

Oracle8i プラットフォーム共通日本語 README

リリース 8.1.7

2001 年 4 月

部品番号: J02828-03

ORACLE®

Copyright © 2000, Oracle Corporation
All Right Reserved

Oracle と Oracle のロゴは Oracle Corporation の登録商標です。Net8、Oracle8i、Oracle Database Configuration Assistant、Oracle *interMedia*、Oracle *interMediaText*、Oracle Names、PL/SQL、Pro*C/C++、Pro*COBOL、SQL*Forms、SQL*Loader および SQL*Plus は、Oracle Corporation の商標です。記載されているその他の製品名および社名は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、それぞれ該当する所有者の商標です。

目次

はじめに.....	16
マニュアルに記載されている名称について	16
英語オンライン・ドキュメントの扱いについて	16
Oracle JServer の名称について	16
最新情報の入手について	16
第 1 章 RDBMS の Readme ファイル	17
ユーティリティについて	17
ユーザーSYS としてデータベースに接続する方法	17
INTEGER と SMALLINT 用の新規の SIGNED および UNSIGNED キーワード	18
SQL*Loader 入カルーチンの動作の変更	18
CONTINUEIF と新規 PRESERVE キーワード	19
例 1-1 PRESERVE キーワードなしの CONTINUEIF THIS	19
例 1-2 PRESERVE キーワードありの CONTINUEIF THIS	20
例 1-3 PRESERVE キーワードなしの CONTINUEIF NEXT	20
例 1-4 PRESERVE キーワードありの CONTINUEIF NEXT	20
SQL*Loader に関するドキュメントの訂正	21
BAD および DISCARD ファイルの形式	21
Bad ファイルのデフォルトのファイル名	21
空白のみを含む日付フィールドのロード	21
フィールドに位置とデリミタの両方が指定されている場合の動作	22
WHEN、NULLIF および DEFAULTIF 句の使用について	22
例 1-5 DEFAULTIF 句が評価されない場合	24
例 1-6 DEFAULTIF 句が評価される場合	24
例 1-7 DEFAULTIF 句で位置を指定する場合	25
例 1-8 EFAULTIF 句でフィールド名を指定する場合	25
エクスポート/インポートについて	26
増分エクスポート機能のサポートに関して	26
ユーザーSYS としてデータベースに接続する方法	26

処理制限事項.....	27
エクスポートとインポートに関するドキュメントの訂正.....	27
表名の大 / 小文字区別の保持.....	27
各国語サポートについて.....	28
Unicode 3.0 のサポート.....	28
データベース・キャラクタ・セット移行ツール.....	28
中国語の簡字体と繁体字の変換.....	28
改善された ALTER DATABASE CHARACTER SET のサポート.....	29
AL24UTF8SS キャラクタ・セットの非サポート.....	29
NCHAR のサポート.....	29
制限付き ALTER DATABASE CHARACTER SET コマンドのサポート (CLOB および NCLOB)	30
レプリケーションについて.....	31
プロシージャ.....	31
SSL 接続.....	31
XML Parser について.....	31
第 2 章 Oracle Advanced Security.....	32
Oracle Advanced Security の概要.....	32
このリリースでの変更内容.....	33
このリリースでの新機能.....	33
データの暗号化と整合性.....	34
外部認証とシングル・サインオン.....	34
ENTRUST サポート.....	35
公開鍵インフラストラクチャ.....	36
暗号化と認証.....	36
判明している不具合とその回避策.....	36
Secure Sockets Layer.....	37
判明している不具合とその回避策.....	37
Oracle Wallet Manager.....	37
判明している不具合とその回避策.....	37
Oracle Enterprise Security Manager 2.0	38
判明している不具合とその回避策.....	38

Java SSL	39
判明している不具合とその回避策	39
インストール	39
旧リリースの問題点に対する拡張機能	39
第 3 章 Net8 について	41
Net8 の概要	41
インストール	42
廃止された NET8 パラメータ	42
names.ora パラメータ	42
マルチスレッド・サーバー (MTS) ・パラメータ	42
Net8 の既知の問題	43
ディレクトリ・ネーミング	43
Net8 の制限事項	43
Net8 Assistant および Net8 Configuration Assistant の制限事項	45
Net8 インストールの既知の問題	46
マニュアルの正誤表	46
第 4 章 Oracle JSP について	49
Oracle JSP の新機能	49
Oracle JSP リリース 1.1.0.0.0 での問題点と制限事項	50
Oracle JSP リリース 1.0.0.6.1 からの不具合の修正	51
<jsp:expression>および<jsp:declaration>が機能しない (Bug#1358636)	51
errorPage / jsp:forward / jsp:include で「..」が機能しない (Bug#1288730)	51
jsp:forward で HTML ページに転送できない (Bug#1261458)	52
場合によっては静的テキスト内の改行文字が欠落する (Bug#1250326)	52
Oracle JSP で html ファイルを動的にインクルードできない (Bug#1238238)	52
Apach でサーブレット別名を使用して Oracle JSP が起動される場合に、 ファイル位置の特定に失敗 (Bug#1237578)	53
エラー・メッセージでのマルチバイト文字「?」の表示 (Bug#1222788)	53
Short が正しくキャストされない (Bug#1213897)	53
jml:useForm - キャストの問題 (Bug#1213777)	54
複数の行にわたるインポート・リストのサポート (Bug#1207720)	54

JSP ファイル名に「-」（ダッシュ）を使用できない（Bug#1155470）	55
Oracle JSP リリース 1.0.0.6.0 以降の不具合の修正	55
JSP とサーブレットでセッションを共有できない（Bug#1225531）	55
別名パスのルート・ディレクトリにあるページに対して Oracle JSP が機能しない（Bug#1230626）	56
別名を使用している場合に request.getPathTranslated() から 不適切な値が戻る（Bug#1233612）	56
アプリケーション関連とページ関連を混在して使用する場合の jsp:include 障害（Bug#1235873）	56
第 5 章 Oracle SQLJ について	57
SQLJ の概要	57
SQLJ 配布の内容	58
インストール	60
注意:	60
新機能	61
Oracle8i リリース 8.1.6.0.0 以降の変更	61
言語	61
トランスレータ	61
ランタイム	61
Oracle8i リリース 8.1.5.0.0 以降の変更	62
言語	62
トランスレータ	62
チェック	63
パフォーマンス	63
ランタイム	63
プラットフォーム固有の情報	64
一般的なトラブルシューティング	64
UNIX プラットフォーム	64
Windows プラットフォーム	65
修正済みの不具合	65
SQLJ リリース 8.1.7 での修正された不具合について	65
トランスレータ	65

ランタイム	65
SQLJ リリース 8.1.6 での修正された不具合について	66
言語	66
トランスレータ	66
オンライン・チェック	67
ランタイム	67
判明している制限事項と問題点	67
言語	67
トランスレータ	68
ランタイム	70
サーバー側 JVM における SQLJ の使用: 判明している制限事項と問題点	70
言語	70
トランスレータ	71
第 6 章 Oracle8i JVM、EJB / CORBA および Oracle Servlet Engine.....	72
Oracle のエンタープライズ Java サーバー・プラットフォーム : Oracle8i JVM.....	72
ドキュメント	72
開始方法	73
構成	73
Java 互換性	74
互換モード	74
CORBA と Enterprise JavaBeans	75
判明している問題	75
メモリー使用量	75
initjvm.sql 実行時のメモリーまたはリソースの不足	76
コンパイル時のメモリー不足	77
Java クラス・ロード時のメモリー不足	77
サーバーに常駐する Java ソースのアクセス可能性	78
Java との DESCRIBE の問題	78
Loadjava と Dropjava	78
エラー・レポート	79
java.version、oracle.jsrserver.version および oracle.server.version	79

SQL*PLUS の Java &&.....	79
RMI とソケット.....	79
マルチバイト文字出力の 255 バイトでの切捨てについて (Bug#895586)	80
Java から SQL、および SQL から Java へのコール.....	80
Java と PL/SQL.....	80
EJB/CORBA - CLASSPATH、LD_LIBRARY_PATH および PATH.....	80
CORBA と EJB の SSL 設定	81
CORBA と EJB のサンプル.....	82
EJB の制限事項.....	84
CORBA の制限事項.....	86
SSL の制限事項.....	87
リリース 8.1.6 との EJB の相違.....	87
リリース 8.1.5 またはリリース 8.1.6 からリリース 8.1.7 へのアップグレード.....	88
CD から利用可能なパッチ	88
Oracle Servlet Engine	88
Oracle Servlet Engine の概要.....	88
一般的な OSE 管理	89
トラブルシューティング.....	90
404 が発生した場合の対策.....	91
一般的な問題に対する対策	91
ファイル・アクセス権とサーブレット・コンテキスト所有権.....	91
Oracle Servlet Engine (mod_ose) の Apache モジュール.....	92
mod_ose モジュールの概要.....	92
構成上の問題.....	93
mod_ose と SSL	94
既知の問題	95
Oracle8i JVM Accelerator (NComp) –ネイティブ・コンパイルの Java コード.....	96
アップグレード、ダウングレード、インポートおよびエクスポート.....	96
セキュリティ.....	97
受入可能な C コンパイラのバージョン.....	97
既知の問題	97

回避する必要がある JDK リリース	97
変換時における Java のメモリ不足	97
C コンパイル時における C コンパイラのメモリ不足	98
フラグ-update が JDK 1.1.x では無効	98
配置 jar のディレクトリ名にピリオド (.) がある場合、 配置は異常終了します。	98
以前 ncomp したパッケージの半分以上が ncomp されていない状態になる	98
すべてのトップレベルが同一のパッケージ UnnamedPackage にある	99
パッケージ P には何も存在しないため、ncomp は不要	99
NEED_NCOMPING の後、ncomp がフリーズ	99
Solaris 固有の「Settings_*.properties」ファイル	99
C 最適化の安全レベル	100
Javavm / admin におけるファイル領域の管理	100
第 7 章 Oracle Internet Directory リリース 2.1.1	102
概要	102
Oracle Internet Directory について	102
この README について	102
カバー・レターおよびライセンス	103
Oracle Internet Directory リリース 2.1.1 のマニュアル	103
インストール・プロセス	103
新規のインストール	103
インストールに必要な Oracle Internet Directory リリース 2.1.1 の コンポーネントの選択方法	103
Windows 固有のインストール事前要件	104
既存のリリース 8.1.7 データベースへの Oracle Internet Directory の インストール	104
インストーラを使用したリモート・データベースへの Oracle Internet Directory インストールは不可	104
カスタム・インストール	104
インストール後の構成ツール	104
Oracle 固有のスキーマのディレクトリへのインストール	105
サイレント・インストール	105

Oracle Internet Directory リリース 2.1.1 のダウングレードを サポートしません.....	105
アップグレード・インストール	106
Oracle Internet Directory サーバー リリース 2.1.1	108
データベースの互換性	108
LDAP サーバーの新機能と拡張機能	108
言語コード (RFC 2596) の使用をサポート	108
拡張パスワード暗号化のサポート	109
複雑な問合せに対するパフォーマンスの強化	109
LDAPSEARCH でのデフォルト参照に対するサポート	110
ランタイムのデバッグ・レベル切替えに対するサポート	110
オブジェクトの追加時における objectClass 階層の解決に対する サポート	110
レプリケーションの新機能と拡張機能	110
管理者操作キューに対するレプリケーション・サイクルが 構成可能となりました。	110
管理者操作キューの操作ツールが使用可能となりました。	110
レプリケーション調整ツールが使用可能になりました。	111
ノードの削除プロシージャが使用可能になりました。	111
他のディレクトリに対する変更サブスクリプションのサポート	111
LDAP サーバーの制限事項と問題	111
Oracle Internet Directory サーバー リリース 2.1.1 の 複数インスタンスは同一マシン上でのみ実行可能です。	111
LDAP サーバーへのレプリケーション・サーバー接続に対する SSL サポートはありません。	112
SSL V2 クライアントはサーバーに接続できません。	112
Oracle Internet Directory データベースのユーザー / パスワードを 変更する必要があります。	112
デフォルト ACP では、スーパー・ユーザーを除く 全ユーザーに対する書込みアクセスが制限されます。	112
索引付き属性名は 28 文字以内です。	112
索引付けできるのは等価の一致規則を持つ属性のみです。	112

索引付き属性の等価の整数一致は、文字列の一致と 同じように動作します。	113
別名の参照解除は LDAP 操作ではサポートされません。	113
構文チェックは LDAP サーバーではサポートされません。	113
高い選択性を持つシングル AVA フィルタでは速度が低下します。	113
必須属性リストの属性オプションの使用はサポートされません。	113
Oracle Internet Directory サーバー接続中に発生する SSL の断続的な ハンドシェイク・エラー (Bug#1398892)	113
レプリケーションの制限事項	114
新しいディレクトリ・レプリケーション・グループ (DRG) の作成	114
既存のディレクトリ・レプリケーション・グループに対する 新しいノードの追加	114
ACL ポリシー情報以外のローカル・システム固有のメタデータは、 レプリケートされません。	114
レプリケーション・サーバーは、RDN コンポーネント間で 空白を保持しません。	114
すでにアクティブなレプリケーション承諾の一部であるノードに、 bulkload.sh を使用してデータを追加できません。	114
ログ・ファイルのロケーション	115
Oracle Internet Directory クライアント リリース 2.1.1	116
Oracle Internet Directory コマンドライン・ツールの拡張機能	116
パラレル・バルク・ロードが Solaris でサポート	116
バルク・ロード・ツールによるバイナリ・データ・ロードのサポート	116
Oracle Internet Directory コマンドライン・ツールの制限事項	116
不適切な LDIF データは、LDAP ツールにおける セグメンテーション障害の原因になります。 (Bug#1103958)	116
LDAP 検索の制限事項	116
ldapsearch は、LDIF 出力をデフォルトでは生成しません。	117
カタログ・マネージャの使用上の問題	117
-U オプションが指定されていないと、SSL モードにおける OID へのアクセス時にコマンドライン・ツールが停止します。	117
フィルタにカタログ化されていない属性が含まれていると、 bulkmodify は誤解を招くようなエラーを戻します。	117
-r オプション付き LDAPADD はサポートされません。	117

Oracle Directory Manager (OID 管理ユーティリティ、oidadmin)	118
リリース 2.0.6 以降の ODM における不具合の修正と拡張機能	118
oidadmin に関する現在の制限事項	120
Oracle Internet Directory の PL/SQL API (DBMS_LDAP)	121
Windows 95、98、2000 および NT 固有の問題	121
データのロードに BULKMODIFY と BULKLOAD は使用できません。	121
パスの設定は、サーバーを再起動するまで有効化されません。	121
Oracle Directory Manager を Windows 98 にインストール後、 最初に ODM の起動時に「Bind Failed」が発生します。	121
同時接続に関する Windows NT ワークステーションの制限事項	122
バルク・ツールの制限事項	122
第 8 章 Oracle Enterprise Manager リリース 2.2	123
Enterprise Manager リリース 2.2 の新機能	124
Enterprise Manager コンソール	124
パフォーマンスの向上	124
Apache Web サーバーおよび Apache for iAS Web サーバーのサポート	124
ジョブ・システムおよびイベント・システム	125
拡張通知	125
エンタープライズ・レポート	125
DBA Studio	126
OMS チューニング用パラメータ	127
Intelligent Agent リリース 8.1.7	127
サポートされるオペレーティング・システム	128
互換性情報	128
Enterprise Manager 製品と Management Server リポジトリ	129
インストール	130
コンソール	130
イベント・システムおよびジョブ・システム	132
イベントの手動消去	132
重複イベント	132
同義イベント・テスト	133
Enterprise Manager リリース 2.0.4 リポジトリからの同義イベントの移行	133

ジョブ履歴またはイベント履歴をリフレッシュ	133
Windows 2000 のノードでのジョブの実行	133
Compaq Tru64 での「パラメータ」タブの表示	134
グループ	134
ポケットベルおよび電子メールのブラックアウト	134
DBA Management Pack	134
Oracle Management Server	136
Windows での Oracle Management Server サービスの作成	136
強化された Oracle Management Server 起動	136
Oracle Management Server の停止	136
Oracle Management Server のステータス	137
Solaris から HP への接続	137
Enterprise Manager Configuration Assistant	137
リリース 2.2 より前のリポジトリ	137
Enterprise Manager Configuration Assistant によるリポジトリの アップグレード	137
Enterprise Manager Migration Assistant	138
ブラウザ・ベースの Enterprise Manager	138
その他	139
第 9 章 Oracle HTTP Server	140
サポートする JDK/JRE	140
Oracle HTTP Server powered by Apache の Global Server ID	140
既知の制約事項と制限事項	141
Oracle HTTP Server におけるサード・パーティ製コンポーネントのサポート	142
mod_perl DBI/DBD-Oracle および Apache::DBI	145
mod_plsql 要件	145
Oracle PL/SQL Web ツールキットのインストールに関する障害	146
Oracle8 リリース 8.0.x への PL/SQL Web Toolkit の インストールに関する障害	146
Oracle Demo 証明書の置換	146
新しい SSL 証明書が必要	146
OPENSSL コマンドの「-CONFIG」オプションの要件	147

mod_ssl の制限事項	147
インストール時の Apache 構成エラー	147
仮想ホストの使用時に Apache を「HTTPDCTL START」で 始動した場合のエラー	148
各国語サポート（NLS 考慮事項）	148
Apache JServ アプリケーションのメモリー増加	149
その他のリリース・ノート	149
mod_ose に関する障害	149
mod_plsql 日付型の引数について	150
mod_rewrite のセキュリティ上の危険性	151
OpenSSL の CA 機能の非サポート	151
第 10 章 Oracle JDBC ドライバ	152
このリリースでの新機能について	152
ドライバのバージョンについて	154
このリリースの内容について	155
NLS 拡張 Zip ファイル（クライアント側のみ）	156
インストールレーションについて	156
環境変数の設定について	157
Windows95 / Windows98 / Windows NT の場合	157
Solaris / Digital Unix の場合	157
HP / UX の場合:	157
AIX の場合:	157
JDBC ドライバを使用する上で役立つヒント	158
Java ストアド・プロシージャ	159
このリリースで判明している問題点および制限事項について	159
第 11 章 JPublisher	162
はじめに	162
JPub のインストールおよび起動方法	162
JPub に関して頻繁に寄せられる質問	164
JPub の新機能	166
JPub に関する既知の不具合と制限事項	166

第 12 章 <i>interMedia Text</i>	168
新しい索引タイプ CTXCAT	168
パラメータ文字列の制限.....	168
列型をサポートする CTXCAT.....	168
CATSEARCH structured_query パラメータの制限.....	169
CTXCAT システム・パラメータ	169
フィルタの改善	170
PL/SQL プロシージャ・フィルタの新しい機能.....	170
ワードリストの改善.....	171
プリフィックス索引付けの新機能.....	171
新しいプロシージャ CTX_THES.HAS_RELATION	174
新しいプロシージャ CTX_DDL.REMOVE_INDEX	174
オプションの FILE/URL セキュリティ	175
URL データストアのユーザー名/パスワードのサポート	175
第 13 章 <i>interMedia Image</i> 、 <i>Audio</i> 、 <i>Video</i> 、 <i>Text</i> および <i>Locator</i>	176
スタート・ガイド	176
Oracle <i>interMedia</i> リリース 8.1.7.....	176
アップグレードの注意点.....	177
Oracle <i>interMedia</i> のインストール	180
インストール決定.....	180
インストール前の作業	180
インストール.....	180
デモの実行	182
Oracle <i>interMedia Image</i> デモ	182
Oracle <i>interMedia Audio</i> デモ	182
Oracle <i>interMedia Video</i> デモ.....	182
Oracle <i>interMedia Locator</i> デモ	183
ダウングレードの注意点.....	183
削除の注意点.....	183
第 12 章 <i>interMedia</i> タイプの削除.....	184
第 13 章 <i>interMedia Java</i> クラスの使用方法.....	184

次期リリースとの互換性のための重要な情報.....	185
互換性初期化ファンクションを使用する場合とその使用方法.....	185
リリース 8.1.7 で修正された問題	187
Oracle <i>interMedia</i> リリース 8.1.7 で修正された問題	187
Oracle <i>interMedia</i> Image リリース 8.1.7 で修正された問題.....	188
Oracle <i>interMedia</i> Audio リリース 8.1.7 で修正された問題	188
Oracle <i>interMedia</i> Video リリース 8.1.7 で修正された問題	188
Oracle <i>interMedia</i> Locator リリース 8.1.7 で修正された問題.....	188
リリース 8.1.7 で追加された機能	189
リリース 8.1.7 <i>interMedia</i> Image、Audio および Video の新機能.....	189
<i>interMedia</i> Image リリース 8.1.7 の新機能	191
<i>interMedia</i> Audio および Video リリース 8.1.7 の新機能.....	191
<i>interMedia</i> Locator リリース 8.1.7 の新機能	191
このリリースの確認されている問題.....	191
Oracle <i>interMedia</i> Image リリース 8.1.7 で確認されている問題.....	191
Oracle <i>interMedia</i> Audio リリース 8.1.7 で確認されている問題	191
Oracle <i>interMedia</i> Video リリース 8.1.7 で確認されている問題	191
Oracle <i>interMedia</i> Locator リリース 8.1.7 で確認されている問題.....	191
廃止予定の旧式のオブジェクト型	192
廃止されたメソッド	192

はじめに

- ※ 米国ではホームページ、ホワイトペーパー等で Oracle8i Release 3 と呼ばれていますが、国内ではこれまでの呼び方との整合性を保つため、「Release 3」とは呼ばずに、「リリース 8.1.7」あるいは「R8.1.7」と呼びます。
- ※ このマニュアルは、Solaris の英語版を翻訳しています。プラットフォームによって記述が異なる場合があります。各プラットフォームのメディアにある英語版リリース・ノートを参照してください。

マニュアルに記載されている名称について

Oracle8i 関連マニュアルは、英語版を翻訳しているため、マニュアル中で参照されている情報には、日本では提供されていないものも含まれます。

- インターネット URL
- マニュアル名
- ソフトウェア名

英語オンライン・ドキュメントの扱いについて

CD 媒体上の英語のドキュメントと同一のドキュメントが日本語で提供されている場合は、日本語版を参照してください。

Oracle JServer の名称について

リリース 8.1.7 より Oracle JServer の名称が変更になりました。新しくは「Oracle8i JVM」となります。Oracle8i 関連マニュアルおよび CD-ROM 製品で Oracle JServer の名称が混在して表示される場合がありますが同じ製品を示しておりますので、注意してください。

最新情報の入手について

日本オラクルでは、インターネット開発者向けのあらゆる技術リソースを、24 時間 365 日提供するコミュニティ・サイト OTN-J (Oracle Technology Network Japan) を運営しています。OTN-J では、最新の技術情報、オンライン・マニュアル、ソフトウェア・コンポーネントなどを、無料で入手できます。

<http://otn.oracle.co.jp/>

第 1 章 RDBMS の Readme ファイル

原典情報：\$ORACLE_HOME/relnotes/README_rdbms.txt ファイル

この章では、次の項目について説明します。

- ユーティリティについて
- エクスポート/インポートについて
- 各国語サポートについて
- レプリケーションについて
- XML Parser について

ユーティリティについて

ユーザーSYS としてデータベースに接続する方法

リリース 8.1.7 以降では、ユーザーSYS としてデータベースに接続する場合、接続文字列で AS SYSDBA も定義する必要があります。たとえば次のようになります。

```
sqlldr ¥'SYS/password AS SYSDBA¥' foo.ctl
```

この例では、接続文字列全体が一重引用符と¥（円マーク）で囲まれています。これは、文字列 AS SYSDBA に空白が含まれており、この場合たいていのオペレーティング・システムは接続文字列全体を引用符で囲むか、何らかの方法でリテラルとしてマークすることが要求されるためです。オペレーティング・システムによっては、コマンドラインの引用符の前にエスケープ文字を付ける必要もあります。この例では、¥（円マーク）がエスケープ文字として使用されています。¥（円マーク）がない場合、SQL*Loader で使用するコマンドライン・パーサーは引用符を認識せず、SQL*Loader をコールする前に引用符を削除します。

使用しているシステムの特許文字および予約語の詳細は、そのオペレーティング・システム固有の Oracle マニュアルを参照してください。

INTEGER と SMALLINT 用の新規の SIGNED および UNSIGNED キーワード

リリース 8.1.6 の『Oracle8i ユーティリティ・ガイド』の 5-59 ページにある INTEGER と SMALLINT データ型に関する内容は、データ型を「符号なし」と誤って定義しています。デフォルトでは、INTEGER と SMALLINT 制御ファイルのデータ型は、符号付き数量とみなされます。SIGNED と UNSIGNED の 2 つの新規キーワードにより、INTEGER と SMALLINT を符号付きあるいは符号なし数量とみなすことができます。

UNSIGNED キーワードを使用すると、次の例に示されているとおり、INTEGER と SMALLINT 制御データ型は符号なし数量となります。

```
(column1 position(1) INTEGER UNSIGNED,  
column2 INTEGER UNSIGNED,  
column3 SMALLINT UNSIGNED,  
...  
)
```

SIGNED キーワードを使用すると、デフォルトの動作になります。

SQL*Loader 入カルーチンの動作の変更

以前のバージョンの SQL*Loader には、レコード境界を誤って認識する可能性のある不具合がありました。このため、バイト固有でないキャラクタ・セットを使用するストリーム・レコード書式データ・ファイルをロードするときに不適切な結果が生成される可能性があります。

キャラクタ・セットは、各バイト・コード化が固有である場合は、バイト固有です。シングルバイト・キャラクタ・セットは、常にバイト固有です。シングルバイト・キャラクタ・セットのコード化がマルチバイト・キャラクタのバイト位置に表示された場合は、マルチバイト・キャラクタ・セットの可変幅はバイト固有ではありません。バイト固有ではないマルチバイト・キャラクタ・セットの例は、utf8 と ja16sjis です。

リリース 8.1.7 では、SQL*Loader の動作に変更が加えられ、この問題が解決されています。この問題が解決したことで、バイト固有ではないキャラクタ・セットに対して SQL*Loader の動作が遅くなる可能性があります。レコード終了記号が通常通りシングル・バイトの 0x0a (ASCII では"¥n") である場合、SQL*Loader の動作は遅くなりません。

CONTINUEIF と新規 PRESERVE キーワード

この項では、リリース 8.1.6 の『Oracle8i ユーティリティ・ガイド』（5-36 ページより）に記載されている、物理レコードからの論理レコードの作成方法に関してわかりやすく説明しています。また、CONTINUEIF THIS および CONTINUEIF NEXT で使用する新規 PRESERVE キーワードについても説明します。

PRESERVE キーワードを使用しない場合、論理レコードが作成されるときにすべての物理レコードから継続フィールドが削除されます。つまり、データ値は中間に余分な文字（継続文字）なしに、レコードを延長することができます。

PRESERVE キーワードを使用する場合、論理レコードが作成されるときに継続フィールドはすべての物理レコード内に保持されます。

次の例は、PRESERVE キーワードがある場合とない場合の CONTINUEIF THIS および CONTINUEIF NEXT の使用方法を示しています。

例 1-1 PRESERVE キーワードなしの CONTINUEIF THIS

14 文字長の物理レコードがあり、ピリオドは空白を表していると仮定します。

```
%aaaaaaaaa....  
%bbbbbbbbb....  
..cccccccc....  
%dddddddddd..  
%eeeeeeeeeee..  
..fffffffffff..
```

この例では、CONTINUEIF THIS 句は PRESERVE キーワードを使用していません。

```
CONTINUEIF THIS (1:2) = '%%'
```

したがって、論理レコードは次のように作成されます。

```
aaaaaaaaa....bbbbbbbbb....cccccccc....  
dddddddddd..eeeeeeeeeee..fffffffffff..
```

1 行目および 2 行目（たとえば、物理レコード 1 の%%）は、論理レコードが作成されるときに物理レコードから削除されることに注意してください。

例 1-2 PRESERVE キーワードありの CONTINUEIF THIS

例 1-1 と同じ物理レコードがあると仮定します。

この例では、CONTINUEIF THIS 句は PRESERVE キーワードを使用しています。

```
CONTINUEIF THIS PRESERVE (1:2) = '%%'
```

したがって、論理レコードは次のように作成されます。

```
%aaaaaaaaa...%bbbbbbbbb...cccccccc...  
%ddddddddd...%eeeeeeeeee...fffffffffff..
```

論理レコードが作成されるときに物理レコードから削除されていないことに注意してください。

例 1-3 PRESERVE キーワードなしの CONTINUEIF NEXT

14 文字長の物理レコードがあり、ピリオドは空白を表していると仮定します。

```
..aaaaaaaaa...  
%bbbbbbbbb...  
%cccccccc...  
..ddddddddd..  
%eeeeeeeeee..  
%fffffffffff..
```

この例では、CONTINUEIF NEXT 句は PRESERVE キーワードを使用していません。

```
CONTINUEIF NEXT (1:2) = '%%'
```

したがって、論理レコードは次のように作成されます（例 1-1 と同じ結果）。

```
aaaaaaaaa...bbbbbbbbb...cccccccc...  
ddddddddd...eeeeeeeeee...fffffffffff..
```

例 1-4 PRESERVE キーワードありの CONTINUEIF NEXT

例 1-3 と同じ物理レコードがあると仮定します。

この例では、CONTINUEIF NEXT 句は PRESERVE キーワードを使用しています。

```
CONTINUEIF NEXT PRESERVE (1:2) = '%%'
```

したがって、論理レコードは次のように作成されます。

```
..aaaaaaaaa...%bbbbbbbbb...%cccccccc...  
..ddddddddd...%eeeeeeeeee...%fffffffffff..
```

SQL*Loader に関するドキュメントの訂正

この項では、リリース 8.1.6 の『Oracle8i ユーティリティ・ガイド』の SQL*Loader に関する章に対する追加、削除および訂正を示します。

BAD および DISCARD ファイルの形式

リリース 8.1.6 の『Oracle8i ユーティリティ・ガイド』のいくつかの箇所では、BAD ファイルと DISCARD ファイルが入力データ・ファイルと同じ形式を持つと誤って記述されています。

SQL*Loader は、常に入力ファイルの同じレコード形式を使用して BAD あるいは DISCARD ファイルにレコードを書き込むとは限りません。書き込まれる内容は、SQL*Loader を実行しているシステムによって異なります。たとえば、Sun Solaris がサポートしている各レコード形式に対して、SQL*Loader は次のように動作します。

- 入力レコードがストリーム・レコード形式を使用している場合、BAD および DISCARD ファイルに書き込まれるレコードは改行で終了します。これは、入力レコードに別の終了記号が指定されている場合も同じです。
- 入力レコードが固定長形式を使用している場合、BAD および DISCARD ファイルに書き込まれるレコードには適切な固定長データが含まれます。
- 入力データが可変長形式を使用している場合、BAD および DISCARD ファイルに書き込まれるレコードは、ストリーム・レコードであるかのように書き込まれます。先頭の件数データおよびレコード終了記号はレコードには含まれません。

Bad ファイルのデフォルトのファイル名

リリース 8.1.6 の『Oracle8i ユーティリティ・ガイド』の 6-3 ページには、Bad ファイルを生成するのに使用される BAD キーワードに関して、次の誤った記述があります。

「ファイル名が指定されていない場合、拡張子.BAD の付いた制御ファイル名が、デフォルトで使用されます。」

正しい内容は、

「ファイル名が指定されていない場合、拡張子.BAD の付いたデータ・ファイル名が、デフォルトで使用されます。」（制御ファイル名ではありません）

空白のみを含む日付フィールドのロード

5-66 ページの日付フィールドおよび 5-82 ページの全空白フィールドに関する内容には、古い情報が含まれています。リリース 8.1.7 では、空白のみを含む日付フィールドは NULL フィールドとしてロードされるので、エラーは発生しません。

フィールドに位置とデリミタの両方が指定されている場合の動作

リリース 8.1.6 の『Oracle8i ユーティリティ・ガイド』の 5-84 ページには、次の不適切な段落があります。

「デリミタ・フィールドのサイズが事前に決まっており、サイズ指定が示す境界内にデリミタが見つからない場合は、エラーが生成されます。たとえば、次のように指定し、

```
loc POSITION(19:31) CHAR TERMINATED BY ","
```

入力レコードの位置 19 と 31 の間にカンマがない場合、レコードは拒否されます。カンマがある場合、そのカンマでフィールドが区切られます。」

正しい内容は、同じフィールドに POSITION および終了記号あるいは囲み文字列が指定されている場合は、POSITION のみが有効です。終了記号あるいは囲み記号は無視されます。

WHEN、NULLIF および DEFAULTIF 句の使用について

この項では、SQL*Loader がどのように WHEN、NULLIF および DEFAULTIF 句を使用するかわかりやすく説明します。この情報はリリース 8.1.6 の『Oracle8i ユーティリティ・ガイド』の 5-80 ページの「列への Null またはゼロの設定」および 5-41 ページの「ロードする行の選択」の情報に代わるものです。

次の情報は、スカラー・フィールドに適用されます。非スカラー・フィールド（列オブジェクト、LOB およびコレクション）については、スカラー・フィールドより複雑なので WHEN、NULLIF および DEFAULTIF 句が別々に処理されます。WHEN、NULLIF または DEFAULTIF 句の結果は、この句でフィールド名あるいは位置を指定するかどうかにより異なることがあります。

WHEN、NULLIF または DEFAULTIF 句でフィールド名を指定する場合、SQL*Loader はこの句をフィールドの評価値と比較します。評価された値は、トリミングされた空白を考慮に入れます。空白トリミングについては『Oracle8i ユーティリティ・ガイド』の空白およびタブのトリミングに関する情報を参照してください。

WHEN、NULLIF または DEFAULTIF 句で位置を指定する場合、SQL*Loader は句をデータ・ファイル内の元の論理レコードと比較します。この場合、論理レコードで空白トリミングは実行されません。

フィールドにトリミングされた空白がある場合や、WHEN、NULLIF または DEFAULTIF 句に空白あるいはタブが含まれていたり、BLANKS キーワードを使用している場合は、結果が異なることがあります。名前指定したフィールドと位置指定した同じフィールドに関して同じ結果が必要な場合は、PRESERVE BLANKS キーワードを使用してください。

PRESERVE BLANKS キーワードは、SQL*Loader に対し、フィールドの値を評価するときに空白をトリミングしないよう指示します。

WHEN、NULLIF または DEFAULTIF 句の結果は、次の手順で説明するように、SQL*Loader で実行する順番にも影響されます。SQL*Loader は次に示す手順を順番に実行しますが、すべてが実行されるとは限りません。フィールドを設定すると、進行中の残りの手順はすべて無視されます。たとえば、手順 5 でフィールドを設定する場合、SQL*Loader は手順 6 には進みません。

1. SQL*Loader は入力レコードの各フィールドの値を評価し、トリミングすべき空白をすべてトリミングします（空白およびタブのトリミングに関する既存のガイドランに従います）。

SQL*Loader はその行にある表の WHEN 句をすべて評価します。

2. レコードが表の WHEN 句を満たしている場合、あるいは WHEN 句が指定されていない場合、SQL*Loader は各フィールドをチェックして NULLIF 句を探します。
3. NULLIF 句が存在する場合、SQL*Loader はそれを評価します。
4. NULLIF 句が満たされている場合、SQL*Loader はそのフィールドを NULL に設定します。
5. NULLIF 句が満たされていなかったり、NULLIF 句がない場合、SQL*Loader はフィールド評価からフィールド長をチェックします。フィールド長がフィールド評価から 0 とわかった場合（たとえば、これは NULL フィールドあるいはトリミングの結果 NULL フィールドになった空白です）、SQL*Loader はこのフィールドを NULL に設定します。この場合、このフィールドに指定された DEFAULTIF 句は評価されません。
6. 指定された NULLIF 句が FALSE であつたり NULLIF 句がない場合、およびフィールド評価からフィールド長が 0 でないとわかった場合、SQL*Loader はフィールドをチェックして DEFAULTIF 句を探します。
7. DEFAULTIF 句が存在する場合、SQL*Loader はそれを評価します。
8. DEFAULTIF 句が満たされており、データ・ファイル中のフィールドが数値フィールドである場合は、そのフィールドは 0 に設定されます。フィールドが数値フィールドでない場合は、NULL に設定されます。次のフィールドは数値フィールドで、DEFAULTIF 句を満たしている場合 0 に設定されます。

- BYTEINT
- SMALLINT
- INTEGER
- FLOAT
- DOUBLE
- ZONED
- (PACKED) DECIMAL

- EXTERNAL NUMERICS (INTEGER, FLOAT, DECIMAL, and ZONED)

DEFAULTIF 句が満たされていなかったり DEFAULTIF 句がない場合、SQL*Loader は手順 1 で評価した値でフィールドを設定します。

SQL*Loader が実行する順番により、期待していない結果が生じることもあります。たとえば、DEFAULTIF 句が 0 ではなく NULL に数値フィールドを設定しているように見えることもあります。

次に示す例で、様々な状況に対する結果をわかりやすく示します。例では、空白はピリオド (.) で示されています。col1 と col2 は、データベース内の VARCHAR2 (5) 列と仮定します。

例 1-5 DEFAULTIF 句が評価されない場合

制御ファイルは次のように指定します。

```
(col1 position (1:5),
 col2 position (6:8) char) integer external defaultif col1 = 'aname')
```

データ・ファイルには次が含まれます。

```
aname...
```

この行の col1 は、「aname」と評価されます。col2 は長さ 0 の NULL と評価されます（これは「...」ですが、後続の空白は位置フィールドのためにトリミングされます）。

SQL*Loader が col2 の最終ロード値を判断するとき、SQL*Loader では WHEN 句も NULLIF 句も検出されません。SQL*Loader はフィールド長をチェックし、フィールド評価から 0 であることがわかります。したがって、SQL*Loader は col2 の最終値を NULL に設定します。DEFAULTIF 句は評価されず、その行は col1 については「aname」、col2 については NULL としてロードされます。

例 1-6 DEFAULTIF 句が評価される場合

制御ファイルは次のように指定します。

```
.
.
.
PRESERVE BLANKS
.
.
.
(col1 position (1:5),
 col2 position (6:8) integer external defaultif col1 = 'aname')
```

データ・ファイルには次が含まれます。

```
aname...
```

この行の col1 は再度「aname」と評価されます。PRESERVE BLANKS が指定されている場合は後続の空白がトリミングされないので、col2 は「...」と評価されます。

SQL*Loader が col2 の最終ロード値を判断するとき、SQL*Loader では WHEN 句も NULLIF 句も検出されません。SQL*Loader はフィールド長をチェックし、フィールド評価からフィールド長が 0 ではなく 3 であることがわかります。

SQL*Loader は DEFAULTIF 句を評価します。col1 が「aname」と同じ「aname」であるため、TRUE と評価します。

col2 は数値フィールドであるため、SQL*Loader は col2 の最終値を「0」に設定します。この行は col1 については「aname」として、col2 については「0」としてロードされます。

例 1-7 DEFAULTIF 句で位置を指定する場合

制御ファイルは次のように指定します。

```
(col1 position (1:5),  
 col2 position (6:8) integer external defaultif (1:5) = BLANKS)
```

データ・ファイルには次が含まれます。。

```
.....123
```

この行の col1 は長さ 0 の NULL と評価されます（これは「.....」ですが、後続の空白はトリミングされます）。col2 は「123」と評価されます。

SQL*Loader が col2 の最終ロード値を設定すると、WHEN 句も NULLIF 句も検出されません。この場合、SQL*Loader はフィールド長をチェックし、フィールド評価からフィールド長が 0 ではなく 3 であることがわかります。

SQL*Loader は DEFAULTIF 句を評価します。SQL*Loader は「.....」である (1:5) を BLANK と比較し、TRUE と評価します。したがって、col2 が数値フィールドであるため (integer external は数値です)、SQL*Loader は col2 の最終値を「0」に設定します。この行は col1 については NULL、col2 については「0」としてロードされます。

例 1-8 DEFAULTIF 句でフィールド名を指定する場合

制御ファイルは次のように指定します。

```
(col1 position (1:5),  
 col2 position (6:8) integer external defaultif col1 = BLANKS)
```

データ・ファイルには次が含まれます。

```
.....123
```

この行の col1 は長さ 0 の NULL と評価されます（これは「.....」ですが、後続の空白はトリミングされます）。col2 は「123」と評価されます。

SQL*Loader が col2 の最終値を判断するとき、SQL*Loader では WHEN 句も NULLIF 句も検出されません。SQL*Loader はフィールド長をチェックし、フィールド評価からフィールド長が 0 ではなく 3 であることがわかります。

次に SQL*Loader は DEFAULTIF 句を評価します。評価の一部として、SQL*Loader は col1 がフィールド評価から NULL であるかどうかチェックします。col1 は NULL であるため、DEFAULTIF 句は FALSE と評価されます。したがって、SQL*Loader では、col2 の最終値を、フィールド評価からの元の値である「123」に設定します。この行は、col1 については NULL、col2 については「123」とロードされます。

エクスポート/インポートについて

増分エクスポート機能のサポートに関して

エクスポートの INCTYPE パラメータで、INCREMENTAL/CUMULATIVE/COMPLETE サポートされなくなり、増分エクスポートの機能はサポートされません。データベースのバックアップを増分バックアップ機能を使用して行うには、Recovery Manager を使用する必要があります。詳細は、『Oracle8i バックアップおよびリカバリ・ガイド』を参照してください。

ユーザーSYSとしてデータベースに接続する方法

リリース 8.1.7 以降では、ユーザーSYSとしてデータベースに接続する場合、接続文字列で AS SYSDBA も定義する必要があります。たとえば次のようになります。

```
exp ¥'SYS/password AS SYSDBA¥'
```

これは、文字列 AS SYSDBA が空白を含んでおり、この場合たいいていのオペレーティング・システムでは接続文字列全体を引用符で囲むか、何らかの方法でリテラルとしてマークすることが要求されるためです。オペレーティング・システムによっては、コマンドラインの引用符の前にエスケープ文字が必要な場合もあります。この例では、¥ (円マーク) がエスケープ文字として使用されています。¥ (円マーク) がない場合、エクスポートとインポートで使用するコマンドライン・パーサーは引用符を認識せず、エクスポートあるいはインポートをコールする前に引用符を削除します。

使用しているシステムの特許文字および予約語の詳細は、そのオペレーティング・システム固有の Oracle ドキュメントを参照してください。

処理制限事項

次の制限事項は、エクスポートおよびインポート・ユーティリティを使用してデータを処理するときに適用されます。

- EJB (Enterprise JavaBeans) を使用して作成された Java クラス、リソースおよびプロシージャは、ダンプ・ファイルには入りません。
- RELY キーワードを使用して変更された制約は、エクスポートされるときに RELY 属性を失います。

エクスポートとインポートに関するドキュメントの訂正

この項では、リリース 8.1.6 の『Oracle8i ユーティリティ・ガイド』のエクスポートとインポートに関する章に対する追加、削除および訂正を示します。

表名の大 / 小文字区別の保持

Oracle8i リリース 8.1.6 の『Oracle8i ユーティリティ・ガイド』の 1-26 ページと 2-28 ページにある文章では、名前を引用符で囲むことで表名の大 / 小文字区別を保持できるとしています。

さらに、オペレーティング・システムによって、コマンドラインの引用符の前にエスケープ文字を付ける必要があることに注意してください。エクスポートおよびインポートのモード別の大 / 小文字区別の保持方法の例を次に示します。

- コマンドライン・モード:

```
tables='¥"Emp¥"'
```

- インタラクティブ・モード:

```
Table(T) to be exported: "Emp"
```

- パラメータ・ファイル・モード:

```
tables='"Emp"'
```

各国語サポートについて

Unicode 3.0 のサポート

Unicode 3.0 は現在、Oracle の文字セット UTF8 と UTFE でサポートされています。これは ISO/IEC 10646-1 第 2 版と同期しており、追加の 10,307 コード・ポイントを含み、一般カテゴリおよびケーシング機能について更新しています。他の Unicode 3.0 機能は、データベースの将来のリリースで提供される予定です。

データベース・キャラクタ・セット移行ツール

Oracle8i リリース 8.1.7 には、キャラクタ・セット・スキャナ (csscan) と呼ばれる新規 NLS ユーティリティが付属しています。このユーティリティは、インポートおよびエクスポート・ツールと同様に、インストール時に \$ORACLE_HOME/bin ディレクトリに作成されます。このツールにより、ユーザーは既存のデータベースを新規キャラクタ・セットに移行する場合の影響を評価できます。このツールは、データ・ディレクトリを含むデータベース内のデータをスキャンし、現行のデータを分析して、新規データベース・キャラクタ・セットに移行したとき変換されるデータを選択します。

詳細は、『Oracle8i リリース 8.1.7 追加情報』を参照してください。

中国語の簡字体と繁体字の変換

この機能は中国語の簡字体と繁体字の変換をサポートするもので、データベースに格納されている中国語データ（簡字体あるいは繁体字）のタイプに関係なく、繁体字のクライアントで簡字体のデータを表示したり、簡字体のクライアントで繁体字のデータを表示したりできます。これは、各中国語の文字を相当する簡字体および繁体字にマッピングすることで実行されます。ただし、変換マッピングはすべての中国語文字について可能ではありません。このため、置換文字がまだ存在する場合があります。

以前のリリースでは、このマッピング機能がなかったため、Unicode で別々にコード化される中国語の簡字体と繁体字文字書式は、「?」文字にマッピングされていました。この新しい変換機能により、情報の損失が最小限に抑えられます。

変換は文字ごとに実行されることに注意してください。文字変換中にコンテキストが失われることがあります。たとえば、1 文字の簡字体文字が 2 文字以上の繁体字中国語文字に対応していることがあります。通常はコンテキストによって、そのうちの 1 文字のみが正しい文字です。Oracle では、1 文字の簡字体中国語文字から 1 文字の繁体字中国語文字、あるいはその反対というように一対一のマッピングしか提供しないため、単語の意味がなくなる場合があります。

改善された ALTER DATABASE CHARACTER SET のサポート

以前のリリースでは、ALTER DATABASE CHARACTER SET コマンドは、US7ASCII キャラクタ・セットの厳密な ASCII スーパーセットへのアップグレードのみサポートしていました。リリース 8.1.7 では、これがすべてのサブセットとスーパーセット・キャラクタ・セットに対応するよう拡張されました。サブセットとスーパーセット・キャラクタ・セットのリストについての詳細は、『Oracle8i リリース 8.1.7 追加情報』を参照してください。

AL24UTFFSS キャラクタ・セットの非サポート

オラクル社は、次のメジャー・リリースでデータベース・キャラクタ・セットの AL24UTFFSS キャラクタ・セットに対するサポートを廃止する予定です。AL24UTFFSS は、Oracle7 で UTF-8 コード体系をサポートする Unicode キャラクタ・セットとして導入されました。UTF-8 コード体系は Unicode 標準 1.1 に基づいていますが、現在は廃止されています。Oracle8 と Oracle8i には、データベース・キャラクタ・セット UTF8 がありますが、これは最新の Unicode3.0 に基づいた Unicode 拡張が含まれています。

オラクル社は、Oracle8 と Oracle8i では UTF8 を Unicode キャラクタ・セットとして使用することをお勧めします。AL24UTFFSS への移行方法は、UTF8 にアップグレードすることです。新規データベース・キャラクタ・セットへのすべての移行において、オラクル社は既存のデータベース・キャラクタ・セットを UTF8 に移行する前に、キャラクタ・セット・スキャナ(csscan)を使用してデータ分析をすることをお勧めします。

NCHAR のサポート

Oracle8 リリース 8.0 では、元のデータベース・キャラクタ・セットに加えて 2 番目の代替キャラクタ・セットを可能にする各国語キャラクタ (NCHAR) ・データ型が導入されました。NCHAR は、アジアの文字データ処理をより高速で実行するために導入されたいくつかの特別な固定幅のアジア・キャラクタ・セットをサポートしています。固定幅キャラクタ・セットの例は、次のとおりです。

- JA16EUCFIXED
- JA16SJISFIXED
- ZHT32EUCFIXED

Oracle8i Server の次のメジャー・リリースから、NCHAR、NVARCHAR および NCLOB データ型は、Unicode キャラクタ・セットのコード化のみに制限されます。アジア・キャラクタ・セット (JA16SJISFIXED など) を含む、NCHAR データ型で現在利用できる他の Oracle8i Server キャラクタ・セットはサポートされません。既存の NCHAR、NVARCHAR および NCLOB 列への移行方法は、エクスポートおよびインポート経由となります。

制限付き ALTER DATABASE CHARACTER SET コマンドのサポート (CLOB および NCLOB)

Oracle8i リリース 8.1.5、8.1.6 および 8.1.7 では、マルチバイト・データベース・キャラクタ・セットの CLOB と NCLOB データ型は常に UCS-2 (2 バイトの固定幅 Unicode) として格納されます。シングルのバイト・データベース・キャラクタ・セットについては、CLOB と NCLOB はそれぞれデータベース・キャラクタ・セットおよび各国語キャラクタとして格納されます。データベースまたは各国語キャラクタ・セットが、ALTER DATABASE CHARACTER SET あるいは DATABASE NATIONAL CHARACTER SET コマンド経由で、シングルのバイトからマルチバイトに移行した場合、CLOB または NCLOB 列はオリジナルのデータベース・キャラクタ・セットのコード化に残るため、CLOB 列に不整合なデータが導入されます。

CLOB および NCLOB 列への移行方法は、まず CLOB および NCLOB 列を含む表をエクスポートして削除し、次に ALTER DATABASE CHARACTER SET または ALTER DATABASE NATIONAL CHARACTER SET (あるいはその両方)、その後インポートとなります。

レプリケーションについて

この項目については、『Oracle8i リリース 8.1.7 追加情報』を参照してください。

プロシージャ

制限: レプリケーション環境で使用されるスナップショットをリフレッシュする場合には、
DBMS_SNAPSHOT.REFRESH_ALL_MVIEWS および
DBMS_SNAPSHOT.REFRESH_DEPENDENT プロシージャを使用しないでください。

処置: レプリケーション環境でのスナップショットをリフレッシュするには、
DBMS_REFRESH.REFRESH または DBMS_SNAPSHOT.REFRESH プロシージャを使用してください。

SSL 接続

Java RepAPI を使用する SSL 接続は、リリース 8.1.7 ではサポートされていません。

XML Parser について

XML Parser for PL/SQL と Class Generator for Java は、リリース 8.1.5 および 8.1.6 からのアップグレード時にリリース 8.1.7 データベースにロードされません。

XML Parser for PL/SQL を使用するためには、\$ORACLE_HOME/lib/xmlplsql.jar をロードし、\$ORACLE_HOME/xdk/plsql/parser/bin/load.sql を実行する必要があります。

Class Generator for Java を使用するには、\$ORACLE_HOME/lib/classgen.jar をロードする必要があります。

第 2 章 Oracle Advanced Security

原典情報：\$ORACLE_HOME/relnotes/README_aso.txt ファイル

Oracle Advanced Security の概要

Oracle Advanced Security リリース 8.1.7 は、Oracle8i のセキュリティ・サービスをまとめたものです。すべてのプロトコルによる Oracle8i への接続を保護し、公開鍵インフラストラクチャ (PKI) を統合します。Oracle Advanced Security により、データベースへのすべてのネットワーク・プロトコルに対して、データの暗号化と整合性が提供されます。これには、システム固有の暗号化を伴った Net8、Net8/SSL、IIOP/SSL および JDBC Thin クライアント用の Java ベースの暗号化などが含まれます。また、このリリースでは、サード・パーティによる認証、認可およびシングル・サインオン・サービスが統合されています。Secure Sockets Layer (SSL) および X.509 バージョン 3 の証明書 (サード・パーティの証明書サーバーにより提供される) を含む公開鍵の解決をサポートし、関連ツール、Oracle Wallet Manager 2.1 および Oracle Enterprise Login Assistant 1.1 をまとめています。さらに、SSL ベースのシングル・サインオンに対応し、証明書ベースのサーバー間認証とデータベース・リンクに対応しています。このリリースの Oracle Advanced Security は、Entrust PKI を統合しています。

Oracle Advanced Security は、LDAP バージョン 3 準拠のディレクトリ・サーバーを使用してユーザー管理を集中化し、Oracle Enterprise Security Manager 2.0 を採り入れてエンタープライズ・ユーザーとエンタープライズ・ロールを管理します。また、ユーザーおよび認可の保存のために、Oracle Internet Directory リリース 2.1.1 の使用を制限し、ロール管理のために Microsoft Active Directory を統合しています。

このリリースの Oracle Advanced Security は、一般的なインストール時に自動的にインストールされます。使用するには、適切なライセンスを購入し、一般的なインストール時に自動的にインストールするか、あるいは Custom インストールを使用して手動でインストールした後、サーバーとクライアント上で構成します。すべてのコンポーネントは、一般的なインストールでインストールされます。ただし、CyberSafe、DCE および Entrust (サーバー上) とエンタープライズ・ユーザーのセキュリティ機能 (クライアント上) は、Custom インストールでインストールします。Oracle Advanced Security はデフォルトでインストールされますが、別売品のため、使用時に購入手続きが必要です。このライセンス要件は、JavaBeans (IIOP/SSL 上の EJB) または Database Enterprise Users (Net8/SSL) と組み合わせてセキュリティ機能を使用する顧客にも適用されます。唯一の例外は RDBMS への HTTPS (HTTP/SSL) 接続で、この接続には Oracle Advanced Security のライセンスが不要です。

このリリースでの変更内容

この項の内容は、次のとおりです。

- このリリースでの新機能
- データの暗号化と整合性
- 外部認証とシングル・サインオン
- ENTRUST サポート
- 公開鍵インフラストラクチャ
- 暗号化と認証
- Secure Sockets Layer
- Oracle Wallet Manager
- Oracle Enterprise Security Manager 2.0
- Java SSL
- インストール
- 旧リリースの問題点に対する拡張機能

このリリースでの新機能

- 強力な暗号化を備えた国際的に使用可能なシングル・バージョン
- ネイティブ Net8 をサポートする 3DES と RSA RC4_256
- ネイティブ Net8 をサポートする SHA-1
- Net8 Assistant における暗号化アルゴリズムの順序
- JDBC Thin クライアント用の強力な暗号化
- Entrust の統合
- エンタープライズ・ユーザーの管理とシングル・サインオンの改善

このリリースには、強力な暗号化が組み込まれた Oracle Advanced Security の国際使用バージョンが含まれています。以前のバージョンは、国内版または輸出版として個別に出荷されていました。このシングル・バージョンでは、最大 256 ビットの RC4 および 3DES 暗号化による強力な暗号化が、ネイティブ Net8 と SSL の 3DES に対して提供されます。リリース 8.1.7 以前のバージョンを使用しているユーザーは、使用しているバージョンの国内版にアップグレードできます。

データの暗号化と整合性

このリリースの Oracle Advanced Security は、256 ビット、128 ビット、56 ビットおよび 40 ビットの各キーを持つ RSA RC4、DES と DES40、および 3 キー (168 ビット) または 2 キー (112 ビット) の Triple-DES を使用してネイティブ Net8 を暗号化します。さらに、Net8 に対して MD5 と SHA-1 のアルゴリズムを使用し、データの整合性をチェックします。

アルゴリズムとキーの長さは、Oracle Advanced Security の暗号化と整合性の構成時に、ユーザーの優先順位に従って選択します。ネゴシエーションは、`sqlnet.ora` では左から右に、Net8 Assistant では上から下の順序に従います。

Oracle Advanced Security リリース 8.1.6 は、ネイティブ Net8 暗号化コンポーネントの FIPS (Federal Information Processing Standard) 140-1 Level 2 の検証を完了しています。FIPS 140-1 の必須構成パラメータについては、『Oracle Advanced Security 管理者ガイド』の付録 D を参照してください。Oracle Advanced Security を FIPS 140-1 Level 2 の検証済み構成で使用するユーザーは、それぞれのマシンのカバーに暗号化済み証明シールを貼る必要があります。これによってカバーの取外しが検出できます。

Oracle Advanced Security は、256 ビット、128 ビット、56 ビットおよび 40 ビットの各キーを持つ RSA RC4 と、DES および DES40 を使用して JDBC Thin クライアントの RDBMS への接続を暗号化します。また、MD5 を使用してデータの整合性をチェックします。

Oracle Advanced Security の SSL 機能は、Triple-DES、128 ビット、56 ビットおよび 40 ビットの各キーを持つ RSA RC4 と、DES および DES40 を使用して Cipher Suites をサポートしています。MD5 と SHA-1 のアルゴリズムは、SSL Cipher Suites に含まれています。

外部認証とシングル・サインオン

Oracle Advanced Security は、RADIUS 準拠サーバーと認証デバイスをサポートしています。RADIUS に対して Java ベースのクライアント・インタフェースを使用するには、`LD_LIBRARY_PATH` を `$ORACLE_HOME/JRE/lib/sparc/native_threads` に設定する必要があります。RADIUS を使用するには、ネイティブ・スレッドを使用する必要があります。`THREADS_FLAG` 変数を Java ランタイム環境内で「native」に設定します。

このリリースの RSA ACE/Server とトークンは、SecurID アダプタの他にも RADIUS アダプタを使用して、Oracle ユーザーを認証できます。ACE/Server に対して RADIUS プラグインを使用すると、ACE/Server は RADIUS サーバーおよび認証サーバーとして動作します。Oracle Advanced Security が、RADIUS と ACE/Server の統合を採用して両モードをサポートするのは、このリリースが最後です。機能の面では変化はありません。

このリリースの Oracle Advanced Security は、Kerberos V5 リリース 1.0 の認証をデフォルトの構成モード（現在サポートされているのは、des-cbc-crc 暗号化モードのみ）でサポートしています。Kerberos V5 リリース 1.0 は、Massachusetts Institute of Technology (MIT) の Web サイト <http://web.mit.edu/kerberos/www> から取得できます。このリリースの Oracle Advanced Security は、MIT の新しい Kerberos 構成ファイル形式をサポートしています。

さらに、Solaris 2.6 上の Transarc の DCE/DFS 2.0 をサポートしています。DCE を使用するには、DCE 統合を Custom インストールでインストールする必要があります。

このリリースの Oracle Advanced Security は、CyberSafeTrustBroker Client Suite v1.2 をサポートしています。CyberSafe を Oracle Advanced Security と併用するには、CyberSafe サポート機能を Custom インストールでインストールする必要があります。複数の製品を併用する場合は、CyberSafe のインストール時に CyberSafe GSS Runtime Library v1.1 をインストールする必要があります。

Oracle Advanced Security が、バイオメトリックをサポートする RADIUS を採用して Identix アダプタをサポートするのは、このリリースが最後です。

外部認証を必須にし、ユーザー名およびパスワード認証を無効にするには、sqlnet.ora のパラメータを SQLNET.AUTHENTICATION_REQUIRED=TRUE に設定します。デフォルトは FALSE です。

ENTRUST サポート

Net8 SSL アダプタの Entrust 対応バージョンを構成するには、最初に EntrustIPSEC Negotiator Toolkit バージョン 5.0 を Oracle クライアントとサーバーを実行するすべてのコンピュータにインストールする必要があります。

警告： Entrust 統合機能がサポートする Net8 SSL Adapter の構成は、Entrust プロファイルとのみ動作します。したがって、Oracle Installation を Oracle Wallets で動作するように構成している場合は、Entrust 統合機能をインストールしないでください。

公開鍵インフラストラクチャ

今回のリリースは、公開鍵インフラストラクチャ（PKI）でネットワークを保護する Secure Sockets Layer（SSL）プロトコルをサポートしています。SSL プロトコルによって、Oracle は、X.509 バージョン 3 の証明書を相互に認証して処理します。SSL は、Cipher Suites による暗号化とデータの整合性を提供します。この Cipher Suites は、認証、暗号化およびデータの整合性の型のセットです。SSL は、X.509 バージョン 3 の証明書を RSA 非対称型キー認証に対して使用するだけでなく、非証明書ベースのネゴシエーションに対して Diffie-Hellman Anonymous Cipher Suites を提供します。

Oracle Wallet Manager は Wallet を作成し、証明書要求を発行して公開鍵資格証明を管理します。Oracle Wallet Manager を起動するには、`$ORACLE_HOME/bin/owm` を実行します。Oracle Wallet Manager は、デフォルトの Wallet ディレクトリを検索して Wallet を作成またはオープンします。たとえば、Solaris では、`/etc/ORACLE/WALLETS/<ユーザーID>`を検索します。デフォルトのディレクトリの作成は、必要なシステム権限を所有しているシステム管理者が行う必要があります。セキュリティ上の理由から、ユーザーID のディレクトリへのアクセスは、それぞれのユーザーに限定する必要があります。

Enterprise Login Assistant は、`$ORACLE_HOME/bin/elogin` を実行して起動できます。このツールにも、ログイン用のデフォルトの Wallet ディレクトリが必要です。

暗号化と認証

判明している不具合とその回避策

このリリースの多重化プーリングと接続プーリングの NS 機能は、SSL トランスポートが使用中のときは、動作しません。

Identix アダプタは、指紋データベースが Windows NT サーバーの場合、次の `sqlnet.ora` パラメータ（指紋データベース・サーバー用）を設定する必要があります。
`SQLNET.AUTHENTICATION_SERVICES=(NTS,NONE)`

Radius アダプタが CHAP（要求/応答）モード用に構成されていると、Oracle クライアント（`sqlplus` または `svrmgrl`）がエラーとなる既知の問題があります。この問題を回避するには、環境変数 `LD_LIBRARY_PATH` に `$ORACLE_HOME/JRE/lib/sparc/native_threads` を挿入するように設定します。

Secure Sockets Layer

判明している不具合とその回避策

SSL アダプタの場合、SSL_VERSION、SSL_CIPHER_SUITES、SSL_CLIENT_AUTHENTICATION などの sqlnet.ora パラメータを TNS 別名の一部として動的に指定しても、データベース・サーバーにリダイレクトされた接続にはまったく影響ありません。

Oracle Wallet Manager

判明している不具合とその回避策

Oracle Wallet Manager のスプラッシュ・スクリーンが適切なフォントを表示しません。
\$JREHOME/lib/font.properties を変更して「SansSerif.alias.helvetica=sansserif」の別名を挿入して回避します。(Bug#1093958)

英語以外のバージョンの Oracle Wallet のツールでは、ヘルプが英語のみで表示されます。これを回避するには、Wallet ツールのスクリプト (UNIX 上では owm、NT 上では owm.cl) を編集し、環境変数 CLASSPATH の先頭に適切な言語のヘルプ・ファイル・ディレクトリを追加します。これは、Enterprise Login Assistant にも該当します。(Bug#1098469)

一部のマルチバイト・キャラクタ・セットを使用すると、証明書要求の作成に失敗します。(Bug#1077099)

証明書要求情報を入力するためのモーダル・ダイアログ・ボックスがポップアップするときなどに、Oracle Wallet Manager のオンライン・ヘルプが応答しなくなります。モーダル・ダイアログ・ボックスをクローズすると、オンライン・ヘルプが応答するようになります。(Bug#1114710)

Oracle Enterprise Security Manager 2.0

判明している不具合とその回避策

- データベースを、すでに別のエンタープライズ・ドメインのメンバーであるエンタープライズ・ドメインに配置しようとする、エラー・メッセージ「General Net8 LDAP Error.」が表示されます。このリリースの Enterprise Security Manager (ESM) では、データベースを複数のエンタープライズ・ドメインに追加できません。このエラー・メッセージが表示された場合は、そのエンタープライズ・ドメインのデータベース・プロパティ・ページを終了し、再入力した後でエンタープライズ・ドメインに対するデータベースの追加または削除を実行します。(Bug#1064954)
- データベースをエンタープライズ・ドメインから削除すると、エンタープライズ・ドメイン内のエンタープライズ・ロールに、そのデータベースのグローバル・ロールが残っている場合があります。(Bug#736159)
- ディレクトリに新しいオブジェクトを作成しようとして、操作に失敗すると（たとえば、無効なベース DN でエンタープライズ・ユーザーを作成しようとする）、ESM は、適切なエラー・メッセージをレポートします。しかし、同じ「Create Object」ダイアログを使用してオブジェクトを再度作成すると、このダイアログは、オブジェクトを作成せずに終了します。操作を再実行するには、「Create Object」ダイアログを再起動する必要があります。(Bug#1041890)
- エンタープライズ・ドメインの「Database Schema Mapping」プロパティ・ページを使用して、エンタープライズ・ドメインに対する新しいデータベース・スキーマ・マッピング・エントリを入力する場合、ESM によって後続のデータベース・スキーマ・マッピング・エントリを入力するための新しい行が、スキーマ割当てグリッド内に自動的に用意されることはありません。新しい行を取得するには、プロパティ・ページを終了し、再入力する必要があります。(Bug#1019924)
- 「Browse Directory」ダイアログを使用する場合、ESM は、ディレクトリ・サーバー内のオブジェクトを検索するのに必要なディレクトリ・ネーミング・コンテキストの初期値を選択します。ディレクトリ・サーバーにネーミング・コンテキストが構成されていない場合、ESM は、そのディレクトリのデフォルト値として、「dc=com」という最も一般的なトップレベルのディレクトリ・エントリに戻ります。ディレクトリが、「dc=com」をサポートしていない場合は、適切なディレクトリ・ネーミング・コンテキストを手動で入力しない限り、検索に失敗します。(Bug#1098590)
- 「netadmins」グループのユーザーには、データベース・オブジェクトの下にマッピング・オブジェクトを追加できるアクセス権限があります。この操作は通常、データベース管理者が行います。

Java SSL

判明している不具合とその回避策

JNDI と JDK 1.2 を使用した LDAP/SSL 接続では、コア・ダンプが発生します。この不具合は、複数のスレッドが同じ SSL 接続を共有すると、Java SSL がスレッド・セーフでなくなるというものです。回避するには、複数スレッドが同じ SSL 接続上で同時に読み込みと書き込みを実行する場合は、アプリケーションが読み込みと書き込みを同期化する必要があります。

(Bug#1328188)

インストール

Oracle Advanced Security をインストールすると、「naect.o.bak」、「naect.o.bak」および「naedhs.o.bak」という 3 つの「.bak」ファイルが作成されます。これらのファイルは \$ORACLE_HOME/lib にありますが、削除しないでください。Oracle Advanced Security の削除時に実行可能ファイルを再リンクするのに必要です。

Oracle Advanced Security をクライアント専用マシン（データベースのないマシン）で使用する場合、インストールの開始前に環境変数 TWO_TASK を設定することが必須要件です。この TWO_TASK 変数は、サーバー・マシン上のデータベースを示す別名をポイントします。環境変数 TWO_TASK によって、Oracle Advanced Security がクライアント専用モードでインストールされます。環境変数 TWO_TASK の詳細は、『Oracle8i 管理者ガイド』を参照してください。

旧リリースの問題点に対する拡張機能

- JDBC Thin クライアントは、最大 256 ビットの RC4 と 56 ビットの DES 暗号化によって安全に接続できます。
- このリリースの Oracle Advanced Security では、1 つの版にすべての暗号化アルゴリズムが含まれているため、輸出版と国内版における暗号化または Cipher Suites に関する問題はすべて解決されました。
- Net8 のレイヤーが SQLNET 構成パラメータ情報を伝播できないために、接続がリダイレクトできないという問題は解決されました。(Bug#1113580)
Identix サポートや SSL サポートに影響を与えることは今後ありません。
- 同じノード上で 2 つ目の DCE リスナーを開始できないという不具合は解決されました。(Bug#834788)
- 無名のデータベース・リンクが Kerberos 認証で動作するようになりました。(Bug#863864)

- 環境変数 `TWO_TASK` の設定とリモート OS 認証の使用を同時に実行できないという不具合は解決されました。 (Bug#1139334)
- データベースが SSL 上にリスナーを持つ MTS ディスパッチャを登録できないという不具合は解決されました。リスナー付きの MTS ディスパッチャの登録は、SSL で適切に機能します。
- Windows 上の Oracle Internet Directory の場合に、Database Configuration Assistant (DBCA) が原因で、ディレクトリへのデータベース登録に失敗するという不具合は解決されました。 (Bug#1266289)
- ユーザーごとに個別の `sqlnet.ora` ファイルが必要になるという不具合は解決されました。複数のユーザーが 1 つのクライアント・マシンで 1 つの `sqlnet.ora` ファイルを共有できるようになりました。
- リスナーのリダイレクトによってエンタープライズ・ユーザーのセキュリティが阻害され、ユーザーが `sysdba` で接続することが妨害されるという不具合は解決されました。 (Bug#1113580)
- Oracle Enterprise Security Manager (ESM) は、起動時に初期ログイン・ボックスを表示するようになり、適切な権限でログインとログアウトを繰り返す必要がなくなりました。ただし、Microsoft Active Directory を使用している場合、ESM は、依然としてネイティブ認証を使用したディレクトリへの自動ログインを起動時に実行します。 (Bug#1098580)
- ESM が `sqlnet.ora` ファイルの読み込みに失敗すると、ネーミングや ASO 暗号化などの Net8 機能が ESM の構成で使用できないという不具合は解決されました。 (Bug#1368685)
- ESM を使用したドメイン管理者の削除によって、そのドメイン内の既存のデータベース ACL がすべて削除されるという不具合は解決されました。 (Bug#1298185)

第 3 章 Net8 について

原典情報：\$ORACLE_HOME/relnotes/README_net8.txt ファイル

Net8 の概要

このでは、次のマニュアルの補足情報を記載しています。

- J00959-01 - 『Oracle8i Net8 管理者ガイド リリース 8.1』
- J02329-01 - 『Oracle8i リリース 8.1.7 追加情報』

Net8 は Oracle のネットワーク製品ファミリーの基盤であり、エンタープライズ・ワイドのデータ・アクセスに最も包括的なソリューションを提供し、異種間分散コンピューティング環境を実現します。Net8 により、クライアント / サーバー間およびサーバー / サーバー間の通信がネットワークの種類を問わず可能になります。

Net8 は、次の要素に利点を提供します。

- ネットワークの拡張性およびパフォーマンス
- ネットワークのロード・バランス
- グラフィカル構成ツールを使用した統合管理
- 構成の柔軟性
- プロトコルの相互依存
- 包括的なプラットフォーム・サポート
- 単一コンピュータでの複数プロトコルのサポート
- 広範囲をカバーするトレースと診断

インストール

Oracle8i リリース 8.1.7 データベースには、Net8 リリース 8.1.7 リスナーが必要です。旧バージョンのリスナーは、Oracle8i リリース 8.1.7 データベースとの併用がサポートされていません。ただし、旧バージョンの Oracle8 データベースは Net8 リリース 8.1.7 リスナーと同時に使用できます。

クライアントおよびサーバーのインストールの詳細は、各オペレーティング・システムのインストール・ガイドを参照してください。

廃止された NET8 パラメータ

オラクル社では、今後、次のパラメータを使用しないことをお勧めします。

names.ora パラメータ

次の 3 つのパラメータは、認識されなくなります。ドメインおよびトポロジーのチェック・ポイント・ファイルのパラメータは、NAMES.REGION_CHECKPOINT_FILE パラメータに置き換えられています。

- NAMES.USE_PLUG_AND_PLAY
- NAMES.DOMAIN_CHECKPOINT_FILE
- NAMES.TOPOLOGY_CHECKPOINT_FILE

マルチスレッド・サーバー (MTS) ・パラメータ

次の 2 つのパラメータは、サポートされません。

- MTS_RATE_LOG_SIZE
- MTS_RATE_SCALE

次の 3 つのパラメータは、今後のリリースでサポートされなくなる可能性があります。

- MTS_SERVICE
- MTS_LISTENER_ADDRESS
- MTS_MULTIPLE_LISTENERS

ただし、MTS_DISPATCHERS パラメータの SERVICE 属性および LISTENER 属性は、MTS ディスパッチャのサービス名およびリスナーの設定に使用してください。

Net8 の既知の問題

ディレクトリ・ネーミング

- ディレクトリ・ネーミングは、次のディレクトリをサポートします。
 - Oracle Internet Directory
 - Microsoft 社の Active Directory
 - Novell Directory Services (NDS) V8
- Net8 ディレクトリ・ネーミングは、JDBC Thin ドライバではサポートされません。次の URL で更新パッチを入手してください。
<http://www.oracle.com/java/jdbc/>

Net8 の制限事項

- Net8 リリース 8.0 とは異なり、リスナーの構成時にホスト・ネーミングの使用をデフォルトで設定することはできません。ホスト・ネーミングを使用するようにリスナーを構成する方法については、『Oracle8i Net8 管理者ガイド リリース 8.1』の第 6 章「ネーミング・メソッドの構成」を参照してください。
- 既存の tnsnames.ora ファイルは、sqlnet.ora ファイル内の一定のエントリ (NAMES.DEFAULT_DOMAIN パラメータなど) に依存している可能性があります。tnsnames.ora ファイルのエントリが sqlnet.ora ファイルのエントリに依存している場合は、既存の sqlnet.ora ファイルを使用してください。
- Trace Assistant は、トレース・ファイル内のすべての Oracle8 メッセージおよび Oracle8i メッセージを解析するわけではありません。
- TRCROUTE ユーティリティ (クライアントからサーバーへの接続ルートを管理者が特定する際に使用するユーティリティ) は、サーバーに接続するクライアントが 2 つ以上ある場合 (Oracle Connection Manager を使用する場合) には無効です。
- Net8 OPEN (Net8 とのインタフェースとなるアプリケーションを作成する際に開発者が使用) は、このリリースでは出荷されません。
- Oracle Connection Manager は、LU6.2 プロトコルをサポートしていません。
- Solaris では SPX プロトコル・サポートをインストールできますが、これはオラクル社が認証するインストールではありません。このプロトコル・サポートは、リリース 8.0 で使用されたライブラリと同じものを使用します。
- Solaris では、LU6.2 プロトコルが無効です。

- コマンド SET LOG_FILE および SET TRC_FILE を使用すると、リスナー・プログラムが書き込みを行うログ・ファイルおよびトレース・ファイルを、リスナー・プログラムの実行中に動的に変更できます。リスナー・プログラムは、ログ情報およびトレース情報を、Oracle 所有者が書き込み可能な任意のオペレーティング・システムのファイル（アラート・ファイルやデータベース・ファイルなど）に付加または上書き（あるいは両方）するように構成できます。そのため、Oracle データベースを破損し、不正なコードをオペレーティング・システムに混入させる可能性があります。

新規パラメータ ADMIN_RESTRICTIONS_listener_name が listener.ora に導入されたため、データベース管理者は、リスナーのランタイム管理を制限できるようになりました。ADMIN_RESTRICTIONS_listener_name=on と設定すると、listener.ora のパラメータのランタイム変更が使用禁止になるため、変更が容易であるという listener.ora パラメータの特性が活用できなくなります。具体的には、パラメータを変更する SET コマンドをリスナーが拒否するため、SET コマンドを発行しようとするエラー・メッセージが生成されます。

listener.ora 内のパラメータ (ADMIN_RESTRICTIONS_listener_name を含む) を変更するには、listener.ora ファイルを手動で編集し、Listener Control ユーティリティの RELOAD コマンドを使用して、それらのパラメータを再ロードする必要があります。この方法で実行された新しい変更は、リスナーを明示的に停止および再起動することなく有効になります。listener.ora の編集には、Oracle アカウント所有者の保護付きディレクトリおよびファイルにオペレーティング・システムがアクセスできる必要があります。Oracle アカウント所有者の保護付きディレクトリおよびファイルにアクセス制御権限を設定して、それらのディレクトリおよびファイルをオペレーション・システム内で保護してください。

現行のカスタマ環境および下位互換性の維持を目的としてリスナーをインストールする場合には、ADMIN_RESTRICTIONS_listener_name=off をデフォルト値とします。リスナー・プログラムのランタイム動作またはこの操作モードでの SET コマンドの構文に変更はありません。

この操作モードを実施する場合、オラクル社では、リスナーのパスワードを設定することをお勧めします。(Bug#1361722)

- このリリースでは、Oracle Advanced Security の暗号化アルゴリズムおよび暗号化キー長のすべてを、各国の Oracle カスタマに公開しています。

Net8 Assistant および Net8 Configuration Assistant の制限事項

- Net8 Configuration Assistant を使用してディレクトリ・アクセスを構成するときには、マルチバイト・キャラクタを使用する管理コンテキストを指定できません。
(Bug#1102675)
- Net8 Assistant は、tnsnames.ora ファイルまたは LDAP 準拠のディレクトリ・サーバーに格納されているすべてのエントリを表示するわけではありません。表示するのは、ネット・サービス名のエントリのみです。

ネット・サービス名には、完全な接続記述子（記述、アドレスおよび接続データ情報）が含まれている必要があります。

次の例は、「sales」というネット・サービス名です。「sales」には、リスナー・アドレス情報および「sales.us.acme.com」というデータベースへの接続情報が含まれています。この情報が Net8 Assistant に表示されます。

```
sales=
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=sales.us.acme.com))
```

Oracle Database Configuration Assistant によって作成される一部のエントリは、一定のアドレス情報のみを含んでいます。次の例は、ネット・サービス名ではなく名前付きアドレスのエントリです。「sales」の名前付きアドレスには、リスナー・アドレス情報のみが含まれています。この名前付きアドレスは Net8 Assistant には表示されませんが、「sales」別名をリスナー・アドレスのリストに対して解決するために使用できます。名前付きアドレスは、おもに非デフォルト・ローカル・リスナーまたはリモート・リスナーの別名に使用します。

```
sales=
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales1-server) (PORT=1521))
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales2-server) (PORT=1521))
```

関連項目：非デフォルト・アドレスを使用するリスナーの構成については、『Oracle8i Net8 管理者ガイド リリース 8.1』の第7章「リスナーの構成」を参照してください。

- ディレクトリに格納されている、データベース・サービスまたはネット・サービスの名前付きエントリの接続タイプは、Net8 Assistant では変更できません。接続タイプは、常に Database Default と設定されます。(Bug#1111589)
- ギリシャ語およびロシア語の言語環境では、Net8 Assistant の「Net8 Assistant」ダイアログ・ボックスに、リリース 8.1.7 ではなくリリース 8.1.6 と表示されます。

- Net8 Assistant または Net8 Configuration Assistant を使用してリスナーの構成を削除する場合、事前にリスナーを停止してください。
- Net8 Assistant の Oracle Names Server フォルダから Oracle Names Server を削除する場合、names.ora ファイルおよび ckp*.ora ファイルの Oracle Names Server エントリも手動で削除する必要があります。

Net8 インストールの既知の問題

Oracle Universal Installer では、LU6.2 プロトコルのライブラリが/opt/SUNWappc ディレクトリにあることを想定しています。LU6.2 プロトコルのライブラリが別のディレクトリにある場合には、そのディレクトリから/opt/SUNWappc へのリンクを作成してください。

たとえば、ライブラリが/opt/SUNWlu62 にある場合、次の構文を使用して/opt/SUNWappc へのリンクを作成できます。

```
ln -s /opt/SUNWlu62 /opt/SUNWappc
```

マニュアルの正誤表

『Oracle8i Net8 管理者ガイド リリース 8.1』には、次の機能およびエラーが記載されていません。

- sqlnet.ora ファイルのパラメータに指定する次の整合性および暗号化アルゴリズムの値は、リリース 8.1.6 の機能に対応しています。

```
SQLNET_CRYPTOCHECKSUM_TYPE_SERVER
SQLNET_CRYPTOCHECKSUM_TYPE_CLIENT
SQLNET_ENCRYPTION_TYPES_SERVER
SQLNET_ENCRYPTION_TYPES_CLIENT
```

関連項目：『Oracle Advanced Security 管理者ガイド リリース 8.1』では、このパラメータは変更されています。

- Oracle Connection Manager のリモート管理機能の詳細が説明されていません。

リモート管理を使用すると、別のコンピュータに常駐する Oracle Connection Manager をリモートの CMCTL セッションで制御できます。

リモート・アクセス構成は、管理対象の Oracle Connection Manager コンピュータおよび CMCTL の実行を計画しているリモート・コンピュータに必要です。

Oracle Connection Manager を実行するローカル・コンピュータにリモート・アクセスを構成する方法は、次のとおりです。

- 次のパラメータを使用して、cman.ora ファイルを構成します。

CMAN	—	Oracle Connection Manager ゲートウェイ・プロセス (CMGW) のリスニング・アドレスを指定します。
CMAN_ADMIN	—	Oracle Connection Manager 管理プロセス (CMADMIN) のリスニング・アドレスを指定します。
CMAN_PROFILE	—	Oracle Connection Manager に固有のパラメータを指定します。特に、リモート CMCTL セッションへアクセスする場合には、REMOTE_ADMIN=yes と設定する必要があります。

次は、サンプルの cman.ora ファイルです。

```
cman=(ADDRESS=(PROTOCOL=tcp)(HOST=sun1)(PORT=1630))
CMAN_ADMIN=(ADDRESS=(PROTOCOL=tcp)(HOST=sun1)(PORT=1830))
CMAN_PROFILE=
  (PARAMETER_LIST=
    (REMOTE_ADMIN=yes)
    ...
  )
```

これらのパラメータの詳細は、『Oracle8i Net8 管理者ガイド リリース 8.1』の付録 C 「構成パラメータ」を参照してください。

- CMGW プロセスを開始します。オペレーティング・システムから、次のコマンドを入力します。

```
CMCTL START cm
```

または、

```
CMCTL START [cman]
```

Oracle Connection Manager をリモートで管理できるように別のコンピュータを構成する方法は、次のとおりです。

- Oracle Connection Manager がインストールされていることを確認します。
- ローカルの Oracle Connection Manager コンピュータに設定した値と同じ値を CMAN および CMAN_ADMIN パラメータに指定して、`cman.ora` ファイルを構成します。

```
CMAN=(ADDRESS=(PROTOCOL=tcp)(HOST=sun1)(PORT=1630))
CMAN_ADMIN=(ADDRESS=(PROTOCOL=tcp)(HOST=sun1)(PORT=1830))
```

START コマンド以外の CMCTL コマンドを使用して、Oracle Connection Manager をリモートで管理できます。

CMCTL で使用可能なコマンドについては、『Oracle8i Net8 管理者ガイド リリース 8.1』の付録 A「制御ユーティリティ」を参照してください。

- ENVS パラメータが説明されていません。ENVS パラメータを使用すると、`listener.ora` ファイルの `SID_DESC` セクションの環境変数を指定できます。次に例を示します。

```
SID_LIST_listener_name=
(SID_LIST=
(SID_DESC=
(SID_NAME=plsextproc)
(ORACLE_HOME=/oracle81)
(PROGRAM=extproc)))
(ENVS=LD_LIBRARY_PATH=/sw0/app/oracle/product/8.1.7/lib:/sw0/app/oracle/product/8.1.7/ctx/lib))
```

リリース 8.1 では、たとえば (`ENVS='VISUAL=vi'`) のように、環境変数の値を一重引用符で囲まないでください。一重引用符は、リリース 8.0 ではサポートされていました。リリース 8.0 からリリース 8.1 に移行する場合は、二重引用符を使用するか、引用符を使用しないようにしてください。

複数の環境変数は、カンマで区切る必要があります。

Windows NT の場合、このリリースの ENVS パラメータはサポートされていません。リスナーによって起動されたプロセスは、いずれもリスナーの環境をそのまま継承します。`listener.ora` ファイルの ENVS 設定は無視されます。(Bug#1082940)

第 4 章 Oracle JSP について

原典情報：\$ORACLE_HOME/relnotes/README_jsp.txt ファイル

Oracle JSP の新機能

- Oracle JSP のこのリリースでは、カスタム・タグのタグ拡張メカニズムも含めて JSP 1.1 仕様をサポートしています。
- Oracle JSP は、埋込み Oracle Servlet Engine (OSE) を使用して、リリース 8.1.7 データベース内の JSP ページを変換および実行します。詳細は『Oracle8i JavaServer Pages 開発者ガイドおよびリファレンス』を参照してください。
- Oracle JSP からのデータベースへの操作を補助するために、一連の Database-Access JavaBeans と SQL タグが用意されています。
- Oracle JSP リリース (1.0.0.6.1) 以降、多くのコードが拡張され、いくつかの不具合が修正されました。後述の不具合の修正の項目を参照してください。
- 今回リリースの Oracle JSP は、次の環境でテストを行いました。
 - JDK 1.1.8 および 1.2.2
 - Oracle8i リリース 8.1.7 データベースの Oracle Servlet Engine
 - Jserv Servlet Engine (iAS) を使用した Apache Web サーバー、Solaris および Windows NT

Oracle JSP は Tomcat サーバー環境でも動作します。Tomcat で Oracle JSP を設定する方法は、『Oracle8i JavaServer Pages 開発者ガイドおよびリファレンス』を参照してください。

- Bug#1155470 の修正/拡張（後述を参照してください）に伴い、「Foo.jsp」ファイルは「_Foo.java」ファイルに変換されることに注意してください。この生成ファイルに対するネーミング・スキーマは、Foo.java に変換されていた Oracle JSP リリース 1.0.0.6.1 のネーミング・スキーマとは異なります。ただし、両方のネーミング・スキーマで ojspc コマンドライン変換ツール（リリース 1.0.0.6.1 ではベータ版でした）を使用している限り、この変更はユーザーの目に直接触れることはありません。

Oracle JSP リリース 1.1.0.0.0 での問題点と制限事項

- NT の場合、ojspc 変換ツールはファイル・リスト内のワイルド・カードをサポートしません。Solaris (UNIX) シェルは、ワイルド・カードを拡張するため、ワイルド・カードは Solaris (UNIX) シェル上で動作します。
- Database-Access JavaBeans は、oracle.jdbc2 パッケージのいずれのクラスもサポートしません。これは、異なる JDK バージョンで一貫性を保持するためです。
- リリース 8.1.7 では、Oracle Servlet Engine、PSP 上で実行する PL/SQL Gateway Servlet を JSP からコールできません（また、その逆もコールできません）。この相互運用性は、今後のリリースで提供される予定です。
- Oracle Internet Server 8i 1.0.1 NT リリース・ノート

Oracle Internet Server 8i 1.0.1 では、SQLJ に 8.1.6 ライブラリを使用します。また、Oracle JSP 1.1 は、SQLJ 8.1.7 で動作するように設計されています。そのため、SQL JSP (SQLJ コードを含む JSP) を動的にコンパイルするときに、「そのようなメソッドはありません」というエラーが発生します。この問題を回避するには、sqljcmd 構成パラメータを使用します。zone.properties には、次の行が必要です。

```
servlet.oracle.jsp.JspServlet.initArgs=sqljcmd=%ORACLE_HOME%\%apache%\jdk\bin\java.exe sqlj.tools.Sqlj
```

%ORACLE_HOME%には、正しい値の指定が必要であることを注意してください。他の代替策としては、ojspc のコマンドライン・トランスレータを使用して、SQLJSP をプリコンパイルすることです。

- ページ・ディレクティブ: language="sqlj"は不十分 (Bug#1285959)

SQLJ コードを使用する JSP の場合、次のディレクティブでは不十分です。

```
<%@ page language="sqlj" %>
```

OJSP トランスレータは、言語をスクリプトするときに SQLJ ではなく Java を使用する「.jsp」ファイルとして処理します。これを回避するには、ファイル名を「.sqljsp」に変更します。

- jsp:include にページ値が未定義の場合 (Bug#1234581)

```
StringIndexOutOfBoundsException
```

このエラーは、次のディレクティブに起因します。

```
<jsp:include page="" flush="true" />
java.lang.StringIndexOutOfBoundsException: String index out of range:
    at java.lang.String.charAt(String.java)
    at oracle.jsp.runtime.OraclePageContext.include(OraclePageContext.java)
    at empty._jspService(empty.java:59)
    at oracle.jsp.runtime.HttpJsp.service(HttpJsp.java)
    at oracle.jsp.app.JspApplication.dispatchRequest(JspApplication.java)
    at oracle.jsp.JspServlet.doDispatch(JspServlet.java)
    ...
```

これを回避するには、ページに対して空でない文字列を指定するか、または条件によってインクルード文を削除します。

- JSP ページのロード時の `Java.lang.VerifyError` (Bug#1138828)

これは、エラーが消えてからしばらく後に、JSP ページに対して突発的に発生します。スタック・トレースがクラス・ローダーを指し示していると考えられます。

```
JspServlet:unable to dispatch to requested page:java.lang.VerifyError
    at java.lang.ClassLoader.resolveClass(ClassLoader.java)
    at oracle.jsp.app.JspClassLoader.loadClass(JspClassLoader.java:538)
    at java.lang.ClassLoader.loadClass(ClassLoader.java)
    ...
```

この問題は、JDK 1.1.6 の不具合 (JDK Bug ID:4211181) としてトレースされていました。JDK 1.1.8 以降では、このエラーは発生しません。

Oracle JSP リリース 1.0.0.6.1 からの不具合の修正

`<jsp:expression>` および `<jsp:declaration>` が機能しない (Bug#1358636)

OJSP では `<jsp:expression>` および `<jsp:declaration>` が機能しないという不具合がありましたが、標準ではないその短縮形式 `<jsp:expr>` および `<jsp:decl>` は問題ありませんでした。Oracle JSP 1.1 では、この不具合は修正されています。

`errorPage` / `jsp:forward` / `jsp:include` で「..」が機能しない (Bug#1288730)

次の `errorPage` ディレクティブが機能しないという不具合がありました。

```
<%@ page language="java" errorPage="../control/error.jsp" %>
```

Oracle JSP 1.1 では、この不具合は修正されています。

jsp:forward で HTML ページに転送できない (Bug#1261458)

この制限は、Oracle Internet Application Server 8i リリース 1.0 などの Servlet 2.0 環境に適用されていました。次の文では例外が発生します。

```
<jsp:forward page="hello.html" />
javax.servlet.ServletException: Forward /bugs/hello.html failed: can only
forward to a JSP page.
    at java.lang.Throwable.(Compiled Code)
    at java.lang.Exception.(Compiled Code)
    at javax.servlet.ServletException.(Compiled Code)
    at oracle.jsp.runtime.OraclePageContext.forward(Compiled Code)
    ...
```

これは、実際には RequestDispatcher インタフェースがない Jserv 内の Servlet 2.0 API の制限です。Oracle JSP 1.1 では、この不具合は修正されています。

場合によっては静的テキスト内の改行文字が欠落する (Bug#1250326)

静的テキスト（たとえば、html 部分）内の改行文字が作成されたクラスで欠落する場合があります。たとえば、"<html>¥r¥n"は、次のように変換される場合があります。

```
out.println("<html>");
```

本来は、次のように変換されます。

```
out.println("<html>¥r");
```

または、次のように変換されます。

```
out.print("<html>¥r¥n");
```

Oracle JSP で html ファイルを動的にインクルードできない (Bug#1238238)

動的インクルード・コマンドにより、Apache/Jserv で例外が発生する場合があります。

```
<jsp:include page="hello.html" />
javax.servlet.ServletException: Include hello.html failed: can only include
a JSP page.
    at oracle.jsp.runtime.OraclePageContext.include(OraclePageContext.java)
    ...
```

これは、（RequestDispatcher インタフェースがない状態での）動的インクルードおよび転送に対する Servlet 2.0 API の制限です。Oracle JSP 1.1 では、この不具合は修正されています。

Apach でサーブレット別名を使用して Oracle JSP が起動される場合に、ファイル位置の特定に失敗 (Bug#1237578)

Oracle JSP 1.0.0.6.1 では、`alias_translation` が TRUE (デフォルト) に設定されている場合、フル・クラス名「`oracle.jsp.JspServlet`」のかわりにサーブレットの別名「`orajsp`」を使用するなど、サーブレット別名を使用して Oracle JSP サーブレットを起動すると、一部のファイルの位置特定がエラーとなりました。(回避方法は、`alias_translation` を FALSE にするか、JSP サーブレットをフル・クラス名「`oracle.jsp.JspServlet`」で起動することでした。) Oracle JSP 1.1 では、この不具合は修正されています。

エラー・メッセージでのマルチバイト文字「?」の表示 (Bug#1222788)

Oracle JSP には、エラー・メッセージでのマルチバイト文字の表示に不具合がありました。例えば、JSP に次の行があると仮定します。

```
<jsp:setProperty name="XXX" property="YYY" value="ZZZ">
```

XXX、YYY、ZZZ はマルチバイト文字です。

JSP のこの行がなんらかの理由で失敗すると、次のエラー・メッセージが表示されました。

例外:

```
oracle.jsp.parse.JspParseException: Line # 10, name="???" property="???"  
value="" /> Error: Error in attribute list: ??? is not a defined bean.
```

Oracle JSP 1.1 では、この不具合は修正されています。

Short が正しくキャストされない (Bug#1213897)

テスト・ケース:

```
<%@ taglib uri="oracle.jsp.parse.OpenJspRegisterLib" prefix="jml" %>  
<jml:useBean id="shortBean" class="Bean.ShortBean" scope="page" />  
<jml:setProperty name="shortBean" property="shortValue" />  
<%= shortBean.getShortValue() %>
```

Bean には Short 型で定義されたプロパティがありますが、エラーが戻されます。

```
Errors compiling: jml_setProperty3.  
  Incompatible type for method. Can't convert short to java.lang.Short.  
    shortBean.setShortValue(Short.valueOf( __paramList[0]).shortValue());  
      at oracle.jsp.app.JspJavacCompiler.compile(JspJavacCompiler.java:16  
7)  
    at oracle.jsp.app.JspAppLoader.reloadPage(JspAppLoader.java:870)  
    at oracle.jsp.app.JspAppLoader.loadPage(JspAppLoader.java:749)  
    at oracle.jsp.app.JspAppLoader.getPage(JspAppLoader.java:501)  
    ...
```

Oracle JSP 1.1 では、この不具合は修正されています。

jml:useForm - キャストの問題 (Bug#1213777)

テスト・ケース:

```
<%@ taglib uri="oracle.jsp.parse.OpenJspRegisterLib" prefix="jml" %>  
<jml:useForm id = "user" type = "fpnumber" param="plo" />
```

例外:

```
oracle.jsp.provider.JspCompileException:  
Errors compiling:JML_useForm4.java  
  Incompatible type for method. Can't convert java.lang.String to double.  
user.setValue( __paramVals[0]);  
  at oracle.jsp.app.JspJavacCompiler.compile(JspJavacCompiler.java:167)  
  at oracle.jsp.app.JspAppLoader.reloadPage(JspAppLoader.java:870)  
  at oracle.jsp.app.JspAppLoader.loadPage(JspAppLoader.java:749)  
  at oracle.jsp.app.JspAppLoader.getPage(JspAppLoader.java:501)  
  at oracle.jsp.app.JspApplication.dispatchRequest(JspApplication.java:  
325)  
  at oracle.jsp.JspServlet.doDispatch(JspServlet.java:227)
```

複数の行にわたるインポート・リストのサポート (Bug#1207720)

以前の Oracle JSP では、インポート・パッケージで複数行にわたるリストのブレイクをサポートしていませんでした。したがって、次の場合は有効でした。

```
<%@ page import="package1,package2,package3" %>
```

または、次の場合も有効でした。

```
<%@ page import="package1" %>  
<%@ page import="package2" %>  
<%@ page import="package3" %>
```

次の構文は認められていませんでしたが、Oracle JSP 1.1 ではサポートされるようになりました。

```
<%@ page import="package1"
      import="package2"
      import="package3" %>
```

JSP ファイル名に「-」（ダッシュ）を使用できない（Bug#1155470）

JSP 1.0 の仕様では、jsp ファイル名に「-」（ダッシュ）の使用が制限されており、jsp ファイルに a-b.jsp などの名前を付けることはできませんでした。jsp エンジンでは a-b.jsp に基づいて a-b.java を作成しますが、これが Oracle JSP でコンパイル・エラーが発生する原因でした。この java ファイルには、「a-b」と呼ばれる公開クラスがあり、これが Java 言語では無効な名前であるため、コンパイル・エラーが発生しました。

注意：html ファイルには a-b.html などの名前を指定できます。

現在、Oracle JSP 1.1 では、デフォルトで JSP ファイル名を正当な Java 識別子にコード化しています。たとえば、/public/dir1/a-b.jsp の場合、パッケージ名「_public_dir1」となり、クラス名「_a_2d_b」となります。したがって、Oracle JSP 1.1 では、JSP のファイル名にある予約語および無効な Java 識別子の文字が原因で問題が発生することはありません。

Oracle JSP リリース 1.0.0.6.0 以降の不具合の修正

JSP とサーブレットでセッションを共有できない（Bug#1225531）

JSP がセッション・オブジェクトに値を入力すると、同一の Oracle8i JVM 内のサーブレットは、その値を直接参照できません。これは、JSP セッションがセッション・ラッパー・オブジェクトであるためです。JSP セッションに格納されている値は、基礎となるサーブレット・セッションに直接公開されません。

globals.jsa が検出されない場合、JSP のセッション・オブジェクトは、実サーブレット・セッション・オブジェクトになります。また、ディレクトリ内で globals.jsa が検出された場合、JSP セッション・ラッパー・オブジェクトは、デフォルトで、基礎となるサーブレット・セッションにセッション・データが公開されるようになりました。

別名パスのルート・ディレクトリにあるページに対して Oracle JSP が機能しない (Bug#1230626)

ユーザーが別名（たとえば、/vpath/->/ppath/）を使用して Apache/Jserv を構成し、別名のルートの右側のページ（たとえば、http://host/vpath/test.jsp）にアクセスしようとすると、ページが見つからないというエラーが Oracle JSP から戻されました。

この不具合は、別名に関連するルーチンでパス一致ロジックを変更することにより修正されました。

別名を使用している場合に request.getPathTranslated() から不適切な値が戻る (Bug#1233612)

JSP で別名ディレクトリを使用している場合に、ドキュメント・ルートにその JSP があるかのような不適切な値が request.getPathTranslated() から戻りました。

この不具合は、要求ラッパー・オブジェクトが、システム固有の要求オブジェクトの基礎となる正しいメソッドをコールすることにより修正されました。

アプリケーション関連とページ関連を混在して使用する場合の jsp:include 障害 (Bug#1235873)

この不具合は、Apache/Jserv の構成の場合にのみ影響があります。ページに、あるアプリケーション・パスに関連する jsp:include が存在し、さらにページ・パスに関連する jsp:include が存在する場合、2 番目の jsp:include のファイル位置の特定が失敗となります。

この不具合は、インクルード処理が終了した後に、サーブレットのインクルード属性を要求オブジェクトから削除することによって修正されました。

第 5 章 Oracle SQLJ について

原典情報：\$ORACLE_HOME/sqlj/README.txt ファイル

SQLJ の概要

Oracle、IBM、Compaq、Sybase、Informix、Sun Microsystems などの各社は、静的な SQL 文と SQL 構造体の Java プログラム (ANSI X3.135.10-1998) への埋込みに関して SQLJ 規格に貢献してきました。この規格の更新バージョンは、現在 ISO がファイナライズされています。

SQLJ は、データベースへの接続が必要な Java プログラムの開発とメンテナンスにかかるコストを削減する 1 つの手段です。静的な SQL 文 (アプリケーションまたはアプレットの実行によって変化しない SQL。通常、アプリケーションには動的 SQL よりも静的 SQL が多く含まれています。) を含んだ Java コードに対して、簡素化されたモデルを提供します。開発とメンテナンスのためのコストが削減される理由は、SQL を直接 Java プログラムに埋め込むための容易な方法論を SQLJ が採用している点にあります。このため、SQLJ は簡潔で読みやすいコードとなっています。アプリケーションが使用する SQL 文に対してコンパイル時のチェックを提供し、コーディングをさらに簡素化しています。JDBC などの動的 SQL API を使用する場合は、実行するコードの行をテストするまで、その SQL が構文的または意味的に間違っているかどうかの検出は困難です。

Oracle の SQLJ 実装である今回のリリース 8.1.7 は、SQLJ プログラムを受け入れ、JDBC を (SQLJ ランタイムを介して) コールする Java プログラムに戻すトランスレータの形式になっています。トランスレータ自体は純粋に Java で記述されています。

今回のリリースでは、Java の埋込み SQL に関する ANSI 規格がサポートされています。さらに、このリリースの JDK 1.2 ランタイムは、Java の埋込み SQL に対する今後の ISO 規格もサポートします。

また、Oracle8i リリース 8.1.7 JDBC ドライバの機能もサポートしています。さらに、今回のリリースは、Oracle データベースに対して実行する Oracle JDBC ドライバの他に、任意のデータベース・システムに対して実行する標準的な JDBC ドライバにも使用できます。

SQLJ 配布の内容

ここでは、SQLJ 配布ファイルが Oracle8i リリース 8.1.7 のインストーラを使用して、ディレクトリ \$ORACLE_HOME/sqlj にすでにインストール済みであることを前提に説明しています。\$ORACLE_HOME は、Oracle 製品がインストールされるルート・ディレクトリです。

SQLJ 配布の内容は、次のとおりです。

- [\$ORACLE_HOME/sqlj/README.txt] — この章の英語版。
- [\$ORACLE_HOME/sqlj/index.html] — SQLJ のインストールの詳細を記述した HTML ファイル。このファイルには、その他のドキュメントへの参照も含まれています。
- [\$ORACLE_HOME/sqlj/doc/whitepaper.pdf] — PDF 形式のホワイト・ペーパー「SQLJ: Embedded SQL in Java」
- [\$ORACLE_HOME/sqlj/doc/sqlj-overview.pdf] — PDF 形式のテクニカル・ホワイト・ペーパー「Overview of SQLJ - Embedded SQL in Java」
- [\$ORACLE_HOME/sqlj/doc/sqlj-objects.pdf] — PDF 形式のテクニカル・ホワイト・ペーパー「Using Oracle Objects in SQLJ programs」
- [\$ORACLE_HOME/sqlj/doc/sqlj-primer.pdf] — PDF 形式の SQLJ 入門書「SQLJ: Tricks, Traps, and Gems」
- [\$ORACLE_HOME/sqlj/doc/runtime] — SQLJ ランタイム API: javadoc 形式でドキュメント化したランタイム・パッケージ。
- 個別の SQLJ ランタイムのバージョンは、次のとおりです。
 - [\$ORACLE_HOME/sqlj/lib/runtime11.zip] — Oracle8i リリース 8.1.7 JDBC の JDK 1.1 用ランタイム
 - [\$ORACLE_HOME/sqlj/lib/runtime12.zip] — Oracle8i リリース 8.1.7 JDBC の JDK 1.2 用ランタイム
 - [\$ORACLE_HOME/sqlj/lib/runtime.zip] — 全 JDBC ドライバの全 JDK 用ランタイム
- [\$ORACLE_HOME/sqlj/lib/translator.zip] — SQLJ トランスレータの *.class ファイルを含んだ Zip ファイル。トランスレータを使用するには、前述のランタイム Zip ファイルが必要です。
- [\$ORACLE_HOME/sqlj/demo/TestInstall*] — インストールのテスト用ファイル。

- [`$ORACLE_HOME/sqlj/demo`] — サンプル・ファイル。
 - 詳細は、`[$ORACLE_HOME/sqlj/demo/README.txt]`を参照してください。
 - オブジェクト型を使用した追加サンプルは、`[$ORACLE_HOME/sqlj/demo/Objects]`を参照してください。
 - SQLJ ISO 規格から導出された JDBC 2.0 のサンプルは、`[$ORACLE_HOME/sqlj/demo/jdbc20]`のサンプルを参照してください。
 - Java ストアド・プロシージャを使用したサンプルは、`[$ORACLE_HOME/sqlj/demo/server]`を参照してください。
 - アプレットの SQLJ プログラムは、`[$ORACLE_HOME/sqlj/demo/applets]`を参照してください。
 - SQLJ のカスタマイズとコンポーネントの SQL チェックのサンプルは、`[$ORACLE_HOME/sqlj/demo/components]`を参照してください。
- [`$ORACLE_HOME/bin`] — SQLJ と関連ツールの起動を簡素化するためのラッパー・スクリプトまたはラッパー実行ファイル（あるいはその両方）。

この SQLJ のインストールで提供される SQLJ ドキュメントの索引は、Web ブラウザで `$ORACLE_HOME/sqlj/index.html` を参照してください。

『Oracle8i SQLJ 開発者ガイドおよびリファレンス』が Oracle8i リリース 8.1.7 ドキュメントの一部としてリリースされています。このマニュアルには、インストールのテスト方法や製品の使用方法など、詳細が説明されています。

SQL のオブジェクト型とコレクション型に対する Java クラスを生成するための JPublisher ツールの詳細は、JPublisher のオンライン・ヘルプを参照してください。Oracle の JDBC ドライバの詳細は、『Oracle8i JDBC 開発者ガイドおよびリファレンス』を参照してください。

インストール

JDK 1.1.x または JDK 1.2 互換の Java 環境が必要です。さらに、Oracle8i リリース 8.1.7 JDBC ドライバを事前にインストールする必要があります。

SQLJ シェル・スクリプトまたは実行ファイルを含む \$ORACLE_HOME/bin ディレクトリが環境変数 PATH に設定してあることを確認してください。

- \$ORACLE_HOME/sqlj/lib/translator.zip ファイルを環境変数 CLASSPATH に追加します。
- JDK 1.1.x と Oracle8i JDBC リリース 8.1.7 を使用している場合は、\$ORACLE_HOME/sqlj/lib/runtime11.zip ファイルを環境変数 CLASSPATH に追加します。
- JDK 1.2 以降と Oracle8i JDBC リリース 8.1.7 を使用している場合は、\$ORACLE_HOME/sqlj/lib/runtime12.zip ファイルを環境変数 CLASSPATH に追加します。
- Oracle8i JDBC リリース 8.1.7 を使用していない場合は、\$ORACLE_HOME/sqlj/lib/runtime.zip ファイルを環境変数 CLASSPATH に追加します。

注意:

- 以前のリリースの SQLJ のみ、SQLJ プログラムの変換または実行（あるいはその両方）をするために、translator.zip を環境変数 CLASSPATH に設定する必要があります。
- SQLJ リリース 8.1.7 の場合は、SQLJ プログラムを変換するために、translator.zip とランタイムの Zip ファイル (runtime.zip、runtime11.zip または runtime12.zip) の 1 つを環境変数 CLASSPATH に設定する必要があります。

環境変数 PATH と CLASSPATH に必要な設定の詳細は、『Oracle8i SQLJ 開発者ガイドおよびリファレンス』を参照してください。

JDBC がインストールされていることを確認してから、SQLJ を使用してください。

SQLJ と JDBC のインストールのテストについては、『Oracle8i SQLJ 開発者ガイドおよびリファレンス』を参照してください。

今回の SQLJ は、次の構成で実行されています。

- **アプリケーションの場合:** Oracle8i リリース 8.1.7 の JDBC OCI8 とリリース 8.1.7 JDBC Thin ドライバを使用した Solaris 2.6、Oracle8i リリース 8.1.7 の JDBC OCI8 とリリース 8.1.7 JDBC Thin ドライバを使用した Windows NT 4.0、サーバー側 (kprb) JDBC ドライバを使用した Oracle8i リリース 8.1.7 サーバー側 JavaVM
- **アプレットの場合:** Netscape 4.7 (Windows NT、Solaris)、Microsoft Internet Explorer 5.0 (Windows NT、Windows 95)、AppletViewer (Solaris) および Java plug-in 1.1.2 (Windows NT - Explorer、Solaris - Netscape および Windows NT - Netscape)

新機能

Oracle8i リリース 8.1.6.0.0 以降の変更

言語

- SQLJ ISO 仕様で定義されている JDBC 2.0 機能に対するサポート
(`$ORACLE_HOME/sqlj/demo/jdbc20` のサンプル・プログラムを参照してください)。
 - `java.sql.SQLData` のインスタンスなどのユーザー定義型の読み込みと書き込みを行うための接続コンテキスト型のマップ機能。この機能には、JDK 1.2 と SQLJ ランタイム `runtime12.zip` が必要です。
 - スクロール可能なイテレータ（位置指定と名前指定の両方）に対するサポート
 - 接続コンテキストと `DataSources` との関連付け
 - `ExecutionContext` を使用したバッチ管理と行のプリフェッチ
 - JDBC 2.0 接続プーリングに対するサポート
- RAW 列と BLOB 列における Java オブジェクトのシリアライズ化に対する接続コンテキスト型のマップ機能を使用したサポート。
- 位置指定イテレータに対する新しい「`FETCH CURRENT FROM ...`」構文によって、SQLJ イテレータへの結果セットの移行時に、プログラム・ロジックを保持できます。

トランスレータ

トランスレータの `statusMain` メソッドは繰り返しコールできます。これにより、Oracle JSP トランスレータへの SQLJ 埋込みが可能となりました。（Bug#1321820）

ランタイム

- 異なる SQLJ ランタイム・ライブラリを使用できます。
 - `runtime11.zip`、`runtime12.zip` : アプレット対応で、Oracle JDBC ドライバ リリース 8.1.7 での使用に最適化されています。
 - `runtime.zip` : すべての Oracle JDBC ドライバと互換性があります。

関連項目：「修正済みの不具合」項目を参照してください。

Oracle8i リリース 8.1.5.0.0 以降の変更

言語

- Java 言語に対する SQLJ サポートが改善されました

関連項目: 後述の Bug#808400、Bug#808729、Bug#808856 および Bug#845528 を参照してください。

- 変換時に `Public static` の最終フィールド値（WITH 句の属性など）が正しく完全に評価されるようになりました。（Bug#808856）
- イテレータをサブクラス化できるようになりました。SQLJ で `oracle.sql.CustomDatum` クラスのサブクラスをホスト表現に使用できます。（Bug#879170）
- JDBC 2.0 の `java.sql.Struct/Ref/Array/Blob/Clob` 型をサポートします。
- JPublisher が生成する JDBC 2.0 の Java ラッパー・クラスをサポートし、`java.sql.SQLData` インタフェースを実装します。

このリリースでは、ANSI 規格 ANSI X3.135.10-1998 をサポートしています。

トランスレータ

- SQLJ には、`javac` と同様に制限された `make` 機能のフォームがあります。（Bug#801780）
注意: 関連する「.sqlj」ソース・ファイルすべてを常に SQLJ コマンドラインに指定する必要があります。トランスレータは追加の「.java」ソースを自動的にチェックできますが、必要な「.sqlj」ソースはチェックできません。

次のコマンドライン・オプションを使用して、この機能をオフに切り替えてください。

```
checksource=false
```

- `jdb` デバッガで、インストールされたクラス・ファイルをデバッグする場合は、「`-linemap`」ではなく、新しい「`-jdblinemap`」オプションを使用してください。（Bug#868524）
- 「`-help`」オプションで、使用方法の概要を出力できます。すべてのオプション設定の完全ヘルプは、「`-help-long`」オプションに移動しました。コマンドラインの短縮表記は、「`help-alias`」オプションで確認できます。
- 「`version-long`」オプションで、環境変数 `CLASSPATH` でアクセス可能な JDBC ドライバのバージョン（ドライバがある場合）や使用中の Java のバージョンなどを含む追加バージョン情報を印刷できます。
- 「`-explain`」フラグが追加されました。このフラグは、SQLJ トランスレータが発行したエラーと警告メッセージの原因および処置情報を表示します。

チェック

デフォルトのオフラインとオンラインのチェッカー・クラスは、`oracle.sqlj.checker.OracleChecker` となりました。構成内で検出された JDBC ドライバに従って、適切なチェッカーが選択されます。リリース 8.0.x または 8.1.x の JDBC ドライバを使用して、プログラムがサポートしている型を Oracle7 の機能にダウングレードする場合のみ、特定のチェッカーを明示的に指定する必要があります。

パフォーマンス

SQLJ では、以前からのパフォーマンスに関する拡張機能を Oracle の JDBC ドライバを使用してすべてサポートしています。詳細は『Oracle8i SQLJ 開発者ガイドおよびリファレンス』の「付録 A」を参照してください。

- 文のキャッシュ

SQLJ ランタイムは、デフォルトで、指定した JDBC 接続で実行した SQL 文のうち最新の 5 つをキャッシュします。「-P-Cstmtcache=NN」フラグを使用して、キャッシュ・サイズを NN に設定できます。0（ゼロ）に設定すると、文のキャッシュは使用禁止になります。

- バッチ処理

SQLJ は、INSERT、DELETE および UPDATE 文のバッチ処理をサポートします。バッチ処理機能は、`ExecutionContext` を介して管理されます。

注意： 提供されるバッチ処理 API は、SQLJ に関する今後の ISO 規格内にある SQLJ バッチ処理サポートに準拠しています。

- パラメータ・サイズの登録

新しいフラグ「-P-Coptcols」と「-P-Coptparams」を使用すると、可変長のパラメータと結果セットの列を最適化できます。

ランタイム

ランタイムのサポートの変更については、`[$ORACLE_HOME/sqlj/doc/runtime/RELEASENOTES.txt]` ファイルを参照してください。

プラットフォーム固有の情報

プラットフォーム固有の問題の処理に役立つトラブルシューティングの一般的なヒントをいくつか紹介します。

一般的なトラブルシューティング

- JIT (Just In Time compilation) をオフにしてください。大半の場合、これで問題を回避できます。次の環境変数を設定してください。

```
JAVA_COMPILER=NONE
```

または、SQLJ コマンドラインに次の追加フラグを指定します。

```
J-Djava.compiler=none
```

注意： 環境変数 SQLJ_OPTIONS は、常に SQLJ コマンドラインに送信する追加フラグに設定できます。

- 環境変数 PATH を縮小して、単一の既知の JDK バージョンを組み込んでください。
- 環境変数 CLASSPATH を縮小して、次の内容のみを組み込んでください。
 - ソース・ファイル (カレント・ディレクトリであることが多い)
 - Oracle JDBC ドライバの classes111.zip または classes12.zip
 - SQLJ トランスレータの translator.zip
 - SQLJ ランタイムのうちの 1 つ (runtime.zip、runtime11.zip または runtime12.zip)
- Java のコンパイルが中断された場合は、(SQLJ コマンドラインの -status を使用して、SQLJ トランスレータの進行状態を調べた後) SQLJ コマンドラインに次のフラグを追加します。

```
-passes
```

- 異なる JDK バージョンを使用します。

UNIX プラットフォーム

- JavaVM と Java コンパイラをプラットフォーム上で起動する方法を SQLJ がカスタマイズする必要がある場合は、シェル・スクリプトを編集します。

```
$ORACLE_HOME/bin/sqlj
```

Windows プラットフォーム

ラッパー実行ファイル\$ORACLE_HOME/bin/sqlj.exe は、Windows NT でのみテストされていることに注意してください。

- Windows の一部のバージョンには、環境内での文字数とコマンドラインに関する制限事項があります。
- Windows 95 の場合、環境変数に割り当てた文字列に「=」は使用できません。SQLJ_OPTIONS では、かわりに「#」を使用します。たとえば、JIT をオフに切り替える場合、通常は次のように指定します。

```
set SQLJ_OPTIONS=-J-Djava.compiler=none
```

Windows 95 では、次のように指定します。

```
set SQLJ_OPTIONS=-J-Djava.compiler#none
```

- 環境変数 CLASSPATH にスペース文字は指定できません。

修正済みの不具合

SQLJ リリース 8.1.7 での修正された不具合について

トランスレータ

- SQLJ 文で任意のサイズの SQL ソース・テキストがサポートされるようになりました。以前のリリースでは、65K に制限されていました。(Bug#1018117)
- 「-ser2class」変換で、任意のサイズの「.ser」ファイルがサポートされるようになりました。以前のリリースでは、32K に制限されていました。(Bug#1061663)

ランタイム

- 新しい Oracle SQLJ ライブラリ runtime11.zip により、Oracle SQLJ プログラムがアプレット環境で正しく動作するようになりました。(Bug#814704)

注意：使用するブラウザによっては、Reflection の使用が必要なため、一部の機能 (CAST 文、CustomDatum および SQLData のサポート) が使用できない場合があります。この場合、Sun Microsystems 社の Java Plug-In を使用する方法があります。これによって、4.x 以前のブラウザでもすべての SQLJ 文を実行できます。

アプレットのサンプル (\$ORACLE_HOME/sqlj/demo/applets) も参照してください。

- 現在、SQLJ アプレットでは、次のエラーは発生しません。(Bug#850197)

```
java.lang.IllegalAccessError:
sqlj.runtime.ExecutionContext.DEBUG
at sqlj.runtime.ExecutionContext$StatementStack.setStatement
```

SQLJ リリース 8.1.6 での修正された不具合について

言語

- SQLJ では、int[] a,b[];などの複合配列宣言が可能になりました。(Bug#808400)
- SQLJ ホスト式は、配列を戻すファンクションをコールできます。(Bug#845528)
- メソッド・ブロック内での SQLJ イテレータとコンテキスト宣言が可能になりました。(Bug#808729)

注意: JDK 1.1.x の場合、ブロック・レベルで宣言されたクラスは参照できません。このようなクラスを使用すると、後で Java コンパイラからエラーが表示されます。ただし、JDK 1.2 では、ロック・レベルの宣言は完全にサポートされています。

- ¥uff70や¥uff65などの日本語半角カタカナ文字がSQL識別子に使用できるようになりました。一般的に、SQL 識別子は、Unicode 2.0、Java およびすべての Oracle SQL キャラクタ・セットにある文字と数字定義のスーパーセットからの文字で構成できます。

トランスレータ

- SQLJ トランスレータが JDK 1.2 で動作するようになりました。適切な Oracle JDBC ドライバ・ファイル (classes12.zip) を使用していることを確認してください。(Bug#814692)
- SQLJ トランスレータの現行のリリースは、Microsoft の J++ Java VM (jview) で実行され、作成された「*.java」ファイルは Microsoft の J++ Java Compiler (jvc) でコンパイルされます。
- 「-linemap」オプションは、インナー・クラスも含め、「.sqlj」ソース・ファイルで宣言されているすべての Java クラスをインストールメントするようになりました。唯一の例外は、現在インストールされていない無名クラスです。インストールメント時に発生していた問題は修正済みです。(Bug#814687、Bug#814680)
- SQLJ 文の SQL ソース・テキストは、任意のサイズにできます。(Bug#1018117)
- ファイルの数にかかわらず、SQLJ の「-ser2class」変換を実行できます。(Bug#859265)
- また、「-ser2class」は、任意のサイズの「.ser」ファイルを「.class」表現に変換できるようになりました。(Bug#1061663)

- 一部の警告が誤ったソース・ファイルで報告されることがなくなりました。
(Bug#859307)
- Windows NT プラットフォーム上で、ディレクトリ・セパレータに「¥」ではなく「/」が使用されると、SQLJ でそれを検出し、エラーを発生します。(Bug#550152)
- Windows NT プラットフォーム上で SQLJ が中断しないようになりました。これにより、「-passes」オプションを使用する必要はなくなりました。(Bug#749636)

オンライン・チェック

- 複数のデータベース・エラーが正しく報告されるようになりました。(Bug#460846)
- ストアド・プロシージャのオーバーロードが PL/SQL ルールと一致するようになりました。(Bug#808874)

ランタイム

SQLJ 接続コンテキストが NULL の JDBC 接続で初期化されると、SQLException が発行されます。

判明している制限事項と問題点

SQLJ リリース 8.1.7 で判明している問題点と不具合は、次のとおりです。

言語

- 実行可能な SQLJ 文を含むクラスには、「java」、「sqlj」または「oracle」の名前を付けることができません (Bug#808835)。さらに、SQLJ 文で変数の型がユーザー定義型「x.y.MyClass」のホスト変数が使用されている場合、このホスト変数を使用しているクラスの名前に「x」は指定できません。この問題を回避するには、パッケージ名は小文字で開始し、クラス名は大文字で開始する Java のネーミング規則に従うことをお勧めします。
- SQLJ は、「a\$b」形式のクラスをインポートできません (Bug#1083057)。「a\$b.c」を参照せずに、間違って「a\$b\$c」を使用してしまいます。名前に「\$」を含むクラスは参照しないでください。

- SQLJ 変換は、次のようなリモート・プロシージャ・コールをサポートしていません (Bug#1275178)。

```
#sql { call remote_procedure@db_link(...) };
```

この問題を回避するには、SQLJ リリース 8.1.7 では、リモート・プロシージャ・コールをオフライン変換時にサポートします。ただし、オンライン変換時にはエラーとなります。

トランスレータ

- フランス語、ドイツ語および日本語のメッセージ変換のみ使用可能です。ただし、これらの変換にはヘルプ・メッセージは含まれていません。(Bug#803875、Bug#803869)。
- SQLJ トランスレータは、スペースを含むファイル名またはディレクトリ名をサポートしていません (Bug#864304)。これは、「-d」、「-dir」および「-classpath」オプションに特に関係があります。

Windows NT プラットフォームでは、Sun の JDK は、スペースを含むファイルおよびディレクトリ名をサポートしていることに注意してください。この問題を回避するために、「-d」、「-dir」または「-classpath」オプションには、DOS 形式と同等の短いファイル名またはディレクトリ名を使用してください。

- SQLJ トランスレータが Solaris 2.6 で SQLJ トランスレータを実行中に、中断する場合があります。

これは、Solaris オペレーティング・システムとともに配布された/usr/bin/java (Java リリース 1.1.3) の JavaVM を使用中に発生しました。

この問題は、オペレーティング・システムに付属している JavaVM ではなく、JDK の中の JavaVM を使用することで回避できます。

- SQLJ 変換で、JavaVM でコア・ダンプが発生する場合があります (Bug#1102856)。次のリストは、発行された Java スタック・ダンプの冒頭の部分です。

```
java.util.zip.ZipFile.findEND(ZipFile.java)
java.util.zip.ZipFile.readCEN(ZipFile.java)
java.util.zip.ZipFile.<init>(ZipFile.java)
java.util.zip.ZipFile.<init>(ZipFile.java)
sun.tools.java.ClassPath.<init>(ClassPath.java)
```

この問題が発生した場合は、次の方法で回避してください。

- a. \$ORACLE_HOME/lib/lclasses11.zip と [JDK Home]/lib/classes.zip を CLASSPATH から削除します。
- b. JDK 1.1.6 の使用を中止します。この問題は、他の JDK バージョンでは発生しません。
- c. コマンドラインから SQLJ をコールします。この問題は、SQLJ ラッパー・スクリプトがコマンドラインから直接起動された場合には発生しません。
- Java でサポートされている一部のコード化が SQLJ では無効です (Bug#1051122) 。
 - JDK 1.1.3 を使用している場合、次のコード化は javac では機能しますが sqlj では機能しません。
Cp1125、GBK、ISO2022CN、ISO2022CN_CNS、ISO2022CN_GB、ISO2022KR、JIS0208
 - JDK 1.1.6 を使用している場合、次のコード化は javac では機能しますが sqlj では機能しません。
Cp1125、ISO2022CN、ISO2022CN_CNS、ISO2022CN_GB、JIS0208
 - JDK 1.2 を使用している場合、次のコード化は javac では機能しますが sqlj では機能しません。
ISO2022CN、ISO2022CN_CNS、ISO2022CN_GB
- Windows NT で JDK 1.1.8 を使用している場合、SQLJ トランスレータが不完全な .java ファイルを生成します (Bug#1080299) 。この場合は、別の JDK バージョンを使用する必要があります。
- 「Unable to convert XXXX.ser to a class file」が発生します (Bug#1070362) 。「-ser2class」オプションを使用して、パッケージがある場合は、次のいずれかの方法を選択する必要があります。
 - ソース階層でパッケージ階層をミラー化し、常にルート（この場合、環境変数 CLASSPATH に存在していることが必要です）から変換します。
 - 「-d」オプションを使用して、クラス・リポジトリ・ディレクトリ（この場合、環境変数 CLASSPATH に存在していることが必要です）に変換します。

ランタイム

- JDK 1.2.1 では、イテレータ・インスタンスの `close()` を実行すると、`NullPointerException` が発生する場合があります。これは、次のような形式のコードで発生します。

```
NamedOrPositionalIterator iter;  
#sql iter = { SELECT .. };  
... while loop to process iter ..  
iter.close(); // <= NullPointerException on iter!
```

この問題は、JDK 1.2.1 のガベージ・コレクションの欠陥に起因しています。別の JDK バージョンで SQLJ プログラムを実行する必要があります (Bug#1130431)。

- JDBC Thin ドライバを使用している場合は、ロケール依存リテラルを使用できません (Bug#810245)。

サーバー側 JavaVM における SQLJ の使用: 判明している制限事項と問題点

Oracle8i リリース 8.1.7 サーバー側 JavaVM の一部である SQLJ トランスレータに関するその他の制限事項と相違点について説明します。

この内容は、クライアントとサーバーの両方で同じコードをコンパイルする場合に役立ちます。Oracle8i JVM 以外でソースをコンパイルし、コンパイルしたファイルを `loadjava` でアップロードすることにより、後述の制限事項 (Bug# 1094935) の例外) も回避できます。

重要: サーバー側 JavaVM は JDK 1.2 を実装しているため、JDK 1.2、Oracle8i リリース 8.1.7 JDBC ドライバ・ライブラリ `classes12.zip` および SQLJ ランタイム・ライブラリ `runtime12.zip` を使用し、クライアントのプログラムをコンパイルおよびテストすることをお勧めします。これにより、クライアントの環境とサーバー側の環境の対応が最高に達します。

言語

- ネストされたインタフェース宣言はサポートされません (Bug# 1049817)。たとえば、次の宣言はサポートされません。

```
interface Nested { #sql static iterator NestedCur(int); }
```

- WITH 句の宣言はサポートされません (Bug# 1049818)。
- ブロック・レベルでのクラス宣言は許可されません (Bug# 1034704)。この宣言は JDK 1.2 では許可しています。

トランスレータ

- 重複するクラス宣言が発生すると、SQLJ は、クラス循環エラーではなく、スタック・トレースを発行する場合があります (Bug# 1018565)。
- サーバー側の SQLJ トランスレータは、実際のソース・ファイル名の一部に、ソースの先頭にある公開クラス名、(または公開クラスがない場合は、先頭のクラス名) をエラー・メッセージに表示します (Bug# 1018533)。
- loadjava を使用して char[][]などの配列を持つクラスをアップロードすると、Oracle8i JVM Verifier エラーが発生することがあります (Bug# 1094935)

第 6 章 Oracle8i JVM、EJB / CORBA および Oracle Servlet Engine

原典情報 : \$ORACLE_HOME/relnotes/javavm/README_javavm.txt ファイル

Oracle のエンタープライズ Java サーバー・プラットフォーム : Oracle8i JVM

リリース 8.1.7 には、Sun の Java Development Kit (JDK) 1.2.1 の Java クラス・ライブラリに加え、Java 仮想マシン (VM) の完全な機能が含まれています。Oracle の JDBC、SQLJ、Enterprise JavaBeans (EJB)、Aurora/ORB、Oracle Servlet Engine (OSE) とネイティブ・コンパイルの実装内容を組み合わせることによって、今回のリリースは、サーバー・ベースの Java アプリケーションの開発と配布のためのエンタープライズ・クラス・プラットフォーム、つまり Oracle8i JVM を提供します。

ドキュメント

今回のリリースの主な Java 関連コンポーネントについては、ハードコピーとオンライン・ドキュメントをそれぞれ提供しています。

- 『Oracle8i Java 開発者ガイド』
- 『Oracle8i Java Tools リファレンス』
- 『Oracle8i JavaServer Pages 開発者ガイドおよびリファレンス』
- 『Oracle8i Enterprise JavaBeans 開発者ガイドおよびリファレンス』
- 『Oracle8i CORBA 開発者ガイド』
- 『Oracle8i Oracle Servlet Engine ユーザーズ・ガイド』
- 『Oracle8i JDBC 開発者ガイドおよびリファレンス』
- 『Oracle8i SQLJ 開発者ガイドおよびリファレンス』
- 『Oracle8i JPublisher ユーザーズ・ガイド』
- 『Oracle8i Java ストアド・プロシージャ 開発者ガイド』

ドキュメントや判明している問題も含め、Oracle Java 製品に関する最新情報は、<http://www.oracle.com/java> を参照してください。

『Oracle8i Java 開発者ガイド』には、Java 全製品の概要が説明されており、Oracle8i の Java の機能を知るためのスタート・ガイドになっています。このマニュアルには、JDBC、SQLJ、EJB/CORBA などの個々の製品に展開されている Oracle 固有の Java の詳細も含まれています。

各分野の詳細情報は、前述のドキュメントと該当する README ファイルを参照してください。JVM、CORBA、EJB、OSE および NComp に関する情報は、この README ファイルにあります。Java Server Pages は、個別の README ファイルが、`$ORACLE_HOME/jsp/doc/releasenotes_11000` にありますので、ご注意ください。

開始方法

最初に『Oracle8i Java 開発者ガイド』に示されているサンプルに従うことをお勧めします。`$ORACLE_HOME/javavm/demo.tar` (または `demo.zip`) のデモおよびサンプルには、HelloWorld のサンプルが含まれています。このサンプルまたは他のサンプルを実行することによって、インストールが正しく完了し、Oracle8i JVM が使用中のデータベースで適切に対応していることを確認できます。

構成

『Oracle8i Java 開発者ガイド』には、Java 対応データベースのインストール方法と構成方法の詳細が示されています。この README の「判明している問題」の「メモリー使用量」の項も参照してください。

Java の使用とパフォーマンスに影響を与える `init.ora` ファイルの主なパラメータは、「`shared_pool_size`」と「`java_pool_size`」です。

「`shared_pool_size`」パラメータに指定したメモリーは、`loadjava` の使用時に一時的に使用されます。データベースの初期化プロセスでは、インストールされているシード・データベースではなく、クリーンなデータベースに対して `initjvm.sql` が実行されます。したがって、約 8000 クラスの Java バイナリをロードし、解決するために、「`shared_pool_size`」パラメータを 50MB に設定する必要があります。`shared_pool` のリソースは、コールの仕様を定義および実行時にロードされた Java クラスをシステムで動的に追跡するときにも使用されます。

Java のランタイム実行時における他のすべての Java 状態は、「`java_pool_size`」パラメータを使用して割り当てたメモリー量から取得されます。このメモリーには、Java メソッドとクラス定義の共有メモリー内表記の他に、コール終了時にセッション領域に移行した Java オブジェクトも含まれています。前者の場合、メモリー・コストはすべての Java ユーザーの間で分担されます。後者の場合は、MTS モードで、各セッションの静的変数に保持されている実際の状態量に基づいて、「`java_pool_size`」パラメータの割当てを調整する必要があります。

Sun の JDK と同様に、Java コンパイラ（Java ソース・コードを Java バイナリにコンパイルする、Java で記述されたコンパイラ）は、メモリーの大量コンシューマとして知られています。Oracle8i JVM 上にある大量の Java ソースをコンパイルする場合、または複雑な EJB（サーバー上のコンパイラを使用する）を配布する場合は、「java_pool_size」パラメータの増加を考慮してください。「java_pool_size」パラメータのデフォルト値は 20MB です。

注意： シード・データベースは、Net8 クライアント用の専用サーバーおよび CORBA / EJB 用の MTS を使用するように構成されています。MTS は、CORBA / EJB (IIOP 接続) には必要ですが、Net8 クライアントには必要ありません。インストールおよび構成の詳細は、『Oracle8i Net8 管理者ガイド』を参照してください。

Java 互換性

今回のリリースは、JDK 1.2.1 に対する Sun の Java 互換キットで徹底してテストされています。『Oracle8i Java 開発者ガイド』で説明しているように、互換性に関するプラットフォーム固有の例外は、ユーザー・インタフェースの実現方法のみです。

Java 開発者が Oracle Server 内でユーザー・インタフェースのコンポーネントを表示可能にすることは適切ではありません。ユーザー・インタフェースとは、サーバー上で Oracle と対話するユーザーの存在を示すものです。一般的に、Oracle の実行中にはそのようなユーザーは存在しないため、任意のユーザー・コードを許可して、応答が必要な可能性のあるユーザー・インタフェースを作成することは適切ではありません。この制限は、ユーザー・インタフェースを実現しようとするコードをサーバー上で実行しない限り、AWT または Swing クラスを参照または使用する Java コードの記述を妨げるものではありません。

Oracle は、Java と他のインターネット標準のペースに合せた Oracle8i JVM をコミットしています。

互換モード

Oracle8i で Java を使用するには、次の設定が必要です。

```
compatibility = 8.1.0
```

または、必要に応じてそれ以降を指定します。

CORBA と Enterprise JavaBeans

Oracle8i リリース 8.1.7 では、統合化された ORB と Enterprise JavaBeans (EJB) に対するサポートを提供しています。リリース 8.1.7 における CORBA または EJB オブジェクトの配布に関する詳細は、ハードコピーまたはオンラインによるドキュメントを参照してください。

Oracle8i リリース 8.1.7 では、Oracle ツールによる CORBA サーバー・オブジェクトの作成と配布をサポートするために、Java 用の Visibroker の開発盤バージョンとランタイム・バージョンが統合されています。

Oracle Tools 以外または Oracle 製品以外での、Java 用の Visibroker を使用したスタンドアロン CORBA オブジェクトの開発と配布はサポートされません。Inprise 社から有効な Visibroker のライセンスを取得してください。

判明している問題

VM 自体、Java スタッド・プロシージャ、および大半の Java 開発者、loadjava および dropjava が使用するユーティリティで判明している問題を説明します。EJB および CORBA での問題も含まれています。SQL と Java との対話、特に SQL と Java との型変換のサポートに関して判明している問題は、JDBC の README ファイルで説明されています。

メモリー使用量

Java は、開発時と実行時ではメモリー使用量に相当な差異があることが公表されていますが、このことは、提供されているインストール時の構成支援には反映されていません。具体的には、インストール構成は「Minimum」、「Typical」および「Custom」インストールに制限されていますが、今回のリリースでは、インストール時に Java をどのように使用するかを判断するための問合せ方法がありません。

Oracle8i JVM アプリケーションの開発時には、実行時の Java アプリケーションとは異なる方法でメモリーを使用する loadjava などのツールを使用します。これらのツールは、開発時のみのツールで、実行時のアプリケーション環境には関係ありません。『Oracle8i Java 開発者ガイド』で説明されているように、クライアント側の IDE を使用した開発スタイルを選択するか、または Oracle8i JVM 上の Java コンパイラを使用するように選択できます。Sun の JDK コンパイラに対応しているこのコンパイラは、メモリーを集中的に使用するため、インストールされている構成の値から「java_pool_size」パラメータを大きくする必要があります。

次の経験則が適用できます。

- クライアント上でコンパイルしてサーバーにロードする場合よりも、サーバー上でコードをコンパイルする場合の方が、より多くの「java_pool_size」が必要です。EJB の配布にはサーバー上の Java コンパイラを使用するため、大きな「java_pool_size」が必要です。
- 多数のクラスをサーバーにロードして解決するには、大きな「shared_pool_size」が必要です。
- MTS では、セッションの数とコール終了時に Java の静的変数に保持されているメモリー量に基づいて、「java_pool_size」を増加する必要があります。
- 実行時のランタイム・メモリー要件は、Java 開発時に必要なメモリーとはかなり異なります。Java 開発を行う場合は、次の設定をお薦めします。

```
shared_pool_size = 50000000  
java_pool_size = 20000000
```

これらの値は、Oracle8i JVM の「Custom」インストールに対するデフォルトのインストール値です。

Java アプリケーションを配布する場合のメモリーの要件は、アプリケーション自体とそのアプリケーションを同時に実行するユーザーの数によって異なります。

initjvm.sql 実行時のメモリーまたはリソースの不足

initjvm.sql が正しく実行されると、Java スキーマ・オブジェクトの合計数は 8,000 を超えます。次のようにして確認できます。

```
connect internal  
select count (*) from user_objects where object_type='JAVA CLASS';
```

すべてのクラスが有効にマークされ、無効なクラスの数はいくらにもなります。次のようにして確認できます。

```
select count(*) from user_objects  
where object_type='JAVA CLASS' and status != 'VALID';
```

initjvm に失敗した場合は、init.ora ファイルをチェックして、「shared_pool_size」と「java_pool_size」がそれぞれ少なくとも 50MB と 20MB であるかどうかを確認してください。Java 対応データベースで開始していることを前提とした、最小または標準インストールを選択した場合は、これらの値が不適切に設定されている可能性があります。

コンパイル時のメモリー不足

loadjava でコンパイラを使用中、または deployejb の途中でメモリーが不足すると、次のエラーが表示されます。

```
A SQL exception occurred while compiling:  
<Java class that was being compiled> : ORA-04031:  
unable to allocate <some number of> bytes of shared memory  
("shared pool","unknown object","joxlod: init h",  
"JOX: ioc_allocate_pal")
```

回復するにはデータベースを一度シャットダウンして、「java_pool_size」パラメータを大きな値に再設定します。エラー・メッセージにある「shared pool」（共有プール）は、誤解を招く恐れがありますが、「shared_pool_size」パラメータではなく、「java_pool_size」パラメータを設定し直してください。

Java クラス・ロード時のメモリー不足

クラス・スキーマ・オブジェクトの作成を処理する場合に「shared_pool_size」パラメータが小さすぎると、エラーの表示なしにロードが失敗し、データベースに無効なクラスが残ります。以降のランタイム実行時に、システムは無効なクラスの解決を試みます。正常に解決しない場合は、実行時に「ClassNotFoundException」または「NoClassDefFoundException」が表示されます。サーバーへのクラスのインストールに失敗しているため、これらの実行時エラーの発生は妥当であることに注意してください。これらの例外は、必ずしもロード中に「shared_pool_size」パラメータに問題があったことを意味するものではありません。サーバーにロードするセットに、クラスが含まれていたことを常に確認し、loadjava の「-force」オプションを使用して、サーバーに常駐しているクラスを、ロード中のクラスで強制的に置換することを考慮してください。

loadjava の「-resolve」オプションを使用して、クラスのロード時に loadjava によるクラスの解決を試みるようにしてください。解決に失敗すると、通常、loadjava によってエラー・メッセージが出力されます。これらのメッセージがメモリーの問題または「接続が切断されました」などの失敗を示す場合は、「shared_pool_size」パラメータおよび「java_pool_size」パラメータのサイズを大きくして、再度試行してください。

loadjava を使用して作成したクラスの状態を再確認するためには、そのクラスが含まれているスキーマのデータベースに対して接続を行い、次を実行してください。

```
select * from user_objects where object_name =  
dbms_java.shortname('<クラス名>') ;
```

user_objects の列の中に状態があります。

サーバーに常駐する Java ソースのアクセス可能性

以前の Oracle8i JVM (JServer) リリースでは、PL/SQL に使用されていた標準的な手段では、サーバーに常駐する Java ソースにアクセスすることはできませんでした。リリース 8.1.7 では、Java ソース・スキーマ・オブジェクトを作成すると、`user_source` を使用してソースを表示できます。具体的には、次を実行すると Java ソースが表示されます。

```
select distinct type from user_source;
```

ただし、Java バイナリ規格では、クライアントで Java コードをコンパイルし、バイナリのみをサーバーにロードできることに注意してください。これは、Oracle8i JVM (JServer) を使用して Java を開発する場合にお薦めできるアプローチです。

Java との DESCRIBE の問題

メソッドに PL/SQL ではなく Java の実装があると、`package.procedure` の DESCRIBE が「オブジェクトは存在しません」を表示します。パッケージの DESCRIBE には、PL/SQL プロシージャのみがリストされ、Java ストアド・プロシージャは除外されます。

Loadjava と Dropjava

Loadjava と Dropjava は Java ベースのユーティリティで、Java バイナリ、ソースまたは Jar ファイルをサーバーにロードし、対応するライブラリ・ユニットを削除するために、クライアント側で実行します。詳細は『Oracle8i Java 開発者ガイド』を参照してください。

- ソースとバイナリを 1 つの jar ファイルにロードしないでください。同一クラスのソース・ファイル (.java) とバイナリ・ファイル (.class) は、同じ jar ファイルにはロードできません。
- Java の重要な一面は、バイナリ規格をサポートしていることです。Java バイナリ規格では、開発はクライアント側で、サーバーにロードできるのは Java バイナリのみです。結果のバイナリのロードには、クライアント側のコンパイラと loadjava を使用した方が、ソースで loadjava を使用するより効率的です。
- dropjava でパッケージを削除できるのは望ましいことですが、そのような機能はありません。
- 大きな jar ファイルをロードするには、`shared_pool` の追加が必要です。Java スキーマ・オブジェクトの作成は、「`shared_pool_size`」パラメータに指定されているメモリーを使用します。大きな jar ファイルのロードで問題が発生した場合は、問題を回避するために `SHARED_POOL_SIZE` の値の増加を考慮してください。

loadjava でのクラスのロードと解決に問題があると思われる場合は、クラスをロードしたユーザー（scott/tiger など）で接続し、ロードされているクラスの状態を確認できます。次のように実行します。

```
select * from user_objects;
```

正しくロードされ、解決されたクラスは、「VALID」（有効）のステータスで表示されます。クラスが「VALID」（有効）かどうかを簡単に判断するには、次を実行します。

```
select dbms_java.longname(object_name) from user_objects
where object_type = 'JAVA CLASS' and status != 'VALID';
```

エラー・レポート

エラーがトレース・ファイルにレポートされていることがよくあります。問題を整理する最初のステップとして、このトレース・ファイルを調べてください。

java.version、oracle.jserver.version および oracle.server.version

リリース 8.1.7 では、JDK デフォルトに対応するために、java.version プロパティに「1.2.1」が戻されています。したがって、コードがクライアントではなく Oracle8i JVM 上で実行されているかどうかの判断に、java.version の Oracle の存在を使用している場合は、oracle.jserver.version プロパティが NULL でないかどうかをテストするように変更してください。（リリース 8.1.5 の場合、java.version の値は「Oracle Server 1.1.6」でした。リリース 8.1.6 では、java.version の値は「Oracle Server 1.2.1」です。）現在、oracle.server.version プロパティは廃止されています。このプロパティはリリース 8.1.5 に設定されていました。互換性のみの理由から、リリース 8.1.6 にも設定されています。

SQL*PLUS の Java &&

SQL*PLUS は、Java &&を置換変数として解析します。これを回避するには、loadjava を使用してソースをロードするか、loadjava を使用してバイナリをロードします。SQL*PLUS で Java ソースを記述する場合は、最初に「set define off」を記述してください。

RMI とソケット

今回のリリースにおける O/S リソース、ソケットおよび RMI サポートの情報は、『Oracle8i Java 開発者ガイド』を参照してください。RMI は完全にサポートされていますが、拡張性のあるサーバー・アプリケーションを構築するためには、RMI を使用する上での影響を理解する必要があります。コール終了以降のソケットのサポートは、『Oracle8i Java 開発者ガイド』で説明されています。

マルチバイト文字出力の 255 バイトでの切捨てについて (Bug#895586)

`dbms_java.set_output` を使用して `System.out.println()` を標準出力にリダイレクトする場合に、その 255 バイト目がマルチバイト文字の一部である場合、この文字は出力では 2 つの 1 バイト文字に分割されます。これを回避するには、出力を複数の小さい単位に分割します。

Java から SQL、および SQL から Java へのコール

SQL コールインの制限に関する詳細は、JDBC README を参照してください。あるデータ型を使用すると、Java と SQL との間の変換に問題が生じることが判明しています。総合的に見ると、この問題は、JDBC ではなく Aurora のランタイム機構に起因しています。しかし、大半の JDBC ユーザーがこの問題に直面することから、この問題は JDBC README (`$ORACLE_HOME/jdbc/Readme.txt`) に記述しています。

Java と PL/SQL

PL/SQL と Java との対話の詳細は、『Oracle8i Java ストアド・プロシージャ開発者ガイド』を参照してください。コール仕様が提供するメソッド・シグネチャと、実行時に起動されるメソッドとのチェックが定義時に行われないことは、コール仕様定義の既知の機能です。設計上は、不一致は実行時にのみ検出されることになっています。

EJB/CORBA - CLASSPATH、LD_LIBRARY_PATH および PATH

`idl2java` コンパイラを実行、または EJB を配布するには、環境変数 `PATH` にある JDK から `java` コマンドおよび `javac` コマンドが必要です。JDK 1.1.3 以降を使用できます。Java2 (JDK 1.2.x) はサポートされますが、必要ではありません。

```
PATH -> jdk1.1.x/bin
```

または

```
PATH -> jdk1.2.x/bin
```

CORBA オブジェクトまたは EJB をコンパイルおよび実行可能にするには、環境変数 `CLASSPATH` にある次の 3 つの jar ファイルが必要です。

```
CLASSPATH -> ORACLE_HOME/lib/aurora_client.jar
             ORACLE_HOME/lib/vbjapp.jar
             ORACLE_HOME/lib/vbjorb.jar
```

JDK 1.1.x を使用する場合は、次のファイルも必要です。

```
CLASSPATH -> jdk1.1.x/lib/classes.zip
```

SSL 接続を介して CORBA オブジェクトまたは EJB にアクセスする場合は、Net8 のドキュメントの説明に従って、GIOP SSL 用のデータベースを構成する必要があります。環境変数 CLASSPATH にある次の jar ファイルおよび Solaris の場合は、環境変数 LD_LIBRARY_PATH に \$ORACLE_HOME/lib が必要です。

JDK 1.1.x の場合：

```
CLASSPATH->ORACLE_HOME/jlib/javax-ssl_1.1.jar:ORACLE_HOME/jlib/jssl-1_1.jar
```

JDK 1.2.x の場合：

```
CLASSPATH->ORACLE_HOME/jlib/javax-ssl_1.2.jar:ORACLE_HOME/jlib/jssl-1_2.jar
```

いずれの場合も、環境変数 LD_LIBRARY_PATH に \$ORACLE_HOME/lib が必要です。

```
LD_LIBRARY_PATH -> ORACLE_HOME/lib
```

サーバー・オブジェクトが、DbmsJava などの oracle.aurora.rdbms パッケージのクラスを使用している場合は、次の Zip ファイルを環境変数 CLASSPATH に追加して、クライアント側でそれらのクラスをコンパイルできるようにする必要があります。

```
CLASSPATH -> ORACLE_HOME/javavm/lib/aurora.zip
```

この資料（環境変数 CLASSPATH の設定など）は、EJB および CORBA のサンプル・プログラムに関する個別の README ファイルでもカバーされています。

CORBA と EJB の SSL 設定

IIOP を使用し、SSL を介して Oracle8i JVM に接続する場合は、ドキュメントの説明に従って SSL 用のデータベースを構成する必要があります。要約は、次のとおりです。

1. init<SID>.ora で、次の内容に該当する行のコメントを外します。

```
mts_dispatchers="(PROTOCOL=TCPS) (PRE=oracle.aurora.server.SGiopServer) "
```

2. \$TNS_ADMIN/listener.ora に次の行を追加します。

```
(LISTENER = (DESCRIPTION_LIST=
...
  (DESCRIPTION =
    (ADDRESS=
      (PROTOCOL=tcps)
      (HOST=<ホスト名>)
      (PORT=2482)
    )
    (PRESENTATION=GIOP)
  )
...
))
```

クライアント側の認証の場合は、次の行のコメントを外してください

```
#SSL_CLIENT_AUTHENTICATION= TRUE
```

クライアント側の認証の場合は、次の行にコメントを入れてください

```
SSL_CLIENT_AUTHENTICATION=FALSE  
oss.source.my_wallet=  
    (SOURCE=(METHOD=FILE) (METHOD_DATA=  
        (DIRECTORY=<Wallet のディレクトリ>)))
```

3. \$TNS_ADMIN/sqlnet.ora に次の行を追加します。

クライアント側の認証の場合は、次の行のコメントを外してください

```
#SSL_CLIENT_AUTHENTICATION= TRUE
```

クライアント側の認証の場合は、次の行にコメントを入れてください

```
SSL_CLIENT_AUTHENTICATION=FALSE  
oss.source.my_wallet=  
    (SOURCE=(METHOD=FILE) (METHOD_DATA=  
        (DIRECTORY=<Wallet のディレクトリ>)))
```

これらの変更を行った後は、データベースとリスナーを再起動する必要があります。

CORBA と EJB のサンプル

Oracle8i での CORBA と EJB の使用方法を示すサンプルを多数提供しています。サンプルは、\$ORACLE_HOME/javavm/demo/demo.tar ファイルにあります。

これらのサンプルは、デモを行うテクノロジー別（CORBA または EJB など）に分類されており、各ディレクトリ内では、用途別（基本的使用、セッション管理、トランザクション管理、IIOP を介した RMI など）に区分けされています。使用しているテクノロジーを提示するために、若干大きく、複雑なデモもあります。いずれの場合も、サンプルの目的や、そのサンプルをコンパイル、ロードおよび実行する Make ファイルについて説明した README ファイルが用意されています。

これらのサンプルを実行するには、前述の「EJB/CORBA -- CLASSPATH、LD_LIBRARY_PATH および PATH」項目で説明した java、javac に対する JDK コマンドが PATH に存在している必要があります。また、ORACLE_HOME、CLASSPATH、JAVA_HOME、LD_LIBRARY_PATH が正しく設定されていることも必要です。さらに、init.ora ファイル内の「java_pool_size」および「shared_poll_size」の値を確認してください。少なくとも 50MB が必要です。

次のサンプルを使用するときは、判明している問題があるため特別なステップが必要です。これ以外のすべてのサンプルは、変更なしで、個々の README ファイルの説明どおりに正常に動作します。

- corba/basic/pureCorba

このサンプルを実行する場合は、Giop をリスニングするためのデータベースを構成する必要があります。ディスパッチャを「PRE=oracle.aurora.server.SgiopServer」ではなく

「PRE=oracle.aurora.server.GiopServer」で構成し、データベースを再起動する必要があります。

- corba/transaction
- ejb/transaction

Java Transaction API (JTA) のサンプルでは、JTA トランザクション・マネージャまたは JTA データソース (あるいはその両方) を名前空間にバインドする必要があります。したがって、Make ファイルでは、複数の変数の受渡しが必要です。詳細は README ファイルを参照してください。

- corba/transaction/jta/twophase
- ejb/transaction/jta/twophase

多くの JTA 2 フェーズ・コミットのサンプルでは、Make ファイルを実行するために、データベース・リンクの設定と複数の変数の受渡しが必要です。JTA 2 フェーズ・コミットのサンプルを実行する前には、README ファイルをよく読んでください。

- ejb/ssl/*
- corba/ssl/*

これらのサンプルでは、この項目の説明に従って SSL を設定する必要があります。

EJB の制限事項

- エンティティ Bean の実装では現在、検索メソッドがコレクションを戻すことはサポートされていません。その代替方法として、列挙を使用します。次のコードを使用すると、コレクションを検索メソッドが戻す列挙に変換できます。

```
import java.util.Collection;
import java.util.Enumeration;
import java.util.Iterator;
public class CollectionEnumeration implements Enumeration {
    Collection coll;
    Iterator it;

    public CollectionEnumeration (Collection coll) {
        this.it = coll.iterator();
    }
    public boolean hasMoreElements() {
        return it.hasNext ();
    }
    public Object nextElement() {
        return it.next ();
    }

    public Collection getCollection () {
        return coll;
    }
}
```

- XML ディプロイメント・ディスクリプタには複数の Bean を指定できますが、配布は現在、1 回に 1 つの Bean のみです。数個の Bean を含んだディプロイメント・ディスクリプタがある場合は、そのディプロイメント・ディスクリプタを 1 つの Bean のみに関する情報を含んだ数個の XML ディプロイメント・ディスクリプタ・ファイルに分割する必要があります。
- コンテナ管理の永続 Bean を配布する `deployejb` の実行時に、次のエラーが発生する可能性があります。

```
An exception occurred during code generation: null
```

エラーが発生した場合は、標準ディプロイメント・ディスクリプタ内の `<ejb-name>` に対応して、Oracle 固有のディプロイメント・ディスクリプタ内の `<ejb-name>` が指定されていることを確認してください。

- #1334001 EJB ディプロイメント・ディスクリプタ内のアクセス制御リストに対してロールを指定し、そのロールをユーザーに付与しても、ユーザーには依然として Bean へのアクセス権がありません。

- #1379854 EJB ハンドルを保存し、後で再利用しようとしても機能しません。
- #1378775 XML ディプロイメント・ディスクリプタには、Bean に対するデフォルトのトランザクション属性を指定する必要があります。

```
<container-transaction>
<description>no description</description>
  <method>
    <ejb-name>/test/purchase</ejb-name>
    <method-name>*</method-name>
  </method>
  <trans-attribute>Required</trans-attribute>
</container-transaction>
```

- クライアントが remove メソッドの起動後に、EJBObject を使用し続けると、指定した javax.ejb.ObjectNotFoundException のかわりに、例外 CORBA.OBJECT_NOT_FOUND が発生します。
- deployejb は、引数のない ejbCreate メソッドを持たない Bean の配布を誤って拒否します。引数なしの ejbCreate を EJB に配置してください。このメソッドは、引数なしの create メソッドがホーム・インタフェースにもある場合に、コールされます。

```
public class myBean extends SessionBean {
    ...
    void ejbCreate(){}
}
```

- 次を実行すると、同一セッション内の Java ストアド・プロシージャまたはサーブレットから EJB をコールできます。

```
InitialContext ic = new InitialContext();
HelloHome = (HelloHome)ic.lookup("/test/helloworld");
.....
```

ただし、この使用法は、セッション同期を実装していないセッション Bean をコールする目的のみに制限されます (Bug#1390678)。Java ストアド・プロシージャまたはサーブレットからエンティティ Bean をコールする場合は、初期コンテキストを作成して、クライアントから行うように検索を実行する必要があります。これによって、EJB を実行するための第 2 のデータベース・セッションが作成されます。

CORBA または RMI/IIOP を介しても純粋な SQL または HTTP セッションにはアクセスできないため、ストアド・プロシージャまたはサーブレットを実行しているセッションに RMI/IIOP のコールバックを行うことはできません。

- ホーム・インタフェース、リモート・インタフェースおよび Bean 実装クラスがパッケージにない EJB は配布できません (Bug#951186)。Bean インタフェースとクラスは、常にパッケージに入れてください。
- #1268913 リモート・インタフェースに java.awt.color.ColorSpace が含まれていると、deployejb がエラーになります。

CORBA の制限事項

- SSL IIOP リスナーとディスパッチャは、シード・データベースに構成されていません。『Oracle8i Net8 管理者ガイド』の説明に従って、ユーザー自身で SSL を構成する必要があります。
- サーバー・アプリケーションは、パラメータ付きまたはパラメータなしで ORB.init() をコールして、ORB の新しいインスタンスをいつでも取得できます。通常、これは必要ではありません。

Oracle8i JVM が作成した ORB インスタンスへのアクセスが必要な場合は、ORB インスタンスを常に 1 個ずつ戻すメソッド `oracle.aurora.jndi.orb_dep.Orb.init()` を使用してください。このインタフェースは、アプリケーション・インタセプタのインストールも適切に処理します。

- データベースが共有パスワード・ファイルを使用するように構成されていて、データベースの SYS パスワードがパスワード・ファイルの SYS パスワードと異なる場合は、CORBA または SQL*PLUS を介して SYS で接続できません。ALTER USER.. コマンドを使用して、パスワードを再度一致させます。
- セッション IIOP ではなく、標準の IIOP 用にデータベースを構成した場合は、CORBA または EJB クライアントの InitialContext に、次のプロパティを配置する必要があります。

```
import oracle.aurora.jndi.sess_iiop.ServiceCtx;
...
env.put("TRANSPORT_TYPE", ServiceCtx.IIOP);
```

さらに、「PRE=oracle.aurora.server.SgiopServer」ではなく、「PRE=oracle.aurora.server.GiopServer」を指定してディスパッチャを構成し、対応する sess_iiop の URL を使用する必要があります。

提供しているすべてのツール (publish、deployejb、sess_sh) は「-iiop」オプションをサポートしており、これによりツールが純粋な IIOP を使用します。

「PRE=oracle.aurora.server.GiopServer」で設定されているディスパッチャに接続する場合は、この引数を指定します。

- /etc/IFR のセッション参照では機能しません (Bug#1359098)。

SSL の制限事項

- あるサーバーから別のサーバーに SSL コールアウトを行う場合に SSL クライアント側の認証を使用することは、今回のリリースではサポートされません。
- 今回のリリースでは、SSL_CREDENTIAL ログインはサポートしていません。たとえば、次のコードは機能しません。

```
env.put (Context.SECURITY_AUTHENTICATION, ServiceCtx.SSL_CREDENTIAL)
```

かわりに、次のコードを使用します。

```
env.put (Context.SECURITY_AUTHENTICATION, ServiceCtx.SSL_LOGIN)
```

リリース 8.1.6 との EJB の相違

Oracle8i リリース 8.1.7 では、EJB 1.1 をサポートしています。リリース 8.1.6 のテキスト・バージョンのかわりに、XML ディプロイメント・ディスクリプタを使用できます。Bean 管理の永続性とコンテナ管理の永続性の両方のエンティティ Bean もサポートしています。新規の CMP は、その永続性ソリューションに Persistence Service Interface Reference Implementation を使用します。これは、Bean を表にマップする非常に単純な永続レイヤーで、多くの制限事項があります。永続性をサポートするための複雑で高度な機能セットは、JDeveloper 3.2 を参照してください。

今回のリリースでは、2 フェーズ・コミット機能を含めた JTA を完全にサポートしています。

クライアント側のツールの動作方法を設計し直したため、リリース 8.1.6 で動作する以前のツール名を「toolname_816」と改名してあります。これは、リリース 8.1.7 クライアントを使用してリリース 8.1.6 サーバーに通信する場合の下位互換性を維持するためのものです。次のツールがこれに該当します。

- sess_sh_816
- publish_816

クライアント・ツール sess_sh の構文が変更されました。http、sess_iiop および jdbc の 3 つの異なるプロトコルを使用して、サーバーの名前空間にアクセスできるようになりました。詳細は、関連するドキュメントを参照してください。

リリース 8.1.5 またはリリース 8.1.6 からリリース 8.1.7 へのアップグレード

リリース 8.1.5 またはリリース 8.1.6 からリリース 8.1.7 にアップグレード時に、PL/SQL の XML パーサーおよび Java のクラス・ジェネレータがリリース 8.1.7 データベースにロードされません。PL/SQL の XML パーサーを使用するには、`$ORACLE_HOME/lib/xmlplsql.jar` をロードして、`$ORACLE_HOME/xdk/plsql/parser/bin/load.sql` を実行する必要があります。Java のクラス・ジェネレータを使用するには、`$ORACLE_HOME/lib/classgen.jar` をロードする必要があります。

CD から利用可能なパッチ

- C++クライアント ORB に対するログイン・オブジェクトの C++バージョンを使用できます。このパッチは、`$ORACLE_HOME/jis/patch/c++`にあります。詳細は、該当する README ファイルを参照してください。
- #1169155 以前に、リリース 8.1.6 クライアントが CORBA または EJB オブジェクトをリリース 8.1.6 サーバーに発行していた環境で、サーバーをアップグレードしようとする、この互換性の問題が発生します。リリース 8.1.7 サーバーと引き続き通信を行うためのこのパッチは、リリース 8.1.6 クライアントの環境変数 `CLASSPATH` に含まれています。ただし、リリース 8.1.7 に CORBA オブジェクトを公開するか、または EJB を再配布する場合は、リリース 8.1.7 のツールとライブラリを使用する必要があります。このパッチは、`$ORACLE_HOME/jis/patch/816`にあります。

Oracle Servlet Engine

Oracle Servlet Engine の概要

初回リリースとなる今回の Oracle Servlet Engine (OSE) は、Oracle8i リリース 8.1.7 の一部です。OSE はサーブレット仕様のバージョン 2.2 に準拠しています。OSE は、データベース内で実行する Java サーブレット・エンジンです。OSE は、HTTP 要求を Apache モジュールからの接続を介して受け取るか、または登録したエンドポイントを介して、データベース自体に直接受け取ります。OSE には、静的な内容を提供する機能の他に、高度に拡張可能な方式でステートフル・サーブレットを提供するように作成されています。OSE は、『Oracle8i Java Tools リファレンス』に説明されているように、セッション・シェルのツールでアクセスできる JNDI 名前空間から、その構成情報を導出します。

今回のリリースには、Apache インストールと OSE への接続に使用する Apache `mod`、`mod_ose` が含まれています。Apache が提供する静的な内容の要求および `mod_ose` を介して OSE に転送するステートフル・サーブレットの要求には、Apache を介して OSE を使用することをお勧めします。

詳細は、『Oracle8i Java Tools リファレンス』および『Oracle8i Oracle Servlet Engine ユーザーズ・ガイド』を参照してください。

管理サービスは、OSE インストールの一部としてインストールされます。このサービスは OSE の管理目的でのみ使用してください。このサービスは、次の手順を実行することにより、ポート 8080 を使用して利用可能になります。

```
cd $ORACLE_HOME/jis/install
sess_sh -s jdbc:oracle:oci8:@ -u internal/knl_test7
@serverendp.ssh admin 8080 9090 -register
exit
```

Web サーバーのデモを実行する前に、前述の手順を実行する必要があることに注意してください。その他のサービスの作成と構成方法を示すサンプルは、`$ORACLE_HOME/javavm/demo/examples/web/service` にあります。前述の手順を実行すると、`http://<マシン名>:8080/` から管理サービスにアクセスできます。このインストールの一部として、『Oracle8i Java Tools リファレンス』を `http://<マシン名>:8080/doc/index.htm` からアクセスできます。

管理サーバーは、`OSE$HTTP$ADMIN` ユーザーで実行します。セキュリティ上の目的から、管理ドメインはランダムなパスワードで出荷されています。管理サーバーの管理には、`OSE$HTTP$ADMIN` のパスワードを設定する必要があります。

```
sqlplus> alter user ose$http$admin identified by foo;
```

「foo」はパスワードです。

SYS で実行すると、要求されたサーブレットを公開でき、管理サーバー内の必要なサーブレット・コンテンツのメンテナンスも可能です。ただし、管理サービスは `OSE$HTTP$ADMIN` ユーザーで実行するため、ドメイン管理下にあるアクセス可能なすべてのサーブレットとサーブレット・コンテンツに対して、`OSE$HTTP$ADMIN` ユーザーへの読み込みアクセス権を付与する必要があります。

一般的な OSE 管理

- `Apache/mod_ose` を使用しないで OSE に直接通信する場合は、MTS 対応のデータベース（汎用 MTS ディスパッチャと MTS サーバーを備えたデータベース）が必要です。
- `addendpoint` ツールを使用してデータベースを動的に構成し、ユーザーが指定したポートで HTTP 着信要求をリスニングします。
- `mts_dispatchers` 宣言を使用して、Web サーバーのエンドポイント定義を `init.ora` ファイルに宣言することもできます。ただし、名前空間構成の一貫性のために依然として `addendpoint` を使用する必要があるため、`-register` フラグは使用しないでください。

大量ロードの環境では、次の構成が参考になります。

- `init.ora` の `SESSIONS` パラメータには、次の値が必要です。
 - * 特定時点でのデータベース・セッションの合計数
(1 つのデータベース・セッションには、複数の HTTP セッション・オブジェクトを保有できます - 『Oracle8i Oracle Servlet Engine ユーザーズ・ガイド』を参照してください。)
- `JAVA_POOL_SIZE = 1MB`
 - * 特定時点でのデータベース・セッションの合計数
- `mod_ose` と `TCPS` プロトコルを使用して大量ロードを行う場合は、次のように、複数の `TCPS` ディスパッチャの使用およびディスパッチャのキュー・サイズを増やすことをお勧めします。

```
mts_dispatchers="(address=(protocol=tcps) (queuesize=128)) (dispatchers=2) "
```

エラー、イベントおよびアクセス・ロギングの設定方法のサンプルは、`$ORACLE_HOME/javavm/demo` のデモを参照してください。

トラブルシューティング

サーバーから接続を拒否された場合は、次のことを確認してください。

- 要求に対して正しいポートが使用されていたかどうか
- 要求に対して正しいホスト名が使用されているかどうか
- Web サービスに対して正しいポート番号が構成されているかどうか
最良の確認方法は、セッション・シェルから「`getgroup <サービス名> endpoint`」を実行し、結果を検証することです。結果は、サービス用に構成されている名前付きの各エンドポイントに対するエントリの一覧です。これらのエントリの 1 つは、「.port」で終了しています。このエントリによって、指定ポートに着信した要求をそのサービスが処理していることが示されます。

404 が発生した場合の対策

- 要求に正しいポートが使用されていたかどうかを確認します。
- 正しい `ServletContext` を指定する適切な仮想パス文字列が URL に含まれているかどうかを確認します。
- `ServletContext` 内で正しいサーブレットを指定する適切な仮想パス文字列が URL に含まれているかどうかを確認します。
- HTTP コンテンツのテストに使用している Web ブラウザのキャッシュを使用不可にします（可能な限り）。
- サーブレットの公開直後または構成情報の変更直後の場合は、アクティブ・セッションのタイムアウトまたは現行のすべてのブラウザのクローズを待機してから、新しいブラウザを開始します。

一般的な問題に対する対策

- セッション・シェルを使用して名前空間を検索し、サービス、ドメインおよびサーブレットのコンテキスト構成を検査します。
- 操作を実行しようとしている有効なユーザーに対する名前空間のアクセス権を確認します。
- エラー・ログを確認し、正常ではないことを示唆するトレース・ファイルの内容を検索します。
- 特定の開発に関するトラブルシューティングに対しては、`init.ora` の「`mts_servers`」と「`mts_max_servers`」の両方を「1」に設定してください。

ファイル・アクセス権とサーブレット・コンテキスト所有権

サーブレット・コンテキストを所有するユーザーには、誤ったエラーを回避するための、一定のファイル・アクセス権が必要です。ファイル・アクセス権が付与されず、エラー・ページがファイル・システムから読み込まれるようにサーブレット・コンテキストのエラー・コードが構成されている場合（デフォルト）は、そのコンテキストの所有者には、これらのファイルを読み込むためのアクセス権が必要です。このようなアクセス権がない場合、あるエラーが発生することによって、ファイル・アクセス権がないことを示す他のエラーが発生します。これによって、根本的なエラー・コードは隠されてしまいます。

ファイル・アクセス権を取得するには、いくつかの方法があります。不確実な方法ですが、ロール JAVAUSERPRIV をユーザーに付与するという方法があります。詳細な管理が必要な安全な環境では、Oracle8i JVM のパーミッション・コールを使用して、的確なファイルのみアクセス権を付与します。このアクセス権を付与する場合は、正規のパスを使用する必要がありますことに注意してください。正規のパスは、プログラムで検出できます。次のクラス定義では、コンパイルされて Oracle8i JVM にロードされます。

```
import java.io.*;
public class FileTest {
    public FileTest() {
        super();
    }
    public static void main(String[] argv) throws IOException {
        for (int i = 0; i < argv.length; i++) {
            File f = new File(argv[i]);
            System.out.println("file: " + f + " -> " + f.getCanonicalFile());
        }
    }
}
```

次に、セッション・シェルから次のコマンドを実行すると、サンプル・ファイル /home/scott/somefile の正規パスが出力されます。

```
java FileTest /home/scott/somefile
file: /home/scott/somefile -> /private/scott/somefile
```

次のように、/private/scott/somefile を使用して権限を付与する必要があることを意味します。

```
dbms_java.grant_permission('SCOTT','SYS:java.io.FilePermission',
    '/private/scott/somefile','read');
```

Oracle Servlet Engine (mod_ose) の Apache モジュール

mod_ose モジュールの概要

Oracle HTTP Server の mod_ose モジュール (Apache により動作) は、Apache と OSE との間の連絡役を果たします。mod_ose モジュールは、HTTP の Net8 へのトンネリングを Apache と OSE 間のプロトコルとして使用します。このため、ロード・バランシング、ファイアウォール・サポート、Connection Manager などの Net8 の機能が活用されます。各要求ごとに接続をクローズする mod_jserv とは異なり、mod_ose は、ステートフル・アプリケーションにクライアントが接続している間、OSE への接続を保持します。ステートレス接続の場合は、Apache プロセスの範囲内で接続状態が続きます。mod_ose は、有効なデータ転送に HTTP チャンク機能を使用します (mod_ose の詳細は、『Oracle8i Oracle Servlet Engine ユーザーズ・ガイド』の第 5 章を参照してください)。

構成上の問題

- `mod_ose` のデフォルト構成（デフォルト構成のサブレットに対して最低限必要な Apache ディレクティブと位置のディレクティブ）は、`$APACHE_HOME/Apache/conf` の `mod_ose.conf` ファイルにあります。このファイルは `oracle_apache.conf` に含まれており、この `oracle_apache.conf` は `httpd{s}.conf` に含まれています。OSE の JNDI 名前空間で新しいサブレットが公開されている場合は、`exportwebdomain` を使用して構成を再生成する必要があります。新規構成を有効にするためには、Apache を再起動する必要があります。
- `mod_ose` は、あらかじめインストールおよび構成されていますが、構成ファイルに修正を加えない限り、そのままでは実行されません。
 - MTS 対応のデータベースの場合、`CONNECT_DATA` 句に `PRESENTATION`、`SERVICE_NAME` および `SERVER` を設定した `tnsnames.ora` に別名があることを確認し、`mod_ose.conf` の `AuroraService` コマンドに対して（`IfModule` ディレクティブで）このエントリを使用します。次の例を参考にしてください。

`tnsnames.ora` の記述：

```
inst1_http = (DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp) (HOST=server27) (PORT=5521))
  (CONNECT_DATA=
    (SERVICE_NAME=rdbms817.rdbms.dev.us.oracle.com)
    (SERVER=shared)
    (SENTATION=http://admin)
  )
)
```

`mod_ose.conf` の記述：

```
AuroraService inst1_http
```

- MTS に対応していないデータベースの場合、`mod_ose` には実行する MTS が必要です。したがって、データベースに MTS ディスパッチャまたは MTS サーバーがないと、`mod_ose` は実行されません。次の例は、これを回避する方法です。
 - サービス名 `MODEOSE`（実際には、データベース・サービス名以外のサービス名を指定可）で、MTS ディスパッチャおよび MTS サーバーを作成します。

```
mts_dispatchers=" (PROTOCOL=tcp) (SER=MODEOSE) "
mts_dispatchers=" (PROTOCOL=tcps) (SER=MODEOSE) "
```

- TNS エントリの別名には、データベース・サービス名ではなくこのサービス名を使用します。

```
inst1_http = (DESCRIPTION=
  (ADDRESS= (PROTOCOL=tcp) (HOST=server27) (PORT=5521))
  (CONNECT_DATA=
    (SERVICE_NAME=MODESE)
    (SERVER=shared)
    (PRESENTATION=http://admin)
  )
)
```

このようにして、他のクライアントが使用しない、`mod_ose` のみが使用する MTS ディスパッチャと MTS サーバーを保持します。

mod_ose と SSL

クライアントと Apache 間の SSL 接続のように、`mod_ose` と OSE 間の SSL 接続を保持できます。この機能を動作させるには、次のことを確認してください。

- 正しい Wallet
- TCPS リスナー
- TCPS プロトコルと TCPS リスナー・ポートを持つ TNS 別名が参照する Aurora サービス

次の例を参考にしてください。

```
inst1_https = (DESCRIPTION=
  (ADDRESS= (PROTOCOL=tcps) (HOST=server27) (PORT=5524))
  (CONNECT_DATA=
    (SERVICE_NAME=rdbms817.rdbms.dev.us.oracle.com)
    (SERVER=shared)
    (PRESENTATION=http://admin)
  )
)
```

既知の問題

- 管理 Web サーバーには `index.html` がなく、かわりに `index.htm` があります。デフォルトでは、`mod_ose` を介して管理 Web サービスのルートを要求すると、Apache は `index.html` を送信するため、エラーとなります。これを解決するには、`index.html` を明示的に要求します。
- `mod_ose` を使用して `jsp` ファイルを要求した場合、そのファイルのコピーが同じパス内のローカルの Apache htdocs に存在していると、Apache の不具合のために要求が失敗します。たとえば、`$APACHE_DOCUMENT_ROOT/demo/basic` に `hellouser.jsp` がある場合は、`mod_ose` への `/demo/basic/hellouser.jsp` (`http://server27:3000/demo/basic/hellouser.jsp`) の要求は失敗します。これを解決するには、同じパスにローカル・コピーが存在しないようにします。

Oracle8i JVM Accelerator (NComp) –ネイティブ・コンパイルの Java コード

今回のリリースには、コアとなるすべての Java クラス、Aurora / ORB、JDBC および SQLJ の Java クラスがネイティブ・コンパイルの形式で含まれています。

注意: swing と awt クラスは、サーバーでの互換性が目的で含まれており、ネイティブ・コンパイルされていません。

『Oracle8i Java 開発者ガイド』に説明されているとおり、ネイティブ・コンパイル・コードは ORACLE_HOME/javavm/admin ディレクトリの共有オブジェクト・ライブラリを使用して、実行時に動的にオープンします。

ネイティブ・コンパイル・コードを使用しないように切り替える必要がある場合は、次のようにします。

```
connect internal
alter java class "java/io/Serializable" check;
alter system flush shared_pool;
alter system flush shared_pool;
alter system flush shared_pool;
```

これらの式を実行すると、実行時にクラスのロード時にネイティブ・コンパイル・コードの使用がオフになります。次の式を使用してネイティブ・コンパイル・コードの使用をオンに切り替えるまで、オフの状態が続きます。

```
alter java class "java/io/Serializable" if needed check;
```

ネイティブ・コンパイル・コードの使用をオフにするのはデバック目的か、または変則的な動作に対する他の説明理由が見つからず、動作がネイティブ・コンパイルと解析との間で一致していることを確認する場合です。今回のリリースでは、この点について不明な問題点はありません。今回のリリースに関してオラクル社が実施したすべての Java Compatibility Kit テストは、完全にネイティブ・コンパイルされたコア・クラス、ORB などを使用して実行され、それが配布されていることに注意してください。つまり、すべてのテストがネイティブ・コンパイルがオンの状態で実行されています。

アップグレード、ダウングレード、インポートおよびエクスポート

今回のリリースでネイティブ・コンパイルした Java コードは、将来の Oracle リリースに自動的にアップグレードされません。新しい Oracle リリースに移行するには、新規リリースにアップグレードしてから、アップグレードしたサーバーでコードを再度ネイティブ・コンパイルする必要があります。同様に、Oracle データベース間での Java コードのインポートとエクスポートでは、ネイティブ・コンパイルした Java コードのフォームは含まれません。このような場合は、オペレーティング・システムとプラットフォームが同一である限り、ネイティブ・コンパイルした Java コードを別の場所にインストールするために当初作成した配布 jar ファイルを再度配布します。

セキュリティ

ネイティブ・コンパイルしたコードのサーバーへの配布は、ドキュメントに説明されているように、Java 権限で保護されています。この権限の付与は、ネイティブ・コンパイルしたコードのインストールまたは供給に関与している当事者が既知で、信頼がおける場合に限って行われることを理解してください。共有ライブラリの作成に使用するコード生成プロセスは、オラクル社が提供し、広範囲に及ぶテストが行われています。ただし、解析済みの Java コードや PL/SQL プログラムとは異なり、ネイティブ・コードは、実行時に動的にオープンされる共有ライブラリから実行されます。したがって、このような共有ライブラリのインストール権限は、厳格に管理してください。

受入可能な C コンパイラのバージョン

プラットフォームの `Settings_<オペレーティング・システム>.properties` ファイルを検討し、今回リリースの Oracle8i JVM Accelerator でサポート使用するために、オラクル社が認定している C コンパイラを判断してください。オラクル社では引き続き、今回および今後のリリースに対してより多くの C コンパイラを認証を行い、関連した情報を Oracle TechNet で継続的に公開します。

既知の問題

回避する必要がある JDK リリース

Ncomp 化の目的で、ネイティブ・スレッドを持つ Solaris JDK 1.1.7p、1.2、1.2.1p および 1.1.3 の各リリースの使用は極力避けてください。これらのリリースの使用に問題があることが判明しています。

変換時における Java のメモリ不足

大容量のパッケージを変換中に、Java のメモリが不足することがあります。より多くのメモリを Java に対して強制的に割り当てることにより、この問題を回避できます。次のように指定します。

```
-ms300000000 -mx1000000000
```

`$ORACLE_HOME/javavm/jahome/Settings.properties` ファイルに定義されている `translate.command` マクロで指定します。

C コンパイル時における C コンパイラのメモリ不足

大容量のパッケージのコンパイル中に、C コンパイラのメモリが不足することがあります。各クラスを個別にコンパイルするように、非バンドル・モードに切り替えることができます。次の設定を削除します。

```
makefile.maker = $(one.c.unit.per.dll.makefile.maker)
```

対象ファイルは、次のとおりです。

```
javavm/jahome/Settings_solaris.properties
```

および

```
javavm/jahome/Settings_sunos.properties
```

フラグ-update が JDK 1.1.x では無効

JDK 1.1.x の各リリースの jar 機能では、コマンドライン・オプション「-u」が提供されないため、ncomping のコマンドライン・オプション「-update」は、これらのリリースでは動作しません。

配置 jar のディレクトリ名にピリオド（.）がある場合、配置は異常終了します。

完全パスの配置 jar 名を deploync に渡す場合、パスまたは配置 jar 内のディレクトリに「.」が含まれている場合、配置は異常終了します。

```
deploync ... /foo.bar/mydepl.jar  
deploync ... /foo_bar/my.depl.jar
```

回避策：

- 「.」をディレクトリと jar 名に使用しないようにします。
- 次のように-dを使用します。

```
deploync ... -d /foo.bar mydepl.jar
```

以前 ncomp したパッケージの半分が ncomp されていない状態になる

2 つの jar ファイルにアプリケーションがあり、パッケージ P に関しては、そのクラスの半分は最初の jar ファイルに含まれており、残りの半分のクラスは 2 番目の jar ファイルに含まれている場合、2 番目の jar ファイルを ncomp すると、最初の jar ファイルに含まれているクラスが ncomp されていない状態になります。これは、ncomp が各パッケージを基準として DLL を配置するためです。

このような状況を回避するには、指定のパッケージ P に対する全クラスが単一の jar ファイルに含まれるように、ソース jar ファイルを構成します。この構成によって、最大のパフォーマンスを取得できます。

すべてのトップレベルが同一のパッケージ `UnnamedPackage` にある

トップレベルのクラスには注意が必要です。以前に `ncomp` したトップレベルのクラス「X」があり、次にトップレベルのクラス「Y」を個別に `ncomp` すると、「X」は `ncomp` されていない状態になります。これは、すべてのトップレベルのクラスが 1 つの特別なパッケージ「`UnnamedPackage`」にあるとみなされているためです。

パッケージ P には何も存在しないため、`ncomp` は不要

サーバーに提示するクラスのないパッケージがソース jar ファイルに含まれている場合に `ncomp` を実行すると、個々のパッケージは `ALREADY_NCOMPED` であるにもかかわらず、各クラスが `NOT_FOUND_IN_SCHEMA` であるとレポートされます。

```
NOT_FOUND_IN_SCHEMA oracle/aurora/mts/iiop/ORB
NOT_FOUND_IN_SCHEMA oracle/aurora/mts/iiop/SessIiopService
...
ALREADY_NCOMPED oracle/aurora/mts/iiop/
```

この場合、`ALREADY_NCOMPED` は、「何も存在しないため、`ncomp` は不要」の意味合いを示します。

`NEED_NCOMPING` の後、`ncomp` がフリーズ

一連の `NEED_NCOMPING` をレポートした後、`ncomp` は、配布 jar ファイルを作成して配布の準備が整うまで、静止状態を保持します。マシンによっては、この遅延がフリーズしているように見えます。進行状況を確認するには、`ncomp.log` ファイルの最後尾を参照してください。-verbose を使用して画面に出力をリダイレクトする方法もあります。

Solaris 固有の「`Settings_*.properties`」ファイル

Solaris については、プラットフォーム固有の 2 種類の「`Settings_*.properties`」ファイルを提供しています。

```
Settings_solaris.properties
```

および

```
Settings_sunos.properties
```

「`Settings_solaris.properties`」ファイルは JDK 1.1.x が選択し、「`Settings_sunos.properties`」ファイルは JDK 1.2.x が選択します。

C 最適化の安全レベル

一部の C オプティマイザが、try-catch-finally を頻繁に使用する変換済み Java メソッドに対して不適切なコードを生成します。Sunpro 4.2 の「-O」レベルで開始する最適化に問題があります。したがって、Sunpro 4.2 の場合は「-xO1」レベルの使用をお勧めします。Sunpro 6.0 のコンパイラを選択している場合は、「-O」を使用できます。

Javavm / admin におけるファイル領域の管理

ncomp に対するリリース 8.1.7 のユーザー・インタフェースは、javavm / admin ディレクトリ内の DLL を管理することが目的ではありません。DLL の名前は一意であるため、javavm / admin には古い大量の DLL が時間の経過とともに蓄積される可能性があります。

指定スキーマの所定の各パッケージでは、最新の配布 DLL のみが使用されます。表 (jaccelerator\$dlls) を分析するか、または次のスクリプトを実行して、使用されている DLL を確認できます。

```
#!/bin/sh
# list of ncomp dlls in use
usage () {
    echo "libs used by ncomped classes foo/bar*"
    echo "usage: $0 foo/bar"
    exit 1
}
if [ $# -lt 1 ]; then usage; fi
DIR=`dirname $0`
cat >$DIR/DLLsReport.java <<EOJAVA
package oracle.jaccelerator.utl;
import oracle.aurora.rdbms.Schema;
import oracle.aurora.rdbms.ClassHandle;
import oracle.aurora.rdbms.Handle;
import oracle.aurora.rdbms.ObjectTypeChangedException;
import oracle.sql.CHAR;
import java.io.*;
import java.util.Date;
import java.sql.*;
import oracle.jdbc.driver.*;
public class DLLsReport {
    public static String dll (String schemaAndName)
        throws SQLException
    {
        int colonIdx = schemaAndName.indexOf(':');
        String name = schemaAndName.substring(colonIdx + 1);
        String schema = schemaAndName.substring(0, colonIdx);
        String result = "bad handle";
        ClassHandle h = Handle.lookupClass(name, Schema.lookup(schema));
        if (h == null) {
            result = "null handle" + h;
        }
    }
}
```

```

    }
    else {
        result = h.getNcompDllNameAsCHAR().toString();
    }
    return result;
}
}

EOJAVA
set -x
loadjava -oci8 -resolve -user internal/change_on_install
DIR/DLLsReport.java
svrmgrl <<EOI
connect internal/change_on_install
create or replace function isvnc_get_class_dll (pkg varchar2) return
VARCHAR2 as
    language java name
    'oracle.jaccelerator.utl.DLLsReport.dll(java.lang.String) return
    java.lang.String';
/
show errors;
select distinct isvnc_get_class_dll(OWNER || ':' ||
bms_java.longname(OBJECT_NAME))
from all_objects where OBJECT_TYPE = 'JAVA CLASS' and
dbms_java.longname(OBJECT_NAME) like '$1%' ;
EOI

```

第 7 章 Oracle Internet Directory リリース 2.1.1

原典情報：\$ORACLE_HOME/relnotes/javavm/README_oid.txt ファイル

概要

Oracle Internet Directory について

Oracle Internet Directory (OID) リリース 2.1.1 は Oracle8i によって機能強化された LDAP バージョン 3 準拠のディレクトリ・サーバーで、Oracle RDBMS テクノロジーを利用した高い拡張性と高度なデータ管理機能を実現しています。この Oracle Internet Directory リリース 2.1.1 は Oracle8i リリース 8.1.7 にバンドルされ、Oracle8i ディレクトリ対応製品のディレクトリ・ニーズを満たします。また、その設計には、適切な製品ライセンスを持つ汎用エンタープライズ・ディレクトリ・サービスまたはインターネット・ディレクトリ・サービスとして配布する目的も含まれています。

Oracle Internet Directory の 2 つのコンポーネントは、次のとおりです。

- Oracle Internet Directory サーバー リリース 2.1.1
このコンポーネントは、OID の LDAP サーバーとその関連コンポーネントすべてをインストールします。
- Oracle Internet Directory クライアント リリース 2.1.1
このコンポーネントは、OID のデータをリモートでアクセス / 管理する LDAP クライアントおよび管理ツールをインストールします。クライアント・インストールの一部としてインストールされるファイルは、サーバー・インストールの一部としてインストールされるファイルのサブセットです。

この README について

この README ファイルの内容は、Oracle Internet Directory リリース 2.1.1 と Oracle8i リリース 8.1.7 のバンドルの一部として提供される統合コンポーネントにのみ関連する内容です。

ここでは、出荷したソフトウェア（およびその統合部品）とそのマニュアルに記載されている機能上の相違点の他に、修正済みの不具合、既知の問題とその回避策が記載されており、リリース・ノートなどの代替となります。

カバー・レターおよびライセンス

Oracle8i リリース 8.1.7 の配布に添付されているカバー・レターをお読みください。Oracle Internet Directory リリース 2.1.1 のライセンス条件に関する重要な情報が記載されています。

Oracle Internet Directory リリース 2.1.1 のマニュアル

このリリースの Oracle Internet Directory で使用するマニュアルは、次のとおりです。

- プラットフォーム固有の『Oracle8i インストレーション・ガイド』
- プラットフォーム固有の『Oracle8i 管理者リファレンス』
- 『Oracle Internet Directory 管理者ガイド』
- 『Oracle Internet Directory アプリケーション管理者ガイド』
- 『Oracle8i 管理者ガイド』
- 『Oracle8i リファレンス・マニュアル』

インストール・プロセス

新規のインストール

インストールに必要な Oracle Internet Directory リリース 2.1.1 のコンポーネントの選択方法

Oracle Internet Directory サーバー リリース 2.1.1 は、リリース 8.1.7 CD-ROM の「Oracle8i Management and Integration」オプション内にあります。OID の専用データベースをインストールすることをお薦めしますが、このインストールに個別の手順を実行する必要はありません。OID のインストール・プロセスでは、専用データベースが自動的にインストールされます。OID サーバーのインストールによって、OID クライアントのコンポーネントもインストールされます。

Oracle Internet Directory クライアント 2.1.1 のコンポーネントのみをインストールするには「Oracle8i Client 8.1.7」オプションを選択します。OID クライアントのコンポーネントは、Oracle8i クライアント・インストールの「Administrator」または「Custom」オプションのいずれかを選択するとインストールされます。

Windows 固有のインストール事前要件

Oracle Internet Directory リリース 2.1.1 をインストールする前に、ORACLE_HOME が Windows システム内に設定されていないことを確認してください。ORACLE_HOME が設定されている場合、Net8 Configuration Assistant はインストーラの起動時に停止します。

既存のリリース 8.1.7 データベースへの Oracle Internet Directory のインストール

同じ ORACLE_HOME に常駐している既存のリリース 8.1.7 のデータベースを OID で使用できます。この場合、データベースはインストーラの開始前に起動され、稼動している必要があります。さらに、UTF8 キャラクタ・セットを使用している必要があります。データベース・インスタンスの識別に SID を指定するときは、大文字を使用してください。Windows へインストールする場合は、インストーラを起動する前に、環境変数 ORACLE_HOME が、前述のように一時的に設定されていないことを確認してください。

インストーラを使用したリモート・データベースへの Oracle Internet Directory インストールは不可

Oracle Internet Directory リリース 2.1.1 をリモート・データベースに対して動作させるには、リモート・マシンに加えてローカル・マシンにも OID とデータベースをインストールする必要があります。両方のインストールの完了後、管理者は、tnsnames.ora ファイルに適切な変更を行い、最初のマシンで実行中の OID を、2 番目のマシンで実行中のデータベースに接続するように構成できます。

カスタム・インストール

OID のカスタム・インストール時には、DBCA がカスタム・モードで起動されます。インストール中に以前に選択した SID とグローバル・データベース名が表示されます。これらの値は変更しないでください。変更すると、OIDCA はエラーとなります。

インストール後の構成ツール

NetCA - OID の一般的なインストールでは、一般的な構成を選択します。OID のインストール中は、Directory Service Access Configuration を実行しないでください。

Oracle 固有のスキーマのディレクトリへのインストール

すべてのインストール・タイプについて、OID インストール・プロセスは、リリース 8.1.7 バンドルに含まれる他の製品に必要な製品固有のスキーマをロードします。このスキーマは、リリース 8.1.6 バンドルで出荷された Oracle 製品との間で下位互換性があります。

LDAP スキーマのロードは、インストールの最後に自動的に実行されます。この手順を通過しない場合は、次のファイルをリストに表示されている順序でディレクトリにロードする必要があります。

- \$ORACLE_HOME/ldap/admin/oidbaseacl.ldif -> デフォルトのセキュリティ・ポリシーを実装します。
- \$ORACLE_HOME/ldap/admin/oidbase.ldif -> すべての Oracle LDAP 対応製品に必要な共通スキーマをロードします。
- \$ORACLE_HOME/ldap/admin/oidnet.ldif -> Net8 での LDAP サポートに必要なスキーマをロードします。
- \$ORACLE_HOME/ldap/admin/oidrdbms.ldif -> Oracle8i RDBMS が Oracle Internet Directory を使用するために必要なスキーマをロードします。

前述のスクリプトは、次のファイルを使用して一度に起動できます。

```
$ORACLE_HOME/ldap/admin/schema_ext.sh
```

サイレント・インストール

一般的なサイレント・インストールでは omioid.rsp の他に、NetCA の動作を制御する net8ca.rsp を使用する必要があります。omioid.rsp には、この net8ca.rsp ファイルのロケーションが必要です。カスタム・サイレント・インストールでは omicustom.rsp の他に、NetCA および DBCA に対してそれぞれ net8ca.rsp および dbca.rsp を使用する必要があります。omicustom.rsp には、これらのファイルのロケーションが必要です。

Oracle Internet Directory リリース 2.1.1 のダウングレードをサポートしません

Oracle Internet Directory リリース 2.1.1 のダウングレードをサポートしていません。

アップグレード・インストール

1. Oracle Internet Directory リリース 2.1.1 は、OID リリース 2.0.4.0.0 および 2.0.6.0.0 から
のアップグレードをサポートします。

いずれかの旧リリースからアップグレードするには、OID リリース 2.1.1 を選択し、同じ ORACLE_HOME にインストールします。インストーラがアップグレードのためのプロンプトを表示します。

プロンプトが表示された場合は、Oracle Data Migration Assistant (ODMA) によって、既存のデータベースのバックアップ・スクリプトまたはリストア・スクリプト（あるいはその両方）をアップグレード処理の初期段階に作成することをお勧めします。

Windows でのアップグレードの場合は、インストーラの起動前に、環境変数 ORACLE_HOME が一時的に設定されていないことを確認してください。

2. アップグレード・インストールを開始する前に、init<ORACLE_SID>.ora ファイルの
job_queue_processes パラメータをコメント化してください。

コメント化されていない場合、ODMA は、job_queue_processes パラメータが 0（ゼロ）に設定されていないことを示します。この場合は、ODMA を終了せずに、データベースを停止して別のウィンドウで job_queue_processes パラメータをコメント化してから、ODMA に戻って次の手順に進んでください。

アップグレード後は、ユーザーは job_queue_processes パラメータをリストアし、データベースと OID サーバーを再起動する必要があります。

3. 既存のインストールから製品を削除するときに、Make ファイルの起動に関するエラーが表示されることがあります。

このエラーは「Ignore」をクリックして無視し、次の手順に進んでください。

4. インストール時に、インストーラが次のエラー・メッセージを表示します。

```
「Error in creating link from <some location>/jre/1.1.8 to <$ORACLE_H  
OME>/JRE」
```

別のウィンドウを使用して、\$ORACLE_HOME/JRE サブディレクトリを削除してから、「Retry」をクリックして次の手順に進んでください。

5. インストーラは、「ods」ユーザーのパスワードまたは OID 管理者のパスワードの入力時に、その内容を検証できません。無効なパスワードは OID Upgrade Assistant のエラーの原因となるため、パスワードの入力には十分な注意が必要です。

6. アップグレードの際、ODMA の停止アップグレード時に ODMA が停止した場合は、データをリカバリするために手順 1 で作成したデータベースのリストア・スクリプトが必要です。
 - a. インストール・プロセスを手動で中断します。
 - b. ODMA で作成したデータベースのリストア・スクリプトを実行して、データベースをリストアします。
 - c. `$ORACLE_HOME/ldap/postcfg/OidUpgrade` を手動で実行して、アップグレード・プロセスを完了します。
7. データベースの移行時に、ODMA はアドバンスド・レプリケーション・オプションがアップグレードされていないことを示します。次の手順に進んでください。
8. OID Upgrade Assistant の実行中、`$ORACLE_HOME/rdbms/admin/catrep.sql` の実行に長い間、プログレス・バーが「4%」を指した状態になります。
9. `$ORACLE_HOME/network/admin` 内の次のファイルが、アップグレード・プロセスの実行時に ODMA によって変更されます。
 - `listener.ora`
 - `tnsnames.ora`
 - `sqlnet.ora`

オリジナル・ファイルは、同じディレクトリに<ファイル名><バックアップの日時> .ora.bak としてバックアップされます。これらのファイルをリストアし、OID の起動前にリスナーを再起動してください。

10. アップグレードからインストーラを終了した後は、`$ORACLE_HOME/ldap/bin/cryptupgrd.sh` を実行する必要があります。

このスクリプトは、サービス名と「ods」ユーザーのパスワードを入力通りに受け入れます。手順 9 が適切に実行されないと、このスクリプトは停止することがあります。スクリプトを中断し、適切なサービス名を入力して再度実行してください。
11. アップグレード後、スーパー・ユーザー（`cn=orcladmin`）、ゲスト・ユーザーおよびプロキシ・ユーザーのパスワードは、デフォルト値にリセットされます。LDIF ベースのアップグレードでは、「ods」データベース・ユーザーのパスワードのほかに前述のパスワードもそれぞれのデフォルト値にリセットされます。
12. LDIF ベースのアップグレードでは、`backup_oid.sh` は `configset` をエクスポートしません。したがって、ユーザーはアップグレード後にその `configset` を再度作成する必要があります。

13. LDIF ベースのアップグレードでは、`backup_oid.sh` によって「`cn=oracleschemaversion`」サブツリーが `$ORACLE_HOME/ldap/load/orcl_schemaver.ldif` にエクスポートされます。
ただし、`restore_oid.sh` は、このサブツリーをリストアしません。
`$ORACLE_HOME/ldap/load/orcl_schemaver.ldif` へのユーザー定義エントリはすべて、手動で OID サーバーを介して追加する必要があります。
14. レプリケーション環境でのマルチノード・アップグレードの実行中は、すべてのネットワークのアップグレードが完了するまで、新しいパスワード暗号化スキームは使用しないでください。使用すると、パスワード値の一貫性が失われ、認証が無効になります。
15. レプリケーション環境でのアップグレード後、個別のノードごとに、DSE ルートの属性 `orclupgradeinprogress` を `FALSE` に手動で設定する必要があります。この設定は、すべてのノードのリリース 2.1.1 へのアップグレードの完了後にのみ実行する必要があります。

Oracle Internet Directory サーバー リリース 2.1.1

Oracle Internet Directory サーバー リリース 2.1.1 には、Oracle ホームから LDAP サーバーと関連コンポーネントを実行するために必要なすべての実行ファイルが含まれています。

データベースの互換性

Oracle Internet Directory サーバー リリース 2.1.1 は、Oracle8i リリース 8.1.7 のみに対応して動作するものとして認定されています。OID のデータストアとして使用されているデータベースは、OID 専用とする必要があります。このデータベースを有効化し、Oracle Advanced Security のオプションなどの LDAP 対応プログラムを使用して、LDAP サーバー内のユーザー情報を参照しないでください。

LDAP サーバーの新機能と拡張機能

言語コード (RFC 2596) の使用をサポート

Oracle Internet Directory サーバーでは、言語コードを使用して、同じ属性値を RFC 2596 で定義されている異なる言語に格納できます。ユーザーは、適切な言語コードを持つデータの格納および問合せが実行できるようになりました。

- LDAPSEARCH、LDAPADD、LDAPDELETE および LDAPMODIFY の参照に対するサポート

参照サポートが LDAP V3 クライアントで使用できるようになりました。検索ベースの下にあるすべての参照オブジェクトは、`ldapsearch` 要求の一部として戻されます。

- LDAPSEARCH での LDAP V2 クライアントに対する参照サポート

OID は、ドキュメント

<http://www.umich.edu/~dirsvcs/ldap/doc/other/ldap-ref.html> で定義されているように、LDAP V2 クライアントに戻す参照をサポートします。OID は、LDAP_PARTIAL_RESULT (0x09) リターン・コードを使用して、LDAP Message の追加情報フィールドに参照情報が含まれていることを示します。

- OID の参照を管理する manageDSA 制御のサポート

ディレクトリ管理者は、manageDSA 制御を使用して OID の参照を管理できます。manageDSA 制御が ldapsearch 操作に設定されている場合、OID は、通常エントリとして参照を戻します。manageDSA 制御の制御タイプは、2.16.840.1.113730.3.4.2 です。

拡張パスワード暗号化のサポート

OID は、userPassword の値を暗号化するために、複数の暗号化スキームをサポートしています。ディレクトリ管理者は、次のスキームの 1 つをデフォルトの暗号化スキームとして選択できます。サポートされる暗号化スキームは、次のとおりです。

非暗号化、MD4、MD5、SHA-1 または UNIX Crypt

注意：ディレクトリ管理者は、デフォルトの暗号化スキームを随時変更できます。ただし、結果として、既存のエントリが再度ハッシュされることはありません。

複雑な問合せに対するパフォーマンスの強化

次のフィルタ・タイプでは、このリリースでのパフォーマンスの改善が見られます。

- NOT フィルタ付きと NOT フィルタなしのすべての単純 AND フィルタ・タイプ。次にその例を示します。

```
"(&(postalcode=wx*)(cn=hm))",
"(&(postalcode=*)(!(sn=johnson))"
```

- NOT フィルタなしで、同じ属性タイプにのみ適用される OR 演算子を持つ単純 OR フィルタ。次にその例を示します。

```
"(|(objectclass=person)(objectclass=inetorgperson))",
"(|(postalcode=*wv*)(postalcode=B1 9XX))"
```

パフォーマンスの差が見られない単純 OR フィルタは、次のとおりです。

```
"(|(objectclass=person)(postalcode=*))", => 異なる属性タイプ
"(|(!(postalcode=B1 9XX))(postalcode>=1000))", => NOT フィルタが存在
```

- 単純 OR フィルタおよび単純 OR フィルタに対する制限を含んだ複合 AND フィルタ。
次にその例を示します。

```
"(&(objectclass=inetorgperson)(cn=hm)(|(postalcode=*vv*)(postalcode=*XX))
(! (sn=johnson)))"
```

LDAPSEARCH でのデフォルト参照に対するサポート

管理者は、参照ネットワークにある別の既知の OID ディレクトリ・サーバーのホスト名とポートを含んだ ref 属性を作成して、OID にデフォルト参照を構成できます。この ref 属性は、DSE（ルート）エントリで作成する必要があります。

ランタイムのデバッグ・レベル切替えに対するサポート

ディレクトリ管理者は、異なるデバッグ・レベルでルート DSE のエントリに orcldebugflag パラメータを設定して、Oracle Internet Directory サーバーのデバッグ・レベルを変更できるようになりました。この変更を有効化するために、サーバーを再起動する必要はありません。Oracle Internet Directory の configset にあった古い orcldebuglevel パラメータは削除されました。

オブジェクトの追加時における objectClass 階層の解決に対するサポート

新規エントリの追加時に、Oracle Internet Directory が objectClass 階層を解決するようになりました。ユーザーは、新規エントリに関する最も固有な objectClass のみを指定し、他の必要な objectClass すべてを Oracle Internet Directory に透過的に含めることができます。たとえば、エントリを定義するとき、organizationalPerson クラス内の新規エントリでは、「top」および「person」の objectClass（organizationalPerson の祖先クラス）を指定する必要はありません。

レプリケーションの新機能と拡張機能

管理者操作キューに対するレプリケーション・サイクルが構成可能となりました。

コンシューマ・ディレクトリへの適用が数回にわたって失敗すると、その変更内容は、管理者操作キューに移動します。これらの変更は、レプリケーション承諾の orclHIQSchedule パラメータに定義されている構成可能な間隔で再度試行されます。orclHIQSchedule のデフォルト値は 10 分です。

管理者操作キューの操作ツールが使用可能となりました。

管理者操作キューにある変更内容は、不要になるとページ・キューに移動します。高い頻度で再試行する必要がある場合は、再試行キューに移動します。

レプリケーション調整ツールが使用可能になりました。

Oracle Internet Directory では、異なるディレクトリ間で一貫性のないデータを修正できるようになりました。OID の新しい調整ツールを使用すると、コンシューマ・ノード上のエントリをサプライヤ・ノード上のエントリと同期化できます。このツールを使用すると、サブツリー全体の他に、リーフ・エントリも修正できます。

ノードの削除プロシージャが使用可能になりました。

Oracle Internet Directory 管理者は、DRG からノードを削除できるようになりました。管理者がノードを削除するのは、次の場合です。

- LDIF データのロード時における表領域の不足などの予想外のシステム・エラーによって、新しいノードを追加できなかった場合。この場合、管理者はその新しいノードを DRG から削除する必要があります。
- DRG 要件が変更された場合。管理者はノードを DRG から削除し、別の目的に使用できます。

他のディレクトリに対する変更サブスクリプションのサポート

他のディレクトリは、それ自体を変更サブスクリプション・クライアントとして、OID に登録できるようになりました。これによって、クライアント・ディレクトリは、アクセス権限を与えられ、OID に格納されているログ・オブジェクトを変更し、それを OID と同期化できます。

LDAP サーバーの制限事項と問題

Oracle Internet Directory サーバー リリース 2.1.1 の複数インスタンスは同一マシン上でのみ実行可能です。

Oracle Internet Directory サーバー リリース 2.1.1 のインストール後は、LDAP サーバーの複数のインスタンスを同じマシン上で実行できます。たとえば、1 つのサーバーを SSL モードで実行し、別のサーバーを SSL 以外のモードで実行することができます。ただし、特定のデータベース・サーバーを使用している OID サーバーのインスタンスすべてを、同じマシンで実行する必要があります。たとえば、マシン C にあるデータベースに対して、2 つの OID サーバー（マシン A とマシン B）を実行することは、サポートされていません。ただし、マシン B にあるデータベースに対して、マシン A で両方の OID サーバーを実行することは、サポートされています。

LDAP サーバーへのレプリケーション・サーバー接続に対する SSL サポートはありません。

このリリースの Oracle Internet Directory では、oidrepld サーバー・プロセスは、SSL を使用して SSL ベースの oidldapd プロセスに接続できません。

SSL V2 クライアントはサーバーに接続できません。

SSL V2 を使用している LDAP クライアントは、Oracle Internet Directory リリース 2.1.1 サーバーにバインドしようとする、「Can't Contact LDAP server」というエラー・メッセージが表示される場合があります。

Oracle Internet Directory データベースのユーザー / パスワードを変更する必要があります。

Oracle Internet Directory をインストールした後で、管理者は、Oracle Internet Directory サーバー・プロセスで使用されるパスワードをリセットし、OID データベース・パスワード・ユーティリティ (oidpasswd) を実行して、基礎となる Oracle8i データベース表に接続およびアクセスすることができます。oidpasswd の使用手順は、『Oracle Internet Directory 管理者ガイド』を参照してください。初期パスワードは「ODS」です。

デフォルト ACP では、スーパー・ユーザーを除く全ユーザーに対する書込みアクセスが制限されます。

Oracle Internet Directory リリース 2.1.1 に同梱されているデフォルトのアクセス制御ポリシーは、スーパー・ユーザー (cn=orcladmin) を除く全ユーザーに対してディレクトリでの書込みアクセスを制限しています。サイト管理者は、実際のセキュリティ考慮事項に基づいてこのポリシーを緩和することができます。デフォルトの ACL ポリシーは、インストールが正常に完了した後で、oidbaseacl.ldif と呼ばれる LDIF ファイルを使用してロードされます。詳細は、「Oracle 固有のスキーマのディレクトリへのインストール」の項を参照してください。

索引付き属性名は 28 文字以内です。

属性名が 28 文字を超えていると、catalog.sh を使用して属性に索引を作成できません。

索引付けできるのは等価の一致規則を持つ属性のみです。

索引が作成できるのは、等価の一致規則が属性定義に指定されている属性のみです。等価の一致規則が指定されていない属性の場合は、属性に索引付けを実行する前に等価の一致規則を割り当てる必要があります。

catalog.sh ユーティリティの使用とサポートされる一致規則の詳細は、『Oracle Internet Directory 管理者ガイド』を参照してください。

索引付き属性の等価の整数一致は、文字列の一致と同じように動作します。

等価に対する integerMatch を持つ属性に、catalog.sh を使用して索引付けを行うと、その属性の一致規則は、整数の一致ではなく文字列の一致のように動作します。

別名の参照解除は LDAP 操作ではサポートされません。

Oracle Internet Directory リリース 2.1.1 は、LDAP 操作における別名の参照解除をサポートしていません。

構文チェックは LDAP サーバーではサポートされません。

LDAP サーバーはエントリの追加および変更時にユーザーが入力した属性値の構文を検証しません。

高い選択性を持つシングル AVA フィルタでは速度が低下します。

サブツリーおよび 1 レベルの検索にシングル AVA フィルタを使用すると、そのフィルタが固有でない場合、つまり、対象のカatalogの選択性が高い場合は速度が低下します。この検索では、DN がきわめて固有であっても、速度が低下します。

必須属性リストの属性オプションの使用はサポートされません。

現行のリリースの Oracle Internet Directory は、属性オプションの使用をサポートしていません。したがって、必須属性リストにある言語コードの検索操作での使用もサポートされません。

Oracle Internet Directory は、すべての言語コードまたは必須属性で指定された属性オプションを無視します。かわりに、必須属性リストに記載されている属性のすべての値（属性オプションの有無にかかわらず）を戻します。

Oracle Internet Directory サーバー接続中に発生する SSL の断続的なハンドシェイク・エラー (Bug#1398892)

LDAP クライアントでは、SSL モードで実行中の LDAP サーバーへの接続（バインド）を試行中に、断続的な接続エラーが発生する場合があります。接続を何回か試行して回避してください。

レプリケーションの制限事項

新しいディレクトリ・レプリケーション・グループ (DRG) の作成

『Oracle Internet Directory 管理者ガイド』の新しいディレクトリ・レプリケーション・グループ (DRG) の作成に関する項では、DRG に使用するすべてのノードに既存ディレクトリのデータが存在しないことを前提としています。

既存のディレクトリ・レプリケーション・グループに対する新しいノードの追加

既存のディレクトリ・レプリケーション・グループに新しいノードを追加する場合は、その新しいノードに以前のディレクトリ・データが存在しないようにする必要があります。以前のデータは、DRG 内の別の参加者にレプリケートされません。以前のデータをレプリケートする場合は、最初に `ldapsearch -L` オプションを使用してそのデータを LDIF ファイルに抽出し、新しいノードが DRG に追加され、新しいデータが他のノードにレプリケートできるようになった後で、`ldapadd` ユーティリティを使用して再ロードする必要があります。

ACL ポリシー情報以外のローカル・システム固有のメタデータは、レプリケートされません。

DSE ルート固有のデータ、サーバー構成データおよびレプリケーション承諾データは、ディレクトリ・レプリケーション・グループのサーバー間でレプリケートされるデータに含まれていません。このルールの唯一の例外は、DSE ルート固有の ACL ポリシーの属性である `orclaci` と `orclentrylevelaci` はレプリケートされることです。

レプリケーション・サーバーは、RDN コンポーネント間で空白を保持しません。

レプリケーション・サーバーは、エントリ・レプリケーション時に DN の RDN コンポーネント間で空白を必ずしも保持しません。DN の文字の大/小文字を保持しない場合もまれにあります。

すでにアクティブなレプリケーション承諾の一部であるノードに、`bulkload.sh` を使用してデータを追加できません。

LDAP サーバー・インスタンスが、レプリケーション承諾に関連している場合は、`bulkload.sh` を使用してノードにデータを追加しないでください。データのロードには、`ldapadd` ツールを使用してください。

ログ・ファイルのロケーション

Oracle Internet Directory のコンポーネントは、それぞれのログ情報とトレース情報を、ORACLE_HOME 環境内でメンテナンスされているログ・ファイルに出力します。コンポーネントとそれぞれのログ・ファイルのロケーションのリストは、次のとおりです。

コンポーネント	ログ・ファイル名
LDAP ディスパッチャプロセス oidldapd	\$ORACLE_HOME/ldap/log/oidldapdXX.log XX = サーバー・インスタンス#
LDAP サーバープロセス oidldapd	\$ORACLE_HOME/ldap/log/oidldapdXXs<pid>.log <pid> = サーバー・プロセス ID
レプリケーション・サーバー・プロセス oidrepld	\$ORACLE_HOME/ldap/log/oidrepldXX.log XX = レプリケーション・サーバー・インスタンス#
監視プロセス oidmon	\$ORACLE_HOME/ldap/log/oidmon.log
バルク・ローダー bulkload.sh	\$ORACLE_HOME/ldap/log/install.log
カタログ・マネージャ catalog.sh	\$ORACLE_HOME/ldap/log/catalog.log
レプリケーション設定 ldaprepl.sh	\$ORACLE_HOME/ldap/admin/logs/ldaprepl.log

Oracle Internet Directory クライアント リリース 2.1.1

OID クライアント 2.1.1 には、次のソフトウェアが含まれています。

- 様々な LDAP クライアントが LDAP サーバーと対話するために必要な LDAP ライブラリ。
- ldapsearch、ldapadd などの各種汎用 LDAP コマンドライン・ツール。
- OID LDAP サーバーを管理するための管理ツールである Oracle Directory Manager。oidadmin または ODM としても知られています。

Oracle Internet Directory コマンドライン・ツールの拡張機能

パラレル・バルク・ロードが Solaris でサポート

bulkload.sh ツールは、-parallel オプションをサポートします。このオプションを指定すると、パラレルでデータをロードできます（Solaris のみ）。

バルク・ロード・ツールによるバイナリ・データ・ロードのサポート

bulkload.sh ツールが、バイナリ・データ・ロードをサポートするようになりました。これにより、既存ノードにバイナリ・データが含まれている場合は、ldifwrite ベースのノード追加プロシージャを使用して新規ノードを追加することができます。

Oracle Internet Directory コマンドライン・ツールの制限事項

**不適切な LDIF データは、LDAP ツールにおけるセグメンテーション障害の原因になります。
(Bug#1103958)**

LDIF 形式を使用してデータをロードするときに、オブジェクトの識別名を挿入しない場合は、ldapadd ツールまたは ldapmodify ツールがクラッシュする可能性があります。このクラッシュからリカバリするには、正常にロードを完了したすべてのエントリを削除し、LDIF ファイルを修正してから、コマンドを再実行します。

LDAP 検索の制限事項

エントリのファジー・マッチはサポートされません。

ldapsearch は、LDIF 出力をデフォルトでは生成しません。

LDIF 形式の出力を `ldapsearch` コマンドライン・ツールで生成するには、`-L` フラグを使用します。

カタログ・マネージャの使用上の問題

カタログ索引管理ツール (`catalog.sh`) では、索引付けによって、以前は検索不可能だった属性を検索可能な属性に変換できます。また、新しい属性に対する索引の定義と削除を実行できます。OID とともにインストールされたベース・スキーマによって作成された索引ではないことに確信がない場合は、属性の索引の削除に `catalog.sh` の `-delete` オプションを使用しないように注意してください。ベース・スキーマの属性から索引を削除すると、OID の操作に悪影響を与えます。サーバー側の索引付き属性については、前述の「索引付き属性名は 28 文字以内です。」から「索引付き属性の等価の整数一致は、文字列の一致と同じように動作します。」の項で説明した制限事項を参照してください。新規にカタログ化された属性（複数可）を実行中の OID プロセスに認識させるために、その OID プロセスを再起動する必要があります。

-U オプションが指定されていないと、SSL モードにおける OID へのアクセス時にコマンドライン・ツールが停止します。

LDAP コマンドライン・ツールを使用し、SSL モードで LDAP サーバーに接続するときは、`-U` オプションを指定する必要があります。指定しないと、コマンドライン・ツールは停止します。

フィルタにカタログ化されていない属性が含まれていると、bulkmodify は誤解を招くようなエラーを戻します。

非索引付け属性（検索不能な属性など）が含まれている検索フィルタを使用して `bulkmodify` ユーティリティを実行しようとする、と、「Subtree Search failed」というエラー・メッセージがユーザーに戻されます。状況を調整するには、該当の属性で `catalog.sh` を実行して索引付けするか、または索引付け属性のみを使用するようにフィルタ基準を変更します。

-r オプション付き LDAPADD はサポートされません。

`-r` オプション付きの `ldapadd` は、同じ `dn` を持ったエントリがすでにディレクトリに存在する場合に、そのエントリを置換します。ディレクトリの情報ツリーに同じ識別名がすでに存在していると、「object already exists」というメッセージが表示されます。この場合は、更新済みのバージョンを作成する前に既存のエントリを手動で削除するか、かわりに `ldapmodify` を使用します。

Oracle Directory Manager (OID 管理ユーティリティ、oidadmin)

Oracle Directory Manager では、OID におけるサーバー、データおよびポリシーを管理するための、使用が容易なグラフィカル・ユーザー・インタフェースを提供しています。このインタフェースはコマンドライン起動 (oidadmin) を使用して起動できます (Windows の場合には[スタート]メニューから起動できます)。

このドキュメントでは、Oracle Directory Manager、ODM および oidadmin という用語は、同義で使用されています。

リリース 2.0.6 以降の ODM における不具合の修正と拡張機能

- ODM エントリ管理は、操作属性の表示をサポートします。

操作属性には、作成者名、作成時刻スタンプ、変更者名および変更時刻スタンプがあります。

- ODM は、カスタマイズしたアクセス制御ポリシー・ポイント (ACP) の表示をサポートします。 (Bug#988852)

5,000 を超える ACP が存在する場合は、ODM での潜在的な停止を回避するために、この新機能を使用することをお薦めします。『Oracle Internet Directory 管理者ガイド』の第 9 章「ディレクトリのアクセス制御の管理」を参照してください。

- オブジェクト・クラスの編集フォームの「Make Optional」ボタンが正常に動作しません。 (Bug#918761)

「Schema Management」の下にあるオブジェクト・クラスを変更する場合、必須属性値を選択して「Make Optional」ボタンをクリックすると、選択された属性が必須リストから削除されますが、その選択された属性の下にある属性はオプション・リストに表示されます。つまり、同じ属性が必須とオプションの両方のリスト・ボックスに表示されるため、変更内容を正常に保存することができません。

- ODM が構成セットの変更の場合にキャッシュを適切に処理しません。 (Bug#918691)

「Server Management」->「Directory Server」->「Default Conf Set」を選択して「SSL Setting」タブを選択し、変更したとします。この状態で、別のウィンドウをクリックすると、ODM は確認のためのダイアログ・ボックスを表示します。変更を廃棄するために「NO」をクリックすると、ダイアログ・ボックスが消失します。以前に編集した構成セットに戻って別のウィンドウをクリックしても、「confirmation」ダイアログ・ボックスが表示されます。

- 重複したエントリが、グループ・エントリの作成時に **member** 属性に表示されます。
(Bug#914256)
「Entry Management」を選択して **member** 属性または **uniquemember** 属性を持つエントリを選択します。この状態で、右側のパネル属性の隣の「Browse」ボタンをでクリックすると、「search engine」ダイアログ・ボックスが表示されます。検索を実行し、「result」リスト・ボックス内で1つのエントリを選択して「OK」をクリックすると、エントリが重複します。1つは検索エンジンから選択されたエントリ、もう1つは「attribute」フィールドに表示されているエントリです。
- 2つのオーバーラップしたメニューが、一部の管理コンテナの右クリック時に表示されます。
(Bug#908781)
「Subtree Access Management」、「Entry Management」、「Server Management」、「Schema Management」または「Audit log Management」が選択されていないと、そのいずれかを右クリックすると、2つのメニューがオーバーラップして表示されます。
- ODM がエラー・メッセージ「Function Not Implemented」をレポートします。
(Bug#903472)
LDAP サーバーの停止時に、oidadmin は LDAP の停止をレポートせず、「Function Not Implemented」というあいまいなエラー・メッセージをレポートします。
- 検索エンジンでの検索基準操作の変更はできません。
(Bug#903463)
「Entry Management」を選択した場合、検索エンジンは右側のパネルに表示されます。ある期間にわたって検索基準を構成した後、ユーザーによる検索基準の変更ができなくなります。この問題を回避するためには、検索基準全体を削除してから、新規の検索基準を作成してください。
- パスワードなしのユーザー・ログインは無名ログインとして処理されます。
(Bug#902957)
- ODM が更新要求のタイムアウト時に警告メッセージを表示しません。
(Bug#797892)
- ODM が操作の正常完了のダイアログをポップアップ表示しません。
(Bug#760412)
操作がエラーなしで完了すると、「Executed Successfully」というメッセージがステータス・バーに表示されます。
- ODM がセッション全体のユーザー設定を保存しません (Bug#760406)。
ODM は、ACP 表示のカスタマイズに関する構成情報および最新のサーバー/ポートの組合せを\$HOME/osdadmin.ini ファイルに保存します。

oidadmin に関する現在の制限事項

- oidadmin リリース 2.1.1 による OID の旧リリースの管理
リリース 2.1.1 に同梱されている ODM のリリースは、OID サーバーの次のリリースとのみ動作します。
 - 2.0.4.x
 - 2.0.5.x
 - 2.0.6.x
 - 2.1.1
- oidadmin を使用したサード・パーティ・ディレクトリの管理
oidadmin を使用する OID 以外の LDAP ディレクトリの管理は、サポートされません。
- マシンのメモリ不足または低速なネットワークによって発生する
イメージ・フェッチの例外
この例外は、既知の JDK (Bug#4112007, JavaSoft) の不具合です。回避策の 1 つは、oidadmin の実行前に ulimit をコールし、ファイル記述子の数を 1024 に増やすことです。
- oidadmin を使用したディレクトリへのログイン時に、メッセージ「Bind Failed」が表示されます。
このエラー・メッセージは、無効なユーザー名とパスワードが原因です。SSL の使用時にこのエラー・メッセージが表示された場合は、SSL のロケーション、パスワードまたは認証レベルが無効であることを示している可能性があります。
- 空の属性が表示されません (Bug#1370786)。
ODM では、エントリの objectClass 属性の最後の値に「top」が定義されていない場合、「Entry Management」の下にある「show all」属性タブにすべての空の属性が表示されません。
回避策: そのエントリの objectClass 属性に最後の値として top を定義します。
- ODM 構造型アクセス項目で「Help」ボタンが動作しません (Bug#1369808)。
ディレクトリ・サーバーに接続した後、「Access Control」をクリックし、「Default ACP」をクリックします。「Create via Wizard」ボタンをクリックすると、「Structural Access Item」タブが表示されます。このウィンドウにある「Help」が正常に機能しません。また、ラベル「Create via Wizard」が ODM の他の言語バージョンに変換されていません。

- ODM は、他のどのオブジェクト・クラスからも継承されない objectClass を持つエントリの作成に失敗します (Bug#1360090)。

最初に、親なしの objectclasses を作成します。次に、この objectclasses を含んだエントリを作成します。すると、エントリの作成の失敗を示すメッセージが表示されます。
回避策: 他のどのオブジェクト・クラスからも継承されない objectClass の場合は、構造的なオブジェクト・クラスの作成時に、親として top を必ず含めます。

- エラー・コード=112 がドキュメント化されていません (Bug#1271471)。
- CN=AUDITLOG の場合、ODM では ACP が表示できません (Bug#1261398)。

Oracle Internet Directory の PL/SQL API (DBMS_LDAP)

Oracle8i リリース 8.1.7 には、PL/SQL プログラミング環境から LDAP サーバーと対話する機能があります。この機能は、DBMS_LDAP と呼ばれる新しい PL/SQL パッケージを使用して実現されました。このパッケージを使用するために、Oracle Internet Directory サーバーをインストールする必要はありません。Oracle8i リリース 8.1.7 で DBMS_LDAP パッケージの使用を有効化するには、\$ORACLE_HOME/rdbms/admin ディレクトリにある catldap.sql と呼ばれる SQL スクリプトを実行します。DBMS_LDAP パッケージの使用の詳細は、『Oracle Internet Directory アプリケーション管理者ガイド』を参照してください。

Windows 95、98、2000 および NT 固有の問題

データのロードに BULKMODIFY と BULKLOAD は使用できません。

これらのツールを使用して、Windows 95 および 98 のマシンから Windows NT または Windows 2000 上で実行中の OID サーバーにデータをロードすることはできません。かわりに、ldapaddmt または ldapmodifymt を使用してバルク操作を実行します。

パスの設定は、サーバーを再起動するまで有効化されません。

インストール後にマシンを再起動しない限り、パスの設定は有効化されません。よって、マシンを再起動しないと Oracle Directory Manager を開始できません。

Oracle Directory Manager を Windows 98 にインストール後、最初に ODM の起動時に「Bind Failed」が発生します。

回避策: Oracle Directory Manager をクローズし、再起動します。

同時接続に関する Windows NT ワークステーションの制限事項

Windows NT ワークステーション上に OID をインストールしている場合、同時インバウンド接続が最大 6 に制限されていることを認識しておいてください。これ以上の負荷で OID を使用する場合は、Windows NT サーバー上に OID をインストールし、追加のユーザー・ライセンスを購入する必要があります。同時使用に関する詳細は、Microsoft 社に連絡してください。

バルク・ツールの制限事項

すべてのバルク・ツールは UNIX のシェル・スクリプトで実装され、NT および Windows 2000 上の UNIX シェル・エミュレータのもとでのみ使用できます。OID 2.1.1 シェル・スクリプトに対して認定されているツールキットは、次のとおりです。

MKS Toolkit V5.1 以上 - <http://webstore.mks.com/webstore/>
Red Hat Cygwin -- <http://sources.redhat.com/cygwin/>

第 8 章 Oracle Enterprise Manager リリース 2.2

原典情報: A85246-01 Oracle Enterprise Manager Readme Release 2.2

この章には、出荷された Oracle Enterprise Manager リリース 2.2 製品と、ドキュメントに記載されている機能との相違点が記載されています。Enterprise Manager に関する最新情報が含まれてますので、製品を使用する前に、この章を必ずお読みください。さらに、統合アプリケーション（Oracle Developer Server Forms Manager など）を使用する前に、各アプリケーションの Readme ファイルをお読みください。すでにインストールされている場合は、`$ORACLE_HOME/relnotes/EM` ディレクトリにあります。

- Enterprise Manager リリース 2.2 の新機能
- サポートされるオペレーティング・システム
- 互換性情報
- インストール
- コンソール
- イベント・システムおよびジョブ・システム
- グループ
- ポケットベルおよび電子メールのブラックアウト
- DBA Management Pack
- Oracle Management Server
- Enterprise Manager Configuration Assistant
- Enterprise Manager Migration Assistant
- ブラウザ・ベースの Enterprise Manager
- その他

Enterprise Manager リリース 2.2 の新機能

この項では、Enterprise Manager リリース 2.2 での新機能について説明します。

Enterprise Manager コンソール

- コンソールのフォント・サイズ設定の変更が可能になりました。
- コンソール内の全ノードのリフレッシュ方法が強化されました。
- ノードおよびエージェントの詳細をコンソールのナビゲータから表示できます。この情報を表示するには、ナビゲータ・ツリーでノードを右クリックし、表示されたメニューから「プロパティ」を選択します。ノードで Agent リリース 8.1.7 が稼働している場合は、オペレーティング・システム情報も表示されます。

パフォーマンスの向上

コンソールおよび DBA Studio のパフォーマンスを向上させるため、大幅な改良が加えられました。次の操作を行う際のパフォーマンスが向上しています。

- コンソールおよび DBA Studio の起動
- コンソールのナビゲータ・ツリーの拡張
- ナビゲータ・ツリーで選択したオブジェクトの詳細表示

ブラウザベースのコンソールの場合、Web サーバーに初めて接続する際には、jar ファイルをダウンロードする間、進捗ダイアログが表示されます。

Apache Web サーバーおよび Apache for iAS Web サーバーのサポート

検出、ステータス・イベント、ジョブおよび通知などの新機能により、一般的な Apache および Oracle Internet Application Server 8i (iAS) Apache Web サーバーの中間層環境を管理できます。パフォーマンス監視、拡張イベントおよびレポートなどの拡張機能が Diagnostics Pack に組み込まれています。

基本的な管理およびイベントの詳細は、『Oracle Enterprise Manager 管理者ガイド リリース 2.2』の第 2 章「ナビゲータ」を参照してください。

拡張イベントおよびパフォーマンス監視の詳細は、『Oracle Enterprise Manager Oracle Diagnostics Pack スタート・ガイド』の第 5 章「Oracle E-Business Management Tools」と第 7 章「Oracle Advanced Event Tests」を参照してください。

ジョブ・システムおよびイベント・システム

- イベント・システムに追加された新機能: 新しいオペレーティング・システム固有のノード・イベント、拡張されたノードの動作・停止イベント、Agent リリース 8.1.7 による重複イベントのサポート、イベント・テストの状況依存ヘルプ、ターゲット・サブコンポーネントの拡張された監視。『Oracle Enterprise Manager 管理者ガイド リリース 2.2』の第 5 章「イベント」を参照してください。
- ジョブ・システムの新機能: ジョブ・タスクの状況依存ヘルプ、電子メール通知にジョブ出力を含めるオプション。

拡張通知

- ポケットベル・サーバーが、Intelligent Agent を必要としない独立したサービスになりました。コンソール内からポケットベル・サーバーを構成することもできます。
- 拡張通知（ポケットベルおよび電子メール）のコンテキストとフォーマットをコンソール内からカスタマイズできるようになりました。イベントおよびジョブ・ステータス通知に応じて、電子メールおよびポケットベルに異なるフィルタを指定することもできます。

詳細は、『Oracle Enterprise Manager 管理者ガイド リリース 2.2』の第 1 章「コンソール」を参照してください。

エンタープライズ・レポート

Management Server で管理している全ターゲットの詳細を含む HTML レポートを生成できるようになりました。レポートに含まれる情報は、イベント警告、スケジュール済ジョブ、グループおよび検出済サービスです。ウィザードを使用して、これらのレポートが定期的に生成されるように設定できます。詳細は、『Oracle Enterprise Manager 管理者ガイド リリース 2.2』の第 1 章「コンソール」を参照してください。

また、オブジェクト定義、オブジェクト従属情報およびデータベース構成の HTML データベース・レポートや、カスタム問合せなどのその他のレポートも生成できます。詳細は、Enterprise Manager のオンライン・ヘルプを参照してください。

DBA Studio

- リソース管理の強化: リソース消費グループのリソース計画を作成するウィザード、リソース消費アクティビティのグラフィカル表示、リソース計画スケジューリングのサポート。
- データベースへの接続時にアプリケーションが行った DDL および DML の全変更のロギングをサポートします。
- Replication Manager の強化: DBA Studio へのレプリケーション管理の統合、ツリーおよびプロパティ・ページの直観的な新しいレイアウト。
- JServer オブジェクトの参照、クラスの実行、リンクおよびディレクトリの作成、およびセキュリティ・オプションの指定が可能です。
- データベース・オブジェクトの DDL の表示。
- 分析ウィザードは、Oracle8i の新しい統計生成方法 (DBMS_STATS) をサポートしており、データベースに対して分析をただちに実行できます。
- マテリアライズド・ビューの強化: ジョブ・システムを使用したマテリアライズド・ビューの作成/移入、作成後に分析を行うオプション。
- バックアップ・ウィザードでのイメージ・コピーをサポートします。
- 従属オブジェクトをリストとツリーのいずれかで表示し、表示からオブジェクトを直接編集することができます。
- 表に列権限を付与できます。
- データ管理ウィザードでのジョブのエラーおよび警告に対する失敗と成功のインジケータをサポートします。
- レンジ・パーティション化の新しいグラフィカル・インタフェースをサポートします。
- フォント設定を変更できます。

OMS チューニング用パラメータ

特殊な状況の下では、特定の OMS チューニングが必要な場合があります。チューニングを行うには、一部の OMS 構成パラメータを設定します。

設定するパラメータは、次のとおりです。

- oms.vdp.ping_interval
- oms.vdg.max_out_conns
- oms.vdg.max_in_conns
- oms.repository.connect_timeout
- oms.repository.connect_numTries

詳細は、『Oracle Enterprise Manager 構成ガイド』を参照してください。

Intelligent Agent リリース 8.1.7

- 複数のネットワーク・インターフェース・カード (NIC) の明確なサポート: この機能により、Agent はマシン上の主 NIC で受信ジョブおよびイベント要求をリスニングできます (デフォルト)。その他のオプションを使用すると、Agent によるリスニングをユーザー指定の NIC と全 NIC のどちらで行うかを構成できます。
- 重複イベントのサポート: この機能により、同じパラメータを持つ重複イベント・テストを同じターゲット (データベース、リスナーなど) に対して登録できます。
- データ収集用移行ユーティリティ: nmumigrate という新しいコマンド行ユーティリティにより、ユーザーは、既存 Agent の Data Gatherer で実行されている既存のデータベース収集を、Agent リリース 8.1.7 の Data Gatherer で認識される新しいフォーマットに移行できます。
- Agent リリース 8.1.5 および 8.1.6 から Agent リリース 8.1.7 へのアップグレード時に、既存の Enterprise Manager のジョブまたはイベントを移行: この方法により、ユーザーは既存のジョブまたはイベントを手動で素早く移行することができます。ジョブまたはイベント・ライブラリに保存してから、登録解除して再発行する必要はありません。
- NT での自動スタートアップ: 特定の状況では、Agent および Data Gatherer サービスがインストール後に自動的に起動されます。

注意: これらの機能および強化の詳細は、Oracle Enterprise Manager ドキュメント・セットに含まれる『Oracle Intelligent Agent ユーザーズ・ガイド』、および \$ORACLE_HOME/network/doc ディレクトリにある Oracle Intelligent Agent リリース 8.1.7 の Readme ファイルを参照してください。

サポートされるオペレーティング・システム

この項では、Enterprise Manager リリース 2.2 の各種コンポーネントでサポートされるオペレーティング・システムを示します。

Enterprise Manager 2.2 コンポーネント	Windows NT/2000	Windows 95/98	Sun Solaris 2.6/2.7	HP-UX *	IBM AIX *	Compaq Tru64 *	Linux *
EM コンソール	○	○	○	○	○	○	○
Oracle Management Server	○		○	○	○	○	○
DBA Management Pack	○	○	○	○	○	○	○

注意: これらのオペレーティング・システム・プラットフォーム用の Enterprise Manager を入手するには、各オペレーティング・システムに対応する CD パックを注文してください。

互換性情報

次の表は、Enterprise Manager の各バージョンをサポートする Intelligent Agent のバージョンを示します。また、この表には Enterprise Manager の各バージョンで管理可能な Oracle データベースのバージョンも記載されています。特記事項は注記されています。

Enterprise Manager の リリース	サポートする Intelligent Agent のリリース	管理可能なデータベースの リリース
2.2	8.1.7	8.1.7
	8.1.6.x	8.1.6.x
	8.1.5.x	8.1.5.x
	8.0.6.x	8.0.6.x
	7.3.4.x	7.3.4.x
2.1.x	8.1.6.x	8.1.6.x
	8.1.5.x	8.1.5.x
	8.0.6.x	8.0.x
	7.3.4.x	7.3.4.x
2.0.x	8.1.5.x	8.1.5.x
	8.0.6.x	8.0.x
	7.3.4.x	7.3.4.x

注意:

- Intelligent Agent リリースで監視または管理できるのは、同一リリースまたは旧リリースのデータベースです。たとえば、Intelligent Agent リリース 8.1.7 で監視または管理できるのは、リリース 8.1.7 以前のデータベースです。新しいリリースの Intelligent Agent を使用して以前のリリースのデータベースを管理する場合、Intelligent Agent は、それ自身の Oracle ホーム、または Intelligent Agent と同じバージョンの Oracle ホームにインストールする必要があります。
- Intelligent Agent リリース 8.0.5 以降には、データ収集サービス (Data Gatherer) が付属しています。
- Enterprise Manager リポジトリの作成は、8.1.7、8.1.6.x、8.1.5.x、8.0.6.x および 7.3.4.x のデータベース・リリースで保証されています。
- リリース 2.2 のバックアップ管理ウィザードは、リリース 8.0.6 以降のデータベースをサポートします。
- JServer 管理は、リリース 8.1.7 データベースに接続している場合のみサポートされます。

Enterprise Manager 製品と Management Server リポジトリ

Enterprise Manager コンソール、DBA Management Pack、個別にライセンス契約が可能な Management Pack、Management Server およびリポジトリは、すべて同じリリースである必要があります。たとえば、リリース 2.1 の Management Server とリポジトリをリリース 2.2 のコンソールとともに使用することはできません。また、リリース 2.2 の Management Server とリポジトリをリリース 2.1 の Management Pack とともに使用することはできません。Enterprise Manager のリリース 2.2 製品では、リリース 2.2 の Management Server とリポジトリを使用する必要があります。Enterprise Manager のリリース 2.2 製品とは、以下のいずれかを指します。

- Oracle Enterprise Manager コンソール
- Oracle DBA Management Pack
- Oracle Tuning Pack
- Oracle Diagnostics Pack
- Oracle Change Management Pack
- Oracle Management Pack for Oracle Applications
- Oracle Management Pack for SAP R/3
- Oracle Standard Management Pack

既存の Management Server とリポジトリのバージョンが古い場合は、最新のバージョンに移行またはアップグレードできます。Management Server とリポジトリをリリース 2.2 に移行またはアップグレードする場合、リポジトリとともに使用する Enterprise Manager 製品がすべてリリース 2.2 であることを確認してください。たとえば、Diagnostics Pack リリース 2.1 や Change Management Pack リリース 2.1 をまだ使用しているのであれば、Management Server とリポジトリはリリース 2.2 にアップグレードしないでください。使用するリリース 2.2 の Management Pack ソフトウェアを入手する前に、Management Server とリポジトリをリリース 2.2 にアップグレードすると、古い Management Pack 製品は使用できなくなります。

移行とアップグレードの詳細は、『Oracle Enterprise Manager 構成ガイド』を参照してください。

Compaq Tru64 では、リポジトリの作成中、「進行中の作業」ダイアログ内の「詳細の表示」および「詳細の非表示」ボタンをクリックすると、リポジトリ作成の作業が停止してしまう場合があります。この場合、リポジトリ作成のプロセスをもう一度実行してください。

(Bug#1403809)

インストール

Enterprise Manager Web サイトをインストールする場合、「サマリー」ページに一覧表示される領域要件に加えて、一時ディレクトリに 150～200MB 程度の空き領域があることを確認してください。この領域は、インストール中に Installer がファイルの中間処理を行うために使われます。Windows NT プラットフォームでは、一時ディレクトリは TEMP システム環境変数で指定します。Solaris では、/tmp にあります。(Bug#1357839)

コンソール

- コンソールの起動後に別のウィンドウを選択した場合、ログイン・ダイアログが手前に表示されません。アクティブなウィンドウをすべて最小化するか、[Alt]+[Tab]キーを使用してコンソール・アプリケーションを選択し、ログイン・ダイアログを手前に表示する必要があります。(Bug#806713)
- コンソールからの接続用に Management Server 名を設定するときには、ノードの完全修飾名 (foo.us.oracle.com など) を使用することをお勧めします。完全修飾名を使用しないと、Management Server と同じ DNS ドメインに属していないマシンでコンソールが実行されている場合に、一部の Management Pack ツールをコンソールから起動できない可能性があります。完全修飾名を使用してもコンソールから一部のツールを起動できない場合は、ローカルの 'hosts' ファイルに、Management Server が実行されているホストの IP アドレスを示すエントリを追加してください。

- 1つのウィンドウで項目を選択したときに、その前に選択した項目が消去されないため、2つの項目が同時に選択されているように見えることがあります。最後に選択した項目が実際に選択されている項目です。通常、新しいウィンドウで別の項目を選択すると、この問題を解決できます。(Bug#618771)
- コンソールのトレース・パラメータにおける構文の変更

すべてのクライアント・トレース・パラメータから OEM.接頭辞が削除されました。現在のトレース・パラメータは次のとおりです。

```
TRACING.ENABLED
```

```
TRACING.LEVEL
```

\$ORACLE_HOME/sysman/config ディレクトリにある clientconfig.properties ファイルに、これらのパラメータを追加してください。詳細は、『Oracle Enterprise Manager 構成ガイド』を参照してください。

- AIX 上でコンソールを実行している場合は、JRE に不具合があるため、コンソールでグループの移入や、その他のドラッグ・アンド・ドロップ操作ができません。
(Bug#1341098) グループを移入するには、サポートされている他の任意のオペレーティング・システム上でコンソールを起動します。
- HP-UX、IBM AIX および Compaq Tru64 では、ノードを検出するために「ナビゲータ」メニューから「ノードの検出」オプションを最初に使用したときに「ノードの検出」ダイアログ・ウィンドウが出現せず、エラーが発生する場合があります。この場合、「ナビゲータ」メニューから「ノードのリフレッシュ」を選択するか、もしくはマウスを右クリックして出現するメニューから「ノードの検出」を選択してください。
(Bug#1377707)

イベント・システムおよびジョブ・システム

イベントの手動消去

次の 5 つの拡張イベントは自動的に消去されません。

- データベース警告
- データ・ブロック破壊
- セッション終了
- アーカイバの停止
- Data Gatherer 警告

つまり、これらのイベントのいずれかが発生し、問題が手動で解決された場合、問題が解決されたことを自動的に検出する方法がないため、イベントは自動的に消去されません。

したがって、イベント発生の原因となった問題の解決後には、次のようにしてイベントを手動で削除する必要があります。

1. Enterprise Manager コンソールにあるイベント画面の「警告」タブで、発生したイベントを「履歴」タブに移動します。
2. 「履歴」タブで、このイベント発生を削除します。

これらの手順によりイベントが削除され、このイベントのために警告として設定されていたグループ画面の赤い旗も緑に変わります。

重複イベント

Enterprise Manager リリース 2.2 では、Agent リリース 8.1.7 に対してイベントが登録されている場合は、重複イベントを発行できます。重複イベントとは、同じパラメータを使用する同じイベント・テストを持ち、同一ターゲットに登録されたイベントのことです。たとえば、警告しきい値が 80% および警戒しきい値が 60% の表領域満杯テストを、同じ USERS 表領域に対して複数回登録できるということです。ただし、同じイベント内にこれと同じイベント・テストを持つことはできません。したがって、この例の場合、重複する表領域満杯テストは、異なるイベントに属する必要があります。

重複イベントは、リリース 8.1.7 より前の Agent では使用できません。特定のノード上で稼働している Agent のバージョンを調べるには、Enterprise Manager コンソールのナビゲータ・ツリーでそのノードを右クリックし、「プロパティ」を選択します。

リリース 8.1.6 以前の Agent で稼働している Enterprise Manager リリース 2.1 では、重複イベントは使用できません。

リリース 8.1.5 以前の Agent で稼働している Enterprise Manager リリース 2.0.4 でも、重複イベントは使用できません。

同義イベント・テスト

同義イベント・テストは、同じイベント内で同じ測定基準を監視するテストですが、使用するパラメータは異なります。たとえば、異なる表領域に登録されているが、同じイベントに属しているという 2 つの表領域満杯イベント・テストがそれに当たります。

同義イベント・テストは、Enterprise Manager リリース 2.2 およびそのすべてのサポート Agent でサポートされます。

Enterprise Manager リリース 2.0.4 リポジトリからの同義イベントの移行

Enterprise Manager リリース 2.0.4 リポジトリから Enterprise Manager リリース 2.2 リポジトリへアップグレードする際、Enterprise Manager リリース 2.0.4 に同義イベント（同一だが使用するパラメータが異なるイベント・テストを複数持つイベント）がある場合、リポジトリのアップグレード後に、リリース 2.2 コンソールを起動してから、これらの同義イベントの登録を解除し、再登録する必要があります。

ジョブ履歴またはイベント履歴をリフレッシュ

ジョブ履歴またはイベント履歴をリフレッシュする際、リフレッシュする履歴の量が多いと、最初は画面に何も表示されません。この問題を解決するには、スクロール・バーの位置を調整してください。（Bug#819774）

Windows 2000 のノードでのジョブの実行

Windows 2000 のノード上でジョブを実行する際、ジョブは「C:\winnt\temp\21041345280.99が見つかりません」というエラーで失敗する可能性があります。Windows 2000 では異なるオペレーティング・システム・レベルのセキュリティを設定できるため、TEMP ディレクトリに対する権限が不十分なことから、このエラーが生じます。この問題が発生した場合は、どのユーザーも TEMP ディレクトリ内でファイルを作成、更新および削除できることを確認してください。TEMP ディレクトリは、TEMP 環境変数で表されるディレクトリです。また、TEMP 環境変数で示されるディレクトリにも、作成、更新および削除権限があることを確認してください。最後に、ジョブを再発行する必要があります。

Compaq Tru64 での「パラメータ」タブの表示

Compaq Tru64 では、新規ジョブまたはイベントが作成された場合、「パラメータ」タブは正しく表示されません。この場合、一旦異なるタブを選択し、その後に再度「パラメータ」タブを選択すると、正常に表示されます。(Bug#1377606)

グループ

- ナビゲータ・ツリーで「グループ」フォルダまたは特定のグループを選択した場合、ナビゲータのメイン・メニューで「グループ作成」を選択しても、何も起こりません。これを回避するには、「グループ」メイン・メニューの「グループ作成」メニュー項目を選択するか、または右クリックをして「作成」を選択します。(Bug#1283307)
- AIX 上でコンソールを実行している場合は、JRE に不具合があるため、コンソールでグループの移入や、その他のドラッグ・アンド・ドロップ操作ができません(Bug #1341098)。グループを移入するには、サポートされている他の任意のオペレーティング・システム上でコンソールを起動し、そのコンソールからグループを移入します。

ポケットベルおよび電子メールのブラックアウト

- ポケットベルと電子メールのブラックアウトを特定の曜日に設定し、指定した曜日の 24 時を超えるブラックアウトを実行できません。たとえば、水曜日の午後 11 時 30 分に開始して木曜日の午前 1 時 30 分まで続く 2 時間のブラックアウトは作成できません。これを行うには、水曜日の午後 11 時 30 分から午前 12 時と、木曜日の午前 12 時 1 分から午前 1 時 30 分という、2 つの別々のブラックアウトを作成する必要があります。(Bug#1036820)
- 「間隔指定」で実行されるブラックアウトを定義する場合は、ブラックアウトの「有効期間」には終了日を設定しないでください。(たとえば、有効期間は、「開始: 07/30/00 終了: <指定しない>」のように指定します。) 終了日を指定すると、その期間の最終サイクルでブラックアウトが実行される問題が発生します。終了日を指定しない状態にして、この問題を回避してください。(Bug#1367312)

DBA Management Pack

- スキーマ名にデリミタとして"."が含まれていると、このスキーマの下にあるオブジェクトを表示する際に、詳細が空になります。(Bug#1055740)
- サマリー・アドバイザ・ウィザードを使用するには、listener.ora ファイルおよび tnsnames.ora ファイルに外部プロシージャ・エージェントを設定する必要があります。詳細は、『Oracle8i Net8 管理者ガイド』を参照してください。

- OS 認証は、データベースに直接接続しているとき（クライアント / サーバー・モード）にのみ機能します。Management Server を介して接続しているときには、OS 認証はサポートされません。
- SQL*Plus Worksheet の実行中に「SP2-0738 Restricted command」エラーが発生すると、SQL*Plus が Oracle Management Server に接続する際、サーバー側のファイル・システムにアクセスできません。この場合、Oracle Management Server マシン上でスクリプトの実行が試みられ、SQL*Plus プロセスが制限モードで実行されます。セキュリティ上の理由で、クライアント側のスクリプトのみが、「ローカル・スクリプトの実行...」メニューにより実行できます。詳細は、「SQL*Plus Worksheet ヘルプ」を参照してください。
- レプリケーション（サーバー側）は、レプリケーション環境を定義するために、DBA Studio Replication Management オプションを使用する前にインストールする必要があります。レプリケーションがインストールされていないと、ウィザードで実行されるタスクはすべて失敗します。
- LONG または LONG RAW データ型の列が含まれる索引構成表（IOT）はパーティション化できないため、LONG または LONG RAW データ型の列が主キー列でない場合は、エラー・メッセージが正しく表示されなくなり、パーティションが入力可能になります。IOT に LONG または LONG RAW データ型の列が含まれ、それが主キーの場合は、「LONG/LONG RAW データ型列を持つ表はパーティション化できません」というエラー・メッセージが正しく表示されます。
- データベース・レポートのスケジューリングは、ユーザーが SYSDBA としてログインし、そのユーザーのパスワードが NORMAL での接続時と SYSDBA での接続時で異なる場合は失敗します。これを回避するため、レポートのスケジューリングには NORMAL で接続してください。SYSDBA で接続すると、レポートのスケジューリングでは無視されますので、注意してください。（Bug#1360896）
- レプリケーション・ウィザードは、DBA Studio からのみ起動できます。Enterprise Manager コンソールを使用する場合、まず DBA Studio を起動してからレプリケーション・ウィザードを起動してください。（Bug#1365244）
- SQL*Plus Worksheet を英語以外の言語で使用するには、最初に<Oracle ホーム>%sysman%config%ディレクトリにある dbappscfg.properties ファイルを変更する必要があります。このファイルを変更するには、SQLPLUS_NLS_LANG の値を追加します。この値はデフォルトの NLS_LANG パラメータと同じになるようにしてください。
- DBA Studio で格納された構成を編集するとき、マウスの右クリックにより出現したメニューで「編集」を選択し、さらに出現したダイアログで「OK」をクリックした場合、エラーが発生する場合があります。このとき、変更はセーブされていますので「取消」を選択してダイアログを終了してください。（Bug#1377721）

Oracle Management Server

Windows での Oracle Management Server サービスの作成

Windows NT および Windows 2000 では、サービスとしての OMS の作成は、Enterprise Manager Configuration Assistant (EMCA) によって実行されるようになりました。つまり、OMS サービスは、インストール時に、新規リポジトリを作成または既存リポジトリを使用するために EMCA を使用した場合にのみ作成されます。

Oracle Management Server (OMS) サービスは「手動」起動タイプで作成されます。これは、構成が不適切または不十分な OMS の自動起動を防ぐために設計されました。OMS が適切に構成され、テストが行われると、OMS のサービス・タイプを必要に応じて「自動」に変更できます。

次の手順で、これを実行できます。

1. 「コントロール パネル」の「サービス」オプションを起動します。
2. OMS サービス（「Oracle<Oracle ホーム名>ManagementServer」という名前）を選択します。
3. 「スタートアップ」ボタンをクリックし、適切な「スタートアップの種類」タイプのラジオ・ボタンを選択します。

強化された Oracle Management Server 起動

oemctrl ユーティリティを介して OMS を起動すると、OMS の起動時に進捗インジケータが表示されます。OMS が完全に起動した場合にのみ、OMS が起動したことを示すメッセージが表示されます。OMS の起動時に問題が生じた場合は、該当するエラーが発生します。oemctrl を使用して Windows NT または Windows 2000 で OMS を起動し、起動が完了すると、「コントロール パネル」内の対応するサービスにも「実行中」ステータスが反映されます。

Windows NT または Windows 2000 上のサービスとして OMS を起動すると、OMS が完全に起動した後にのみ、ステータスが「実行中」になります。

起動時に問題が発生した場合は、\$ORACLE_HOME/sysman/log ディレクトリにある oms.nohup ファイルを参照してください。

Oracle Management Server の停止

Windows NT および Windows 2000 上で、「コントロール パネル」からサービスとしての OMS を停止すると、停止を実行するための接続情報を尋ねるダイアログ・ボックスが表示されます。指定する接続情報は、有効な Oracle Enterprise Manager スーパーユーザー・アカウントです。

Oracle Management Server のステータス

OMS のステータスを調べるには、新しい `oemctrl` オプション、`oemctrl ping oms` を使用できます。このオプションは OMS が使用可能かどうかを調べるのに適しています。接続情報は必要ありません。

Solaris から HP への接続

OMS が Solaris ノードで、Enterprise Manager リポジトリ・データベースが HP 10.20 ノードで稼働している場合は、次の `SQLNET.ORA` パラメータを OMS ノードで設定する必要があります。

```
DISABLE_OOB=ON
```

`SQLNET.ORA` ファイルは、`$ORACLE_HOME/network/admin` ディレクトリにあります。`$ORACLE_HOME` は、OMS がインストールされているディレクトリです。このパラメータは、Solaris と HP の TCP 層による同じソケットでの同時緊急メッセージの処理方法が異なるために生じる `sqlnet` 接続の停止を防ぎます。

このパラメータを設定した後、OMS を停止し、再起動する必要があります。

Enterprise Manager Configuration Assistant

リリース 2.2 より前のリポジトリ

リリース 2.2 より前の既存リポジトリは、インストール時に自動的にアップグレードされません。リポジトリに対するアップグレードは、各 Enterprise Manager 製品でそのリポジトリを使用する既存ユーザーに影響を与えます。リポジトリのユーザー全員が、コンソール、DBA Studio、Management Pack など、各自のソフトウェアをリリース 2.2 にアップグレードするまで、リポジトリのアップグレードは実行しないでください。リポジトリのアップグレードに適した時期をスケジュールした後、EMCA を使用してアップグレードを実行します。

Enterprise Manager Configuration Assistant によるリポジトリのアップグレード

リポジトリのアップグレードを実行するには、明示的に EMCA を実行し、「アップグレード」オプションを選択する必要があります。「アップグレード」オプションを選択し、DBA 権限を持つアカウントを使用してログインすると、EMCA にはアップグレード可能な既存のリポジトリが表示されます。リポジトリがリストにない場合は、EMCA によってアップグレードできないことを意味します。アップグレードするリポジトリを選択し、操作を続けます。

Windows NT または Windows 2000 上で EMCA を実行中の場合は、EMCA のアップグレードにより、まだ作成されていない OMS サービスも作成されます。

Enterprise Manager Migration Assistant

一部のサービスについて、優先接続情報リストをリリース 1.x リポジトリからリリース 2.2 リポジトリに移行できない場合があります。この場合は、リリース 2.2 コンソールで、そのサービスの優先接続情報リストを手動で設定する必要があります。(Bug#751627)

ブラウザ・ベースの Enterprise Manager

- ブラウザ・ベースの Enterprise Manager にログインした後、コンソールが停止していることに気づいた場合は、次の項目を確認します。

- (1) 使用している Web ブラウザでプロキシが設定されていないこと
- (2) 使用されている MSVCRT.DLL がバージョン 5.00.7303 より後のバージョンであること

MSVCRT.DLL のバージョンは、システム・パスでこの dll を検索することにより調べられます。パスで見つかった最初の dll は、使用中のものではないことがときどきあります（この dll は c:\winnt\40\system32 からロードされたものと考えられます）。使用中の dll を確認するには、コンソールの停止中にその dll の名前を変更してみてください

（Windows NT エクスプローラでは、使用中の dll の名前を変更できません）。dll のバージョンは、プロパティを参照することで確認できます。dll を置換するには、次の手順で行います。

DOS プロンプトからの場合:

1. dll が存在するディレクトリにディレクトリを変更します。
 2. msvcrt.dll の名前を「_msvcrt.dll」に変更します。
 3. 正しいバージョンの msvcrt.dll を元のディレクトリにコピーします。
 4. マシンを再起動します。
- Web サーバーに初めて接続すると、jar ファイルがダウンロードされます。ブラウザ・プロセスは、ダウンロード時に多くのメモリーを消費するように見えることがあります。ダウンロードの完了後、ブラウザを終了し、Enterprise Manager コンソールを起動する前に再起動してください。
 - Enterprise Manager の起動ページから Enterprise Manager コンソール、DBA Studio または SQL Worksheet を起動すると、選択したアプレットを実行するために別のブラウザ・ウィンドウが作成されます。これにより、元のブラウザ・ウィンドウは別の目的に使用できます。作成されたブラウザ・ウィンドウは閉じないでください。閉じると、アプレットも停止してしまいます。

- 複数の言語をインストールして複数の言語を使用して Web ブラウザから Enterprise Manager を使用するには、以下の URL を使用してください。
http://<マシン名>:3339/EmWebSite_xx.html (xx は言語名)
 もし一つの言語で使用する場合は、EmWebSite_xx.html を EmWebSite.html にコピーし、
<http://<マシン名>:3339/> という URL を使用してください。

その他

- Enterprise Manager、または Enterprise Manager に統合されているアプリケーションを Windows 95 または Windows 98 で起動すると、そのアプリケーションと同時に起動された DOS ウィンドウが表示されます。DOS ウィンドウは閉じないでください。DOS ウィンドウを閉じると、起動されていた Enterprise Manager アプリケーションが終了してしまいます。(Bug#1161051)
- JRE は、各 Oracle 製品によりインストールされます。ただし、共有 JRE は 1 システムにつき 1 つに限られます。したがって、Enterprise Manager リリース 2.2 のインストール後に前のリリースの Enterprise Manager (リリース 2.0 または 2.1) をインストールすると、リリース 2.2 の一部としてインストールされた JRE (Java Runtime Environment) が、自動的にダウングレードされ、エラー・メッセージは表示されません。JRE リリース 1.1.7.30o から JRE リリース 1.1.7.5o または 1.1.7.24o へのダウングレードは、Enterprise Manager リリース 2.2 の問題の原因となります。したがって、Enterprise Manager リリース 2.2 のインストール後に、Enterprise Manager リリース 2.0 または 2.1 をインストールしないでください。

リリース 8.1.7 より前の Oracle データベースをインストールしても、同様に JRE のダウングレードという問題が生じます。

ダウングレードした場合は、Enterprise Manager リリース 2.2 をインストールした CD-ROM (ほとんどの場合はリリース 8.1.7 CD-ROM) から JRE リリース 1.1.7.30o を再インストールすることにより、問題を回避できます。Oracle Universal Installer を再インストールして、JRE を再インストールします。Installer は、JRE に従属しているためです。

- Enterprise Manager の英語以外のバージョンでは、英語以外のオンライン・ヘルプで一部矛盾が生じる可能性があります。最新情報はこのリリース・ノートを参照してください。
- サイレント・インストール中、リポジトリを作成/編集する「emca」ツールは自動的に起動しません。サイレント・インストール終了後、手動で「emca」ツールを起動してください。(Bug#1379570)

第 9 章 Oracle HTTP Server

次の項目について説明します。

- サポートする JDK/JRE
- Oracle HTTP Server powered by Apache の Global Server ID
- 既知の制約事項と制限事項

サポートする JDK/JRE

オラクル社では、`mod_jserv` でユーザー Java アプリケーションを構築および配置するために、JDK/JRE 1.1.8 と JDK/JRE 1.2.x の両方の使用を検証しました。このリリースでは、すべての Java クラス・ライブラリ（たとえば、Oracle BC4J (Business Components for Java)、Oracle XML Developer's Kit など）を JDK 1.1.8_10 でコンパイルして出荷しています。これらのコンパイル済クラスは、Java JRE でサポートされている両方のバージョンで実行できることが検証されていますので、再コンパイルの必要はありません。

Oracle HTTP Server powered by Apache の Global Server ID

Apache サーバーに適切な Global Server ID (GS-ID) 証明書が含まれていて、ブラウザで GS-ID 証明書を受け入れられるようにパッチが適用されている場合は、GS-ID を使用してエクスポート・レベルのブラウザを正式にアップグレードし、高度な暗号化（128 ビット）を使用を可能にします。すべてのブラウザが高度な暗号化で出荷されますが、エクスポート製品については、暗号化は使用できません。米国輸出法が最近更新されたため、GS-ID の使用が将来的に必要なことに注意してください。しかし、現時点では、エクスポート級のブラウザがあり高度な暗号化を必要とする場合には、下記の手順に従って、GS-ID 証明書を入手してブラウザの使用を可能にしてください。

1. GS-ID 証明書の購入

適切なベンダーから GS-ID 証明書を入手してください。オラクル社では、Verisign (<http://www.verisign.com/server/prd/g/index.html>) の GS-ID 証明書をテストしています。証明書をダウンロードし、サーバーに格納するための手順に従ってください。証明書を入手した後、Oracle HTTP Server の管理者は、`httpd.conf` ファイルを更新する必要があります。更新する行は、次のとおりです。

```
SSLCertificateFile <pathname>/gsid.crt
SSLCertificateKeyFile <pathname>/gsid.key
SSLCertificateChainFile <pathname>/gsidintermediate.crt
```

この場合の *pathname* には、インストール済みの Verisign ファイルへの完全修飾パスが入ります。

2. ブラウザ・パッチの購入

ブラウザが暗号化メソッドをアップグレードできるようにするパッチを入手してください。ソースには、Apache (www.apache.org) と Fortify (www.fortify.net/intro.html) があります。オラクル社では Fortify のパッチをテストしています。ダウンロードおよびブラウザへの適用が容易に行うことができます。

既知の制約事項と制限事項

Oracle HTTP Server を使用する前に、各項目を読み、リリースの制約／制限事項を理解してください。

- Oracle HTTP Server におけるサード・パーティ製コンポーネントのサポート
- `mod_perl` DBI/DBD-Oracle および `Apache::DBI` `mod_plsql` 要件
- `mod_plsql` 要件
- Oracle PL/SQL Web ツールキットのインストールに関する障害
- Oracle8 リリース 8.0.x への PL/SQL Web Toolkit のインストールに関する障害
- Oracle Demo 証明書の置換
- 新しい SSL 証明書が必要
- OPENSSL コマンドの「-CONFIG」オプションの要件
- `mod_ssl` の制限事項
- インストール時の Apache 構成エラー
- 仮想ホストの使用時に Apache を「HTTPDSCTL START」で始動した場合のエラー
- 各国語サポート (NLS 考慮事項)
- Apache JServ アプリケーションのメモリー増加
- その他のリリース・ノート
- `mod_ose` に関する障害
- `mod_plsql` 日付型の引数について
- `mod_rewrite` のセキュリティ上の危険性
- OpenSSL の CA 機能の非サポート

Oracle HTTP Server におけるサード・パーティ製コンポーネントのサポート

－ すべてのサード・パーティ製コンポーネント

Oracle HTTP Server の一部は、Apache Software Foundation によってライセンスされている Apache Web Server バージョン 1.3.12 などのように、サード・パーティのライセンス下でオラクル社によって配布されています（サード・パーティ製コンポーネント）。オラクル社は、これらのサード・パーティ製コンポーネントを、Oracle HTTP Server 製品の一部として配布し、特にここで指定している場合を除いて、サード・パーティ製コンポーネントに対して、標準の製品サポートを提供しています。そのサポートの詳細は、オラクル社の技術サポートの方針に定義されています。オラクル社は、サード・パーティ製コンポーネントの Oracle HTTP Server で出荷されるバージョンのみをサポートしています。インターネット上で自由に入手できる他のバージョンについては、サポート対象外です。

オラクル社は、サービスとして、サード・パーティ製コンポーネントに関する拡張要求を、そのコンポーネントをオラクル社にライセンスしている団体に対して問い合せていますが、オラクル社がその義務を負っているものではありません。さらに、非サード・パーティ製コンポーネント（オラクル・コンポーネント）に対する技術的な支援要求（トラブルなど）も、オラクル・コンポーネントのみで構成されている環境で再現できない限り、正式に提起することはできません。

－ Apache モジュール

Oracle HTTP Server の一部として Apache Web Server で提供されている機能を拡張するために、オラクル社は Apache Web Server に対する拡張部分（Apache モジュール）を配布する場合があります。オラクル社から配布されている Apache モジュールは、このリリース・ノートでは、Oracle Apache モジュールとして参照します。

オラクル社は、このリリースで以下の Oracle Apache モジュールをサポートしています。

Oracle Apache モジュール	説明
mod_ssl	SSL サポート。
mod_perl	Perl で Apache モジュールを作成するためのサポート。
mod_jserv	サーブレット・エンジンとの通信。
mod_plsql	PL/SQL サポート。
mod_ose	Oracle Servlet Engine (OSE) のステートフル Java と PL/SQL サーブレットに URL を委譲する。
http_core	中核の Apache 機能。

Oracle Apache モジュール	説明
<code>mod_access</code>	ホスト・ベースのアクセス制御。クライアントのホスト名またはIPアドレスに基づいたアクセス制御を提供します。
<code>mod_actions</code>	ファイル・タイプ/メソッド・ベースのスクリプトの実行。メディア・タイプまたは要求メソッドに基づいた CGI スクリプト用に提供します。
<code>mod_alias</code>	別名およびリダイレクト。ドキュメント・ツリー上のホスト・ファイル・システムの異なる部分のマッピングおよび URL のリダイレクト用に提供します。
<code>mod_auth</code>	テキスト・ファイルを使用したユーザー認証。
<code>mod_auth_anon</code>	無名ユーザー認証、FTP 形式。
<code>mod_autoindex</code>	自動ディレクトリ・リスティング。
<code>mod_cgi</code>	CGI スクリプトの実行。MIME タイプ・アプリケーション/x-httpd-cgi のファイル进行处理します。
<code>mod_define</code>	構成の定義。
<code>mod_digest</code>	MD5 の認証。MD5 Digest 認証を使用する、ユーザー認証用に提供します。
<code>mod_dir</code>	基本ディレクトリ処理。後書きスラッシュのリダイレクトおよびディレクトリ索引ファイル用に提供します。
<code>mod_env</code>	環境を CGI スクリプトにパスするもの。環境変数を CGI/SSI スクリプトに渡すために提供します。
<code>mod_expires</code>	Expires:ヘッダーのリソースへの適用。ユーザー指定の基準に従って、Expires ヘッダーを生成するために提供します。
<code>mod_headers</code>	任意の HTTP ヘッダーをリソースに追加する。ヘッダーは、マージ、置換または削除することができます。
<code>mod_include</code>	サーバーで解析されるドキュメント。サーバー解析 HTML ドキュメント用に提供します。
<code>mod_info</code>	サーバー構成情報。構成ファイル上のすべてのインストール済モジュールとディレクティブを含む、サーバー構成の包括的な概要を提供します。

Oracle Apache モジュール	説明
mod_log_config	mod_log_common に置き換えられるユーザーによる構成可能なロギング。Common Log Format またはユーザー指定フォーマットを使用した、サーバーに対するロギング要求用に、提供します。
mod_log_referer	ドキュメント参照のロギング。サーバー上のドキュメントを参照するドキュメントのロギング用に提供します。
mod_mime	ファイル拡張子を使用するドキュメント・タイプの決定。
mod_mime_magic	マジック番号を使用するドキュメント・タイプの決定。コンテンツの数バイトを見てファイルの MIME タイプを決定するために使用されます。
mod_negotiation	コンテンツ・ネゴシエーション。
libproxy (mod_proxy)	キャッシング・プロキシ機能。HTTP 1.0 キャッシング・プロキシ・サーバー用に提供します。
mod_rewrite	正規表現を使用する URL とファイル名の強力なマッピング。稼働中に要求 URL を書き換えるためにルールベースの書き換えエンジンを提供します。
mod_setenvif	クライアント情報に基づいて環境変数を設定する。要求の属性に基づいて環境変数を設定する機能用に提供します。
mod_so	実行時のモジュールのローディング・サポート。起動時あるいは再起動時に、実行可能プログラムとモジュールをサーバーにローディングするために提供します。
mod_speling	URL の微妙な入力ミスを自動的に修正する。ユーザーが入力した URL の複数のスペルミス、大文字使用を無視したり、スペルミスを 1 つまで容認するなどして、修正をはかります。
mod_status	サーバーのステータス表示。現行サーバーの統計を読みやすい形で示す HTML ページで表示して、サーバー管理者がサーバーの稼働状況を確認できるようにします。
mod_unique_id	各要求に対する固有の要求識別子を生成する。 特定の状況下のすべての要求にわたって固有であると保証されている各要求ごとのマジック・トークンを提供します。

Oracle Apache モジュール	説明
mod_userdir	ユーザーのホーム・ディレクトリ。ユーザー固有のディレクトリ用に提供します。
mod_usertrack	Cookies を使用したユーザー追跡。
mod_vhost_alias	動的に構成される大規模仮想ホスティングのサポート。

Oracle HTTP Server に含まれるその他の Oracle Apache モジュールは、保証やサポートのないそのままの状態を提供されています。Apache Software Foundation またはそのユーザーを含む、オラクル社以外から提供されている Apache モジュールは、サポート対象外です。

さらに、Oracle Apache モジュールのみを含む Apache の構成で再現できる問題に対してのみ、技術的なサポートを提供します。

mod_perl DBI/DBD-Oracle および Apache::DBI

今回のリリースでは、以下のものが含まれません。

- Apache::DBI
- Perl DBI
- Perl DBD-Oracle

今回のリリースでは、DBI/DBD-Oracle および Apache::DBI を用いたデータベース接続は、サポート対象外です。Perl インタプリタ本体はサポートします。

mod_plsql 要件

特定のバックエンド・データベースに対して mod_plsql モジュールを使用するには、バックエンド・データベースに手動で Oracle PL/SQL Web Toolkit (OWA PL/SQL パッケージ) をインストールする必要があります。

Oracle8i リリース 8.1.7 には、あらかじめ OWA PL/SQL パッケージがインストールされています。

Oracle8 リリース 8.0.x および Oracle8i リリース 8.1.6 に対しては、手動で OWA PL/SQL パッケージをインストールする必要があります。OWA PL/SQL パッケージは、SYS データベース・スキーマにインストールしてください。OWA PL/SQL パッケージは必ずシステムに 1 つのみインストールするようにしてください。既存の Oracle Application Server ユーザーがアップグレードする場合は、旧バージョンを新しいバージョンに置き換えてください。

詳細は、『Oracle9i Application Server PL/SQL ゲートウェイの使用』を参照してください。または、http://hostname.domain:port/pls/admin_/help/title.htm で参照可能なオンライン・ドキュメントにアクセスすることもできます。

Oracle PL/SQL Web ツールキットのインストールに関する障害

特定のバックエンド・データベースに、\$ORACLE_HOME/Apache/modplsql/owa 上の owaload.sql を使用した場合、手動による Oracle PL/SQL Web Toolkit (OWA PL/SQL パッケージ) のインストールは正常に終了しません。

OWA PL/SQL パッケージをインストールする際には、\$ORACLE_HOME/rdbms/admin 上の owaload.sql を使用してください。

Oracle8 リリース 8.0.x への PL/SQL Web Toolkit のインストールに関する障害

Oracle PL/SQL Web Toolkit (OWA PL/SQL パッケージ) をインストールする前に以下のスクリプトを一部変更する必要があります。

- \$ORACLE_HOME/rdbms/admin/privcach.sql
- \$ORACLE_HOME/rdbms/admin/pubcach.sql

上記スクリプトを<スクリプト名>.nocopy などの別名でバックアップを取得してから、下記の 1 行を修正してください。

《変更前》

```
PROCEDURE init(p_htbuf IN OUT NOCOPY http.htbuf_arr,... (以下省略)
```

《変更後》

```
PROCEDURE init(p_htbuf IN OUT http.htbuf_arr,... (以下省略)
```

Oracle Demo 証明書の置換

Oracle Demo 証明書(これは架空の証明書です)が Oracle HTTP Server のビルドに含まれていますが、これはテスト環境でのみ使用可能です。本番環境に移る前に、Oracle Demo 証明書を実際に利用する証明書に置き換える必要があります。

新しい SSL 証明書が必要

Oracle Application Server (OAS) から Oracle HTTP Server にアップグレードするには、リスナーを Spyglass から Apache へ変換されるため、新しい SSL 証明書を取得する必要があります。

OPENSSL コマンドの「-CONFIG」オプションの要件

証明書または証明書要求を作成する openssl コマンドは、
\$ORACLE_HOME/Apache/open_ssl/bin 上の openssl.cnf ファイルの位置を指定するために、
「-config」オプションを付ける必要があります。設定しない場合、このコマンドは失敗します。

たとえば、次のように行います。

```
$ openssl req -config $ORACLE_HOME/Apache/open_ssl/bin -new -x509 -days  
365 -key ca.key -out ca.crt
```

mod_ssl の制限事項

このリリースでは、次のアルゴリズムはサポート対象外です。

- 対称型暗号化アルゴリズム : RC2、RC5、IDEA
- ハッシュ・アルゴリズム : MD5

インストール時の Apache 構成エラー

Oracle HTTP Server をインストール中、同じ ORACLE_HOME 上にすでに別の Apache がインストールされている場合、Apache の構成が次のエラーで失敗します。

```
Syntax Error on line 14 of  
<ORACLE_HOME>/Apache/Apache/conf/mod_ose.conf  
Aurora Service - directive already effect for this server  
<ORACLE_HOME>/Apache/Apache/bin/httpsdctl start: httpd could not be  
started
```

このエラーを回避するには、次のようにします。

1. \$ORACLE_HOME/Apache/Apache/conf/oracle_apache.conf ファイル上の二重の項目を削除します。
2. 次のように Oracle HTTP Server を再起動します。

```
$ $ORACLE_HOME/Apache/Apache/bin/httpsdctl start
```

仮想ホストの使用時に Apache を「HTTPDCTL START」で始動した場合のエラー

ポート番号のディレクティブと NameVirtualHost のディレクティブ・ポートが一致しない場合、このエラーが発生します。この問題を解決するには、ポートと NameVirtualHost が同一になるよう設定する必要があります。

また、NameVirtualHost ディレクティブに port#を指定しないことによっても、この問題を解決できます。この問題は、SSL なしで始動する場合にのみ発生します。

各国語サポート（NLS 考慮事項）

– mod_plsql

mod_plsql を構成する時、環境変数 NLS_LANG は、DAD レベルではなく、Web サーバー・インスタンスごとのレベルで構成します。

Oracle HTTP Server を開始する前に、正しい環境変数 NLS_LANG を設定してください。

– Oracle PSP

Oracle PSP では、loadpsp コマンドを使用して PL/SQL Server Pages（PSP）をデータベースにロードする前に、環境変数 NLS_LANG を定める必要があります。

– Jserv および cleJSP で JDBC OCI8 ドライバを使用する場合

JDBC OCI8 ドライバ経由で Oracle に接続する場合、jserv.properties に適切な環境変数 NLS_LANG 設定してください。たとえば、次のようにします。

```
wrapper.env=NLS_LANG=AMERICAN_AMERICA.UTF8
```

環境変数 NLS_LANG の詳細は、『Oracle8i NLS ガイド』を参照してください。

Apache JServ アプリケーションのメモリー増加

Oracle8i JVM により、JServ ログおよびブラウザでメモリー不足が発生する場合があります。これは、通常 Oracle8i JVM が扱うデータが、メモリー割当てプールの超過したときに発生します。

Oracle8i JVM 用のメモリー割当てプールの最大サイズを増加させるには、次の手順を使用してください。

1. \$ORACLE_HOME/Apache/Jserv/etc/jserv.properties に次の行を追加します。

```
wrapper.bin.parameters=-mx<size>m
```

この場合の<size>は、メモリー割当てプールのメガバイト単位での大きさです。デフォルト値は、1 メガバイトです。オラクル社は、128 メガバイトのサイズの使用をお勧めします。値を 128 メガバイトに設定するには、次の行を追加してください。

```
wrapper.bin.parameters=-mx128m
```

2. 変更後、この設定を有効にするために Oracle HTTP Server を再起動してください。

その他のリリース・ノート

各 Oracle HTTP Server コンポーネントをインストールすると、各コンポーネントのリリース・ノートは、\$ORACLE_HOME/relnotes ディレクトリにインストールされます。リリース・ノートの詳細は、インストール後に次のファイルを参照してください。

- Oracle XML Developer's Kit (XDK)
- \$ORACLE_HOME/xdk/index.html
- OracleJSP
- \$ORACLE_HOME/jsp/doc/releasenotes_11000.txt

mod_ose に関する障害

Oracle HTTP Server では、\$ORACLE_HOME/Apache/Apache/mod_ose.conf 中で、SetHandler aurora-stateless-server と設定された URL を呼び出すと、ブラウザの画面に表示されるコンテンツの先頭にそのサイズが表示されることがあります。

SetHandler aurora-server と設定することで、この問題を回避してください。

mod_plsql 日付型の引数について

mod_plsql を使用したデータベースのストアード・プログラムを呼び出し時に、日付型の引数のフォーマットが正しく解釈されない場合があります。この不具合は将来のリリースで修正されます。

この問題を回避するためには、以下の作業を実行してください。

1. SYS ユーザーでデータベースにログインし、次のコマンドを実行し、ストアード・プロシージャを作成します。日付書式フォーマットは、必要に応じて変更します。

```
create or replace procedure fix_date_format is
begin
  DBMS_SESSION.SET-NLS('NLS_DATE_FORMAT','YYYY-MM-DD');
end;
/
```

2. 次のコマンドを実行して、手順 1 で作成したストアード・プロシージャを全ユーザーより実行可能にします。

```
grant execute on fix_date_format to public;
```

3. Oracle HTTP Server を停止します。

```
$ $ORACLE_HOME/Apache/Apache/bin/httpdctl stop
```

4. \$ORACLE_HOME/Apache/modplsql/cfg/wdbsvr.app ファイルを編集し、各 DAD の before_proc を以下のように変更します。

《変更前》

```
;before_proc =
```

《変更後》

```
before_proc = sys.fix_date_format
```

5. 最後に、Oracle HTTP Server を再起動してください。

```
$ $ORACLE_HOME/Apache/Apache/bin/httpdctl start
```

mod_rewrite のセキュリティ上の危険性

mod_rewrite で一部のルールを使用することで、Web サーバー上のあらゆるファイルにアクセス可能になるというセキュリティ上の脆弱性が判明しています。この問題を回避するには、書き換えルールを直接ファイルにマップするのではなく、すべて URL にマップします。

たとえば、ご使用の Web サーバーで DocumentRoot が /webroot に設定されている場合、次のルールは使用しないでください。

```
RewriteRule /foobar/(.*) /webroot/myfiles/$1
```

この場合、リクエストが直接ファイル・システムの場所にマップされているためです。かわりに、次のルールを使用します。

```
RewriteRule /foobar/(.*) http://myserver.mydomain.com/myfiles/$1
```

この場合、アクセスは、Apache インスタンスによってアクセス可能なファイル（つまり、DirectoryRoot ディレクトリ・ツリーに含まれるファイル）のみに制限されます。

OpenSSL の CA 機能の非サポート

OpenSSL の認証局（CA）機能はサポートされていません。

第 10 章 Oracle JDBC ドライバ

このリリースでの新機能について

このリリースでの主な新機能/拡張機能は、次のとおりです。

- 文のキャッシュ
 - 暗黙的な文のキャッシュ
 - 明示的な文のキャッシュ
- XA の完全サポート
 - XA Recover と Forget の包含
 - リリース 8.1.7 以降のサーバーに対する OracleXid の個別実装
- 接続キャッシュ
 - 新規スキーム (FIXED_WAIT_SCHEME)
 - 文のキャッシュの結合
- スカラー型の PL/SQL 表のサポート (OCI ドライバのみ)
- ユーザー定義データ型のパフォーマンス強化
- オブジェクト型の拡張
 - シリアライズ可能な型記述子
 - Java 基本型のコレクション要素へのアクセス
 - コレクション要素のバッファリングと索引付け
 - 空の LOB の作成
- JDBC Thin ドライバを使用した接続に対する 56 ビット暗号化アルゴリズムのサポート

このリリースで修正された不具合は、主に次のとおりです。

- **Bug#903011**
サーバーが UTF8 キャラクタ・セットを使用すると、JDBC Thin ドライバでは Latin-1 のキャラクタを含んだユーザー名を使用できませんでした。リリース 8.1.7 では、この問題が修正されています。
- **Bug#1052489**
PreparedStatements が複数回実行され、バインド変数 (setBinaryStream または setCharacterStream にバインドされている) の 1 つの長さが増加する問題がありました。リリース 8.1.7 では、この問題が修正されています。
- **Bug#1069768**
JDBC Thin ドライバには 4K を超えるイメージの ADT の挿入に問題がありました。リリース 8.1.7 では、この問題が修正されています。
- **Bug#1247015**
スクロール可能な結果セットの CHAR または VARCHAR の列へのアクセスに ResultSet::getObject() を使用すると、ResultSet::getObject() から NULL が戻りました。リリース 8.1.7 では、この問題が修正されています。
- **Bug#1349713**
列データにマルチバイト文字が含まれていると、スクロール可能な結果セットの getString() で不適切な値が戻りました。リリース 8.1.7 では、この問題が修正されています。

ドライバのバージョンについて

リリース 8.1.7 でのドライバのバージョンは、次のとおりです。

- JDBC OCI ドライバ 8.1.7
OCI 8.1.7 がインストールされているマシンで使用するクライアント側の JDBC です。
- JDBC Thin ドライバ 8.1.7
アプレットとアプリケーションで使用する 100% Java のクライアント側 JDBC です。
- JDBC Thin サーバー側ドライバ 8.1.7
Oracle8i リリース 8.1.7 で実行する Java ストアド・プロシージャまたは Java CORBA オブジェクトが使用する JDBC です。一般的に、このドライバは中間層サーバーで使われます。
- JDBC サーバー側内部ドライバ 8.1.7
Oracle リリース 8.1.7 で実行する Java ストアド・プロシージャまたは Java CORBA オブジェクトが使用するサーバー側の JDBC です。このドライバは、JDBC Kprb ドライバと呼ばれていました。

詳細：『Oracle8i JDBC 開発者ガイドおよびリファレンス』を参照してください。

このリリースの内容について

\$ORACLE_HOME/jdbc/lib ディレクトリには、次のファイルが含まれています。

- `classes111.zip`
JDK 1.1.x で使用するクラスです。オブジェクト型とコレクション型の NLS サポートに必要なクラスを除く JDBC ドライバ・クラスが含まれています。
- `nls_charset11.zip`
JDK 1.1.x で使用する NLS クラスです。オブジェクト型とコレクション型の NLS サポートに必要なクラスが含まれています。
- `classes111_g.zip`
クラスが `javac -g` でコンパイルされていることを除き、`classes111.zip` と同じです。
- `classes12.zip`
JDK 1.2.x で使用するクラスです。オブジェクト型とコレクション型の NLS サポートに必要なクラスを除く JDBC ドライバ・クラスが含まれています。
- `nls_charset12.zip`
JDK 1.2.x で使用する NLS クラスです。オブジェクト型とコレクション型の NLS サポートに必要なクラスが含まれています。
- `classes12_g.zip`
クラスが `javac -g` でコンパイルされていることを除き、`classes12.zip` と同じです。

NLS をサポートしていた JDBC クラスのパッケージが、Oracle8i で変更されたことに注意してください。オブジェクト型およびコレクション型でサポートしている特定のキャラクタ・セットに関連するクラスは、基本 Zip ファイルから分割されました。これらの NLS クラスは拡張 Zip ファイルにパッケージされています。これによって、ユーザーは必要な場合にのみ NLS クラスを含めることができます。詳細は、「NLS 拡張 Zip ファイル (クライアント側のみ)」を参照してください。

\$ORACLE_HOME/lib ディレクトリには、`libocijdbc8.so` および `libocijdbc8_g.so` (Solaris の場合) が含まれています。これらは JDBC OCI ドライバが使用する共有ライブラリです。

\$ORACLE_HOME/jdbc/doc/javadoc.tar には、JDBC Javadoc が含まれています。今回のリリースでは、Oracle JDBC の公開クラスの公開 API に関する Javadoc ファイルのベータ・リリースが収められています。

\$ORACLE_HOME/jdbc/demo/demo.tar には、JDBC サンプル・プログラムが含まれています。JDK 1.1 用に記述されたデモ・プログラムは、JDK 1.2 で実行するために変更が必要です。

移植の詳細は「JDK 1.2 のサポート」を参照してください。

NLS 拡張 Zip ファイル（クライアント側のみ）

JDBC サーバー側内部ドライバは、NLS を完全にサポートしています。したがって、NLS 拡張 Zip ファイル（`nls_charset*.zip`）は不要です。この項の以降の説明は、JDBC サーバー側内部ドライバには適用されません。サーバー側内部ドライバのみを使用する場合は、この項をスキップしてもかまいません。

基本 Zip ファイル（`classes111.zip` および `classes12.zip`）には、NLS を完全にサポートするために必要なすべてのクラスが含まれています。

- Oracle8 のオブジェクト型またはコレクション型のデータ・メンバーとして取得または挿入されない CHAR/VARCHAR/LONGVARCHAR/CLOB 型のデータに対する Oracle キャラクタ・セット。
- オブジェクトおよびコレクションの CHAR/VARCHAR データ・メンバーで、ある程度共通して使用されているキャラクタ・セットに対する NLS サポート。このキャラクタ・セットは、US7ASCII、WE8DEC、WE8ISO8859P1 および UTF8 です。

オブジェクト/コレクションの CHAR/VARCHAR データ・メンバーにその他のキャラクタ・セットを利用する場合は、適切な拡張 Zip を環境変数 `CLASSPATH` に含める必要があります。膨大な数のキャラクタ・セットをサポートする必要性から、拡張 Zip ファイルのサイズが大きくなっていることに留意することが重要です。ユーザーは、拡張 Zip ファイルから必要なクラスのみを挿入するように選択できます。選択するには、最初に拡張 Zip ファイルを解凍してから、必要なクラスのみを環境変数 `CLASSPATH` に入れます。キャラクタ・セットの拡張クラス・ファイルの名前は、次の形式に従って指定されています。

```
CharacterConverter<Oracle キャラクタ・セット ID>.class
```

<Oracle キャラクタ・セット ID>には、対応するキャラクタ・セットの Oracle キャラクタ・セット ID の 16 進数表記が入ります。

インストールについて

環境変数 `CLASSPATH` には、複数のバージョンの Oracle JDBC ドライバを配置しないでください。JDBC ドライバは、Oracle Installer によって `$ORACLE_HOME/jdbc` ディレクトリにインストールされます。

環境変数の設定について

Windows95 / Windows98 / Windows NT の場合

- %ORACLE_HOME%\jdbc\lib\classes111.zip および
%ORACLE_HOME%\jdbc\lib\nls_charset11.zip を環境変数 CLASSPATH に追加します。
(JDK 1.2.x を使用する場合は、classes12.zip と nls_charset12.zip を追加します。)
- %ORACLE_HOME%\jdbc\lib を環境変数 PATH に追加します。

Solaris / Digital Unix の場合

- \$ORACLE_HOME/jdbc/lib/classes111.zip および
\$ORACLE_HOME/jdbc/lib/nls_charset11.zip を環境変数 CLASSPATH に追加します。
(JDK 1.2.x を使用する場合は、classes12.zip と nls_charset12.zip を追加します。)
- \$ORACLE_HOME/jdbc/lib を環境変数 LD_LIBRARY_PATH に追加します。

HP / UX の場合:

- \$ORACLE_HOME/jdbc/lib/classes111.zip および
\$ORACLE_HOME/jdbc/lib/nls_charset11.zip を環境変数 CLASSPATH に追加します。
(JDK 1.2.x を使用する場合は、classes12.zip と nls_charset12.zip を追加します。)
- \$ORACLE_HOME/jdbc/lib を環境変数 SHLIB_PATH と LD_LIBRARY_PATH に追加します。

AIX の場合:

- \$ORACLE_HOME/jdbc/lib/classes111.zip および
\$ORACLE_HOME/jdbc/lib/nls_charset11.zip を環境変数 CLASSPATH に追加します。
(JDK 1.2.x を使用する場合は、classes12.zip と nls_charset12.zip を追加します。)
- \$ORACLE_HOME/jdbc/lib を環境変数 LIBPATH と LD_LIBRARY_PATH に追加します。

JDBC ドライバを使用する上で役立つヒント

Oracle の JDBC ドライバの使用に関する詳細は、『Oracle8i JDBC 開発者ガイドおよびリファレンス』を参照してください。この項では、ヒントのみを提供します。これらのヒントはすべてを網羅するものではありません。

JDBC プログラムで実行したほうがよい簡単な事項を、次に列記します。

1. JDBC を使用するプログラムに、必要な JDBC クラスをインポートします。

例:

```
import java.sql.*;
import java.math.*;
```

2. 他の JDBC API をコールする前に、Oracle ドライバを登録します (JDBC サーバー側内部ドライバを使用している場合は不要です。登録はサーバーで自動的に行われます)。Oracle ドライバを登録する場合は、次の文が Java セッションで少なくとも 1 回実行されていることを確認します。

```
DriverManager.registerDriver(
    new oracle.jdbc.driver.OracleDriver());
```

3. `getConnection` コールでデータベースへの接続をオープンします。異なる JDBC ドライバには異なる接続 URL を使用してください。次の例では、異なる URL を使用しています。

《JDBC OCI8 ドライバの場合》

```
Connection conn = DriverManager.getConnection(
    "jdbc:oracle:oci8:@<データベース>",
    "scott", "tiger");
```

<データベース>には、`tnsnames.ora` または `SQL*net` 名-値ペアのエントリが入ります。

《JDBC Thin ドライバまたはサーバー側 Thin ドライバの場合》

```
Connection conn = DriverManager.getConnection(
    "jdbc:oracle:thin:@<データベース>",
    "scott", "tiger");
```

<データベース>には、<ホスト>:<ポート>:<SID>形式の文字列、または `SQL*net` 名-値ペアが入ります。

《JDBC サーバー側内部ドライバの場合》

```
Connection conn = DriverManager.getConnection(
    "jdbc:oracle:kprb:");
```

最後に「:」文字が必要であることを注意してください。サーバー側内部ドライバを使用する場合は、実行中のデータベースに常に接続します。次のようにして接続することもできます。

```
Connection conn
    = new oracle.jdbc.driver.OracleDriver().defaultConnection();
```

Java ストアド・プロシージャ

Oracle8 オブジェクト型を使用したコールとインスタンス・メソッドの例は、次のファイルで提供しています。

```
$ORACLE_HOME/javavm/demo/demo.zip
```

解凍すると、例は、次のディレクトリに格納されます。

```
$ORACLE_HOME/javavm/demo/examples/jsp
```

このリリースで判明している問題点および制限事項について

判明している問題点および制限事項を次に示します。

- LOB 型のストリーム入力の使用に関して制限があります。LOB 型のストリーム入力は、Oracle Server リリース 8.1.7 に接続しているリリース 8.1.7 JDBC OCI ドライバに対してのみ使用できます。これ以外の構成で LOB 型のストリーム入力を使用すると、データが破損する可能性があります。PreparedStatement ストリーム入力 API には、setBinaryStream()、setAsciiStream()、setUnicodeStream()、setCharacterStream() および setObject() が含まれています。
- Bug#1018797
CLOB 値の最後に誤って余分な文字が付加される場合があります。
 - CLOB 値の挿入に setCharacterStream() を使用する場合
 - Oracle Server がマルチバイト・キャラクタ・セットを使用する場合。
(LOB 型のストリーム入力の制限は、項目 1 を参照してください。)
- クライアント側ドライバを使用して 1 回に 16 接続以上のオープンを試みると、オープンに失敗する場合があります。これは、JDBC ドライバの制限が原因ではありません。1 プロセスあたりのファイル記述子の制限を超えたことが原因と思われます。解決方法は、この制限を緩和することです。

- サーバー側内部ドライバには、次の制限があります。
 - LONG および LONG RAW 型のデータ・アクセスは 32K までのデータに制限されます。
 - データベースの互換モードが 8.0 に設定されている場合、オブジェクト型 (Oracle8 オブジェクト、コレクションおよび参照) の挿入は機能しません。この制限は、互換モードが 8.1 に設定されている場合には適用されません。
 - Statement.cancel() は実装されていません。
 - SQLExceptions の連鎖内では、連鎖の先頭のみ `getSQLState` 値があります。
- Java 仮想マシンがグリーンスレッド・モードで実行されていると、SSL 接続の JDBC OCI ドライバは停止します。これを回避するには、Java 仮想マシンをネイティブ・スレッド・モードで実行します。
- 日付時刻フォーマット、通貨記号および小数点符号は、常に米国の表記規則で示されます。
- `OracleStatement.defineColumnType()` を使用する場合は、データベースで宣言されている列と同じ型になるように、列の型を定義する必要はありません。型が異なる場合、取り出した値は `defineColumnType` で指定した型に変換されます。

注意： 大半の変換は正しく行われていますが、完全ではありません。正しいと思われる変換が機能していないことを発見した場合は、オラクル社カスタマ・サポート・センターにお問い合わせください。

- JDBC トレース機能がオンに設定されている場合は、JSP の `System.out.println()` の標準出力へのリダイレクトに使用されているユーティリティ `dbms_java.set_output` または `dbms_java.set_stream` を使用しないでください。これは、`dbms_java.set_output` および `set_stream` の現在の実装では、標準出力の書き込みに JDBC を使用しているためです。そのため、JDBC トレース機能がオンの状態でこれらのユーティリティを使用すると、無限ループに陥ります。
- JDBC OCI および Thin ドライバは、CHAR データをバイナリ・ストリームを介して正しい読み込みができません。つまり、CHAR データを取得するために `getBinaryStream()` を使用すると不適切な結果になります。これを回避するには、`getBinaryStream()` のかわりに `getCHAR()` または `getAsciiStream()` を使用します。他の代替策として `getUnicodeStream()` の使用がありますが、お勧めしません。
- Bug#899078 (リリース 8.1.6 SDK 以降)

JDBC サーバー側内部ドライバには、PL/SQL CHAR OUT で埋め込まれた余分な空白があります (キャラクタ・セットによって異なりますが、2~3 つの空白)。この問題は、UTF8 を除くマルチバイト・データベース・キャラクタ・セットの大半で発生します。

- Java およびオブジェクト型で実装されるトリガーに制限があります。この制限は、クライアント側の Java を使用して実装されたトリガーの IN タイプの引数にのみ影響します。サーバー内部で実行されている JDBC プログラムには適用されません。Java メソッドとして実装されるトリガーには、Oracle8 オブジェクト型またはコレクション型の IN 引数はありません。つまり、トリガーを実装するために使用する Java メソッドには、次の型の引数は指定できません。
 - java.sql.Struct
 - java.sql.Array
 - oracle.sql.STRUCT
 - oracle.sql.ARRAY
 - oracle.jdbc2.Struct
 - oracle.jdbc2.Array
 - oracle.jdbc2.SQLData を実装するすべてのクラスまたは oracle.sql.CustomDatum
- スクロール可能な結果セットの実装には、次の制限があります。
 - ScrollableResultSet での setFetchDirection() はサポートされません。
 - ScrollableResult での SetrefreshRow() は、同期性と同時実行性のすべての組合せをサポートしているわけではありません。次の表はサポート対象の組合せです。

サポート	型	同時実行性
いいえ	TYPE_FORWARD_ONLY	CONCUR_READ_ONLY
いいえ	TYPE_FORWARD_ONLY	CONCUR_UPDATABLE
いいえ	TYPE_SCROLL_INSENSITIVE	CONCUR_READ_ONLY
はい	TYPE_SCROLL_INSENSITIVE	CONCUR_UPDATABLE
はい	TYPE_SCROLL_SENSITIVE	CONCUR_READ_ONLY
はい	TYPE_SCROLL_SENSITIVE	CONCUR_UPDATABLE

- Bug#1324918
 JDBC OCI、プリコンパイルされた SQL 文、および setCharacterStream を使用して CLOB を繰返し更新すると、一時表領域が次々に使用されます。CLOB を反復的に更新すると、一時表領域が大きくなり続けるか、またはスレッド main の例外 (java.sql.SQLException: ORA-01652: unable to extend segment if you have a limit on the tablespace size) を受け取る可能性があります。これを回避するには、oracle.sql.CLOB::setCharacterOutputStream() をかわりに使用します。

第 11 章 JPublisher

はじめに

この章では、Java Publisher ユーティリティ (JPub) リリース 8.1.7.0.0 に関する情報が収められています。

JPub は Java ソース・コードを生成します。このコードを使用すると、Oracle オブジェクト型と PL/SQL パッケージを、ユーザーの SQLJ と Java プログラムで簡単に表現できます。JPub は『Oracle 8i JPublisher ユーザーズ・ガイド』で説明されており、『Oracle8i SQLJ 開発者ガイドおよびリファレンス』に補足情報があります。この README ファイルの一部は、JPub に関連するマニュアルを読み終えていることを前提としています。

この章は、次の 4 つの項で構成されています。

- JPub のインストールおよび起動方法
- JPub に関して頻繁に寄せられる質問
- このリリースの新機能
- 既知の不具合と制限事項

JPub のインストールおよび起動方法

JPub は Java プログラムで、Oracle SQLJ に同梱されています。SQLJ がインストールされている場合は、JPub もインストールされています。

JPub を使用するために SQLJ が必要な理由は、次のとおりです。

- JPub は必要に応じて、.sqlj コードを生成します。このコードは SQLJ でコンパイルする必要があります。
- JPub 自体が SQLJ プログラムです。したがって、実行時には SQLJ が必要です。

JPub は、JDBC 経由でデータベースと通信します。JPub の実行には、JDBC ドライバのインストールも必要です。Oracle オブジェクトを使用している場合は、Oracle 8.1.x JDBC ドライバが必要です。

デフォルトでは、JPub は Oracle JDBC OCI ドライバを使用して Oracle データベースと通信しますが、-url コマンドライン・パラメータを設定して、Oracle Thin driver などの別のドライバも選択できます。

SQLJ と JDBC のインストールについては、次のドキュメントを参照してください。

- 『Oracle8i SQLJ 開発者ガイドおよびリファレンス』
- 『Oracle8i JDBC 開発者ガイドおよびリファレンス』
- SQLJ README ファイル
- JDBC README ファイル

JPub をインストールするには、前述の SQLJ マニュアルの SQLJ インストール手順に従ってください。その概要は、次のとおりです。

Oracle 製品をインストールしてある ORACLE_HOME を格納場所とします。

- 環境変数 CLASSPATH に、使用する環境に対応した Oracle JDBC ドライバを設定します。

JDK 1.1 の場合： \$ORACLE_HOME/jdbc/lib/classes11.zip

JDK 1.2 の場合： \$ORACLE_HOME/jdbc/lib/classes12.zip

Windows のプラットフォームに次のファイル名が書き込まれます。

JDK 1.1 の場合： %ORACLE_HOME%\jdbc\lib\classes11.zip

JDK 1.2 の場合： %ORACLE_HOME%\jdbc\lib\classes12.zip

- 環境変数 CLASSPATH に、\$ORACLE_HOME/sqlj/lib/translator.zip ファイルを設定します。

translator.zip ファイルには、SQLJ と JPub 両方の.class ファイルが含まれています。

- 環境変数 CLASSPATH に、\$ORACLE_HOME/sqlj/lib/runtime.zip ファイルを設定します。

runtime.zip ファイルには、SQLJ と JPub 両方の.class ファイルが含まれています。他の選択肢としては、JDK 1.1 と Oracle JDBC 8.1.7 を使用している場合に、\$ORACLE_HOME/sqlj/lib/runtime11.zip ファイルを指定できます。

また、JDK 1.2 以降と Oracle JDBC 8.1.7 を使用している場合は、\$ORACLE_HOME/sqlj/lib/runtime12.zip ファイルを指定できます。

- \$ORACLE_HOME/bin (または\$ORACLE_HOME\bin) ディレクトリでコマンドを実行できることを確認します。UNIX 環境では、\$ORACLE_HOME/bin ディレクトリを環境変数 PATH に設定するとコマンドを実行できます。\$ORACLE_HOME/bin ディレクトリには、JPub の起動を簡素化するためのラッパー・スクリプトまたは実行可能ファイルが含まれています。

JPublisher を起動するには、"jpub"の後ろに 0 (ゼロ) 以上のコマンドライン引数を指定します。引数をまったく指定しないと、JPub は使用可能なコマンドライン引数に関する情報を表示します。

JPub で生成されたクラスを使用する手順は、次のとおりです。

- 環境変数 CLASSPATH に、\$ORACLE_HOME/jdbc/lib/classes12.zip ファイル (JDK 1.2 の場合) または \$ORACLE_HOME/jdbc/lib/classes111.zip ファイル (JDK 1.1 の場合) を設定します。
- SQLJ コンパイラを使用して、JPub で生成された .sqlj ファイルをコンパイルします。
- SQLJ コンパイラまたは Java コンパイラを使用して、JPub で生成された .java ファイルをコンパイルします。
- 環境変数 CLASSPATH に、JPub で生成されたクラスの .class ファイルを含むパッケージのディレクトリを設定します。

JPub に関して頻繁に寄せられる質問

質問: 必要な Java クラスを、オブジェクト型やパッケージに手動で書き込むことができないのはなぜですか?

手動で書き込むことはできますが、エラーが発生しやすくなります。

JDBC プログラムで Oracle8 オブジェクト型を表現する Java クラスは、oracle.sql.CustomDatum インタフェースまたは java.sql.SqlData インタフェースを実装する必要があります。SQLJ プログラムで Oracle8 オブジェクト型を表現する Java クラスは、フォームのクラス内の宣言の中で、次のようにしてそのオブジェクト型に名前を指定する必要があります。

```
public static final String _SQL_NAME = "<オブジェクト型の名前>";
```

JPub で生成されたコードは、ユーザーに代わってこれらのタスクを処理します。さらに、JPub は、オブジェクト型と PL/SQL パッケージの両方に対して、サーバー・メソッドの起動時に、引数渡しとメソッド起動の詳細を処理するコードを生成します。

質問: JPub で生成されたクラスにメソッドをいくつか追加する予定です。JPub で生成されたクラスを編集するのですか、それとも新しいクラスを作成して、JPub で生成されたそのクラスを拡張するのですか?

JPub で生成されたクラスを編集する場合、および将来 JPub を再度実行してそのクラスを再生成する必要がある場合は、実行した変更内容を保存して、手動で再挿入する必要があります。したがって、JPub で生成されたクラスを拡張するほうがはるかに便利です。拡張時には、-input ファイル構文を使用して、次のように書込み対象のサブクラスを JPub に通知する必要があります。

```
TYPE <型名>  
AS <サブクラス名>  
GENERATE <JPub 生成クラスの名前>
```

この機能の詳細は、JPub のマニュアルを参照してください。

質問: JPub を使用して `java.sql.SqlData` インタフェースを実装するクラスを生成できますか?

はい。JPub、Oracle JDBC ドライバおよび Oracle SQLJ は、`oracle.sql.CustomDatum` インタフェースまたは `java.sql.SqlData` インタフェースの選択をサポートしています。

質問: コマンドライン上でオプションを再指定すると、`-props` ファイルのオプションをオーバーライドできますか?

はい。コマンドライン上で `-props` オプションの後にそのオプションを指定します。オプションは、コマンドライン上に表示される順序で左から右に処理されます。`-props` ファイルのオプションは、コマンドライン上に表示された時点で処理されます。

質問: "`jpub -user=scott/tiger`" と入力し、JPub に対して変換対象のオブジェクト型もパッケージも指定しなかったにもかかわらず、JPub からの要求がありませんでした。なぜですか?

`-input`、`-sql` および `-types` などのオプションがいずれも存在しない場合は、ユーザー・スキーマのすべての型とパッケージが変換されます。

質問: JPub `-encoding` オプションを指定する必要があるのはどのような場合ですか?

一般的なルールとして、Java コンパイラの起動時に `-encoding` オプションを指定しない場合は、JPublisher の起動時にも `-encoding` オプションを指定する必要はありません。`-encoding` オプションが指定されていない場合は、システム・プロパティの `file.encoding` の値が使用されます。

`encoding` オプションは、JPub で生成された入力ファイル、`.sqlj` ファイルおよび `.java` ファイルに使用する文字コード化を指定します。`-encoding` の設定は、コマンドラインまたは `-props` ファイルの読み込み方法に影響を与えません。

質問: JPub のオブジェクト型から Java への変換方法に満足しています。しかし、C プログラムでもオブジェクト型を使用する予定です。これを達成するにはどのようにしたらよいですか?

オブジェクト型を C に変換するには、別の Oracle ユーティリティ、Object Type Translator (OTT) を使用します。

JPub の新機能

JPub リリース 8.1.6.0.0 とリリース 8.1.7.0.0 の間で変更された機能を次に示します。

- 次のフィールドは、SQL オブジェクト型に対する JPub 生成コードで保護されるようになりました。これにより、JPub 生成クラスของผู้ユーザー書き込み用サブクラスに機能が追加されます。
 - `_ctx` - SQLJ 接続コンテキスト・オブジェクト
 - `_struct` - 元の SQL 形式のデータを持つ可変的な構造体のオブジェクト
- Java キーワード"NULL"は、属性またはメソッドをマッピングするための Java 名として使用されると、次のように重要な意味を持ちます。
 - SQL メソッドが NULL にマップされると、メソッドはまったく生成されません。
 - SQL 属性が NULL にマップされると、取得/設定メソッドはまったく生成されません。

JPub に関する既知の不具合と制限事項

現在の JPub は、次の PL/SQL 引数型をサポートしていません。

BOOLEAN

NCHAR

NCLOB

NVARCHAR2

レコード・タイプ

索引付き表

厳密な型指定の REF CURSOR

これらの型を引数型または結果タイプとして使用するサーバー・メソッド用のラッパー・メソッドの生成を JPub に要求すると、JPub はサポートされていない型を使用したというエラー・メッセージを表示し、.sql ファイルにそのラッパー・メソッドに対するパーシャル・コードを含むコメントを生成します。

JPub は、`-input` ファイル内の大部分の（すべてではない）エラーをユーザーにレポートします。`-input` ファイルのエラーが少数の場合、レポートは作成されません。このエラーの例は、次のとおりです。

- ユーザーが 2 つの異なるオブジェクト型に対して同じ Java クラス名を要求した場合、2 番目のクラスが最初のクラスを暗黙的に上書きします。たとえば、`-input` ファイルに次のものが含まれているとします。

```
type PERSON1 as person
TYPE PERSON2 as person
```

この場合、`person.java` ファイルが `type PERSON1` に対して作成され、次にそのファイルは、`TYPE PERSON2` によって上書きされます。

- ユーザーが 2 つの異なるオブジェクト属性に対して同じ属性名を要求すると、両属性に対して取得と設定メソッドが生成されます。この場合、警告メッセージは発行されません。Java は、生成クラスが正当であるかどうかを、同じ名前を持つ 2 つの取得メソッドと、同じ名前を持つ 2 つの設定メソッドが、オーバーロードの可能性のある異なる引数型を持っているかどうかによって判断します。
- ユーザーが `TRANSLATE` 句で、存在しないオブジェクト属性を指定した場合、その属性は無視され、警告メッセージは発行されません。

たとえば、`-input` ファイルに次の句が含まれているとします。

```
type PERSON translate X as attr1
```

`X` が `PERSON` の属性でないとしてします。この場合、警告メッセージは発行されません。

第 12 章 *interMedia Text*

今回のリリースの *interMedia Text* の新機能に関しては、『Oracle8i リリース 8.1.7 追加情報』の第 2 章に記載されています。ここでは、『Oracle8i リリース 8.1.7 追加情報』には含まれない *interMedia Text* の新しい機能および拡張機能が記載されています。

新しい索引タイプ CTXCAT

この項では、新しい索引タイプ CTXCAT の使用の制限について説明します。この索引タイプに関連するシステム・パラメータについても説明します。

CTXCAT の使用方法の詳細は、『Oracle8i リリース 8.1.7 追加情報』の第 2 章を参照してください。

パラメータ文字列の制限

CTXCAT 索引を作成するには、CREATE INDEX を使用して、ConText 索引を作成するときと同じようにオプションでパラメータ文字列でプリファレンスを指定します。

ただし、ctxcat パラメータ文字列で使用できるキーワードは次のように制限されています。

INDEX SET

LEXER

MEMORY

STOPLIST

STORAGE

WORDLIST

列型をサポートする CTXCAT

索引セットは、索引を 99 まで取ることができます。それぞれの索引は、実表から列の順序付けられたリストから作成されます。ただし、これらの列には、次のような 2 つの重要な制限事項があります。

- 使用可能な型は、最大長 30 バイトの NUMBER、DATE および CHAR/VARCHAR2 型のみです。
- NULL 値は、使用できません。ユーザー列の NULL は索引エラーとなり、行が索引付けられません。

CATSEARCH structured_query パラメータの制限

CATSEARCH 演算子の structured_query パラメータに対して保持される制限事項を次に示します。

- 左側（列名）は、索引セットの索引内の 1 つ以上の列名である必要があります。
- 左側は、プレーン列名である必要があります。 -- 関数、式などがない
- 演算子は、<、<=、=、>=、>、BETWEEN および IN に制限されます。
- 右側は、リテラル値で構成されている必要があります。 -- 関数、式、他の列、副選択などがない
- 複数の条件は、AND で結合できます。OR は、サポートされていません。

サポートされている文を次に示します。

```
catsearch(text, 'dog', 'foo > 15')
catsearch(text, 'dog', 'bar = 'SMITH'')
catsearch(text, 'dog', 'foo between 1 and 15')
catsearch(text, 'dog', 'foo = 1 and abc = 123')
```

サポートされていない文を次に示します。

```
catsearch(text, 'dog', 'upper(bar) = 'A'')
catsearch(text, 'dog', 'bar LIKE 'A%'')
catsearch(text, 'dog', 'foo = abc')
catsearch(text, 'dog', 'foo = 1 or abc = 3')
```

CTXCAT システム・パラメータ

カタログ索引は、それ自体がプリファレンスのデフォルトを設定するシステム・パラメータのセットを持っています。これを使用して、インストールがデフォルトのプリファレンスのセットを 1 つ、ctxcat 索引のセットをもう 1 つ持つことができます。次に、新しいシステム・パラメータを示します。

DEFAULT_CTXCAT_LEXER

DEFAULT_CTXCAT_STOPLIST

DEFAULT_CTXCAT_WORDLIST

DEFAULT_CTXCAT_STORAGE

DEFAULT_CTXCAT_INDEX_SET

ctxcat 索引のワードリストのプリファレンスを指定できますが、CATSEARCH 問合せ言語がワイルド・カード、ファジーおよびステミングをサポートしていないので、適用されない属性が多くあります。日本語用データの場合、使用可能なワードリスト・プリファレンスの属性は、PREFIX_INDEX のみです。

フィルタの改善

PL/SQL プロシージャ・フィルタの新しい機能

前回のリリースでは、ユーザー・フィルタ・オブジェクトは、それぞれのドキュメントに 1 度起動される外部実行可能ファイルを使用します。これは、それぞれの起動にシェルのオーバーヘッドおよび起動コストで負荷がかかるので、遅くなります。

PL/SQL プロシージャ・フィルタは、ユーザー・データストアのように、外部実行可能ファイル・インタフェースを PL/SQL インタフェースと置き換えます。

Oracle のシャドウ・プロセスへの索引付けの統合があるので、セーフ・コールアウトのユーザー・フィルタがサポートされています。さらに、PL/SQL インタフェースを使用すると、様々な環境で使用できる PL/SQL または Java スタッド・プロシージャのフィルタリングが使用できます。

プロシージャ・フィルタを作成する場合、新しい PROCEDURE 属性にプロシージャ名を指定する必要があります。例を次に示します。

```
ctx_ddl.create_preference('myfilt','procedure_filter');
ctx_ddl.set_attribute('myfilt','procedure','normalize');
```

プロシージャ・フィルタは、ユーザー・データストアと同じセキュリティ制約を必要とします。プロシージャは、CTXSYS が所有する必要があります。索引を作成するプリファレンスを使用するユーザーはすべて、スタッド・プロシージャに EXECUTE 権限を付与されている必要があります。

デフォルトでは、フィルタ・プロシージャには 2 つのパラメータがあります。最初のパラメータは BLOB で、データストアが提供する入力するバイナリのバージョンです。2 番目のパラメータは CLOB で、フィルタが提供する出力するテキストのバージョンです。前述の例から次のようなプロシージャが予想されます。

```
PROCEDURE NORMALIZE(IN BLOB, IN OUT NOCOPY CLOB)
```

入力および出力タイプは、INPUT_TYPE および OUTPUT_TYPE 属性を使用して、変更できます。例を次に示します。

```
ctx_ddl.set_attribute('myfilt','input_type','clob');
ctx_ddl.set_attribute('myfilt','output_type','varchar2');
```

フィルタ・プロシージャに(IN CLOB, IN OUT NOCOPY VARCHAR2)の署名を指定します。

BLOB、CLOB および VARCHAR2 に加えて、入力および出力両タイプは、FILE に設定されます。入力が FILE に設定される場合、入力ドキュメントは、ファイルおよび最初に引数として受け渡されるファイル名にスプールされます。フィルタ・プロシージャは、フィルタされた内容がそのファイルに置かれます。プロシージャ・パラメータの型は、VARCHAR2 で

す。この機能は、OS ファイルの扱いが LOB にアクセスするより単純なので、フィルタ・プロシージャが実際にセーフ・コールアウトの場合に役に立ちます。

ファイル・プロシージャは、ROWID、キャラクタ・セットまたは列フォーマットが渡される必要があります。例を次に示します。

```
ctx_ddl.set_attribute('myfilt','ROWID_PARAMETER','TRUE')
```

最初のパラメータで Oracle に ROWID を渡し、入力、出力が続いています。

ROWID_PARAMETER、CHARSET_PARAMETER および FORMAT_PARAMETER は、すべて独立しています。順序は、ROWID、フォーマット、それからキャラクタ・セットです。ただし、フィルタ・プロシージャは、最小のパラメータを渡されます。例を次に示します。

```
ctx_ddl.create_preference('myfilt','procedure_filter');
ctx_ddl.set_attribute('myfilt','procedure','normalize');
ctx_ddl.set_attribute('myfilt','input_type','clob');
ctx_ddl.set_attribute('myfilt','output_type','varchar2');
ctx_ddl.set_attribute('myfilt','rowid_parameter','TRUE')
ctx_ddl.set_attribute('myfilt','charset_parameter','TRUE')
```

CTXSYS.NORMALIZE が署名を持ちます。

```
(IN ROWID, IN VARCHAR2, IN CLOB, IN OUT NOCOPY VARCHAR2)
```

最初のパラメータは、ドキュメントの ROWID です。2 番目はドキュメントのキャラクタ・セット列の値で、3 番目は入力するドキュメントの内容で、4 番目は出力するドキュメントの内容です。

INSO フィルタがフィルタリングの外部実行可能メソッドを使用しているので注意してください。ただし、これは、将来のリリースではセーフ・コールアウトに変更する可能性があります。

ワードリストの改善

プリフィックス索引付けの新機能

リリース 8.1.6 のワイルド・カード問合せは、入力パターンを LIKE で \$I の一致トークンに拡張し、これらの一致の同等化に問合せを実行して処理されます（同等化に詳しくない場合は、OR とみなしてください）。問合せの例を次に示します。

```
contains(text, 'wood%')
```

問合せが通過し、次のようにパターンを拡張します。

```
select distinct token_text
from dr$index$I
where token_text like 'WOOD%'
```

wood、wooden、woody および woods を戻すようにするには、次のように問合せを書き直します。

```
contains(text, 'wood = wooden = woody = woods')
```

これには、次の3つの理由による問題があります。

1. 一致する用語が多い場合、または用語が\$Iに多くの行を持つ場合、拡張ステップに長時間かかる可能性があります。
2. パターンが多くのワードと一致する可能性があります。また、「問合せが複雑すぎます」というエラーが表示され問合せが失敗します。
3. 多くの個別トークンがある場合、すべての情報の読み込みおよびマージに長時間かかります。

そのため、より速いワイルド・カード検索が必要なアプリケーションに対して、リリース 8.1.7 にプリフィックス索引付けが追加されました。次のようにプリフィックス索引付けを使用します。

```
ctx_ddl.create_preference('mywordlist', 'BASIC_WORDLIST');  
ctx_ddl.set_attribute('mywordlist', 'PREFIX_INDEX', 'YES');
```

それぞれのトークンは、索引作成時に複数の接頭辞に分割され、それぞれの接頭辞は、トークン・タイプ 6 で索引付けされます。たとえば、「TOKEN」という単語があるドキュメントの場合、次の\$I 行のような結果が出力されます。

TOKEN_TEXT	TOKEN_TYPE	TOKEN_INFO
TOKEN	0	DOC 1 OFFSET 1
T	6	DOC 1 OFFSET 1
TO	6	DOC 1 OFFSET 1
TOK	6	DOC 1 OFFSET 1
TOKE	6	DOC 1 OFFSET 1
TOKEN	6	DOC 1 OFFSET 1

次のように接頭辞ワイルド・カード問合せが実行されます。

```
contains(text, 'TOK%')
```

interMedia Text は、後続のワイルド・カードを削除し、トークン "TOK" タイプ 6 の情報を参照します。LIKE 拡張または複数トークンのマージは必要ありません。その結果、接頭辞問合せがすばやく実行され、拡張数の制限がなくなります。そのために、TOKEN がタイプ 6 で索引付けされる必要があります。これを使用すると、TOKEN%での検索で TOKEN を検索できます。

問合せパフォーマンスの改善の価値は、索引サイズおよび時間を増加させます。これは、オプションで最小長および最大長の設定を使用して制御できます。例を次に示します。

```
ctx_ddl.set_attribute('mywordlist', 'PREFIX_MIN_LENGTH', 3);  
ctx_ddl.set_attribute('mywordlist', 'PREFIX_MAX_LENGTH', 4);
```

3～4 文字の接頭辞のみが索引付けされます。トークン TOKEN の場合、次のような結果になります。

TOKEN_TEXT	TOKEN_TYPE	TOKEN_INFO
-----	-----	-----
TOKEN	0	DOC 1 OFFSET 1
TOK	6	DOC 1 OFFSET 1
TOKE	6	DOC 1 OFFSET 1

最小長より短い、または最大長より長いパターンのワイルド・カード検索は、古いスタイルの解決方法を実行します（たとえば、LIKE を使用してパターンを拡張したり、同等化を使用して結果をマージしたりします）。

問合せが接頭辞問合せに変換され、開始するにはかなり短い V-GRAM トークンの日本語環境では、接頭辞索引作成は、特に役に立ちます。

プリフィックス索引付けは、ctxcat 索引にサポートされています。ただし、catsearch には、ワイルド・カード演算子がありません。そのため、通常、それを設定しても、役に立ちません。日本語の VGRAM レクサーの接頭辞索引作成で索引付けされた ctxcat がある場合は、問合せ処理に必要になります。

プリフィックス索引付けおよびサブ文字列索引付け（リリース 8.1.6 で提供）は、個別にあり、独立しています。プリフィックス索引付けは、右側にワイルド・カードを使用した問合せ（たとえば、ABC%）のみを高速化します。サブ文字列索引付けは、左側および両側にワイルド・カードを使用した問合せ（たとえば、%ABC、%ABC%）を高速化します。プリフィックス索引付けは、拡張および同等化のマージを排除して高速化を実現しています。サブ文字列索引付けは、拡張する \$I の全表スキャンを排除して高速化を実現しています。サブ文字列索引付けには、まだ拡張リストで同等化のマージをする必要があります。

新しいプロシージャ CTX_THES.HAS_RELATION

プロシージャ CTX_THES.HAS_RELATION を使用して、実際に拡張せずにシソーラス関係が存在することをテストします。

```
CTX_THES.HAS_RELATION(phrase in varchar2, rel in varchar2,  
tname in varchar2 default 'DEFAULT')  
returns boolean;
```

phrase : 句を指定します。

rel : PT を除く、単一シソーラス関係または関係のカンマで区切られたリストを指定します。すべての関係に'ANY'を指定します。

tname : シソーラス名を指定します。

句がリスト内の関係のいずれかを持つ場合、関数は TRUE を返します。

例を次に示します。

```
ctx_thes.has_relation('cat','BT,BTG')
```

DEFAULT の句"cat"がいずれかの広義の用語または広義の汎用用語がある場合、TRUE を返します。

新しいプロシージャ CTX_DDL.REMOVE_INDEX

『Oracle8i リリース 8.1.7 追加情報』に記述されているこのプロシージャの指定は、不完全です。このプロシージャの正しい指定を次に示します。

```
CTX_DDL.REMOVE_INDEX(set_name in varchar2, column_list in varchar2);
```

このプロシージャは、CTXCAT 索引プリファレンスから指定された列リストを使用して索引を削除します。

set_name : 索引セット名を指定します。

column_list : 索引セット・プリファレンスから削除する column_list を指定します。

オプションの FILE/URL セキュリティ

FILE および URL データストアは、データベース・マシンのファイルに接続します。すべてのユーザーがこれらのデータストアを使用して Oracle ユーザーにアクセスできるファイル・システムをブラウズできるので、依存性のあるサイトには望ましくありません。リリース 8.1.7 では、このようなサイトには FILE_ACCESS_POLE システム・パラメータを提供しています。ユーザーまたは DBA は、このシステム・パラメータをデータベース・ロールの名前に設定できます。設定した場合、FILE または URL データストアを使用して索引を作成しようとするユーザーはすべて、このロールを持つ必要があります。そうしないと、索引作成が失敗します。たとえば、DBA が次のように設定します。

```
ctx_adm.set_parameter('FILE_ACCESS_ROLE','WHITEHAT');
```

ユーザー SCOTT が次のようにします。

```
create index foox on foo(text) indextype is ctxsys.context  
parameters('datastore ctxsys.file_datastore')
```

エンジンは、SCOTT がロール WHITEHAT を持つかどうかをテストします。SCOTT がロール WHITEHAT を持つ場合、CREAT INDEX は通常どおり処理されます。持たない場合は、CREAT INDEX は失敗します。

これは索引作成時にのみチェックされることを注意してください。このパラメータを設定したり、または名前付きロール権限を付与/取り消したりしても、ファイル・データストアを使用した索引付けには影響ありません。

URL データストアのユーザー名/パスワードのサポート

URL データストアの FTP プロトコル URL は、次に示すサイトで埋込みユーザー名およびパスワードをサポートしています。

```
ftp://username:password@ftp.hostname.com/dir/file.doc
```

第 13 章 *interMedia* Image、Audio、Video、Text および Locator

重要： このオプションを使用するには、Oracle リスナーを実行している必要があります。詳細は、「リスナーを起動します」の項を参照してください。

警告： Oracle *interMedia* はデータベース・ユーザーORDSYSの下にインストールされます。ORDSYSはインストール中に作成されるもので、次回ベータ版または製品版では変更される可能性があります。このパッケージにより提供される機能、パッケージおよびタイプとその外部インタフェースはOracleが権利を留保しており、次期リリースで予告なしに変更することがあります。このパッケージをお客様が修正することはできません。DBMSで内部エラーおよびセキュリティ違反を引き起こす可能性があります。

スタート・ガイド

Oracle *interMedia* の一般的なリリース情報および Oracle *interMedia* Image、Audio および Video コンポーネントの固有のリリース情報を説明しています。

Oracle *interMedia* リリース 8.1.7

Oracle *interMedia* を使用すると、Oracle8i は他のエンタープライズ情報を持つ統合方式で image、audio、video、text および locator 関数を管理できます。Oracle *interMedia* は、Oracle8i の信頼性、可用性、およびデータ管理をインターネット、電子商取引およびメディアリッチ・アプリケーションのマルチメディア・コンテンツへ拡張します。*interMedia* の機能には、Oracle8i が管理するマルチメディア・データの格納、検索、管理および操作が含まれます。

Oracle *interMedia* Image は、2 つのディメンションの静的ビットマップ・イメージの格納、検索および処理を提供します。画像は、一般的な業界標準のデスクトップ・パブリッシングの画像交換フォーマットを使用して圧縮され、効果的に格納されます。

Oracle *interMedia* Audio および *interMedia* Video は、各種の音声やビデオ・ファイル形式を格納および検索することができます。

アップグレードの注意点

interMedia リリース 8.0.x、リリース 8.1.x をリリース 8.1.7 へアップグレードする際のアップグレード用のスクリプトが用意されています。これらのスクリプトでは *interMedia* Image、Audio および Video のみアップグレードできます。*interMedia* Text および Locator は、このスクリプトではアップグレードできません。*interMedia* Locator のアップグレードについては『Oracle Spatial ユーザーズ・ガイドおよびリファレンス』を、*interMedia* Text のアップグレードについては、『Oracle8i *interMedia* Text リファレンス』を参照してください。

注意：*interMedia* をアップグレードする前に、データベースのアップグレードを行ってください。データベース（Oracle JVM を含む）のアップグレードの方法については、『Oracle8i 移行ガイド』を参照してください。

データベースのアップグレード後、次の手順に従って *interMedia* のアップグレードを行ってください。

1. Oracle8 Image、Audio または Video がすでにインストールされていて、Oracle *interMedia* リリース 8.1.7 をインストールしたい場合は、imdbma を実行し、アップグレードが必要かどうか判断してください。

Server Manager を起動します。（UNIX および Windows NT の場合は、svrmgrl）

```
% svrmgrl
```

SYSDBA として SYS ユーザーで接続します。

```
SVRMGR> connect SYS/<SYS_PASSWORD> as SYSDBA
```

```
SVRMGR> @<ORACLE_HOME>/ord/im/admin/imdbma.sql (UNIX の場合)
```

```
SVRMGR> @<ORACLE_HOME>%ord¥im¥admin¥imdbma.sql (Windows NT の場合)
```

<ORACLE_HOME>は ORACLE_HOME ディレクトリです

次のいずれかのメッセージが表示されます。

NOT_INSTALLED システムには以前のバージョンの *interMedia* コンポーネントがインストールされていません。アップグレードではなく、インストールを実行してください。

INSTALLED *interMedia* リリース 8.1.7 がすでにインストールされています。

MANUAL アップグレードが必要です。

アップグレードが必要な場合は、手順 2 に進んでください。

2. imuplhelp を実行し、*interMedia* のアップグレードの準備ができているかどうか判断します。

Server Manager を起動します。（UNIX および Windows NT の場合は、svrmgrl）

```
% svrmgrl
```

SYSDBA として SYS ユーザーで接続します。

```
SVRMGR> connect SYS/<SYS_PASSWORD> as SYSDBA
```

```
SVRMGR> @<ORACLE_HOME>/ord/im/admin/imuphelp.sql (UNIX の場合)
```

```
SVRMGR> @<ORACLE_HOME>%ord%im%admin%imuphelp.sql (Windows NT の場合)
```

<ORACLE_HOME>は ORACLE_HOME ディレクトリです

次のメッセージが表示されます。

- アップグレードを行う前に Audio の古いバージョンを削除してください。
- アップグレードを行う前に Video の古いバージョンを削除してください。

続行する前に imuphelp.sql で見つかった問題を修正してください。

3. アップグレードが必要で、準備ができている場合はアップグレードを行います。

- a. ORD のアップグレードを行います。\$ORACLE_HOME/ord/admin/u08xxx0.sql を実行します。

Server Manager を起動します。

```
% svrmgrl
```

SYSDBA として SYS/<SYS_PASSWORD>に接続します。

```
SVRMGR> @<ORACLE_HOME>/ord/admin/u080xxx0.sql (UNIX の場合)
```

```
SVRMGR> @<ORACLE_HOME>%ord%admin%u080xxx0.sql (Windows NT の場合)
```

xxxには現在インストールされている *interMedia* または Image Cartridge のバージョンが入ります。たとえば、Oracle8 Image リリース 8.0.3.0.0 からアップグレードを行う場合は、u0800030.sql になります。

注意： ORDSYS のパスワードの入力が必要です。

- b. *interMedia* のアップグレードを行います。

<ORACLE_HOME>/ord/im/admin/u08xxx0.sql を実行します。

Server Manager を起動します。

```
% svrmgrl
SVRMGRL> @<ORACLE_HOME>/ord/im/admin/u080xxx0.sql (UNIX の場合)
SVRMGRL> @<ORACLE_HOME>%ord¥im¥admin¥u080xxx0.sql (Windows NT の場合)
```

xxx には現在インストールされている *interMedia* または Image Cartridge のバージョンが入ります。たとえば、Oracle8 Image リリース 8.0.3.0.0 からアップグレードを行う場合は、u0800030.sql になります。

注意： ORDPLUGINS および ORDSYS のパスワードの入力が必要です。

4. アップグレードの検証のため、inchk.sql を実行します。

```
% svrmgrl
```

ORDSYS として接続します。

```
SVRMGRL> @<ORACLE_HOME>/ord/im/admin/imchk.sql (UNIX の場合)
SVRMGRL> @<ORACLE_HOME>%ord¥im¥admin¥imchk.sql (Windows NT の場合)
```

Oracle *interMedia* のインストール

Oracle *interMedia* をインストールする前に、Oracle8i をインストールしてください。Oracle8i のインストールと同時に Oracle *interMedia* をインストールした場合は、この項で説明する手順を行う必要はありません。Oracle *interMedia* は自動的に構成されます。

- インストール決定
- インストール前の作業
- インストール

インストール決定

インストール中に ORDSYS、ORDPLUGINS および MDSYS のユーザーが作成されます。これらのユーザーID はオプションとして特別な権限を持つ標準的な Oracle のデータベース・アカウントです。

決定： ORDERSYS、ORDERPLUGINS および MDSYS ユーザーのパスワードを決定します。ORDSYS および ORDERPLUGINS にはデフォルトのパスワードが使用されます。インストール終了後にパスワードを変更してください。

自動インストールでの ORDSYS ユーザーのデフォルトのパスワードは「ORDSYS」、ORDPLUGINS ユーザーのデフォルト・パスワードは「ORDPLUGINS」です。

インストール前の作業

Oracle *interMedia* のインストールを行う前に、次の作業を行ってください。手順については、ご使用の OS に対応したインストレーション・ガイドを参照してください。

- PL/SQL および Oracle JVM (JServer) オプションを含む Oracle 8i のインストール
- データベースの作成
- データベースの起動

インストール

指示に従ってインストールを行ってください。この作業は、Oracle *interMedia* を手動でインストールする場合のみ行ってください。Database Configuration Assistant を使用する場合は、この作業は必要ありません。また、次の手順で *interMedia* Image、Audio、Video および Locator をインストールします。

この手順では *interMedia Text* のインストールは行いません。

1. Oracle Universal Installer を使用して、Oracle *interMedia* のデータベースではないコンポーネントをインストールします。
2. ユーザーを作成し権限を付与します。

Server Manager を起動します。（UNIX および Windows NT では svrmgrl）

```
% svrmgrl
```

SYSDBA として SYS ユーザーで接続します。

```
SVRMGR> connect SYS/<SYS_PASSWORD> as SYSDBA.
```

<SYS_PASSWORD>には SYS のパスワードが入ります。

ordinst.sql を実行します。

```
SVRMGR> @<ORACLE_HOME>/ord/admin/ordinst.sql (UNIX の場合)
```

```
SVRMGR> @<ORACLE_HOME>%ord%admin%ordinst.sql (Windows NT の場合)
```

<ORACLE_HOME>は ORACLE_HOME ディレクトリです。

3. Oracle *interMedia* を作成します。

```
SVRMGR> @<ORACLE_HOME>/ord/im/admin/imininst.sql (UNIX の場合)
```

```
SVRMGR> @<ORACLE_HOME>%ord%im%admin%imininst.sql (Windows NT の場合)
```

<ORACLE_HOME>は ORACLE_HOME ディレクトリです。

4. リスナーを起動します。

リスナーは外部プロシージャ・コールを使用するように構成してください。tnsnames.ora ファイルでエントリが extproc_connection_data をコールしているか、listner.ora ファイルでエントリが extproc. をコールしているか確認してください。詳細は、ネットワーク・ドキュメントを参照してください。正しく設定されていない場合、Oracle *interMedia* は動作しません。設定終了後に、Oracle *interMedia* は使用可能になります。

デモの実行

Oracle *interMedia* インストール後に、Oracle *interMedia* デモを実行するかどうかを選択することができます。この項では、*interMedia* デモの参照先を記述しています。

Oracle *interMedia* Image デモ

interMedia Image OCI デモについては、README.txt を参照してください。

UNIX の場合： \$ORACLE_HOME/ord/img/demo/README.txt

Windows NT の場合： %ORACLE_HOME%\ord\img\demo\README.txt

interMedia Image Java デモについては、README.txt を参照してください。

UNIX の場合： \$ORACLE_HOME/ord/img/demo/java/README.txt

Windows NT の場合： %ORACLE_HOME%\ord\img\demo\java\README.txt

Oracle *interMedia* Audio デモ

interMedia Audio SQL デモについては、README.txt を参照してください。

UNIX の場合： \$ORACLE_HOME/ord/aud/demo/README.txt

Windows NT の場合： %ORACLE_HOME%\ord\aud\demo\README.txt

interMedia Audio Java デモについては、README.txt を参照してください。

UNIX の場合： \$ORACLE_HOME/ord/aud/demo/java/README.txt

Windows NT の場合： %ORACLE_HOME%\ord\aud\demo\java\README.txt

Oracle *interMedia* Video デモ

interMedia Video SQL デモについては、README.txt を参照してください。

UNIX の場合： \$ORACLE_HOME/ord/vid/demo/README.txt

Windows NT の場合： %ORACLE_HOME%\ord\vid\demo\README.txt

interMedia Video Java デモについては、README.txt を参照してください。

UNIX の場合： \$ORACLE_HOME/ord/vid/demo/java/README.txt

Windows NT の場合： %ORACLE_HOME%\ord\vid\demo\java\README.txt

Oracle *interMedia* Locator デモ

interMedia Locator については、次のファイルを参照してください。

UNIX の場合： \$ORACLE_HOME/MD/doc/README.doc
Windows NT の場合： %ORACLE_HOME%\MD\doc\README.doc

ダウングレードの注意点

Oracle *interMedia* のダウングレードの準備を行います。次の手順に従ってください。

1. ディレクトリを変更します。

UNIX の場合： \$ORACLE_HOME/ord/im/admin
Windows NT の場合： %ORACLE_HOME%\ord\im\admin

2. Server Manager を起動します。（UNIX および Window NT の場合は、svrmgrl）

3. @d080xxx0.sql

xxx にはダウングレードを行う *interMedia* または Image Cartridge のリリースが入ります。
たとえば、Oracle8 Image のリリース 8.0.3.0.0 へダウングレードする場合は、d0800030.sql
となります。

注意： ORDPLUGINS および ORDSYS のパスワードの入力が必要です。

削除の注意点

次の手順に従って、Oracle *interMedia* をアクティブでない状態にします。

1. ディレクトリを変更します。

UNIX の場合： \$ORACLE_HOME/ord/im/admin
Windows NT の場合： %ORACLE_HOME%\ord\im\admin

2. Server Manager を起動します。（UNIX および Windows NT の場合は、svrmgrl）

3. @imdinst.sql

注意： MDSYS、ORDPLUGINS および ORDSYS のパスワードの入力が必要です。

第 12 章 *interMedia* タイプの削除

interMedia を削除した後に、*interMedia* オブジェクトのすべての表とタイプを削除します。Server Manager で次のスクリプトを実行すると、*interMedia* Image、Audio および Video のタイプを削除できます。

UNIX の場合： @<ORACLE_HOME>/ord/im/admin/imdtyp.sql

Windows NT の場合： @<ORACLE_HOME>%ord%im%admin%imdtyp.sql

第 13 章 *interMedia* Java クラスの使用方法

リリース 8.1.7 には新しい *interMedia* Java クラスが追加されています。詳細は、「このリリースで追加された機能」を参照してください。

リリース 8.1.7 *interMedia* Java クラスを使用するには、現在の Java 環境のセットアップと修正を行い、Java プログラムをコンパイルおよび実行します。さらに、次のステップを行ってください。

環境変数 CLASSPATH には jdk クラス、*interMedia* Java ライブラリ、JDBC および SQLJ ランタイム・ライブラリが必要です。ライブラリは次の場所にあります。

注意：jdk クラスの場所はバージョンおよびインストールに依存するため、ここでは説明しません

Unix:

```
$ORACLE_HOME/jdbc/lib/classes111.zip or classes12.zip  
$ORACLE_HOME/ord/jlib/ordim817.zip  
$ORACLE_HOME/sqlj/lib/runtime.zip
```

Windows NT:

```
%ORACLE_HOME%\jdbc\lib\classes111.zip or classes12.zip  
%ORACLE_HOME%\ord\jlib\ordim817.zip  
%ORACLE_HOME%\sqlj\lib\runtime.zip
```

JDBC OCI 共有ライブラリの場所は環境変数 LD_LIBRARY_PATH (UNIX の場合) または環境変数 PATH (Windows NT の場合) で定義されている必要があります。SQL*Plus などの他のクライアント・アプリケーションを使用するために、同じ共有ライブラリがすでに定義されていることもあります。

Unix:

```
<ORACLE_HOME>/lib (for libocijdbc8.so)
```

Windows NT:

```
<ORACLE_HOME>%bin (for ocijdbc8.dll)
```

新しい *interMedia* Java クラスを使用しない場合は、8.1.5 API を使用することもできます。リリース 8.1.5 の *interMedia* Java クラスを使用するには、環境変数 CLASSPATH に ordim817.zip ではなく、ordim.zip (リリース 8.1.5 ライブラリ) があることを確認してください。

次期リリースとの互換性のための重要な情報

重要： *interMedia* の次期リリースでは、新しいオブジェクト属性を追加することにより *interMedia* のオブジェクト型 (ORDAudio、ORDImage、ORDVideo および ORDSource) を改良する可能性があります。サーバーがオブジェクト型を改良し、アップグレードした後もクライアント側アプリケーションが、*interMedia* オブジェクト型のバージョンである 8.1.7 との互換性を維持したい場合は、次の作業を行うことをお勧めします。

- 必要に応じて、アプリケーション起動時に互換性初期化ファンクションをコールしてください。(次の項を参照)
- *interMedia* オブジェクト型が新しい属性を追加することで改良された場合、挿入された文がデフォルトのコンストラクタを使用します。そのため、デフォルトのコンストラクタは正しく動作しない可能性があるので、使用しないでください。かわりに、8.1.7 で提供されている新しい静的コンストラクタ・ファンクションを使用できます。

詳細は、「このリリースで追加された機能」の項を参照してください。

互換性初期化ファンクションを使用する場合とその使用方法

注意： クライアント側アプリケーションが *interMedia* オブジェクト型の構造体を静的に認識している場合にのみ、互換性初期化ファンクションをコールしてください。サーバー側ストアド・プロシージャはアップグレード後に、新しくインストールされた (改良された可能性がある) *interMedia* オブジェクト型を自動的に使用するので、互換性初期化ファンクションをコールする必要はありません。(コンパイル時に) *interMedia* オブジェクト型の構造体を静的に認識しないクライアント側アプリケーションの場合は、互換性初期化ファンクションをコールしないでください。実行時に OICDescribeAny コールを経由して *interMedia* オブジェクト型の構造体を決定する OCI アプリケーションの場合は互換性初期化ファンクションをコールする必要はありません。

(OTT が生成する) *interMedia* オブジェクト型を C でコンパイルした OCI で書かれているクライアント側アプリケーションの場合は、アプリケーションのはじめにサーバー側 PL/SQL ファンクションである ORDSYS.IM.compatibilityInit をコールしてください。

```
FUNCTION compatibilityInit(release IN    VARCHAR2,  
                           errmsg OUT   VARCHAR2)  
  return NUMBER;
```

この PL/SQL ファンクションはアプリケーションで使用している Oracle *interMedia* のバージョンをあらわす文字列型のパラメータを使用します。*interMedia* オブジェクト型を改良した Oracle の次期リリースで、リリース 8.1.7 アプリケーションを使用する（アップグレードせずに）ためには、この文字列を「8.1.7」に設定します。2 つめのパラメータは文字列出力パラメータです。このファンクションが 0 以外の値を返す場合、エラー・メッセージに問題の原因が含まれています。

リリース 8.1.7 用の *interMedia* Java クラスを使用して Java で書かれているクライアント側アプリケーションの場合は、Oracle に接続後に `OrdMediaUtil.imCompatibilityInit` 関数をコールしてください。

```
public static void imCompatibilityInit(OracleConnection con)
    throws Exception
```

この Java 関数は、引数に `OracleConnection` を渡します。挿入された *interMedia* リリース 8.1.7 Java API により、改良されたオブジェクト型の *interMedia* の次期リリースを、現在ご使用のリリース 8.1.7 アプリケーションで使用できます。

クライアント側 PL/SQL アプリケーションでは、次期リリースでオブジェクトが改良され、新しい属性が追加された場合、*interMedia* オブジェクトのリリース 8.1.7 との互換性を維持する方法はありません。

詳細説明および実行例については、『Oracle8i *interMedia* Audio,Image および Video ユーザーズ・ガイドおよびリファレンス』および『Oracle8i *interMedia* Audio,Image および Video Java Classes ユーザーズ・ガイドおよびリファレンス』を参照してください。

リリース 8.1.7 で修正された問題

Oracle *interMedia* リリース 8.1.7 で修正された問題

- srcType 'FILE'で動作する export()メソッド。export()セマンティクスは次のとおりです。
 - export()はファイルをコピーする操作に似ています。つまり、BLOB に格納されているオリジナルのデータは読取り以外の目的でアクセスされません。
 - export()は import()とまったく逆の操作をするものではありません。clearLocal()は export()によって自動的にコールされません。一方、import()は setLocal()をコールします。
 - export()メソッドを使用した後に、データベースでマルチメディア・データを使用しないが、参照を継続したい場合（たとえば BFILE として）は、deleteContent()をコールして BLOB 記憶域に開放してください。

データベース・サーバーへの書き込み能力は強力です。この能力は悪用される可能性があります。重要なデータベース・ファイルを上書きしようと試みたり、サーバーのディスク容量を限界にするために、多数の大量のファイルを書き込む可能性があります。

このため、export()には特殊な権限が設定されています。

- export()ではユーザーがアクセスする権限を持っているディレクトリ・オブジェクトにのみ書き込みを行うことができます。つまり、ユーザー自身が SQL の「CREATE DIRECTORY」コマンドで作成したディレクトリ以外に対しては、読取り権限が与えられています。

CREATE DIRECTORY コマンドを実行するには、CREATE ANY DIRECTORY 権限が必要です。

- より制限するために、DBMS_JAVA.GRANT_PERMISSION を使用して、ファイルに書き込みを許可するファイルを指定します。

例：

```
CALL DBMS_JAVA.GRANT_PERMISSION(  
    'MEDIAUSER',  
    'java.io.FilePermission',  
    '/actual/server/directory/path/filename.dat',  
    'write');
```

DBMS_JAVA.GRANT_PERMISSION については、『Oracle8i Java 開発者ガイド』の「セキュリティおよびパフォーマンス」を参照してください。

- ORDSource.import メソッドがさらに定義されています。

```
MEMBER PROCEDURE import (
    SELF      IN OUT NOCOPY ORDSource,
    ctx       IN OUT RAW,
    mimetype  OUT VARCHAR2,
    format    OUT VARCHAR2)
```

このインポート・メソッドは既存のインポート・メソッドとほぼ同一のものです。宛先 BLOB は個別パラメータとして渡されません。冗長な宛先 BLOB パラメータは次期リリースで廃止されます。

Oracle *interMedia* Image リリース 8.1.7 で修正された問題

deleteContent()メソッドはほとんどの ORDImage のメタデータ属性を無意識に NULL に設定するようになっていました。deleteConent()がメタデータ属性にアクセスすることはありません。

Oracle *interMedia* Audio リリース 8.1.7 で修正された問題

なし

Oracle *interMedia* Video リリース 8.1.7 で修正された問題

なし

Oracle *interMedia* Locator リリース 8.1.7 で修正された問題

interMedia Locator の情報については、次のファイルを参照してください。

UNIX の場合：	\$ORACLE_HOME/<MD>/doc/README.doc
Windows NT の場合：	%ORACLE_HOME%\<MD>\doc\README.doc

リリース 8.1.7 で追加された機能

リリース 8.1.7 *interMedia* Image、Audio および Video の新機能

- ORDIImage、ORDAudio、ORDVideo の各メディア・タイプに static メソッドが追加されたことにより、インスタンスの初期化が簡略化されました。このメソッドは「init」と呼ばれ、2つのバリエーションのうちのどちらかを使用できます。

1 つめの「init」にはパラメータがないため、すべてのメディア・タイプの属性を NULL にしますが、次のような例外があります。

- source updateTime は SYSDATE に設定されます。
- source local は 1 に設定されます。
- BLOB および CLOB はそれぞれ empty_blob および empty_clob に設定されます。

2 つめの「init」には 3 つの文字列型のパラメータがあり、source.srcType、source.srcLocation および source.srcName に直接マップされます。他のすべての属性は NULL に設定されますが、次のような例外があります。

- source updateTime は SYSDATE に設定されます。
- source.local は 0 に設定されます。
- BLOB および CLOB はそれぞれ empty_blob および empty_clob に設定されます。

重要：「init」メソッドの使用をできるだけ早くはじめることをお勧めします。次期バージョンで、*interMedia* オブジェクト型は改良される可能性があります。
interMedia オブジェクト型に属性を追加する場合、挿入された文がデフォルトのコンストラクタ（各オブジェクト属性を初期化する）を使用するため、デフォルトのコンストラクタは正しく動作しない可能性があります。

使用例：SQL

```
INSERT INTO AUDIOTABLE VALUES ( ORDSYS.ORDAudio.init(), ... );
INSERT INTO IMAGETABLE VALUES (
    ORDSYS.ORDImage.init('FILE','IMGDIR','product3154.jpg'), ... );
PL/SQL:
DECLARE
    myImage ORDSYS.ORDImage;
    myVideo ORDSYS.ORDVideo;
BEGIN
    myImage := ORDSYS.ORDImage.init;
    myVideo := ORDSYS.ORDVideo.init('FILE','VIDDIR','interMedia.rm');
    ...
    INSERT INTO MEDIATABLE VALUES ( myImage, myVideo, .... );
END;
```

2. リッチ・コンテンツ用の新しい *interMedia* Java クラス

Oracle *interMedia* Java クラスによりどの階層（クライアント、アプリケーション・サーバーまたはデータベース・サーバー）の Java アプリケーションでも Oracle8i に格納されている音声、画像および映像データを操作および修正することができます。

アプリケーションはデータベースに接続し、メディア・オブジェクトを選択し、データベースからオブジェクトをコピーするなどのさまざまな操作を行い、Oracle8i に新しいメディア・オブジェクトを格納できます。

interMedia Java クラスは JDBC のリッチ・コンテンツへのアクセスをサポートします。*interMedia* 列と他のリレーショナル・データの集合を含む結果セット上でアプリケーションを選択し、操作することができます。Java クラスはまた、オブジェクト属性にアクセスし、オブジェクト・メソッドの起動を可能にします。

新しい *interMedia* Java クラスは Oracle8i サーバー（リリース 8.1.5 またはそれ以降）と Oracle8i クライアント（リリース 8.1.6 またはそれ以降）とともに動作します。

JDBC をサポートするために、*interMedia* Java クラスの構文とセマンティクスが変更されました。

オブジェクトのアプリケーションのコピーをデータベース・オブジェクトに明示的にバインドする必要はなくなりました。また、アプリケーションのコピーを、対応するデータベース・オブジェクトと同期化するためのリフレッシュやフラッシュは不要になりました。かわりに、ユーザーは標準 JDBC `executeQuery()` ファンクション・コールを使用して Java プログラムで *interMedia* オブジェクトを取り出してください。データベースにアプリケーションのコピーをフラッシュするかわりに、JDBC API を使用して標準 SQL アップデート文を実行することで明示的にアップデートが行われます。

その他の基本的な変更はすべてメソッドに関するもので、外部データ・ソースから BLOB ヘデータをロードする場合を除いて、オブジェクトのアプリケーションのコピーの状態を変更します。ユーザーはデータベース・オブジェクトに変更を伝えるために上記の明示的アップデートを行ってください。

さらに、新しい Java クラスで廃止された API や、小規模の変更が行われた API もあります。

***interMedia Image* リリース 8.1.7 の新機能**

interMedia Image は、デジタル・カメラの「Exif」フォーマットを認識できます。Exifフォーマットは本来は JFIF フォーマットのバリエーションです。ORDImage.setProperties メソッドは fileFormat 属性に JFIF を設定します。

***interMedia Audio* および *Video* リリース 8.1.7 の新機能**

なし

***interMedia Locator* リリース 8.1.7 の新機能**

interMedia Locator の情報については、次のファイルを参照してください。

UNIX の場合 : \$ORACLE_HOME/<MD>/doc/README.doc

Windows NT の場合 : %ORACLE_HOME%\<MD>\doc\README.doc

このリリースの確認されている問題

Oracle *interMedia Image* リリース 8.1.7 で確認されている問題

段階的にエンコードした JPEG ファイルを使用する「process」メソッドは、失敗します。

Oracle *interMedia Audio* リリース 8.1.7 で確認されている問題

なし

Oracle *interMedia Video* リリース 8.1.7 で確認されている問題

なし

Oracle *interMedia Locator* リリース 8.1.7 で確認されている問題

interMedia Locator については、次のファイルを参照してください。

UNIX の場合 : \$ORACLE_HOME/<MD>/doc/README.doc

Windows NT の場合 : %ORACLE_HOME%\<MD>\doc\README.doc

廃止予定の旧式のオブジェクト型

ORDImgB および ORDImgF はリリース 8.1.7 以降では出荷されません。次期バージョンでは、新しい image オブジェクト型である `ORDImage` が使用されます。

廃止されたメソッド

ctx パラメータを許可する `ORDAudio` および `ORDVideo` get メソッドはリリース 8.1.6 で廃止されました。

`ORDAudio`:

```
getFormat(ctx IN OUT RAW) RETURN VARCHAR2
getEncoding(ctx IN OUT RAW) RETURN VARCHAR2
getNumberOfChannels(ctx IN OUT RAW) RETURN INTEGER
getSamplingRate(ctx IN OUT RAW) RETURN INTEGER
getSampleSize(ctx IN OUT RAW) RETURN INTEGER
getCompressionType(ctx IN OUT RAW) RETURN VARCHAR2
getAudioDuration(ctx IN OUT RAW) RETURN INTEGER
```

`ORDVideo`:

```
getFormat(ctx IN OUT RAW) RETURN VARCHAR2
getFrameSize(SELF IN OUT NOCOPY ORDVideo,
             ctx IN OUT RAW,
             retWidth OUT INTEGER,
             retHeight OUT INTEGER)
getFrameResolution(ctx IN OUT RAW) RETURN INTEGER
getFrameRate(ctx IN OUT RAW) RETURN INTEGER
getVideoDuration(ctx IN OUT RAW) RETURN INTEGER
getNumberOfFrames(ctx IN OUT RAW) RETURN INTEGER
getCompressionType(ctx IN OUT RAW) RETURN VARCHAR2
getNumberOfColors(ctx IN OUT RAW) RETURN INTEGER
getBitRate(ctx IN OUT RAW) RETURN INTEGER
```

これらのメソッドはリリース 8.1.7 では出荷されますが改良は行われません。次期バージョンでは削除されます。

上記のメソッドはそれぞれ必要とする属性を抽出するためにメディア・データを読み込みます。それぞれの属性に対してメディア・データを読み込むのではなく、`setProperties` メソッドをコールすることをお勧めします。このメソッドはメディアから属性を抽出し、オブジェクト属性を形成します。`getxxx` メソッド (ctx パラメータなし) はオブジェクト属性に格納された値を返します。これらはリリース 8.1.7 およびそれ以降のバージョンで推奨されるメソッドです。