

Oracle9i Lite

Palm 開発者ガイド

リリース 5.0

2001 年 7 月

部品番号 : J04360-01

ORACLE®

Oracle9i Lite Palm 開発者ガイド, リリース 5.0

部品番号 : J04360-01

原本名 : Oracle9i Lite Developers Guide for Palm, Release 5.0

原本部品番号 : A90105-01

Copyright © 2001, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

プログラム（ソフトウェアおよびドキュメントを含む）の使用、複製または開示は、オラクル社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権に関する法律により保護されています。

当プログラムのリバース・エンジニアリング等は禁止されております。

このドキュメントの情報は、予告なしに変更されることがあります。オラクル社は本ドキュメントの無謬性を保証しません。

* オラクル社とは、Oracle Corporation（米国オラクル）または日本オラクル株式会社（日本オラクル）を指します。

危険な用途への使用について

オラクル社製品は、原子力、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションを用途として開発されておりません。オラクル社製品を上述のようなアプリケーションに使用することについての安全確保は、顧客各位の責任と費用により行ってください。万一かかる用途での使用によりクレームや損害が発生いたしましても、日本オラクル株式会社と開発元である Oracle Corporation（米国オラクル）およびその関連会社は一切責任を負いかねます。当プログラムを米国国防総省の米国政府機関に提供する際には、『Restricted Rights』と共に提供してください。この場合次の Notice が適用されます。

Restricted Rights Notice

Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このドキュメントに記載されているその他の会社名および製品名は、あくまでその製品および会社を識別する目的にのみ使用されており、それぞれの所有者の商標または登録商標です。

目次

はじめに	xiii
------------	------

1 概要

概要	1-2
Oracle Lite DBMS	1-2
Mobile Sync Client	1-2
Mobile SQL	1-3
サンプル	1-3
Mobile サーバー	1-3
環境	1-4
開発インタフェース	1-4
ODBC	1-4
Oracle Lite OKAPI	1-4
Mobile Sync Client API	1-5
開発過程	1-5
開発システムの構成	1-5
パブリケーション項目の作成	1-6
パブリケーション項目の宣言型作成	1-6
プログラムでのパブリケーション項目の作成	1-7
手動でのデータベースの作成	1-8
Palm OS Emulator のセットアップ	1-8
エミュレータでのレプリケーション	1-9
Palm 用のモバイル・アプリケーションの開発	1-9
デバイス上でのテスト	1-9
アプリケーションのパッケージ化	1-9

2 開発システムの構成

一般要件	2-2
開発環境の構成	2-2
Palm Computing Platform の開発環境	2-2
開発システムでの Oracle Lite のインストール	2-2
ディレクトリ構造	2-2
静的ライブラリ、共有ライブラリおよびスタブ・ライブラリ	2-4
Palm Emulator の構成	2-4
TCP/IP 構成	2-5

3 同期

概要	3-2
パブリケーションとサブスクリプション	3-2
クライアント・デバイス・データベースの DDL 操作	3-2
データベース・サポート	3-3
ユーザー定義の PL/SQL パッケージのバインド	3-3
ネーミングの柔軟性	3-3
ビューの高速リフレッシュおよび更新操作	3-3
更新可能な親表	3-3
親表のヒントと INSTEAD OF トリガーの使用	3-4
ビューの高速リフレッシュ	3-4
ビューの完全リフレッシュ	3-4
更新可能パブリケーション項目の外部キー制約	3-5
外部キー制約違反の例	3-5
BeforeApply および AfterApply での制約違反の回避	3-5
表の比率を使用した制約違反の回避	3-6
レプリケーション・エラーと競合	3-7
バージョンング	3-7
ウィニング・ルール	3-7
Oracle サーバーとデバイス間でのデータ型のマッピング	3-8
Oracle Lite データベースのデータ型	3-8
Satellite Forms データ型	3-9
パブリッシュ・サブスクライブ・モデル	3-9
パブリッシュ・サブスクライブ・モデルの実装	3-10
Mobile サーバーへの接続	3-11
Mobile Server Admin API のユーザー関数	3-11

ユーザーの作成	3-11
パスワードの変更	3-12
ユーザーの削除	3-12
パブリケーションの作成	3-13
パブリケーション項目の定義	3-13
データのサブセット化	3-13
パブリケーション項目の作成	3-14
Null Sync コールアウト	3-15
パブリケーション項目名の取得	3-16
パブリケーション項目索引の作成	3-16
クライアント索引の定義	3-17
パブリケーションへのパブリケーション項目の追加	3-17
読取り専用パブリケーション項目	3-19
競合ルールの定義	3-19
表の比率の使用	3-19
シーケンスの作成	3-20
パブリケーションへのユーザーのサブスクライブ	3-20
クライアント・デバイスに対するシーケンスのパーティション化	3-21
パブリケーションに対するクライアント・サブスクリプション・パラメータの定義	3-22
サブスクリプションのインスタンス化	3-22
代替パブリケーション項目を使用したスキーマの発展	3-23
パブリケーション項目の変更	3-23
パブリッシュ・サブスクライブ・メソッド固有の関数	3-24
構成と適用を使用したコールバックのカスタマイズ	3-24
カスタマイズ DML 操作の定義	3-25
仮想主キー	3-25
仮想主キー列の作成	3-25
仮想主キー列の削除	3-26
制限選択条件	3-26
エラー・キューを使用した競合の解決	3-26
トランザクションの実行	3-27
トランザクションのページ	3-27
Palm OS Emulator のセットアップ	3-28
Emulator でのレプリケーションのテスト	3-30
Mobile サーバーのシステム・カタログ・ビュー	3-31

4 Orders サンプル・アプリケーション

概要	4-2
Orders サンプル・アプリケーションのファイル	4-2
Sample Orders アプリケーションを使用可能にする方法	4-3
Palm Computing Platform 用のサンプル・アプリケーションの使用方法	4-3
Main フォーム	4-3
ソート順序の変更	4-4
Order フォーム	4-4
Order Item フォーム	4-5
デバイスでのサンプル・アプリケーションのインストール	4-5
デバイスへのデータのレプリケート	4-6

5 Palm アプリケーションのプログラミング

開発インタフェース	5-2
ODBC	5-2
OKAPI (Oracle Lite Object Kernel API)	5-2
Mobile Sync API	5-3
ocSessionInit	5-3
ocSessionTerm	5-3
ocSaveUserInfo	5-4
ocDoSynchroize	5-4
ocSetTableSyncFlag	5-5
Mobile Sync API のデータ構造	5-6
ocEnv	5-6
ocTransportEnv	5-7
Palm Computing Platform 用アプリケーションの開発	5-7
ツール	5-8
Oracle Lite Mobile SQL for Palm Computing Platform	5-8
PUMATECH 社の Satellite Forms	5-8
GNU GCC PRC ツール	5-8
Metrowerks 社の CodeWarrior	5-9
Palm OS Emulator	5-10
開発システムでの Oracle Lite データベースの作成	5-10
Oracle Lite Mobile SQL for Palm Computing Platform の使用方法	5-10
Oracle Lite Mobile SQL for Palm Computing Platform のインストール	5-10
Oracle Lite Mobile SQL for Palm Computing Platform の起動	5-11

Oracle Lite Mobile SQL でのデータベースの新規作成	5-11
Oracle Lite Mobile SQL での表の表示	5-12
Oracle Lite Mobile SQL での SQL の使用方法	5-13
Metrowerks 社の CodeWarrior でのプロジェクトの作成	5-14
Satellite Forms でのアプリケーションの作成	5-17
要件	5-17
Satellite Forms アプリケーションの作成	5-17
Satellite Forms のサンプル・アプリケーション	5-18
Palm OS Emulator でのアプリケーションのテスト	5-19
Metrowerks 社の CodeWarrior で作成したアプリケーションのテスト	5-19
PUMATECH 社の Satellite Forms で作成したアプリケーションのテスト	5-20
Palm OS マシンでのアプリケーションのテスト	5-20
Palm OS マシンでの Mobile Development Kit のインストール	5-21
Palm OS マシンでのアプリケーションのインストール	5-22
Palm Computing Platform デバイスでの転送の構成	5-23
HTTP の使用	5-23
HTTP Sync への HotSync 方式の再割当て	5-25
HotSync の使用	5-26
Mobile Sync Client for Palm 用のセットアップ	5-26

6 パッケージ・ウィザードの使用

パッケージ・ウィザードの概要	6-2
パッケージ・ウィザードの起動	6-2
新規アプリケーションの命名	6-3
プラットフォーム・ファイルの命名	6-5
アプリケーション・ファイルのリスト表示	6-5
ソート	6-6
フィルタ	6-6
データベース情報の入力	6-7
レプリケーション用スナップショットの定義	6-8
新規スナップショットの作成	6-12
スナップショットのインポート	6-13
スナップショットの編集	6-15
アプリケーションの完了	6-16
XML ファイル	6-17
JAR ファイルの作成	6-17

SQL ファイルの作成	6-17
パッケージ・ウィザードの再起動	6-18
アプリケーションのパブリッシュ	6-18
アプリケーションの編集	6-19

A Mobile サーバー・システム・カタログ・ビュー

"MOBILEADMIN".CV\$ALL_CLIENTS	A-2
"MOBILEADMIN".CV\$ALL_PUBLICATIONS	A-2
"MOBILEADMIN".CV\$ALL_SUBSCRIPTIONS	A-2
"MOBILEADMIN".CV\$ALL_SEQUENCES	A-3
"MOBILEADMIN".CV\$ALL_SEQUENCE_PARTITIONS	A-3
"MOBILEADMIN".CV\$ALL_PUBLICATION_ITEMS_ADDED	A-3
"MOBILEADMIN".CV\$ALL_PUBLICATION_ITEMS	A-4
"MOBILEADMIN".CV\$ALL_PUBLICATION_INDEXES	A-4
"MOBILEADMIN".CV\$ALL_SUBSCRIPTION_PARAMS	A-5

B SQL サポート

Oracle Lite for the Palm Computing Platform の SQL サポート	B-2
サポートされる SQL 文	B-2
CREATE TABLE	B-2
DROP TABLE	B-3
CREATE INDEX	B-3
DELETE	B-4
DROP INDEX	B-4
INSERT	B-5
SELECT	B-5
UPDATE	B-6
COMMIT	B-7
ROLLBACK	B-7
条件	B-8
比較演算子	B-8
リテラル値	B-9
サポートされるデータ型	B-11
サポートされるグループ関数	B-12
サポートされる日付関数	B-13

C ODBC サポート

Oracle Lite for the Palm Computing Platform の ODBC サポート	C-2
SQLAllocConnect	C-3
SQLAllocEnv	C-4
SQLAllocHandle	C-4
SQLAllocStmt	C-6
SQLFreeConnect	C-7
SQLFreeEnv	C-7
SQLFreeHandle	C-8
SQLFreeStmt	C-8
SQLConnect	C-9
SQLDisconnect	C-10
SQLBindParameter	C-11
SQLPrepare	C-11
SQLExecDirect	C-12
SQLExecute	C-13
SQLFetch	C-13
SQLBindCol	C-14
SQLDescribeCol	C-15
SQLError	C-15
SQLGetData	C-16
SQLNumResultCols	C-17
SQLRowCount	C-17
SQLTransact	C-18

D Mobile SQL

概要	D-2
データベース・アクセス	D-2
インストールとセットアップ	D-3
Mobile SQL の起動と停止	D-3
Main フォーム	D-3
Table フォーム	D-4
SQL フォーム	D-6
「行の編集」オプション	D-7
Palm OS 固有の機能	D-8

用語集
索引



3-1	Mobile SQL アイコンと Mobile Sync Client アイコン	3-29
3-2	インストールされるファイル	3-29
3-3	Oracle Mobile Sync Client 画面	3-31
4-1	Palm 用の Main フォーム	4-3
4-2	Order フォーム	4-4
4-3	Order Item フォーム	4-5
5-1	開発インタフェース	5-2
5-2	Code Warrior の画面の例	5-9
5-3	Mobile SQL	5-11
5-4	Mobile SQL の表	5-12
5-5	Mobile SQL の表の例	5-12
5-6	SELECT 文の例	5-14
5-7	「New Project」メニュー	5-15
5-8	ターゲットの設定	5-16
5-9	アクセス・パス	5-16
5-10	CodeWarrior の「Preferences」	5-19
5-11	Mobile Development Kit アイコン	5-21
5-12	Mobile サーバー用の転送の構成	5-23
5-13	「ネットワーク環境設定」画面	5-24
5-14	ログイン画面	5-25
5-15	「mSync」画面	5-27
6-1	「ようこそ」パネル	6-3
6-2	「アプリケーション」パネル	6-4
6-3	「ファイル」パネル	6-6
6-4	「フィルタ」パネル	6-7
6-5	「データベース」パネル	6-8
6-6	「スナップショット」パネル	6-11
6-7	「新規スナップショット」パネル	6-12
6-8	「データベースへの接続」パネル	6-14
6-9	「表」パネル	6-14
6-10	「スナップショットの編集」パネル	6-15
6-11	「アプリケーションの定義が完了しました。」ウィンドウ	6-17
6-12	「アプリケーションをパブリッシュします。」ウィンドウ	6-18
B-1	CREATE TABLE 文の構文	B-2
B-2	DROP TABLE 文の構文	B-3
B-3	CREATE INDEX 文の構文	B-3
B-4	DELETE 文の構文	B-4
B-5	DROP INDEX 文の構文	B-4
B-6	INSERT 文の構文	B-5
B-7	SELECT 文の構文	B-6
B-8	UPDATE 文の構文	B-7
B-9	COMMIT 文の構文	B-7
B-10	ROLLBACK 文の構文	B-7

B-11	WHERE 句の構文	B-8
D-1	Mobile SQL の概要	D-2
D-2	Mobile SQL データベース画面	D-3
D-3	表ビュー	D-4
D-4	コマンド入力メニュー	D-6
D-5	行編集の例	D-7

表

2-1	ディレクトリ構造	2-3
3-1	Oracle Lite データ型	3-8
3-2	Satellite Forms データ型	3-9
3-3	パブリッシュ・サブスクライブ・モデルの要素	3-9
3-4	パブリッシュ・サブスクライブ・モデルを実装する方法	3-11
3-5	ユーザー作成パラメータの例	3-12
3-6	パスワード設定パラメータの例	3-12
3-7	ユーザー削除パラメータの例	3-13
3-8	パブリケーション作成パラメータの例	3-14
3-9	パブリケーション項目作成パラメータの例	3-15
3-10	パブリケーション項目名取得パラメータの例	3-16
3-11	パブリケーション項目索引作成パラメータの例	3-17
3-12	パブリケーション項目追加パラメータの例	3-18
3-13	サブスクリプション作成パラメータの例	3-20
3-14	シーケンス・パーティション作成パラメータの例	3-21
3-15	サブスクリプション・パラメータ設定パラメータの例	3-22
3-16	サブスクリプションのインスタンス化パラメータの例	3-23
3-17	パブリケーション項目変更パラメータの例	3-24
3-18	仮想主キー列作成パラメータ	3-25
3-19	仮想主キー列の削除パラメータ	3-26
3-20	トランザクション実行パラメータ	3-27
3-21	トランザクションのページ・パラメータ	3-27
3-22	Mobile クライアント・パラメータ	3-30
4-1	Palm Computing Platform のファイル	4-2
5-1	Palm デバイスのネットワーク設定	5-24
5-2	Palm のログイン・スクリプト	5-25
5-3	Mobile クライアント・パラメータ	5-26
6-1	「ようこそ」パネルのオプション	6-2
6-2	「アプリケーション」パネルのオプション	6-3
6-3	「ファイル」パネルのオプション	6-5
6-4	「データベース」パネルのオプション	6-7
6-5	スナップショット・パラメータ	6-9
6-6	「新規スナップショット」パネルのオプション	6-13
6-7	「スナップショットの編集」パネルのオプション	6-15
6-8	「アプリケーションをパブリッシュします。」ウィンドウのオプション	6-18
A-1	ALL_CLIENTS のパラメータ	A-2
A-2	ALL_PUBLICATIONS のパラメータ	A-2
A-3	ALL_SUBSCRIPTIONS のパラメータ	A-2
A-4	ALL_SEQUENCES のパラメータ	A-3
A-5	ALL_SEQUENCE_PARTITIONS のパラメータ	A-3
A-6	ALL_PUBLICATION_ITEMS_ADDED のパラメータ	A-3
A-7	ALL_PUBLICATION_ITEMS のパラメータ	A-4
A-8	ALL_PUBLICATION_INDEXES のパラメータ	A-4

A-9	ALL_SUBSCRIPTIONS_PARAMS のパラメータ	A-5
B-1	サポートされる SQL 文	B-2
B-2	CREATE TABLE の引数	B-3
B-3	DROP TABLE の引数	B-3
B-4	CREATE INDEX の引数	B-4
B-5	DELETE の引数	B-4
B-6	DROP INDEX の引数	B-5
B-7	INSERT の引数	B-5
B-8	SELECT の引数	B-6
B-9	UPDATE の引数	B-7
B-10	WHERE 句のパラメータ	B-8
B-11	比較演算子	B-8
B-12	リテラル値パラメータ	B-10
B-13	サポートされるデータ型	B-11
B-14	サポートされるグループ関数	B-12
B-15	サポートされる日付関数	B-13
C-1	ODBC API 関数	C-2
C-2	SQLAllocConnect の引数	C-3
C-3	SQLAllocEnv の引数	C-4
C-4	SQLAllocHandle の引数	C-5
C-5	ハンドル・パラメータ	C-5
C-6	SQLAllocStm の引数	C-6
C-7	SQLFreeConnect の引数	C-7
C-8	SQLFreeEnv の引数	C-7
C-9	SQLFreeHandle の引数	C-8
C-10	SQLFreeStmt の引数	C-9
C-11	SQLConnect の引数	C-9
C-12	SQLDisconnect の引数	C-10
C-13	SQLBlindParameter の引数	C-11
C-14	SQLPrepare の引数	C-12
C-15	SQLExecDirect の引数	C-12
C-16	SQLExecute の引数	C-13
C-17	SQLFetch の引数	C-13
C-18	SQLBindCol の引数	C-14
C-19	SQLDescribeCol の引数	C-15
C-20	SQLError の引数	C-16
C-21	SQLGetData の引数	C-16
C-22	SQLNumResultCols の引数	C-17
C-23	SQLRowCount の引数	C-18
C-24	SQLTransact の引数	C-18
D-1	Main フォームのオプション	D-4
D-2	「行の編集」ダイアログ・ボックスのオプション	D-7

はじめに

ここでは、このマニュアルの内容を紹介します。このマニュアルでは、**Oracle Lite** データベースを使用する携帯アプリケーションの開発および配布について説明します。内容は、次のとおりです。

第 1 章「概要」

レプリケーションや携帯端末用のアプリケーション開発を含めた、**Mobile Development Kit (Palm OS マシン用)** の概要を説明します。開発者に対してはレプリケーション可能な携帯アプリケーションの開発過程も説明します。

第 2 章「開発システムの構成」

携帯端末用のアプリケーションを開発するための環境の構成について、サポートされるプラットフォーム、一般的な要件、開発環境の構成、レプリケーション用のサーバー構成などを説明します。

第 3 章「同期」

Mobile サーバーとのレプリケートについて、**Mobile** サーバーの機能、データ型のマッピング、転送の構成、パブリッシュ・サブスクライブ・モデル、ウィニング・ルール、索引、シーケンスおよび拡張機能などを説明します。

第 4 章「Orders サンプル・アプリケーション」

Mobile Development Kit (Palm OS マシン用) の **Orders** サンプル・アプリケーションについて説明します。このアプリケーションでは、**Oracle Lite** データベースとレプリケーション・エンジンを使用してアプリケーションを作成する方法が示されます。

第 5 章「Palm アプリケーションのプログラミング」

携帯端末への **Oracle Lite** データベース・アプリケーションの配布について説明します。

第 6 章「パッケージ・ウィザードの使用」

パッケージ・ウィザードを使用した、アプリケーションの作成と配布について説明します。

付録 A 「Mobile サーバー・システム・カタログ・ビュー」	Mobile サーバー・システム・カタログのビューを説明します。
付録 B 「SQL サポート」	Palm OS マシン用 Oracle Lite データベースに対する SQL サポートについて説明します。
付録 C 「ODBC サポート」	Palm OS マシン用 Oracle Lite データベースに対する ODBC サポートについて説明します。
付録 D 「Mobile SQL」	クライアント・デバイス上で実行される GUI ベースのアプリケーションである Mobile SQL について説明します。これを使用すると、ローカル・データベースに対して SQL 文を実行できます。

この章では、『Oracle9i Lite Palm 開発者ガイド』の概要を説明します。内容は次のとおりです。

- [概要](#)
- [環境](#)
- [開発インタフェース](#)
- [開発過程](#)

概要

Oracle9i Lite Mobile Development Kit (Palm 用) には、次のものが含まれています。

- Oracle Lite データベース (Palm 用) — 軽量の埋込みデータベース
- Mobile Sync Client — トランザクショナル・レプリケーション・エンジン

Mobile Development Kit (Palm 用) を使用すると、分散されたモバイル・プラットフォームに企業データとアプリケーションを配布できます。携帯端末上のデータは、各プラットフォーム専用設計された Oracle Lite データベース・オブジェクト・リレーショナル・データベースに格納されます。一度配布されると、Oracle Lite データベースはエンド・ユーザーに対し透過的であり、最小限のチューニングや管理しか必要ありません。

注意： このドキュメントでは、Palm Computing Platform でのアプリケーション開発とデータベースとの同期についてのみ説明します。Windows CE については、『Oracle9i Lite Windows CE 開発者ガイド』を参照してください。EPOC については、『Oracle9i Lite Developer's Guide for EPOC』を参照してください。

Oracle Lite DBMS

Oracle Lite データベースはラップトップ、携帯用コンピュータ、PDA および情報アプリケーション用に特別に作成されたフットプリントの小さい Java 対応リレーショナル・データベースです。Oracle Lite データベースは、Windows 95/98/NT/2000、Windows CE/Pocket PC、Palm Computing および EPOC 上で稼働します。Oracle Lite データベースは、C/C++、Visual Basic および Satellite Forms などの様々なプログラミング言語からデータベース・アプリケーションを作成するための ODBC および OKAPI プログラミング・インタフェースを提供します。これらのデータベース・アプリケーションは、切断状態でも使用できます。

Mobile Sync Client

Mobile Sync Client は、携帯端末上に常駐するフットプリントの小さいアプリケーションで、Replication API や IConnect Consolidator によって以前に実行された機能を継承します。Mobile Sync Client により、携帯端末と Oracle データベースの間でデータを同期できます。Mobile Sync Client は、Windows 95/98/NT/2000、Windows CE/Pocket PC、Palm Computing および EPOC 上で稼働します。

次のような様々なコンポーネントが含まれています。

- Oracle8 データベース・サーバー。
- 携帯端末と Oracle8 データベース・サーバー間のゲートウェイとなる Mobile サーバー。

- Message Generator and Processor (MGP)。Mobile サーバーと Oracle8 データベース・サーバー間のトランザクションとデータ交換を管理するバックグラウンド・プロセスです。Mobile Sync Client は携帯端末上に常駐し、指定された任意の通信メカニズムで Mobile サーバーと通信します。
- Mobile サーバーとの通信メカニズム。HTTP や Palm HotSync などがあります。

携帯端末は一般に、中央のデータベースに含まれているデータのごく一部しか扱う必要がありません。たとえば、特定の地域の営業部隊は、その地域の顧客の連絡先情報にアクセスする必要しかありません。Mobile サーバーは、データやアプリケーションの管理にパブリッシュ・サブスクライブ・モデルを使用します。このモデルでは、Mobile Server Admin API を使用して作成したパブリケーションにより、特定のサブスクライバに対して使用可能にするデータのサブセットを定義します。サブスクリプションは、ユーザーをパブリケーションに関連付けます。クライアントとサーバーが同期を介してデータを交換する際、Mobile サーバーはユーザーの定義したルールを使用してデータの競合を検出および解決し、クライアント上で使用するためのパブリケーション項目のスナップショットを作成します。

Mobile SQL

Mobile SQL により、Palm Computing Platform デバイス上で Oracle Lite データベースを作成、アクセスおよび管理できます。Mobile SQL を使用して次のことができます。

- データベースの作成
- 表の表示
- SQL 文の実行

サンプル

Mobile Development Kit には、サンプル・アプリケーションがいくつか含まれています。サンプルは、Mobile Development Kit とともに自動的にインストールされませんが、インストール後のタスクを実行することで適用できます。サンプルのインストール方法は、『Oracle9i Lite インストレーションおよび構成ガイド』を参照してください。

Mobile サーバー

Mobile サーバーにより、携帯端末上の Oracle Lite データベースと Oracle サーバーの間のシームレスな同期が実現されます。

注意： Mobile Development Kit (Palm OS マシン用) では、Palm Computing Platform 固有のデータベース形式もサポートされます。この形式は、PUMATECH 社の Satellite Forms やその他の Palm Computing Platform 開発ツールに基づいたアプリケーションの作成に使用できます。

データ配分のパブリッシュ・サブスクライブ・モデルを使用すると、Mobile サーバーでは個々のクライアント・デバイスで、ユーザー ID や位置などのパラメータに基づいた様々なデータ・サブセットをサブスクライブできます。

Mobile サーバーは Palm OS マシン上の Oracle Lite データベースをサポートし、さらに、Palm Computing Platform では、次のものをサポートしています。

- Address Book などの Palm Computing Platform デバイス固有のアプリケーション
- Palm Computing Platform データベース (PDB)
- Satellite Forms データベース・アプリケーション

Mobile サーバーでは、携帯端末とサーバー間の転送プロトコルを複数サポートしています。次のプロトコルがあります。

- HotSync
- HTTP

環境

Mobile Development Kit (Palm OS マシン用) は、Palm Operating System のバージョン 3.5 をサポートしています。

開発インタフェース

Mobile Development Kit (Palm 用) の提供する開発インタフェースは、次のとおりです。

- [ODBC](#)
- [Oracle Lite OKAPI](#)
- [Mobile Sync Client API](#)

ODBC

Open Database Connectivity (ODBC) は、SQL データベースにアクセスするためのコールレベル・インタフェースです。Oracle Lite データベース (Palm Computing Platform 用) は、ODBC 3.0 インタフェースのサブセットをサポートしています。詳細は、[付録 C 「ODBC サポート」](#) を参照してください。

Oracle Lite OKAPI

Oracle Lite Object Kernel API (OKAPI) は、アプリケーションから Oracle Lite オブジェクト・カーネル機能へのアクセスを可能にする、C++ 言語のオブジェクト・データベース・インタフェースです。Oracle Lite OKAPI プログラムは、ターゲットとなるプラットフォームに適した開発ツール (たとえば、CodeWarrior for Palm Computing Platform) で開発でき

ます。詳細は、[第 5 章「Palm アプリケーションのプログラミング」](#)を参照してください。
また、Oracle Lite OKAPI の詳細は、『Oracle9i Lite (C および C++) オブジェクト・カーネル API リファレンス』を参照してください。

Mobile Sync Client API

この API により、アプリケーションはレプリケーション処理をプログラムによって制御できます。アプリケーションは Mobile Sync Client API 関数を起動して、レプリケーション・プロセスの開始や、Mobile Sync Client API によって生成されたエラー・メッセージの捕捉を行います。

開発過程

ここでは、Mobile Development Kit (Palm 用) の開発過程の概要を説明します。開発過程には次の手順が含まれています。

1. [開発システムの構成](#)
2. [パブリケーション項目の作成](#)
3. [Palm OS Emulator のセットアップ](#)
4. [エミュレータでのレプリケーション](#)
5. [Palm 用のモバイル・アプリケーションの開発](#)
6. [デバイス上でのテスト](#)
7. [アプリケーションのパッケージ化](#)

開発システムの構成

Mobile Development Kit を使用した開発を開始するには、Windows で稼働する開発システム上にいくつかのコンポーネントをインストールし、構成する必要があります。

- Mobile サーバー
- Mobile Development Kit
- Palm 用統合開発環境 (IDE)

サーバーおよび開発環境のインストールと構成の詳細は、[第 2 章「開発システムの構成」](#)を参照してください。

パブリケーション項目の作成

パブリケーション項目を作成するには、次の手法を使用できます。同期の発生時には、パブリケーション項目によりモバイル・データベース上に自動的に表が作成されます。パブリケーション項目の作成には、次の3つのオプションがあります。

1. **パブリケーション項目の宣言型作成** - パッケージ・ウィザードを使用してパブリケーション項目を作成します。この方法をお薦めます。
2. **プログラムでのパブリケーション項目の作成** - Mobile Server Admin API を使用してプログラムでパブリケーション項目を作成します。
3. **手動でのデータベースの作成** - 手動でモバイル・データベースを作成します。

パブリケーション項目の宣言型作成

この方法では、パッケージ・ウィザードと呼ばれる Mobile サーバーの一部を利用します。パブリケーション項目は、Mobile サーバー・リポジトリへのアプリケーションのパブリッシュ時に自動的に作成されるため、プログラミングの必要はありません。開発者が同期プロセスを実行すると、パッケージ・ウィザードで指定したスナップショット（データ・サブセット）情報に基づいて、表や索引などのモバイル・データベース・パブリケーション項目オブジェクトが自動的に作成されます。このグラフィカル・ツールは、スナップショット定義の含まれた「アプリケーション」を Mobile サーバー・リポジトリにパッケージ化してパブリッシュするために使用します。スナップショット以外のアプリケーション情報が実装されていない場合でも、Mobile サーバーにより、データを同期するために必要なレプリケーション環境が作成されます。

このグラフィカル・ツールの便利な点は、開発者がモバイル・データベースを作成するために安全で、エラーの起こりにくい手法であることです。実際にアプリケーションのプログラミングを開始する前に、次の手順を実行する必要があります。

- アプリケーションのパッケージ化
- Oracle データベース・サーバーでのデータベースの作成
- Mobile サーバー・リポジトリへのアプリケーションのパブリッシュ
- Mobile クライアントのセットアップ
- アプリケーションおよびデータの配布

Mobile サーバー・アーキテクチャは、集中化されたサーバーをモバイル・アプリケーションの管理と配布に使用できるように設計されています。このドキュメントでは、[第6章「パッケージ・ウィザードの使用」](#)で順を追ったアプローチを説明します。

プログラムでのパブリケーション項目の作成

データベースを作成してデータを移入する第2の方法としては、Mobile Server Admin API を使用してプログラムでパブリケーション項目を作成します。パブリケーション項目には、パブリケーションやサブスクリプションなどのデータ・レプリケーション・オブジェクトが含まれます。データベース・スキーマは、Mobile Server Admin API を起動する前に存在している必要があります。分散データベース・スキーマを作成するには、3つの基本的な手順が必要です。

- パブリケーションの作成
- パブリケーション項目の作成
- ユーザー ID の作成
- サブスクリプションの作成

パブリケーションの作成

パブリケーションは、定義や索引をサブセット化した表などのメタデータが含まれたテンプレート・グループです。パブリケーションは、Mobile Server Admin API を使用して作成できます。この API には、パブリッシュ・サブスクライブ・モデルを実装する Java 関数が含まれています。API 中のこれらの関数は、Java プログラムの内部から標準のファンクション・コールとしてコールできます。

パブリケーション項目の作成

パブリケーション項目は、クライアントがアクセスできるデータ・サブセットを指定する SQL の SELECT 文です。パブリケーション項目は、通常クライアント・デバイス上のレプリカ表に対応します。パブリケーション項目は、Mobile Server Admin API を使用して作成できます。この API には、パブリッシュ・サブスクライブ・モデルを実装する Java 関数が含まれています。API 中のこれらの関数は、Java プログラムの内部から標準のファンクション・コールとしてコールできます。

ユーザー ID の作成

各クライアントはユーザー ID で識別されます。開発の目的では、ユーザー ID は、特定のユーザーにデータ・サブスクリプションを割り当てるために、Mobile Server Admin API を使用して作成する必要があります。

サブスクリプションの作成

サブスクリプションは、ユーザーをパブリケーションにリンクさせます。サブスクリプションは、Mobile Server Admin API を使用して作成できます。この API には、パブリッシュ・サブスクライブ・モデルを実装する Java 関数が含まれています。API 中のこれらの関数は、Java プログラムの内部から標準のファンクション・コールとしてコールできます。Java を使用してパブリケーションとサブスクリプションを作成するには、「[パブリッシュ・サブスクライブ・モデル](#)」を参照してください。

重要： プログラムによるパブリケーション項目の作成には、Java と Mobile Server Admin API の高度なスキルが必要とされるため、より多くの手間が必要です。パッケージ・ウィザードを使用すれば、必要な手順が増えても理解がしやすいため、お薦めします。パッケージ・ウィザードは [第 6 章「パッケージ・ウィザードの使用」](#) で説明します。

手動でのデータベースの作成

Oracle Lite データベースを作成する 3 番目の方法としては、SQL を使用します。Mobile Development Kit には、基本的なデータベースを手動で作成して特定の種類のデータを挿入するためのツールがいくつかバンドルされています。Mobile SQL は、この目的で使用できるアプリケーションです。詳細は、[付録 B「SQL サポート」](#) を参照してください。

この方法を使用する開発者は、後でパッケージ・ウィザードを使用して指定するスナップショット定義と、手動で作成する表定義を必ず完全に一致させる必要があります。このため、スナップショット情報の定義に特に注意が必要となり、レプリケーション中に作成されるスナップショットと手動で作成した表定義が一致しないと、アプリケーション・エラーが発生します。

Palm OS Emulator のセットアップ

開発過程の 3 番目のステップは、開発システムに Oracle Lite データベースを作成するデータをレプリケートすることです。レプリケーションを実行する前に、Mobile Development Kit (Palm OS マシン用) ランタイム・ライブラリをインストールしておく必要があります。

Mobile サーバーと Palm OS Emulator の間の転送を構成した後、Palm OS Emulator に Mobile Sync Client をインストールして構成する必要があります。また、Mobile サーバーと同期する前に、エミュレータでユーザー・パラメータや接続パラメータを入力し、保存する必要があります。

インストールと構成の方法は、[「Palm OS Emulator のセットアップ」](#) の項を参照してください。

注意： アプリケーションは、Palm Computing Platform エミュレータを使用して開発する必要があります。エミュレータは、デスクトップ上でのデバイス・アプリケーションの開発およびテストを可能にする仮想デバイスです。

エミュレータでのレプリケーション

4 番目のステップには、開発システム上に常駐する Oracle Lite データベースに Oracle データベース・サーバーのデータをレプリケートする、最初のレプリケーション・サイクルが含まれます。Palm OS Emulator は、Windows マシン上で Palm OS マシンをシミュレートするもので、データ・レプリケーションのモニターとテストに便利です。Mobile Development Kit に含まれているサンプル・ファイルをインストールすると、Windows 開発システムに Oracle Lite データベースのサンプルが作成されます。

サンプル・ファイルのインストールの詳細は、『Oracle9i Lite インストールガイド』を参照してください。エミュレータのインストールと構成の方法は、[第 3 章「同期」](#)を参照してください。

Palm 用のモバイル・アプリケーションの開発

開発過程の 5 番目のステップは、アプリケーション開発です。同期をとって Palm OS Emulator 上にスナップショット・データを作成した後、なんらかの形で Palm IDE を使用して Windows 開発システム (PC) 上でアプリケーションを開発します。Palm OS マシンへのアプリケーションの配布と実行を試みる前に、Palm OS Emulator を使用して Palm アプリケーションをテストしてください。

アプリケーション開発の詳細は、[第 5 章「Palm アプリケーションのプログラミング」](#)を参照してください。

デバイス上でのテスト

開発過程の 6 番目のステップは、実際の Palm Computing 携帯端末でアプリケーションをテストする作業です。デバイス上で Palm アプリケーションをテストするには、開発者が手動でインストールおよび構成タスクを実行する必要があります。インストールツールを使用して、Palm OS マシンにランタイム・ライブラリおよびアプリケーション・ライブラリをインストールします。『Oracle9i Lite パブリッシュおよびディプロイ・ガイド』には、ランタイム・ライブラリおよびアプリケーション・ライブラリをインストールするための便利な方法が説明されています。

アプリケーションのパッケージ化

Palm Computing アプリケーションを開発およびテストした後、アプリケーションをパッケージ化する必要があります。パッケージ・ウィザードを使用すると、アプリケーション・ファイルとアプリケーション・ディプロイメント・ディスクリプタの含まれた自己完結型パッケージを作成できます。Mobile サーバー管理者は、このパッケージを使用して Mobile サーバー・リポジトリに Palm Computing アプリケーションをパブリッシュします。

パッケージ・ウィザードの使用方法は、[第 6 章「パッケージ・ウィザードの使用」](#)を参照してください。

開発システムの構成

この章では、携帯アプリケーション開発環境の構成を説明します。内容は次のとおりです。

- [一般要件](#)
- [開発環境の構成](#)
- [Palm Emulator の構成](#)

一般要件

Mobile Development Kit (Palm OS マシン用) でアプリケーションを開発および配布するための一般的なソフトウェア要件は、次のとおりです。

- Oracle8 リリース 8.1.6 または 8.1.7
- Java Development Kit (JDK) 1.2.2 以降

プラットフォーム固有のソフトウェア要件は、「[開発環境の構成](#)」の項を参照してください。

注意： CLASSPATH に必ず次の Mobile Development Kit クラスを追加してください。

開発環境の構成

ターゲットとなる各プラットフォームに応じた開発環境を構成する必要があります。各プラットフォームごとにエミュレータをインストールしてください。携帯端末上でアプリケーションをテストする前に、エミュレータを使用してアプリケーションをテストします。

Palm Computing Platform の開発環境

ここでは、開発システム上の Mobile Development Kit (Palm 用) コンポーネントについて説明します。

開発システムでの Oracle Lite のインストール

開発システムには、Oracle Universal Installer を使用して、Mobile Development Kit をインストールする必要があります。詳細は、『Oracle9i Lite インストレーションおよび構成ガイド』を参照してください。

ディレクトリ構造

Mobile Development Kit コンポーネントをインストールすると、`Oracle_Home¥Mobile¥SDK¥Palm` ディレクトリに常駐します。このディレクトリには、次のサブディレクトリがあります。

表 2-1 ディレクトリ構造

ディレクトリ	説明
Runtime¥shlib	odbc.prc : SQL ランタイムとの共有ライブラリ・バージョンの ODBC。 okapi.prc : 共有 OKAPI ライブラリ。 okindex.prc : 共有索引エンジン・ライブラリ。 okapi_noddl.prc : データ定義言語なしの共有 OKAPI ライブラリ。
Runtime¥bin	ConsClientStatic.prc : 静的バージョンの Mobile Sync Client。共有ライブラリは必要ありません。 ConsolidatorClient.prc : okapi.prc 、 okindex.prc 、 odbc.prc が必要。 msql.prc : 共有バージョンの Oracle Lite Mobile SQL アプリケーション。 msql_static.prc : 静的リンク・バージョンの Oracle Lite Mobile SQL アプリケーション。
SDK¥lib	ocapi.lib : アプリケーションでのレプリケーションのインスタンス化を可能にする Mobile Sync API。 odbc.lib : 静的 ODBC および SQL エンジン・ライブラリ。 okapi.lib : 静的 OKAPI ライブラリ。 okindex.lib : 静的索引エンジン・ライブラリ。 okapi_stub.lib : 共有ライブラリ用スタブ。 okapi.prc または okindex.prc あるいはその両方が必要。 odbc_stub.lib : 共有ライブラリ用スタブ。 okapi.prc 、 okindex.prc および odbc.prc が必要。 okapi_noddl.lib : データ定義言語なしの静的 OKAPI ライブラリ。 okrepex*.lib 、 pslim.lib 、 olrl.lib : 内部使用専用。
SDK¥Include	Oracle Lite インクルード・ファイル。
SDK¥examples	サンプル・アプリケーション・ファイル。

注意： *Oracle_Home* は、このマニュアルでは Oracle インストール・ディレクトリ名のプレースホルダとして使用されます。

静的ライブラリ、共有ライブラリおよびスタブ・ライブラリ

静的ライブラリ（*.lib）は、コンパイル時に静的にリンクされデバッグ処理が簡単になるため、開発に最もよく使用されます。共有ライブラリ（.prc）は、実行時に動的にリンクされます。スタブ・ライブラリは静的にリンクされる小さいライブラリで、共有ライブラリ・ファイルのプロキシです。

Palm Emulator の構成

Palm OS Emulator は、PalmPilot、Palm III、Palm V、Palm VII その他の様々なモデルのほとんどの Palm Computing Hardware デバイスをエミュレートするアプリケーションです。このエミュレータは、Windows 95、Windows 98、Windows 2000 または Windows NT 4.0 が稼働する大部分の標準デスクトップ・コンピュータで実行されます。開発者は、Palm OS アプリケーションの開発、テストおよびデバッグの補助に Palm OS Emulator を使用する必要があります。Palm OS Emulator は Palm Computing Platform のマシン・レベルでのエミュレーションなので、テストのために実際のデバイスにアプリケーションをダウンロードする必要がなくなります。かわりに、開発者は、エミュレートされた環境にアプリケーションをインストールして、開発ツールが含まれているのと同じデスクトップ・コンピュータでテストできます。

Palm OS Emulator for Windows の最新バージョンは、常にインターネットの次の Palm 開発者用ゾーンに置かれています。

<http://www.palmos-japan.com/dev/>

Emulator のリリース・バージョンを取得するには、開発者ゾーンのメイン・ページからエミュレータ・ページにリンクをたどります。

Palm OS Emulator は Palm Computing Platform ハードウェアをエミュレートするので、ハードウェアのすべてのコンポーネントが存在する必要があります。これには、Emulator とともに出荷されない ROM イメージ・ファイルが含まれます。Palm からデバッグ ROM イメージをダウンロードするには、Palm 開発者ゾーンの Web サイトに移動します。

ROM イメージ・ファイルは、Palm Provider Pavilion にあります。Palm OS Emulator の構成と使用の詳細は、Palm OS Emulator のドキュメントを調べてください。

TCP/IP 構成

Palm OS Emulator for TCP/IP を構成するには、次の手順を実行します。

1. Windows から Palm OS Emulator を起動します。
2. 適切な ROM イメージを選択します（最初のセッションでのみ必要）。
3. 「Palm OS Emulator のイメージ」を右クリックし、「Settings」→「Properties」を選択します。
4. チェック・ボックス「Redirect NetLib calls to host TCP/IP」を選択します。
5. 「OK」ボタンを選択します。

この章では、**Mobile** サーバーを使用したレプリケーションについて説明します。内容は次のとおりです。

- 概要
- [Oracle](#) サーバーとデバイス間でのデータ型のマッピング
- [パブリッシュ・サブスクライブ・モデル](#)
- [パブリッシュ・サブスクライブ・メソッド固有の関数](#)
- [Mobile](#) サーバーのシステム・カタログ・ビュー
- [Palm OS Emulator](#) のセットアップ
- [Emulator](#) でのレプリケーションのテスト

概要

Mobile サーバーにより、携帯端末上の新規または既存のアプリケーションやデータを Oracle サーバーとレプリケート、同期および共有できます。デバイス上のデータは、Oracle8 データベース上の Mobile サーバー・エージェントを介して Oracle8i データ・サーバーに直接マップできます。Mobile サーバーでは、モバイル・デバイスのデータ・サブセット化ポリシーを管理するパブリッシュ・サブスクリブ・モデルが使用されます。デバイス上のデータは、次のいずれかの転送方法を使用して Oracle8i と同期されます。

- HTTP
- HotSync、Network HotSync (Palm OS マシン用)

MGP (Message Generator and Processor) は、クライアント・デバイスで実行中のアプリケーションからトランザクションをアップロードする Java バックグラウンド・プロセスです。MGP は、Oracle データベースにトランザクションを適用します。クライアント・デバイスがダウンロードする新規の更新内容 (データ) も生成します。MGP の管理の詳細は、『Oracle9i Lite パブリッシュおよびディプロイ・ガイド』を参照してください。

パブリケーションとサブスクリプション

Mobile サーバーでは、パブリケーションとサブスクリプションが使用されます。パブリケーションは、Oracle8i の表またはビューに対して定義された問合せで、オプションでパラメータ化されます。パブリケーションは、1 人以上のユーザーによってサブスクリブされます。Mobile サーバーは、デバイスのユーザーを追跡します。確立されたサブスクリプションを介して、Mobile サーバーは各クライアント・デバイスに対する新規データを準備します。データは、行単位で水平に、または列単位で垂直にパーティション化できます。データのうち必要なサブセットのみが各クライアントにダウンロードされます。

Mobile サーバーのパブリッシュ・サブスクリブ・モデルでは、各パブリケーションに複数のパブリケーション項目を含むことができます。パブリケーション項目は、通常デバイス上の表にマップされます。パブリケーション項目内の表の列は改名できます。各パブリケーション項目には、複数のパラメータを含むことができます。詳細は、「[パブリッシュ・サブスクリブ・モデル](#)」の項を参照してください。

クライアント・デバイス・データベースの DDL 操作

クライアント・デバイスの最初のレプリケート時に、Mobile サーバーは、クライアント・デバイスにデータベース・オブジェクトを自動的に作成します。デフォルトでは、表の主要索引がサーバーから自動的にレプリケートされます。パブリケーション項目を使用して、デバイス上で 2 次索引を作成できます。1 次索引が必要ない場合は、パブリケーション項目から明示的に削除する必要があります。特定の API 情報は、『Oracle9i Lite Consolidator Admin API リファレンス』を参照してください。

データベース・サポート

サーバー側では、Mobile サーバーは Oracle8 に対して動作します。クライアント側では、Mobile Sync は、Oracle Lite によってサポートされる任意のプラットフォーム上の Oracle Lite データベースとのレプリケーションをサポートします。さらに、Mobile サーバーは、Palm Computing Platform 組込みアプリケーション固有のデータベース形式（Palm データベース）や Puma Technology 社の Satellite Forms アプリケーション形式を使用してレプリケートできます。

ユーザー定義の PL/SQL パッケージのバインド

Mobile サーバーの同期プロセスは、多くの方法でカスタマイズできます。アプリケーション・ロジックは、PL/SQL パッケージをパブリケーション項目にバインドすることで、Mobile サーバーにアタッチできます。パッケージは、BeforeCompose、AfterCompose、BeforeApply および AfterApply の各メソッドを公開する必要があります。Mobile サーバーは、次の作業の前後にこれらのメソッドをコールします。

- クライアントの変更内容を Mobile Sync Client にかかわってサーバーの表に適用する
- 指定されたパブリケーション項目に対する高速リフレッシュ変更を構成する

Mobile サーバーは、現在の Mobile Sync Client ユーザーをこれらのメソッドに渡します。

ユーザー定義の PL/SQL パッケージは、データのキャッシュや事前計算ができます。外部キー制約違反の問題も解決できます。詳細は、「[更新可能パブリケーション項目の外部キー制約](#)」の項を参照してください。これらのコールの使用の詳細は、「[構成と適用を使用したコールバックのカスタマイズ](#)」の項を参照してください。

ネーミングの柔軟性

クライアント、表、ビューおよび列の名前には、スペースを含む特殊文字がサポートされません。コール側は Mobile サーバーに対して、データベースに格納されているとおりに（多くの場合、大文字を使用する）正確にオブジェクト名を指定する必要があります。

ビューの高速リフレッシュおよび更新操作

Mobile サーバーでは、特定の条件を満たす複合複数表パブリケーションに対する高速リフレッシュ操作と更新操作をサポートしています。

更新可能な親表

ビューを更新可能にするには、親表が必要です。親表には、ビューの任意の実表が格納されています。実表のビューの列リストには主キーが含まれており、これはビューの行セットで一意です。ビューを更新可能にする場合は、ビューでパブリケーション項目を作成する前に、Mobile サーバーに適切なヒントとビューの親表を指定する必要があります。

親表のヒントと INSTEAD OF トリガーの使用

ビュー・ベースのパブリケーション項目を更新可能にするには、次の 2 つの方式を使用する必要があります。

- 親表のヒント
- INSTEAD OF トリガー

親表のヒントは、指定されたビューの親表を定義します。親表のヒントは、Consolidator Admin API の ParentHint 関数を使用して指定します。詳細は、『Oracle9i Lite Consolidator Admin API リファレンス』の「ParentHint」を参照してください。

INSTEAD OF トリガーは、INSTEAD OF INSERT、INSTEAD OF UPDATE または INSTEAD OF DELETE の各コマンドを実行するために使用します。さらに INSTEAD OF トリガーは、ビューの実表に対して実行される操作にこれらの DML コマンドをマップします。INSTEAD OF トリガーは、Oracle8i の機能です。INSTEAD OF トリガーの詳細は、Oracle8i のドキュメントを参照してください。

ビューの高速リフレッシュ

パブリケーション項目は、デフォルトでは高速リフレッシュ用に作成されます。高速リフレッシュでは、増分変更のみがレプリケートされます。高速リフレッシュの利点は、同期セッション間での変更が限られている場合に大量のデータを持つデータ・ストアをレプリケートするときのオーバーヘッドが軽減され速度が向上することです。

ビューが次の条件を満たす場合、Mobile サーバーはビューの高速リフレッシュを実行します。

- 各ビューの実表には主キーが必要です。
- すべての実表の主キーは、すべてビューの列リストに含まれている必要があります。
- 項目がビューで、項目の選択条件に複数の表が含まれている場合、選択条件定義に含まれているすべての表が主キーを持ち、対応するパブリケーション項目を持っている必要があります。

ビューでは、親表に対してのみ一意の主キーが必要です。その他の表の主キーは重複してもかまいません。実表の主キー列ごとに、ビューでその列をコールする方法に関するヒントを Mobile サーバーに提供する必要があります。これは、Consolidator Admin API の PrimaryKeyHint を使用して実現できます。詳細は、『Oracle9i Lite Consolidator Admin API リファレンス』を参照してください。

ビューの完全リフレッシュ

パブリケーション項目は、Consolidator Admin API の CompleteRefresh コールを使用して、完全リフレッシュ用に作成できます。このモードを指定した場合、クライアント・データは同期のたびにサーバーの現在のデータで完全にリフレッシュされます。管理者は、API コールを介して、パブリケーション全体に完全リフレッシュを強制できます。完全リフレッシュ関数は、指定したクライアントに対するパブリケーションの完全リフレッシュを強制的

に実行します。詳細は、『Oracle9i Lite Consolidator Admin API リファレンス』を参照してください。

更新可能パブリケーション項目の外部キー制約

Oracle8i とクライアント・デバイスの間で更新可能モードで表をレプリケートするとき、表に参照整合性制約があると、外部キー制約違反が起きる場合があります。外部キー制約違反が発生した場合、サーバーはクライアント・トランザクションを拒否します。

外部キー制約違反の例

たとえば、2つの表 EMP と DEPT に参照整合性制約があるとします。DEPT 表の DeptNum (部門番号) 属性は、EMP 表の外部キーです。EMP 表の各従業員の DeptNum 値は、DEPT 表の有効な DeptNum 値である必要があります。

Mobile サーバー・ユーザーが DEPT 表に新しい部門を追加し、次に EMP 表でこの部門に新しい従業員を追加します。トランザクションはまず DEPT を更新し、次に EMP 表を更新します。しかし、データベース・アプリケーションでは、これらの操作を実行したシーケンスを保存しません。

ユーザーが Mobile サーバーをレプリケートする際、Mobile サーバーは最初に EMP 表を更新します。この作業で、DeptNum に対して無効な外部キー値を持つ新規レコードを EMP に作成しようとします。Oracle8i により参照整合性違反が検出されます。Mobile サーバーはトランザクションをロールバックし、トランザクション・データを Mobile サーバーのエラー・キューに置きます。この場合、トランザクション内の操作が元と違うシーケンスで行われたために、外部キー制約違反が発生しました。

BeforeApply および AfterApply での制約違反の回避

PL/SQL を使用すると、DEFERRABLE 制約を BeforeApply 関数および AfterApply 関数とともに使用することで、シーケンスどおりでない操作による外部キー制約違反を回避できます。DEFERRABLE 制約は、INITIALLY IMMEDIATE または INITIALLY DEFERRED のいずれかにできます。DEFERRABLE INITIALLY IMMEDIATE 外部キー制約の動作は、通常の即時制約と同じです。どちらの制約をアプリケーションに適用しても、機能に違いはありません。

Mobile サーバーは、サーバーにクライアント・トランザクションを適用する前に BeforeApply 関数をコールし、トランザクションを適用した後で AfterApply 関数をコールします。BeforeApply 関数を使用して、制約を DEFERRED に設定し、参照整合性検査を遅延できます。トランザクションが適用された後、AfterApply 関数をコールして制約を IMMEDIATE に設定します。この時点で、クライアント・トランザクションが参照整合性に違反している場合は、ロールバックされエラー・キューに移動されます。

DEFERRABLE 制約を使用して外部キー制約違反を回避するには、次の手順に従います。

1. 外部キー制約をすべて削除して DEFERRABLE 制約として作成し直します。
2. ユーザー定義の PL/SQL パッケージを、参照整合性制約を持つ表が含まれたパブリケーションにバインドします。
3. 次のように、BeforeApply 関数で DEFERRED 制約を、AfterApply 関数で IMMEDIATE 制約を定義します。

```
procedure BeforeApply(clientname varchar2) is
cur integer;
begin
  cur := dbms_sql.open_cursor;
  dbms_sql.parse(cur, 'SET CONSTRAINT SAMPLE3.address14_fk
                     DEFERRED', dbms_sql.native);
  dbms_sql.close_cursor(cur);
end;
procedure AfterApply(clientname varchar2) is
cur integer;
begin
  cur := dbms_sql.open_cursor;
  dbms_sql.parse(cur, 'SET CONSTRAINT SAMPLE3.address14_fk
                     IMMEDIATE', dbms_sql.native);
  dbms_sql.close_cursor(cur);
end;
```

表の比率を使用した制約違反の回避

表の比率は、パブリケーションとパブリケーション項目を関連付ける整数プロパティです。Mobile サーバーは、表の比率を使用して、マスター表にクライアント操作を適用する順序を次のように決定します。

1. 最初に、クライアントの INSERT 操作が、表の比率の低いものから高いものへと順に実行されます。
2. 次に、クライアントの DELETE 操作が、表の比率の高いものから低いものへと順に実行されます。
3. 最後に、クライアントの UPDATE 操作が、表の比率の低いものから高いものへと順に実行されます。

「外部キー制約違反の例」の項にリストされた例では、DEPT 表に EMP 表よりも低い比率を割り当てることで、制約違反エラーを解決できます。たとえば、次のようになります。

(DEPT weight=1, EMP weight=2)

レプリケーション・エラーと競合

Mobile サーバーでは、サーバーが行を削除するのと同時にクライアントが更新する場合、Oracle8i のアドバンスド・レプリケーションとの互換性エラーが発生します。NULL の使用違反や外部キー制約違反などのその他のエラーはすべてレプリケーション・エラーです。

Mobile サーバーでは、レプリケーション・エラーは自動的に解決されません。かわりに、Mobile サーバーは対応するトランザクションをロールバックし、トランザクション操作を Mobile サーバーのエラー・キューに移動します。Mobile サーバーのデータベース管理者は、後でこれらのトランザクション操作を変更して再実行したり、エラー・キューからパージできます。

Mobile サーバーのレプリケーション競合は、次の場合に発生します。

- クライアントとサーバーが同じ行を更新する場合
- クライアントとサーバーが同じ主キー値を持つ行を作成する場合
- サーバーが更新する行とクライアントが削除する行が同じ場合

競合解決手法の詳細は、「[エラー・キューを使用した競合の解決](#)」の項を参照してください。

バージョンニング

Mobile サーバーでは、内部バージョンニングを使用してレプリケーションの競合を検出します。バージョン番号は、各サーバー・レコードにかぎらず、各クライアント・レコードに対しても管理されます。クライアントの変更内容がサーバーに適用される際、Mobile サーバーはバージョンの不一致を検出し、ウィニング・ルールに従って競合を解決します。

ウィニング・ルール

Mobile サーバーは、ウィニング・ルールを使用してレプリケーションの競合を自動的に解決します。次のウィニング・ルールが組み込まれています。

- クライアント優先
- サーバー優先

クライアント優先の場合、Mobile サーバーはクライアントの変更内容を自動的にサーバーに適用します。サーバー優先の場合、Mobile サーバーはクライアントに対する変更内容を自動的に構成します。

Mobile サーバーの競合解決方式は、ウィニング・ルールを「クライアント優先」に設定し、データベース表に BEFORE INSERT、BEFORE UPDATE および BEFORE DELETE の各トリガーをアタッチすることで、カスタマイズできます。トリガーにより、新旧の行の値を比較して指定されたとおりにクライアントの変更内容を解決します。

Oracle サーバーとデバイス間でのデータ型のマッピング

Mobile サーバーによって同期される Oracle8i と Oracle Lite データベースの表では、互換性のあるデータ型を使用している必要があります。Oracle8i のデータ型は、次のデータ型といくつか互換性があります。

- Satellite Forms データ型
- Oracle Lite データベースのデータ型

Oracle Lite データベースのデータ型

Oracle Lite データベースのスナップショットはすべて、レプリケーション時に Consolidator クライアントによって作成されます。Consolidator では、Oracle8i でのデータ精度に応じて自動的に Oracle Lite データベースのデータ型が選択されます。次の表にデータ変換を示します。Oracle データ型は左側の列に、Oracle Lite データベースのデータ型は最上部行に表示されています。「X」は無条件にサポートされ、「-」はサポートされないことを示します。1B、2B および 4B のデータ型は、OKAPI 専用のデータ型です。詳細は、『Oracle9i Lite (C および C++) オブジェクト・カーネル API リファレンス』を参照してください。

表 3-1 Oracle Lite データ型

	1B	2B	4B	FLOAT	DOUBLE	DATETIME	LONG-VARBINARY	VARCHAR
INTEGER	X	X	X	X	X	-	-	-
VARCHAR2	-	-	-	-	-	-	-	X
VARCHAR	-	-	-	-	-	-	-	X
CHAR	-	-	-	-	-	-	-	X
SMALLINT	X	X	X	X	X	-	-	-
FLOAT	X	X	X	X	X	-	-	-
DOUBLE PRECISION	X	X	X	X	X	-	-	-
NUMBER	X	X	X	X	X	-	-	-
DATE	-	-	-	-	-	X	-	-
LONG RAW	-	-	-	-	-	-	X	-
BLOB	-	-	-	-	-	-	X	-

Satellite Forms データ型

Mobile サーバーでは、Oracle8i と Satellite Forms のデータ型の間のデータ変換が必要に応じて自動的に実行されます。次の表にデータ変換を示します。Oracle データ型は左側の列に、Satellite Forms データ型は最上部行に表示されています。「X」は無条件にサポートされること、「S」は検証が必要なこと、「-」はサポートされないことを示します。

表 3-2 Satellite Forms データ型

	Character	Numeric	Date	Time	Ink	Timestamp
INTEGER	S	X	-	-	-	-
VARCHAR2	X	S	S	S	-	S
VARCHAR	X	S	S	S	-	S
CHAR	X	S	S	S	-	S
SMALLINT	S	X	-	-	-	-
FLOAT	S	X	-	-	-	-
DOUBLE PRECISION	S	X	-	-	-	-
NUMBER	S	X	-	-	-	-
DATE	S	-	X	X	-	X
LONG RAW	X	S	S	S	S	S

パブリッシュ・サブスクライブ・モデル

Mobile サーバーではパブリッシュ・サブスクライブ・モデルを使用して、Oracle サーバーと携帯端末間のデータ配分を集中管理します。パブリッシュ・サブスクライブ・モデルは、Mobile Server Admin API および Consolidator Admin API を使用して、特定の専用機能を起動するようプログラムで実装できます。これらの API はどちらも Mobile サーバーの一部です。このモデルの内容は次のとおりです。

表 3-3 パブリッシュ・サブスクライブ・モデルの要素

項目	説明
パブリケーション	パブリケーションは、パブリケーション項目のグループです。
パブリケーション項目	パブリケーション項目は、ユーザーがアクセスできるデータ・サブセットを指定する SQL の Select 文です。パブリケーション項目は、通常クライアント・デバイス上のレプリカ表に対応します。

表 3-3 パブリッシュ・サブスクライブ・モデルの要素（続き）

項目	説明
サブスクリプション	サブスクリプションは、ユーザーをパブリケーションに対応付けます。サブスクリプション・パラメータを含む場合もあります。
ユーザー	<p>ユーザーは、ユーザー名とパスワードで定義されます。Mobile サーバーは、クライアントのサブスクリプションに従ってデータを同期します。</p> <ul style="list-style-type: none">■ ユーザーは単一のユーザー名を使用して、複数のデバイスに格納されたデータを同期できます。ユーザーがデバイスを変更すると、Mobile サーバーにより新規デバイス上でそのユーザーの全サブスクリプションの完全リフレッシュが実行されます。■ ユーザーは、複数のユーザー名を使用して、単一のデバイス上のデータを同期できます。ユーザー名を変更しようとする、Mobile サーバーによりクライアントの更新内容はすべて無視され、クライアントの全サブスクリプションの完全リフレッシュが実行されます。
サブスクリプション・パラメータ	サブスクリプション・パラメータは、名前と文字列値を使用して、個々のパブリケーションに対する個々のクライアントのサブスクリプションを定義します。サブスクリプション・パラメータを使用すると、クライアントはデータのサブセット化を実行し、各クライアントに割り当てられる行数を制限できます。一般的なサブスクリプション・パラメータには、ユーザー名と市外局番を含めることができます。

パブリッシュ・サブスクライブ・モデルの実装

Mobile サーバーのパブリッシュ・サブスクライブ・モデルを実装するには、次の手順を実行します。

1. Mobile サーバーに接続します。
2. ユーザーを作成します。
3. パブリケーションを作成します。
4. パブリケーション項目を作成し、ウィニング・ルールを設定します。
5. パブリケーションにパブリケーション項目を追加します。
6. 必要に応じて、パブリケーション項目索引を作成します。
7. シーケンスを作成します。
8. ユーザーをパブリケーションにサブスクライブします。
9. クライアントのシーケンスをパーティション化します。
10. パブリケーションに対してユーザーのサブスクリプション・パラメータを定義します。
11. サブスクリプションをインスタンス化します。

次のいずれかを使用して、Mobile サーバーのパブリッシュ・サブスクリブ・モデルを実装できます。

表 3-4 パブリッシュ・サブスクリブ・モデルを実装する方法

実装メソッド	定義
パッケージ・ウィザード	お薦めする方法です。詳細は、 第 6 章「パッケージ・ウィザードの使用」 を参照してください。
Pure Java メソッド	Mobile Server Admin API は、Pure Java メソッドを使用して、Java プログラムから実行します。

注意： Consolidator Admin API では、大 / 小文字を区別されます。
Mobile Server Admin API では、大 / 小文字の区別はありません。

Mobile サーバーへの接続

ここでは、Java メソッドを使用して Mobile サーバーに接続する方法を説明します。

Java メソッドの例

```
ResourceManager.openConnection("MOBILEADMIN", "MANAGER");
```

Mobile Server Admin API のユーザー関数

ここでは、Mobile Server Admin API を使用してユーザーの作成、パスワードの変更、ユーザーの削除を行う方法を説明します。

ユーザーの作成

CreateUser() 関数を使用して、Mobile サーバー・ユーザーを作成できます。構文は次のとおりです。

```
public static boolean createUser(String userName, String password, String fullName, String privilege);
```

次の例では、表にリストされたパラメータを使用してユーザー「MOBILE」を作成します。

表 3-5 ユーザー作成パラメータの例

パラメータ	値	定義
userName	"MOBILE"	モバイル・クライアントのユーザー名を指定します。
password	"MOBILE"	ユーザー名 MOBILE のパスワードを指定します。
fullName	"MOBILEUSER"	ユーザー MOBILE のフル・ネームを指定します。
privilege	"C"	このユーザーが Mobile サーバーに接続できることを指定します。

Java メソッドの例

```
ResourceManager.createUser("MOBILE","MOBILE","MOBILEUSER","C");
```

パスワードの変更

Mobile サーバー・ユーザーのパスワードは、SetPassword() 関数を使用して変更できます。構文は次のとおりです。

```
public static void setPassword(String userName, String newpwd);
```

次の例では、ユーザー「MOBILE」のパスワードを変更します。

表 3-6 パスワード設定パラメータの例

パラメータ	値	定義
userName	"MOBILE"	モバイル・クライアントのユーザー名を指定します。
newpwd	"MOBILENEW"	モバイル・クライアントの新しいパスワードを指定します。

Java メソッドの例

```
ResourceManager.setPassword("MOBILE","MOBILENEW");
```

ユーザーの削除

dropUser() 関数を使用して、既存の Mobile サーバー・ユーザーを削除できます。構文は次のとおりです。

```
public static void dropUser(String userName);
```

次の例では、ユーザー「MOBILE」を削除します。

表 3-7 ユーザー削除パラメータの例

パラメータ	値	定義
userName	"MOBILE"	モバイル・クライアントのユーザー名を指定します。

Java メソッドの例

```
ResourceManager.dropUser("MOBILE");
```

パブリケーションの作成

ここでは、パブリケーションを作成する方法を説明します。

パブリケーション項目の定義

パブリケーション項目名は、26 文字に制限され、すべてのパブリケーションで一意である必要があります。パブリケーション項目は、表とビューの両方に対して定義できます。オブジェクトをパブリッシュするとき、ユーザーはオブジェクトの常駐するパブリケーション項目を指定する必要があります。更新可能な複数表ビューをパブリッシュする場合、特定の制限が適用されます。

- ビューには、主キーの定義された親表が含まれている必要があります。
- ビューの DML 操作に対して INSTEAD OF トリガーを定義する必要があります。
- ビューの実表は、すべてパブリッシュする必要があります。

データのサブセット化

パブリケーション項目を作成するとき、ユーザーは最大 8K の文字制限を持つ、パラメータ化された Select 文を定義できます。サブスクリプション・パラメータはこの時点で指定でき、レプリケーション時に、各クライアントに対してパブリッシュされるデータを制限するために使用されます。文字列代入値は、サブスクライブ時にパラメータ値を置き換えるために使用されます。

Consolidator Admin API の使用方法

CreatePublication 関数を使用して、パブリケーションを作成できます。構文は次のとおりです。

```
public static void CreatePublication(String name, int client_storage_type,  
    String client_name_template, String enforce_ri) throws ThrowableConsolidator
```

次の例では、表にリストしたパラメータを持つ「T_SAMPE1」という名のパブリケーションを作成します。

表 3-8 パブリケーション作成パラメータの例

パラメータ	値	定義
client_storage_type	Consolidator.OKPI_PALM	プラットフォーム・タイプを定義する定数。詳細は、『Oracle9i Lite Consolidator Admin API リファレンス』を参照してください。
client_name_template	'%s'	デフォルト。
enforce_ri	NULL	このパラメータは常に NULL です。

Java メソッドの例

```
Consolidator.CreatePublication("T_SAMPLE1", Consolidator.OKPI_PALM, "%s", null);
```

注意： クライアントの格納タイプとして Oracle Lite データベースを使用する場合、データベースには拡張子は付きません。

パブリケーション項目の作成

ここでは、パブリケーション項目を作成する方法を説明します。

Consolidator Admin API の使用方法

CreatepublicationItem 関数を使用してパブリケーション項目を作成できます。構文は次のとおりです。

```
public static void CreatePublicationItem(String name, String owner,
    String store, String refresh_mode, String select_stmt, String cbk_owner, String
    cbk_name) throws Throwable
```

次の例では、下の表にリストしたパラメータを持つ「P_SAMPLE1」という名のパブリケーション項目を作成します。

表 3-9 パブリケーション項目作成パラメータの例

パラメータ	値	定義
owner	SAMPLE1	SAMPLE1 がベース・オブジェクトの所有者であることを指定します。
store	ADDROLRL4P	ADDROLRL4P がベース・オブジェクト名であることを指定します。
refresh_mode	F または C	リフレッシュ・モードを高速または完全として定義します。詳細は、「 ビューの高速リフレッシュおよび更新操作 」の項を参照してください。
select_stmt	例を参照	sample1.addrolrl4p の指定した列からデータを選択します。
cbk_owner	NULL	コールバック・パッケージの所有者を NULL に指定します。詳細は、「 構成と適用を使用したコールバックのカスタマイズ 」の項を参照してください。
cbk_name	NULL	コールバック・パッケージの所有者名を NULL に指定します。詳細は、「 構成と適用を使用したコールバックのカスタマイズ 」の項を参照してください。

Java メソッドの例

```
Consolidator.CreatePublicationItem("P_SAMPLE1", "SAMPLE1", "ADDROLRL4P", "F" ,
    "SELECT \"LastName\",\"FirstName\", company, phone1, phone2, phone3, phone4,
    phone5, phone1id, phone2id, phone3id, phone4id, phone5id, displayphone,
    address, city, state, zipcode, country, title, custom1, custom2, custom3,
    custom4, note
FROM sample1.addrolrl4p" + " WHERE upper(company) > " + ":COMP", null, null);
```

Null Sync コールアウト

Mobile サーバーは、同期中に、クライアント・デバイスが Null Sync を試行中かどうかを示すコールアウトを作成します。Null Sync とは、クライアントにアップロード対象の変更がないことを言います。このコールアウトは、Mobile サーバーのリポジトリ内に PL/SQL パッケージを作成することで実装できます。パッケージでは、次の指定を行う必要があります。

```
create or replace package CUSTOMIZE as procedure
    NullSync(p_Client IN varchar2, p_NullSync as boolean);
end CUSTOMIZE;
```

パブリケーション項目名の取得

ここでは、パブリケーション項目名を取得する方法を説明します。

Mobile Server Admin API の使用方法

```
public static String getPublicationItemName(String Publication Item, int platform);
```

定数 `int platform` は、パブリケーション項目の作成対象となったプラットフォームを識別します。定数引数は、「ResourceManager.」で、プラットフォームに応じて、「WTG」、「PALM」、「EPOC」または「WINCE」を後に付けます。詳細は、『Oracle9i Lite Web-to-Go API リファレンス』を参照してください。

次の例では、Palm OS マシン・アプリケーションに対して「DEPT」というパブリケーション項目が返されます。

表 3-10 パブリケーション項目名取得パラメータの例

パラメータ	値	定義
PublicationItem	DEPT	スナップショットまたは表を識別する文字列名です。
platform	ResourceManager.PALM	スナップショットの作成対象となったプラットフォーム・タイプを表す定数です。

Java メソッドの例

```
ResourceManager.getPublicationItemName("DEPT",ResourceManager.PALM);
```

パブリケーション項目索引の作成

Mobile サーバーでは、クライアント・デバイス上の Oracle Lite データベースへの索引の自動配布をサポートしています。Mobile サーバーは、主キー索引をサーバー・データベースから自動的にレプリケートします。Consolidator Admin API では、クライアント・デバイスに一意、標準および主キーの各索引を明示的に配布するコールも同様に提供しています。

Consolidator Admin API の使用方法

CreatePublicationItemIndex 関数を使用して、パブリケーション項目索引を配布できます。構文は次のとおりです。

```
public static void CreatePublicationItemIndex(String name,  
                                             String publication_item, String pmode,  
                                             String columns) throws Throwable
```


この例では、パブリケーション項目 PSAMPLE1 に INDEX001 という名のパブリケーション項目索引を作成します。パブリケーション項目索引は、表に定義されたように ZIPCODE 列を含む標準索引です。

表 3-11 パブリケーション項目索引作成パラメータの例

パラメータ	値	定義
name	INDX001	作成するパブリケーション項目として INDX001 を定義します。
publication_item	P_SAMPLE1	索引のパブリケーション項目として P_SAMPLE1 を定義します。
pmode	I	索引モードを標準として定義します。
columns	ZIPCODE	索引に ZIPCODE 列を含めます。

Java メソッドの例

```
Consolidator.CreatePublicationItemIndex("INDX001", "P_SAMPLE1", "I", "ZIPCODE");
```

クライアント索引の定義

クライアント側の索引は、既存のパブリケーション項目に対して定義できます。3 種類の索引を指定できます。

- P - 主キー
- U - 一意
- I - 標準

注意：パブリケーション項目に「U」または「P」タイプの索引を定義した場合、サーバー上では重複キーの検査は行われません。パブリケーション項目のベース・オブジェクトに同一の制約がない場合、Mobile Sync は重複キー違反で失敗する可能性があります。詳細は、『Oracle9i Lite Consolidator Admin API リファレンス』を参照してください。

パブリケーションへのパブリケーション項目の追加

パブリケーション項目を作成した後、これをパブリケーションに対応付ける必要があります。定義を変更するには、パブリケーション項目を削除して新しい定義で作成しなおすか、要件に応じてスキーマの発展を使用します。詳細は、『Oracle9i Lite Consolidator Admin API リファレンス』の「DropPublicationItem」と「AlterPublicationItem」をそれぞれ参照してください。

Consolidator Admin API の使用方法

AddPublicationItem 関数を使用して、パブリケーション項目をパブリケーションに追加できます。構文は次のとおりです。

```
public static void AddPublicationItem(String publication, String item,
    String columns, String disabled_dml, String conflict_rule, String
    restricting-predicate, String weight) throws Throwable
```

次の例では、P_SAMPLE1 という名のパブリケーション項目をパブリケーション T_SAMPLE1 に追加します。追加されるパブリケーション項目のパラメータは次のとおりです。

表 3-12 パブリケーション項目追加パラメータの例

パラメータ	値	定義
publication	T_SAMPLE	新規項目を受け取るパブリケーションとして T_SAMPLE を定義します。
item	P_SAMPLE	追加するパブリケーション項目として P_SAMPLE を定義します。
columns	NULL	列の名前を変更しないことを指定します。
disabled_dml	NULL	DML を使用禁止するためのオプションを選択しないことを指定します。その他の値は、『Oracle9i Lite Consolidator Admin API リファレンス』を参照してください。
conflict_rule	S	競合解消においてサーバー優先を定義します。その他の値は、『競合ルールの定義』の項を参照してください。
restricting_predicate	NULL	制限選択条件は、パブリケーションにパブリケーション項目を追加するときに、パブリケーション項目に割り当てることができます。クライアントが高優先順位モードで同期される場合、選択条件はデバイスにブッシュされるデータを制限するために使用されます。
weight	NULL	NULL、またはマスター表に対してクライアント操作を実行する優先順位を指定する整数を指定します。詳細は、『表の比率の使用』の項を参照してください。

Java メソッドの例

```
Consolidator.AddPublicationItem("T_SAMPLE1", "P_SAMPLE1", null, null, "S");
```

読取り専用パブリケーション項目

パブリケーション項目は、DML を使用禁止にすることで、読取り専用として定義できます。詳細は、『Oracle9i Lite Consolidator Admin API リファレンス』を参照してください。

競合ルールの定義

パブリケーションにパブリケーション項目を追加する場合、クライアント「C」とサーバー「S」のいずれかを優先してレプリケーションの競合を解決するウィニング・ルールを指定できます。Mobile サーバーのレプリケーション競合は、次のいずれかの状況で検出されます。

- クライアントとサーバーで同一行が更新された場合。
- クライアントとサーバーの両方で同じ主キーを持つ行を作成した場合。
- クライアントが削除した行をサーバーが更新した場合。
- クライアントが更新した行をサーバーが削除した場合、Oracle8i のアドバンスト・レプリケーションとの互換性のためにレプリケーション・エラーと見なされます。
- クライアントのデータが実表に直接適用されない遅延データ処理があるシステム（たとえば、3 層アーキテクチャ）の場合、クライアントが最初に行を挿入し、次に同じ行を更新するときに、サーバーが実表にまだ行を挿入していない状況が発生する可能性があります。この場合、C\$ALL_CONFIG の DEF_APPLY パラメータが TRUE に設定されている場合、UPDATE でなく INSERT 操作が実行されます。結果として生じる PK 競合を解決するのは、アプリケーション開発者の責任です。ただし、DEF_APPLY が設定されていない場合、「NO DATA FOUND」例外が発生します（レプリケーション・エラーの処理は後述を参照してください）。
- NULL の使用違反や外部キー制約違反などのその他のエラーはすべてレプリケーション・エラーです。
- レプリケーション・エラーが自動的に解決されない場合、対応するトランザクションがロールバックされ、トランザクション操作は Mobile サーバーのエラー・キューに移動されます。Mobile サーバーのデータベース管理者は、後でこれらのトランザクション操作を変更して再実行したり、エラー・キューからページできます。

詳細は、『Oracle9i Lite Consolidator Admin API リファレンス』を参照してください。

表の比率の使用

表の比率は、パブリケーションとパブリケーション項目を関連付ける整数プロパティです。Mobile サーバーは、表の比率を使用して、マスター表にクライアント操作を適用する順序を次のように決定します。

1. 最初に、クライアントの INSERT 操作が、表の比率の低いものから高いものへと順に実行されます。
2. 次に、クライアントの DELETE 操作が、表の比率の高いものから低いものへと順に実行されます。

3. 最後に、クライアントの UPDATE 操作が、表の比率の低いものから高いものへと順に実行されます。

シーケンスの作成

Mobile サーバーでは、クライアント・デバイスに対してシーケンスのパーティション化とレプリケーションをサポートしています。

Java API の使用方法

CreateSequence 関数を使用してシーケンスを作成できます。構文は次のとおりです。

```
public static void CreateSequence(String name) throws Throwable
```

次の例では、CUSTOM1 という名のシーケンスを作成します。

Java メソッドの例

```
Consolidator.CreateSequence("CUSTOM1");
```

パブリケーションへのユーザーのサブスクライブ

次のいずれかの方法を使用して、パブリケーションに対してユーザー・サブスクリプションを作成できます。

Consolidator Admin API の使用方法

CreateSubscription 関数を使用して、ユーザーをパブリケーションにサブスクライブできます。構文は次のとおりです。

```
public static void CreateSubscription(String publication, String clientid)
    throws Throwable
```

次の例では、下の表にリストしたパラメータを使用して、クライアント DAVIDL をパブリケーション T_SAMPLE1 にサブスクライブします。

表 3-13 サブスクリプション作成パラメータの例

パラメータ	値	定義
publication	T_SAMPLE1	パブリケーションを T_SAMPLE1 として定義します。
clientid	DAVIDL	クライアントを DAVIDL として定義します。

Java メソッドの例

```
Consolidator.CreateSubscription("T_SAMPLE1", "DAVIDL");
```

クライアント・デバイスに対するシーケンスのパーティション化

Mobile サーバー・シーケンスを使用して、クライアント・デバイス上で一意の主キーを生成できます。レプリケート・シーケンスを使用するには、最初にこれを作成し、Mobile Sync Client ユーザー用にパーティション化する必要があります。

Consolidator Admin API の使用方法

CreateSequencePartition 関数を使用して、シーケンス・パーティションを作成できます。構文は次のとおりです。

```
public static void CreateSequencePartition(String name, String clientid,  
                                           long curr_val, long incr) throws Throwable
```

次の例では、表に指定したパラメータを使用して CUSTOM1 シーケンスをパーティション化します。

表 3-14 シーケンス・パーティション作成パラメータの例

パラメータ	値	定義
name	CUSTOM1	パーティション化するシーケンスとして CUSTOM1 を定義します。
clientid	DAVIDL	シーケンスを割り当てるクライアントとして DAVIDL を定義します。
curr_val	1000	シーケンスの初期値として 1000 を定義します。
incr	1	シーケンスの増分値として 1 を定義します。

Java メソッドの例

```
Consolidator.CreateSequencePartition("CUSTOM1", "DAVIDL", 1000, 1);
```

注意： シーケンスが一意であることを保証するには、一意の開始位置を使用し、サーバー上のクライアント数に等しい増分値を設定します。

パブリケーションに対するクライアント・サブスクリプション・パラメータの定義

一部のパブリケーションにはパラメータがあります。パブリケーションにパラメータがある場合、これらのパラメータをパブリケーションのサブスクリプション用に設定する必要があります。

Consolidator Admin API の使用方法

SetSubscriptionParameter 関数を使用して、サブスクリプションのインスタンス化パラメータを定義できます。構文は次のとおりです。

```
public static void SetSubscriptionParameter(String publication, String clientid,
                                           String param_name, String param_value) throws Throwable
```

次の例では、表にリストしたパラメータを使用して、クライアント DAVIDL をパブリケーション T_SAMPLE1 にサブスクライブします。

表 3-15 サブスクリプション・パラメータ設定パラメータの例

パラメータ	値	定義
publication	T_SAMPLE1	パブリケーション T_SAMPLE1 を定義します。
clientid	DAVIDL	クライアントを DAVIDL として定義します。
param_name	COMP	パラメータ名を COMP として定義します。
param_value	P	パラメータ値を P として定義します。

Java メソッドの例

```
Consolidator.SetSubscriptionParameter("T_SAMPLE1", "DAVIDL", "COMP", "'P'");
```

サブスクリプションのインスタンス化

サブスクリプションのパブリケーション・パラメータを設定し、サブスクリプションをインスタンス化してサブスクリプション処理を完了します。Mobile サーバーでサブスクリプションをインスタンス化する際、サブスクリプションの完全な内部表現が作成されます。

Consolidator Admin API の使用方法

InstantiateSubscription 関数を使用して、サブスクリプションをインスタンス化できます。構文は次のとおりです。

```
public static void InstantiateSubscription(String publication,
                                           String clientid) throws Throwable
```

次の例では、表に指定した値を使用して、クライアントのサブスクリプションをパブリケーションに対してインスタンス化します。

表 3-16 サブスクリプションのインスタンス化パラメータの例

パラメータ	値	定義
publication	T_SAMPLE1	パブリケーションを T_SAMPLE1 として定義します。
clientid	DAVIDL	クライアントを DAVIDL として定義します。

Java メソッドの例

```
Consolidator.InstantiateSubscription("T_SAMPLE1", "DAVIDL");
```

代替パブリケーション項目を使用したスキーマの発展

既存のパブリケーション項目に列を追加できます。これらの新しい列は、次に同期される時点で、全サブスクライブ・クライアントにプッシュされます。これは変更済み全パブリケーション項目の完全リフレッシュで達成されます。

- 管理者は、複数の列を追加できます。
- この機能は、すべてのクライアント形式に対してサポートされます。
- クライアントは、サーバーにスナップショット情報をアップロードしません。これは、クライアントがクライアント・データベースでスナップショットを直接変更できないことを意味します。たとえば、Palm 上で Mobile SQL を使用して表を変更することはできません。
- パブリケーション項目のアップグレードは、高優先順位の同期中は遅延されます。これは、ワイヤレスなどの低帯域幅ネットワークの場合に必要です。パブリケーション項目のアップグレードには常に、変更済みパブリケーション項目の完全リフレッシュが必要なためです。高優先順位フラグが設定されている間、高優先順位のクライアントは古いスナップショット形式を受信し続けます。
- サーバーは、パブリケーション項目のバージョンを最大 2 個サポートする必要があります。

パブリケーション項目の変更

これにより、既存のパブリケーション項目に列を追加できます。

Consolidator Admin API の使用方法

```
public static void AlterPublicationItem(String name, String select_stmt)
    throws Throwable
```

表 3-17 パブリケーション項目変更パラメータの例

パラメータ	値	説明
name	P_SAMPLE1	パブリケーション項目名を指定する文字列です。
select_stmt	select * from EMP	追加列の含まれた新規パブリケーション項目の Select 文です。

Java メソッドの例

Consolidator.AlterPublicationItem("P_SAMEPLE1", "select * from EMP");

パブリッシュ・サブスクライブ・メソッド固有の関数

次の機能には、ほとんどのアプリケーション設計で必要でない特殊関数が含まれています。

構成と適用を使用したコールバックのカスタマイズ

パブリケーション項目を作成する際、ユーザーは MGP バックグラウンド・プロセスの適用と構成段階でコールされるカスタマイズ可能パッケージを指定できます。これらのプロシージャにより、カスタマイズされたコードをプロセスに取り込むことができます。ユーザー・レベルおよびトランザクション・レベルでのカスタマイズを可能にするために、clientname と tranid が渡されます。

このプロシージャは、インキューのクライアント・データが適用される前にコールする必要があります。

procedure BeforeApply(clientname varchar2)

このプロシージャは、クライアントのデータがすべて適用された後でコールする必要があります。

procedure AfterApply(clientname varchar2)

このプロシージャは、tranid を持つクライアントのデータが適用される前にコールする必要があります。

procedure BeforeTranApply(tranid number)

このプロシージャは、tranid を持つクライアントのデータが適用された後にコールする必要があります。

procedure AfterTranApply(tranid number)

このプロシージャは、アウトキューが構成される前にコールする必要があります。

procedure BeforeCompose(clientname varchar2)

このプロシージャは、アウトキューが構成された後にコールする必要があります。

```
procedure AfterCompose(clientname varchar2)
```

カスタマイズ DML 操作の定義

パブリケーション項目を作成すると、そのパブリケーション項目の DML 操作のかわりにコールされるカスタマイズ PL/SQL プロシージャを指定できます。パブリケーションまたは pub_item あたり、mobile_dml_procedure は 1 つしか指定できません。たとえば、次の問合せを持つパブリケーション項目 A に対して、DML プロシージャを 1 つ持つとします。

```
select A,B,C from pub_item_a_table
```

A は主キーです。DML プロシージャのシグネチャは次のようになります。

```
any_schema.any_package.any_name (DML in CHAR(1), A in TYPE, B in TYPE, C in TYPE, A_
OLD in TYPE)
```

このプロシージャは、DML タイプとして「I」、「U」、「D」でコールされます。挿入および削除の場合、A_OLD は NULL になります。更新の場合、更新される行の主キーに設定されます。

仮想主キー

ベース・オブジェクトに主キーが定義されていないパブリケーション項目に対して、仮想主キーを指定できます。仮想主キーの作成および削除には、いくつかの方法があります。

仮想主キー列の作成

これは仮想主キー列を作成します。

Consolidator Admin API の使用方法

これは、主キーが仮想列である更新可能パブリケーション項目を作成するために使用します。

```
public static void CreateVirtualPKColumn(String owner, String store, String column)
throws Throwable
```

表 3-18 仮想主キー列作成パラメータ

パラメータ	値	説明
owner	SAMPLE_1	ベース・オブジェクトの所有者を指定する文字列です。
store	DEPT	ベース・オブジェクトを指定する文字列です。
column	DEPT_ID	主キー列を指定する文字列です。この文字列はカンマで区切ることができます。

Java メソッドの例

```
Consolidator.CreateVirtualPKColumn("SAMPLE1", "DEPT", "DEPT_ID");
```

仮想主キー列の削除

これにより、主キーを削除できます。

Consolidator Admin API の使用方法

```
public static void DropVirtualPKColumn(String owner, String store) throws Throwable
```

表 3-19 仮想主キー列の削除パラメータ

パラメータ	値	説明
owner	SAMPLE_1	ベース・オブジェクトの所有者を指定する文字列です。
store	DEPT	ベース・オブジェクトを指定する文字列です。

Java メソッドの例

```
Consolidator.DropVirtualPKColumn("SAMPLE1", "DEPT");
```

制限選択条件

選択条件は、パブリケーションにパブリケーション項目を追加するときに、パブリケーション項目に割り当てることができます。クライアントが高優先順位モードで同期される場合、選択条件はデバイスにプッシュされるデータを制限するために使用されます。このパラメータは、NULL にできます。このパラメータは、上級者向けです。

エラー・キューを使用した競合の解決

作成したパブリケーション項目ごとに、対応するエラー・キューが別個に作成されます。このキューの目的は、未解決の競合が原因で失敗するトランザクションを格納することです。管理者は、エラー・キュー・データまたはサーバーのエラー・キューを変更することで競合の解決を試みることができ、続いて Execute Transaction API コールを介してトランザクションの再適用を試みることができます。管理者は、Purge Transaction API コールを介してエラー・キューのページを試みることもできます。

トランザクションの実行

トランザクション実行関数は、Mobile サーバーのエラー・キュー内のトランザクションを再実行します。

Consolidator Admin API の使用方法

```
public static void ExecuteTransaction(String clientid, long tid)
    throws Throwable
```

表 3-20 トランザクション実行パラメータ

パラメータ	説明
clientid	Mobile Sync Client 名です。
tid	トランザクション ID です。

Java メソッドの例

```
Consolidator.ExecuteTransaction("DAVIDL", 100002);
```

トランザクションのパージ

トランザクションのパージ関数は、Mobile サーバーのエラー・キューからトランザクションをパージします。

Consolidator Admin API の使用方法

```
public static void PurgeTransaction(String clientid, long tid) throws Throwable
```

表 3-21 トランザクションのパージ・パラメータ

パラメータ	値	説明
clientid	DAVIDL	Mobile サーバー・ユーザー名です。
tid	100001	トランザクション ID です。

Java メソッドの例

```
Consolidator.PurgeTransaction("DAVIDL", 100001);
```

Palm OS Emulator のセットアップ

Mobile Development Kit (Palm 用) は、Windows Development システムにすでにインストールされています。Palm Computing Platform では、いくつかの共有ライブラリと Palm アプリケーションを使用できます。

- **okapi.prc** - これは、全 OKAPI 関数の含まれたコア共有ライブラリです。
- **okindex.prc** - パフォーマンスを向上させるために使用する、オプションの索引エンジンです。このライブラリをインストールしないと、**okCreateIndex** から **OK_INDEXLIB_NOTFOUND_ERROR** が戻されますが、その他の関数はすべて正常に動作します。
- **odbc.prc** - ODBC 共有ライブラリです。
- **ConsolidatorClient.prc** - Mobile Sync Client プログラムです。
- **msql.prc** - Mobile SQL を使用して開発者が Palm 用の Oracle Lite データベースを問い合わせることを可能にします。

okapi_noddl.prc は、**okapi.prc** の小規模なバージョンで、スナップショットの作成および削除の関数は含まれていません。このライブラリを使用して、オブジェクトの問合せ、挿入、更新および削除ができますが、クラス、グループまたはデータベースの追加や削除はできません。大部分のアプリケーションには、**odbc.prc**、**okapi.prc**、**okindex.prc** および **ConsolidatorClient.prc** を使用します。

重要： 同じデバイスに **okapi.prc** と **okapi_noddl.prc** を同時にインストールしないでください。

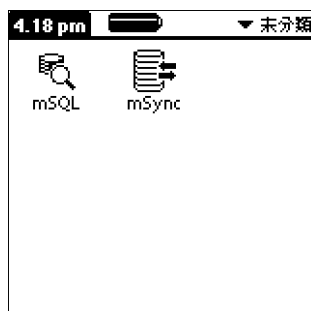
Palm 共有ライブラリとアプリケーションを Palm OS Emulator で使用するには、次の手順を実行します。

1. Palm OS Emulator アプリケーションを起動します。
2. *Oracle_Home¥Mobile¥SDK¥Palm* ディレクトリにある次の共有ライブラリとアプリケーション・ファイルを、Palm OS Emulator にドラッグ・アンド・ドロップします。
 - **okapi.prc**
 - **okindex.prc**
 - **odbc.prc**
 - **ConsolidatorClient.prc**
 - **msql.prc**

Palm Desktop 上に次の 2 つのアイコンが新しく表示されます。

- Mobile SQL アプリケーション用の **mSQL**
- Mobile Sync Client アプリケーション用の **mSync**

図 3-1 Mobile SQL アイコンと Mobile Sync Client アイコン



これらのファイルが正しくインストールされていることを確認するには、次のようにします。

1. Palm Desktop から「メニュー」を選択します。
2. メニュー選択バーから「アプリケーション」→「削除」を選択します。
3. インストール済みのアプリケーションと共有ライブラリが、すべてリストされます。
OL_SYNC、mSQL、ORACLE LITE 索引および ORACLE LITE DBMS を検索します。
4. 削除しないで終了するには、「終了」をクリックします。

図 3-2 インストールされるファイル



Emulator でのレプリケーションのテスト

mSYNC クライアント・アプリケーションを起動します。

Palm 用の Mobile クライアントでは、同期に次のパラメータが必要です。

表 3-22 Mobile クライアント・パラメータ

パラメータ	説明
ユーザー名	Mobile クライアント・ユーザー名です。このフィールドは大 / 小文字を区別しません。
パスワード	Mobile クライアント・パスワードです。このフィールドは大 / 小文字を区別します。
変更	このボックスは空白のままにします。
パスワードを保存	パスワードを保存するには、このチェック・ボックスを選択します。
http:// <mobile server>	Mobile サーバーの IP アドレスです。
プロキシ使用	必要であれば、選択します。

重要： レプリケーションのデモを行うために、この章では Sample Orders デモが使用されています。このサンプル・デモは、別個にインストールする必要があります。Sample Orders デモ・アプリケーションのインストールに関する情報は、『Oracle9i Lite インストレーションおよび構成ガイド』を参照してください。Sample Orders デモの詳細は、[第 4 章「Orders サンプル・アプリケーション」](#)を参照してください。

SAMPLE11 サンプル・アプリケーションを利用するには、次のユーザー名とパスワードを使用します。ユーザー作成アプリケーションおよびユーザー・スナップショットには、別個のユーザー名とパスワードを指定する必要があります。

- ユーザー名 = S11U1
- パスワード = MANAGER

1. Mobile クライアントを起動するには、「mSync」アイコンをクリックします。「mSync」画面が表示されます。

図 3-3 Oracle Mobile Sync Client 画面

2. 必要なフィールドに情報を入力します。「パスワードを保存」チェック・ボックスを忘れずに選択してください。「http://」フィールドには、画面名または IP アドレスを含めることができます。
3. 「適用」ボタンをクリックします。
4. 「同期」ボタンをクリックします。

構成、送信、受信および処理の各同期タスクの完了を示す進行状況バーが表示されます。進行状況バーには、各作業の完了に要する時間も表示されます。同期が正常に実行されると、同期成功画面が表示されます。同期に失敗すると、同期が失敗したというメッセージが表示されます。同期に失敗した原因を判断するには、サーバー管理者は Mobile サーバーのログファイルで追跡情報を表示できます。

Mobile サーバーのシステム・カタログ・ビュー

Mobile サーバーには、Mobile サーバーのコンポーネントを検索するためのシステム・カタログ・ビューが含まれています。Consolidator Admin API を使用せずにシステム・カタログ・ビューを変更しないでください。次の項目を検索するには、Mobile サーバーのシステム・カタログ・ビューを使用します。

- Mobile サーバーのユーザー
- パブリケーション
- サブスクリプション
- シーケンス
- シーケンス・パーティション

- スタンドアロン・パブリケーション項目
- パブリケーションに追加されたパブリケーション項目
- パブリケーション項目索引
- サブスクリプション・パラメータ

詳細は、[付録 A「Mobile サーバー・システム・カタログ・ビュー」](#)を参照してください。

Orders サンプル・アプリケーション

この章では、Mobile Development Kit（Palm 用）のサンプル・アプリケーションについて説明します。内容は次のとおりです。

- [概要](#)
- [Orders サンプル・アプリケーションのファイル](#)
- [Sample Orders アプリケーションを使用可能にする方法](#)
- [デバイスでのサンプル・アプリケーションのインストール](#)
- [デバイスへのデータのレプリケート](#)
- [Palm Computing Platform 用のサンプル・アプリケーションの使用方法](#)

概要

Mobile Development Kit（Palm OS マシン用）には、Oracle Lite データベースと Mobile サーバーを使用してレプリケーション可能なアプリケーションを作成する方法を示す Orders サンプル・アプリケーションが含まれています。

アプリケーションのユーザー・インタフェースには、受注のマスター・リスト、各受注の詳細な明細品目、各明細品目の数量確認が含まれています。サンプル・アプリケーションは、受注データを Oracle サーバー上の 2 つの表の間でレプリケートします。また、サンプルでは索引を使用して、個々の受注明細品目に対して実行される検索の速度を向上させています。

Orders サンプル・アプリケーションのこのリリースでは、Palm OS マシンおよび Palm OS Emulator をサポートしています。

Orders サンプル・アプリケーションのファイル

Palm 用 Orders サンプル・アプリケーションは、次のファイルで構成されます。

表 4-1 Palm Computing Platform のファイル

ディレクトリ/ファイル	説明
Oracle_Home¥Mobile¥SDK¥palm¥sdk¥examples¥cw¥okDemo¥okDemo.mcp	サンプル・アプリケーション・プロジェクト・ファイル。
Oracle_Home¥Mobile¥SDK¥palm¥sdk¥examples¥cw¥okDemo¥okDemo.prc	コンパイル済みアプリケーション・ファイル。すぐにインストールおよび実行できます。
Oracle_Home¥Mobile¥SDK¥palm¥sdk¥examples¥cw¥okDemo¥Src	サンプル・アプリケーションのソース・コード・ファイルの含まれるディレクトリ。

Sample Orders アプリケーションを使用可能にする方法

Palm OS Emulator で Sample Orders アプリケーションを使用するには、次のようにします。

- Oracle_Home¥Mobile¥SDK¥palm¥sdk¥examples¥cw¥okdemo ディレクトリの okdemo.prc ファイルを Palm OS Emulator にドラッグします。

Palm Computing Platform 用のサンプル・アプリケーションの使用方法

サンプル・アプリケーションでは次のフォームが使用されます。

- Main
- Order
- Item

Main フォーム

アプリケーションを起動すると、Main フォームが表示されます。Main フォームには、データベース内の受注がリストされます。

図 4-1 Palm 用の Main フォーム

ORACLE Orders			
ID	Date	Status	Name
100	07/04	Open	XYZ Comp
101	09/06	Open	Home Excha
102	04/10	Open	S&T
103	10/21	Open	Total Soluti
104	05/25	Open	Total Soluti
105	12/24	Open	S&T
106	05/09	Open	Partners
107	12/03	Open	S&T
108	10/21	Open	Smith & Wis
109	03/16	Open	Smith & Wis
110	06/25	Open	Lab Technic
111	06/06	Open	Estes

ソート順序の変更

Main フォームのソート列を変更するには、変更するソート列のヘッダーをクリックします。表の選択した列がソートされます。昇順と降順を切り替えるには、表ヘッダーの右側の矢印をクリックします。

Order フォーム

単一受注の情報を表示するには、表で受注品目をクリックします。Order フォームが表示されます。

図 4-2 Order フォーム

XYZ Company			
Order ID: 100			
Date: 07/05/1998			
Description: Original Order			
Status: Closed			
ID	Date	Description	
1	07/04	Nails (1000)	↑ ↓
2	07/04	Hammer	
3	07/04	Wrench	
4	07/04	Plyers	
5	07/04	Handsaw	

Done

Order フォームには、受注に関する次の情報が表示されます。

- 会社名
- 受注 ID
- 作成日
- 説明
- ステータス
- （受注の）受注品目のリスト

Main フォームに戻るには、「Done」ボタンをクリックします。

受注品目を表示または変更するには、リスト・ボックスで品目をクリックします。Order Item フォームが表示されます。

Order Item フォーム

Order Item フォームにより、受注の特定の品目に関する情報の表示や変更ができます。

図 4-3 Order Item フォーム

The screenshot shows a form titled "Order Item" with the following fields and values:

- Order ID: 3
- Date: 07/05/1998
- Description: Wrench
- Cost: 3.45
- Qty Ordered: 5
- Qty Shipped: 4
- Qty Recv'd: 3

At the bottom of the form are two buttons: "Save" and "Cancel".

Order Item フォームでは、次の情報を編集できます。

- 受注数量
- 出荷数量
- 受入数量

変更内容をコミットする場合は「Save」をクリックし、破棄する場合は「Cancel」をクリックして、Order フォームに戻ります。

デバイスでのサンプル・アプリケーションのインストール

Palm Desktop からインストール ツールを使用します。

注意： サンプル・アプリケーションをインストールするには、デスクトップ PC に Palm Desktop をインストールしておく必要があります。

サンプル・アプリケーションをインストールするには、次の手順を実行します。

1. Palm Desktop プログラム・グループのインストール ツールを起動します。インストール ツールの画面が表示されます。
2. 「追加」 ボタンをクリックします。
3. `Oracle_Home¥Lite¥pda40¥palm¥SDK¥examples¥cw¥okdemo` ディレクトリの **okdemo.prc** ファイルを選択します。
4. 「終了」 ボタンをクリックします。
5. デバイスを HotSync クレードルに置きます。
6. 「HotSync」 ボタンをクリックします。Orders サンプル・アプリケーションがデバイスにインストールされます。

デバイスへのデータのレプリケート

okdemo を実行する前に、Palm デバイス上の Mobile Sync Client で「同期」 ボタンをクリックする必要があります。

Mobile Sync Client 名として「S11U1」を、パスワードとして「MANAGER」を使用します。パスワードは大 / 小文字が区別されます。

Palm アプリケーションのプログラミング

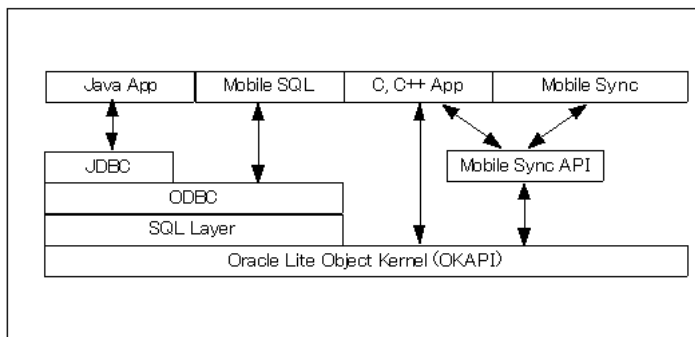
この章では、Palm Computing Platform デバイス用の Oracle Lite データベースのプログラミングについて説明します。内容は次のとおりです。

- [開発インタフェース](#)
- [Palm Computing Platform 用アプリケーションの開発](#)
- [Palm OS Emulator でのアプリケーションのテスト](#)

開発インタフェース

Oracle Lite データベース・アプリケーションを開発するには、ODBC または OKAPI のインタフェースを使用できます。

図 5-1 開発インタフェース



ODBC

Open Database Connectivity (ODBC) は、SQL データベースにアクセスするためのコール・レベル・インタフェースです。ODBC にはデータベースへの接続、SQL 文の実行、問合せ結果の取得などのための一連の関数が指定されています。ODBC は、ほとんどのデータベース・ベンダーでサポートされ幅広く使用されている仕様であり、最も可搬性があります。

ODBC for the Palm Computing Platform では、ODBC 3.0 インタフェースのサブセットをサポートしています。詳細は、[付録 C「ODBC サポート」](#)を参照してください。ODBC インタフェースの詳細は、ODBC 3.0 のドキュメントを参照してください。

OKAPI (Oracle Lite Object Kernel API)

Oracle Lite Object Kernel API (OKAPI) は、アプリケーションが Oracle Lite のオブジェクト・カーネルに直接アクセスできるようにする C/C++ 言語のインタフェースです。OKAPI インタフェースを使用して、限定されたシステム・リソースを最も有効に活用する Oracle Lite アプリケーションを作成できます。OKAPI を使用するアプリケーションは、オブジェクト・カーネルに直接アクセスするため、その他のインタフェースにアクセスするアプリケーションより少ないリソースで高速に実行できます。

OKAPI ライブラリは、トランザクション、問合せ、挿入、更新および削除などの一般的な操作のほとんどをサポートしています。Oracle Lite OKAPI の詳細は、『Oracle9i Lite (C および C++) オブジェクト・カーネル API リファレンス』を参照してください。

Mobile Sync API

Mobile Sync API を使用すると、アプリケーションはデータベースとの同期をクライアント・デバイスから開始および監視でき、**Mobile** サーバーから起動する必要はありません。デフォルトの転送方法は HTTP ですが、他の転送形式が使用できる場合は、それを指定できます。

Mobile Sync は、5 個のファンクション・コールと 1 つの制御構造で構成されます。

ocSessionInit

同期環境を初期化します。

構文

```
int ocSessionInit( ocEnv *env );
```

パラメータ

Env

戻される同期環境を保持する ocEnv 構造体バッファへのポインタ。

コメント

このコールは、ocEnv 構造体を初期化し、最後の ocSaveUserInfo() コールで保存されたユーザー設定をリストアします。ocEnv 構造体は、パラメータとして渡し、コール元で初期化する必要があります。ocSessionInit() コールの後、コール側でユーザー設定情報を上書きする場合は、ocSaveUserInfo() をコールします。

ocSessionTerm

同期環境を解放し、クリーン・アップします。

構文

```
int ocSessionTerm( ocEnv *env );
```

パラメータ

Env

ocSessionInit によって戻された環境構造体へのポインタ。

コメント

ocSessionInit() コールによって作成された構造体とメモリーをすべて消去します。ocSessionInit() と ocSessionTerm() は常に一対でコールする必要があります。

ocSaveUserInfo

ユーザー設定を保存します。

構文

```
int ocSaveUserInfo( ocEnv *env );
```

パラメータ

Env

同期環境へのポインタ。

コメント

この関数は、ユーザー設定をクライアント側のファイルまたはデータベースに保存します。環境構造体で指定された次の情報が保存されます。

- Username
- Password
- SavePassword
- NewPassword
- Priority
- Secure
- PushOnly
- SyncApps
- SyncNewPublication

これらのフィールドの使用方法は、「[Mobile Sync API のデータ構造](#)」の項を参照してください。

ocDoSynchroize

同期プロセスを開始します。

構文

```
int ocDoSynchronize( ocEnv *env );
```

パラメータ

Env

同期環境へのポインタ。

コメント

この関数は、同期サイクルを開始します。`syncDirection` が 0（デフォルト）の場合、ラウンドトリップ同期がアクティブにされます。`syncDirection` が 1 の場合、アップロード（送信）のみが実行されます。`syncDirection` が 2 の場合、ダウンロード（受信）のみが実行されます。クライアント・デバイスがサーバーからのデータのダウンロードを必要としない場合、アップロードのみの同期を実行すると便利です。

ocSetTableSyncFlag

選択同期用の表フラグを更新します。

構文

```
ocSetTableSyncFlag(ocEnv *env, const char* publication_name,  
const char* table_name, short sync_flag)
```

パラメータ

Env

同期環境へのポインタ。

publication_name

Consolidator Admin API の一部である `CreatePublication()` で使用されます。詳細は、『Oracle9i Lite Consolidator Admin API リファレンス』を参照してください。

`publication_name` が NULL の場合、`sync_flag` はパブリケーションのすべての項目に対して適用されます。

table_name

この文字列は、`CreatePublication()` の `client_name_template` と同じです。OKAPI の場合、これは `database_name + '.' + store` になります。ここで `store` は、`CreatePublicationItem()` の 3 番目のパラメータです。`CreatePublication()` および `CreatePublicationItem()` の詳細は、『Oracle9i Lite Consolidator Admin API リファレンス』を参照してください。

sync_flag

1 の場合、同期します。0 の場合、同期しません。`Sync_flag` は持続的に格納されているわけではありません。`ocDoSynchronize()` の前に毎回、`ocSetTableSyncFlag()` をコールする必要があります。

コメント

この関数により、クライアント・アプリケーションは、特定の表の同期方法を選択できます。

個々の表または個々のパブリケーションに対して `sync_flag` を設定します。`sync_flag = 0` の場合、表は同期されません。

Mobile Sync API のデータ構造

ocEnv

`ocEnv` は、内部的なメモリー・バッファと状態情報を保持するためにすべての Mobile Sync 関数によって使用されるデータ構造体です。この構造体を使用する前に、アプリケーションはこれを `ocSessionInit` に渡して環境を初期化する必要があります。

環境構造体には、プログラムで Mobile Sync 関数の動作方法を変更するために、コール側が更新できるフィールドも含まれています。

```
typedef struct ocEnv_s {
    // User infos
    char username[MAX_USERNAME];    // Mobile Sync Client Mobile Sync Client id
    char password[MAX_USERNAME];    // Mobile Sync Client password for
    // authentication during sync
    char newPassword[MAX_USERNAME]; // resetting Mobile Sync Client password
    // on server side if this field is not blank
    short savePassword;             // if set to 1, save 'password'
    char appRoot[MAX_PATHNAME];     // dir path on client device for deploying files
    short priority;                 // High priority table only or not
    short secure;                   // if set to 1, data encrypted over the wire
    enum {
        OC_SENDRECEIVE = 0,        // full step of synchronize
        OC_SENDONLY,              // send phase only
        OC_RECEIVEONLY,           // receive phase only

        // For Palm Only
        OC_SENDTOFILE,            // send into local file | pdb
        OC_RECEIVEFROMFILE        // receive from local file | pdb
    } syncDirection;               // synchronize direction

    enum {
        OC_BUILDIN_HTTP = 0,       // Use build-in Http transport method
        OC_USER_METHOD             // Use user defined transport method
    } trType;                       // type of transport

    ocError rexError;               // extra error code

    ocTransportEnv transportEnv;    // transport control information

    // GUI related function entry
    progressProc fnProgress;        // callback to track progress; this is optional
};
```

```

        // Values used for Progress Bar. If 0, progress bar won't show.
long totalSendDataLen;    // set by Mobile Sync API informing transport total
number of
                        // bytes to send; set before the first fnSend() is called
long totalReceiveDataLen; // to be set by transport informing Mobile Sync API
                        // total number of bytes to receive;
                        // should be set at first fnReceive() call.
void* userContext;       // user defined context
void* ocContext;         // internal use only
short logged;            // internal use only
long bufferSize;        // send/receive buffer size, default is 0
short pushOnly;          // Push only flag
short syncApps;          // Application deployment flag
} ocEnv;

```

ocTransportEnv

この構造体は、組み込み転送関数を上書きするために使用します。構造体に関数のリストを指定することにより、アプリケーションは同期エンジンによって使用されるトランスポート層に対して独自の実装を定義できます。

```

typedef struct ocTransportEnv_s {
void* ocTrInfo;          // transport internal context
                        // for built-in Http, mapped to ocTrHttp
connectProc fnConnect;   // plug-in callback to establish a connection from
                        // device to server
disconnectProc fnDisconnect; // plug-in callback to dismantle connection from
                        // device to server
sendProc fnSend;         // plug-in callback to send data
receiveProc fnReceive;   // plug-in callback to receive data
}ocTransportEnv;

```

Palm Computing Platform 用アプリケーションの開発

Palm Computing Platform は 32 ビットの POSE ベースのオペレーティング・システムです。CodeWarrior などの Palm Computing Platform 開発ツールを使用するとプロジェクトに Palm OS ファンクション・コールをリンクできます。CodeWarrior を使用して、OKAPI ライブラリへの静的なリンクや実行時の動的なリンクも可能です。

静的リンクでは、リンクした関数はアプリケーション・コードの一部になります。静的リンクには次の利点があります。

- デバッグ時にライブラリ・コードを表示できるため、プログラムがよりデバッグしやすくなります。
- アプリケーションで必要とするコードのみがリンクされます。使用しない関数はリンクされません。

動的リンクには次の利点があります。

- コード・サイズが小さくなります。ライブラリが共有されるため、アプリケーションのコード・サイズはかなり小さくなります。
- 管理が容易です。ライブラリは実行時にバインドされるため、OKAPI が変更されてもユーザーはプログラムを再コンパイルしたり、再リンクする必要がありません。

「[Metrowerks 社の CodeWarrior でのプロジェクトの作成](#)」の項で説明するように、OKAPI ライブラリをアプリケーションに静的にリンクするには、プロジェクトに **okapi.lib** を含めます。ライブラリを動的にリンクするには、**okapi_stub.lib** を含めます。**okapi_stub.lib** 使用時の依存性の詳細は、第 2 章の「[ディレクトリ構造](#)」の項を参照してください。

ツール

Oracle Lite for Palm Computing Platform アプリケーションを作成するために次の開発ツールが使用できます。

- [Oracle Lite Mobile SQL for Palm Computing Platform](#)
- [PUMATECH 社の Satellite Forms](#)
- [GNU GCC PRC ツール](#)
- [Metrowerks 社の CodeWarrior](#)
- [Palm OS Emulator](#)

Oracle Lite Mobile SQL for Palm Computing Platform

Oracle Lite Mobile SQL for the Palm Computing Platform を使用すると、Palm Computing Platform デバイス上の Oracle Lite データベースにアクセスし操作できます。詳細は、「[Oracle Lite Mobile SQL for Palm Computing Platform の使用方法](#)」を参照してください。

PUMATECH 社の Satellite Forms

PUMATECH 社の Satellite Forms は、Palm Computing Platform デバイス用の高速ビジュアル・アプリケーション開発ツールです。詳細は、「[Satellite Forms でのアプリケーションの作成](#)」を参照してください。

GNU GCC PRC ツール

Palm Computing Platform 用の GNU ツールは、インターネット経由で無料ソフトウェアとして入手できます。Windows の 32 ビット・システム用 GNU C コンパイラ (GCC) とデバッグは、次の場所からダウンロードできます。

<http://www.land-j.com/gccwin32.html>

ほとんどの UNIX や Macintosh プラットフォームでは、ツールセットも使用できます。

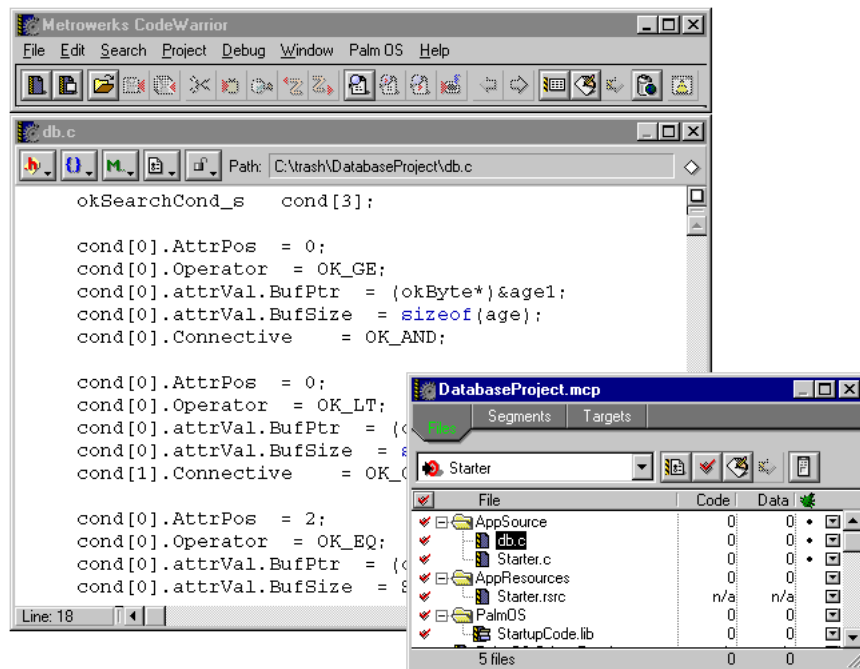
ただし、静的リンクのサポートは開発中です。

Metrowerks 社の CodeWarrior

Metrowerks 社の CodeWarrior リリース 7 は、Palm Computing Platform 用の公式開発ツールとして Palm Computing 社が薦める完全対話型の開発環境です。CodeWarrior には、プロジェクト・ビューア、コード・エディタ、リソース・コンストラクタ、コンパイラおよびリモート・デバッガが含まれています。

CodeWarrior のプロジェクト・ビューアとエディタを次に示します。

図 5-2 Code Warrior の画面の例



CodeWarrior は、Metrowerks 社の Web サイトから購入するか、体験版をダウンロードできます。

CodeWarrior で OKAPI アプリケーションを起動する方法の手順は、「[Metrowerks 社の CodeWarrior でのプロジェクトの作成](#)」を参照してください。

Palm OS Emulator

Palm OS Emulator (POSE) は、デスクトップ・コンピュータ上で Palm Computing Platform アプリケーションを実行できるようにするハードウェア・エミュレータです。Palm OS Emulator をデバッグ・バージョンの Palm OS ROM と自動ソフトウェア・テストの Gremlin とともに使用すると、特にアプリケーションのテストとデバッグが簡単になります。

Palm OS Emulator は、Palm Computing 社の Web サイトから入手できます。Windows の 32 ビットおよび Macintosh プラットフォームで使用できます。

開発システムでの Oracle Lite データベースの作成

モバイル・データベースを作成するには、次の 3 つのオプションがあります。

1. 宣言型 — 詳細は、「[パブリケーション項目の宣言型作成](#)」の項を参照してください。
2. プログラムを使用 — 詳細は、「[プログラムでのパブリケーション項目の作成](#)」の項を参照してください。
3. 手動 — 「[Oracle Lite Mobile SQL for Palm Computing Platform の使用方法](#)」の項を参照してください。

場合によっては、Mobile サーバーとは別にクライアント・データベースを作成します。Oracle Lite Mobile SQL for the Palm Computing Platform を使用すると、Palm Computing Platform デバイス上に直接データベースを作成して操作できます。OKAPI ファンクショナル・コールを使用してもクライアント・データベースを作成できます。Mobile サーバーとは別に作成したデータベースはレプリケート可能にはなりません。

Oracle Lite Mobile SQL for Palm Computing Platform の使用方法

Oracle Lite Mobile SQL for the Palm Computing Platform を使用すると、Palm Computing Platform デバイス上で Oracle Lite データベースを作成、アクセスおよび操作できます。Oracle Lite Mobile SQL を使用して次のことができます。

- データベースの作成
- 表の表示
- SQL 文の実行

Oracle Lite Mobile SQL は、サポートされるすべてのプラットフォームに標準のツールです。

Oracle Lite Mobile SQL for Palm Computing Platform のインストール

Oracle Lite Mobile SQL for the Palm Computing Platform をインストールするには、次の手順を実行します。

1. Palm Desktop プログラム・グループのインストール ツールを起動します。インストール ツールの画面が表示されます。

2. 「追加」ボタンをクリックし、`Oracle_Home¥Mobile¥SDK¥palm¥runtime¥bin` ディレクトリの **msql.prc** ファイルを選択します。
3. 「開く」ボタンをクリックします。インストールするファイルのリストにファイル名が表示されます。
4. 「終了」ボタンをクリックします。メッセージ・ボックスで「OK」ボタンをクリックします。
5. Palm Computing Platform デバイスを HotSync クレードルに置き、「HotSync」ボタンをクリックします。

Oracle Lite Mobile SQL for Palm Computing Platform の起動

Oracle Lite Mobile SQL for the Palm Computing Platform を起動するには、「アプリケーション」アイコンをクリックし、「MSQL」アイコンをクリックします。「Oracle Lite Mobile SQL」画面にデバイス上の Oracle Lite データベースのリストが表示されます。接続先のデータベース名を選択します。

図 5-3 Mobile SQL



Oracle Lite Mobile SQL でのデータベースの新規作成

Palm Computing Platform デバイスでデータベースを新規作成するには、次の手順を実行します。

1. 「メニュー」アイコンをクリックし、「DB 作成」を選択します。ダイアログ・ボックスが表示され、データベース名を要求されます。
2. 新規データベースの名前を入力します。「OK」ボタンをクリックします。Oracle Lite Mobile SQL により新規データベースが作成されます。

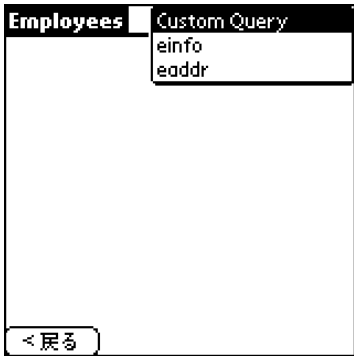
Oracle Lite Mobile SQL での表の表示

Oracle Lite Mobile SQL を使用して、データベースの内容を表形式で表示できます。

データベースに接続すると、「テーブルの参照」フォームが表示されます。データベース内の表を表示するには、次の手順を実行します。

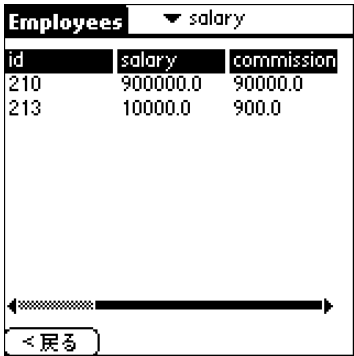
- 1. 「テーブルを選択」ボタンをクリックします。リスト・ボックスに現在のデータベースにある表のリストが表示されます。

図 5-4 Mobile SQL の表



- 2. リストから表を選択します。Oracle Lite Mobile SQL により表が表示されます。

図 5-5 Mobile SQL の表の例



結果が 1 画面に表示できない場合は、画面最下部および右側のスクロールバーを使用してデータを表示します。

表のソート順序を変更するには、変更するソート列名をクリックします。ソートの昇順と降順を切り替えるには、列名を 2 回クリックします。列名の最後にある正符号 (+) はソート順が昇順であることを示し、負符号 (-) は降順を示します。

Oracle Lite Mobile SQL での SQL の使用方法

Oracle Lite Mobile SQL の SQL ビュー機能を使用すると、SQL 文を実行して Palm Computing Platform デバイスに結果を表示できます。SQL ビューでは、SQL 文を使用して次のことができます。

- 表の作成と削除
- 索引の作成と削除
- 行の挿入、削除、選択および更新

SQL ビューをオープンするには、次の手順を実行します。

1. 「メニュー」アイコンをクリックします。
2. 「ビュー」メニューから「SQL 文実行」を選択します。「SQL 参照」画面が表示されます。

SQL ビューで文を実行するには、次の手順を実行します。

1. 画面最上部のテキスト・フィールドに SQL 文を入力します。SQL ビュー機能は大 / 小文字を区別します。たとえば、サンプル表 ORD_MASTER からすべての値を選択する場合は、次のように入力する必要があります。

```
select * from ORD_MASTER
```

2. 「実行」ボタンをクリックします。結果は「結果」テキスト・ボックスに表示されます。

図 5-6 SELECT 文の例



Metrowerks 社の CodeWarrior でのプロジェクトの作成

ここでは、Metrowerks CodeWarrior 統合開発環境を使用して、Mobile Development Kit (Palm 用) OKAPI アプリケーションを作成する方法を説明します。始める前に、リンクが必要です。

- 静的にリンクする場合は、プロジェクトに **okapi.lib** ライブラリを含めます。アプリケーション開発には、静的ライブラリの使用をお薦めします。
- 動的にリンクする場合は、プロジェクトに **okapi_stub.lib** を含めます。このライブラリには、共有ライブラリ関数をコールするためのスタブが含まれています。
okapi_stub.lib を使用するには、2 つの共有ライブラリ **okapi.prc** と **okindex.prc** のいずれか一方または両方が必要です。

OKAPI を静的にリンクする場合も動的にリンクする場合も、プログラムに同じヘッダー・ファイルを含めます。

注意： MetroWerks 社の CodeWarrior リリース 7 の下で実行するようコンパイルされた **oktest.r7.mcp** プロジェクト・フォルダを出発点として使用することをお薦めします。これは、適切なコンパイル・パラメータで事前に構成されています。次のディレクトリにあります。

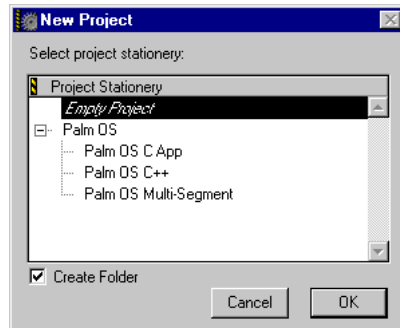
Oracle_Home¥SDK¥Mobile¥palm¥sdk¥examples¥cw¥oktest

リリース 6 を使用している場合は、**.r6.mcp** というラベルの付いたファイルを使用してください。

静的リンクか動的リンクかを決めた後、プロジェクトを作成します。

1. CodeWarrior IDE プログラムを実行します。
2. 「File」メニューから「New Project」を選択します。Project stationery のオプションのダイアログ・ボックスが表示されます。

図 5-7 「New Project」メニュー



CodeWarrior 環境では、stationery は、初期プロジェクト設定、ソース・ファイルおよびリソース・ファイルを提供する初期プロジェクトです。「New Project」ダイアログ・ボックスで、C、C++ またはマルチセグメントのプロジェクトを開始できます。

3. 「New Project」ダイアログ・ボックスにリストされている stationery を選択し、「OK」ボタンをクリックします。新規プロジェクトの名前を要求するダイアログ・ボックスが表示されます。
4. 開発ディレクトリを選択し、プロジェクト名を入力します。

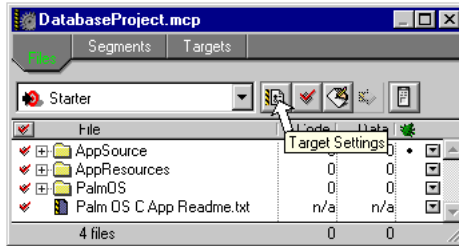
プロジェクトに OKAPI ライブラリを追加します。

1. 「Project」メニューの「Add Files」を選択します。
2. 適切な OKAPI ライブラリ・ファイル（静的リンク用 **okapi.lib** または動的リンク用 **okstub.lib**）を検索し、選択します。

次に、検索パスに OKAPI ヘッダー・ファイルのあるディレクトリを追加します。

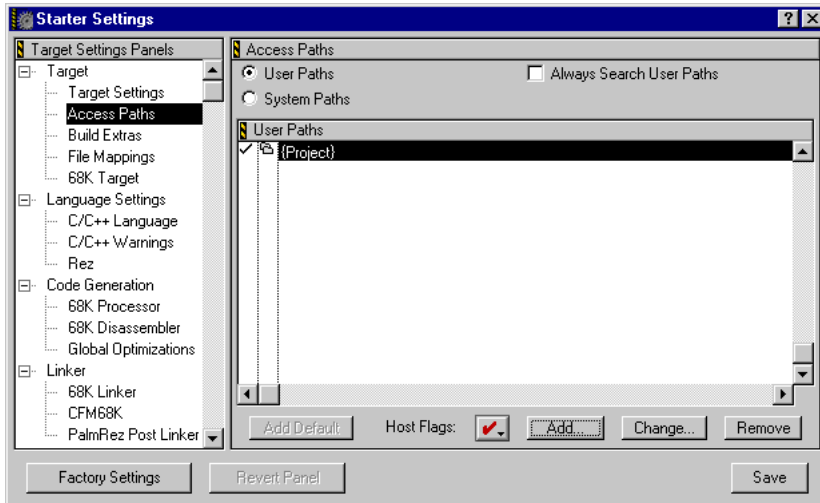
1. プロジェクト・ウィンドウのツールバーの「Target Settings」ボタンをクリックします。

図 5-8 ターゲットの設定



2. 「Starter Settings」ダイアログ・ボックスの「Access Paths」をクリックします。

図 5-9 アクセス・パス



3. 「User Paths」ラジオ・ボタンを選択します。
4. 「Add」ボタンをクリックし、OKAPI ヘッダー・ファイルの入っているディレクトリを選択します。

5. 「Save」 ボタンをクリックします。

これで、ファイルを追加してコーディングを開始できます。

プロジェクトを作成すると、バイナリのプログラム・ファイル（たとえば、**Starter.prc**）が生成されます。プログラムをテストするには、Palm Computing Platform デバイスまたは Palm OS Emulator にインストールします。

詳細は、Metrowerks 社の CodeWarrior のドキュメントを参照してください。OKAPI の使用方法の詳細は、『Oracle9i Lite（C および C++）オブジェクト・カーネル API リファレンス』を参照してください。

Satellite Forms でのアプリケーションの作成

ここでは、PUMATECH 社の Satellite Forms を使用して Oracle Lite アプリケーションを開発する過程を説明します。

要件

PUMATECH 社の Satellite Forms を使用して Oracle Lite for Palm Computing Platform アプリケーションを開発するには、次のものがが必要です。

- Oracle Mobile サーバーおよび Mobile Sync Client
- PUMATECH Satellite Forms Enterprise Edition
- Mobile サーバーのサポートする任意の転送
- Palm Computing Platform デバイスまたはエミュレータ

Satellite Forms アプリケーションの作成

パブリケーションを作成した後、Satellite Forms アプリケーションを作成します。ここでは、Satellite Forms でアプリケーションを開発する過程を簡単に説明します。詳細は、Satellite Forms のドキュメントを参照してください。

Satellite Forms アプリケーションを作成するには、次の手順を実行します。

1. Satellite Forms Application Designer をオープンします。
2. 「Create New Application」を選択します。
3. 「Contents of Application」フィールドで、「Properties」をダブルクリックします。
「Application Properties」ダイアログ・ボックスが表示されます。
4. 必要な情報を入力します。
5. 「Oracle Lite Compatible Tables」が選択されていることを確認します。「OK」ボタンをクリックします。
6. 「Tables」アイコンをマウスの右ボタンでクリックし、「Import Table」を選択します。

7. 「ODBC CONNECTION」 ウィンドウで、「ODBC driver for ORACLE」を選択します。
8. ユーザー名、パスワードおよび接続文字列を入力します。
9. 「ODBC tables」 から、インポートする表を選択します。
10. 「Forms」 アイコンをマウスの右ボタンでクリックし、「Properties」を選択します。
11. フォームにリンクする表を選択します。
12. 「Form 1」 をダブルクリックします。
13. フォーム・レイアウトを作成します。
14. 「Edit Controls」 プロパティを適切な表列にリンクします。
15. 「Oracle Lite Compatible Tables」を選択します。
16. 「Creator ID」を「SMSF」に設定します。
17. アプリケーションを保存して Satellite Forms を終了します。

これで、アプリケーションを Palm Computing Platform デバイスにロードし、データを Mobile サーバーと同期させる準備ができました。詳細は、[第3章「同期」](#)を参照してください。

Satellite Forms のサンプル・アプリケーション

Oracle Lite for Palm のこのリリースには、Satellite Forms のサンプル・アプリケーションが含まれています。このアプリケーションを使用するには、Palm Computing Platform デバイスまたはエミュレータに、**EMPDEPT.sfa** ファイルをインストールします。このファイルは、`Oracle_Home¥Mobile¥SDK¥palm¥Samples¥SatForms` ディレクトリに置かれます。

注意： サンプル・アプリケーションを初めて起動するとき、エラー・メッセージが表示される可能性があります。このメッセージは無視してかまいません。

Palm OS Emulator でのアプリケーションのテスト

次の項では、Metrowerks 社の CodeWarrior および Satellite Forms を使用して開発したアプリケーションをテストするために必要な手順を説明します。

Metrowerks 社の CodeWarrior で作成したアプリケーションのテスト

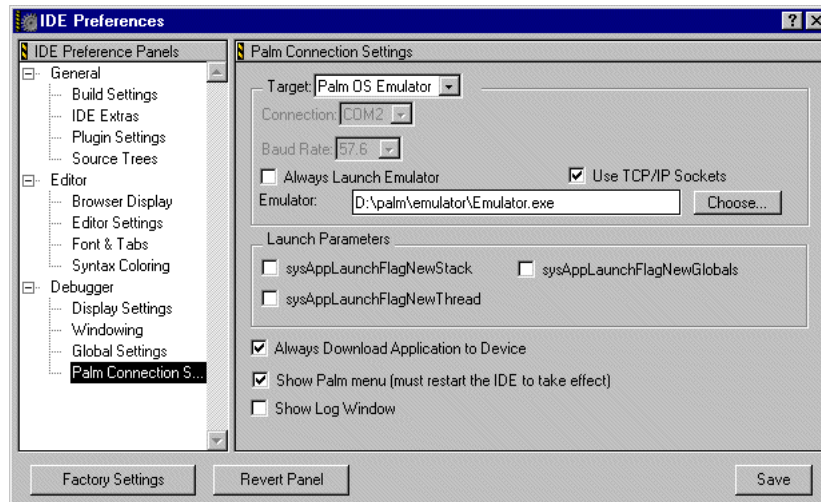
開発者は、**Starter.prc** というバイナリ・ファイルを生成する Metrowerks プロジェクトを作成する必要があります。アプリケーションをテストするには、次の手順を実行します。

1. Palm OS Emulator を起動します。
2. **Starter.prc** を Palm OS Emulator にドラッグします。
3. Palm OS Emulator からアプリケーションを起動します。
4. アプリケーションをテストします。

CodeWarrior に組み込まれたツールを使用してアプリケーションをデバッグする場合は、デバッグ・ツールを構成した後でアプリケーションを実行する必要があります。

1. 「Edit」メニューから「Preferences」を選択し、「Debugger」ブランチをオープンし、「Palm Connection Setting」ブランチ・メニューをオープンします。続いて、「Target」に「Palm OS Emulator」を選択します。

図 5-10 CodeWarrior の「Preferences」



2. プロジェクト・ファイルをオープンします。
3. メニュー・バーから「Palm」を選択し、「Launch Emulator」オプションを使用して Palm OS Emulator を起動します。
4. 「Project」メニューから「Enable Debugger」を選択します。
5. 「Project」メニューから「Debug」を選択するか、[F5] キーを押します。
6. アプリケーションを実行します。

PUMATECH 社の Satellite Forms で作成したアプリケーションのテスト

PUMATECH 社の Satellite Forms を使用して作成したアプリケーションには、機能するためのランタイム環境に必要な Satellite Forms Application ファイル (*.sfa) が作成されます。次の手順は、Palm OS Emulator での Satellite Forms アプリケーションの実行を説明します。

1. Satellite Forms ランタイム環境 SF31RDK.prc を Palm OS Emulator にドラッグします。
2. Satellite Forms アプリケーション（たとえば、EMPDEPT.sfa）を Palm OS Emulator にドラッグします。
3. Mobile Development Kit (Palm OS マシン用) のランタイム・ライブラリを使用可能にしていない場合は、Oracle_Home¥Mobile¥SDK¥Palm¥SDK¥runtime¥ の次のディレクトリをオープンして使用可能にします。
 - a. このディレクトリにある ¥shlib ディレクトリをオープンし、okapi.prc と okindex.prc を Palm OS Emulator にドラッグ・アンド・ドロップし、使用可能にします。
 - b. ¥bin ディレクトリをオープンし、odbc.prc、ConsolidatorClient.prc および msq1.prc を Palm OS Emulator にドラッグ・アンド・ドロップし、使用可能にします。
4. Palm OS Emulator から Satellite Forms アプリケーションを起動します。
5. アプリケーションをテストします。

Palm OS マシンでのアプリケーションのテスト

Palm OS マシンでのアプリケーションのテストは、次の 2 つのステップに分けられます。

- データ・レプリケーションのテスト
- アプリケーション・ロジックのテスト

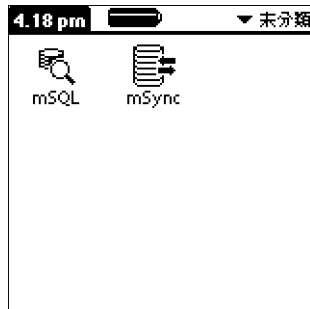
開発者は、最初にデータベース / パブリケーション項目をテストした後、アプリケーション・ロジックをテストすることをお勧めします。パブリケーション項目はすでに作成されているので、レプリケーションを使用して Palm OS マシンにスナップショットを作成するのが便利であり、お勧めします。

Palm OS マシンでの Mobile Development Kit のインストール

使用できる Palm OS マシンおよび開発システムに付属のクレードルを使用して、次の手順を実行します。

1. 「スタート」メニューからインストール ツール プログラムを実行するか、**instapp.exe** ファイルを検索して実行します。
2. 「追加」ボタンをクリックします。
3. 「ブラウズ」ボタンをクリックして、`Oracle_Home¥Mobile¥SDK¥Palm¥runtime¥shlib` ディレクトリにナビゲートします。ここには、2つのサブディレクトリがあります。`shlib` ディレクトリからは、**odbc.prc**、**okapi.prc** および **okindex.prc** をインストール・リストに追加します。`bin` サブディレクトリからは、**ConsolidatorClient.prc** と **msql.prc** をインストール・リストに追加します。
4. 「終了」ボタンをクリックして、インストール ツールをクローズします。
5. HotSync アプリケーションが実行されていない場合は、これを実行します。
6. Palm OS マシンを HotSync クレードルに置きます。クレードルの「HotSync」ボタンをクリックします。
7. 共有ライブラリをインストールした後、Palm OS マシンを再起動します。

図 5-11 Mobile Development Kit アイコン



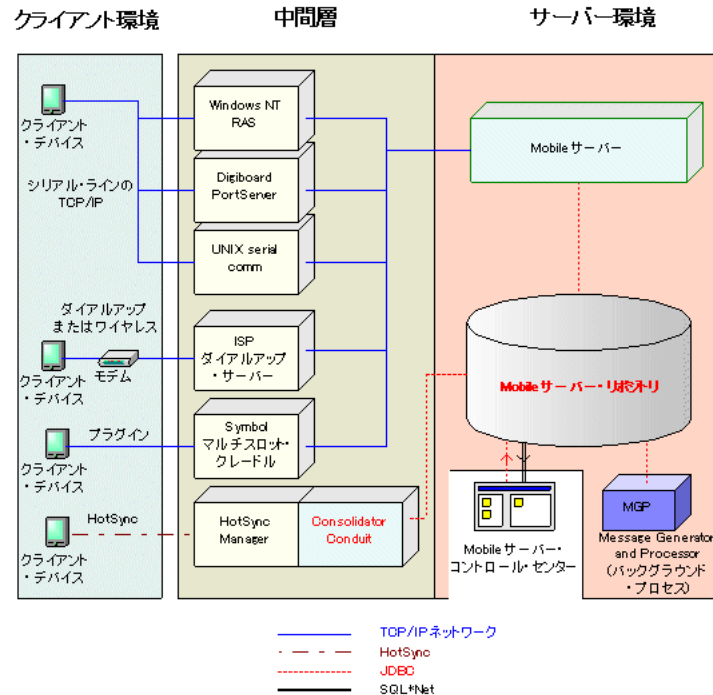
Palm ワークスペースに、新しいアイコンが2つ表示されます。

- Mobile SQL アプリケーション用の **mSQL**
- Mobile Sync Client アプリケーション用の **mSync**

Palm OS マシンでのアプリケーションのインストール

1. 「スタート」メニューからインストール ツール プログラムを実行するか、**instapp.exe** ファイルを検索して実行します。
2. 「追加」 ボタンをクリックします。
3. 「ブラウズ」 ボタンをクリックして次のファイルを検索し、インストール・リストに追加します。
 - **starter.prc** または自分の Metrowerks アプリケーション
 - Satellite Forms ランタイム環境。Satellite Forms v. 3.1 の場合は **sf31rdk.prc**
 - 自分の Satellite Forms アプリケーション (*.sfa)
4. 「終了」 ボタンをクリックして、インストール ツールをクローズします。
5. HotSync アプリケーションが実行されていない場合は、これを実行します。
6. Palm OS マシンを HotSync クレードルに置きます。クレードルの「HotSync」 ボタンをクリックします。
7. アプリケーションをインストールした後、Palm OS マシンを再起動します。

図 5-12 Mobile サーバー用の転送の構成



Palm Computing Platform デバイスでの転送の構成

HTTP、Scout または HotSync を使用して、Palm Computing Platform デバイスでの転送を構成できます。次の事項を説明します。

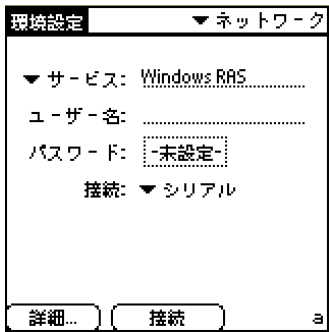
- HTTP の使用
- HotSync の使用

HTTP の使用

HTTP を転送方式として構成するには、次の手順を実行します。

1. Mobile サーバーをインストールおよび構成し、実行します。必要に応じて、Windows の「コントロール パネル」の「サービス」メニューを見て、Mobile サーバーが稼働していることを確認します。
2. Palm Computing Platform デバイスの「環境設定」画面で、メニューから「ネットワーク」を選択します。「ネットワーク環境設定」画面が表示されます。

図 5-13 「ネットワーク環境設定」画面



3. 「ネットワーク環境設定」画面のフィールドに必要な情報を入力します。

表 5-1 Palm デバイスのネットワーク設定

フィールド	値
サービス	Windows NT RAS
ユーザー名	ユーザーのログオン名
パスワード	ユーザーのパスワード
電話	00

注意： 複数のデバイスに対して同じ RAS ユーザー名とパスワードを使用できます。

4. 「詳細 ...」 ボタンをクリックします。
「詳細」 ダイアログ・ボックスが表示されます。
5. 「IP アドレス: 自動」 が選択されていることを確認します。
6. 「スクリプト」 ボタンをクリックします。
「ログイン スクリプト」 ダイアログ・ボックスが表示されます。
7. プルダウン・メニューを使用して次の事項を入力します。

表 5-2 Palm のログイン・スクリプト

フィールド	値
改行を送信	不要
時間待ち	1
送信	CLIENT
プロンプト待ち	CLIENTSERVER

図 5-14 ログイン画面



8. 「OK」ボタンをクリックして「ログイン スクリプト」ダイアログ・ボックスをクローズします。
9. 「OK」ボタンを再度クリックして「詳細」ダイアログ・ボックスをクローズします。
10. デバイスを HotSync クレードルに置きます。
11. 「接続」ボタンをクリックして接続をテストします。PPP 接続が成功した場合、デバイスはデータの同期の準備が整っています。

HTTP Sync への HotSync 方式の再割当て

Palm Computing Platform デバイスで HTTP 転送を構成した後、HotSync 方式を再割当てして HTTP をデフォルトの同期方式にする必要があります。

HotSync 方式を HTTP 同期に再割当てするには、次の手順に従います。

1. Palm Computing Platform デバイスに **OracleLauncher.prc** をインストールします。
2. Palm OS マシンで環境設定アイコンを選択します。「環境設定」画面が表示されます。
3. 「環境設定」画面でメニューから「ボタン」を選択します。「ボタン設定」画面が表示されます。
4. 「HotSync」をクリックします。

- 5. 「クレードル」メニューの「OracleLauncher」を選択します。「OK」をクリックします。
- 6. デバイスを HotSync クレードルに置き、「HotSync」ボタンをクリックします。HTTP Sync 方式がアクティブになります。

HotSync の使用

HotSync を転送方式として構成するには、次の手順を実行します。

- 1. Oracle_Home¥Mobile¥SDK¥Palm¥HotSync から Oracle_Home¥Bin に cnshscnd.dll をコピーします。
- 2. Oracle_Home¥Bin ディレクトリの regset.exe ファイルを実行して、Mobile サーバー HotSync パイプを HotSync に登録します。
- 3. HotSync マネージャを再起動します。

Mobile Sync Client for Palm 用のセットアップ

Mobile Sync Client は、Palm OS マシン上のローカル・データベースを、接続先のシステム（この場合は、開発システム）で稼働中の Mobile サーバーのインスタンスに同期させます。Mobile Sync Client for Palm では、次のパラメータを同期に使用します。

表 5-3 Mobile クライアント・パラメータ

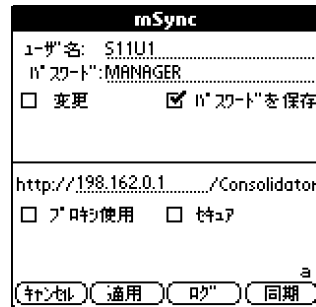
パラメータ	説明
ユーザー名	Mobile クライアント・ユーザー名です。このフィールドは大 / 小文字を区別しません。
パスワード	Mobile クライアント・パスワードです。このフィールドは大 / 小文字を区別します。
変更	このボックスは空白のままにします。
パスワードを保存	パスワードを保存するには、このチェック・ボックスを選択します。
http:// <mobile server>	Mobile サーバーの IP アドレスです。
プロキシ使用	必要であれば、選択します。

SAMPLE11 サンプル・アプリケーションを利用するには、次のユーザー名とパスワードを使用します。ユーザー作成アプリケーションおよびユーザー・スナップショットには、別個のユーザー名とパスワードを指定する必要があります。

- ユーザー名 = S11U1
- パスワード = MANAGER

1. Mobile Sync Client を起動するには、「ORACLE」アイコンをクリックします。
「mSync」画面が表示されます。

図 5-15 「mSync」画面



mSync

ユーザー名: S11U1
パスワード: MANAGER

☐ 変更 ☒ パスワードを保存

http://198.162.0.1/Consolidator

☐ プロキシ使用 ☐ キャッシュ

キャンセル 適用 OK 同期

2. 必要なフィールドに情報を入力します。「パスワードを保存」チェック・ボックスを忘れずに選択してください。
3. 「適用」ボタンをクリックします。
4. 「同期」ボタンをクリックします。

構成、送信、受信および処理の各同期タスクの完了を示す進行状況バーが表示されます。進行状況バーには、各タスクの完了に要する時間も表示されます。同期が正常に実行されると、同期成功画面が表示されます。同期に失敗すると、同期が失敗したというメッセージが表示されます。同期に失敗した原因を判断するには、サーバー管理者は Mobile サーバー・ログファイルで追跡情報を表示できます。

パッケージ・ウィザードの使用

この章では、Mobile Development Kit のパッケージ・ウィザード・ユーティリティについて説明します。内容は次のとおりです。

- [パッケージ・ウィザードの概要](#)
- [パッケージ・ウィザードの起動](#)
- [新規アプリケーションの命名](#)
- [アプリケーション・ファイルのリスト表示](#)
- [データベース情報の入力](#)
- [レプリケーション用スナップショットの定義](#)
- [アプリケーションの完了](#)

パッケージ・ウィザードの概要

パッケージ・ウィザードは、次の目的に使用できるグラフィカル・ツールです。

- Win32、Palm、Windows CE または EPOC プラットフォーム用の新規 Mobile サーバー・アプリケーションを作成する。
- 既存の Mobile サーバー・アプリケーションを編集する。

新規 Mobile サーバー・アプリケーションを作成するときは、コンポーネントを定義し Mobile サーバー・リポジトリにパブリッシュします。場合によっては、既存の Mobile サーバー・アプリケーションのコンポーネントの定義を編集することがあります。たとえば、アプリケーションの新規バージョンを開発する場合は、パッケージ・ウィザードを使用してアプリケーション定義を更新します。パッケージ・ウィザードを使用すると、アプリケーション・コンポーネントを **.jar** ファイルにパッケージ化することもできます。

パッケージ・ウィザードの起動

パッケージ・ウィザードを起動するには、DOS プロンプトで次のように入力します。

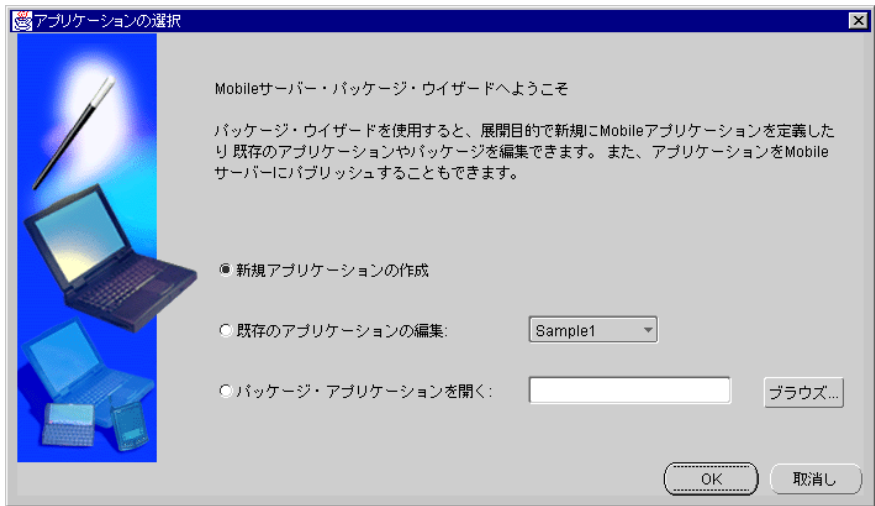
wtgpack

パッケージ・ウィザードが表示され、デフォルトで「ようこそ」パネルが表示されます。「ようこそ」パネルでは、次の機能を使用して、パッケージ化されたアプリケーションの作成、編集またはオープンができます。

表 6-1 「ようこそ」パネルのオプション

機能	説明
新規アプリケーションの作成	このオプションを選択すると、新規アプリケーションを定義できます。
既存のアプリケーションの編集	このオプションを選択すると、既存のアプリケーションを編集できます。隣にあるドロップダウン・リストから既存のアプリケーションを選択できます。
パッケージ・アプリケーションを開く	このオプションを選択すると、 .jar ファイルとしてパッケージ化されているアプリケーションを選択できます。隣にあるフィールドにパッケージ・アプリケーションの名前を入力するか、「ブラウズ」ボタンを使用してアプリケーションを選択します。

図 6-1 「ようこそ」 パネル



新規アプリケーションの命名

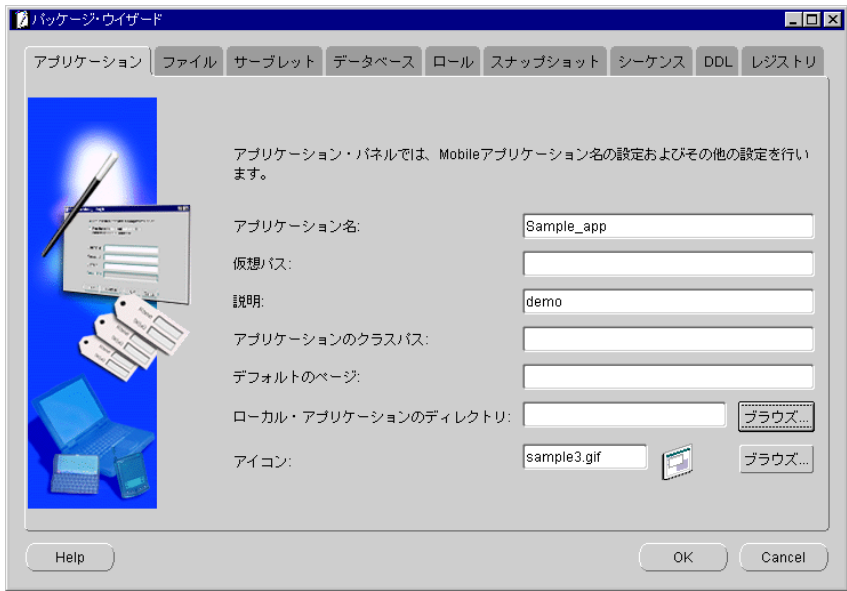
「アプリケーション」 パネルは、Mobile サーバー・アプリケーションに名前を付けて、このアプリケーションを Mobile サーバー上のどこに格納するかを指定するために使用します。このパネルに含まれるフィールドは次のとおりです。

表 6-2 「アプリケーション」 パネルのオプション

フィールド	説明	必須
アプリケーション名	Windows アプリケーションの名前。	YES
仮想パス	サーバー・リポジトリのルート・ディレクトリから Web-to-Go アプリケーション自身にマップされるパス。仮想パスにより、アプリケーションのディレクトリ構造全体を参照する必要がなくなります。アプリケーションに一意の ID も提供します。	YES
説明	Windows アプリケーションの概要。	YES
アプリケーションのクラスパス	セミコロン (;) で区切られたクラス・ディレクトリ。Web-to-Go は、ここに指定された順序でアプリケーション・クラスを検索します。	NO

フィールド	説明	必須
デフォルトのページ	Web-to-Go アプリケーションのエントリ・ポイントとして機能する Web ページのサーバー位置。これは、「リポジトリのディレクトリ」に対する相対パスです。たとえば、サーバー・ディレクトリが /apps でデフォルト・ページが index.html の場合、「デフォルトのページ」は /apps/index.htm になります。デフォルト・ページはサブレットにできます。ユーザーがデフォルト・ページを指定しない場合、汎用ページが発行されます。	NO
ローカル・アプリケーションのディレクトリ	ローカル・マシン上でこのアプリケーションの全コンポーネントが含まれているディレクトリです。この場所は、入力するかまたは「ブラウズ」ボタンをクリックして選択します。	NO
アイコン	Mobile サーバー・ワークスペースで Mobile サーバー・アプリケーションのアイコンとして使用される GIF イメージです。隣にあるフィールドにアイコンの名前を入力するか、「ブラウズ」ボタンを使用してアイコン・ファイルを選択します。	NO

図 6-2 「アプリケーション」パネル



プラットフォーム・ファイルの命名

ローカル・アプリケーションのディレクトリが必要です。アプリケーションに Win32、Palm、EPOC または Windows CE のファイルが含まれている場合は、これらのファイルをローカル・アプリケーション・ディレクトリの次のサブディレクトリに配置します。

- Win32 ファイルは、「win32」というサブディレクトリに入れます。
- Palm ファイルは、「palm」というサブディレクトリに入れます。
- EPOC ファイルは、「epoc」というサブディレクトリに入れます。
- Windows CE ファイルはすべて、「wince」というサブディレクトリに入れます。

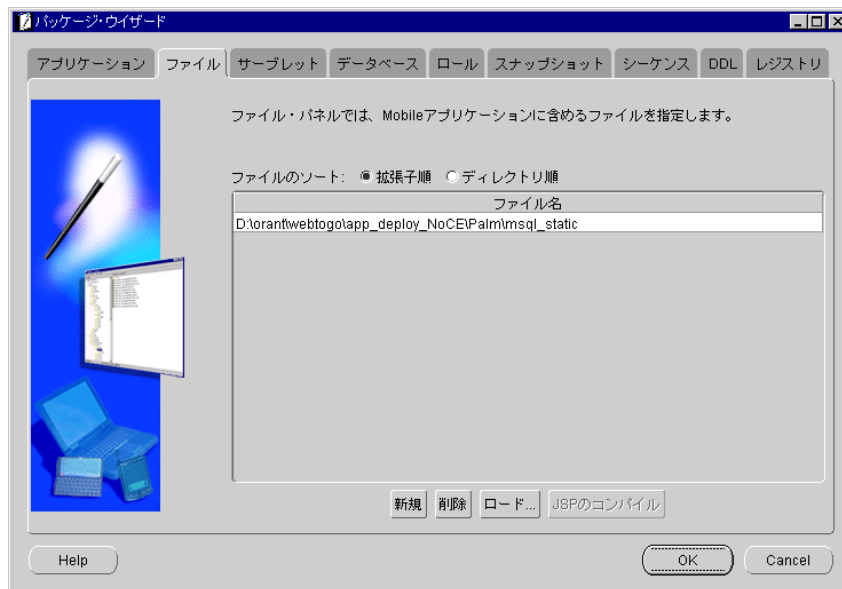
アプリケーション・ファイルのリスト表示

「ファイル」パネルは、アプリケーション・ファイルを表示し、ファイルがローカル・マシン上のどこにあるかを示すために使用します。パッケージ・ウィザードはローカル・アプリケーションのディレクトリの内容を分析し、各ファイルのローカル・パスを表示します。このパネルに含まれるフィールドは次のとおりです。

表 6-3 「ファイル」パネルのオプション

フィールド	説明	必須
ローカルのパス	各 Mobile サーバー・アプリケーション・ファイルの絶対パスです。リスト内の各エントリには、各ファイルまたはディレクトリの完全なパスが含まれています。	YES
ファイルのソート:	<ul style="list-style-type: none">■ 拡張子順 - ファイルを拡張子ごとにアルファベット順に表示します。■ ディレクトリ順 - ファイルをディレクトリごとにアルファベット順に表示します。	

図 6-3 「ファイル」 パネル



「ファイル」パネルにリストされているファイルは、すべて追加、削除またはロードできます。新規アプリケーションを作成する場合、「ファイル」パネルに進むと、ローカル・ディレクトリにリストされているすべてのファイルが自動的に分析されロードされます。既存アプリケーションを編集する場合は、「ロード」ボタンを使用して個々のファイルをロードできます。

ソート

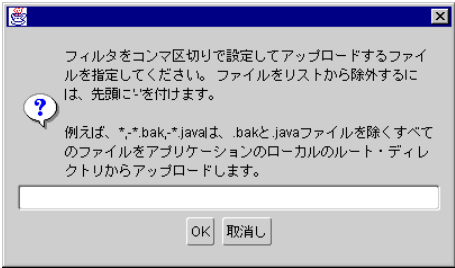
ファイルは、拡張子または含まれているディレクトリごとにソートできます。ファイルをソートするには、「拡張子順」または「ディレクトリ順」ラジオ・ボタンをクリックします。

フィルタ

「ロード」ボタンをクリックすると「入力」ダイアログ・ボックスが表示されます。「入力」ダイアログ・ボックスは、アップロード・プロセスからのアプリケーション・ファイルを含めるか除外するかを指定する（カンマで区切られた）フィルタのリストを作成するために使用します。ファイルを除外するには、ファイル名の前に負符号（-）を付けます。たとえば、**.prc** および **.exe** 拡張子の付いたファイルを除くすべてのファイルをロードするには、次のように入力します。

***,-*.prc,-*.exe**

図 6-4 「フィルタ」 パネル



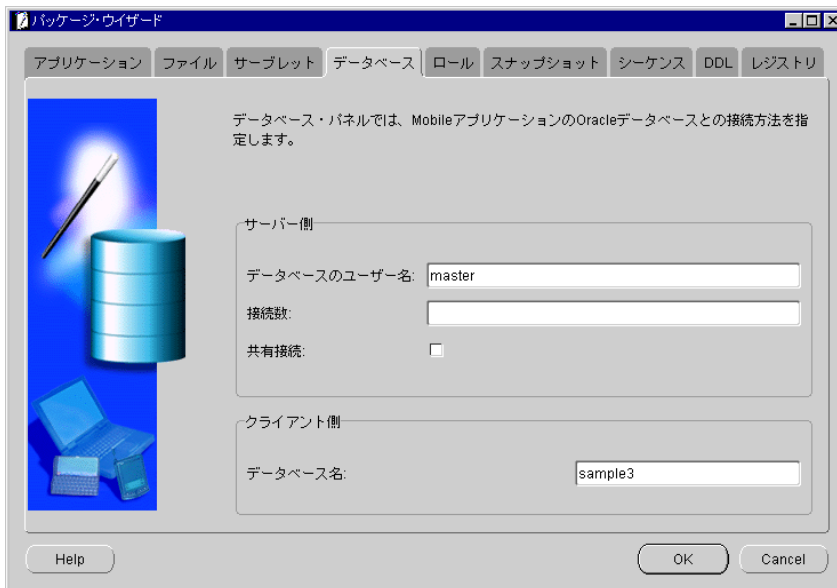
データベース情報の入力

「データベース」パネルは、Oracle サーバー上のデータベースを指定するために使用します。このパネルに含まれるフィールドとチェックボックスは次のとおりです。

表 6-4 「データベース」パネルのオプション

フィールド	説明	必須
データベースのユーザー名	データの同期をとるためにアプリケーションによって使用される Mobile サーバー・データベースのユーザー名。デフォルトは「master」です。	YES
接続数	Web-to-Go アプリケーションが Mobile サーバーに対して持つことのできる同時接続数。	NO
共有接続	チェックした場合、複数のサブレットで同一のデータベース接続を共有できます。複数のサブレットで接続を共有することにより、Oracle の使用可能な最大同時接続数を超えることができます。ただし、1 つのサブレットがコミットを実行すると、その接続を共有する他のすべてのサブレットに自動的に影響します。	NO
データベース名	Mobile クライアント・デバイス上で接続中のデータベースの名前。アプリケーションで使用される名前である必要があります。	YES

図 6-5 「データベース」 パネル



レプリケーション用スナップショットの定義

「スナップショット」パネルは、アプリケーションのレプリケーション・スナップショットを作成するために使用します。スナップショットは、全アプリケーション間で一意にする必要があります。パッケージ・ウィザードでは、次のプラットフォームに対してスナップショットを作成できます。

- Win32
- Palm
- EPOC
- Windows CE

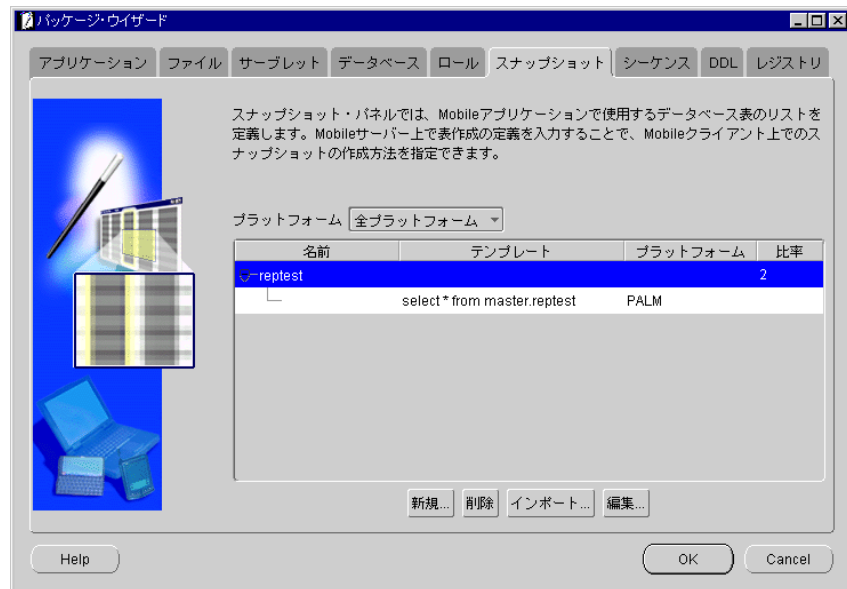
このパネルに含まれるフィールドは次のとおりです。

表 6-5 スナップショット・パラメータ

フィールド	説明	必須
名前	Mobile サーバー・アプリケーションに関連付けられているスナップショット（複数も可）の名前（複数も可）です。この名前は、Oracle Lite データベース上のデータベース表と同じである必要があります。	YES
テンプレート	利用可能なスナップショット・テンプレートのリストです。テンプレートとは、スナップショットの作成に使用される SQL 文です。テンプレートには変数を含められます。テンプレートを Mobile サーバーにパブリッシュした後は、Mobile サーバー・コントロール・センターを使用して、ユーザー固有のテンプレート変数を指定できます。ただし、Mobile サーバー・コントロール・センターではスナップショット・テンプレートは変更できません。	YES
プラットフォーム	スナップショットのプラットフォームです。ユーザーは異なるプラットフォームに対してスナップショットを作成できます。クライアントのデータと同期をとった場合、クライアント・アプリケーションを実行中のプラットフォームに適したスナップショットのみが取得されます。	YES
比率	これは、データベース表の同期順序を決定する正の整数です。「マスター」/「ディテール」関連を持つ表の場合、マスター表はディテール表より先にレプリケートされるように、低い比率を持つ必要があります。	YES

フィールド	説明	必須
全プラットフォーム	<p>現在のスナップショットのプラットフォームのドロップダウン・リストです。ドロップダウン・リストには、次のプラットフォームをすべて含めることができます。</p> <ul style="list-style-type: none">■ Win32■ Palm■ EPOC■ Windows CE <p>ドロップダウン・リストからプラットフォームを選択すると、そのプラットフォーム用のスナップショットのみが「スナップショット」パネルに表示されます。たとえば、「全プラットフォーム」ドロップダウン・リストから「Win32」を選択すると、Win32 ベースのスナップショットのみが表示されます。ドロップダウンから「全プラットフォーム」オプションを選択すると、現在使用中のプラットフォームごとにすべてのスナップショットが表示されます。ユーザーが新しいスナップショットを追加した場合、ドロップダウン・リストには追加のプラットフォームがリスト表示されます。</p>	NO

図 6-6 「スナップショット」パネル



「スナップショット」パネルで「新規」ボタンまたは「削除」ボタンをクリックすると、「スナップショット」パネルにスナップショットを追加または削除できます。スナップショットはインポートまたは編集することもできます。

注意：「スナップショット」パネルから複数のスナップショットをインポートできますが、「新規表」ダイアログ・ボックスから新規表を作成するときにインポートできるスナップショットは1つのみです。

新規スナップショットの作成

新規スナップショットを作成するには、「新規」ボタンをクリックします。「新規スナップショット」表が表示されます。

図 6-7 「新規スナップショット」パネル

新規スナップショット

サーバーWIN32PALMEPOCWINCE

☒ クライアントで作成

☒ 更新可能

競合解決:

☒ サーバー優先☐ クライアント優先

DMLプロシージャ

リフレッシュ・タイプ:

☒ 高速リフレッシュ☐ 完全リフレッシュ

テンプレート:

select * from master.reptest

索引

名前	タイプ	列
----	-----	---

新規

削除

OK

取消し

新規スナップショットを作成するには、新規表の画面の次の機能を変更します。

表 6-6 「新規スナップショット」パネルのオプション

機能	説明
名前	サーバー上の新規表の名前です。
プラットフォーム	<p>タブに、スナップショットのデバイス・プラットフォームが表示されます。現在、Mobile サーバーでは次のプラットフォームがサポートされます。</p> <ul style="list-style-type: none"> ■ Win32 ■ Palm ■ EPOC ■ Windows CE
SQL の生成	この機能を選択すると、パッケージ・ウィザードにより、データベース表の作成を可能にする SQL スクリプトへの出力情報が収集されます。
比率	この表に対する表の比率を設定できます。表の比率は、同期時の競合を解決するために使用されます。詳細は、「 表の比率の使用 」の項を参照してください。
SQL	名前付きの表を定義する SQL 文を表示します。この文は変更できます。

スナップショットのインポート

Oracle8 データベースまたは Oracle Lite データベースからスナップショットをインポートするには、「インポート」ボタンをクリックします。接続を指定していない場合は、データベース接続ウィンドウが表示されます。

注意： 一度指定したデータベース接続は、パッケージ・ウィザードの残りの部分でも使用されます。Oracle8 と Oracle Lite データベースを切り替える必要がありますが、すでに接続が確立されている場合は、パッケージ・ウィザード・アプリケーションを完全に終了して、再度 **wtgpack.exe** を実行します。

図 6-8 「データベースへの接続」パネル



スナップショットのインポート元の Oracle8 データベースのユーザー名、パスワードおよびデータベース URL を入力します。「表」ウィンドウが表示されます。

注意： Oracle8 データベースのデータベース URL を入力するときは、次の書式を使用します。

`jdbc:oracle:thin:@o8host:o8 port:SID`

たとえば、`jdbc:oracle:thin:@o8-db:1521:orcl` と指定します。

Oracle Lite の場合は、`jdbc:polite:webtogo` を使用します。

図 6-9 「表」パネル



表のインポート元のパブリケーション項目を選択してから、表を選択します。「追加」をクリックしてから「閉じる」をクリックします。パッケージ・ウィザードの「スナップショット」パネルに表が表示されます。

スナップショットの編集

スナップショットを編集するには、「スナップショット」パネルからスナップショットを選択し、「編集」をクリックします。表の編集ウィンドウが表示されます。

図 6-10 「スナップショットの編集」パネル

スナップショットの編集

サーバー: WIN32 PALM EPOC WINCE

名前: reptest

比率: 2

☐ SQLの生成

SQL:

OK 取消し

スナップショットを編集するには、表の編集ウィンドウの次の機能を変更します。

表 6-7 「スナップショットの編集」パネルのオプション

機能	説明
プラットフォーム	タブに、スナップショットのデバイス・プラットフォームが表示されます。 <ul style="list-style-type: none">■ Win32■ Palm■ EPOC■ Windows CE

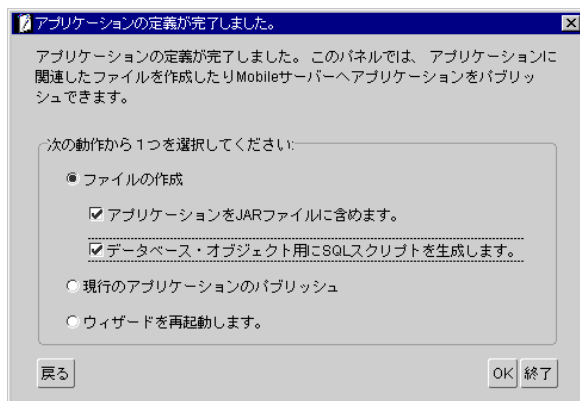
機能	説明
クライアントで作成	このチェックボックスが選択されていると、次のことを実行できます。 <ul style="list-style-type: none">■ 更新可能スナップショットの作成。■ スナップショット・テンプレートの作成。管理者は、Mobile サーバー・コントロール・センターを使用して、複数の異なるユーザー用変数をこのテンプレートに対してインスタンス化できます。
更新可能	このチェックボックスが選択されていると、名前付きの表の更新可能スナップショットが Mobile サーバーの表に作成されます。
テンプレート	名前付きの表のスナップショット・テンプレートを表示します。スナップショット・テンプレートは変更できます。管理者は、 Mobile サーバー・コントロール・センターを使用して、複数の異なるユーザー用変数をこのテンプレートに対してインスタンス化できます。テンプレート変数の詳細は、「 レプリケーション用スナップショットの定義 」の項を参照してください。

アプリケーションの完了

パッケージ・ウィザードの全パネルを完了すると、次のオプションの含まれた「アプリケーションの定義が完了しました。」ウィンドウが表示されます。

- ファイルの作成
- 現行のアプリケーションのパブリッシュ
- ウィザードを再起動します。

図 6-11 「アプリケーションの定義が完了しました。」ウィンドウ



XML ファイル

パッケージ・ウィザードは、アプリケーション情報のすべてを XML ファイルに自動的に出力します。パッケージ・ウィザードはローカル・マシン上で XML ファイルを保持します。さらに、Mobile サーバーに XML ファイルをパブリッシュするオプションも提供しています。XML ファイルを Mobile サーバーにパブリッシュできるのは、サーバーが実行されているときのみです。

JAR ファイルの作成

「ファイルの作成」オプションを使用すると、アプリケーション・コンポーネントを **.jar** ファイルにパッケージ化できます。アプリケーション・コンポーネントを **.jar** ファイルにパッケージ化するには、「ファイルの作成」をクリックしてから「アプリケーションを JAR ファイルに含めます。」をクリックします。**.jar** ファイルの位置を指定するよう要求されます。

SQL ファイルの作成

SQL スクリプトを生成するには、「ファイルの作成」をクリックしてから「データベース・オブジェクト用に SQL スクリプトを生成します」をクリックします。生成されたスクリプトは、アプリケーションのローカル・ルート・ディレクトリの下に SQL サブディレクトリ内に入れます。SQL スクリプトは、サーバー側の表、シーケンスおよび DDL に関して指定した情報を使用します。この SQL スクリプトをデータベースに対して実行して、これらのデータベース・オブジェクトを作成できます。

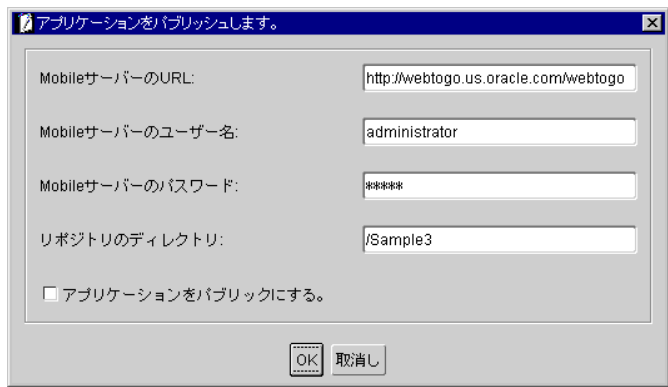
パッケージ・ウィザードの再起動

「ウィザードを再起動します。」オプションを使用すると、パッケージ・ウィザードを再起動できます。このオプションを使用すると、パッケージ・ウィザードの「ようこそ」パネルに戻ります。パッケージ・ウィザードを再起動するには、「ウィザードを再起動します。」をクリックしてから「OK」をクリックします。

アプリケーションのパブリッシュ

「現行のアプリケーションのパブリッシュ」オプションを使用すると、パッケージ・ウィザードで作成し定義したアプリケーションをパブリッシュできます。Mobile サーバー・アプリケーションをパブリッシュするには、「現行のアプリケーションのパブリッシュ」チェックボックスを選択してから「OK」をクリックします。「アプリケーションをパブリッシュします。」ウィンドウが表示されます。

図 6-12 「アプリケーションをパブリッシュします。」ウィンドウ



「アプリケーションをパブリッシュします。」ウィンドウの各フィールドに必要な情報を入力します。

表 6-8 「アプリケーションをパブリッシュします。」ウィンドウのオプション

フィールド	説明	必須
Mobile サーバーの URL	サーバー名とポート番号を含む、Mobile サーバーの URL です。サーバー名とポート番号は、次の書式で指定します。 http://mobileserver/webtogo/:port。ポート 80 はデフォルト・ポートです。	YES

フィールド	説明	必須
Mobile サーバーのユーザー名	Mobile サーバー・ユーザーの名前です。	YES
Mobile サーバーのパスワード	Mobile サーバー・ユーザーのパスワードです。	YES
リポジトリのディレクトリ	Mobile サーバー上の Mobile サーバー・リポジトリの場所です。パッケージ・ウィザードはアプリケーション・ファイルをこのディレクトリにパブリッシュし、ローカル・アプリケーションのディレクトリ上でディレクトリ構造をメンテナンスします。	YES
アプリケーションをパブリックにする。	このアプリケーションをパブリック・アプリケーションとしてパブリッシュするには、これを選択します。パブリック・アプリケーションに対しては、すべてのユーザーがアクセス権を持ちます。	NO

注意： アプリケーションを Mobile サーバーにパブリッシュするには、publish 権限が必要です。Mobile サーバー管理者は Mobile サーバー・コントロール・センターを使用して権限を割り当てます。

アプリケーションの編集

パッケージ・ウィザードを起動して「既存のアプリケーションを編集」を選択すると、アプリケーションを編集できます。Mobile サーバーの DTD ファイルに準拠する XML 文書を作成または変更すると、アプリケーションを手動で作成または編集できますが、アプリケーションの作成または変更ではパッケージ・ウィザードの使用が最善の方法です。

Mobile サーバー・システム・カタログ・ビュー

この章は、Mobile Admin スキーマの一部である Mobile サーバー・システム・カタログ・ビューの参考資料です。Mobile Admin スキーマは Mobile サーバーの一部としてインストールされます。Mobile Admin スキーマは Mobile Development Kit の一部ではありません。

システム・カタログ・ビューは読取り専用なので、変更できません。ビューには次のものが含まれています。

- ["MOBILEADMIN".CV\\$ALL_CLIENTS](#)
- ["MOBILEADMIN".CV\\$ALL_PUBLICATIONS](#)
- ["MOBILEADMIN".CV\\$ALL_SUBSCRIPTIONS](#)
- ["MOBILEADMIN".CV\\$ALL_SEQUENCES](#)
- ["MOBILEADMIN".CV\\$ALL_SEQUENCE_PARTITIONS](#)
- ["MOBILEADMIN".CV\\$ALL_PUBLICATION_ITEMS_ADDED](#)
- ["MOBILEADMIN".CV\\$ALL_PUBLICATION_ITEMS](#)
- ["MOBILEADMIN".CV\\$ALL_PUBLICATION_INDEXES](#)
- ["MOBILEADMIN".CV\\$ALL_SUBSCRIPTION_PARAMS](#)

"MOBILEADMIN".CV\$ALL_CLIENTS

このビューには、Mobile サーバーのクライアントに関する情報が表示されます。列の内容は次のとおりです。

表 A-1 ALL_CLIENTS のパラメータ

列	データ型	NULL	説明
CLIENT	VARCHAR2(30)		Mobile サーバーのクライアント
LASTREFRESH_STARTTIME	VARCHAR2(19)		最終リフレッシュの開始時間
LASTREFRESH_ENDTIME	VARCHAR2(19)		最終リフレッシュの終了時間

"MOBILEADMIN".CV\$ALL_PUBLICATIONS

このビューには、Mobile サーバーのパブリケーションに関する情報が表示されます。列の内容は次のとおりです。

表 A-2 ALL_PUBLICATIONS のパラメータ

列	データ型	NULL	説明
NAME	VARCHAR2(30)		パブリケーション名
TYPE	VARCHAR2(40)		パブリケーション・タイプ
NAME_TEMPLATE	VARCHAR2(30)		スナップショット名テンプレート
ENFORCE_RI	CHAR(1)	NOT NULL	予約済

"MOBILEADMIN".CV\$ALL_SUBSCRIPTIONS

このビューには、Mobile サーバーのサブスクリプションに関する情報が表示されます。列の内容は次のとおりです。

表 A-3 ALL_SUBSCRIPTIONS のパラメータ

列	データ型	NULL	説明
CLIENT	VARCHAR2(30)		サブスクリプションのクライアント
PUBLICATION	VARCHAR2(30)		サブスクリプションのパブリケーション
INSTANTIATED	CHAR(1)		サブスクリプションがインスタンス化されているかどうかを示すブール値

"MOBILEADMIN".CV\$ALL_SEQUENCES

このビューには、Mobile サーバーのシーケンスに関する情報が表示されます。列の内容は次のとおりです。

表 A-4 ALL_SEQUENCES のパラメータ

列	データ型	NULL	説明
NAME	VARCHAR2(30)		シーケンス名

"MOBILEADMIN".CV\$ALL_SEQUENCE_PARTITIONS

このビューには、Mobile サーバーのシーケンス・パーティションに関する情報が表示されます。列の内容は次のとおりです。

表 A-5 ALL_SEQUENCE_PARTITIONS のパラメータ

列	データ型	NULL	説明
CLIENT	VARCHAR2(30)		シーケンスを割り当てるクライアント
NAME	VARCHAR2(30)		シーケンス名
CURR_VALUE	NUMBER(38)		現在のシーケンス値
INCREMENT_BY	NUMBER(38)		シーケンスの増分値。シーケンスはこの数値で増加します。

"MOBILEADMIN".CV\$ALL_PUBLICATION_ITEMS_ADDED

このビューには、パブリケーションに追加された Mobile サーバーのパブリケーション項目に関する情報が表示されます。列の内容は次のとおりです。

表 A-6 ALL_PUBLICATION_ITEMS_ADDED のパラメータ

列	データ型	NULL	説明
PUB_NAME	VARCHAR2(30)		パブリケーション名
ITEM_NAME	VARCHAR2(30)		パブリケーション項目名
OWNER	VARCHAR2(30)	NOT NULL	ベース・オブジェクトの所有者
OBJECT_NAME	VARCHAR2(30)	NOT NULL	ベース・オブジェクト名
TEXT	VARCHAR2(2048)	NOT NULL	Select 文
UPDATABLE	VARCHAR2(1)		更新可能オプション

列	データ型	NULL	説明
REFRESH_METHOD	CHAR(1)	NOT NULL	リフレッシュ方法。オプションには、高速リフレッシュと完全リフレッシュがあります。
WINNING_RULE	VARCHAR2(30)		レプリケーションの競合を解決するためのウィニング・ルール・オプション。オプションには、「クライアント優先」と「サーバー優先」があります。

"MOBILEADMIN".CV\$ALL_PUBLICATION_ITEMS

このビューには、Mobile サーバーのパブリケーション項目に関する情報が表示されます。
列の内容は次のとおりです。

表 A-7 ALL_PUBLICATION_ITEMS のパラメータ

列	データ型	NULL	説明
NAME	VARCHAR2(30)		パブリケーション項目名。
OWNER	VARCHAR2(30)	NOT NULL	パブリケーション項目のベース・オブジェクトの所有者。
OBJECT_NAME	VARCHAR2(30)	NOT NULL	ベース・オブジェクト名。
TEXT	VARCHAR2(2048)	NOT NULL	Select 文。
REFRESH_METHOD	CHAR(1)	NOT NULL	リフレッシュ方法。オプションには、高速リフレッシュと完全リフレッシュがあります。

"MOBILEADMIN".CV\$ALL_PUBLICATION_INDEXES

このビューには、Mobile サーバーのパブリケーション索引に関する情報が表示されます。
列の内容は次のとおりです。

表 A-8 ALL_PUBLICATION_INDEXES のパラメータ

列	データ型	NULL	説明
NAME	VARCHAR2(30)		索引名
PUB_ITEM	VARCHAR2(30)	NOT NULL	パブリケーション項目名
INDX_TYPE	CHAR(1)	NOT NULL	索引タイプ
COLUMN_LIST	VARCHAR2(2048)	NOT NULL	列リスト

"MOBILEADMIN".CV\$ALL_SUBSCRIPTION_PARAMS

このビューには、Mobile サーバーのサブスクリプション・パラメータに関する情報が表示されます。列の内容は次のとおりです。

表 A-9 ALL_SUBSCRIPTIONS_PARAMS のパラメータ

列	データ型	NULL	説明
NAME	VARCHAR2(30)		パブリケーション名
CLIENT	VARCHAR2(30)		クライアント
PARAM_NAME	VARCHAR2(30)		パラメータ名
PARAM_VALUE	VARCHAR2(30)		パラメータ値

SQL サポート

この章では、Oracle Lite for the Palm Computing Platform の SQL サポートについて説明します。内容は次のとおりです。

- [Oracle Lite for the Palm Computing Platform の SQL サポート](#)

Oracle Lite for the Palm Computing Platform の SQL サポート

この項では、Oracle Lite for the Palm Computing Platform の SQL サポートについて説明します。

サポートされる SQL 文

Oracle Lite for the Palm Computing Platform でサポートされる SQL 文を次にリストします。SQL 文は、大文字である必要があります。

表 B-1 サポートされる SQL 文

文	説明
CREATE TABLE	表を作成します。
DROP TABLE	表を削除します。
CREATE INDEX	表に索引を作成します。
DROP INDEX	索引を削除します。
DELETE	表から 1 つまたは複数の行を削除します。
INSERT	表に行を挿入します。
SELECT	一連の行を検索して取得します。
UPDATE	1 つまたは複数の行を更新します。
COMMIT	変更内容をコミットします。
ROLLBACK	変更内容をロールバックします。

CREATE TABLE

データベースに表を作成します。

図 B-1 CREATE TABLE 文の構文

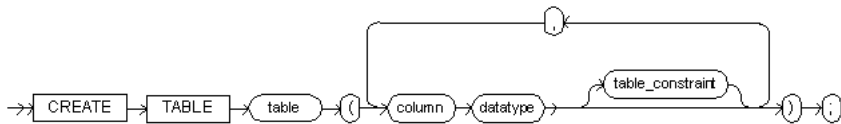


表 B-2 CREATE TABLE の引数

引数	説明
<i>table</i>	作成する表の名前。
<i>column</i>	列の名前。
<i>datatype</i>	列のデータ型。サポートされるデータ型のリストは、「 サポートされるデータ型 」を参照してください。
<i>table_constraint</i>	オプションの制約。PRIMARY KEY または NOT NULL。

DROP TABLE

DROP TABLE 文は、データベースから表を削除します。

図 B-2 DROP TABLE 文の構文



表 B-3 DROP TABLE の引数

引数	説明
<i>table</i>	削除する表の名前。

CREATE INDEX

CREATE INDEX 文は、表に索引を作成します。

図 B-3 CREATE INDEX 文の構文

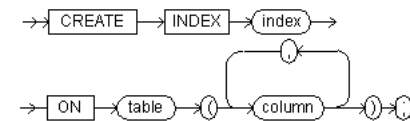


表 B-4 CREATE INDEX の引数

引数	説明
<i>index</i>	索引の名前。データベースで一意である必要があります。
<i>table</i>	索引を作成する表の名前。
<i>column</i>	索引を構成するために使用する列の名前。

DELETE

DELETE 文は、WHERE 句で設定した条件に一致する行をすべて削除します。WHERE 句を指定しない場合、表のすべての行が削除されます。

図 B-4 DELETE 文の構文

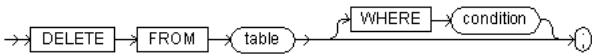


表 B-5 DELETE の引数

引数	説明
<i>table</i>	表の名前。
<i>condition</i>	オプションの検索条件。条件に一致する行のみが削除されます。WHERE 句を指定しない場合、表のすべての行が削除されます。

DROP INDEX

DROP INDEX 文は、索引を削除します。

図 B-5 DROP INDEX 文の構文

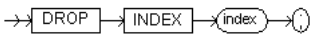


表 B-6 DROP INDEX の引数

引数	説明
<i>index</i>	削除する索引の名前。

INSERT

INSERT 文は、表に新規行を挿入します。列リストを指定しない場合、データは表に列が定義されている順に更新されます。

図 B-6 INSERT 文の構文

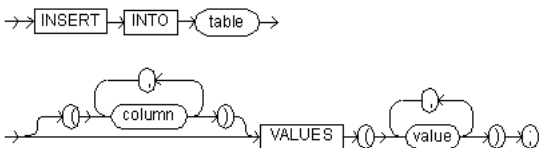


表 B-7 INSERT の引数

引数	説明
<i>table</i>	行を挿入する表の名前。
<i>column</i>	列の名前。
<i>value</i>	列に格納するリテラル値。詳細は、「 リテラル値 」を参照してください。

SELECT

SELECT 文は、1 つまたは複数の表から行を問い合わせます。

図 B-7 SELECT 文の構文

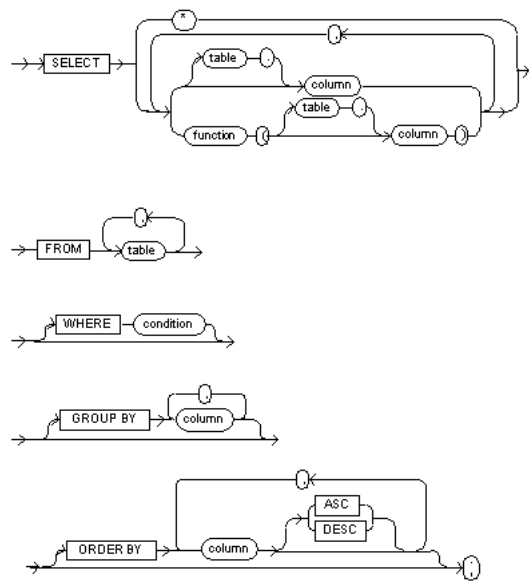


表 B-8 SELECT の引数

引数	説明
<i>table</i>	表の名前。
<i>column</i>	列の名前。
<i>function</i>	列関数。サポートされる列関数のリストは、 「サポートされるグループ関数」 を参照してください。
<i>condition</i>	選択条件。条件に一致する行のみが選択されます。指定しない場合、表のすべての行が選択されます。詳細は、 「条件」 を参照してください。
ORDER BY	SELECT 文によって戻された行を、指定された <i>column</i> に従って順序付けします。

UPDATE

UPDATE 文は、指定した条件に一致する行をすべて更新します。

図 B-8 UPDATE 文の構文

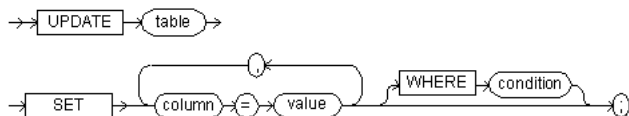


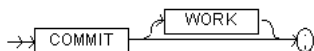
表 B-9 UPDATE の引数

引数	説明
<i>table</i>	表の名前。
<i>column</i>	列の名前。
<i>value</i>	列に格納するリテラル値。詳細は、「 リテラル値 」を参照してください。
<i>condition</i>	検索条件。条件に一致する行のみが更新されます。WHERE 句を指定しない場合、表のすべての行が更新されます。詳細は、「 条件 」を参照してください。

COMMIT

COMMIT 文は、現在のトランザクションで行われた変更をコミットします。

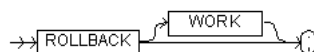
図 B-9 COMMIT 文の構文



ROLLBACK

ROLLBACK 文は、現在のトランザクションで行われた変更をロールバックします。

図 B-10 ROLLBACK 文の構文



条件

WHERE 句は、行の結果リストを選択する条件を定義します。WHERE 句は、SELECT、UPDATE および DELETE の各コマンドとともに使用できます。

図 B-11 WHERE 句の構文

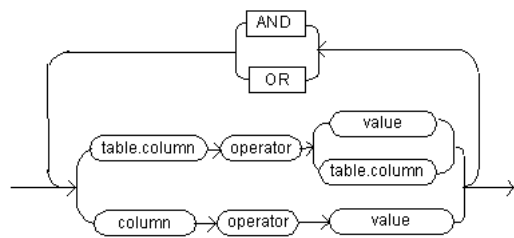


表 B-10 WHERE 句のパラメータ

要素	説明
column	列の名前。
operator	比較演算子。「 比較演算子 」を参照してください。
value	リテラル値。詳細は、「 リテラル値 」を参照してください。

比較文にはそれぞれ、左オペランド、比較演算子および右オペランドがあります。

左オペランドは、有効な列名である必要があります。SELECT 文の場合、列名は *table.column* 形式で指定できます。*table* は表の名前です。表名を指定しない場合、FROM 句で指定した最初の表が使用されます。

比較演算子

比較演算子は、左オペランドと右オペランドを比較する方法を定義します。次の表にサポートされる演算子をリストします。

表 B-11 比較演算子

演算子	説明
=	等しい。
<>, !=, ^=	等しくない。

演算子	説明
<	より小さい。
>	より大きい。
<=	以下。
>=	以上。
LIKE	右オペランドに指定した文字から開始します。「%」記号は、文字列の最後に置かれる場合にのみサポートされます。
NOT LIKE	指定した文字からは開始しません。
IS NULL	NULL 値。
IS NOT NULL	NULL 値ではない。

右オペランドは、リテラル値または列名のいずれかにできます。UPDATE 文と DELETE 文の場合、右オペランドはリテラル値である必要があります。文が有効になるためには、2 つのオペランドが同じ型である必要があります。

比較文は AND 演算子や OR 演算子で連鎖できます。AND 演算子は、OR 演算子に優先します。カッコを使用したグループ化は現在サポートしていません。

たとえば、EmpID 列 (INTEGER 型) が 158 に等しい列を持つ行を検索するには、次のようにします。

```
WHERE EmpID = 158
```

EmpID 列が 50 より大きく 100 未満でかつ 90 でない行、または Name 列 (VARCHAR 型) が 'Sam' で始まる行を検索するには、次のようにします。

```
WHERE EmpID > 50 AND EmpID < 100 AND EmpID <> 90 OR Name LIKE 'Sam%'
```

EmpID 列が Dept 表の行と一致し、Name 列が NULL でない Emp 表の行をすべて検索するには、次のようにします。

```
WHERE Emp.EmpID = Dept.EmpID AND Emp.Name IS NOT NULL
```

リテラル値

リテラル値という語は、固定データ値を意味します。たとえば、'Jack'、'Sam' および '101' はすべて文字リテラル値です。リテラル数値は、-2147483647 から 2147483647 に制限されています。より大きい数が必要な場合は、数値に小数点とゼロ (.0) を付加して、パーサーに強制的に実数として表示させます。たとえば、1234567890 のかわりに 1234567890.0 を使用します。

表 B-12 リテラル値パラメータ

リテラル	説明
テキスト	<p>CHAR、VARCHAR および VARCHAR2 のプロパティを持つリテラル文字列。テキスト・リテラル値は、引用符によってカプセル化されます。</p> <p>例: 'Jack' および 'How are you today?'</p> <p>テキスト・リテラルの内部に引用符を指定するには、内部の引用符の前にもう 1 つ引用符を付けます。</p> <p>例: 'My cat''s name'</p> <p>空の文字列（サイズがゼロ）を指定するには、引用符を 2 個 ("") 指定します。</p>
整数	<p>小数点を含まない数値。整数は数字の配列として指定され、オプションで前に符号文字 (+ または -) が付きます。たとえば、次のようになります。123、256789 および -125。整数リテラル値の型は、INTEGER または CHAR（単一文字）です。</p>
数値	<p>数値リテラル値は、小数点付きの実数です。数値リテラル値は、オプションの符号文字 (+ または -) で始まり、1 個以上の数字、1 つの小数点 (.) および 1 個以上の数字で構成される必要があります。たとえば、次のようになります。12.345、55.0、-1234.5567 および +123.9。数値リテラル値は、NUMBER 型、DECIMAL 型または NUMERIC 型として処理されます。小数点がない場合、値は整数として処理されます。</p>
日付、時間、タイムスタンプ	<p>日付と時間は、文字列を次のいずれかの書式で指定する必要があることを除いて、リテラル・テキストと同様に指定されます。</p> <ul style="list-style-type: none">■ 'yyyy-mm-dd'（日付を指定）■ 'hh:mm:ss'（時間を指定）■ 'yyyy-mm-dd hh:mm:ss'（日付と時間を指定） <p>たとえば、次のようになります。</p> <ul style="list-style-type: none">■ '1970-07-15'■ '1999-12-30 12:50:00'■ '12:59:31'

サポートされるデータ型

Oracle Lite for the Palm Computing Platform では、次のデータ型がサポートされます。

表 B-13 サポートされるデータ型

データ型	説明
BINARY	最高 4,096 バイトのバイナリ・データの格納が可能です。
BLOB	バイナリ・ラージ・オブジェクト (Binary Large Object: BLOB)。精度 (格納できる桁数またはビット数) を指定する必要があります。最大サイズは 64KB です。
CHAR	固定長文字データ。最大サイズは 10KB です。デフォルトおよび最小サイズは 1 バイトです。
DATE	BC4712 年の 1 月 1 日から AD4712 年の 12 月 31 日までの範囲の有効な日付。書式は 'yyyy-mm-dd' です。
DECIMAL	精度 (10 進数値) またはスケール (小数値) で測定可能な数値。精度は DECIMAL (<i>p</i>) を使用して計算できます。スケールは DECIMAL (<i>p, s</i>) を使用して計算できます。スケールは精度より大きくできません。
DOUBLE	仮数の小数精度が 15 の符号付き概数値。絶対値は 0 または 10^{-308} と 10^{308} の間です。
FLOAT	浮動小数点数。
INTEGER および INT	オペレーティング・システムによって精度が定義される整数値。値の範囲は、-214783647 ~ 214783647 です。
NUMBER および NUMERIC	精度 (10 進数値) またはスケール (小数値) で測定可能な数値。精度は NUMERIC (<i>p</i>) を使用して計算できます。スケールは NUMERIC (<i>p, s</i>) を使用して計算できます。精度の最大値は 38 です。スケールは精度より大きくできません。
REAL	オプションなしの単精度浮動小数点を要求できます。精度は実装によって選択され、通常はハードウェア・プラットフォームのデフォルトの単精度データ型となります。
SMALLINT	精度が実装で定義される小さい整数です。値の範囲は、-32768232767 ~ 32768232767 です。
TIME	時間値を時、分および秒で格納します。時は 00 ~ 23 の範囲の 2 つの数字で表され、分は 00 ~ 59 の範囲の 2 つの数字で表されます。秒の範囲は 00 ~ 59 です。書式は 'hh:mm:ss' です。
TIMESTAMP	日付と時刻を格納します。書式は 'yyyy-mm-dd hh:mm:ss' です。

データ型	説明
VARCHAR	最大サイズ 10KB の可変長文字列。サイズを指定する必要があります。精度付きの VARCHAR データ型は、可変長文字列として定義されます。
VARCHAR2	最大サイズ 10KB の可変長文字列。サイズを指定する必要があります。

サポートされるグループ関数

グループ関数は、行のグループを基にした結果を戻します。グループ関数は、SELECT 文の列リストで使用できます。Oracle Lite for the Palm Computing Platform では、次のグループ関数がサポートされます。

表 B-14 サポートされるグループ関数

関数	説明
AVG(<i>column</i>)	行のグループの <i>column</i> の平均値。
COUNT(<i>column</i>)	問合せ内の行数。NOT NULL 値を指定された <i>column</i> を持つ行のみがカウントされます。
COUNT(*)	表内のすべての行数。行内の列が NULL かどうかにかかわらず。
MAX(<i>column</i>)	行のグループの <i>column</i> の最大値。
MIN(<i>column</i>)	行のグループの <i>column</i> の最小値。
SUM(<i>column</i>)	行のグループの <i>column</i> の合計。

注意： AVG と SUM は、INTEGER (INT) データ型および SMALLINT データ型でのみサポートされます。

SELECT 文でグループ関数を使用する場合は、GROUP BY 句を指定する必要があります。

サポートされる日付関数

日付関数は日付を戻します。Oracle Lite for the Palm Computing Platform では、次の日付関数がサポートされます。

表 B-15 サポートされる日付関数

関数	説明
CURRENT_DATE	現在の日付を戻します。
CURRENT_TIME	現在の時刻を戻します。
SYSDATE	現在の日付と時刻を戻します。

ODBC サポート

この章では、Oracle Lite for the Palm Computing Platform に対する Open Database Connectivity (ODBC) サポートについて説明します。内容は次のとおりです。

- [Oracle Lite for the Palm Computing Platform の ODBC サポート](#)

Oracle Lite for the Palm Computing Platform の ODBC サポート

Oracle Lite for the Palm Computing Platform では、ODBC 3.0 Application Programming Interface (API) 標準のサブセットをサポートしています。ODBC API を使用すると、アプリケーションは携帯端末の Oracle Lite データベースに格納されたデータにアクセスできます。

Oracle Lite ODBC ライブラリは動的 SQL モデルをサポートしています。このモデルでは、アプリケーションが実行時に SQL 文を構成して携帯端末上で直接実行できます。

サポートされる全 ODBC API 関数を次にリストし、説明します。

表 C-1 ODBC API 関数

関数	説明
SQLAllocConnect	指定した環境を使用する接続ハンドルにメモリーを割り当てます。
SQLAllocEnv	環境ハンドルにメモリーを割り当てます。
SQLAllocHandle	環境ハンドル、接続ハンドルおよび文ハンドルを割り当てるための汎用関数です。
SQLAllocStmnt	指定した接続を使用する文ハンドルにメモリーを割り当てます。
SQLFreeConnect	指定したハンドルを使用する接続中のデータベースを切断し、ハンドルを解放します。
SQLFreeEnv	指定したハンドルを解放します。ハンドルに関連するコミットされていないトランザクションはロールバックされます。
SQLFreeHandle	環境ハンドル、接続ハンドルおよび文ハンドルを解放するための汎用関数です。
SQLFreeStmnt	指定した文ハンドルとそれに関連する一時メモリーを解放します。
SQLConnect	データベースに接続し、指定した接続ハンドルに接続に関する情報を保存します。
SQLDisconnect	以前接続していたデータベースを切断し、クローズします。
SQLBindParameter	データ・バッファを SQL 文のパラメータ・マーカーにバインドします。
SQLPrepare	SQL 文をコンパイルし、指定した文ハンドルにその情報を格納します。
SQLExecDirect	指定した SQL 文をコンパイルし、実行します。
SQLExecute	SQLPrepare を使用してプリコンパイルされた SQL を実行します。
SQLFetch	結果セットからデータ行を読み取ります。この関数をコールした後、カーソルは次に読み取る行に置かれます。

関数	説明
SQLBindCol	バッファを結果セットの列にバインドします。
SQLDescribeCol	結果セットの列に関する情報を取得します。
SQLError	指定したハンドルに関連する最新のエラーの詳細を取り出します。
SQLGetData	現在行から指定したバッファに単一行を読み取ります。
SQLNumResultCols	結果セット内の列数を戻します。
SQLRowCount	SQL SELECT 文、UPDATE 文または DELETE 文で影響を受けた行数を戻します。
SQLTransact	環境に対応付けられたすべての文のすべてのアクティブな操作に対して、コミットまたはロールバックを要求します。

SQLAllocConnect

指定した環境 hEnv を使用する接続ハンドルにメモリを割り当てます。

構文

```
RETCODE SQLAllocConnect( hEnv, hDbc )
```

引数

表 C-2 SQLAllocConnect の引数

型	名前	説明
SQLHENV	hEnv	環境ハンドル。NULL に設定した場合、新しい環境を作成します。
SQLHDBC*	hDbc	接続ハンドルへのポインタ。ルーチンは新規に割り当てられたメモリのアドレスをこのハンドルに格納します。

使用方法

この関数は、ODBC 2.0 との下位互換性のためにサポートされています。新規アプリケーションは、SQLAllocHandle 関数およびハンドル・タイプ SQL_HANDLE_DBC を使用してコード化する必要があります。SQLAllocConnect は内部的に SQLAllocHandle をコールします。

戻り値

成功した場合、SQLAllocConnect は SQL_SUCCESS を返します。それ以外の場合は、SQL_ERROR を返します。エラーの詳細は、アプリケーションから指定した環境ハンドルを使用して SQLError をコールします。

SQLAllocEnv

SQLAllocEnv は環境ハンドルにメモリーを割り当てます。

複数の接続や文ハンドルに対して単一のトランザクションを共有するには、SQLAllocConnection、SQLAllocStmt または SQLAllocHandle に同じ環境ハンドルを渡します。この方法により、新規ハンドルは同じ環境ハンドルを継承して共有できます。これらのハンドルを解放しても、実際の接続やトランザクションは解放されません。リソースは、元の環境ハンドルが解放されるまで解放されません。

構文

```
RETCODE SQLAllocEnv( hEnv )
```

引数

表 C-3 SQLAllocEnv の引数

型	名前	説明
SQLHENV*	hEnv	環境ハンドルへのポインタ

使用方法

この関数は、ODBC 2.0 との下位互換性のためにサポートされています。新規アプリケーションは、SQLAllocHandle 関数およびハンドル・タイプ SQL_HANDLE_ENV を使用してコード化する必要があります。SQLAllocEnv は内部的に SQLAllocHandle をコールします。

戻り値

成功した場合、SQLAllocEnv は SQL_SUCCESS を返します。それ以外の場合は、SQL_ERROR を返します。エラーの詳細は、アプリケーションから SQLError をコールして、ハンドル・パラメータとして NULL を渡します。

SQLAllocHandle

SQLAllocHandle は、環境ハンドル、接続ハンドルおよび文ハンドルを割り当てるための汎用関数です。

この関数は、個々のハンドル・タイプに対する古い割当て関数（SQLAllocEnv、SQLAllocConnection および SQLAllocStmt）を置き換えます。

新規環境ハンドルごとにトランザクション表（新規 OKAPI 環境）が作成されます。複数の接続や文ハンドルに対して単一のトランザクションを共有するには、同じ環境ハンドルを SQLAllocHandle の inputHandle 引数として渡します。この方法により、新規ハンドルは同じ環境ハンドルを継承して共有できます。これらのハンドルを解放しても、実際の接続やトランザクションは解放されません。リソースは、元の環境ハンドルが解放されるまで解放されません。同じ方法を使用して、接続も共有できます。

構文

```
RETCODE SQLAllocHandle( handleType, inputHandle, outputHandle )
```

引数

表 C-4 SQLAllocHandle の引数

型	説明	名前
SQLSMALLINT	割り当てるハンドルのタイプ。詳細は、次の「使用方法」を参照してください。	handleType
SQLHANDLE	新規ハンドルの基になるハンドル。環境ハンドルまたは接続ハンドルのいずれかです。 新規ハンドルを初めから作成するには、NULL を渡します。	inputHandle
SQLHANDLE*	新規に作成されたハンドル用の記憶域へのポインタ。	outputHandle

使用方法

アプリケーションは、使用する API 関数によって異なる種類のハンドルを割り当てます。ハンドルは各関数に対するコンテキストを提供します。次のハンドルがサポートされています。

表 C-5 ハンドル・パラメータ

ハンドル	型	説明
環境	SQL_TYPE_ENV	環境ハンドルは、環境を作成するために使用します。各環境には、データベースにアクセスするための汎用情報が含まれています。新規トランザクションは、新規に作成された環境ハンドルに対応付けられます。
接続	SQL_TYPE_DBC	接続ハンドルは、特定の Oracle Lite データベースへの接続をオープンするために使用します。同一の環境ハンドルに基づいて複数の接続ができるため、複数のデータベース接続で同一のトランザクションが共有されます。ただし、単一の環境ハンドルを共有できる最大接続数は 8 個です。

ハンドル	型	説明
文	SQL_TYPE_STMT	文ハンドルには、コンパイル済み SQL 文とその結果についての情報が格納されます。

戻り値

成功した場合、SQLAllocHandle は SQL_SUCCESS を戻します。それ以外の場合は、SQL_ERROR を戻します。エラーの詳細は、アプリケーションから inputHandle 引数を指定して SQLError をコールします。

SQLAllocStmt

SQLAllocStmt は、指定した接続 hDbc を使用する文ハンドルにメモリーを割り当てます。

構文

RETCODE SQLAllocStmt (hDbc, hStmt)

引数

表 C-6 SQLAllocStm の引数

型	名前	説明
SQLHDBC	hDbc	新規ハンドルを作成する接続ハンドル
SQLHSTMT*	hStmt	文ハンドルへのポインタ

使用方法

この関数は、ODBC 2.0 との下位互換性のためにサポートされています。新規アプリケーションは、SQLAllocHandle 関数およびハンドル・タイプ SQL_HANDLE_STMT を使用してコード化する必要があります。SQLAllocStmt は内部的に SQLAllocHandle をコールします。

戻り値

成功した場合、SQLAllocStmt は SQL_SUCCESS を戻します。それ以外の場合は、SQL_ERROR を戻します。エラーの詳細は、アプリケーションから指定した接続ハンドルを使用して SQLError をコールします。

SQLFreeConnect

SQLFreeConnect は、指定したハンドルを使用する接続中のデータベースから切断し、ハンドルを解放します。

構文

```
RETCODE SQLFreeConnect (hDbc )
```

引数

表 C-7 SQLFreeConnect の引数

型	名前	説明
SQLHDBC	hDbc	解放する接続ハンドル

使用方法

この関数は廃止され、新規汎用関数 SQLFreeHandle に置き換わりました。

戻り値

成功した場合、SQLFreeConnect は SQL_SUCCESS を返します。それ以外の場合は、SQL_ERROR を返します。エラーの詳細は、アプリケーションから指定した環境ハンドルを使用して SQLError をコールします。

SQLFreeEnv

SQLFreeEnv は指定したハンドルを解放します。ハンドルに関連するコミットされていないトランザクションはロールバックされます。

構文

```
RETCODE SQLFreeEnv( hEnv )
```

引数

表 C-8 SQLFreeEnv の引数

型	名前	説明
SQLHENV	hEnv	解放する環境ハンドル

注意

この関数は廃止され、新規汎用関数 SQLFreeHandle に置き換わりました。

戻り値

成功した場合、SQLFreeEnv は SQL_SUCCESS を戻します。それ以外の場合は、SQL_ERROR を戻します。エラーの詳細は、アプリケーションから指定した環境ハンドルを使用して SQLError をコールします。

SQLFreeHandle

SQLFreeHandle は、環境ハンドル、接続ハンドルおよび文ハンドルを解放するための汎用関数です。

ハンドルには最後に使用した方法に関する情報、つまり解放の方法が内部的に格納されているため、引数 handleType は使用されません。

構文

```
RETCODE SQLFreeHandle( handleType, handle )
```

引数

表 C-9 SQLFreeHandle の引数

型	名前	説明
SQLSMALLINT	handleType	解放するハンドルのタイプ
SQLHANDLE	handle	解放するハンドル

戻り値

成功した場合、SQLFreeHandle は SQL_SUCCESS を戻します。それ以外の場合は、SQL_ERROR を戻します。エラーの詳細は、アプリケーションから指定したハンドルを使用して SQLError をコールします。

SQLFreeStmt

SQLFreeStmt は、指定した文ハンドルとそれに関連する一時メモリーを解放します。

構文

```
RETCODE SQLFreeStmt( hStmt, Option )
```

引数

表 C-10 SQLFreeStmt の引数

型	名前	説明
SQLHSTMT	hStmt	解放する文ハンドル。
SQLUSMALLINT	オプション	この引数は現在サポートされていないため無視されます。

使用方法

この関数は廃止され、新規汎用関数 `SQLFreeHandle` に置き換わりました。

戻り値

成功した場合、`SQLFreeStmt` は `SQL_SUCCESS` を返します。それ以外の場合は、`SQL_ERROR` を返します。エラーの詳細は、アプリケーションから指定した環境ハンドルを使用して `SQLError` をコールします。

SQLConnect

`SQLConnect` は、データベースに接続し、指定した接続ハンドルに接続に関する情報を保存します。ハンドルは、`SQLAllocHandle` 関数を使用して事前に割り当てる必要があります。

構文

```
RETCODE SQLConnect( hConn, dbName, dbNameLen, userName, userNameLen, auth, authLen )
```

引数

表 C-11 SQLConnect の引数

型	名前	説明
SQLHDBC	hConn	新規に割り当てた接続ハンドル。渡された接続ハンドルが使用中の場合、関数は既存の接続をクローズします。
SQLCHAR*	dbName	接続先のデータベース名。
SQLSMALLINT	dbNameLen	データベース名の長さ。
SQLCHAR*	userName	この引数は現在サポートされていないため無視されます。
SQLSMALLINT	userNameLen	この引数は現在サポートされていないため無視されます。

型	名前	説明
SQLCHAR*	auth	この引数は現在サポートされていないため無視されます。
SQLSMALLINT	authLen	この引数は現在サポートされていないため無視されます。

戻り値

成功した場合、SQLConnect は SQL_SUCCESS を返します。それ以外の場合は、SQL_ERROR を返します。エラーの詳細は、アプリケーションから指定した接続ハンドルを使用して SQLError をコールします。

SQLDisconnect

SQLDisconnect は以前接続していたデータベースを切断し、クローズします。

接続するために使用した環境が、接続をクローズする前にコミットされていない場合、以降のコミットは失敗します。

構文

RETCODE SQLDisconnect(hDbc)

引数

表 C-12 SQLDisconnect の引数

型	名前	説明
SQLHDBC	hDbc	切断する接続ハンドル

戻り値

成功した場合、SQLDisconnect は SQL_SUCCESS を返します。それ以外の場合は、SQL_ERROR を返します。エラーの詳細は、アプリケーションから指定した接続ハンドルを使用して SQLError をコールします。

SQLBindParameter

SQLBindParameter は、データ・バッファを SQL 文のパラメータ・マーカーにバインドします。パラメータ・マーカーは、SQL 文に「?」で示されます。

構文

```
RETCODE SQLBindParameter( hStmt, paramNo, paramType, cType, sqlType, colDef, scale,
value, valueMaxSize, valueSize )
```

引数

表 C-13 SQLBindParameter の引数

型	名前	説明
SQLHSTMT	hStmt	文ハンドル。
SQLUSMALLINT	paramNo	バインド先のパラメータ・マーカー番号。1 で始まり、左から右にカウントされます。
SQLSMALLINT	paramType	パラメータ・タイプ。現在、SQL_PARAM_INPUT のみがサポートされます。
SQLSMALLINT	cType	パラメータの C 言語データ型。
SQLSMALLINT	sqlType	パラメータの SQL データ型。
SQLINTEGER	colDef	パラメータの精度。
SQLSMALLINT	scale	パラメータのスケール。
SQLPOINTER	value	パラメータを格納するバッファへのポインタ。
SQLINTEGER	valueMaxSize	パラメータ・バッファのサイズ。
SQLINTEGER*	valueSize	パラメータ値の実際のサイズ。

戻り値

成功した場合、SQLBindParameter は SQL_SUCCESS を戻します。それ以外の場合は、SQL_ERROR を戻します。

SQLPrepare

SQLPrepare は SQL 文をコンパイルし、指定した文ハンドルにその情報を格納します。

構文

```
RETCODE SQLPrepare( hStmt, statement, statementLen )
```

引数

表 C-14 SQLPrepare の引数

型	名前	説明
SQLHSTMT	hStmt	文ハンドル
SQLCHAR*	statement	SQL 文の文字列
SQLINTEGER	statementLen	SQL 文の文字列の長さ

戻り値

成功した場合、SQLPrepare は SQL_SUCCESS を戻します。それ以外の場合は、SQL_ERROR を戻します。エラーの詳細は、アプリケーションから指定した文ハンドルを使用して SQLError をコールします。

SQLExecDirect

SQLExecDirect は指定した SQL 文をコンパイルし、実行します。

構文

```
RETCODE SQLExecDirect( hStmt, statement, statementLen )
```

引数

表 C-15 SQLExecDirect の引数

型	名前	説明
SQLHSTMT	hStmt	文ハンドル
SQLCHAR*	statement	SQL 文の文字列
SQLINTEGER	statementLen	SQL 文の文字列の長さ

戻り値

成功した場合、SQLExecDirect は SQL_SUCCESS を戻します。それ以外の場合は、SQL_ERROR を戻します。エラーの詳細は、アプリケーションから指定した文ハンドルを使用して SQLError をコールします。

SQLExecute

SQLExecute は SQLPrepare を使用してプリコンパイルされた SQL を実行します。

構文

```
RETCODE SQLExecute( hStmt )
```

引数

表 C-16 SQLExecute の引数

型	名前	説明
SQLHSTMT	hStmt	文ハンドル

戻り値

成功した場合、SQLExecute は SQL_SUCCESS を戻します。それ以外の場合は、SQL_ERROR を戻します。エラーの詳細は、アプリケーションから指定した文ハンドルを使用して SQLError をコールします。

SQLFetch

SQLFetch は結果セットからデータ行を読み取ります。この関数をコールした後、カーソルは次に読み取る行に置かれます。

アプリケーションは SQLGetData をコールして、読み取られた行の列を読み取れます。

アプリケーションが SQLBindCol をコールして列をバインドした場合、SQLFetch は指定されたバッファに行のデータを格納します。

構文

```
RETCODE SQLFetch( hStmt )
```

引数

表 C-17 SQLFetch の引数

型	名前	説明
SQLHSTMT	hStmt	文ハンドル

戻り値

新規データ行が正常に読み取られた場合、SQLFetch は SQL_SUCCESS を返します。

読み取る行がない場合、SQLFetch は SQL_NO_DATA_FOUND を返します。

エラーが発生した場合、関数は SQL_ERROR を返します。エラーの詳細は、アプリケーションから指定した文ハンドルを使用して SQLError をコールします。

SQLBindCol

SQLBindCol はバッファを結果セットの列にバインドします。バッファは、SQLFetch がコールされたときに更新されます。その後、結果セットの新規列が読み取られます。

SQLBindCol は、SQLFetch をコールする前であれば、文をプリコンパイルして実行する前でも後でもコールできます。

構文

RETCODE SQLBindCol(hStmt, columnNo, targetType, targetValue, targetSize, actualSize)

引数

表 C-18 SQLBindCol の引数

型	名前	説明
SQLHSTMT	hStmt	文ハンドル。
SQLUSMALLINT	columnNo	バインド先の結果セットの列番号。
SQLSMALLINT	targetType	バッファの C 言語データ型。
SQLPOINTER	targetValue	列データを保持するバッファへのポインタ。
SQLINTEGER	targetSize	バッファのサイズ (バイト単位)。
SQLINTEGER*	actualSize	読み取ったデータのサイズを保持するバッファへのポインタ。この情報が必要ない場合は、NULL を指定できます。

戻り値

成功した場合、SQLBindCol は SQL_SUCCESS を返します。それ以外の場合は、SQL_ERROR を返します。エラーの詳細は、アプリケーションから指定した文ハンドルを使用して SQLError をコールします。

SQLDescribeCol

SQLDescribeCol は結果セットの列に関する情報を取得します。

構文

```
RETCODE SQLDescribeCol( hStmt, columnNo, columnName, columnNameMaxLen,
datatype, columnNameLen, columnSize, decimalDigits, nullable )
```

引数

表 C-19 SQLDescribeCol の引数

型	名前	説明
SQLHSTMT	hStmt	文ハンドル。
SQLUSMALLINT	columnNo	結果セット内の列番号。
SQLCHAR*	columnName	戻された列名を格納する文字列バッファへのポインタ。
SQLSMALLINT	columnNameMaxLen	文字列バッファのサイズ。
SQLSMALLINT	*columnNameLen	戻された列名サイズ (バイト単位)。
SQLSMALLINT*	dataType	戻された SQL データ型。
SQLINTEGER*	columnSize	戻された列サイズ。
SQLSMALLINT*	decimalDigits	戻された列精度。
SQLSMALLINT*	nullable	列が NULL 化可能な場合は 1、可能でない場合は 0 に設定されます。

戻り値

成功した場合、SQLDescribeCol は SQL_SUCCESS を戻します。それ以外の場合は、SQL_ERROR を戻します。エラーの詳細は、アプリケーションから指定した文ハンドルを使用して SQLError をコールします。

SQLError

SQLError は指定したハンドルに関連する最新のエラーの詳細を抽出します。

構文

```
RETCODE SQLError( hEnv, hConn, hStmt, sqlState, nativeError, messageText,
messageMaxSize, messageLength)
```

引数

表 C-20 SQLError の引数

型	名前	説明
SQLHENV	hEnv	環境ハンドル
SQLHDBC	hConn	データベース・ハンドル
SQLHSTMT	hStmt	文ハンドル
SQLCHAR*	sqlState	戻された SQLSTATE を格納する文字列バッファへのポインタ
SQLINTEGER*	nativeError	システム固有のエラー・コード
SQLCHAR*	messageText	エラー・メッセージのテキスト
SQLSMALLINT	messageMaxSize	渡すバッファのサイズ
SQLSMALLINT*	messageLen	戻されたメッセージ・テキストの長さ

戻り値

最新のエラーに関連する情報を取得できた場合、SQLError は SQL_SUCCESS を戻します。指定したハンドルに関連するエラーがなかった場合、関数は SQL_NO_DATA_FOUND を戻します。

SQLGetData

SQLGetData は現在行から指定したバッファに単一列を読み取ります。このルーチンでは、データは宛先バッファの型に変換されます。

構文

```
RETCODE SQLGetData( hStmt, columnNo, targetType, targetValue, targetSize, actualSize )
```

引数

表 C-21 SQLGetData の引数

型	名前	説明
SQLHSTMT	hStmt	文ハンドル
SQLUSMALLINT	columnNo	列数
SQLSMALLINT	targetType	targetValue バッファのタイプ
SQLPOINTER	targetValue	結果の列データを格納するバッファへのポインタ

型	名前	説明
SQLINTEGER	targetSize	バッファのサイズ
SQLINTEGER*	actualSize	指定したバッファに読み取られた実際のバイト数

戻り値

成功した場合、SQLGetData は **SQL_SUCCESS** を返します。それ以外の場合は、**SQL_ERROR** を返します。エラーの詳細は、アプリケーションから指定した文ハンドルを使用して **SQLError** をコールします。

SQLNumResultCols

SQLNumResultCols は結果セット内の列数を返します。

構文

```
RETCODE SQLNumResultCols( hStmt, columnCount )
```

引数**表 C-22 SQLNumResultCols の引数**

型	名前	説明
SQLHSTMT	hStmt	文ハンドル
SQLSMALLINT*	columnCount	結果セット内に戻された列数を格納するバッファへのポインタ

戻り値

成功した場合、SQLNumResultCols は **SQL_SUCCESS** を返します。それ以外の場合は、**SQL_ERROR** を返します。エラーの詳細は、アプリケーションから指定した文ハンドルを使用して **SQLError** をコールします。

SQLRowCount

SQLRowCount は、SQL SELECT 文、UPDATE 文または DELETE 文で影響された行数を返します。

構文

```
RETCODE SQLRowCount( hStmt, rowCount )
```

引数

表 C-23 SQLRowCount の引数

型	名前	説明
SQLHSTMT	hStmt	文ハンドル
SQLINTEGER*	rowCount	結果セット内の行数

戻り値

成功した場合、SQLRowCount は SQL_SUCCESS を戻します。それ以外の場合は、SQL_ERROR を戻します。エラーの詳細は、アプリケーションから指定した文ハンドルを使用して SQLError をコールします。

SQLTransact

SQLTransact は、環境に対応付けられたすべての文に関するすべてのアクティブな操作に対して、コミットまたはロールバックを要求します。

構文

RETCODE SQLTransact(hEnv, hDbc, completionType)

引数

表 C-24 SQLTransact の引数

型	名前	説明
SQLHENV	hEnv	環境ハンドル。
SQLHDBC	hDbc	接続ハンドル。使用しません。
SQLUSMALLINT	completionType	トランザクション・アクション。SQL_COMMIT または SQL_ROLLBACK のいずれかにできます。

戻り値

成功した場合、SQLTransact は SQL_SUCCESS を戻します。それ以外の場合は、SQL_ERROR を戻します。エラーの詳細は、アプリケーションから指定した環境ハンドルを使用して SQLError をコールします。

この章では、Oracle Lite Mobile SQL について説明します。この章の各項では、それぞれ異なるトピックを説明します。内容は次のとおりです。

- 概要
- インストールとセットアップ
- Main フォーム
- Table フォーム
- SQL フォーム
- 「行の編集」オプション
- Palm OS 固有の機能

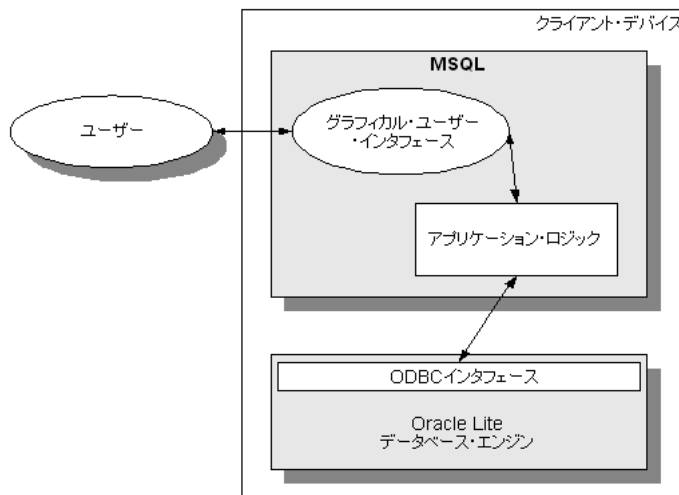
概要

Mobile SQL は、クライアント・デバイス（ラップトップ、Palm OS、EPOC および Windows CE）上で実行されるアプリケーションです。これを使用すると、ローカル・データベースに対して SQL 文を実行できます。このアプリケーションは開発ツールでもありコード例でもあります。ユーザーは、基盤となる Oracle Lite データベース・エンジンの ODBC および Oracle Lite OKAPI インタフェースによって提供される機能にアクセスできます。Mobile SQL（Palm OS マシン用）は、GUI ベースのインタフェースを持ちます。

データベース・アクセス

Mobile SQL は、ODBC と OKAPI の両方のインタフェースを介して、データベースにアクセスします。ほとんどの機能は ODBC を介して実行されますが、ODBC で処理できない機能は OKAPI ファンクション・コールを使用して実装されます。

図 D-1 Mobile SQL の概要



Mobile SQL では、次の機能が提供されます。

- データベースの作成と削除
- データベースへの接続
- データベースで利用できる表のリスト
- 表ビューでの表の行と列の表示
- 表の列の更新

- 表からの行の削除
- SQL 文入力用の対話型コンソール

インストールとセットアップ

Mobile SQL の使用を開始するには、次の構成手順に従います。

1. Java Runtime Engine バージョン 1.2.2 をインストールしてあることを確認します。バージョンが不確実な場合は、コンソール・コマンド「`java -version`」で、使用中の Java バージョンが返されます。
2. Palm Computing のインストール ツールを使用して、**msql.prc**、**okindex.prc** および **odbc.prc** をインストールします。

インストールすると、Palm OS マシン画面に Mobile SQL アイコンが表示されます。

Mobile SQL の起動と停止

Mobile SQL は、デバイス画面で「Mobile SQL」アイコンをクリックして起動します。

Main フォーム

プログラムを実行したときに最初に表示されるフォームで、データベースへの接続をユーザーに要求します。ここで、データベースの作成や削除もできます。

図 D-2 Mobile SQL データベース画面



Main フォームの構成要素は、次のとおりです。

表 D-1 Main フォームのオプション

項目	説明
データベース・リスト	ローカル・デバイス上の使用可能な Oracle Lite データベースを表示します。
オープン	クリックすると、現在選択（または入力）されているデータベースがオープンされます。Mobile SQL により Table フォームが表示されます。
情報	「情報」ダイアログ・ボックスを表示します。
DB 作成	「列の編集」ダイアログ・ボックスを表示します。
削除	「確認」ダイアログ・ボックスに「Are you sure you want to delete the database "<Name>"?» というメッセージを表示します。<Name> は、現在選択されているデータベース名です。「OK」ボタンをクリックすると、アプリケーションはデータベースを削除します。
上向き矢印	現在の選択をリスト内の 1 つ前のデータベースに移動します。
下向き矢印	現在の選択をリスト内の次のデータベースに移動します。

Table フォーム

データベースを選択すると、プログラムはそのデータベースに接続して、Table フォームを表示します。Table フォームでは、データベース内で使用可能なすべての表に対する簡単なビューが提供されます。表を選択すると、表の内容が表示されます。

図 D-3 表ビュー

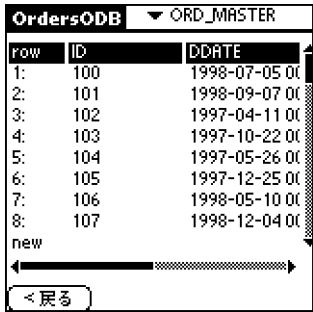


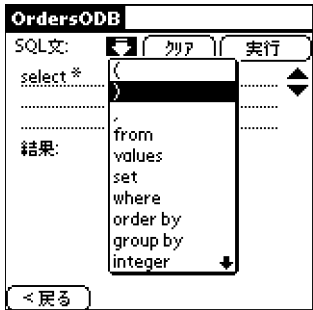
Table フォームの構成要素は、次のとおりです。

項目	説明
表リスト	現在のデータベース内のすべての表をリストします。リストから表を選択すると、「行テーブル」に表の行が表示されます。
テーブルの行および列	<p>最初は空で、表が選択されたときに表コントロールにより表の内容が表示されます。表コントロールの各セルには、表の列が表示されます。</p> <p>ソート: 表コントロールの先頭行には、列の名前が表示されます。先頭行の列をクリックすると、プログラムは列によって表をソートします。最初にクリックしたとき、昇順で列がソートされ、列名の後にプラス (+) 符号が表示されます。同じ列をクリックすると、ソート順が降順 (-) と昇順 (+) で切り替わります。ソートに使用できる列は一度に1つのみです。</p> <p>スクロール: 列の数が表の1画面に表示できる数よりも多い場合、列をスクロールできるよう水平スクロールバーが表示されます。行の数が表の1画面に表示できる数よりも多い場合、行をスクロールできるよう垂直スクロールバーが表示されます。</p>
テーブル	テーブル・フォームに切り替えます。
SQL 文実行	SQL フォームに切り替えます。
挿入	Mobile SQL により、表エントリを動的に編集できます。エントリをクリックすると、Mobile SQL により挿入位置が提供されます。新しい行を挿入します。「行の編集」ダイアログ・ボックスが表示されます。
削除	現在の行を削除します。行番号をクリックすると、行の削除の確認を要求されます。

SQL フォーム

SQL フォームを使用すると、ユーザー・インタフェースから SQL 文を入力し、実行できます。これにより、ODBC レイヤーによって公開された動的 SQL 機能にアクセスできます。

図 D-4 コマンド入力メニュー



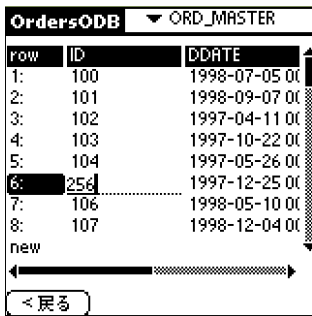
SQL フォームの構成要素は、次のとおりです。

項目	説明
SQL 文	実行する SQL 文の入力をユーザーに要求します。文はセミコロン (;) 文字で終了できますが、必須ではありません。文の最大長は、511 文字です。下向き矢印アイコンをクリックすると、文の作成過程を解析して各ステップで最も使用される可能性の高いオプションをいくつか含んだメニューが表示されます。
クリア	このボタンをクリックすると、「SQL 文」編集ボックスと「結果」テキスト・ボックスの両方が消去されます。
実行	クリックすると、Mobile SQL により SQL 文が実行されます。
結果	SQL 文実行後の結果が表示されます。
テーブル	Table フォームに切り替えます。
SQL 文実行	SQL フォームに切り替えます。
編集、カット	選択されたテキストがある場合は、これを SQL 文からクリップボードにカットします。
編集、コピー	選択されたテキストがある場合は、これを「SQL 文」編集ボックスからクリップボードにコピーします。
編集、ペースト	クリップボードにテキストがある場合は、これを「SQL 文」フィールドの現在のカーソル位置にコピーします。

「行の編集」オプション

「行の編集」ダイアログ・ボックスを使用すると、特定の行の列を変更したり、新しい行を作成できます。

図 D-5 行編集の例



「行の編集」ダイアログ・ボックスの構成要素は、次のとおりです。

表 D-2 「行の編集」ダイアログ・ボックスのオプション

項目	説明
ColumnsList	表内のすべての列名と値のリスト。新しい列が選択された場合、「値」編集ボックスの値が一時列に更新され、列リストが更新されます。
NullItButton	プッシュ・ボタン。このボタンを押すと、現在の列が NULL にされます。
値	現在の選択列の値を入力できる編集ボックス。
OK	プッシュ・ボタン。このボタンを押すと、列が更新され、ダイアログが消去されます。
取消し	プッシュ・ボタン。このボタンを押すと、変更内容が取り消され、ダイアログ・ボックスが消去されます。

Palm OS 固有の機能

次に、特に Palm OS 用に実装された機能をすべてリストします。

- インプレース編集 — 画面上の任意のセルをクリックすることで、Table フォームの列を変更できます。ダイアログをロードせずに列を編集できます。
- 表のビーム転送 — IRDA リンクを使用して、別のデバイスに表（スキーマおよびデータ）を送信できます。
- SQL ヒント — 推奨するすべての SQL キーワードおよび表名のリスト・ボックス。
- ORDER BY — SELECT 文によって戻された行を、指定された列に従って順序付けします。デフォルトの順序は降順です。オプションで、ASC キーワードで昇順を指定できます。

用語集

3 層 Web モデル (Three-Tier Web Model)

クライアント、中間層およびサーバーを含むインターネット・データベース構成。
Web-to-Go アーキテクチャは 3 層 Web モデルに準拠しています。

Apache Server

National Center for Supercomputing Applications (NCSA) から発表されたパブリック・ドメインの HTTP サーバー。

Java Servlet Development Kit

Java サーブレットの開発のために JavaSoft 社が提供しているツール。

JavaServer Pages

JavaServer Pages (JSP) とは、開発者がページの基になるコンテンツを変更せずにページのレイアウトを変更できるようにするテクノロジーです。JSP は HTML と Java コードを使用し、動的コンテンツとビジネス・ロジックを結び付けたプレゼンテーションを可能にします。

Java Web Server Development Kit

Java Web Server Development Kit 1.0.1 は、JavaServer Pages (JSP) と Java サーブレットの開発のために JavaSoft 社が提供しているツールです。

Java アプレット (Java Applets)

ブラウザで実行される小規模なアプリケーションで、動的コンテンツを追加することにより HTML ページの機能を拡張します。

Java サーブレット (Java Servlets)

Java で作成されているプロトコルで、プラットフォームに依存しないサーバー側コンポーネント。Java サーブレットは Java 対応のサーバーを動的に拡張し、要求 - 応答方式を使用して作成されたサービスのための汎用フレームワークを提供します。

JDBC

Java Database Connectivity (JDBC) は Java クラスの標準セットで、リレーショナル・データに対してベンダーに依存しないアクセスを提供します。JDBC クラスは ODBC をモデルにしたもので、複数データベースへの同時接続、トランザクション管理、単純問合せ、バインド変数によるコンパイル済文の操作、ストアド・プロシージャへのコールなどの標準機能を提供します。JDBC では、静的 SQL と動的 SQL の両方がサポートされます。

LEAPFROG シーケンス (Leapfrog Sequence)

Web-to-Go がサポートする 2 つのシーケンスのうちの 1 つで、オフライン・モードの Web-to-Go の Mobile クライアントに対して一意の主キー値を提供するために使用されるもの。LEAPFROG シーケンスの初期値はクライアントごとに異なり、各シーケンスの増分は最大クライアント数より大きい値に設定されます。

MIME

Multipurpose Internet Mail Extensions (MIME) とは、メッセージの内容を記述するためにインターネット上で使用されるメッセージ形式です。MIME は、HTTP サーバーが配布対象ファイルのタイプを記述するために使用します。

MIME タイプ (MIME Type)

Multipurpose Internet Mail Extensions (MIME) により定義されているファイル形式。

Web-to-Go の Mobile クライアント

Web-to-Go の Mobile クライアントは、Web-to-Go の 3 層 Web モデルのクライアント層です。この層には、Mobile サーバーと Oracle Lite データベースが含まれます。Web-to-Go は、ユーザーがオフライン・モードに切り替えたときにユーザー・アプリケーションとデータを Oracle Lite にレプリケートします。ユーザーが元のオンライン・モードに切り替えると、Web-to-Go はすべてのデータ変更を Oracle9i にレプリケートします。

Mobile Development Kit for Web-to-Go

Mobile Development Kit for Web-to-Go を使用すると、アプリケーション開発者は、Java サブレット、JavaServer Pages (JSP) または Java アプレットで構成される Web-to-Go アプリケーションの開発とデバッグを行えます。

Mobile サーバー

Mobile サーバーは、Web-to-Go3 層モデルのアプリケーション・サーバー層に常駐し、データベース・サーバー内での Web-to-Go の Mobile クライアントからのデータ変更要求を処理します。Mobile サーバーは、Oracle HTTP Server、Apache Server、Oracle9i Application Server およびスタンドアロンの Mobile サーバーとともに実行されるように構成できます。

Mobile サーバー・リポジトリ (Mobile Server Repository)

Mobile サーバー・リポジトリとは、Oracle8i に常駐する仮想ファイル・システムです。このリポジトリは、すべてのアプリケーション・ファイルとアプリケーション定義を含む持続リソース・リポジトリです。

ODBC

ODBC (Open Database Connectivity) は Microsoft 社の標準で、様々なプラットフォーム上でのデータベース・アクセスを可能にします。Web-to-Go の Mobile クライアント上では、トラブルシューティング用に ODBC サポートを使用可能にします。ODBC サポートを使用すると、ローカルな Oracle Lite データベースに格納されているクライアントのデータを表示できます。この情報を表示するには、SQL*Plus を使用します。

Oracle8i

Oracle8i は、Mobile サーバーのデータベース・コンポーネントです。Web-to-Go の Mobile クライアントがオンライン・モードのときは、アプリケーションとデータは Oracle9i に格納されます。

Oracle Lite

Oracle Lite は、Web-to-Go の Mobile クライアントのデータベース・コンポーネントです。クライアントがオフライン・モードのときは、アプリケーションとデータは Oracle Lite に格納されます。

SQL

Structured Query Language (SQL) は、リレーショナル・データベース・エンジンのほとんどで使用される非手続き型データベース・アクセス言語です。SQL 文はデータ・セットに対して実行される操作を記述します。SQL 文がデータベースに送られると、データベース・エンジンは指定されたタスクを実行するプロシージャを自動的に生成します。

SQL*Plus

Oracle Lite データベースに接続してデータにアクセスするツール。SQL*Plus を使用するには、ODBC サポートを使用可能にしておく必要があります。

Web-to-Go

Oracle Web-to-Go は、Web ベースのモバイル・データベース・アプリケーションを作成および配布するためのフレームワークです。Web-to-Go には、Web-to-Go の Mobile クライアント、Mobile サーバーおよび Oracle8i で構成される 3 層データベース・アーキテクチャが含まれます。サーバーから一元管理されており、サーバーに接続されたとき (オンライン) またはサーバーから切断されたとき (オフライン) に Web-to-Go アプリケーションを実行できます。オフラインのときは Web-to-Go はデータをローカルにキャッシュし、オンラインに戻ったときにそのデータをサーバーと同期します。

WINDOW シーケンス (Window Sequence)

Web-to-Go がサポートする 2 つのシーケンスのうちの 1 つで、オフライン・モードの Web-to-Go の Mobile クライアントに対して一意の主キー値を提供するために使用されるもの。WINDOW シーケンスには、一意の値範囲が含まれます。他のクライアントと値の範囲は重複しません。クライアントがシーケンスの範囲内の値をすべて使用すると、Web-to-Go は新しい一意の値範囲を持つシーケンスを再び作成します。

一意キー (Unique key)

表の一意キーは、表の各列での一意の列または列グループです。UNIQUE KEY 制約を満たすには、一意キーの値が表の複数の行に出現することはできません。ただし、PRIMARY KEY 制約とは異なり、単一列からなる一意キーは NULL 値を含むことができます。

位置付け DELETE (Positioned DELETE)

位置付け DELETE 文により、カーソルの現在行が削除されます。書式は次のとおりです。

```
DELETE FROM table
WHERE CURRENT OF cursor_name
```

位置付け UPDATE (Positioned UPDATE)

位置付け UPDATE 文により、カーソルの現在行が更新されます。書式は次のとおりです。

```
UPDATE table SET set_list
WHERE CURRENT OF cursor_name
```

オフライン・モード (Offline Mode)

Web-to-Go の Mobile クライアントが Mobile サーバーから切断されている状態。オフライン・モードでは、クライアント・アプリケーションはローカルに実行され、データは Oracle Lite でアクセスおよび格納されます。「[オンライン・モード](#)」も参照。

オンライン・モード (Online Mode)

Web-to-Go の Mobile クライアントが Mobile サーバーに接続されている状態。「[オフライン・モード](#)」も参照。

外部キー (Foreign Key)

外部キーとは表またはビューに存在する列または列グループのことで、その値は別の表またはビューに存在する行を参照します。外部キーには、一般に、別の表の主キー値と一致する値が含まれます。「[主キー](#)」も参照。

結合 (Join)

2 つの異なる表またはビューに存在するキー（主キーと外部キーの両方）の間に確立された関係。結合は、リレーショナル・データベース内の重複したデータを排除するために正規化された表のリンクに使用します。結合リンクの一般的なものとしては、1 つの表の主キーを別の表の外部キーにリンクして、マスター・ディテール関連を確立するものがあります。結合は SQL 文の WHERE 句条件に対応します。

コントロール・センター (Control Center)

Mobile サーバー・コントロール・センターはブラウザ内で実行される Web ベースのアプリケーションで、これを使用すると Web-to-Go アプリケーションとそのユーザーの管理が容易になります。管理者はコントロール・センターを使用して、ユーザーまたはグループに対するアクセス権の付与と取消し、スナップショット・テンプレート変数の変更、Web-to-Go からのアプリケーションの削除などの機能を実行します。

サイト (Site)

Web-to-Go は、Web-to-Go の Mobile クライアント上の各ユーザーに対してデータベースを作成します。このデータベースはサイトと呼ばれます。1つのクライアントに複数のサイトを含められますが、サイトは1人のユーザーに1つのみ可能です。ユーザーは、異なるクライアント上に複数のサイトを所有できます。

索引 (Index)

表内のそれぞれの行に対する高速アクセスを提供するデータベース・オブジェクト。索引を作成すると、表のデータに対して実行される問合せおよびソート操作を高速化できます。また、索引を使用して、UNIQUE KEY 制約や PRIMARY KEY 制約などの制約を表に対して規定することもできます。

索引はいったん作成されると自動的にメンテナンスされ、データベース・エンジンにより可能な限りデータ・アクセスのために使用されます。

参照整合性 (Referential Integrity)

参照整合性は、レコードが追加、修正または削除されたときにメンテナンスされるマスター・ディテール関連内の表間のリンクの精度として定義されます。

マスター・ディテール関連を注意深く定義しておくことにより、参照整合性が高まります。データベース内の制約によって、データベース（クライアント / サーバー環境でのサーバー）レベルの参照整合性が規定されます。

参照整合性の目的は、孤立したレコード（マスター・レコードとの有効なリンクを持たないディテール・レコード）が作成されないようにすることです。参照整合性を規定する規則により、結果として孤立したレコードを作成するような、マスター・レコードの削除や更新、またはディテール・レコードの挿入や更新を予防できます。

シーケンス (Sequence)

順次数を生成するスキーマ・オブジェクト。シーケンスを作成した後は、これを使用してトランザクション処理用の一意の順次数を生成できます。これらの一意の整数には、主キー値を含むことができます。トランザクションで順序番号が生成される場合、トランザクションをコミットしたかロールバックしたかにかかわらずシーケンスが即時増分されます。

[「WINDOW シーケンス」](#) および [「LEAPFROG シーケンス」](#) も参照。

実表 (Base Table)

ビューの基になるデータのソースで、表またはビューのいずれか。ビュー内のデータにアクセスするとき、実際は実表のデータにアクセスしています。

シノニム (Synonym)

表、ビュー、シーケンス、スナップショットまたは別のシノニムに対する代替名（エイリアス）。

主キー (Primary Key)

表の主キーは、表内の各行を一意に識別するのに使用される 1 つの列または列グループです。主キーを使用すると表のレコードにすばやくアクセスできます。また主キーは 2 つの表またはビューの間の結合の基礎として頻繁に使用されます。それぞれの表に対して主キーは 1 つしか定義できません。

PRIMARY KEY 制約を満たすには、主キー値が表の 2 つ以上の列で使用されたり、主キーの一部の列に NULL 値が含まれないようにします。

スキーマ (Schema)

表、ビュー、索引、シーケンスなどを含む、名前の付いたデータベース・オブジェクトの集まり。

スナップショット (Snapshot)

スナップショットとは Web-to-Go が Oracle データベースからリアルタイムで取得するアプリケーション・データのコピーで、オフラインになる前にクライアントにダウンロードされます。スナップショットは、データベース表全体のコピー、または表の行のサブセットのコピーです。ユーザーが初めてオンラインからオフラインに切り替えるとき、Web-to-Go はクライアント・マシン上にスナップショットを自動的に作成します。その後、オンラインまたはオフラインに切り替えるたびに、Web-to-Go はスナップショットの複雑さに応じて、スナップショットを最新のデータでリフレッシュするか、全体を再作成します。

整合性制約 (Integrity Constraint)

表の 1 つ以上の列に入力できる値を制限する規則。

接続 (Connected)

ユーザー、アプリケーションまたはデバイスがサーバーに接続されていることを指す一般的な用語。Web-to-Go の Mobile クライアントは、オンライン・モードのときに「接続」されています。

切断 (Disconnected)

ユーザー、アプリケーションまたはデバイスがサーバーに接続されていないことを指す一般的な用語。Web-to-Go の Mobile クライアントは、オフライン・モードのときに「切断」されています。

データベース・オブジェクト (Database Object)

データベース・オブジェクトとは、表、ビュー、シーケンス、索引、スナップショットまたはシノニムなどの名前の付けられたデータベース構造体です。

データベース・サーバー (Database Server)

Web-to-Go の 3 層 Web モデルの 3 番目の層。アプリケーション・データを格納します。

同期 (Synchronization)

Web-to-Go が Web-to-Go の Mobile クライアントと Oracle9i の間でデータをレプリケートするために使用するプロセス。Web-to-Go は、ユーザーがオフライン・モードに切り替えたときにユーザーのアプリケーションおよびデータを Oracle Lite にレプリケートします。ユーザーが元のオンライン・モードに切り替えると、Web-to-Go はすべてのデータ変更を Oracle9i にレプリケートします。

トランザクション (Transaction)

リレーショナル・データベース内の選択されたデータに対して加えられる一連の変更。トランザクションは通常、ADD、UPDATE、DELETE などの SQL 文を使用して実行します。トランザクションは、コミットされた（変更が永続的になる）とき、またはロールバックされた（変更が破棄された）ときに完了します。

トランザクションの前に問合せが実行されることがよくあります。問合せを使用して、変更対象の特定のレコードをデータベースから選択しておきます。「SQL」も参照。

パッケージ・ウィザード

パッケージ・ウィザードを使用すると、管理者が Web-to-Go アプリケーションを Mobile サーバー・リポジトリにパブリッシュできます。管理者は、パッケージ・ウィザードを使用して新しい Web-to-Go アプリケーションを作成したり、既存のアプリケーション定義を編集できます。

パブリケーション項目 (Publication Item)

パブリケーション項目は、クライアントがアクセスできるデータ・サブセットを指定する SQL の SELECT 文です。パブリケーション項目は、通常クライアント・デバイス上のレプリカ表に対応します。パブリケーション項目は、Mobile Server Admin API を使用して作成できます。この API には、パブリッシュ・サブスクライブ・モデルを実装する Java 関数が含まれています。API 中のこれらの関数は、Java プログラムの内部から標準のファンクション・コールとしてコールできます。

ビュー (View)

1 つ以上の表（または他のビュー）から選択されたデータをカスタマイズして表したもの。ビューは「仮想的な表」のようなもので、複数の表（実表と呼ばれます）およびビューからのデータを関連させ、組み合わせることができます。ビューは表示されるデータの選択条件を指定できるため、一種の「格納された問合せ」といえます。

ビューは、表のように、行と列に編成されます。ただし、ビューには、データそのものは含まれません。ビューを使用すると、複数の表またはビューを 1 つのデータベース・オブジェクトとして扱うことができます。

表 (Table)

行と列に編成されたデータを格納するデータベース・オブジェクト。上手に設計されたデータベースでは、各表に単一のトピックに関する情報（たとえば従業員や顧客の住所など）が格納されます。

マスター・ディテール関連 (Master-Detail Relationship)

1 つの表またはビュー（ディテール表またはビュー）の複数行が、別の表またはビュー（マスター表またはビュー）の単一のマスター行に関連付けられている場合に、マスター・ディテール関連がデータベース内の表またはビューの間に存在すると言います。

マスター行およびディテール行は通常、ディテール表またはビュー内の外部キー列と一致するマスター表またはビュー内の主キー列により結合されます。

主キーの値を変更した場合、アプリケーションでは、外部キーの値が主キーの値と一致するように一連の新しいディテール・レコードを問い合わせる必要があります。たとえば、EMP 表内のディテール・レコードが、DEPT 表内のマスター・レコードと同期される場合、DEPT 内の主キーは DEPTNO で、EMP 内の外部キーは DEPTNO にします。「主キー」および「外部キー」も参照。

モードの切替え (Switching Modes)

Web-to-Go の Mobile クライアントがオフラインに切り替えたりオンラインに戻るために使用するプロセス。クライアントがオフライン・モードに切り替えると、Oracle Lite がオフラインで作業するために必要なすべてのアプリケーションとデータがダウンロードされます。クライアントがオンラインに戻ったときに、対するデータ変更を Oracle9i と同期します。

レジストリ (Registry)

レジストリには、Web-to-Go の一意の名前と値のペアが含まれます。レジストリの名前はすべて一意である必要があります。

レプリケーション (Replication)

分散データベース・システムを構成する複数のデータベース内で、データベース・オブジェクトをコピーしメンテナンスするプロセス。1 つのサイトに適用された変更が取得されローカルに格納されてから、各リモート・サイトに転送され適用されます。レプリケーションは、共有データに対する高速のローカル・アクセスをユーザーに提供し、データ・アクセスの代替オプションを提供してアプリケーションの使用を保護します。1 つのサイトが使用不可になっても、残りのサイトへの問合せや更新ができます。

レプリケーションの競合 (Replication Conflict)

レプリケーションの競合は、1 つのデータに対して矛盾する変更が加えられるときに発生します。Web-to-Go では、切断されているクライアント用のシーケンス値を使用して、レプリケーションの競合を回避します。

ワークスペース (Workspace)

Mobile サーバーのワークスペースとは、Web-to-Go アプリケーションに対するアクセスをユーザーに提供する Web ページです。Web-to-Go は、ユーザーが Web-to-Go にログインした後に、ユーザーのブラウザ内にワークスペースを生成します。ワークスペースは、ユーザーが使用できるすべてのアプリケーションのアイコン、リンクおよび説明を表示します。アプリケーションを使用できるようになるのは、管理者がアプリケーションを Web-to-Go システムにパブリッシュし、ユーザーに対してアクセス権を付与した後です。

索引

B

BLOB, B-11

C

CHAR, 3-9, B-11

CodeWarrior, 5-9

プロジェクトの作成, 5-14

COMMIT, B-7

CREATE INDEX, B-3

CREATE TABLE, B-2

D

DATE, 3-9, B-11

DDL 操作, 3-2

DECIMAL, B-11

DELETE, B-4

DOUBLE, B-11

DOUBLE PRECISION, 3-9

DROP INDEX, B-4

DROP TABLE, B-3

F

FLOAT, 3-9, B-11

G

GNU GCC, 5-8

H

HotSync

ボタンの再割当て, 5-25

HTTP 転送

Palm Computing Platform, 5-23

I

INSERT, B-5

INSTEAD OF トリガー, 3-4

INT, B-11

INTEGER, 3-9, B-11

J

jar ファイル

パッケージ・ウィザードによるパッケージ化, 6-17

L

LONG RAW, 3-9

M

Message Generator and Processor, 3-2

Mobile SQL, 5-8

SQL 文の使用, 5-13

インストール, 5-10

起動, 5-11

データベースの作成, 5-11

表の表示, 5-12

Mobile Sync Client

概要, 1-3, 3-2

構成, 1-8

N

NUMBER, 3-9, B-11
NUMERIC, B-11

O

ODBC, 1-4
 開発, 5-2
ODBC サポート
 Palm Computing Platform, C-2
OKAPI, 1-4
 開発, 5-2
Orders サンプル・アプリケーション
 Palm Computing Platform での使用, 4-3
 サーバーの準備, 4-3
 デバイスでのインストール, 4-5
 ファイル, 4-2

P

Palm Computing Platform
 アプリケーションの開発, 5-7
 固有のアプリケーションのサポート, 1-4
 データベース・サポート, 1-4
PL/SQL, 3-3

R

REAL, B-11
ROLLBACK, B-7

S

Satellite Forms
 アプリケーションの作成, 5-17
 サポート, 1-4
 データ型, 3-9
 要件, 5-17
SELECT, B-5
SMALLINT, 3-9, B-11
SQLAllocConnect, C-3
SQLAllocEnv, C-4
SQLAllocHandle, C-4
SQLAllocStmt, C-6
SQLBindCol, C-14
SQLBindParameter, C-11

SQLConnect, C-9
SQLDescribeCol, C-15
SQLDisconnect, C-10
SQLError, C-15
SQLExecDirect, C-12
SQLExecute, C-13
SQLFetch, C-13
SQLFreeConnect, C-7
SQLFreeEnv, C-7
SQLFreeHandle, C-8
SQLFreeStmt, C-8
SQLGetData, C-16
SQLNumResultCols, C-17
SQLPrepare, C-11
SQLRowCount, C-17
SQLTransact, C-18
SQL サポート
 Palm Computing Platform, B-2
SQL 文
 Palm Computing Platform でサポート, B-2

T

TIME, B-11
TIMESTAMP, B-11

U

UPDATE, B-6

V

VARCHAR, 3-9, B-12
VARCHAR2, 3-9, B-12

X

XML ファイル, 6-17

あ

アプリケーション
 Satellite Forms で作成, 5-17
 パッケージ・ウィザードによる編集, 6-19
 パッケージ・ウィザードによる命名, 6-3
 パブリッシュ, 6-18
 ファイルのリスト表示, 6-5

「アプリケーション」パネル
パッケージ・ウィザード, 6-3

い

インストール, 1-5

う

ウィニング・ルール, 3-7

え

エミュレータ, 1-8
Palm Computing Platform, 5-10
エラー, 3-7

お

オブジェクトのネーミング, 3-3
親表
INSTEAD OF トリガー, 3-4
更新可能, 3-3
ヒント, 3-4

か

開発過程, 1-5
開発環境
Palm Computing Platform, 2-2
インストール, 2-2
構成, 2-2
開発ツール
Palm Computing Platform, 5-8
外部キー制約, 3-5
違反, 3-5
環境サポート
EPOC, 1-4
Palm Computing Platform, 1-4

き

競合, 3-7

く

クライアント
パブリケーションへのサブスクライブ, 3-20
クライアント・データベース
作成, 3-2
グループ関数, B-12

こ

高速リフレッシュと更新, 3-3

さ

索引
パブリケーション項目用の作成, 3-16
サブスクリプション, 3-2, 3-10
インスタンス化, 3-22
サブスクリプション・パラメータ, 3-10
定義, 3-22

し

システム・カタログ・ビュー, A-1
主キー索引, 3-2

す

スナップショット
インポート, 6-13
パッケージ・ウィザードによる作成, 6-12
編集, 6-15
「スナップショット」パネル
パッケージ・ウィザード, 6-8

て

ディレクトリ構造
Palm Computing Platform, 2-2
データ型
Oracle Lite, 3-8
Satellite Forms, 3-9
サポート, B-11
マッピング, 3-8
データベース
Mobile SQL で作成, 5-10, 5-11
作成, 5-10

データベース・インタフェース

ODBC, 1-4

OKAPI, 1-4

データベース・サポート, 3-3

「データベース」パネル

パッケージ・ウィザード, 6-7

テスト

デバイスを使用, 1-9

転送

構成

Palm Computing Platform, 5-23

転送の構成

Palm Computing Platform デバイス, 5-23

転送プロトコル

サポート, 1-4

と

トランザクション

実行, 3-27

ページ, 3-27

は

ページ、トランザクション, 3-27

バージョンニング, 3-7

パッケージ・ウィザード

jar ファイルのパッケージ化, 6-17

「アプリケーション」パネル, 6-3

起動, 6-2

スナップショットのインポート, 6-13

スナップショットの編集, 6-15

「スナップショット」パネル, 6-8, 6-12

「データベース」パネル, 6-7

ファイルのソート, 6-6

「ファイル」パネル, 6-5

パブリケーション, 3-2, 3-9

クライアントのサブスクライブ, 3-20

作成, 3-13

パブリケーション項目, 3-9

索引の作成, 3-16

作成, 3-14

パブリケーションへの追加, 3-17

パブリッシュ・サブスクライブ, 1-4

パブリッシュ・サブスクライブ・モデル, 3-2, 3-9

実装, 3-10

ひ

日付関数, B-13

ビュー

高速リフレッシュと更新, 3-3

ヒント, 3-4

ふ

「ファイル」パネル

パッケージ・ウィザード, 6-5

よ

要件

JDK, 2-2

Oracle8, 2-2

一般, 2-2

れ

レプリケーション

エミュレータを使用, 1-9

エラー, 3-7

競合, 3-7

レプリケート・シーケンス

クライアントごとのパーティション化, 3-21

作成, 3-20

レプリケート・シーケンスのパーティション化, 3-21