方法で行うために JDBC 標準と異なるメソッドが含まれます。以下のメソッドが影響を受けます。

- supportsMixedCaseQuotedIdentifiers()

InterSystems IRIS は false を返します。これは JDBC 準拠ではありません。

```
boolean supportsMixedCaseQuotedIdentifiers()
```

- getIdentifierQuoteString()

区切り識別子のサポートがオンの場合、InterSystems IRIS は " (二重引用符) を返します。これは、JDBC 準拠のドライバが返す値です。そうでない場合、InterSystems IRIS はスペースを返します。

```
String getIdentifierQuoteString()
```

## 2.3.4 Driver : サポートされないメソッド

### サポートされないオプションのメソッド

`java.sql.Driver` は、以下のオプションのメソッドをサポートしません。

- ・ `getParentLogger()`  
  
`void getParentLogger()`

## 2.3.5 PreparedStatement : サポートされないメソッド

### サポートされないオプションのメソッド

`java.sql.PreparedStatement` は、以下のオプションのメソッドをサポートしません。

- ・ `setArray()`  
  
`void setArray(int i, Array x)`
- ・ `setRef()`  
  
`void setRef(int i, Ref x)`
- ・ `setRowId()`  
  
`void setRowId(int parameterIndex, RowId x)`
- ・ `setSQLXML()`  
  
`void setSQLXML(int parameterIndex, SQLXML xmlObject)`
- ・ `setUnicodeStream()`  
Java JDK 仕様で廃止されました。  
  
`void setUnicodeStream(int i, InputStream x, int length)`
- ・ `setURL()`  
  
`void setURL(int i, java.net.URL x)`

## 2.3.6 ResultSet : サポートされないまたは制限のあるメソッド

### 制限のあるオプションのメソッド

InterSystems IRIS は、`TYPE_SCROLL_SENSITIVE` 結果セット・タイプをサポートしません。以下のメソッドは、制限事項付きで実装されています。

- ・ `setFetchDirection()`

`ResultSet.FETCH_REVERSE` をサポートしません (代わりに、結果セットのカーソルを最後の行に移動する場合は `afterLast` を使用し、逆方向にスクロールする場合は `previous` を使用)。

```
void setFetchDirection(int direction)
```

### サポートされないオプションのメソッド

`java.sql.ResultSet` は、以下のオプションのメソッドをサポートしません。

- ・ `getArray()`

```
Array getArray(int i)
Array getArray(String colName)
```
- ・ `getCursorName()`

```
String getCursorName()
```
- ・ `getObject()`

```
Object getObject(int i, java.util.Map map)
Object getObject(String colName, java.util.Map map)
```
- ・ `getRef()`

```
Ref getRef(int i)
Ref getRef(String colName)
```
- ・ `getHoldability()`

```
int getHoldability()
```
- ・ `getUnicodeStream()`

Java JDK 仕様で廃止されました。

```
java.io.InputStream getUnicodeStream(int i)
java.io.InputStream getUnicodeStream(String colName)
```
- ・ `getURL()`

```
java.net.URL getURL(int i)
java.net.URL getURL(String colName)
```
- ・ `updateArray()`

```
void updateArray(int i, Array x)
void updateArray(String colName, Array x)
```
- ・ `updateRef()`

```
void updateRef(int i, Ref x)
void updateRef(String colName, Ref x)
```

## 2.3.7 Statement : サポートされないまたは制限のあるメソッド

### サポートされないオプションのメソッド

`java.sql.Statement` は、以下のオプションのメソッドをサポートしません。

- ・ `cancel()`  
`void cancel()`
- ・ `closeOnCompletion()`  
`void closeOnCompletion()`
- ・ `isCloseOnCompletion()`  
`boolean isCloseOnCompletion()`

### 制限のあるオプションのメソッド

以下のオプションの `java.sql.Statement` メソッドは、制限事項または制約付きで実装されています。

- ・ `getResultSetHoldability()`  
`HOLD_CURSORS_OVER_COMMIT` のみ  
`int getResultSetHoldability()`
- ・ `setCursorName()`  
空命令  
`void setCursorName(String name)`
- ・ `setEscapeProcessing()`  
空命令 (適用されない)  
`void setEscapeProcessing(Boolean enable)`
- ・ `setFetchDirection()`  
`ResultSet.FETCH_REVERSE` をサポートしません (代わりに、結果セットのカーソルを最後の行に移動する場合は `afterLast` を使用し、逆方向にスクロールする場合は `previous` を使用)。  
`void setFetchDirection(int direction)`

InterSystems IRIS がサポートする自動生成キーは、現在 0 個または 1 個のみです。以下の `java.sql.Statement` メソッドが、配列の長さが 1 以外の `columnIndexes` 配列または `columnNames` 配列を指定すると、例外がスローされます。

- ・ `execute()`  
`boolean execute(String sql, int[] columnIndexes)`  
`boolean execute(String sql, String[] columnNames)`
- ・ `executeUpdate()`  
`int executeUpdate(String sql, int[] columnIndexes)`  
`int executeUpdate(String sql, String[] columnNames)`

## 2.4 インターシステムズの拡張機能と拡張メソッド

以下のクラスは、さらに別のインターシステムズ固有の拡張メソッドを提供します。



- ・ [CallableStatement getBinaryStream\(\)](#) 拡張メソッド
- ・ “[ConnectionPoolDataSource の拡張メソッドと拡張機能](#)” では、`javax.sql.ConnectionPoolDataSource` インタフェースのインターシステムズ実装である `com.intersystems.jdbc.IRISConnectionPoolDataSource` について説明します。
- ・ “[DataSource の拡張メソッドと拡張機能](#)” では、`javax.sql.DataSource` のインターシステムズ実装である `com.intersystems.jdbc.IRISDataSource` について説明します。

## 2.4.1 CallableStatement getBinaryStream() 拡張メソッド

`java.sql.CallableStatement` は、以下のさらに別のインターシステムズ固有の拡張メソッドを実装します。

- ・ `getBinaryStream()`

指定されたパラメータの値を `java.io.InputStream` オブジェクトとして取得します（ここで、`i` はパラメータのインデックスです）。

```
java.io.InputStream getBinaryStream(int i)
```

このメソッドは、標準の `setBinaryStream()` メソッドを補完するもので、`getCharacterStream()` (`java.io.Reader` を返す) に代わるものです。

## 2.4.2 ConnectionPoolDataSource の拡張メソッドと拡張機能

`com.intersystems.jdbc.IRISConnectionPoolDataSource` クラスは、`javax.sql.ConnectionPoolDataSource` インタフェースを完全に実装します。このクラスは、InterSystems JDBC ドライバでサポートされていない `javax.sql.CommonDataSource` のメソッドを継承しません。

### 制限のあるメソッド

`getPooledConnection()` は、JDBC 標準で必要なため実装されますが、InterSystems IRIS 実装を直接呼び出すことはできません。InterSystems IRIS ドライバ接続の取得は常に `getConnection()` メソッドを呼び出すことによって行う必要があります（詳細は、“[InterSystems ソフトウェアでの Java の使用法](#)” の “[接続プールの使用法](#)” を参照してください）。

- ・ `getPooledConnection()`

```
javax.sql.PooledConnection getPooledConnection()
javax.sql.PooledConnection getPooledConnection(String usr,String pwd)
```

### 注意

アプリケーションの呼び出しに `getPooledConnection()` メソッドまたは `PooledConnection` クラスを使用しないでください。（`IRISDataSource` から継承された）`getConnection()` メソッドを呼び出して、常に InterSystems IRIS ドライバ接続を取得する必要があります。InterSystems IRIS ドライバは、プーリングが返す `java.sql.Connection` オブジェクトを使用して、このプーリングを透過的に提供します。

`IRISConnectionPoolDataSource` は、さらに別のインターシステムズの拡張メソッドを提供する `IRISDataSource` (“[DataSource の拡張メソッドと拡張機能](#)” を参照) から継承します。

### 2.4.2.1 ConnectionPoolDataSource 拡張メソッド

`IRISConnectionPoolDataSource` は、さらに以下の InterSystems IRIS 限定管理メソッドもサポートします（詳細は、“[InterSystems ソフトウェアでの Java の使用法](#)” の “[接続プールの使用法](#)” を参照してください）。

- ・ `restartConnectionPool()`

接続プールを再起動します。物理接続をすべて切断し、接続プールを空にします。

```
void restartConnectionPool()
```

- `getPoolCount()`

接続プール内の現在のエン트리数を返します。

```
int getPoolCount()
```

- `setMaxPoolSize()`

接続プールの最大サイズを設定します。最大サイズが設定されていない場合、既定値として 40 に設定されます。

```
void setMaxPoolSize(int max)
```

- `getMaxPoolSize()`

現在の接続プールの最大サイズを返します。

```
int getMaxPoolSize()
```

## 2.4.3 DataSource の拡張メソッドと拡張機能

`com.intersystems.jdbc.IRISDataSource` クラスは、`javax.sql.DataSource` インタフェースを完全に実装します。このクラスは、InterSystems JDBC ドライバでサポートされていない `javax.sql.CommonDataSource` のメソッドを継承しません。

### 必須の拡張メソッド

このメソッドの InterSystems IRIS 実装は、自動の透過的な接続プーリングを提供するために拡張されます（詳細は、“[InterSystems ソフトウェアでの Java の使用法](#)” の“[接続プールの使用法](#)”を参照してください）。

- `getConnection()`

```
java.sql.Connection getConnection()  
java.sql.Connection getConnection(String usr,String pwd)
```

### 2.4.3.1 DataSource 拡張メソッド

インタフェースで定義されるメソッドに加えて、`IRISDataSource` は、InterSystems IRIS でサポートされる `DataSource` プロパティの取得と設定に使用できる以下のメソッドも実装します（詳細は、“[InterSystems ソフトウェアでの Java の使用法](#)” の“[接続プロパティ](#)”を参照してください）。

- `getConnectionSecurityLevel()`

現在の接続セキュリティ・レベルの設定を表す `int` を返します。

```
int getConnectionSecurityLevel()
```

- `getDatabaseName()`

現在のデータベース (InterSystems IRIS ネームスペース) の名前を表す文字列を返します。

```
String getDatabaseName()
```

- `getDataSourceName()`

現在のデータ・ソース名を表す文字列を返します。

```
String getDataSourceName()
```

- `getDefaultTransactionIsolation()`

現在の既定のトランザクション分離を取得します。

```
int getDefaultTransactionIsolation()
```

- `getDescription()`

現在の記述を表す文字列を返します。

```
String getDescription()
```

- `getEventClass()`

イベント・クラス・オブジェクトを表す文字列を返します。

```
String getEventClass()
```

- `getKeyRecoveryPassword()`

現在のキー・リカバリ・パスワード設定を表す文字列を返します。

```
getKeyRecoveryPassword()
```

- `getNodeDelay()`

現在の `TCP_NODELAY` オプション設定を表すブーリアンを返します。

```
boolean getNodeDelay()
```

- `getPassword()`

現在のパスワードを表す文字列を返します。

```
String getPassword()
```

- `getPortNumber()`

現在のポート番号を表す `int` を返します。

```
int getPortNumber()
```

- `getServerName()`

現在のサーバ名を表す文字列を返します。

```
String getServerName()
```

- `getServicePrincipalName()`

現在のサービス・プリンシパル名を表す文字列を返します。

```
String getServicePrincipalName()
```

- `getSSLConfigurationName()`

現在の TLS 構成名の設定を表す文字列を返します。

```
getSSLConfigurationName()
```

- `getURL()`

対象の接続の現在の URL を表す文字列を返します。

```
String getURL()
```

- ・ `getUser()`

現在のユーザ名を表す文字列を返します。

```
String getUser()
```

- ・ `setConnectionSecurityLevel()`

接続のセキュリティ・レベルを設定します。

```
Sets the Connection Security Level for this DataSource object.
```

- ・ `setDatabaseName()`

対象の接続のデータベース名 (InterSystems IRIS ネームスペース) を設定します。

```
void setDatabaseName(String dn)
```

- ・ `setDataSourceName()`

対象の接続のデータ・ソース名を設定します。`DataSourceName` はオプション設定で、**IRISDataSource** の接続には使用されません。

```
void setDataSourceName(String dsn)
```

- ・ `setDefaultTransactionIsolation()`

トランザクションの既定の分離レベルを設定します。

```
void setDefaultTransactionIsolation(int level)
```

- ・ `setDescription()`

対象の接続の記述を設定します。`Description` はオプション設定で、**IRISDataSource** の接続には使用されません。

```
void setDescription(String d)
```

- ・ `setEventClass()`

対象の接続のイベント・クラスを設定します。このイベント・クラスは InterSystems IRIS JDBC 固有のメカニズムです。この機能は完全にオプションで、ほとんどのアプリケーションで必要ありません。

InterSystems JDBC サーバは、トランザクションのコミット時およびロールバック時にクラスで実装されているメソッドにディスパッチします。これらのメソッドが実装されているクラスは、“イベント・クラス”と呼ばれます。イベント・クラスがログイン時に指定されると、JDBC サーバは、現在のトランザクションをコミットする直前に `%OnTranCommit` にディスパッチし、現在のトランザクションをロールバック (中止) する直前に `%OnTranRollback` にディスパッチします。ユーザ・イベント・クラスは `%ServerEvent` を拡張する必要があります。このメソッドは、いかなる値も返さず、現在のトランザクションを中止できません。

```
void setEventClass(String e)
```

- ・ `setKeyRecoveryPassword()`

対象の接続のキー・リカバリ・パスワードを設定します。

```
setKeyRecoveryPassword( java.lang.String password)
```

- ・ `setLogFile()`

対象の接続のログ・ファイル名を無条件に設定します。

```
setLogFile(java.lang.String logFile)
```

- `setNodelay()`

対象の接続の `TCP_NODELAY` オプションを設定します。このフラグを切り替えると、アプリケーションのパフォーマンスに影響する場合があります。設定されていない場合、既定値として `true` が設定されます。

```
void setNodelay(boolean nd)
```

- `setPassword()`

対象の接続のパスワードを設定します。

```
void setPassword(String p)
```

- `setPortNumber()`

対象の接続のポート番号を設定します。

```
void setPortNumber(int pn)
```

- `setServerName()`

対象の接続のサーバ名を設定します。

```
void setServerName(String sn)
```

- `setServicePrincipalName()`

対象の接続のサービス・プリンシパル名を設定します。

```
void setServicePrincipalName(String name)
```

- `setSSLConfigurationName()`

対象の接続の TLS 構成名を設定します。

```
setSSLConfigurationName(java.lang.String name)
```

- `setURL()`

対象の接続の URL を設定します。

```
void setURL(String u)
```

- `setUser()`

対象の接続のユーザ名を設定します。

```
void setUser(String u)
```



# 3

## Python DB-API のサポート

インターシステムズでは、以下の 2 種類の実装で、Python Database API Specification がサポートされます。

- ・ InterSystems DB-API は、インターシステムズ・データベースへの直接のインタフェースを提供する [PEP 249 バージョン 2.0](#) に完全に準拠した実装です。

詳細は、“[Native SDK for Python の使用法](#)” の “[Python DB-API の使用法](#)” を参照してください。

- ・ pyodbc は、サードパーティのオープン・ソース Python モジュールです。インターシステムズでは、InterSystems ODBC ドライバ経由でデータベースにアクセスする手段として、**pyodbc** の使用がサポートされています。このモジュールは、2022.1 より前のバージョンの InterSystems IRIS でも使用できます。

詳細は、“[InterSystems ODBC ドライバの使用法](#)” の “[pyodbc Python ODBC ブリッジのサポート](#)” を参照してください。





# 4

## Hibernate のサポート

InterSystems Hibernate 言語は、Hibernate 言語インタフェースの実装です。SQL の実装はベンダごとに若干異なるので、ベンダはこの言語インタフェースを使用することで、特定のデータベースに対して Hibernate のカスタム・マッピングを作成できるようになります。ベンダ提供の言語の実装は Hibernate の一部として配布されます。

以下のトピックで、InterSystems Hibernate 言語の詳細な技術情報を提供します。

- ・ [Hibernate と InterSystems Hibernate 言語](#) – Hibernate 言語の概要とリソースへのリンクを提供します。
- ・ [インストールと構成](#) – インターシステムズ固有の手順を紹介します。
- ・ [言語ファイルの場所](#) – InterSystems 言語ファイルとそれらの必要な場所のリストを示します。

### 4.1 Hibernate と InterSystems Hibernate 言語

Java Persistence Architecture (JPA) は Java プロジェクト内の複雑なオブジェクト階層に推奨される永続性テクノロジーです。インターシステムズは、現在、JPA 仕様の Hibernate 実装を介した JPA をサポートしています。Hibernate は、リレーショナル・データベースにオブジェクト/リレーショナル・マッピング (ORM) サービスを提供する JDBC のラップとして動作する JBoss のオープン・ソース・フレームワークです。Hibernate は、ベンダに依存しない永続性サービスを提供します。このサービスは、一部のプロジェクトの要件である場合があります。

InterSystems Hibernate 言語は、Hibernate 言語インタフェースの実装です。SQL の実装はベンダごとに若干異なるので、Hibernate は、ベンダ提供の “方言 (dialect)” を備え、特定のデータベースへのマッピングをカスタマイズします。現在の Hibernate ディストリビューションには、高性能なカスタマイズされた InterSystems 言語クラスが含まれています。

#### 4.1.1 Hibernate を使用する状況

Hibernate は、オブジェクトをリレーショナル・テーブルに永続化するためのインフラストラクチャを提供します。基本的には、Hibernate は、オブジェクトと SQL クエリ内のテーブルとの間の変換を透過的に処理しながら、ユーザがオブジェクトの操作に集中できるようにする JDBC のラップです。Hibernate はほとんどの環境で使用できますが、常に最適なオプションとは限りません。以下に、留意する必要がある考慮事項をいくつか示します。

- ・ Hibernate は、複雑であると同時に静的なオブジェクト・モデルを使用している場合に役立ちます。InterSystems IRIS テーブル・モデルにマップする前に、データがどのような内容で、クラスがどのようなやり取りを行うかを理解しておく必要があります。
- ・ Hibernate オブジェクトはキャッシュされているため、Hibernate がオブジェクトにアクセスしている間に他のアプリケーションがそのデータを操作することはできません。リアルタイム・データが含まれる環境で作業していて、他のアプリ

ケーションがそのデータにアクセスできる状態を保つ必要がある場合は、考えられる代替手段として XEP を検討してください (“[InterSystems XEP による Java オブジェクトの永続化](#)” を参照してください)。

- ・ Hibernate は、単純なクエリによる一般的な CRUD 操作には適していますが、より複雑なクエリは、JDBC を直接使用した方が容易に作成でき、より効率的です。

## 4.2 インストールと構成

このセクションでは、InterSystems IRIS で Hibernate を使用するためにシステムを設定する手順を説明します。この手順では、正しいバージョンの InterSystems IRIS と Hibernate がインストール済みで操作可能な状態であることが前提となっています。

### 4.2.1 要件

システムに以下のソフトウェアをインストールする必要があります。

- ・ InterSystems IRIS®。
- ・ Hibernate 5.2 または 5.3。Hibernate は、[www.hibernate.org](http://www.hibernate.org) からダウンロードできます。
- ・ Java JDK 1.8 以降のサポート対象バージョン (このリリース用のオンライン・ドキュメント “[インターシステムズのサポート対象プラットフォーム](#)” の “[サポート対象 Java テクノロジ](#)” を参照してください)。

### 4.2.2 ディレクトリ

この章の手順では、以下のディレクトリを参照します。

- ・ `<install-dir>` – InterSystems IRIS のインストール・ディレクトリ。InterSystems IRIS のインスタンスの `<install-dir>` を見つけるには、InterSystems ターミナルを開いて以下のコマンドを発行します。

```
write $system.Util.InstallDirectory()
```

`<install-dir>` の場所に関するシステム固有の情報は、“[インストール・ガイド](#)” の “[インストール・ディレクトリ](#)” を参照してください。

- ・ `<hibernate_root>` – Hibernate のインストール・ディレクトリ。

### 4.2.3 システムの設定

システムに以下の変更を行います。

- ・ `intersystems-jdbc-<version>.jar` ファイル

InterSystems JDBC jar ファイルには、InterSystems JDBC ドライバが含まれています。まだ行っていない場合は、最新バージョンの JDBC jar ファイルを `<hibernate_root>%lib` (`<hibernate_root>` はインストール・ディレクトリ) にコピーしてください。ファイルには `intersystems-jdbc-<version>.jar` という名前が付けられており、`<version>` は、3.3.0 などの数値です (最新のバージョン番号はこれより大きい可能性があります)。最新バージョンのファイルは、“[InterSystems IRIS Driver Packages](#)” ページからダウンロードできます。

- ・ Java Classpath

以下が Java Classpath 上に存在することを確認します。

- `<hibernate_root>%lib` にある jar ファイル。

- Hibernate 構成ファイル (`hibernate.properties` および `hibernate.cfg.xml`) が保持されているディレクトリ既定では、これらのファイルは両方とも `<hibernate_root>%etc` にあります。

## 4.2.4 Hibernate 構成

Hibernate 構成ファイル (`hibernate.properties` または `hibernate.cfg.xml` のいずれか) では、データベースの接続情報、および InterSystems 言語クラスの名前を指定します。

以下の 5 つの構成プロパティが必要です。

- `dialect` - InterSystems 言語クラスの完全修飾名。ベースの言語クラスは以下のとおりです。

```
org.hibernate.dialect.InterSystemsIRISDialect
```

Hibernate 主キー・ジェネレータ・クラスのサポートを有効にする必要がある場合、このベース・クラスから派生したカスタム言語クラスを使用できます。

- `driver_class` - InterSystems JDBC ドライバ・クラスの完全修飾名。

```
com.intersystems.jdbc.IRISDriver
```

このクラスは、InterSystems JDBC ドライバ `jar` ファイルに含まれています (詳細は、“[システムの設定](#)”を参照)。

- `username` - アクセスする InterSystems IRIS ネームスペースのユーザ名 (既定では、`_SYSTEM`)。
- `password` - InterSystems IRIS ネームスペースのパスワード (既定では、`SYS`)。
- `url` - InterSystems JDBC ドライバの URL。URL の形式は以下のとおりです。

```
jdbc:IRIS://<host>:<port>/<namespace>
```

ここで `<host>` は、InterSystems IRIS をホストするマシンの IP アドレス、`<port>` は、InterSystems IRIS インスタンスのスーパーサーバ TCP ポート、`<namespace>` は、InterSystems IRIS データベースのデータを含むネームスペースです (詳細は、“[InterSystems ソフトウェアでの Java の使用法](#)”の“[JDBC 接続 URL の定義](#)”を参照してください)。

`hibernate.properties` の一般的なエントリには、以下の行が含まれます (ご使用のシステムに合わせて適宜 URL、ユーザ名、およびパスワードを変更してください)。

```
hibernate.dialect org.hibernate.dialect.InterSystemsIRISDialect
hibernate.connection.driver_class com.intersystems.jdbc.IRISDriver
hibernate.connection.url jdbc:IRIS://127.0.0.1:51773/USER/
hibernate.connection.username _SYSTEM
hibernate.connection.password SYS
```

以下の例は、hibernate.cfg.xml に表示される情報と同じ情報を示しています。

```
<hibernate-configuration>
  <session-factory>
    <property name="dialect">
      org.hibernate.dialect.InterSystemsIRISDialect
    </property>
    <property name="connection.driver_class">
      com.intersystems.jdbc.IRISDriver</property>
    <property name="connection.username">_SYSTEM</property>
    <property name="connection.password">SYS</property>
    <property name="connection.url">
      jdbc:IRIS://127.0.0.1:51773/USER
    </property>
  </session-factory>
</hibernate-configuration>
```

注意            hibernate.properties および hibernate.cfg.xml の両方に同じプロパティが設定されている場合、Hibernate は hibernate.cfg.xml の値を使用します。

## 4.3 言語ファイルの場所

InterSystems Hibernate 言語は、以下の場所に配置されている 4 つのファイルで構成されています

(`<hibernate>` は `hibernate-orm¥hibernate-core¥src¥main¥java¥org¥hibernate` です)。

- ・ `<hibernate>¥dialect¥` の `InterSystemsIRISDialect.java`
- ・ `<hibernate>¥dialect¥identity¥` の `IntersystemsIRISIdentityColumnSupport.java`
- ・ `<hibernate>¥exception¥internal¥` の `InterSystemsIRISSQLExceptionConversionDelegate.java`
- ・ `<hibernate>¥sql¥` の `InterSystemsIRISJoinFragment.java`

これらのファイルがまだない場合、ダウンロード情報については、[インターシステムズのサポート窓口](#)までお問い合わせください。