



Java™ DataBase Connectivity の使用

Forte™ for Java™ プログラミングシリーズ

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A.
650-960-1300

Part No. 816-7458-10
2002 年 6 月 Revision A

Copyright © 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. は、この製品に組み込まれている技術に関連する知的所有権を持っています。具体的には、これらの知的所有権には <http://www.sun.com/patents> に示されている 1 つまたは複数の米国の特許、および米国および他の各国における 1 つまたは複数のその他の特許または特許申請が含まれますが、これらに限定されません。

本製品はライセンス規定に従って配布され、本製品の使用、コピー、配布、逆コンパイルには制限があります。本製品のいかなる部分も、その形態および方法を問わず、Sun およびそのライセンサーの事前の書面による許可なく複製することを禁じます。

フォント技術を含む第三者のソフトウェアは、著作権法により保護されており、提供者からライセンスを受けているものです。

本製品には、RSA Data Security からライセンスを受けたコードが含まれています。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。

Sun、Sun Microsystems、Forte、Java、NetBeans、iPlanet および docs.sun.com は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

すべての SPARC の商標はライセンス規定に従って使用されており、米国および他の各国における SPARC International, Inc. の商標または登録商標です。SPARC の商標を持つ製品は、Sun Microsystems, Inc. によって開発されたアーキテクチャに基づいています。

サン のロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同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。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

Netscape および Netscape Navigator は、米国ならびに他の国における Netscape Communications Corporation の商標または登録商標です。

Federal Acquisitions: Commercial Software -- Government Users Subject to Standard License Terms and Conditions

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含み、明示的であるか黙示的であるかを問わず、あらゆる説明および保証は、法的に無効である限り、拒否されるものとします。

本製品が、外国為替および外国貿易管理法 (外為法) に定められる戦略物資等 (貨物または役務) に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典 : *Using Java DataBase Connectivity*
Part No: 816-4685-10
Revision A



目次

はじめに	ix
Java Data Base Connectivity の使用方法	1
JDBC プログラミング	1
一般的なプログラミング手順	1
JDBC 関連資料	2
データベースエクスプローラの使用法	4
JDBC コンポーネントの使用法	4
「JDBC」タブ	5
JDBC コンポーネントを使ったプログラミング	12
JDBC フォームウィザードの使用法	17
接続の確立	17
表示する列の選択	21
二次行セットの選択	23
アプリケーションのプレビューと生成	25
JDBC アプリケーションの実行	26
索引	27

図目次

図 1	JDBC フォームウィザード - 開く	17
図 2	JDBC フォームウィザード - データベース接続	18
図 3	JDBC フォームウィザード - 表の選択	20
図 4	JDBC フォームウィザード - 列の選択	22
図 5	JDBC フォームウィザード - 二次行セットの選択	24
図 6	JDBC フォームウィザード - ウィザードの完了	25

表目次

表 1	行セットのプロパティ	8
表 2	行セットの「他のプロパティ」タブのプロパティ	9
表 3	行セットの「イベント」タブのプロパティ	9
表 4	「コード生成」タブのプロパティ	10
表 5	データナビゲータのプロパティ	11
表 6	ストアドプロシージャのプロパティ	12
表 7	トランザクション遮断期間	21

はじめに

Forte™ for Java™ プログラミングの世界によろこそ。このマニュアルは、Forte for Java プログラミングシリーズのマニュアルの 1 つです。このマニュアルでは、持続データ、すなわち、アプリケーションの外部にあるデータベースやその他データストアに保存されているデータを使ったプログラミングに注目しています。Forte for Java がサポートするさまざまなプログラミングモデルを取り上げ、Forte for Java 統合開発環境 (IDE) が提供する透過的な持続性技術に焦点を当てています。

このマニュアルで説明しているプログラム例は、実際に作成することができます。作業環境については、以下の Web サイトにあるリリースノートを参照してください。

<http://sun.co.jp/forte/ffj/documentation/index.html>

使用するプラットフォームによっては、このマニュアルに掲載している画面イメージと異なることがあります。その場合でも表示上の違いはわずかであるため、内容を理解するには問題ありません。ほとんどの手順で Forte for Java のユーザーインタフェースを使用しますが、場合によっては、コマンド行にコマンドを入力する必要があります。その場合は、次のように、Microsoft Windows の「コマンドプロンプトウィンドウ」でのプロンプトと構文が例として示されています。

```
c:\>cd MyWorkspace\MyPackage
```

UNIX® や Linux 環境では、次のようなプロンプトとなり、¥マーク (またはバックslash) ではなくスラッシュを使用します。

```
% cd MyWorkspace/MyPackage
```

お読みになる前に

このマニュアルは、Forte for Java がサポートする持続性プログラミングモデルの使用方法を学びたいと考えているプログラマ向けに書かれています。このマニュアルは、Java およびデータベースアクセス技術に関する一般的な知識を前提としており、本書を読まれるにあたっては、次の事項を理解しておいてください。

- Java プログラミング言語
- 表やキーなどの、リレーショナルデータベースの概念
- 利用するデータベースの使用方法

注 - Sun では、本マニュアルに掲載した第三者の Web サイトのご利用に関しましては責任はなく、保証するものでもありません。また、これらのサイトあるいはリソースに関する、あるいはこれらのサイト、リソースから利用可能であるコンテンツ、広告、製品、あるいは資料に関して一切の責任を負いません。Sun は、これらのサイトあるいはリソースに関する、あるいはこれらのサイトから利用可能であるコンテンツ、製品、サービスのご利用あるいは信頼によって、あるいはそれに関連して発生するいかなる損害、損失、申し立てに対する一切の責任を負いません。

内容の紹介

このマニュアルでは、Forte for Java が提供する JDBC 生産性向上ツールについて説明しています。これらツールは、データベースと対話するクライアントコンポーネントまたはアプリケーションを構築する際の JDBC プログラミング作業の多くを自動化します。

書体と記号について

次の表と記述は、このマニュアルで使用している書体と記号について説明しています。

書体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コーディング例。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 machine_name% You have mail.
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して表わします。	<div style="border: 1px solid black; padding: 5px;"><pre>machine_name% su Password:</pre></div>
AaBbCc123 または ゴシック	コマンド行の可変部分。実際の名前または実際の値と置き換えてください。	rm <i>filename</i> と入力します。 rm ファイル名 と入力します。
『』	参照する書名を示します。	『Solaris ユーザーマニュアル』
「」	参照する章、節、または、強調する語を示します。	第 6 章「データの管理」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合、バックスラッシュは、継続を示します。	machinename% grep `^#define \ XV_VERSION_STRING`
▶	階層メニューのサブメニューを選択することを示します。	作成: 「返信」▶「送信者へ」

シェルプロンプトについて

シェル	プロンプト
UNIX の C シェル	machine_name%
UNIX の Bourne シェルと Korn シェル	machine_name\$
スーパーユーザー (シェルの種類を問わない)	#

関連マニュアル

Forte for Java のマニュアルは、Acrobat Reader (PDF) ファイル、オンラインヘルプ、サンプルアプリケーションの Readme ファイル、Javadoc™ 文書の形式で提供しています。

オンラインで入手可能なマニュアル

次のマニュアルは、Forte for Java のポータルサイトおよび docs.sun.com の Web サイトから入手できます。

Forte for Java ポータルサイトでのマニュアルの入手先は、<http://sun.co.jp/forte/ffj/documentation/index.html> です。
docs.sun.com の URL は、<http://docs.sun.com> です。

- リリースノート (HTML 形式)

Forte for Java の Edition ごとに用意されています。このリリースでの変更情報と技術上の注意事項を説明しています。

- インストールガイド (PDF 形式)

Forte for Java の Edition ごとに用意されています。対応プラットフォームへの Forte for Java のインストール手順を説明しています。さらに、システム要件、アップグレード方法、Web サーバーやアプリケーションサーバーのインストール、コマンド行での操作、インストールされるサブディレクトリ、Javadoc の設定、データベースの統合、アップデートセンターの使用方法などが含まれます。

- Forte for Java プログラミングシリーズ (PDF 形式)

Forte for Java の各機能を使用して優れた J2EE アプリケーションを開発するための方法を詳細に説明しています。

- 『Web コンポーネントのプログラミング』

JSP ページ、サーブレット、タグライブラリを使用し、クラスやファイルをサポートする Web アプリケーションを J2EE Web モジュールとして構築する方法を説明しています。

- 『J2EE アプリケーションのプログラミング』

EJB モジュールや Web モジュールを J2EE にアセンブルする方法を説明しています。また、J2EE アプリケーションの配備や実行についても説明しています。

- 『Enterprise JavaBeans コンポーネントのプログラミング』

Forte for Java の EJB ビルダーウィザードや、他の IDE コンポーネントを使用し、EJB コンポーネント (コンテナ管理や Bean 管理の持続性の機能を持つセッション Bean やエンティティ Bean) を作成する方法を説明しています。

- 『Web サービスのプログラミング』

Web サービスモジュールが提供するツールを使用して Web サービスを構築する方法を説明しています。Web サービスは、XML (Extensible Markup Language) 文書の形式で提供されるアプリケーションビジネスサービスであり、HTTP を介して配信されます。

- 『Java DataBase Connectivity の使用』

Forte for Java の JDBC 生産性向上ツールを使用し、JDBC アプリケーションを作成する方法について説明しています。

- Forte for Java チュートリアル (PDF 形式)

Forte for Java の Edition ごとに用意されています。Forte for Java のツールを使用してアプリケーションを作成する方法を、順を追って説明しています。

チュートリアルアプリケーションは、以下のサイトからもアクセスできます。

<http://forte.sun.com/ffj/documentation/tutorialsandexamples.html>

docs.sun.com (<http://docs.sun.com>) の Web サイトでは、他のサンのマニュアルの参照、印刷、購入をすることもできます。

オンラインヘルプ

オンラインヘルプは、Forte for Java 開発環境内から参照できます。ヘルプキー (Solaris オペレーティング環境では Help キー、Windows および Linux 環境では F1 キー) を押すか、「ヘルプ」>「内容」を選択します。ヘルプの項目と検索機能が表示されます。

プログラム例

Forte for Java の機能を紹介したプログラム例とチュートリアルアプリケーション (各 Edition のチュートリアルで説明されているアプリケーションを含む) を、以下の Forte for Java のポータルサイトからダウンロードすることができます。

<http://forte.sun.com/ffj/documentation/tutorialsandexamples.html>

Javadoc

Javadoc 形式のマニュアルは、Forte for Java の多くのモジュールに用意されており、IDE の中で参照できます。このマニュアルの使用方法については、リリースノートを参照してください。IDE を起動すると、エクスプローラの Javadoc タブで Javadoc マニュアルを参照できます。

ご意見の送付先

Sun のマニュアルについてのご意見やご要望をお寄せください。今後のマニュアル作成の参考にさせていただきます。次のアドレスまで電子メールをお送りください。

docfeedback@sun.com

電子メールのタイトルに、対象マニュアルの Part No. を明記してください。

Java Data Base Connectivity の使用方法

Forte for Java には、データベースと対話するクライアントコンポーネントまたはアプリケーションを構築するときの、プログラミング作業の多くを自動化する JDBC (Java Database Connectivity) モジュールが用意されています。

Forte for Java JDBC モジュールの目標は、データベースのテーブルの取り出し、更新に JDBC を使用する Swing (Java Foundation Class) コンポーネントからなる表示フォームをプログラミングする際の、開発者の生産性を向上することにあります。JDBC モジュールは、簡単な 2 層アプリケーションアーキテクチャの生成に役立ちます。

この章では、Forte for Java が提供する次の JDBC 生産性向上ツールを取り上げ、JDBC アプリケーションを作成する手順の概略を説明します。

- データベースエクスプローラ
- JDBC JavaBeans コンポーネント
- JDBC フォームウィザード

JDBC プログラミング

この節では、JDBC プログラミング入門として、その作業手順を簡単に説明します。

一般的なプログラミング手順

JDBC プログラミングの一般的な手順は以下のとおりです。

1. コードに関連するクラスをインポートする

2. JDBC ドライバを読み込む
3. データベースとの接続を確立する
4. Main メソッドを作成する
5. try & catch ブロックを作成し、例外および警告を取り出す
6. データベーステーブルを作成して、設定する
 - a. テーブルを作成する
 - b. JDBC 文を作成する
 - c. 文を実行して、持続操作を行う
 - i. テーブルにデータを登録する
 - ii. テーブルからデータを取得する
 - iii. 更新可能な結果セット (行セット) を作成する
 - iv. プログラムで行を挿入、削除する
 - d. トランザクション遮断期間を管理することによって結果セット内の変化内容を表示する

Forte for Java は、Forte for Java JDBC JavaBeans コンポーネントのプロパティを編集したり、「JDBC フォームウィザード」を使ってJDBC コードを生成することによって、プログラミング作業を簡素化します。

JDBC 関連資料

この章では、JDBC プログラミングモデルの基礎を理解していることを前提に、Forte for Java IDE の JDBC プログラミングを説明します。JDBC の詳細は、目的ごとにまとめられた以下の資料を参照してください。

JDBC プログラミングの学習

Java Developer Connection は、JDBC に関する優れたチュートリアルを提供します。

<http://developer.java.sun.com/developer/onlineTraining/new2java/programming/learn/jdbc.html>

また、Java Developer Connection には JDBC Short Course もあります。

<http://developer.java.sun.com/developer/onlineTraining/Database/JDBCShortCourse/index.html>

専門記事

以下の資料があります。

“Duke’s Bakery - A JDBC Order Entry Prototype - Part I” :

<http://developer.java.sun.com/developer/technicalArticles/Database/dukesbakery/>

JDBC 入門

JDBC プログラミングを始めるときには、次の索引を参考にしてください。

<http://developer.java.sun.com/developer/technicalArticles/Interviews/StartJDBC/index.html>

“Of Java, Databases, and Really Cool Dead Guys” という資料もあります。

<http://developer.java.sun.com/developer/technicalArticles/Interviews/Databases/index.html>

JDBC の基礎

サンのチュートリアルでも JDBC に関する情報を提供しています。

<http://java.sun.com/docs/books/tutorial/index.html>

以下のチュートリアルも参考になります。

<http://java.sun.com/docs/books/tutorial/jdbc/basics/index.html>

データベースエクスプローラの使用方法

JDBC コードの作成を始める前に、アプリケーションが使用するデータベースを理解しておく必要があります。データベース情報は、Forte for Java データベースエクスプローラを使って入手できます。

Forte for Java データベースエクスプローラでは、以下の作業を行うことができます。

- データベース構造のブラウズ
- 列や索引の情報など、データベース内のすべてのテーブルの確認
- データベースに関連する SQL ビューの確認
- データベースに定義されているすべてのストアードプロシージャの確認
- データベースデータの表示
- テーブルの作成
- ビューの作成
- データベース構造の「スナップショット」の取得
- データベースに送信された SQL コマンドの監視
- データベースへの接続

これらの作業を行う方法については、Forte for Java IDE にある「データベースエクスプローラのヘルプ」を参照してください。

JDBC コンポーネントの使用方法

Forte for Java は、データベース接続機能と、表示フォームや表示コンポーネント用の JDBC コード生成ツールを備えています。特に JDBC アプリケーションで使用される 2 種類の基本コンポーネントがあります。

- 表示コンポーネント - Swing コンポーネントを使用して、表形式のデータベース情報を表示することができます。Forte for Java では、Swing 表示コンポーネントを使って、データベースデータとユーザーを中継するフォームを作成します。Swing コンポーネントは、行データを操作して列を表示するための手段です。Forte for Java は、自動的に適切な Swing コードを生成します。表示コンポーネントには、もう 1 つ、データナビゲータという JDBC コンポーネントがあります。このコンポーネントは、ユーザー向けのデータ表示を操作するときにフォームに追加します。

- 非表示コンポーネント - データベースからのデータの操作に使えるが、ビジュアル表現を持たない **JavaBeans** コンポーネントです。非表示コンポーネントとしては、たとえば行セットがあります。これは、データベースからの情報を含む一種の行グループです。JDBC の **JavaBeans** コンポーネントの使用方法を理解するには、次の項目を学ぶ必要があります。
- 「JDBC」タブ
- JDBC コンポーネントを使ったアプリケーションのプログラミング方法。このためには、次のことが必要になります。
 - Forte for Java での表示フォームの作成
 - JDBC **JavaBeans** コンポーネントに対する Forte for Java コンポーネントインスペクタの利用

「JDBC」タブ

コンポーネントパレットの「JDBC」タブには多数のアイコンがあり、データベースと **Java Swing** コンポーネントの対話を簡単にする **JDBC JavaBeans** コンポーネントに使用します。このコンポーネントにはプロパティがあり、Forte for Java コンポーネントインスペクタを使ってカスタマイズします。以下に **JDBC JavaBeans** コンポーネントを示します。

- 接続ソース (ConnectionSource)
- プールされた接続ソース (PooledConnectionSource)
- NBCachedRowSet
- NBJDBCRowSet
- NBWebRowSet
- ストアドプロシージャ (StoredProcedure)
- データナビゲータ (DataNavigator)

接続ソース (ConnectionSource)

接続ソースは、JDBC 準拠のデータベースに接続するための非表示コンポーネントです。接続ソースを構成するには、以下のプロパティを設定します。

- データベース URL
- JDBC ドライバ名
- ユーザー名
- パスワード

プールされた接続ソース (PooledConnectionSource)

プールされた接続ソースコンポーネントは接続ソースに似ていますが、アプリケーションでプールされた接続ソースを使用するよう指定すると、アプリケーションの実行時に確立されたデータベース接続をアプリケーションが使用中止しようとしてもデータベース接続を閉じられなくなります。

ちなみに、Forte for Java はデータベース接続をプールに保持して、アプリケーション実行時に利用できるようにしています。アプリケーションが接続先のデータベースに対して頻繁に open と close を要求する場合は、プールされた接続ソースを利用することができます。

行セット (Rowset) とは

行セットコンポーネントはデータベースから取り込まれた行を表します。行セットコンポーネントを使用して、複数の Swing コンポーネントに対するデータモデルを構成することができます。

行セットに関する予備知識

行セットオブジェクトには、JDBC 結果セット、またはファイルやスプレッドシートなどの表形式のデータソースから取り込まれた一群の行が含まれます。

コードへの実装方法によっては、行セットはシリアライズしたり、表形式以外のデータソースに拡張したりすることができます。

行セットオブジェクトは、プロパティおよびイベント通知に関しては JavaBeans モデルに従っています。このため、行セットオブジェクトはアプリケーション内で他のコンポーネントと組み合わせることができる JavaBeans コンポーネントです。

行セットは接続された状態でも、接続されていない状態でも実装することができます。接続されていない行セットは、自身にデータを登録するか、データの変更をデータソースに戻すときにデータベースへの接続を取得しますが、たいていの場合、接続を開いたままにすることはありません。

接続されていないときでも、行セットが JDBC ドライバや完全な JDBC API を使用する必要はありません。このため、そのサイズは小さくなります。接続されていない行セットは、thin クライアントにデータをネットワーク送信する場合に理想的な形式です。

行セットの種類

「JDBC」タブには、3種類の行セットがあります。

■ NBCachedRowSet

NBCachedRowSet は接続されていない行セットで、メモリー内にデータをキャッシュしています。この特殊な行セットは小さめのデータセットに適しています。この行セットを使用して、携帯情報端末 (PDA) などの thin Java クライアントを操作するコードを提供する JDBC アプリケーションを作成することができます。

行セットとデータソースの接続が解除されると、アプリケーションが行セットに書き込んだあらゆる更新内容は基のデータベースに送られます。

■ NBJDBCRowSet

NBJDBCRowSet は、接続された結果セットオブジェクトを JavaBeans™ ラッピングしたもので、Swing コンポーネントからなるモデルで使用されます。内部キャッシュにすべてのデータを書き込む NBCachedRowSet に比べると、この NBJDBCRowSet はきわめて長い表の読み取りをより効率良く行うことができます。

■ NBWebRowSet

NBWebRowSet はキャッシュにフェッチされた行セットで、Swing コンポーネントからなるモデルで使用されます。このコンポーネントは NBCachedRowSet のすべての機能を備え、XML 形式で行をインポートおよびエクスポートすることができます。また、この XML 形式のファイルは、HTTP/XML プロトコルを使ってインターネット送信することができます。

JDBC 行セットは、プロパティエディタの「プロパティ」タブにある以下のプロパティを設定することによってカスタマイズできます。

表 1 行セットのプロパティ

プロパティ	定義
コマンド	この行セットを生成するための SQL 照会。正しい構文の任意の SQL 選択照会を指定できる
接続プロバイダ	構成済みの接続ソース。ドロップダウンリストから選択できる
読み取り専用	true の場合、この行セットは読み取り専用になる。行セットのデータはデータベースに書き出せない
トランザクション遮断期間	トランザクションの下での行セットのデータ処理方法を制御する。詳細は、Java マニュアルの <code>java.sql.Connection</code> の説明を参照してください。
XML 出力ディレクトリ (WebRowSet のみ)	WebRowSet からのデータの送信先ディレクトリを指定する
XML 出力ファイル (WebRowSet のみ)	WebRowSet からの XML 出力の書き込み先ファイル名を指定する

行セットの「他のプロパティ」、「イベント」、「コード生成」タブ

行セットの「他のプロパティ」タブでは、上記以外のプロパティを検査・変更することができます。

表 2 行セットの「他のプロパティ」タブのプロパティ

プロパティ	定義
データベース URL	更新するレコードが存在するデータベースの場所。たいていの場合は、接続ソースの「データベース URL」プロパティに示されているのと同じ URLになる
列のデフォルト値	新しい行に挿入する値。「列をフェッチ」をクリックすることによって行セットの列リストを取り出すことができる
読み込み時に実行	true の場合は、読み込み時に NB 行セットを実行することができる。「フォーム接続」から「読み込み時に実行」でパラメータを指定し、初期化コードを生成することができる
パスワード	NB 行セットを含む表にアクセスする際にユーザーが入力するパスワード
表名	更新するレコードが存在するデータベーステーブルの名前
ユーザー名	レコードを更新するユーザー名

行セットの「イベント」タブでは、行セットに関連づけられているイベントを検査・変更することができます。

表 3 行セットの「イベント」タブのプロパティ

プロパティ	定義
cursorMoved	cursorMoved イベントに対するイベントハンドラを指定する。このメソッドは、NBCachedRowSet のカーソルが移動したときに呼び出される
rowChanged	rowChanged イベントに対するイベントハンドラを指定する。このメソッドは、行セットの行が変更されたときに呼び出される
rowInserted	rowInserted イベントに対するイベントハンドラを指定する。このメソッドは、行セットに行が挿入されたときに呼び出される
rowSetChanged	rowSetChanged イベントに対するイベントハンドラを指定する。このメソッドは、行セットが変更されたときに呼び出される
rowCompleted	rowCompleted イベントに対するイベントハンドラを指定する。このメソッドは、挿入された行がデータベースにコミットされた後で呼び出される

「コード生成」タブでは、行セットに関する処理前および処理後のコードを指定することができます。

表 4 「コード生成」タブのプロパティ

プロパティ	定義
コード生成	コンポーネントに対して標準またはシリアライズしたコードのどちらを生成するかを選択する
カスタム作成コード	コンポーネントに対する、変数名と等号(=)を含まない独自の作成コードを入力する。この作成コードは、 <code>initComponents()</code> メソッドで呼び出される。このプロパティが空白の場合は、コンポーネントに対するデフォルトのコードが作成される
生成後のコード、初期化後コード、生成前のコード、初期化前コード	コンポーネントの作成コードの前後、および初期化コードの前後に IDE に挿入させるカスタムコードを入力する。IDE は、必ず <code>initComponents()</code> 内の初期化コードの前に作成コードを挿入する
シリアライズ先	コンポーネントをシリアライズする場合に、そのシリアライズ先のファイルの名前を設定する
デフォルトの修飾子を使用	デフォルトの修飾子を使ってコンポーネントの変数修飾子 (<code>public</code> 、 <code>private</code> など) を生成する場合は <code>True</code> に設定する。デフォルトの修飾子は、「オプション」ウィンドウのフォームオブジェクトノードの「変数修飾子」プロパティで指定する(「オプション」ウィンドウを表示するには、「ツール」->「オプション」を選択)。コンポーネントのプロパティシートに「変数修飾子」プロパティを表示して、デフォルトの修飾子に優先する設定を行えるようにする場合は、 <code>False</code> に設定する
変数名	コンポーネントの変数名を変更する

データナビゲータ

JDBC モジュールには、事前に作成されている GUI に行セットを直接ナビゲートする機能を持つ表示コンポーネントが用意されています。このコンポーネントは、プロトタイプアプリケーションを作成する必要があったり、データ入力アプリケーションを作成したりする場合に役立ちます。

データナビゲータは、データナビゲータの「プロパティ」タブにある以下のプロパティを設定することによってカスタマイズすることができます。

表 5 データナビゲータのプロパティ

プロパティ	定義
自動的に同意	データベースの変更を自動的に受け付ける。このプロパティを指定すると、ナビゲータを使って加えられた変更がただちにデータベースに送られるか、行セットに追加されて要求があったときにデータベースに送られる
行セットに結合	データナビゲータで制御する行セット
ボタンの配置	ボタンを 1 行で表示するか、2 行に分けて表示するか指定する
修正ボタンを表示	修正用のボタンの表示を有効または無効にする

ストアドプロシージャ

ストアドプロシージャは、特定のタスクを行うために論理的にまとめられた SQL 文のグループです。ストアドプロシージャは、データベースサーバー上で実行される処理または照会をカプセル化します。当然、プロシージャは実行されるサーバー上のデータベース管理システム (DBMS) によって異なります。

Forte for Java IDE の内部では、ストアドプロシージャは、JDBC アプリケーション内のデータベースストアドプロシージャを表す、非表示コンポーネントです。ストアドプロシージャは、ボタンのクリックなどのアプリケーション GUI 上でユーザーによって開始されたイベントに対する応答として呼び出すことができます。

ストアドプロシージャの構文は、Forte for Java がサポートする DBMS ごとに異なります。たとえばある DBMS では、プロシージャの定義の開始と終了を表すのに `begin` や `end` を使用するのに対し、別の DBMS では、同じことを表すのに別のキーワードを使用するといった具合です。

JDBC のチュートリアルでは、JDBC アプリケーションからのストアドプロシージャの呼び出し方法や、タイプの異なるいくつかのデータベースに対して作成できるストアドプロシージャについて説明しています。

ストアドプロシージャは、ストアドプロシージャのプロパティエディタの「プロパティ」タブにある以下のプロパティを設定することによって、カスタマイズすることができます。プロパティシートでプロパティを指定すると、ストアドプロシージャを任意のユーザーアクションに接続することができます。

表 6 ストアドプロシージャのプロパティ

プロパティ	定義
引数	ストアドプロシージャがアプリケーションから呼び出されたときに、そのプロシージャに使用させるデータベースデータを表す
行セットに結合	ドロップダウンリストからの行セットの選択を可能にする。このドロップダウンリストは、ストアドプロシージャが呼び出された後、データベースデータに基づいて再表示される
呼び出しフォーマット	ストアドプロシージャを呼び出すフォーマット。たとえば、プロパティシートのプロパティに指定した内容の代わりに使用する名前や引数を指定する
接続プロバイダ	アプリケーションからストアドプロシージャを呼び出すコンテキストを持つ、構成済みの接続ソース
名前	呼び出されるストアドプロシージャの名前

JDBC コンポーネントを使ったプログラミング

JDBC モジュールに用意されている表示および非表示コンポーネントを Swing コンポーネントと併用しながら、データベースデータの取り出しや操作に使うフォームを作成することができます。

たとえば、いくつかの Swing コンポーネント (JList、JTable、JComboBox、JButton、JToggleButton、JRadioButton、JCheckbox) では、表示するデータと対応するデータモデルを関連づけることができます。IDE では、プロパティエディタやコンポーネントインスペクタ上で、データベースと対話的なアクセスを行う JDBC コンポーネントを指定することによって、Swing コンポーネントのためのデータモデルをカスタマイズすることができます。

コンポーネントに対するデータモデルの構成

次の Swing コンポーネントには、関連づけられているデータモデルがあります。

- JList
- JTable
- JComboBox
- JButton
- JToggleButton
- JRadioButton
- JCheckBox

これらのデータモデルは、データベースからのデータを使って構成することができます。

データベースの表を表示するための最も一般的なコンポーネントは JTable です。このモデルは、各 Swing コンポーネントのプロパティシート (モデルプロパティ内) で構成することができます。

データベースの列の選択

JTable や JList など、複数の行を表示することが可能なコンポーネントは、selectionModel というプロパティも備えています。

JList と JComboBox には特殊なモデルもあります。このモデルは、SQL の join を使い、1つの行セットの1つの列に別の行セットの別の列を操作させてデータを表示します。詳細は、この後の説明を参照してください。

JTextField、JTextArea、JPasswordField、JTextPane、JEditorPane などドキュメントプロパティを持つテキストコンポーネントでは、このドキュメントプロパティを設定してデータベースのデータを利用することができます。

▼ JTable のデータモデルを構成する

1. JTable のプロパティシートにある model プロパティの値をクリックすると、省略符号ボタン (...) が表示されます。このボタンをクリックしてカスタムプロパティエディタを開きます。
2. TableEditor モードを選択します。
3. 「行セット」フィールドから、表に表示する行セットを選択します。
4. 「列をフェッチ」をクリックして、リストに列名を読み込みます。

5. 「追加」、「削除」、「編集」、「上へ移動」、「下へ移動」を使って表の列名と表示順を設定します。
6. 「了解」をクリックして変更を保存し、カスタムプロパティエディタを閉じます。

▼ JTable または JList の選択モデルを構成する

1. コンポーネントのプロパティシートにある selectionModel プロパティの値をクリックすると、省略符号ボタン (...) が表示されます。このボタンをクリックしてカスタムプロパティエディタを開きます。
2. 「行セット」フィールドから、表またはリストに表示する行セットを選択します。
3. 「了解」をクリックして変更を保存し、カスタムプロパティエディタを閉じます。

▼ JList または JComboBox のデータモデルを構成する

1. コンポーネントのプロパティシートにある model プロパティの値をクリックすると、省略符号ボタン (...) が表示されます。このボタンをクリックしてカスタムプロパティエディタを開きます。
2. 「ListEditor」モードを選択します。
3. 「一次行セット」フィールドでは、データモデルに使う行を取り出す行セットを選択し、さらに「列」ドロップダウンリストから列を 1 つ選択します。
4. 「二次行セット」フィールドの設定が必要な場合は、SQL の join に従って表示されるデータが含まれる行セットを選択します (省略可能)。一次および二次行セットの対応する列は同じデータ型である必要があります。
5. 「結合」チェックボックスが選択されている場合、対応するコンポーネントはデータベースの結合結果を表示します。選択されていない場合、対応するコンポーネントは、一次行セットに値を設定するためのコードマップとして使用されます。
6. 「データ列」(結合列) と「列を表示」(表示データ) を選択します。「了解」をクリックして変更を保存し、カスタムプロパティエディタを閉じます。

▼ JCheckbox、JRadioButton、JToggleButton のデータモデルを構成する

1. コンポーネントのプロパティシートにある model プロパティの値をクリックすると、省略符号ボタン (...) が表示されます。このボタンをクリックしてカスタムプロパティエディタを開きます。
2. フェッチするデータを含む行セットを選択します。
3. 列を選択します。この列のデータをもとにコンポーネントを選択すべきかどうか判定します。
4. コンポーネントが選択された場合に対応するデータベース値を「選択」フィールドに、またコンポーネントが選択されていない場合の値を「選択解除」フィールドに入力します。
5. 「了解」をクリックして変更を保存し、カスタムプロパティエディタを閉じます。

▼ テキストコンポーネントのドキュメントモデルを構成する

1. コンポーネントのプロパティシートにある document プロパティの値をクリックすると、省略符号ボタン (...) が表示されます。このボタンをクリックしてカスタムプロパティエディタを開きます。
2. フェッチするデータを含む行セットを選択します。
3. テキストコンポーネントを表示する列を選択します。
4. 「了解」をクリックして変更を保存し、カスタムプロパティエディタを閉じます。

表示フォームの作成

アプリケーション内の Swing コンポーネントをプロパティエディタからカスタマイズすると、データベースと対話する Swing コンポーネントに関連づけられた表示フォームを作成することができます。

▼ データベースと対話する Swing コンポーネントを使用して表示フォームを作成する

1. Forte for Java IDE に用意されているテンプレートを使用して Swing コンポーネントフォームを作成します。

2. コンポーネントパレットを使用して、非表示コンポーネントをフォームに追加します。必要な接続ソース (またはプールされた接続ソース)、行セット、ストアードプロシージャを追加してください。
3. 追加した非表示コンポーネントが表すデータベースエンティティに応じて、プロパティエディタから各コンポーネントをカスタマイズします。
4. データナビゲータなどの、必要な表示コンポーネントを追加します。
5. 必要な行セットコンポーネントを参照しながら、表示コンポーネントを対応するプロパティエディタから適宜カスタマイズします。
Forte for Java では、JDBC アプリケーションで使用する Swing コンポーネントを指定すると、アプリケーションで使用する適切な Swing クラスが自動的に作成されます。
6. 実行時の例外については、特定のフォームのプロパティエディタを使って、実行中に例外をキャッチさせ、フォームを実行するように設定します。

JDBC コンポーネントにコンポーネントインスペクタを利用する

Forte for Java コンポーネントインスペクタを使って、JDBC アプリケーションで使用するコンポーネントのプロパティを変更できます。コンポーネントインスペクタの非表示コンポーネントには、以下のコンポーネントがあります。

- NBCachedRowSet
- NBJDBCRowSet
- NBWebRowSet
- 接続ソース (ConnectionSource)
- プールされた接続ソース (PooledConnectionSource)
- ストアドプロシージャ (StoredProcedure)

データナビゲータ (DataNavigator) とその他の Swing コンポーネントは、コンテナ階層内の位置に従って表示されます。

JDBC フォームウィザードの使用方法

JDBC フォームウィザードは、データベースの表との対話できるフォームを作る手引きをします。このウィザードは、4 ページの「JDBC コンポーネントの使用方法」で概略を説明している方法に替わる手段です。ウィザードの実行を終えると、生成されたアプリケーション、アプリケーションのファイル名、パッケージができます。

以下では、Forte for Java IDE に付属しているサンプルの PointBase サーバーデータベースを使用して JDBC フォームウィザードを説明します。

▼ JDBC フォームウィザードを開く

- 「ツール」->「JDBC フォームウィザード」を選択します。

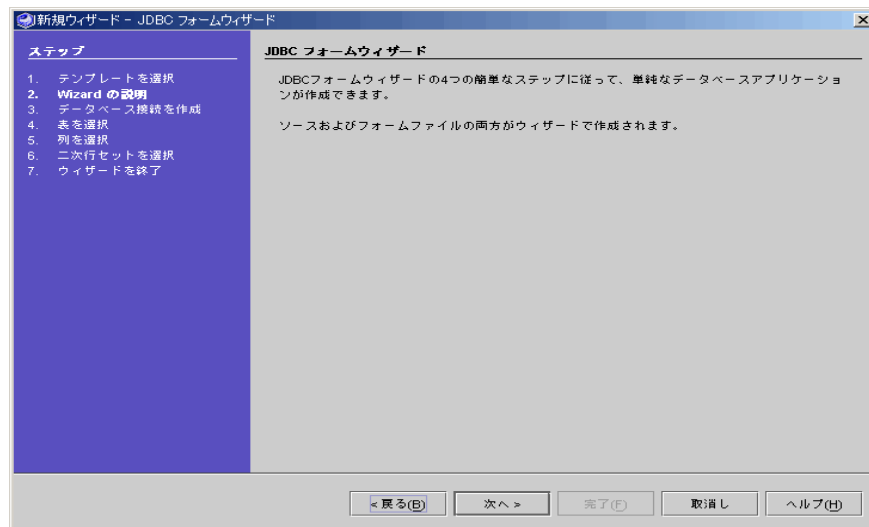


図 1 JDBC フォームウィザード - 開く

接続の確立

JDBC フォームウィザードや「JDBC」タブを使って JDBC クライアントアプリケーションを作成するときには、使用する DBMS との接続を最初に確立します。

一般に、JDBC フォームウィザードまたは Forte for Java 接続は、表示フォームエディタまたは JDBC フォームウィザードを使ってフォームを作成するときに、JDBC アプリケーション内で使用できるコードを生成します。こうしてアプリケーションは、DBMS から取得した情報を生成します。

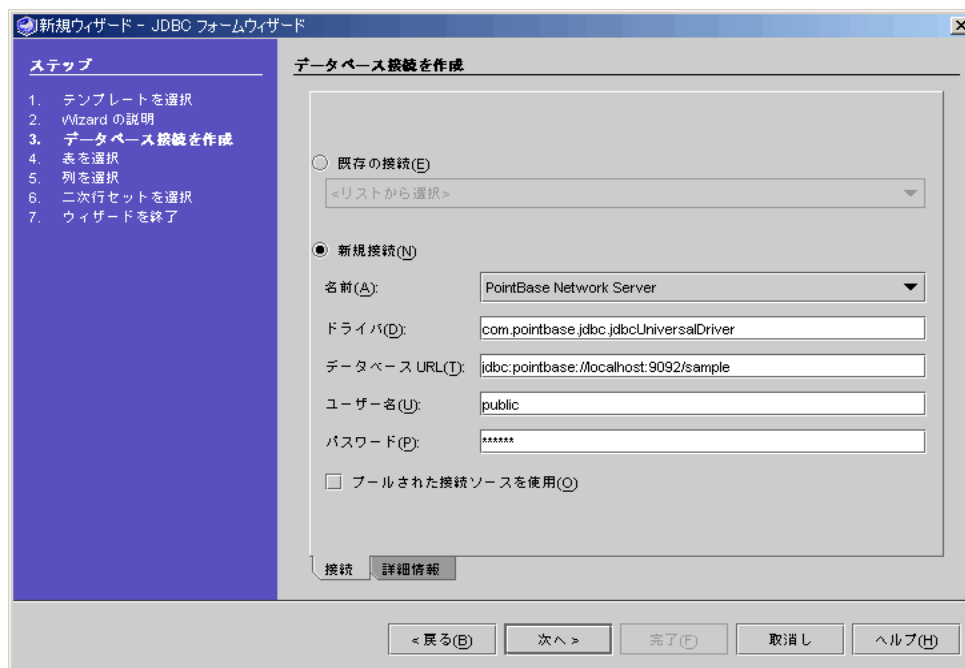


図 2 JDBC フォームウィザード - データベース接続

JDBC フォームウィザードの 2 枚目のパネルでは、データベースとの接続を確立することができます。データソースにプールされた接続を使うように指定できます。

新しい接続が必要な場合は、次の情報を入力するか、操作を行う必要があります。

- データベース名 (例: PointBase Network Server)
- データベース用の JDBC ドライバ名
(例: com.pointbase.jdbc.jdbcUniversalDriver)
- データベースが存在するデータベース URL
(例: jdbc:pointbase://localhost:9092/sample)
- ユーザー名
- パスワード
- プールされた接続の指定 (「プールされた接続ソースを使用」チェックボックスを選択した場合)

- 表を取得するスキーマの指定 (省略可能。指定は「詳細設定」タブで行う)

Forte for Java は、生成する JDBC アプリケーションコードにこれらのパラメータを入力します。

「既存の接続を使用」ラジオボタンをクリックし、ドロップダウンリストから接続を選択すると、既存の接続を選択することができます。

「次へ」ボタンをクリックすると、入力されたパラメータに基づいてデータベース接続を作成するメソッドが呼び出されます。このデータベース接続は、ウィザードを使用して JDBC アプリケーションコードを作成するのと同じ方法で使用することができます。

データベースの表またはビューの選択

JDBC フォームウィザードの 3 枚目のパネルでは、次のことを行うことができます。

- 接続先のデータベースからの表またはビューの選択
- 生成された JDBC アプリケーションを使った特定の表に対するアクセスを読み取り専用にする指定 (アプリケーションはデータベースのデータを変更できない)。
- 表に対する行挿入 (rowInserted) イベントハンドラの追加。このイベントハンドラは、選択された表へのアプリケーションによる行挿入に関連づけられているイベントを監視します。
- 表に対するトランザクション遮断期間の設定 (20 ページの「トランザクション遮断期間」を参照)。
- 指定した表に対して実行する SQL コマンドの入力

JDBC フォームウィザードでは、ウィザードで指定された表に対して SQL 文を実行することができます。SQL 出力のデータを使って、表示フォームを生成します。フォームや生成される SQL コードについて、特定のタイミングやそれぞれのコンポーネントに対応した SQL 文を指定することができます。図 3 は、選択された表に対して Forte for Java がデフォルトで使用する SQL コマンドを示しています。

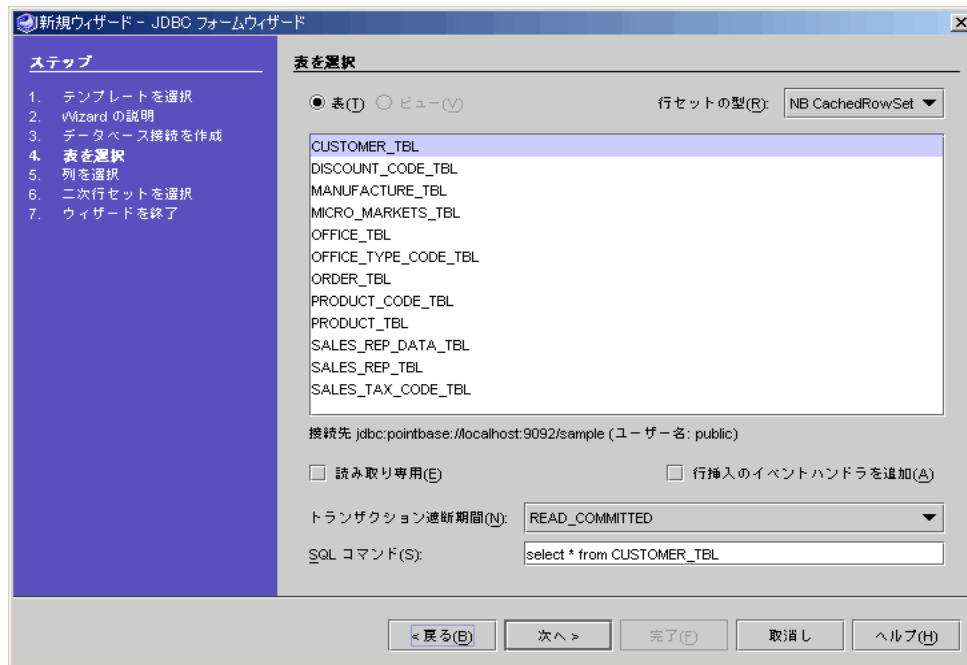


図 3 JDBC フォームウィザード - 表の選択

トランザクション遮断期間

DBMS はロックを使用して、トランザクション中の競合の回避を図ります。ロックは、アプリケーションがトランザクションをコミットするか、データベースからトランザクションをロールバックするまで有効です。

ロックは、トランザクション遮断期間に従って設定されます。したがって、アプリケーションに返されるか、アプリケーションからデータベースにコミットされる結果セット全体にロックが適用されます。

DBMS には、それぞれ独自のデフォルトのトランザクション遮断期間が用意されています。Forte for Java では、JDBC フォームウィザードの 2 枚目のパネルでトランザクション遮断期間を選択できます。

注 - 使用するドライバと DBMS が、選択するトランザクション遮断期間に対応していなければなりません。

表 7 トランザクション遮断期間

プロパティ	定義
READ_COMMITTED	変更がコミットされていない行の読み取りを禁止する
SERIALIZABLE	REPEATABLE_READ の禁止を含む。あるトランザクションが WHERE 条件を満たすすべての行を読み取っている時に、別のトランザクションが WHERE 条件を満たす「本来実在しない」行を挿入し、最初のトランザクションが同じ条件で「本来実在しない」行を含むデータを再読み取りしてしまう状況を回避する
NONE	トランザクションをサポートしない
REPEATABLE_READ	変更がコミットされていない行の読み取りを禁止する。また、最初のトランザクションがある行を読み取る間に、2つ目のトランザクションがその行を変更してしまい、最初のトランザクションが異なる値を含む同じ行を再読み取りすること (non-repeatable read と呼ばれる現象) を禁止する
READ_UNCOMMITTED	トランザクションで変更された行を、変更がデータベースにコミットされる前に別のトランザクションが読み取ることを許可する。この後で変更がロールバックされた場合、2つ目のトランザクションは無効な行を取り出すことになる

表示する列の選択

JDBC フォームウィザードの 4 枚目のパネルでは、データベースの表から、表示するフォームに含める列を選択することができます。具体的には以下を指定できます。

- 生成するアプリケーションで表示する列
- 列の表示順序

- 列のパラメータ
 - 列のタイトル
 - 列の編集
 - 列のデフォルト値
 - アプリケーションで表を表示する Swing コンポーネント

例では、最も一般的な Swing フォームである JTable を使用します。JTable フォームは、アプリケーションで複数の列のデータを表示します。

その他使用できる Swing コンポーネントには以下があります。

- JList - リストに列を 1 つ表示する
- JComboBox - コンボボックスに列を 1 つ表示する
- JTextField - テキストフィールドに 1 以上の列を表示する

図 4 では、最初の列が選択されています。この列を削除したり移動したりできます。

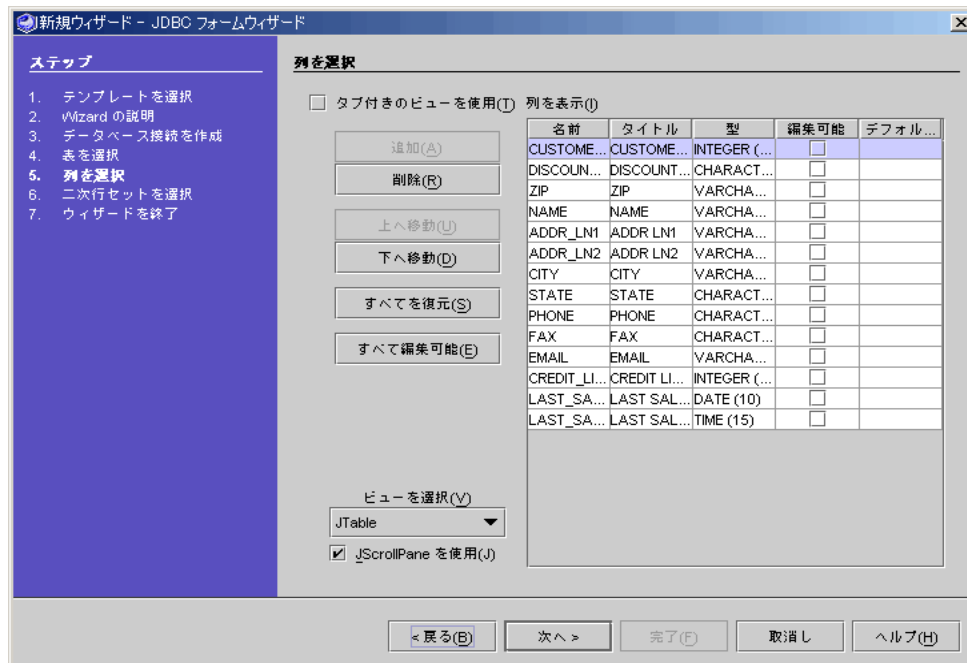


図 4 JDBC フォームウィザード - 列の選択

JList または JComboBox を選択した場合、表示できる列は 1 つだけです。「名前」プロパティから表示する列を選択します。

1. 「名前」列で値を選択します。
2. 組み込みコンボボックスから列の名前を選択します。

▼ 列のタイトルを編集する

1. 編集する「タイトル」フィールドをクリックします。2つのタブを持つ編集ウィンドウが表示されます。
2. 「文字列の値」タブを選択して、簡単な文字列の値として新しい名前を入力します。
3. 「リソースバンドル」を選択して、リソースバンドルを使用する名前を入力します。「バンドル」フィールドにバンドル名を入力し、キーのコンボボックスから使用するキーを選択します。
4. 「了解」を選択して編集ウィンドウを閉じます。

二次行セットの選択

「二次行セットを選択」パネルは、「接続」パネルで作成したデータベース接続に従って使用できるすべての表の一覧です。このパネルは、2つの行セットをサポートするビュー (JCheckbox, JList) が選択された場合にのみ表示されます。

このパネルを使用して、生成されるアプリケーションの二次行セットを生成することができます。

▼ 二次行セットを選択する

1. 「二次行セット (RowSet) を使用」チェックボックスを選択します。
二次行セットを使用する選択を行うと、生成されるアプリケーションでその二次行セットが使用されます。
2. 「表」または「ビュー」ラジオボタンのいずれかを選択します。
3. 行セットの型のコンボボックスから行セットの型を選択します。
4. リストから表またはビューを選択します。
5. 該当する行セットを読み取り専用にする場合は、「読み取り専用」チェックボックスを選択します。

6. 生成されるアプリケーションのソースコードに `rowInserted` イベントハンドラを追加する場合は、「行挿入のイベントハンドラを追加」チェックボックスを選択します。
このハンドラは新しい行が挿入されたときに呼び出され、列のデフォルト値を動的に作成することを可能にします。
7. トランザクション遮断期間のコンボボックスから、行セットに対するトランザクション遮断期間を選択します。
デフォルトの遮断期間は `READ_COMMITTED` です。
8. SQL コマンドのテキストフィールドを使用して、行セットを生成するための SQL を作成します。
デフォルトでは、`select * from table-name` というテキストが生成されます。
9. データベース結合で使用するデータ列を選択します。
この列を選択すると、取り出された主列と異なるフィールドが表示されます。ただし、選択する列は主列と同じデータ型である必要があります。

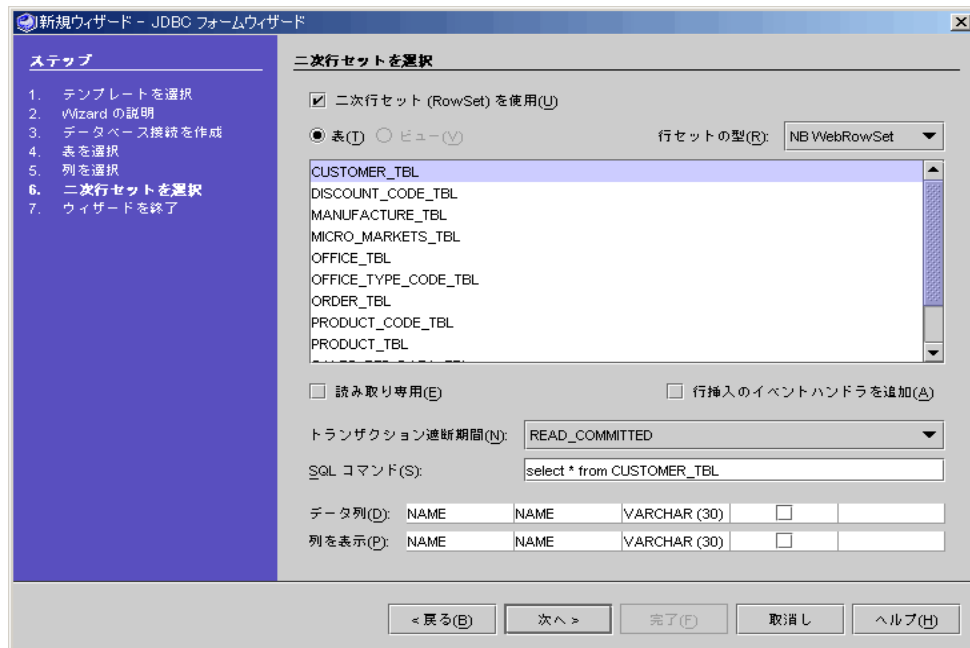


図 5 JDBC フォームウィザード - 二次行セットの選択

アプリケーションのプレビューと生成

JDBC フォームウィザードの最後のパネルでは、生成されるアプリケーションをプレビューすることができます。このパネルを利用して、生成するアプリケーションを完全なものにしてください。また、パッケージおよびファイル名を選択して、完全なアプリケーションを作成することもできます。

「パッケージ」の下にパッケージ名、「ターゲット」の下にターゲットファイル名を入力してください。

コンポーネントのレイアウトや、データナビゲータから見たレイアウトを見ることができます。実際には、アプリケーションで操作するデータを表示するフォームとして選択した Swing フォームによって見え方が異なります。

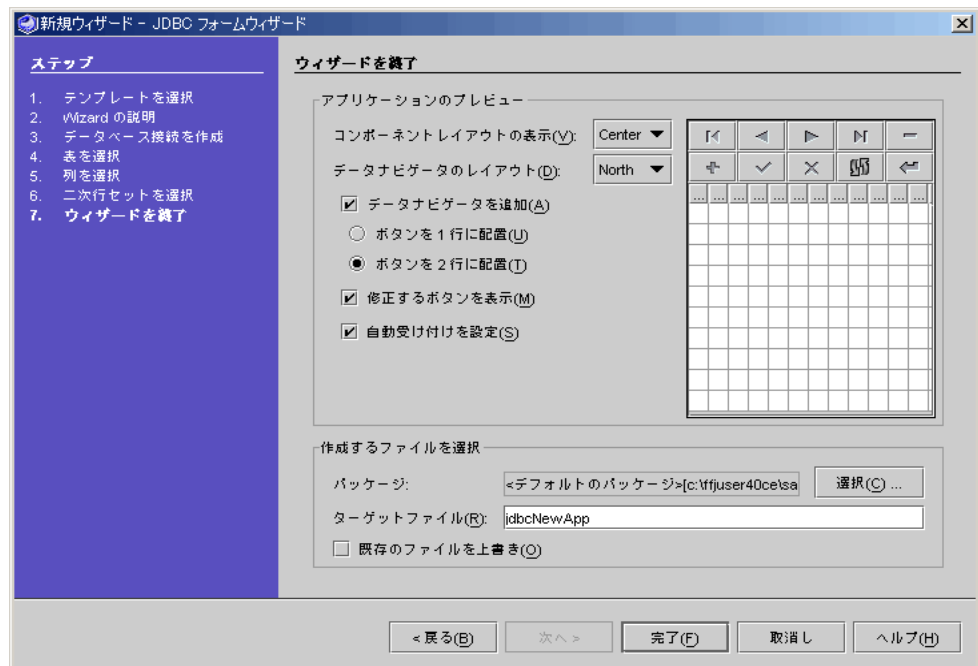


図 6 JDBC フォームウィザード - ウィザードの完了

JDBC アプリケーションの実行

JDBC アプリケーションは他のフォームと同じように、コンパイル、実行、デバッグすることができます。特定の JDBC ドライバが必要な場合は、デフォルト設定で JDBC を用いたフォームの外部コンパイルや実行、デバッグで使用できるように、必ず Forte for Java の CLASSPATH に特定のドライバ類を追加してください。

次のパッケージへのパスを CLASSPATH に追加することによって、IDE の外部でアプリケーションを実行することができます。

- modules/ext/sql.jar
- modules/ext/rowset.jar
- lib/ext/jdbc20x.zip
- 対応する JDBC ドライバ。通常、JDBC ドライバは lib/ext にあります。

JDBC アプリケーションで WebRowSet を使用している場合は、さらに 2 つの JAR ファイルが必要です。

- lib/ext/parser.jar
- lib/ext/xerces.jar

索引

C

CLASSPATH, 26

D

DataNavigator, 5

J

JAR ファイル, 26

Java Database Connectivity, 1

JDBC

JButton, 12

JCheckBox, 12

JComboBox, 12, 22

JList, 12

JList, 22

JRadioButton, 12

JTable, 12

JTextField, 22

JToggleButton, 12

関連資料, 2

データベースの列の選択, 13

表示コンポーネントと非表示コンポーネント, 12

プログラミング, 1

JDBC 表示フォーム、作成, 15

JDBC アプリケーションの実行, 26

JDBC フォームウィザード

アプリケーションのプレビューと生成, 25

データベースの表の選択, 19

N

NBCachedRowSet, 5, 16

行セットの一種, 7

NBJDBCRowSet, 5, 16

行セットの一種, 7

NBWebRowSet, 5, 16

行セットの一種, 7

あ

新しい接続の確率

「詳細設定」タブ, 19

データベース URL, 18

データベース名, 18

ドライバ名, 18

パスワード, 18

プールされた接続ソース, 18

ユーザー名, 18

アプリケーションのプレビューと生成, 25

き

行セットオブジェクト, 6

行セット、「他のプロパティ」と「イベントタブ」, 8

こ

コンポーネントインスペクタ、使用方法, 16
コンポーネントパレットの「JDBC」タブ, 5

し

書体と記号について, xi

す

ストアードプロシージャ, 5, 11, 16

せ

接続ソース, 5, 16
 JDBCドライバ名, 5
 データベース URL, 5
 ユーザー名, 5
接続の確立, 17

て

データナビゲータ, 5, 10, 16
データベースエクスプローラ、JDBC での使用方法, 4
データベースの表の選択, 19
データモデル, 12
 コンポーネントに対する構成, 12

と

トランザクション遮断期間, 20

に

二次行セットの選択, 23

は

パスワード, 5

ひ

非表示コンポーネント, 5
表示コンポーネント, 4
表示する列の選択, 21

ふ

プールされた接続ソース, 5, 16
プロパティエディタ, 11