

Oracle COM Automation 機能

開発者ガイド

リリース 9.2

2002 年 7 月

部品番号 : J06331-01

Oracle COM Automation 機能 開発者ガイド, リリース 9.2

部品番号 : J06331-01

原本名 : Oracle COM Automation Feature Developer's Guide, Release 9.2

原本部品番号 : A95499-01

原本協力者 : Janis Greenberg, Eric Belden, Steven Caminez, Jagadish Changavi, Barmak Meftah

Copyright © 1999, 2002, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

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

当プログラムのリバース・エンジニアリング等は禁止されています。

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

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

危険な用途への使用について

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

Restricted Rights Notice

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

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

目次

はじめに	v
対象読者	vi
このマニュアルの構成	vi
関連資料	vii
表記規則	viii
 Oracle COM Automation の新機能	xiii
Oracle9i リリース 2 (9.2) の Oracle COM Automation における新機能	xiv
Oracle9i リリース 1 (9.0.1) の Oracle COM Automation における新機能	xiv
Oracle8i リリース 8.1.6 の Oracle COM Automation における新機能	xv
 1 Oracle COM Automation の概要	
Oracle COM Automation の概要	1-2
Oracle COM Automation の機能	1-2
PL/SQL 用の Oracle COM Automation	1-2
Java 用の Oracle COM Automation	1-3
Oracle COM Automation の利点	1-3
Oracle COM Automation のアーキテクチャ	1-4
PL/SQL のアーキテクチャ	1-5
COM Automation の外部プロシージャ API のコール	1-6
可用性の問題に関するアーキテクチャ上の影響	1-6
Java アーキテクチャ	1-6
信頼性	1-7

2 Oracle COM Automation のインストールおよび構成

Oracle COM Automation のコンポーネント	2-2
システム要件	2-3
Oracle8i から Oracle9i への移行	2-3
Oracle COM Automation の構成	2-4
PL/SQL 用の Oracle COM Automation の構成	2-4
Java 用の Oracle COM Automation の構成	2-4
PL/SQL のリスナーの構成	2-5
リスナーの問題のトラブルシューティング	2-6
DCOM のサポート	2-7
データベース・インスタンスを実行しているコンピュータの構成	2-8
ドメイン・ユーザーへのサービスの設定	2-8
リモート・オブジェクトを含むコンピュータの構成	2-8

3 Oracle COM Automation の基本機能

データ型変換	3-2
PL/SQL のデータ型変換	3-2
Java のデータ型変換	3-3
HRESULT エラー・コード	3-4
Java 用の Oracle COM Automation の例外処理	3-4
COM Automation の一般的な機能	3-6
COM オブジェクトに関して必要な情報	3-6
OLE-COM オブジェクト ビューア	3-7
COM Automation の API の使用	3-8
API	3-8
PL/SQL API	3-9
CreateObject	3-10
DestroyObject	3-11
GetLastError	3-12
GetProperty	3-13
SetProperty	3-14
InitArg	3-16
InitOutArg	3-16
GetArg	3-17
SetArg	3-18
Invoke	3-20

Java API	3-21
Automation コンストラクタ	3-23
Create	3-24
Destroy	3-25
GetProperty	3-26
SetProperty	3-27
InitArg	3-28
SetArg	3-29
Invoke	3-30
Currency コンストラクタ	3-32
Get	3-33
Set	3-33

4 Oracle COM Automation PL/SQL デモ

PL/SQL デモの概要	4-2
Microsoft Word デモ	4-2
Microsoft Word デモのインストール	4-3
Microsoft Word デモの使用法	4-3
基本機能	4-4
Microsoft Excel デモ	4-7
Microsoft Excel デモのインストール	4-7
Microsoft Excel デモの使用法	4-8
基本機能	4-8
Microsoft PowerPoint デモ	4-11
Microsoft PowerPoint デモのインストール	4-11
Microsoft PowerPoint デモの使用法	4-12
基本機能	4-12
MAPI デモ	4-15
MAPI デモを使用する環境の設定	4-15
MAPI デモのインストールの準備	4-16
MAPI デモのインストール	4-17
MAPI デモの使用法	4-17
基本機能	4-18

5 Oracle COM Automation Java デモ

Oracle COM Automation Java デモの概要	5-2
Microsoft Word Java デモ	5-2
Microsoft Word Java デモのインストール	5-2
Microsoft Word Java デモの使用方法	5-3
カスタム・アプリケーションの作成	5-3
基本機能	5-4

A COM Automation エラー・メッセージ

Oracle COM Automation PL/SQL エラー	A-2
Microsoft COM Automation のエラー	A-4

用語集

索引

はじめに

このマニュアルでは、Oracle COM Automation の概念、インストール、インストール後の構成および使用方法について説明します。

このマニュアルで説明するのは、Windows NT、Windows 2000 および Windows XP オペレーティング・システムに適用される Oracle9i for Windows ソフトウェアの機能のみです。

次の項目について説明します。

- [対象読者](#)
- [このマニュアルの構成](#)
- [関連資料](#)
- [表記規則](#)

対象読者

『Oracle COM Automation 機能 開発者ガイド』は、COM を使用するソリューションを開発する開発者を対象としています。

このマニュアルは、次のことを十分に理解していることを前提にしています。

- Component Object Model (COM)
- Object Linking and Embedding (OLE) オートメーション
- Structured Query Language (SQL)
- データ定義言語 (DDL)
- データ操作言語 (DML)
- PL/SQL または Java
- Oracle のオブジェクト・リレーショナル・データベース管理 (ORDBMS) の概念
- Windows NT

このマニュアルの構成

このマニュアルは、次のように構成されています。

第 1 章「Oracle COM Automation の概要」

Oracle COM Automation および Oracle サーバー・アーキテクチャの概要を説明します。Oracle COM Automation をインストールして使用する前に、この章をお読みください。

第 2 章「Oracle COM Automation のインストールおよび構成」

Oracle COM Automation のインストール方法、および Oracle COM Automation を使用する前に必ず実行しておく構成タスクを説明します。Oracle COM Automation の Software Development Kit (SDK) の内容のリストやシステム要件についても説明します。

第 3 章「Oracle COM Automation の基本機能」

COM オブジェクトの操作に必要な PL/SQL および Java Application Program Interface (API) を含め、Oracle COM Automation の基本機能を説明します。

第 4 章「Oracle COM Automation PL/SQL デモ」

Oracle COM Automation の PL/SQL デモの使用方法を説明します。

第 5 章「Oracle COM Automation Java デモ」

Oracle COM Automation の Java デモの使用方法を説明します。

付録 A 「COM Automation エラー・メッセージ」

Oracle COM Automation PL/SQL のエラー・コードおよび Microsoft COM Automation のエラー・コードを説明します。

用語集

このマニュアルで使用される用語を定義します。

関連資料

詳細は、次の Oracle マニュアルを参照してください。

- 『Oracle9i Database for Windows インストレーション・ガイド』
- 『Oracle9i Database for Windows リリース・ノート』
- 『Oracle9i Database for Windows 管理者ガイド』
- 『Oracle Enterprise Manager 管理者ガイド』
- 『Oracle Services for Microsoft Transaction Server 開発者ガイド』
- 『Oracle9i Net Services 管理者ガイド』
- 『Oracle9i Real Application Clusters 概要』
- 『Oracle9i データベース新機能』
- 『Oracle9i データベース概要』
- 『Oracle9i データベース・リファレンス』
- 『Oracle9i データベース・エラー・メッセージ』
- 『Oracle9i Java Developer's Guide』
- 『Oracle9i Java Stored Procedures Developer's Guide』

このマニュアルの多くの例では、Oracle のインストール時にデフォルトでインストールされるシード・データベースのサンプル・スキーマを使用します。これらのスキーマの作成方法およびその使用方法は、『Oracle9i サンプル・スキーマ』を参照してください。

リリース・ノート、インストール・ドキュメント、ホワイト・ペーパー、またはその他の関連資料を無償でダウンロードするには、OTN-J (Oracle Technology Network Japan) にアクセスしてください。OTN-J を利用する前に、オンライン登録が必要です。次の URL で登録できます。

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

OTN-J のユーザー名およびパスワードをすでにお持ちの場合は、次の OTN-J の Web サイトのドキュメント・セクションに直接アクセスできます。

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

表記規則

ここでは、このマニュアルの本文およびサンプル・コードで使用される表記規則について説明します。表記規則は次の 3 種類です。

- 本文の表記規則
- サンプル・コードの表記規則
- Windows オペレーティング・システムの表記規則

本文の表記規則

本文中では、特定の用語をより簡単に識別できるように、様々な表記規則を使用しています。次の表は、本文中で使用される表記規則とその使用例を説明したものです。

規則	意味	例
太字	太字は、本文中で定義されている用語、または用語集で説明されている用語、あるいはその両方を示します。	この句を指定する場合、 索引構成表 を作成します。
大文字（固定幅） フォント	大文字固定幅フォントは、システムによって指定される要素を示します。これらの要素には、パラメータ、権限、データ型、Recovery Manager のキーワード、SQL のキーワード、SQL*Plus またはユーティリティのコマンド、パッケージ、メソッドの他に、システムで表示される列名、データベースのオブジェクトおよび構造、ユーザー名およびロールがあります。	この句は NUMBER 列に対してのみ指定できます。 BACKUP コマンドを使用して、データベースをバックアップできます。 USER_TABLES データ・ディクショナリ・ビューの TABLE_NAME 列を問い合わせます。 DBMS_STATS.GENERATE_STATS プロシージャを使用します。
小文字（固定幅） フォント	小文字固定幅フォントは、実行可能ファイル、ファイル名、ディレクトリ名、およびサンプルのユーザー指定要素を示します。これらの要素には、コンピュータ名およびデータベース名、ネット・サービス名、および接続識別子の他に、ユーザー指定のデータベースのオブジェクトおよび構造、列名、パッケージおよびクラス、ユーザー名およびロール、プログラム・ユニット、およびパラメータ値があります。 注意： 一部のプログラム要素には、大文字と小文字の両方が使用されます。これらの要素は、記載されているとおりに入力してください。	sqlplus を入力して、SQL*Plus を開きます。 パスワードは、orapwd ファイルで指定されます。 ¥disk1¥oracle¥dbs ディレクトリのデータ・ファイルと制御ファイルをバックアップします。 department_id、department_name および location_id 列は、hr.departments 表にあります。 QUERY_REWRITE_ENABLED 初期化パラメータを true に設定します。 oe ユーザーとして接続します。 JRepUtil クラスは、これらのメソッドを実装します。

規則	意味	例
小文字イタリック (固定幅) フォント	小文字イタリック固定幅フォントは、ブレースホルダまたは変数を示します。	<i>parallel_clause</i> を指定できます。 <i>Uold_release</i> .SQL を実行します。 <i>old_release</i> は、アップグレード前にインストールしたリリースを表します。

サンプル・コードの表記規則

サンプル・コードは、SQL、PL/SQL、SQL*Plus またはその他のコマンドライン文を示します。これらは固定幅フォントで示され、次の例のように、通常の本文とは区別されています。

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

次の表は、サンプル・コードで使用する表記規則とそれらの使用例を説明したものです。

規則	意味	例
[]	大カッコは、1 つ以上のオプション項目を囲みます。大カッコは入力しないでください。	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	中カッコは複数の項目を囲み、そのうちの 1 つが必要であることを示します。中カッコは入力しないでください。	{ENABLE DISABLE}
	縦線は、大カッコまたは中カッコ内にある複数のオプションの選択肢を区切るために使用します。オプションの 1 つを入力します。縦線は入力しないでください。	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	水平の省略記号は、次のいずれかを示します。 <ul style="list-style-type: none">■ 例に直接関係のないコードの一部を省略■ コードの一部の繰り返しが可能	CREATE TABLE ... AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
. . . .	垂直の省略記号は、例に直接関係のないコードの数行を省略したことを示します。	SQL> SELECT NAME FROM V\$DATAFILE; NAME ----- /fsl/dbs/tbs_01.dbf /fsl/dbs/tbs_02.dbf . . . /fsl/dbs/tbs_09.dbf 9 rows selected.

規則	意味	例
その他の表記規則	大カッコ、中カッコ、縦線および省略記号以外の記号は、示されているとおりに入力してください。	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
イタリック	イタリックの文字は、特定の値を指定する必要があるプレースホルダまたは変数を示します。	CONNECT SYSTEM/system_password DB_NAME = database_name
大文字	大文字は、システムによって指定される要素を示します。ユーザーが定義する語句と区別するために、大文字で示しています。語句が大カッコ内に表示されている場合を除き、記載されているとおりの順序とスペルで入力します。ただし、これらの語句には大文字と小文字の区別がないため、小文字で入力できます。	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;
小文字	小文字は、ユーザーが指定するプログラム要素を示します。たとえば、小文字は表、列またはファイルの名前を示します。 注意： 一部のプログラム要素には、大文字と小文字の両方が使用されます。これらの要素は、記載されているとおりに入力してください。	SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjjones IDENTIFIED BY ty3MU9;

Windows オペレーティング・システムの表記規則

次の表は、Windows オペレーティング・システムの表記規則とその使用例を説明したものです。

規則	意味	例
「スタート」→を選択	プログラムの起動方法。たとえば、Database Configuration Assistant を起動するには、タスクバーの「スタート」ボタンをクリックし、「プログラム」→「Oracle - HOME_NAME」→「Configuration and Migration Tools」→「Database Configuration Assistant」を選択します。	「スタート」→「プログラム」→「Oracle - HOME_NAME」→「Configuration and Migration Tools」→「Database Configuration Assistant」を選択します。

規則	意味	例
ファイル名 および ディレクトリ名	ファイルおよびディレクトリ名には、大文字と小文字の区別がありません。＜、＞、：、"、/、 、および-の特殊文字は使用できません。特殊文字¥は、引用符に囲まれている場合でも、要素の区切り文字として扱われます。ファイル名が¥¥で始まる場合、Windows では汎用命名規則を使用しているものと認識されます。	c:¥winnt"¥"system32 は、 C:¥WINNT¥SYSTEM32 と同じです。
C:¥>	現行のハード・ディスク・ドライブの Windows コマンド・プロンプトを示します。コマンド・プロンプトのエスケープ文字は、カレット (^) です。プロンプトは、現在作業中のサブディレクトリを示しています。このマニュアルでは、コマンド・プロンプトと呼びます。	C:¥oracle¥oradata>
特殊文字	特殊文字の円記号 (¥) は、Windows コマンド・プロンプトで特殊文字の二重引用符 (") のエスケープ文字として必要な場合があります。カッコおよび特殊文字の一重引用符 (') は、エスケープ文字を必要としません。エスケープ文字および特殊文字の詳細は、Windows オペレーティング・システムのドキュメントを参照してください。	C:¥>exp scott/tiger TABLES=emp QUERY=¥"WHERE job='SALESMAN' and sal<1600¥" C:¥>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)
HOME_NAME	Oracle ホーム名を示します。 ホーム名は、英数字 16 文字までです。ホーム名で利用できる特殊文字は、アンダースコアのみです。	C:¥> net start OracleHOME_NAME_TNSListener

規則	意味	例
ORACLE_HOME および ORACLE_BASE	<p>Oracle8 リリース 8.0 以下のリリースでは、Oracle コンポーネントをインストールすると、サブディレクトリはすべて、最上位の ORACLE_HOME ディレクトリ（デフォルトでは次のとおり）の下に置かれました。</p> <ul style="list-style-type: none">■ Windows NT の場合は C:¥orant■ Windows 98 の場合は C:¥orawin98 <p>あるいは、Oracle ホームと呼ばれるディレクトリの下に置かれました。</p> <p>今回のリリースは、Optimal Flexible Architecture (OFA) に準拠しています。すべてのサブディレクトリが最上位の ORACLE_HOME ディレクトリの下にあるわけではありません。ORACLE_BASE という最上位ディレクトリがあり、デフォルトは C:¥oracle です。コンピュータに最新の Oracle リリースをインストールし、他の Oracle ソフトウェアをインストールしない場合、最初の Oracle ホーム・ディレクトリのデフォルト設定は、C:¥oracle¥orann です。nn は、最新のリリース番号です。Oracle ホーム・ディレクトリは、ORACLE_BASE の直下に置かれます。</p> <p>このマニュアルでは、ディレクトリ・パスの例は、すべて OFA 表記規則に準拠しています。</p>	<p>%ORACLE_HOME%¥rdbms¥admin ディレクトリに移動します。</p>

Oracle COM Automation の新機能

次の項では、Oracle COM Automation の新機能について説明します。

- [Oracle9i リリース 2 \(9.2\) の Oracle COM Automation における新機能](#)
- [Oracle9i リリース 1 \(9.0.1\) の Oracle COM Automation における新機能](#)
- [Oracle8i リリース 8.1.6 の Oracle COM Automation における新機能](#)

Oracle9i リリース 2 (9.2) の Oracle COM Automation における新機能

次の項目について説明します。

- **Java 用の権限の付与**

新しい SQL スクリプトの `grant.sql` は、Java 用の Oracle COM Automation のユーザーごとに実行する必要があります。このスクリプトにより必要な権限が付与されます。詳細は、2-4 ページの「[Java 用の Oracle COM Automation の構成](#)」を参照してください。

- 多くの新機能が、Oracle9i リリース 1 (9.0.1) で追加されました。リリース 1 (9.0.1) より前のリリースからアップグレードする場合、「[Oracle9i リリース 1 \(9.0.1\) の Oracle COM Automation における新機能](#)」の新機能情報をお読みください。

Oracle9i リリース 1 (9.0.1) の Oracle COM Automation における新機能

次の項目について説明します。

- **Windows 2000 での Oracle9i の使用方法**

Windows 2000 と Windows NT 4.0 では、Oracle9i の使用方法に若干の違いがあります。

関連資料：『Oracle9i Database for Windows スタート・ガイド』

- **サンプル・スキーマ**

このマニュアルの多くの例では、Oracle のインストール時にデフォルトでインストールされるシード・データベースのサンプル・スキーマを使用します。これらのスキーマの作成方法およびその使用法は、『Oracle9i サンプル・スキーマ』を参照してください。

- **Java ストアド・プロシージャ (JSP) をサポートする新機能**

Oracle COM Automation は、Java および PL/SQL で使用できるようになりました。一般的な機能は同じですが、このマニュアルでは、機能、設定およびアーキテクチャの異なる箇所を説明します。

- **Oracle8i から Oracle9i への移行**

`com81.dll` の名前が `orawpcom.dll` に変更されました。Oracle8i から移行する場合、PL/SQL 用の Oracle COM Automation の使用を継続するには `comwrap.sql` を再実行する必要があります。

- **Microsoft Word Java デモ**

Java 用の Oracle COM Automation では、Microsoft Word のコード・サンプルをロードおよび実行する方法を示すデモを使用できます。

- **PL/SQL 用の Oracle COM Automation**

MAPI デモの使用に役立つ追加情報が示されています。

関連項目： 4-15 ページの「[MAPI デモを使用する環境の設定](#)」および 4-16 ページの「[MAPI デモのインストールの準備](#)」

- **PL/SQL OUT パラメータに対するサポート**

SetPtrArg は、IN パラメータおよび OUT パラメータを使用して同じ機能を提供する実装で置き換えられています。この実装は、SetArg 機能と SetArg データ型文字列の変更、および [InitOutArg](#) と [GetArg](#) の導入で構成されています。

OUT パラメータを使用するには、3-18 ページの「[SetArg](#)」の SetArg データ型文字列を参照してください。

Oracle8i リリース 8.1.6 の Oracle COM Automation における新機能

Oracle8i リリース 8.1.6 には、次の機能が含まれていました。

- **2 つの新しいデモ・プログラム**

- [Microsoft PowerPoint デモ](#)： Oracle から PowerPoint にデータを挿入します。
- [MAPI デモ](#)： Oracle から Messaging Application Programming Interface (MAPI) 準拠アプリケーションにデータを挿入します。

関連項目： デモ・プログラムの使用方法は、[第 4 章「Oracle COM Automation PL/SQL デモ」](#)を参照してください。

Oracle COM Automation の概要

この章では、Oracle COM Automation の SDK についての説明と、製品の概要を示します。
Oracle COM Automation をインストールして使用する前に、この章をお読みください。

この章の項目は次のとおりです。

- [Oracle COM Automation の概要](#)
- [Oracle COM Automation の利点](#)
- [Oracle COM Automation のアーキテクチャ](#)

Oracle COM Automation の概要

Oracle COM Automation では、Windows NT、Windows 2000 および Windows XP 上の Oracle データベース・サーバーの機能を、**Component Object Model (COM)** ベースのコンポーネントを使用してカスタマイズし、強化できます。

カスタム・コンポーネントを独自に作成することも、あるいはサード・パーティの独立系ソフトウェア・ベンダー（ISV）から入手できる数多くの作成済コンポーネントを利用することもできます。

Oracle COM Automation の機能

Oracle COM Automation には、**PL/SQL** または Java を使用して **COM** オブジェクトを操作するためのメカニズムが用意されています。Oracle COM Automation は、IDispatch インタフェースの汎用ラッパーとして機能します。

- IDispatch インタフェースによってサポートされるすべてのメソッドを使用可能にします。
- **COM** オブジェクトは、プロパティ、データ属性およびメソッド（アクションを実行するファンクション）を開発者に公開します。
- IDispatch インタフェースは、すべての **COM** オブジェクトに対する 3 つの基本操作をサポートします。
 - 公開されたプロパティの値の取得
 - 公開されたプロパティの値の設定
 - オブジェクトに対するメソッドのコール

Oracle COM Automation の API が **PL/SQL** または JSP からコールされると、Oracle COM Automation によりパラメータが適切な COM オートメーション・データ型に変換されます。次に、変換されたパラメータの値を使用して、対応する IDispatch API がコールされます。

関連項目： データ型および API の説明は、[第 3 章「Oracle COM Automation の基本機能」](#) を参照してください。

PL/SQL 用の Oracle COM Automation

PL/SQL 用の Oracle COM Automation は、**PL/SQL** パッケージを提供し、**COM** オブジェクトをインスタンス化するための一連の API を公開します。開発者は、これらの API を **PL/SQL** サブプログラム、ストアド・プロシージャ、ストアド・ファンクションまたはトリガーからコールして **COM** オブジェクトを操作できます。

COM オブジェクトを置く場所に関する制限はありません。**COM** オブジェクトはデータベース・サーバーに対してローカルに配置するか、**Distributed Component Object Model (DCOM)** を介してリモートでアクセスできます。

Java 用の Oracle COM Automation

Java 用の Oracle COM Automation は、COM オブジェクトをインスタンス化するための一連の Java API を提供します。開発者は、これらの API を JSP、Java ファンクションまたは Java トリガーからコールして COM オブジェクトを操作できます。

Java 用の Oracle COM Automation では、インプロセス COM Automation サーバーを使用できません。開発者は、dllhost を使用し、インプロセス・サーバーをサポートできます。

Oracle COM Automation の利点

Oracle COM Automation は、Windows NT、Windows 2000 および Windows XP における Oracle 開発者のための強力かつ有用なインフラストラクチャ・テクノロジーです。Oracle COM Automation には、次のような利点があります。

- 開発の容易さ

Oracle COM Automation は、COM オブジェクトを操作するための簡単な一連の API を公開します。COM および Microsoft Visual Basic に習熟した開発者であれば、これらの API を既存の **PL/SQL** サブプログラムまたは **Java** プログラムに容易に組み込むことができます。

- 再利用性

Oracle COM Automation により、社内またはサード・パーティの独立系ソフトウェア・ベンダー（ISV）によって開発された、作成済の COM コンポーネントを利用できます。目的に応じて選択できる数多くの COM コンポーネントがすでに存在しています。COM コンポーネントのマーケットは急速に拡大しつつあり、プログラマが解決する必要のあるほとんどの一般的な問題へのソリューションがすでに提供されています。

- 柔軟性と拡張性

Oracle COM Automation を使用することにより、データベース・サーバーの機能をカスタマイズし、強化することができます。COM コンポーネントを使用し、Oracle データベースをカスタマイズして次のことができます。

- Microsoft Word、Microsoft Excel、Microsoft PowerPoint などの生産性アプリケーション間でデータを交換
- Seagate Crystal Reports を使用してレポートを生成
- MAPI 準拠のアプリケーションで電子メールを送受信

データベース・サーバーのカスタマイズおよび機能拡張に制限はありません。

- 強化された統合

Oracle COM Automation では、Oracle と Windows を組み合わせた環境に Oracle を配置できます。Windows NT、Windows 2000、Windows XP、Microsoft BackOffice アプリケーションおよび Microsoft Office アプリケーションで公開されているサービスと Oracle を完全に統合し、Oracle からそれらのサービスを利用できます。

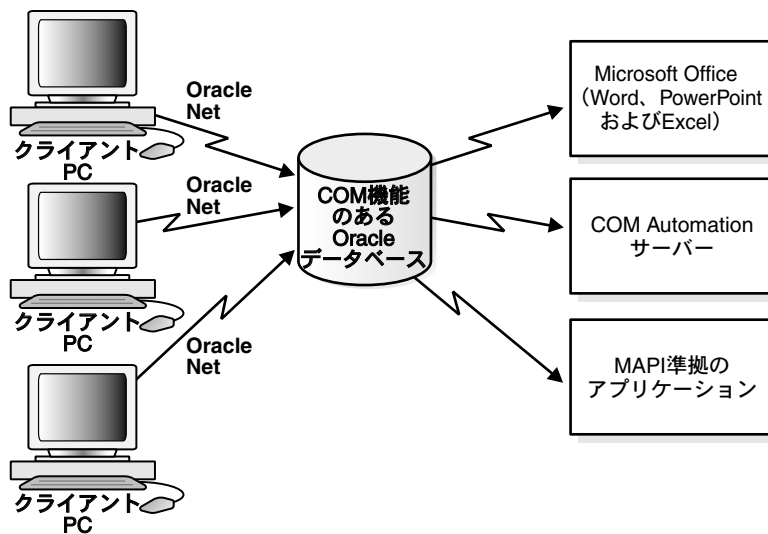
- 移植性およびプラットフォーム固有の要件

Oracle COM Automation を使用しているアプリケーションは、プラットフォームに依存しない Java または **PL/SQL** で作成されています。Windows NT、Windows 2000 または Windows XP で実行する必要があるのは、COM コンポーネントをコールする必要があるデータベース・インスタンスのみです。

Oracle COM Automation のアーキテクチャ

図 1-1 「Oracle COM の相互関係」は、Oracle9i データベース、Oracle COM Automation、クライアント・アプリケーションおよびサーバー・アプリケーションの関係を示したものです。

図 1-1 Oracle COM の相互関係



PL/SQL 用の Oracle COM Automation と Java 用の Oracle COM Automation のアーキテクチャ上の違いについて、次の 2 つの項で説明します。

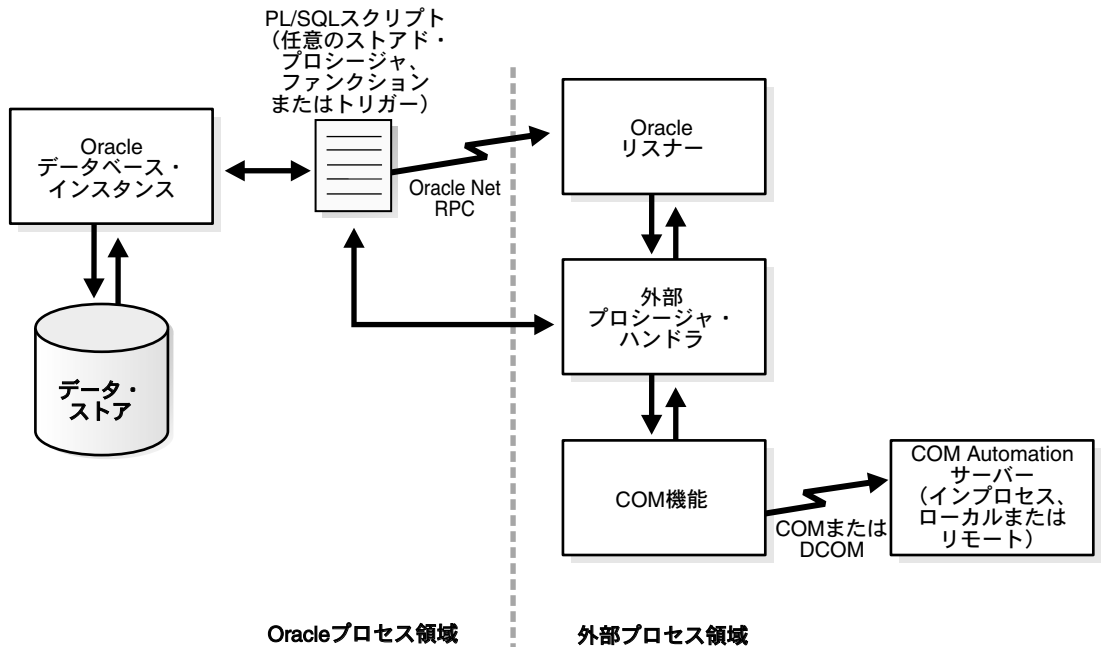
PL/SQL のアーキテクチャ

PL/SQL 用の Oracle COM Automation は、COM オブジェクトを操作する PL/SQL API のパッケージを提供しています。これらの API は、**Dynamic Link Library (DLL)** 内の**外部プロシージャ**として実装されています。

Oracle9i は、開発者がサーバー・ベースのオブジェクト型のメソッドおよびストアード・プロシージャから第 3 世代言語 (3GL) 関数をコールできる**外部プロシージャ**をサポートします。外部プロシージャは、標準の PL/SQL ストアド・プロシージャとまったく同じようにコールされます。ただし、プロシージャの本体が PL/SQL で記述され、データベースに保存されている標準の PL/SQL プロシージャとは異なり、外部プロシージャは **DLL** 内の **C** 言語関数です。Oracle COM Automation の API は、標準の PL/SQL ストアド・プロシージャまたはファンクションをコールするのと同じ方法でコールできます。

図 1-2 「PL/SQL 用の COM Automation のアーキテクチャ」は、Oracle9i データベースによって COM Automation の**外部プロシージャ** API がコールされる様子を示したものです。

図 1-2 PL/SQL 用の COM Automation のアーキテクチャ



COM Automation の外部プロシージャ API のコール

データベース・サーバーは、次のように任意の COM Automation の外部プロシージャ API をコールします。

1. PL/SQL インタプリタは、Oracle COM Automation の DLL (orawpcom.dll) へのパス名を検索します。
2. 現行のユーザー・セッションに対して extproc.exe がまだ起動されていない場合、PL/SQL インタプリタは **Oracle Net** を使用して **リスナー** にメッセージを送信し、extproc.exe を起動します。
3. PL/SQL インタプリタはプロシージャ名、パラメータおよび DLL のパス名を extproc.exe に渡します。
4. extproc.exe は DLL をロードし、**外部プロシージャ** を実行します。それぞれの COM Automation の外部プロシージャ API により、COM オブジェクトのインスタンス化、COM オブジェクトのプロパティの設定または取得、あるいは COM オブジェクトのメソッドのコールをする Win32 API が順次コールされます。
5. extproc.exe は媒介として機能し、Oracle COM Automation とデータベース・サーバー間のすべての対話を処理します。

可用性の問題に関するアーキテクチャ上の影響

PL/SQL 用の Oracle COM Automation による外部プロシージャへの依存性は、データベース・サーバーの可用性と密接に関係します。

Oracle COM Automation と、カスタムまたはサード・パーティ製の COM オブジェクトを本番環境で使用しても、データベース・サーバーの可用性に悪影響は出ません。Oracle COM Automation は、Oracle カーネルのアドレス空間の外部で動作します。これにより、不意に異常終了する COM オブジェクトから Oracle データベースが保護されます。

Java アーキテクチャ

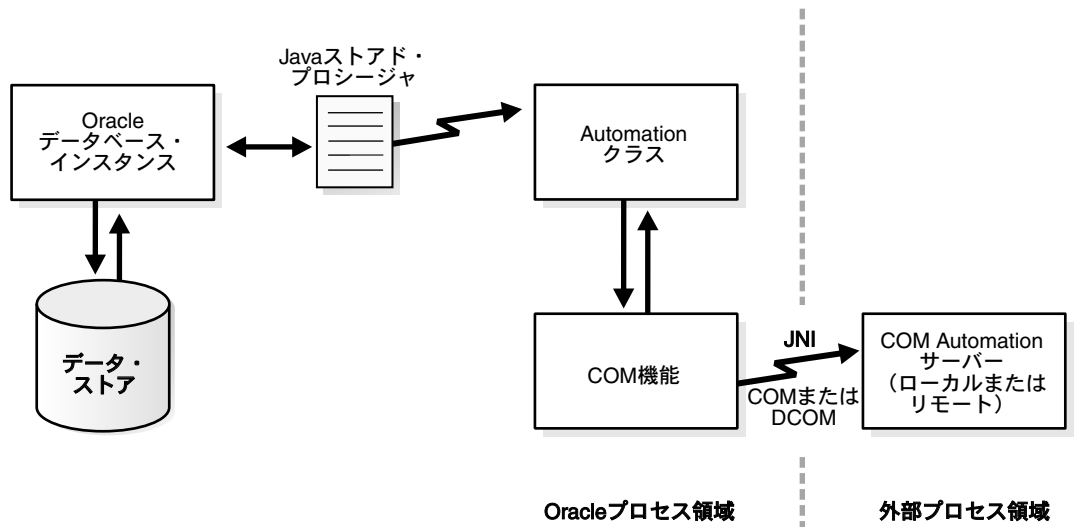
図 1-3 「Java 用の COM Automation のアーキテクチャ」に、Java 用の Oracle COM Automation が Java ネイティブ・インタフェース (JNI) によって実装される様子を示します。

このアーキテクチャの主要コンポーネントは、Automation クラスおよび Java COM プロキシ DLL である orawcom.dll です。

インタフェースは、COM Automation サーバーへの Java プロキシである Automation クラスです。Automation クラスにより、開発者が IDispatch を通じて COM オブジェクトを操作するために必要なメソッドが提供されます。

Java 固有の COM プロキシである orawcom.dll により、Java ファンクションは対応する COM ファンクションをコールできます。

図 1-3 Java 用の COM Automation のアーキテクチャ



信頼性

Java 用の Oracle COM Automation は、データベース・サーバーから COM コンポーネントをコールします。ただし、COM コンポーネントは Oracle9i データベース・プロセスの外部で実行されます。この設計により、データベース・プロセスへの不安定な COM コンポーネントの影響を回避できます。

Oracle COM Automation のインストール および構成

この章では、Oracle COM Automation のインストールおよびインストール後の構成作業の概要を説明します。

この章の項目は次のとおりです。

- Oracle COM Automation のコンポーネント
- システム要件
- Oracle8i から Oracle9i への移行
- Oracle COM Automation の構成
- PL/SQL のリスナーの構成
- DCOM のサポート

Oracle COM Automation のコンポーネント

Oracle COM Automation 機能パッケージは、Oracle のインストール時に一緒にインストールできます。このパッケージには、COM Automation 本体に加えて、この製品を使用して実際の問題を解決する方法を例示するためのデモが含まれています。

関連資料： インストール手順は、『Oracle9i Database for Windows インストール・ガイド』を参照してください。

COM Automation パッケージには、次の **PL/SQL** コンポーネントおよび Java コンポーネントが含まれています。

PL/SQL コンポーネント

- Oracle COM Automation PL/SQL (orawpcom.dll)
- PL/SQL インストールおよび定義スクリプト (comwrap.sql)
- Oracle COM Automation デモ・プログラム
- メッセージ・ファイル (comus.msb など)

Oracle COM Automation PL/SQL の orawpcom.dll は、%ORACLE_HOME%\bin ディレクトリにあります。

他のすべてのコンポーネントは、%ORACLE_HOME%\com ディレクトリにあります。

Java コンポーネント

- JAR ファイル orawcom.jar
- Oracle COM Automation Java (orawcom.dll)
- Oracle COM Automation デモ・プログラム

Oracle COM Automation Java の orawcom.dll は、%ORACLE_HOME%\bin ディレクトリにあります。他のすべてのコンポーネントは、%ORACLE_HOME%\com\java ディレクトリにあります。

システム要件

次のシステム要件をご確認ください。

- Oracle COM Automation では、次の要件を必要とします。
 - Windows NT 4.0、Windows 2000 または Windows XP
 - コンピュータ上で動作している Oracle データベース・サーバー
- PL/SQL の場合、Oracle COM Automation では次の要件を必要とします。
 - Oracle8i 以上
- Java の場合、Oracle COM Automation では次の要件を必要とします。
 - Oracle9i 以上
- Oracle COM Automation デモの場合、デモ・プログラムで使用されるアプリケーションを最初にインストールする必要があります。
 - PL/SQL および Java 用の Word デモには、Microsoft Word 95 以上が必要です。
 - Excel デモには、Microsoft Excel 95 以上が必要です。
 - PowerPoint デモには、Microsoft PowerPoint 97 以上が必要です。
 - MAPI デモには、Microsoft Outlook 2000 以上が必要です。

デモとインストールについては、4-2 ページの「[PL/SQL デモの概要](#)」および 5-2 ページの「[Oracle COM Automation Java デモの概要](#)」で説明します。

Oracle8i から Oracle9i への移行

このリリースで、Oracle は com81.dll の名前を orawpcom.dll に変更しました。Oracle8i から移行する場合、PL/SQL 用の Oracle COM Automation の使用を継続するには comwrap.sql を再実行する必要があります。

関連項目： comwrap.sql の再実行の詳細は、2-4 ページの「[PL/SQL 用の Oracle COM Automation の構成](#)」を参照してください。

Oracle COM Automation の構成

構成手順は、PL/SQL と Java とで異なります。

PL/SQL 用の Oracle COM Automation の構成

PL/SQL 用の Oracle COM Automation を構成するには、次のようにします。

1. SQL*Plus を起動します。
2. データベースに SYSTEM として接続します。

SQL> CONNECT SYSTEM/password@net_service_name
3. Oracle COM Automation を使用するデータベース・ユーザーに CREATE LIBRARY 権限を付与します。入力例は次のとおりです。

SQL> GRANT CREATE LIBRARY TO hr;
4. Oracle COM Automation を使用するユーザーに接続し、次のように、SQL*Plus プロンプトで comwrap.sql スクリプトを実行します。

```
SQL> CONNECT hr/hr;  
SQL> @%ORACLE_HOME%\com\comwrap.sql
```

%ORACLE_HOME% は、Oracle COM Automation のインストール先の Oracle ホーム・ディレクトリを表しています。

このスクリプトを初めて実行するとき、「ORA-04043: オブジェクト XXXX は存在しません。」というメッセージが数行にわたり表示されます。これらは正常なメッセージです。

Java 用の Oracle COM Automation の構成

Java 用の Oracle COM Automation を構成するには、次のようにします。

1. SQL*Plus を使用して、SYSTEM としてデータベースに接続します。入力例は次のとおりです。

SQL> CONNECT SYSTEM/password@net_service_name
2. COM Automation を使用するユーザーの名前で、grant.sql を実行します。ユーザー名をすべて大文字にする必要がある場合もあります。入力例は次のとおりです。

SQL> @%ORACLE_HOME%\com\java\grant.sql HR
3. コマンド・プロンプトで、次のように loadjava を実行します。

```
loadjava -force -resolve -user hr/hr  
%ORACLE_HOME%\com\java\orawcom.jar
```

hr は、Oracle COM Automation を使用するユーザーです。

関連資料： loadjava ユーティリティの詳細は、『Oracle9i Java Developer's Guide』を参照してください。

PL/SQL のリスナーの構成

PL/SQL 用の Oracle COM Automation を使用する場合、**listener.ora** ファイルおよび **tnsnames.ora** ファイルについて、この項で説明する特定の要件があります。

Java 用の Oracle COM Automation では、これらのファイルに特別な変更を加える必要はありません。

PL/SQL 用の Oracle COM Automation は**外部プロシージャ**のコールに依存しているため、Oracle COM Automation を正しく動作させるには、**リスナー**および **Oracle Net** のリモート・プロシージャ・コール (RPC) のメカニズムを構成する必要があります。

次に、外部ストアド・プロシージャをコールするためにプロセス間通信 (IPC) で使用できる **listener.ora** ファイルおよび **tnsnames.ora** ファイルの例を示します。

関連資料： 外部プロシージャ用の listener.ora ファイルおよび tnsnames.ora ファイルの構成の詳細は、『Oracle9i Net Services 管理者ガイド』を参照してください。

listener.ora 構成ファイル

```
LISTENER =
  (ADDRESS_LIST =
    (ADDRESS=
      (PROTOCOL= IPC)
      (KEY= EXTPROC0)
    )
  )
STARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10
TRACE_LEVEL_LISTENER = off
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = ORCL)
    )
    (SID_DESC =
      (SID_NAME = plsextproc)
      (PROGRAM=extproc)
    )
  )
)
PASSWORDS_LISTENER = (oracle)
```

tnsnames.ora 構成ファイル

```
EXTPROC_CONNECTION_DATA=
  (DESCRIPTION=
    (ADDRESS=
      (PROTOCOL=IPC)
    (KEY=EXTPROC0)
    )
    (CONNECT_DATA=(SID=plsextproc)
    )
  )
```

リスナーの問題のトラブルシューティング

「ORA-28575: 外部プロシージャ・エージェントへの RPC 接続をオープンできません。」というエラー・メッセージの原因として、2 種類のリスナーの問題が考えられます。

問題 1

問題： リスナーが起動されていない。

処置： コントロールパネルまたはコマンド・プロンプトから、OracleHOME_NAMETNSListener サービスを開始する必要があります。

コントロールパネルから Oracle サービスを開始するには、次のようにします。

1. 「スタート」→「設定」→「コントロールパネル」を選択します。
「コントロールパネル」ウィンドウが表示されます。
2. 「サービス」をダブルクリックします。
「サービス」ダイアログ・ボックスが表示されます。
3. リストから OracleHOME_NAMETNSListener を探し、その状態が「開始」になっていることを確認します。サービスが開始していない場合、そのサービスを選択して「開始」をクリックします。

コマンド・プロンプトから Oracle サービスを開始するには、次のようにします。

次のコマンドを入力して、コマンド・プロンプトから Oracle サービスを開始します。

```
C:\> net start service
```

service の部分には、OracleHOME_NAMETNSListener のような特定のサービス名を指定します。

問題 2

問題： リスナーが正しく構成されていない。

処置： listener.ora ファイルおよび tnsnames.ora ファイルを変更する必要があります。

関連項目： これらのファイルの構成方法の詳細は、2-5 ページの「[PL/SQL のリスナーの構成](#)」を参照してください。

DCOM のサポート

Oracle COM Automation は、Distributed Component Object Model (**DCOM**) を利用したリモートの **COM** オブジェクトへのネットワーク経由のアクセスをサポートします。

クライアントがリモート・コンピュータへのアクセス権を持つことを認証するために、DCOM は適切なセキュリティ資格証明をリモート・コンピュータに渡します。リモート・コンピュータはセキュリティ資格証明を検証し、DCOM に続行を許可します。

これらのセキュリティ資格証明は、クライアントのリスナー・サービスまたはデータベース・サービスに関連付けられているドメイン・ユーザーの権限に基づきます。表 2-1「[セキュリティ資格証明を決定するサービス](#)」は、PL/SQL および Java 用の COM Automation の、決定するサービスを示したものです。

表 2-1 セキュリティ資格証明を決定するサービス

COM Automation	資格証明を決定するサービス
PL/SQL	リスナー
Java	Oracle データベース・サービス

DCOM を使用するには、次のコンピュータに関するセキュリティ設定を構成する必要があります。

- データベース・インスタンスを実行しているコンピュータ
- リモートの **COM** オブジェクトを含むコンピュータ

データベース・インスタンスを実行しているコンピュータの構成

データベース・インスタンスを実行しているコンピュータの構成では、**リスナー**とデータベース・サービスを同じドメイン・ユーザーに設定する必要があります。

ドメイン・ユーザーへのサービスの設定

設定するサービスは、サービスをドメイン・ユーザーに設定するこの手順の **3** で選択します。

この手順は 2 回実行する必要があります。**リスナー**を設定するために 1 回実行し、データベース・サービスを設定するために 1 回実行します。順序は重要ではありません。

サービスをドメイン・ユーザーに設定するには、次のようにします。

1. 「スタート」→「設定」→「コントロール パネル」を選択します。「コントロール パネル」ウィンドウが表示されます。
2. 「サービス」をダブルクリックします。「サービス」ダイアログ・ボックスが表示されます。
3. サービスを選択し、「スタートアップ」をクリックします。このサービスは、OracleHOME_NAMETNSListener またはデータベース・サービスである必要があります。
4. 「アカウント」ラジオ・ボタンをクリックします。
5. ドメイン・ユーザーの名前を入力するか、「...」をクリックしてユーザーを指定します。
6. 選択したドメイン・ユーザーのパスワードを入力し、確認のために再度入力します。
7. 「OK」をクリックして、変更を保存します。

リモート・オブジェクトを含むコンピュータの構成

リモート・オブジェクトを含むコンピュータの構成では、Microsoft 社が提供している dcomcnfg.exe ツールを使用して、コンピュータの **DCOM** セキュリティ設定を構成する必要があります。

このツールを使用して、特定の **COM** オブジェクト、またはコンピュータ上のすべての COM オブジェクトに対するアクセス権限、起動権限および構成権限を設定できます。

dcomcnfg.exe ツールを使用し、次のように設定します。

1. **DCOM** セキュリティ権限を設定して、ドメイン・ユーザーとして使用する適切なサービス（**PL/SQL** の場合は**リスナー**、**Java** の場合はデータベース・サービス）が、リモートの COM オブジェクトをインスタンス化して操作できるようにします。
2. リモートの **COM** オブジェクトを、サービスと同じ権限で実行されるように設定します。

COM オブジェクトが権限のないアクションを実行しようとする、**DCOM** はその処理を拒否し、**Oracle COM Automation** にセキュリティ違反が返されます。必ず **DCOM** セキュリティを正しく構成し、必要な権限を **Oracle** データベースに指定する必要があります。

関連資料： 次の項目の詳細は、Microsoft 社のドキュメントを参照してください。

- `dcomcnfg.exe` ツールの使用および関連する権限の意味
- **DCOM** を使用するクライアントおよびサーバー・コンピュータの設定

Oracle COM Automation の基本機能

この章では、Oracle COM Automation のプログラム・インタフェースについて説明します。
この章の項目は次のとおりです。

- [データ型変換](#)
- [HRESULT エラー・コード](#)
- [Java 用の Oracle COM Automation の例外処理](#)
- [COM Automation の一般的な機能](#)
- [API](#)
- [PL/SQL API](#)
- [Java API](#)

データ型変換

Microsoft COM Automation では COM オートメーション・データ型が使用され、Oracle COM Automation では PL/SQL または Java のデータ型が使用されるため、Oracle COM Automation では、受け取ったデータを変換して OLE オートメーション・オブジェクトに渡す操作、またはその逆の操作を行う必要があります。

PL/SQL のデータ型変換

表 3-1 「PL/SQL データ型から COM オートメーション・データ型への変換」は、PL/SQL データ型と COM オートメーション・データ型の間のマッピングを示したものです。

このマニュアルでは、COM オートメーション・データ型が IN OUT パラメータまたは OUT パラメータとして使用される場合、先頭に p を付けて表記しています。先頭の文字が p 以外のデータ型は、IN パラメータです。

表 3-1 PL/SQL データ型から COM オートメーション・データ型への変換

PL/SQL データ型	COM オートメーション・データ型
VARCHAR2	BSTR、pBSTR
BOOLEAN	BOOL、pBOOL
BINARY_INTEGER	DISPATCH、pDISPATCH
DOUBLE PRECISION	UI1、pUI1、I2、pI2、I4、pI4、R4、pR4、R8、pR8、SCODE、pSCODE、CY、pCY、DISPATCH、pDISPATCH
DATE	DATE、pDATE

注意： Oracle では、CY および pCY の値を -999999999.9999 ～ 999999999.9999 の範囲内に制限しています。

Java のデータ型変換

表 3-2 「Java データ型から COM オートメーション・データ型への変換」は、サポートされる COM オートメーション・データ型および Java データ型への関連マッピングを示したものです。

戻り値のみに適用される void を除き、すべてのデータ型マッピングは、プロパティ、引数および戻り値に適用されます。

表 3-2 Java データ型から COM オートメーション・データ型への変換

Java データ型	COM オートメーション・データ型
boolean	BOOL
char	CHAR
double	DOUBLE
int	INT
long	LONG
float	FLOAT
short	SHORT
byte	BYTE
java.lang.String	BSTR
oracle.win.com.Currency	CURRENCY
java.util.Calendar	DATE
void	VOID (戻り値のみ)
oracle.win.com.Automation	IDispatch*

HRESULT エラー・コード

HRESULT エラー・コードは、Microsoft Windows API により提供されています。

HRESULT は、0x800nnnnn という形式の 16 進数の COM エラー・コードです。ただし、-214nnnnnn という 10 進数形式もあります。たとえば、COM オブジェクトの作成時に不正なオブジェクト名が渡されると、HRESULT 値 -2147221005 が返されます。これは 16 進数で表すと 0x800401f3 です。

HRESULT リターン・コードの詳細は、Microsoft 社のドキュメントを参照してください。

関連項目： 追加情報は、A-4 ページの「[Microsoft COM Automation のエラー](#)」を参照してください。

PL/SQL での HRESULT の使用

PL/SQL API は、整数のリターン・コードを返します。成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

関連項目： Oracle COM Automation から返されるリターン・コードの解釈方法の詳細は、3-12 ページの「[GetLastError](#)」を参照してください。

Java での HRESULT の使用

Java API では、HRESULT は COMException クラスのデータ・メンバーです。

関連項目： 3-4 ページの「[Java 用の Oracle COM Automation の例外処理](#)」

Java 用の Oracle COM Automation の例外処理

Java 用の Oracle COM Automation は、標準の Java 例外メカニズムを使用します。具体的には、Java 例外クラス oracle.win.com.COMException が、COM エラーを表すために組み込まれています。

この例外は、エラーの発生時に Java の Automation クラスでスローされます。

この例外で提供されるエラー情報は、PL/SQL API の GetLastError ファンクションで提供される情報と同様です。

注意： HRESULT データ・メンバーは、PL/SQL ファンクションから返される HRESULT と同じ意味を持ちます。

excepInfo データ・メンバーで示される COM エラーが DISP_E_EXCEPTION の場合、COMException は source、description、helpfile および helpid の各データ・メンバーを使用します。それ以外の場合、データ・メンバーは無効です。

COMException は、COM エラーを表すエラー・メッセージを errMsg データ・メンバーに書き込みます。

表 3-3 COMException データ・メンバー

メンバー	説明
hresult	Windows API により定義されている HRESULT 値です。
errMsg	適切な言語での HRESULT のテキスト表現です。
source	例外の発生元で、通常はアプリケーション名です。
description	エラーの説明です。
helpfile	エラーに関する詳細情報を含むヘルプ・ファイルの完全修飾パス名です。
helpid	helpfile で指定されたヘルプ・ファイル内のトピックのヘルプ・コンテキスト ID です。
excepInfo	true の場合、HRESULT の値は DISP_E_EXCEPTION で、source、description、helpfile および helpid には詳細情報が含まれます。

コード・サンプル

この例では、COMException 例外を示します。

```
try
{
    // Some code which might throw a COMException exception.
}
catch(COMException e)
{
    System.out.println(e.toString());
    if (e.excepInfo)
    {
        System.out.println(e.source);
        System.out.println(e.description);
        System.out.println(e.helpfile);
        System.out.println(e.helpid);
    }
}
```

COM Automation の一般的な機能

この項では、Oracle COM Automation を使用したソリューションの開発に必要な情報および一般的な開発手順について説明します。

COM オブジェクトに関して必要な情報

使用する **COM** オブジェクトに関して次の情報を確認します。

- COM オブジェクトの ProgID を決定する必要があります。プログラム ID (**progID**) は、Globally Unique Identifier (**GUID**) にマップされる説明的な文字列です。GUID は 16 進の数値で、COM オブジェクトを一意に識別します。

次の文字列は、progID の例です。

```
Excel.Worksheet.1
```

progID は、COM オブジェクトをインスタンス化する API で使用します。

- COM オブジェクトの IDispatch インタフェースを通じて公開されるプロパティとメソッドの種類を知る必要があります。通常、オブジェクトのプロパティの名前とデータ型、および、オブジェクトのメソッドのプロトタイプが記述されたドキュメントが ISV から提供されます。プロパティは xpos や ypos のような説明的な文字列によって参照されます。プロパティには INT や BSTR など、任意の標準 COM オートメーション・データ型を使用できます。API の GetProperty と SetProperty は、プロパティ名と適切なデータ型の変数を受け取ります。メソッドは InsertChart のような説明的な文字列によって参照されます。メソッドは、様々な種類の COM オートメーション・データ型のパラメータ集合を受け取り、COM オートメーション・データ型を返します。

次に示すのは、COM インタフェース定義言語 (IDL) の文法で記述された、COM オートメーション・メソッドのプロトタイプの例です。

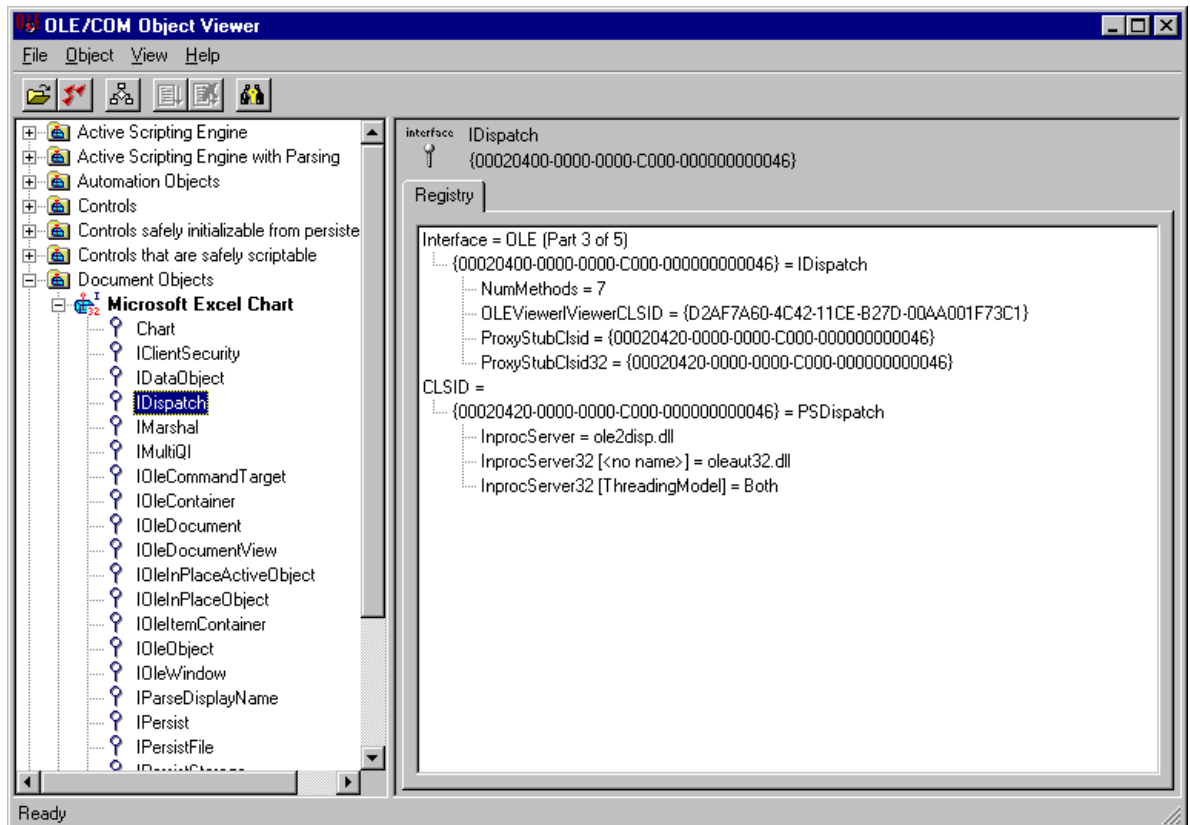
```
[id(0x6003000)]  
long Post([in, out] long* lngAccountNo,  
          [in, out] long* lngAmount,  
          [in, out] BSTR* strResult);
```

インタフェースは、オブジェクト・メソッドおよびプロパティを定義します。COM IDL を使用し、COM オブジェクトに定義されているインタフェースを指定します。

OLE-COM オブジェクト ビューア

Microsoft 社は、ローカル・システム上の COM オブジェクトのプロパティおよびメソッドの表示用に、Microsoft Visual Studio の OLE-COM オブジェクト ビューアというツールを提供しています。このツールを使用することで、各 COM オブジェクトが提供するプロパティとメソッドをすばやく、かつ容易に判断できます。図 3-1 「OLE-COM オブジェクト ビューア」の例を参照してください。

図 3-1 OLE-COM オブジェクト ビューア



COM Automation の API の使用

Oracle COM Automation の代表的な使用例では、Java クラスまたは PL/SQL ブロックを設計し、COM オブジェクトを作成および操作します。クラスまたはコード・ブロックは、次のステップを実行します。

1. COM オブジェクトを次のように作成します。
 - PL/SQL では、`CreateObject` を使用します。
 - Java では、コンストラクタまたは `Create` メソッドを使用します。
2. 次の API をコールして COM オブジェクトを操作します。
 - `GetProperty` をコールして、プロパティ値を取得します。
 - `SetProperty` をコールして、プロパティ値を新しい値に設定します。
3. `Invoke` をコールして、メソッドをコールします。

Invoke API コールの準備の一部として、Java の `InitArg` および `SetArg` を使用、または PL/SQL の `InitArg` および `SetArg` を使用して、COM オートメーション・メソッドに送られる引数をパッケージ化します。
4. PL/SQL の `GetLastError` をコールして、最新のエラー情報を取得します。
5. PL/SQL の `DestroyObject` または Java の `Destroy` を使用してオブジェクトを破棄します。

API

この項では、Oracle COM Automation で使用可能な API をリストし、説明します。

PL/SQL API

COM Automation で使用可能な PL/SQL 開発用の API には、次のようなものがあります。

- `CreateObject`
- `DestroyObject`
- `GetLastError`
- `GetProperty`
- `SetProperty`
- `InitArg`
- `InitOutArg`
- `GetArg`

- [SetArg](#)
- [Invoke](#)

Java API

COM Automation で使用可能な Java 開発用の API には、次のようなものがあります。

- [Automation](#) コンストラクタ
- Automation メソッド
 - [Create](#)
 - [Destroy](#)
 - [GetProperty](#)
 - [SetProperty](#)
 - [InitArg](#)
 - [SetArg](#)
 - [Invoke](#)
- [Currency](#) コンストラクタ
- Currency メソッド
 - [Get](#)
 - [Set](#)

PL/SQL API

この項では、COM オートメーション・インタフェースを使用して COM オブジェクトを操作するための PL/SQL API について説明します。これらの PL/SQL ストアド・プロシージャはそれぞれ ORDCOM パッケージに含まれています。

CreateObject

COM Automation サーバー内の COM オブジェクトをインスタンス化します。

構文

```
FUNCTION CreateObject (progid VARCHAR2, reserved BINARY_INTEGER, servername VARCHAR2,
objecttoken OUT BINARY_INTEGER) RETURN BINARY_INTEGER;
```

引数	説明
progid	<p>作成する COM オートメーション・オブジェクトのプログラム識別子 (progID)。この文字列は、COM オートメーション・オブジェクトのクラスを記述します。形式は次のとおりです。</p> <p><i>COMComponent.Object</i></p> <p><i>COMComponent</i> は COM Automation サーバーのコンポーネント名、<i>Object</i> は COM オートメーション・オブジェクトの名前です。指定された COM オートメーション・オブジェクトは作成可能である必要があり、また IDispatch インタフェースをサポートしている必要があります。</p>
reserved	<p>将来の使用のために現在予約されているパラメータ。値 0 を渡してください。このパラメータは、今後のバージョンの Oracle COM Automation で使用する可能性があります。</p>
servername	<p>COM オブジェクトがインスタンス化されるリモート DCOM サーバーの名前。</p> <p>名前を明示的に指定すると、Oracle COM Automation はリモート・コンピュータ上で COM オブジェクトのインスタンス化を試みます。'' のような空の文字列を渡すと、Oracle COM Automation はレジストリをチェックし、COM オブジェクトの場所を調べます。レジストリ情報は、COM オブジェクトがローカルかリモートかを示します。したがって、ローカルの COM オブジェクトを作成する場合は、COM オブジェクトがローカルに存在することをレジストリが示していることを確認した上で、常に空の文字列を渡すようにします。COM オブジェクトのレジストリ情報は、ツール dcomcnfg.exe を使用して構成できます。</p>
objecttoken	<p>返されるオブジェクト・トークン。データ型 BINARY_INTEGER のローカル変数である必要があります。このオブジェクト・トークンは、作成された COM オートメーション・オブジェクトを識別し、他の Oracle COM Automation API コールで使用されます。</p>

備考

作成された COM オートメーション・オブジェクトは、対応する DestroyObject のコールによって解放されます。このコールにより、Oracle COM Automation 内でのオブジェクトの内部表現が破棄され、オブジェクトに関連付けられたすべてのインタフェースが解放されます。

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

コード・サンプル

```
HRESULT BINARY_INTEGER;  
applicationToken BINARY_INTEGER:=-1;  
  
HRESULT :=ORDCOM.CreateObject('Excel.Application', 0, '', applicationToken);  
IF HRESULT = -1 THEN  
    dbms_output.put_line(HRESULT);  
END IF;
```

DestroyObject

作成された COM オートメーション・オブジェクトを破棄します。

構文

```
FUNCTION DestroyObject(objecttoken BINARY_INTEGER) RETURN BINARY_INTEGER;
```

引数	説明
objecttoken	事前に CreateObject によって作成された COM オートメーション・オブジェクトのオブジェクト・トークン。

備考

DestroyObject をコールすると、Oracle COM Automation 内でのオブジェクトの内部表現が破棄され、オブジェクトに関連付けられたすべてのインタフェースが解放されます。

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

コード・サンプル

```
HRESULT BINARY_INTEGER;
applicationToken BINARY_INTEGER:=-1;

/*
  At some point before this, we called CreateObject and
  got a valid applicationToken.
*/
HRESULT:=ORDCOM.DestroyObject(applicationToken);
```

GetLastError

最後に発生したエラーについての COM オートメーション・エラー情報を取得します。

構文

```
FUNCTION GetLastError(source OUT VARCHAR2, description OUT VARCHAR2, helpfile OUT
VARCHAR2, helpid OUT BINARY_INTEGER) RETURN BINARY_INTEGER;
```

引数	説明
source	エラー情報のソース。指定する場合、この値は CHAR 型または VARCHAR 型のローカル変数である必要があります。戻り値は必要に応じて、ローカル変数として適切な値になるよう切り捨てられます。
description	エラーの説明。指定する場合、この値は CHAR 型または VARCHAR 型のローカル変数である必要があります。戻り値は必要に応じて、ローカル変数として適切な値になるよう切り捨てられます。
helpfile	COM オートメーション・オブジェクトについてのヘルプ・ファイル。指定する場合、この値は CHAR 型または VARCHAR 型のローカル変数である必要があります。戻り値は必要に応じて、ローカル変数として適切な値になるよう切り捨てられます。
helpid	ヘルプ・ファイルのコンテキスト ID。指定する場合、この値は INT 型のローカル変数である必要があります。

備考

(GetLastError を除く) Oracle COM Automation API コールごとにエラー情報がリセットされます。このため、GetLastError では、最新の Oracle COM Automation API コールに対するエラー情報のみが取得されます。GetLastError は最後のエラー情報をリセットしないため、この API を複数回コールすると、同じエラー情報が取得されることがあります。

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

このファンクションによって返される可能性のあるエラーの種類については、A-4 ページの「[Microsoft COM Automation のエラー](#)」を参照してください。

コード・サンプル

```
applicationToken BINARY_INTEGER:= -1;
HRESULT BINARY_INTEGER;
error_src VARCHAR2(255);
error_description VARCHAR2(255);
error_helpfile VARCHAR2(255);
error_helpID BINARY_INTEGER;

HRESULT:=ORDCOM.CreateObject('Excel.Application', 0, '', applicationToken);
IF HRESULT=-1 THEN
    ORDCOM.GetLastError(error_src, error_description, error_helpfile, error_helpID);
    dbms_output.put_line(error_src);
    dbms_output.put_line(error_description);
    dbms_output.put_line(error_helpfile);
    return HRESULT;
END IF;
```

GetProperty

COM オートメーション・オブジェクトのプロパティ値を取得します。

構文

```
FUNCTION GetProperty(objecttoken BINARY_INTEGER, propertyname VARCHAR2, argcount
BINARY_INTEGER, propertyvalue OUT any_PL/SQL_datatype) RETURN BINARY_INTEGER;
```

引数	説明
objecttoken	事前に CreateObject によって作成された COM オブジェクトのオブジェクト・トークン。
propertyname	返される COM オブジェクトのプロパティ名。
argcount	プロパティ配列のインデックス。プロパティが配列でない場合、この値には 0 を指定する必要があります。
propertyvalue	返されるプロパティ値。返されるプロパティの型は、返される COM オートメーションのデータ型によって異なります。COM オートメーション・プロパティの COM オートメーション・データ型に対応する PL/SQL データ型を渡す必要があります。そうしないと、COM Automation によって COM オートメーション・データ型が適切に変換されません。
any_PL/SQL_datatype	COM Automation でサポートされているデータ型。

備考

プロパティが COM オブジェクトを返す場合、propertyvalue パラメータにデータ型 BINARY_INTEGER のローカル変数を指定する必要があります。オブジェクト・トークンはローカル変数に格納されます。格納されたオブジェクト・トークンは、他の COM オートメーション・ストアド・プロシージャで使用できます。

プロパティが配列を返すとき、propertyvalue を指定するとその値は NULL に設定されます。

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

コード・サンプル

```
ChartObject  BINARY_INTEGER:=-1;
ChartToken   BINARY_INTEGER := -1
HRESULT      BINARY_INTEGER;
/* Previously, ChartObject was initialized calling CreateObject */
HRESULT:=ORDCOM.getProperty(ChartObject, 'Chart', 0, ChartToken);
IF HRESULT=-1 THEN
    /* Do error checking here */
    return HRESULT;
END IF;
```

SetProperty

COM オートメーション・オブジェクトのプロパティを新しい値に設定します。

構文

```
FUNCTION SetProperty(objecttoken BINARY_INTEGER, propertyname VARCHAR2, newvalue
any_PL/SQL_datatype, datatype VARCHAR2) RETURN BINARY_INTEGER;
```

引数	説明
objecttoken	事前に CreateObject によって作成された COM オートメーション・オブジェクトのオブジェクト・トークン。
propertyname	新しい値を設定する COM オブジェクトのプロパティ名。
newvalue	プロパティの新しい値。この値は適切なデータ型の値である必要があります。
any_PL/SQL_datatype	COM Automation でサポートされているデータ型。

引数	説明
datatype	渡される値の、明示的に指定されるデータ型。有効なデータ型は次のとおりです。 <ul style="list-style-type: none">■ UI1 - バイト整数■ I2 - 2 バイト整数■ I4 - 4 バイト整数■ R4 - IEEE 4 バイト実数■ R8 - IEEE 8 バイト実数■ SCODE - エラー・コード■ CY - 通貨 (-9999999999.9999 ～ 9999999999.9999 の値) (Oracle の制限)■ DISPATCH - ディスパッチ・ポインタ■ BSTR - 文字列■ BOOL - ブール■ DATE - 日付

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

コード・サンプル

```
RangeToken BINARY_INTEGER:= -1;
HRESULT     BINARY_INTEGER;

/*
   Previously, RangeToken has been initialized to a valid object token with a property
   by the name of value.
*/
HRESULT:=ORDCOM.SetProperty(RangeToken, 'Value', 'EmpNo', 'BSTR');
IF HRESULT=-1 THEN
    /* Do error checking here */
    return HRESULT;
END IF;
```

InitArg

Invoke コールに渡されるパラメータ・セットを初期化します。

構文

```
PROCEDURE InitArg();
```

備考

InitArg のコールにより、パラメータ・セットが初期化されます。InitArg のコール後、SetArg をコールすることにより、最初のパラメータが指定の値に設定されます。SetArg の 2 度目のコールでは、パラメータ・リスト中の 2 番目のパラメータが設定されます。以後のコールにより、パラメータ・リスト中の n 番目のパラメータが設定されます。 n は、InitArg のコール後に SetArg がコールされた回数です。InitArg をもう一度コールすると、引数リストはリセットされます。その後 SetArg をコールすると、再び最初のパラメータが設定されます。

コード・サンプル

3-20 ページの「[Invoke](#)」の項を参照してください。

InitOutArg

InitOutArg は、GetArg を使用して OUT パラメータおよび IN OUT パラメータの値を取得する準備として、COM メソッドがコールされた後にコールする必要があります。InitOutArg のコール後、GetArg の最初のコールで最初の OUT パラメータまたは IN OUT パラメータの値が取得され、2 度目の GetArg コールで 2 番目の OUT パラメータまたは IN OUT パラメータの値が取得されます。以降も同様に取得されます。InitOutArg を再びコールすると、このプロセスが再び開始されます。

構文

```
PROCEDURE InitOutArg();
```

備考

IN パラメータおよび OUT パラメータの詳細は、3-29 ページの「[SetArg](#)」の SetArg データ型に関する記載を参照してください。

コード・サンプル

3-30 ページの「[Invoke](#)」の項を参照してください。

GetArg

COM メソッドがコールされた後に OUT パラメータおよび IN OUT パラメータの引数を取得します。

構文

```
PROCEDURE GetArg(data OUT any_PL/SQL_datatype, type VARCHAR2);
```

引数	説明
data	COM メソッドがコールされた後の OUT パラメータまたは IN OUT パラメータの値。
any_PL/SQL_datatype	COM Automation でサポートされているデータ型。
datatype	パラメータの COM オートメーション・データ型。 使用可能なデータ型は、次のとおりです。 <ul style="list-style-type: none">■ pUI1 - バイト整数■ pI2 - 2 バイト整数■ pI4 - 4 バイト整数■ pR4 - IEEE 4 バイト実数■ pR8 - IEEE 8 バイト実数■ pSCODE - エラー・コード■ pCY - 通貨 (-9999999999.9999 ～ 9999999999.9999 の値) (Oracle の制限)■ pDISPATCH - ディスパッチ・ポインタ■ pBSTR - 文字列■ pBOOL - ブール■ pDATE - 日付

備考

IN パラメータおよび OUT パラメータの詳細は、3-29 ページの「[SetArg](#)」の SetArg データ型に関する記載を参照してください。

コード・サンプル

3-30 ページの「[Invoke](#)」の項を参照してください。

SetArg

この API を使用して、Invoke を次回コールするためのパラメータ・リストを構成します。
SetArg は、渡されるパラメータの値を設定します。

構文

```
PROCEDURE SetArg(paramvalue any_PL/SQL_datatype, datatype VARCHAR2);
```

引数	説明
paramvalue	Invoke コールに渡されるパラメータの値。設定されるパラメータは、パラメータ・リスト中の <i>n</i> 番目のパラメータです。 <i>n</i> は、InitArg のコール後に SetArg がコールされた回数です。
datatype	<p>明示的に指定されるパラメータのデータ型。</p> <p>先頭の文字が <i>p</i> のデータ型は、IN OUT パラメータまたは OUT パラメータです。<i>p</i> は、COM オートメーション・データ型に対して VT_BYREF フラグが設定されることを示します。</p> <p>先頭の文字が <i>p</i> 以外のデータ型は、IN パラメータです。使用可能なデータ型は、次のとおりです。</p> <ul style="list-style-type: none">■ UI1 - バイト整数■ pUI1 - バイト整数■ I2 - 2 バイト整数■ pI2 - 2 バイト整数■ I4 - 4 バイト整数■ pI4 - 4 バイト整数■ R4 - IEEE 4 バイト実数■ pR4 - IEEE 4 バイト実数■ R8 - IEEE 8 バイト実数■ pR8 - IEEE 8 バイト実数■ SCODE - エラー・コード■ pSCODE - エラー・コード

引数	説明
	<ul style="list-style-type: none"> ■ CY - 通貨 (-9999999999.9999 ～ 9999999999.9999 の値) (Oracle の制限) ■ pCY - 通貨 (-9999999999.9999 ～ 9999999999.9999 の値) (Oracle の制限) ■ DISPATCH - ディスパッチ・ポインタ ■ pDISPATCH - ディスパッチ・ポインタ ■ BSTR - 文字列 ■ pBSTR - 文字列 ■ BOOL - ブール ■ pBOOL - ブール ■ DATE - 日付 ■ pDATE - 日付
<i>any_PL/SQL_datatype</i>	COM Automation でサポートされているデータ型。

備考

SetArg プロシージャはそれぞれ、*n* 番目のパラメータ値を設定します。InitArg のコールにより、パラメータ・セットが初期化されます。InitArg のコール後、SetArg をコールすることにより、最初のパラメータが指定の値に設定されます。SetArg の 2 度目のコールでは、パラメータ・リスト中の 2 番目のパラメータが設定されます。以後のコールにより、パラメータ・リスト中の *n* 番目のパラメータが設定されます。*n* は、InitArg のコール後に SetArg がコールされた回数です。InitArg をもう一度コールすると、引数リストはリセットされます。その後 SetArg をコールすると、再び最初のパラメータが設定されます。

先頭の文字が *p* 以外のデータ型は、IN パラメータです。先頭の文字が *p* のデータ型は、IN OUT パラメータまたは OUT パラメータです。

コード・サンプル

3-20 ページの「[Invoke](#)」の項を参照してください。

Invoke

COM オートメーション・オブジェクトのメソッドをコールします。このファンクションでは、COM オートメーション・メソッドへの入力値として、InitArg および SetArg のコールによって事前に作成されたパラメータ・リストが使用されます。

構文

```
FUNCTION Invoke(objecttoken BINARY_INTEGER, methodname VARCHAR2, argcount
BINARY_INTEGER, returnvalue OUT any_PL/SQL_datatype) RETURN BINARY_INTEGER;
```

引数	説明
objecttoken	事前に CreateObject によって作成された COM オートメーション・オブジェクトのオブジェクト・トークン。
methodname	コールする COM オートメーション・オブジェクトのメソッド名。
argcount	COM オートメーション・オブジェクトのメソッドに渡される引数の数。
returnvalue	COM オートメーション・オブジェクトのメソッドの戻り値。指定される場合、この値は適切なデータ型のローカル変数である必要があります。
any_PL/SQL_datatype	COM Automation でサポートされているデータ型。

備考

メソッドの戻り値が COM オブジェクトである場合、returnvalue パラメータに対しては、BINARY_INTEGER データ型のローカル変数を指定する必要があります。オブジェクト・トークンはローカル変数に格納されます。格納されたオブジェクト・トークンは、他の Oracle COM Automation API で使用できます。

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

コード・サンプル

```

/*
 * Following is the IDL definition of the COM Automation method
 * being called:
 *
 * HRESULT TestOutArg([in, out]      short *x1,
 *                    [in]           short  x2,
 *                    [out]          short *x3,
 *                    [out, retval] short *x4);
 */

applicationToken binary_integer:=-1;
i binary_integer:=-1;
x1 double precision:=12;
x2 double precision:=7;
x3 double precision:=0;
x4 double precision:=0;

/* Assume applicationToken is initialized. */

ORDCOM.InitArg();
ORDCOM.SetArg(x1, 'pI2');
ORDCOM.SetArg(x2, 'I2');
ORDCOM.SetArg(x3, 'pI2');

i:=ORDCOM.Invoke(applicationToken, 'TestOutArg', 3, x4);

ORDCOM.InitOutArg();
ORDCOM.GetArg(x1, 'pI2');
ORDCOM.GetArg(x3, 'pI2');

```

Java API

この項では、COM オートメーション・インタフェースを使用して COM オブジェクトを操作するための Java API について説明します。これらの API は、Java の Automation クラスおよび Currency クラスにあります。

Java の Automation クラスを使用し、COM オートメーションをサポートする COM オブジェクトにアクセスできます。この Java クラスでは、COM オブジェクトを作成し、COM オブジェクトの IDispatch インタフェースへのポインタを取得できます。さらに、COM オブジェクトに対してプロパティを取得および設定できる他、COM オブジェクトに対して (引数あり、または引数なしで) メソッドをコールできます。このクラスは、COM オブジェクトのラッパーを提供するため、COM オブジェクトまたはその IDispatch インタフェースには直接アクセスしません。

Java の Currency クラスは、COM オートメーション・データ型の CURRENCY に相当します。CURRENCY は 8 バイトの数値で、最後の 4 桁は値の小数部を表します。たとえば、12345 という数値は、実際には 1.2345 という値を表します。CURRENCY の範囲は (+/-)922337203685477.5807 です。

COM オブジェクト参照カウンティング

COM オブジェクト・インタフェース参照カウンティングは内部的に処理され、IUnknown::AddRef () および IUnknown::Release () は公開されません。ユーザーは、COM オブジェクト・インタフェースを明示的にアドレス指定できません。特定の COM オブジェクトの存続期間は、関連する Java コンストラクタまたは [Create](#) メソッドがコールされたときに開始し、対応する [Destroy](#) メソッドがコールされたときに解放されます。

コンストラクタおよびデストラクタ

デフォルトのコンストラクタは COM オブジェクトを作成しません。COM オブジェクトを作成するには、2 つの方法があります。

- デフォルトのコンストラクタを使用して Java オブジェクトをインスタンス化し、[Create](#) メソッドの 1 つをコールします。使用する [Create](#) メソッドは、サーバー名を指定するかどうかによって決まります。後で [Destroy](#) メソッドをコールし、COM オブジェクトを解放する必要があります。

[Create](#) メソッドはいつでもコールできますが、COM オブジェクトが以前にデフォルト以外のコンストラクタの 1 つまたは [Create](#) メソッドを介して作成された場合は、最初に [Destroy](#) メソッドをコールする必要があります。

- デフォルト以外のコンストラクタを使用して Java オブジェクトをインスタンス化します。使用するデフォルト以外のコンストラクタは、サーバー名を指定するかどうかによって決まります。後で [Destroy](#) メソッドをコールし、COM オブジェクトを解放する必要があります。

COM オブジェクト・エラーの処理

すべての COM エラーは Java の例外にマップされます。ユーザーは、Java の例外処理メカニズムにより COM オブジェクト・エラーを捕捉できます。

注意： Java 用の Oracle COM Automation では、インプロセス COM Automation サーバーを使用できません。開発者は、dllhost を使用し、インプロセス・サーバーをサポートできます。

Automation コンストラクタ

COM オブジェクトを作成します。

構文

```
public Automation()
public Automation(String progID)
public Automation(String progID, String serverName)
```

引数	説明
progid	<p>作成する COM オートメーション・オブジェクトのプログラム識別子 (progID)。この文字列は、COM オートメーション・オブジェクトのクラスを記述します。形式は次のとおりです。</p> <p><i>COMComponent.Object</i></p> <p><i>COMComponent</i> は COM Automation サーバーのコンポーネント名、<i>Object</i> は COM オートメーション・オブジェクトの名前です。指定された COM オートメーション・オブジェクトは作成可能である必要があります、また <i>IDispatch</i> インタフェースをサポートしている必要があります。</p>
servername	<p>COM オブジェクトがインスタンス化されるリモート DCOM サーバーの名前。</p> <p>名前を明示的に指定すると、Oracle COM Automation はリモート・コンピュータ上で COM オブジェクトのインスタンス化を試みます。</p>

備考

デフォルトのコンストラクタ `public Automation()` は何も行いません。このコンストラクタは `Create` メソッドで使用されます。

`progid` パラメータのみを受け取るコンストラクタを使用すると、Oracle COM Automation は COM オブジェクトの場所についてレジストリをチェックします。レジストリ情報は、COM オブジェクトがローカルかリモートかを示します。

デフォルト以外のコンストラクタを使用して作成された COM オートメーション・オブジェクトは、対応する `Destroy` のコールによって解放されます。このコールにより、Oracle COM Automation 内でのオブジェクトの内部表現が破棄され、オブジェクトに関連付けられたすべてのインタフェースが解放されます。

Java 用の Oracle COM Automation では、インプロセス COM Automation サーバーを使用できません。開発者は、`dllhost` を使用し、インプロセス・サーバーをサポートできます。

エラーが発生した場合は、例外 `COMException` がスローされます。

コード・サンプル

次のコード・サンプルは、デフォルト以外のコンストラクタの例です。

```
// Use the registry to determine where to create the COM object.
Automation word = new Automation("Word.Basic");

// Create the COM object on the specified server machine.
Automation excel = new Automation("Excel.Application",
                                   "¥¥ServerMachineName");

// Free the COM objects.
word.Destroy();
excel.Destroy();
```

Create

COM Automation サーバー内の COM オブジェクトをインスタンス化します。

構文

```
public void Create(String progID)
public void Create(String progID, String serverName)
```

引数	説明
progid	作成する COM オートメーション・オブジェクトのプログラム識別子 (progID)。この文字列は、COM オートメーション・オブジェクトのクラスを記述します。形式は次のとおりです。 <i>COMComponent.Object</i> <i>COMComponent</i> は COM Automation サーバーのコンポーネント名、 <i>Object</i> は COM オートメーション・オブジェクトの名前です。指定された COM オートメーション・オブジェクトは作成可能である必要があります、また <i>IDispatch</i> インタフェースをサポートしている必要があります。
servername	COM オブジェクトがインスタンス化されるリモート DCOM サーバーの名前。 名前を明示的に指定すると、Oracle COM Automation はリモート・コンピュータ上で COM オブジェクトのインスタンス化を試みます。

備考

Create メソッドで作成された COM オートメーション・オブジェクトは、対応する Destroy のコールによって解放されます。このコールにより、Oracle COM Automation 内でのオブジェクトの内部表現が破棄され、オブジェクトに関連付けられたすべてのインタフェースが解放されます。

progid パラメータのみを受け取るコンストラクタを使用すると、Oracle COM Automation は COM オブジェクトの場所についてレジストリをチェックします。レジストリ情報は、COM オブジェクトがローカルかリモートかを示します。

Java 用の Oracle COM Automation では、インプロセス COM Automation サーバーを使用できません。開発者は、dllhost を使用し、インプロセス・サーバーをサポートできます。

エラーが発生した場合は、例外 COMException がスローされます。

コード・サンプル

```
// Use the default constructor.
Automation word = new Automation();
Automation excel = new Automation();

// Use the registry to determine where to create the COM object.
word.Create("Word.Basic");

// Create the COM object on the specified server machine.
excel.Create("Excel.Application", "¥¥ServerMachineName");

// Free the COM objects.
word.Destroy();
excel.Destroy();
```

Destroy

作成された COM オートメーション・オブジェクトを破棄します。

構文

```
public void Destroy()
```

備考

Destroy をコールすると、Oracle COM Automation 内でのオブジェクトの内部表現が破棄され、オブジェクトに関連付けられたすべてのインタフェースが解放されます。

コード・サンプル

3-24 ページの「[Create](#)」の項を参照してください。

GetProperty

COM オートメーション・オブジェクトのプロパティ値を取得します。

構文

```
public allowed_type GetProperty(String propName, allowed_type[] propVal)
```

引数	説明
propName	返される COM オブジェクトのプロパティ名。
propVal	返されるプロパティ値。返されるプロパティの型は、返される COM オートメーションの型によって異なります。プロパティを返すためにアクセスされる要素は最初の要素のみですが、配列は少なくとも 1 つの要素を保持できる大きさである必要があります。
allowed_type	次のリストのいずれかの型。 <ul style="list-style-type: none">booleanbytecharshortintlongfloatdoublejava.long.Stringoracle.win.com.Automationoracle.win.com.Currencyjava.util.Calendar

備考

プロパティが COM オブジェクトの場合は、oracle.win.com.Automation の allowed_type を使用して取得できます。返されるオートメーション Java オブジェクトは、プロパティの取得と設定、およびプロパティに対するメソッドのコールに使用できます。

GetProperty は、GetProperty メソッドをオーバーロードするために、配列パラメータを使用してプロパティ値を返します。プロパティ値が単純に戻り値として返された場合はオーバーロードできません。配列は、out パラメータを持たない Java が原因で発生する問題を解決します。

プロパティは、便宜上戻り値として返されます。
 エラーが発生した場合は、例外 `COMException` がスローされます。

コード・サンプル

```
// A Microsoft Excel ChartObject object.
Automation chartObject = null;
// A Microsoft Excel Chart object.
Automation chart = null;
// Used for properties of type Automation.
Automation[] autoProp = { null };

// Assume the Microsoft Excel ChartObject object is initialized.

// Get the Chart property.
chartObject.GetProperty("Chart", autoProp);
chart = autoProp[0];

// Set the Chart property.
chartObject.SetProperty("Chart", chart);
```

SetProperty

COM オートメーション・オブジェクトのプロパティを新しい値に設定します。

構文

```
public void SetProperty(String propName, allowed_type propVal)
```

引数	説明
<code>propName</code>	新しい値を設定する COM オブジェクトのプロパティ名。
<code>propVal</code>	プロパティの新しい値。この値は適切なデータ型の値である必要があります。

引数	説明
<code>allowed_type</code>	次のリストのいずれかの型。 <ul style="list-style-type: none">■ <code>boolean</code>■ <code>byte</code>■ <code>char</code>■ <code>short</code>■ <code>int</code>■ <code>long</code>■ <code>float</code>■ <code>double</code>■ <code>java.lang.String</code>■ <code>oracle.win.com.Automation</code>■ <code>oracle.win.com.Currency</code>■ <code>java.util.Calendar</code>

備考

プロパティが COM オブジェクトの場合は、`oracle.win.com.Automation` の `allowed_type` を使用して設定できます。プロパティ値は、有効なオートメーション Java オブジェクトである必要があります。

エラーが発生した場合は、例外 `COMException` がスローされます。

コード・サンプル

3-26 ページの「[GetProperty](#)」の項を参照してください。

InitArg

Invoke コールに渡されるパラメータ・セットを初期化します。

構文

```
public void InitArg()
```

備考

`InitArg` コールは、パラメータ・セットを初期化し、COM メソッドがパラメータを受け取らない場合でもコールする必要があります。`InitArg` のコール後、`SetArg` をコールすることにより、最初のパラメータが指定の値に設定されます。`SetArg` の 2 度目のコールでは、パラメータ・リスト中の 2 番目のパラメータが設定されます。以後のコールにより、パラ

メータ・リスト中の n 番目のパラメータが設定されます。 n は、InitArg のコール後に SetArg がコールされた回数です。InitArg をもう一度コールすると、引数リストはリセットされます。その後 SetArg をコールすると、再び最初のパラメータが設定されます。

コード・サンプル

3-30 ページの「Invoke」の項を参照してください。

SetArg

この API を使用して、Invoke を次回コールするためのパラメータ・リストを構成します。

構文

```
public void SetArg(allowed_type val)
```

引数	説明
val	Invoke コールに渡されるパラメータの値。設定されるパラメータは、パラメータ・リスト中の n 番目のパラメータです。 n は、InitArg のコール後に SetArg がコールされた回数です。
allowed_type	次のリストのいずれかの型。 <ul style="list-style-type: none">booleanbytecharshortintlongfloatdoublejava.lang.Stringoracle.win.com.Automationoracle.win.com.Currencyjava.util.Calendar

備考

パラメータが COM オブジェクトの場合、対応する引数の `allowed_type` は `oracle.win.com.Automation` である必要があります。引数は、有効なオートメーション Java オブジェクトである必要があります。

この時点で例外はスローされません。ただし、エラーが発生した場合、たとえば不正な引数の型が渡された場合は、`Invoke` メソッドがコールされたときにエラーが捕捉されます。

コード・サンプル

3-30 ページの「[Invoke](#)」の項を参照してください。

Invoke

COM オートメーション・オブジェクトのメソッドをコールします。このファンクションでは、COM オートメーション・メソッドへの入力値として、`InitArg` および `SetArg` へのコールによって事前に作成されたパラメータ・リストが使用されます。

構文

```
public void Invoke(String methodName, allowed_type[] retVal)
public void Invoke(String methodName)
```

引数	説明
methodName	コールする COM オートメーション・オブジェクトのメソッド名。
retVal	COM オートメーション・オブジェクトのメソッドの戻り値。指定される場合、この値は適切なデータ型のローカル変数である必要があります。プロパティを返すためにアクセスされる要素は最初の要素のみですが、配列は少なくとも 1 つの要素を保持できる大きさである必要があります。

引数	説明
<code>allowed_type</code>	次のリストのいずれかの型。 <ul style="list-style-type: none"> ■ <code>boolean</code> ■ <code>byte</code> ■ <code>char</code> ■ <code>short</code> ■ <code>int</code> ■ <code>long</code> ■ <code>float</code> ■ <code>double</code> ■ <code>java.lang.String</code> ■ <code>oracle.win.com.Automation</code> ■ <code>oracle.win.com.Currency</code> ■ <code>java.util.Calendar</code>

備考

COM メソッドが戻り値として COM オブジェクトを返す場合、戻り値の `allowed_type` は `oracle.win.com.Automation` です。返されるオートメーション Java オブジェクトは、プロパティの取得と設定、および戻り値に対するメソッドのコールに使用できます。

`Invoke` メソッドをオーバーロードするには、`Invoke` が配列パラメータを使用して COM オブジェクト・メソッドの値を返す必要があります。プロパティ値が単純に戻り値として返された場合はオーバーロードできません。配列は、`out` パラメータを持たない Java が原因で発生する問題を解決します。

1 つのパラメータのみを受け取る `Invoke` の `public void Invoke(String methodName)` は、戻り値が `void` 型の COM オブジェクト・メソッドで使用されます。

プロパティは、便宜上戻り値として返されます。

エラーが発生した場合は、例外 `COMException` がスローされます。

コード・サンプル

```
// A Microsoft Excel Worksheet object.
Automation workSheet = null;
// A Microsoft Excel ChartObjects collection object.
Automation chartObjects = null;
// A Microsoft Excel ChartObject object.
Automation chartObject = null;
// Used for return values of type Automation.
Automation[] autorv = { null };
// Dimensions for a Microsoft Excel ChartObject object.
short xpos = 100, ypos = 30, width = 400, height = 250;

// Assume the Microsoft Excel Worksheet object is initialized.

// Invoke a method which takes no arguments.
workSheet.InitArg();
workSheet.Invoke("ChartObjects", autorv);
chartObjects = autorv[0];

// Invoke a method which takes multiple arguments.
chartObjects.InitArg();
chartObjects.SetArg(xpos);
chartObjects.SetArg(ypos);
chartObjects.SetArg(width);
chartObjects.SetArg(height);
chartObjects.Invoke("Add", autorv);
chartObject = autorv[0];
```

Currency コンストラクタ

Java オブジェクト currency を作成します。

構文

```
public Currency(long value)
```

引数	説明
value	8 バイトの CURRENCY 数値。

Get

8 バイトの CURRENCY 数値を取得します。

構文

```
public long Get()
```

備考

8 バイトの CURRENCY 数値を返します。

Set

8 バイトの CURRENCY 数値を設定します。

構文

```
public void Set(long value)
```

引数	説明
value	8 バイトの CURRENCY 数値。

Oracle COM Automation PL/SQL デモ

この章では、PL/SQL 用の Oracle COM Automation のデモ・プログラムの使用方法について説明します。

この章の項目は次のとおりです。

- [PL/SQL デモの概要](#)
- [Microsoft Word デモ](#)
- [Microsoft Excel デモ](#)
- [Microsoft PowerPoint デモ](#)
- [MAPI デモ](#)

PL/SQL デモの概要

PL/SQL 用の Oracle COM Automation には、Oracle COM Automation を使用してソリューションを開発する方法を示す例が含まれています。これらのデモは基本機能を提供し、COM Automation を使用するカスタマイズされた複雑なアプリケーションを作成する基礎として機能します。デモは、サンプル・スキーマで使用可能な人事スキーマに基づいています。

それぞれのデモでは、COM Automation を使用して簡単な操作を行うための基本的な API のセットを提供しています。それぞれの COM Automation サーバー（Word や Excel など）は、デモ API を通じて提供されるものよりもさらに高度な機能を提供します。これらの高度な機能を利用するためには、独自の PL/SQL プロシージャを設計し、コーディングする必要があります。

このリリースでは、オラクル社は次のデモを提供しています。

- **Microsoft Word デモ**： Oracle から Word にデータを挿入します。
- **Microsoft Excel デモ**： Oracle から Excel にデータを挿入します。
- **Microsoft PowerPoint デモ**： Oracle から PowerPoint にデータを挿入します。
- **MAPI デモ**： Oracle から Messaging Application Programming Interface (MAPI) 準拠のアプリケーションにデータを挿入します。

Microsoft Word デモ

次の各項では、Microsoft Word デモのインストール方法と、このデモで提供されている API について説明します。このデモは、Oracle と Microsoft Word を使用して開発できるソリューションの例として提供されています。

デモでは、データベース内の従業員の名前を含む Microsoft Word ドキュメントを作成します。

Microsoft Word デモは、次のものを提供します。

- **ORDWord**。これは、Microsoft Word を操作するいくつかの API を公開する PL/SQL パッケージです。このパッケージは、wordsol.sql スクリプトにより作成されます。
- **worddem.sql**。これは、Oracle と Microsoft Word 間でのデータ交換機能を示すスクリプトです。このスクリプトは、EMPLOYEES 表および JOBS 表から Microsoft Word ドキュメントにデータを挿入します。これらの表は、サンプル・スキーマ内の人事スキーマにあります。

Microsoft Word デモのインストール

このデモをインストールする前に、ローカル・コンピュータ上に Microsoft Word がインストールされている必要があります。

Microsoft Word デモをインストールするには、次のようにします。

1. SQL*Plus を起動します。

```
C:¥> sqlplus /nolog
```

2. Microsoft Word デモを使用するユーザーのアカウントで、Oracle データベースのインスタンスに接続します。入力例は次のとおりです。

```
SQL> connect hr/hr
```

3. SQL*Plus プロンプトから、wordsol.sql スクリプトを実行します。

```
SQL> @%ORACLE_HOME%¥com¥demos¥wordsol.sql;
```

このスクリプトにより、現行のユーザーのスキーマ内に ORDWord パッケージが作成されます。このスクリプトを初めて実行するときに、次のエラーが数回発生します。

```
ORA-04043: オブジェクト XXXX は存在しません。
```

これらは正常なメッセージです。

Microsoft Word デモの使用方法

Microsoft Word デモを使用するには、次のようにします。

1. SQL*Plus プロンプトから、worddem.sql スクリプトを実行します。

```
SQL> @%ORACLE_HOME%¥com¥demos¥worddem.sql;
```

このスクリプトにより、C:¥ディレクトリに Microsoft Word ドキュメント (worddemo.doc) が作成されます。このドキュメントには、EMPLOYEES 表および JOBS 表から取得されたデータが含まれています。これらの表は、サンプル・スキーマ内の人事スキーマにあります。

2. worddemo.doc を開いて、その内容を確認します。

基本機能

次の各項では、Microsoft Word デモで提供される API について説明します。これらの API は基本的な機能です。Microsoft Word が COM Automation を通じて提供する機能は、これらの API を通じて提供されるもの以外にも多数存在します。

CreateWordObject

Microsoft Word オートメーション・サーバー内の Word.Basic オブジェクトをインスタンス化します。

構文

```
FUNCTION CreateWordObject() RETURN BINARY_INTEGER;
```

備考

このファンクションがコールされるまでの間は、他のどのような操作も実行できません。このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

FileNew

Microsoft Word ドキュメントを作成します。

構文

```
FUNCTION FileNew() RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

FileLoad

ドキュメントを Microsoft Word にロードします。

構文

```
FUNCTION FileLoad(filename VARCHAR2) RETURN BINARY_INTEGER;
```

引数	説明
filename	ドキュメントの完全修飾ファイル名。

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

FileSave

現行の Microsoft Word ドキュメントをディスクに保存します。

構文

```
FUNCTION FileSave() RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

FileSaveAs

現行の Microsoft Word ドキュメントを特定のファイルに保存します。

構文

```
FUNCTION FileSaveAs(filename VARCHAR2) RETURN BINARY_INTEGER;
```

引数	説明
filename	ドキュメントの完全修飾ファイル名。

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

FileClose

現行の Microsoft Word ドキュメントを閉じます。

構文

```
FUNCTION FileClose() RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

InsertText

現行の Microsoft Word ドキュメントにテキスト文字列を挿入します。

構文

```
FUNCTION InsertText(textstr VARCHAR2) RETURN BINARY_INTEGER;
```

引数	説明
textstr	ドキュメントに挿入されるテキスト。

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

InsertNewLine

現行の Microsoft Word ドキュメントに改行を挿入します。

構文

```
FUNCTION InsertNewLine() RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

FormatFontSize

現行の Microsoft Word ドキュメントに対してフォント・サイズを設定します。

構文

```
FUNCTION FormatFontSize(fontsize BINARY_INTEGER) RETURN BINARY_INTEGER;
```

引数	説明
fontsize	フォントのポイント・サイズ。

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

Microsoft Excel デモ

次の各項では、Microsoft Excel デモのインストール方法と、このデモで提供されている API について説明します。このデモは、Oracle と Microsoft Excel を使用して開発できるソリューションの例として提供されています。

Microsoft Excel デモは、次のものを提供します。

- ORDExcel。これは、Microsoft Excel を操作するいくつかの API を公開する PL/SQL パッケージです。このパッケージは、`excelsol.sql` スクリプトにより作成されます。
- `exceldem.sql`。これは、Oracle と Microsoft Word 間でのデータ交換機能を示すスクリプトです。このスクリプトは、Oracle の EMPLOYEES 表および JOBS 表から Microsoft Excel スプレッドシートにデータを挿入し、そのデータをグラフに描画します。これらの表は、サンプル・スキーマ内の人事スキーマにあります。

Microsoft Excel デモのインストール

このデモをインストールする前に、ローカル・コンピュータ上に Microsoft Excel がインストールされている必要があります。

Microsoft Excel デモをインストールするには、次のようにします。

1. SQL*Plus を起動します。

```
C:¥> sqlplus /NOLOG
```

2. Microsoft Excel デモを使用するユーザーのアカウントで、Oracle データベースのインスタンスに接続します。入力例は次のとおりです。

```
SQL> connect hr/hr
```

3. SQL*Plus プロンプトから、`excelsol.sql` スクリプトを実行します。

```
SQL> @%ORACLE_HOME%\com\demo\excelsol.sql;
```

このスクリプトにより、現行のユーザーのスキーマ内に ORDExcel パッケージが作成されます。このスクリプトを初めて実行するときに、次のエラーが数回発生します。

```
ORA-04043: オブジェクト XXXX は存在しません。
```

これらは正常なメッセージです。

Microsoft Excel デモの使用方法

Microsoft Excel デモを使用するには、次のようにします。

1. SQL*Plus プロンプトから、exceldem.sql スクリプトを実行します。

```
SQL> @%ORACLE_HOME%\com\demos\exceldem.sql;
```

このスクリプトにより、C:\ディレクトリに Microsoft Excel スプレッドシート (excelxxxxx.xls) が作成されます。このドキュメントには、EMPLOYEES 表および JOBS 表から取得されたデータが含まれています。これらの表は、サンプル・スキーマ内の人事スキーマにあります。

2. excelxxxxx.xls ファイルを開いて、その内容を確認します。xxxxxx はタイムスタンプです。

基本機能

次の各項では、Microsoft Excel デモで提供される API について説明します。これらの API は基本的な機能です。Microsoft Excel が COM Automation を通じて提供する機能は、これらの API を通じて提供されるもの以外にも多数存在します。

CreateExcelWorkSheet

Microsoft Excel COM Automation サーバーを起動し、ブックおよびワークシートに対応するオブジェクトをインスタンス化します。

構文

```
FUNCTION CreateExcelWorkSheet() RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

InsertData

現行の Excel ワークシートの特定のセルに、任意の種類のデータを挿入します。

構文

```
FUNCTION InsertData(range VARCHAR2, data ANY PL/SQL DATATYPE, datatype VARCHAR2)
RETURN BINARY_INTEGER;
```

引数	説明
range	現行の Excel ワークシート内の特定のセルを指定する文字列（たとえば「A1」、「B1」など）。

引数	説明
data	現行の Excel ワークシートに挿入するデータ。
datatype	Excel に挿入するデータのデータ型を表す文字列。使用可能なデータ型の一覧は、次のとおりです。 <ul style="list-style-type: none"> ■ I2 - 2 バイト整数 ■ I4 - 4 バイト整数 ■ R4 - IEEE 4 バイト実数 ■ R8 - IEEE 8 バイト実数 ■ SCODE - エラー・コード ■ CY - 通貨 ■ DISPATCH - ディスパッチ・ポインタ ■ BSTR - 文字列 ■ BOOL - ブール ■ DATE - 日付

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

InsertChart

特定範囲のデータのグラフを作成し、現行のワークシートの XY 位置にグラフを挿入します。グラフの高さと幅を任意に指定できます。

構文

```
FUNCTION InsertChart(xpos BINARY_INTEGER, ypos BINARY_INTEGER, width BINARY_INTEGER, height BINARY_INTEGER, range VARCHAR2, type VARCHAR2) RETURN BINARY_INTEGER;
```

引数	説明
xpos	グラフが挿入される現行のワークシート内の X 位置。
ypos	グラフが挿入される現行のワークシート内の Y 位置。
width	グラフの幅。
height	グラフの高さ。
range	グラフ化されるセルの範囲。

引数	説明
type	グラフ化されるデータのデータ型。

備考
このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

SaveExcelFile

現行の Microsoft Excel ブックを特定のファイルに保存します。

構文
FUNCTION SaveExcelFile(filename VARCHAR2) RETURN BINARY_INTEGER;

引数	説明
filename	Excel ブックの完全修飾ファイル名。

備考
このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

ExitExcel

クリーンアップを実行し、Excel COM Automation サーバーへの未解決の参照を破棄します。この API は最後にコールすることをお薦めします。

構文
FUNCTION ExitExcel() RETURN BINARY_INTEGER;

備考
このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

Microsoft PowerPoint デモ

次の各項では、Microsoft PowerPoint デモのインストール方法と、このデモで提供されている API について説明します。このデモは、Oracle と Microsoft PowerPoint を使用して開発できるソリューションの例として提供されています。

Microsoft PowerPoint デモは、次のものを提供します。

- ORDPPT。これは、Microsoft PowerPoint を操作するいくつかの API を公開する PL/SQL パッケージです。このパッケージは、pptsol.sql スクリプトにより作成されます。
- pptdem.sql。これは、Oracle と Microsoft PowerPoint 間でのデータ交換機能を示すスクリプトです。このスクリプトは、Oracle の EMPLOYEES 表および JOBS 表から Microsoft PowerPoint ドキュメントにデータを挿入します。これらの表は、サンプル・スキーマ内の人事スキーマにあります。

Microsoft PowerPoint デモのインストール

このデモをインストールする前に、ローカル・コンピュータ上に Microsoft PowerPoint 97 以上がインストールされている必要があります。

Microsoft PowerPoint デモをインストールするには、次のようにします。

1. SQL*Plus を起動します。

```
C:> sqlplus /NOLOG
```

2. Microsoft PowerPoint デモを使用するユーザーのアカウントで、Oracle データベースのインスタンスに接続します。入力例は次のとおりです。

```
SQL> connect hr/hr
```

3. SQL*Plus プロンプトから、pptsol.sql スクリプトを実行します。

```
SQL> @%ORACLE_HOME%\com%demos\pptsol.sql;
```

このスクリプトにより、現行のユーザーのスキーマ内に ORDPPT パッケージが作成されます。このスクリプトを初めて実行するときに、次のエラーが数回発生します。

```
ORA-04043: オブジェクト XXXX は存在しません。
```

これらは正常なメッセージです。

Microsoft PowerPoint デモの使用方法

Microsoft PowerPoint デモを実行するには、次のようにします。

1. SQL*Plus プロンプトから、pptdem.sql スクリプトを実行します。

```
SQL> @%ORACLE_HOME%\com\demo\pptdem.sql;
```

このスクリプトにより、C:\ディレクトリに Microsoft PowerPoint プレゼンテーション (pptdemo.ppt) が作成されます。ドキュメントには従業員名のリストが含まれます。

2. pptdemo.ppt を開いて、その内容を確認します。

基本機能

次の各項では、Microsoft PowerPoint デモで提供される API について説明します。これらの API は基本的な機能です。Microsoft PowerPoint が COM Automation を通じて提供する機能は、これらの API を通じて提供されるもの以外にも多数存在します。

CreatePresentation

Microsoft PowerPoint COM Automation サーバーを起動し、プレゼンテーションに対応するオブジェクトをインスタンス化します。

構文

```
FUNCTION CreatePresentation (servername IN VARCHAR2) RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

AddSlide

PowerPoint プレゼンテーションに新しいスライドを挿入します。

構文

```
FUNCTION AddSlide (layout IN BINARY_INTEGER) RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

SetTitle

PowerPoint スライドのタイトルを指定します。

構文

```
FUNCTION SetTitle (title IN VARCHAR2) RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

InsertText

スライド上の指定された位置にテキストを挿入します。

構文

```
FUNCTION InsertText (orientation IN BINARY_INTEGER, left IN BINARY_INTEGER, top IN BINARY_INTEGER, width IN BINARY_INTEGER, height IN BINARY_INTEGER, text IN VARCHAR2) RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

PresentationSave

現行の PowerPoint プレゼンテーションを保存します。

構文

```
FUNCTION PresentationSave RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

PresentationSaveAs

名前を指定して現行のプレゼンテーションを保存します。

構文

```
FUNCTION PresentationSaveAs (filename IN VARCHAR2) RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

PresentationClose

現行の PowerPoint プレゼンテーションを閉じます。

構文

```
FUNCTION PresentationClose RETURN binary_integer;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

Exit

PowerPoint プログラムを終了します。

構文

```
FUNCTION Exit RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

MAPI デモ

次の各項では、**Messaging Application Programming Interface (MAPI)** デモのインストール方法と、このデモで提供されている API について説明します。このデモは、Oracle と MAPI 準拠のアプリケーションを使用して開発できるソリューションの例として提供されています。

MAPI デモは、次のものを提供します。

- ORDMAPI。これは、Extended MAPI クライアントを操作するいくつかの API を公開する PL/SQL パッケージです。
- mapidem.sql。これは、Oracle と Extended MAPI クライアント間でのデータ交換機能を示すスクリプトです。

MAPI デモを使用する環境の設定

MAPI デモを使用するには、特定の関連アプリケーションを設定する必要があります。

注意： 次の設定には、Microsoft Outlook 2000 以上が必要です。Outlook Express は動作しません。

MAPI デモ用の環境を設定するには、次のようにします。

1. Exchange Server をインストールして次のように新規アカウントを作成します。
「スタート」→「プログラム」→「Microsoft Exchange」→「Active Directory ユーザーとコンピュータ」を選択します。
ドメインを選択し、フォルダを開きます。ユーザーを選択して右クリックし、新規ユーザーを作成します。
2. 次のように Microsoft Outlook をインストールします。
カスタム・インストールを選択します。「Collaboration Data Objects」を選択します（デフォルトではインストールされません）。
インストール時に「企業 / ワークグループ」オプションを選択します。
3. 次のように Microsoft Outlook を構成して接続情報を設定します。
Exchange Server で作成したアカウントを追加します。
受信メール・サーバーおよび送信メール・サーバーを入力し、アカウント名およびパスワードを入力します。
接続タイプを選択します（たとえば「LAN」など）。

4. Microsoft Outlook を、電子メール、ニュースグループおよびカレンダー・ツールのデフォルト・プログラムとして次のように設定します。

Internet Explorer で「ツール」メニュー→「インターネット オプション」→「プログラム」を選択し、各フィールドに入力します。

MAPI デモのインストールの準備

MAPI ソリューションは、Oracle データベース・サーバーにかわって Extended MAPI クライアントをコールします。Windows NT 上の Oracle データベース・サービスは、デフォルトでは NT システム・ユーザー LocalSystem として実行されます。ユーザー LocalSystem の MAPI プロファイルは簡単には構成できません。MAPI ソリューションを使用する前に、Windows NT ログオン・ユーザー・アカウントを使用して起動するよう Windows NT Oracle データベース・サービスと OracleHOME_NAMETNSListener サービスの両方を変更します。

MAPI デモのインストールを準備するには、次のようにします。

1. DOMAIN-1¥hr など、ローカル・ユーザー・アカウントまたはドメイン・ユーザー・アカウントを使用して Windows NT にログオンします。
2. MAPI サーバー（たとえば、Microsoft Outlook）を起動し、Windows NT ユーザー DOMAIN-1¥hr 用の MAPI プロファイルを構成します。このプロファイルを使用して電子メールを送信できることを確認します。
3. Windows NT の「コントロールパネル」の「サービス」に移動します。
4. OracleHOME_NAMETNSListener サービスを停止します。
5. OracleHOME_NAMETNSListener サービスを選択し、「スタートアップ」をクリックします。
6. 「ログオン」を「アカウント」に変更し、DOMAIN-1¥hr と入力します。
7. DOMAIN-1¥hr のパスワードを入力し、パスワードの確認入力を行います。
8. OracleHOME_NAMETNSListener サービスを再起動します。
9. Oracle データベース・サービスを停止します。
10. Oracle データベース・サービスを選択し、「スタートアップ」をクリックします。
11. 「ログオン」を「アカウント」に変更し、DOMAIN-1¥hr と入力します。
12. DOMAIN-1¥hr のパスワードを入力し、パスワードの確認入力を行います。
13. Oracle データベース・サービスを再起動します。

MAPI デモのインストール

このデモをインストールする前に、ローカル・コンピュータ上に **MAPI** アプリケーション (Microsoft Outlook 2000 以上など) がインストールされている必要があります。

MAPI デモをインストールするには、次のようにします。

1. SQL*Plus を起動します。

```
C:> sqlplus /NOLOG
```

2. MAPI デモを使用するユーザーのアカウントで、Oracle データベースのインスタンスに接続します。入力例は次のとおりです。

```
SQL> connect hr/hr
```

3. SQL*Plus プロンプトから、mapisol.sql スクリプトを実行します。

```
SQL> @%ORACLE_HOME%\com%demos\mapisol.sql;
```

このスクリプトにより、現行のユーザーのスキーマ内に ORDMAPI パッケージが作成されます。このスクリプトを初めて実行するときに、次のエラーが数回発生します。

```
ORA-04043: オブジェクト XXXX は存在しません。
```

これらは正常なメッセージです。

MAPI デモの使用方法

MAPI デモを使用するには、次のようにします。

1. テキスト・エディタで mapidem.sql を開き、ORDMapi.AddRecipient の電子メール・アドレス hr@us.oracle.com を自身の電子メール・アドレスに変更します。変更を保存します。
2. SQL*Plus プロンプトから、mapidem.sql スクリプトを実行します。

```
SQL> @%ORACLE_HOME%\com%demos\mapidem.sql;
```

このスクリプトはデータベース・サーバーに接続し、データを抽出し、指定された受信者に電子メールを送信します。

基本機能

次の各項では、MAPI デモで提供される API について説明します。これらの API は基本的な機能です。MAPI が COM Automation を通じて提供する機能は、これらの API を通じて提供されるもの以外にも多数存在します。

CreateMAPISession

MAPI COM Automation サーバーを起動し、セッションのオブジェクトをインスタンス化します。

構文

```
FUNCTION CreateMAPISession (servername IN VARCHAR2 DEFAULT '', profilename IN  
VARCHAR2 DEFAULT NULL, password IN VARCHAR2 DEFAULT NULL) RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

NewMessage

新しいメッセージを作成します。

構文

```
FUNCTION NewMessage RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

AddRecipient

受信者の電子メール・アドレスを追加します。このアドレスは、電子メール・メッセージの送信先のアドレスです。

構文

```
FUNCTION AddRecipient (emailaddress VARCHAR2) RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

SetSubject

電子メール・メッセージの件名を指定します。

構文

```
FUNCTION SetSubject (subject VARCHAR2) RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

SetBody

電子メール・メッセージの本文テキストを挿入します。

構文

```
FUNCTION SetBody (messagetext VARCHAR2) RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

SendMessage

指定された受信者に電子メール・メッセージを送信します。

構文

```
FUNCTION SendMessage RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

EndMAPISession

MAPI セッションを終了します。

構文

```
FUNCTION EndMAPISession RETURN BINARY_INTEGER;
```

備考

このプロシージャは、成功のときは 0 を返し、エラーが発生したときは 0 以外の HRESULT を返します。

Oracle COM Automation Java デモ

この章では、Java 用の Oracle COM Automation 用に設計されたデモ・プログラムの使用方法を説明します。

この章の項目は次のとおりです。

- [Oracle COM Automation Java デモの概要](#)
- [Microsoft Word Java デモ](#)

Oracle COM Automation Java デモの概要

Java 用の Oracle COM Automation には、Oracle COM Automation を使用してソリューションを開発する方法を示す例が含まれています。このデモは基本機能を提供し、COM Automation を使用するカスタマイズされた複雑なアプリケーションを作成する基礎として機能します。このデモは、サンプル・スキーマで使用可能な人事スキーマに基づいています。

デモでは、COM Automation を使用して簡単な操作を行うための基本的な API のセットを提供しています。それぞれの COM Automation サーバー（Word や Excel など）は、デモ API を通じて提供されるものよりもさらに高度な機能を提供します。これらの高度な機能を利用するためには、独自の Java クラスを設計し、コーディングする必要があります。

このリリースでは、オラクル社は次のデモを提供しています。

- **Microsoft Word Java デモ**： Oracle から Word にデータを挿入します。

Microsoft Word Java デモ

次の各項では、Microsoft Word Java デモのインストール方法と、このデモで提供されている API について説明します。このデモは、Oracle と Microsoft Word を使用して開発できるソリューションの例として提供されています。

デモでは、データベース内の従業員の名前を含む Microsoft Word ドキュメントを作成します。

Microsoft Word Java デモは、`%ORACLE_HOME%\com\java\demos` ディレクトリにインストールされ、次のものを提供します。

- デモの Java ソースである `TestWORD.java`。API のセットに加え、これにはデモ・プログラム `test` が含まれます。
- デモの Java クラスである `TestWORD.class`。
- デモのコール仕様を作成するスクリプトである `TestWORD.sql`。

Microsoft Word Java デモのインストール

このデモをインストールする前に、ローカル・コンピュータ上に Microsoft Word がインストールされている必要があります。

デモをインストールするには、次のようにします。

1. コマンドラインで `loadjava` を実行します。

```
loadjava -force -resolve -user hr/hr %ORACLE_HOME%\com\java\demos\TestWORD.class
```

2. SQL*Plus を起動します。

```
C:> sqlplus /NOLOG
```

3. Microsoft Word デモを使用するユーザーのアカウントで、Oracle データベースのインスタンスに接続します。入力例は次のとおりです。

```
SQL> connect hr/hr
```

4. TestWORD.sql を実行し、コール仕様を作成します。

```
SQL> @%ORACLE_HOME%\com\java\demos\TestWORD.sql
```

関連資料：

- loadjava ツールの詳細は、『Oracle9i Java Developer's Guide』を参照してください。
- コール仕様の詳細は、『Oracle9i Java Stored Procedures Developer's Guide』を参照してください。

Microsoft Word Java デモの使用方法

Word デモを使用するには、次のようにします。

1. SQL*Plus プロンプトで、SERVEROUTPUT を設定します。

```
SQL> SET SERVEROUTPUT ON
```

2. SQL*Plus プロンプトで、TestWORD() をコールします。

```
SQL> CALL TestWORD();
```

このスクリプトにより、C:\ディレクトリに Microsoft Word ドキュメント (worddemoj.doc) が作成されます。このドキュメントには、EMPLOYEES 表および JOBS 表から取得されたデータが含まれています。これらの表は、サンプル・スキーマ内の人事スキーマにあります。

3. worddemoj.doc を開いて、その内容を確認します。

カスタム・アプリケーションの作成

5-4 ページの「基本機能」で説明する public class TestWORD は、Word.Basic COM Automation クラスのラッパーの他に、ラッパーの使用法を示すサンプル・コードも提供します。このコードは、Oracle データベース・サーバーで実行するために作成されました。

このラッパーを使用するカスタム・アプリケーションを作成するには、次のようにします。

1. このクラスのオブジェクトをインスタンス化します。
2. メソッド CreateWordObject をコールすることにより、Word.Basic オブジェクトを作成します。
3. メソッド FileNew で新規の Microsoft Word ドキュメントを作成するか、メソッド FileLoad で既存のドキュメントを開きます。

- 4. メソッド `FormatFontSize`、`InsertText` および `InsertNewLine` を使用して、テキストと書式をドキュメントに追加します。
- 5. `FileSaveAs` または `FileSave` でドキュメントを保存します。
- 6. ドキュメントの終了時に、メソッド `FileClose` をコールします。
- 7. `Word.Basic` オブジェクトの終了時に、メソッド `DestroyWordObject` をコールします。

基本機能

次の各項では、Microsoft Word Java デモで提供される API について説明します。これらの API は基本的な機能です。Microsoft Word が COM Automation を通じて提供する機能は、これらの API を通じて提供されるもの以外にも多数存在します。

TestWORD

コンストラクタ。これは何も行いません。

構文

```
public TestWORD()
```

CreateWordObject

Word.Basic COM オブジェクトを作成します。

構文

```
public void CreateWordObject(java.lang.String servername)
```

引数	説明
servername	COM オブジェクトを作成するサーバー。ローカル・サーバーに対しては null または空の文字列を指定します。

DestroyWordObject

Word.Basic COM オブジェクトを破棄します。

構文

```
public void DestroyWordObject()
```

FileNew

Microsoft Word ドキュメントを作成します。

構文

```
public void FileNew()
```

備考

Word.Basic COM オブジェクトの COM メソッド FileNewDefault のラッパー。

FileLoad

既存の Microsoft Word ドキュメントをロードします。

構文

```
public void FileLoad(java.lang.String filename)
```

引数	説明
filename	ロードするファイルの名前。

備考

Word.Basic COM オブジェクトの COM メソッド FileOpen のラッパー。

FormatFontSize

フォント・サイズを設定します。

構文

```
public void FormatFontSize(long fontsize)
```

引数	説明
fontsize	新しいフォント・サイズ。

備考

Word.Basic COM オブジェクトの COM メソッド FormatFont のラッパー。

InsertText

Microsoft Word ドキュメントにテキストを挿入します。

構文

```
public void InsertText(java.lang.String textstr)
```

引数	説明
textstr	挿入するテキスト。

備考

Word.Basic COM オブジェクトの COM メソッド Insert のラッパー。

InsertNewLine

Microsoft Word ドキュメントに改行を挿入します。

構文

```
public void InsertNewLine()
```

備考

Word.Basic COM オブジェクトの COM メソッド InsertPara のラッパー。

FileSaveAs

指定された名前を使用して Microsoft Word ドキュメントを保存します。

構文

```
public void FileSaveAs(java.lang.String filename)
```

引数	説明
filename	ファイルの名前。

備考

Word.Basic COM オブジェクトの COM メソッド FileSaveAs のラッパー。

FileSave

Microsoft Word ドキュメントを保存します。

構文

```
public void FileSave()
```

備考

Word.Basic COM オブジェクトの COM メソッド FileSave のラッパー。

FileClose

Microsoft Word ドキュメントを閉じ、Microsoft Word を終了します。

構文

```
public void FileClose()
```

備考

Word.Basic COM オブジェクトの COM メソッド FileClose および FileExit のラッパー。

COM Automation エラー・メッセージ

この付録の項目は次のとおりです。

- [Oracle COM Automation PL/SQL エラー](#)
- [Microsoft COM Automation のエラー](#)

Oracle COM Automation PL/SQL エラー

次に示すのは、Oracle COM Automation PL/SQL エラーの一覧と一般的な原因です。

COM-0001: BOOLEAN 型ではありません

処置: 変数が適切なデータ型であることを確認します。

COM-0002: トークンが無効であるか、またはトークンのインタフェースがありません

処置: インタフェースが存在することを確認します。

COM-0003: オブジェクトの数が最大数に達しています

処置: 使用済のオブジェクトは、[DestroyObject](#) をコールして破棄するようにします。

COM-0004: ProgID に登録された CLSID は無効です。

処置: 指定された ProgID の COM コンポーネントが登録されていることを確認します。

COM-0005: CLSID をレジストリへ書き込み中にエラーが発生しました。

処置: レジストリが書き込み可能であることと、破損していないことを確認します。

COM-0006: 指定されたクラスは登録データベースに登録されていません

処置: クラスが登録されていることを確認します。

COM-0007: COM Automation オブジェクトを初期化できませんでした

処置: オブジェクトが COM オートメーション・オブジェクトとして登録されていることを確認します。

COM-0008: インタフェースがサポートされていません

処置: 指定されたインタフェースが有効であることを確認します。

COM-0009: タイプ情報の件数を取得できませんでした

処置: オブジェクトが正しく登録されていることを確認します。

COM-0010: タイプ情報の実装はサポートされていません

処置: オブジェクトが正しく登録されていることを確認します。

COM-0011: タイプ情報を取得できませんでした

処置: オブジェクトが正しく登録されていることを確認します。

COM-0012: タイプ属性を取得できませんでした

処置: オブジェクトが正しく登録されていることを確認します。

COM-0013: 索引でファンクションの説明を取得できませんでした。

処置: オブジェクトが正しく登録されていることを確認します。

COM-0014: 起動できません

処置: メソッド名がそのオブジェクトで有効であることを確認します。

COM-0015: パラメータ・カウントが正しくありません

処置: メソッドのパラメータ数がカウントと等しいことを確認します。

COM-0016: rgvarg で引数の 1 つに無効な VARIANT 型があります。

処置: オブジェクトが正しく登録されていることを確認します。

COM-0017: アプリケーションで例外を呼ぶ必要があります。pexcepinfo に渡される構造体には値がすべて入っている必要があります。

処置: pexcepinfo の構造体が初期化されていることを確認します。

COM-0018: 要求されたメンバーが存在しないか、または Invoke の呼出しで読み込み専用プロパティの値を設定しようとしています。

処置: プロパティ値が書き込めること、およびメンバーが存在することを確認します。

COM-0019: Idispatch のこの実装では名前付き引数をサポートしていません。

処置: 名前付き引数は使用しないでください。標準のパラメータ受渡し方法を使用します。

COM-0020: rgvarg の引数の 1 つに、指定タイプへ強制変換できなかったものがあります。

処置: 変換される引数のデータ型に互換性があることを確認します。

COM-0021: パラメータ・ディスパッチ ID の 1 つにメソッドのパラメータと対応しないものがあります。

処置: 引数が正しく渡されることを確認します。

COM-0022: 1 つまたは複数の引数を強制変換できませんでした。

処置: 引数に互換性があることを確認します。

COM-0023: riid に渡されたインタフェース ID は IID_NULL ではありません。

処置: 渡されるインタフェース ID が IID_NULL であることを確認します。

COM-0024: 呼び出されたメンバーでは認識されていない locale ID (LCID) に従って STRING 引数を解釈します。

処置: localeID が正しいことを確認します。

COM-0025: オプション・パラメータではありません

処置: 引数カウントが、渡されるパラメータ数に対応していることを確認します。

COM-0026: 最大許容文字数を越えた名前

処置: 入力する名前の文字数は 1024 文字以下です。

COM-0027: このクラスは集合体の一部として作成できません。

処置: このクラスを集合体の一部として作成しないでください。

Microsoft COM Automation のエラー

次に示すのは、Microsoft COM Automation エラーの一覧と一般的な原因です。エラー・コードは、16 進および 2 進の両方でリストされています。

(0x800401f3) (-2147221005) クラス文字列が不正です。

原因: 指定された progID または CLSID が、ローカル・コンピュータのレジストリに COM オブジェクトとして登録されていません。

(0x8007007e) (-2147024770) 指定されたモジュールが見つかりませんでした

原因: 指定された COM オブジェクトはインプロセス COM サーバー (DLL ファイル) として登録されていますが、DLL ファイルが見つからないかロードされていません。

(0x80020004) (-2147352572) パラメータが見つかりません

原因: 名前付きパラメータが位置パラメータの前に指定されました。

処置: すべての名前付きパラメータを、位置パラメータの後に指定します。

(0x80020005) (-2147352571) 型が一致しません。

原因: プロパティ値またはメソッドの戻り値を格納するために使用する PL/SQL ローカル変数のデータ型が、プロパティ値またはメソッドの戻り値の Visual Basic データ型と一致しませんでした。または、プロパティまたはメソッドの戻り値が要求されましたが、そのプロパティまたはメソッドは戻り値を返しません。

処置: ローカル変数を適切なデータ型にします。

(0x80020006) (-2147352570) 名前が定義されていません。

原因: 指定されたプロパティ名またはメソッド名が、指定されたオブジェクトに見つかりませんでした。

(0x80020008) (-2147352568) 変数型が不正です。

原因: メソッド・パラメータとして渡された PL/SQL または Java 値のデータ型と、メソッド・パラメータの COM オートメーション・データ型が一致しないか、またはメソッド・パラメータとして NULL 値が渡されました。

処置: メソッド・パラメータとして使用しているローカル変数を適切なデータ型にし、さらに NULL 以外の値を設定するようにします。

(0x80080005) (-2146959355) サーバーの実行に失敗しました。

原因: 指定された COM オブジェクトはローカル COM サーバー (.EXE ファイル) として登録されていますが、.EXE ファイルが見つからないか、起動できません。

用語集

Component Object Model (COM)

オブジェクト間通信を、個々のオブジェクトの記述に使用されるプログラム言語の種類に関係なく実現するためのバイナリ標準。

Distributed Component Object Model (DCOM)

オブジェクトがネットワークを介して他のオブジェクトと対話できるようにするための COM 拡張。

Dynamic Link Library (DLL)

必要に応じて Windows アプリケーションがロードできる実行可能ファイル。

GUID

COM オブジェクトを一意に識別する識別子。GUID は Globally Unique Identifier の略称。

IID

COM インタフェースを識別する [GUID](#)。

listener.ora

サーバー上の 1 つまたは複数の Transparent Network Substrate (TNS) リスナーが記述されている構成ファイル。

Messaging Application Programming Interface (MAPI)

MAPI は一連の共通 API で構成されるメッセージ・アーキテクチャであり、様々なハードウェア・プラットフォーム間で、複数のアプリケーションが複数のメッセージ・システムを使用して対話することを可能にする。

Optimal Flexible Architecture (OFA)

Oracle ソフトウェアおよびデータベースのファイルのネーミングおよび配置に関する一連のガイドライン。

Oracle COM Automation

PL/SQL 開発者が、COM オートメーション・インタフェース (IDispatch) を通じて COM オブジェクトをプログラム上で操作するための Oracle の機能の 1 つ。

Oracle Net

Oracle クライアント / サーバー通信ソフトウェア。任意のネットワーク・プロトコルおよびオペレーティング・システムを介して、Oracle ツールまたはデータベースへの透過的操作を提供する。

PL/SQL

SQL を拡張した、オラクル社の手続き型言語。

progid

GUID にマップされる記述文字列。

tnsnames.ora

ネット・サービス名にマップされる接続記述子を含むファイル。すべてのクライアントまたは各クライアントで使用するために、このファイルを集中してメンテナンスすることも、ローカルでメンテナンスすることもできる。

外部プロシージャ (external procedure)

C などの第 3 世代言語 (3GL) で書かれた関数。PL/SQL ファンクションまたはプロシージャと同様に、PL/SQL や SQL 内でコールできる。

リスナー (listener)

クライアント・アプリケーションからの着信接続要求をリスニングして受け入れるサーバー・プロセス。Oracle リスナー・プロセスにより、クライアントとの後続の通信を処理する Oracle データベース・プロセスが起動する。

索引

A

API

- Java, 3-21
- PL/SQL, 3-9

Application Program Interface (API), 3-8

- Java, 3-21
- PL/SQL, 3-9

Automation

- Java API, 3-23

C

COM Automation

- PL/SQL エラー, A-2
- コール, 1-6

comwrap.sql, 2-2, 2-4

COM オブジェクト

- 必要な情報, 3-6
- 表示, 3-7
- プログラム ID, 3-6
- プロパティとメソッド, 3-6

Create

- Java API, 3-24

CreateObject

- PL/SQL API, 3-10

Currency

- Java API, 3-32

D

DCOM

- 構成, 2-7

Destroy

- Java API, 3-25

DestroyObject

- PL/SQL API, 3-11

E

Exchange Server, 4-15

EXTPROC

- extproc.exe, 1-6

G

GetArg

- PL/SQL API, 3-17

GetLastError

- PL/SQL API, 3-12

GetProperty

- Java API, 3-26
- PL/SQL API, 3-13

grant.sql, 2-4

GUID (Globally Unique Identifier), 3-6

H

HRESULT

- リターン・コード, 3-4

I

IDispatch インタフェース, 3-10, 3-21, 3-23

IDL, 3-6

InitArg

- Java API, 3-28
- PL/SQL API, 3-16

InitOutArg

- PL/SQL API, 3-16

Internet Explorer, 4-16

Invoke

Java API, 3-30

PL/SQL API, 3-20

J

Java

構成, 2-4

Java API, 3-21

Create, 3-24

Destroy, 3-25

GetProperty, 3-26

InitArg, 3-28

Invoke, 3-30

SetArg, 3-29

SetProperty, 3-27

Java Automation コンストラクタ, 3-23

Java Currency コンストラクタ, 3-32

Java コンポーネント, 2-2

L

loadjava, 5-2

M

MAPI

PL/SQL の例, 4-15

デモ・スクリプト mapidem.sql, 4-15

MAPI デモ, 4-16

Messaging Application Programming Interface,

「MAPI」を参照

Microsoft Excel

PL/SQL の例, 4-7

デモ・スクリプト exceldem.sql, 4-7

Microsoft Outlook, 4-16

Microsoft PowerPoint

PL/SQL の例, 4-11

デモ・スクリプト pptdem.sql, 4-11

Microsoft Word

デモ・スクリプト worddem.sql, 4-2

例, 4-2

Microsoft Word Java デモ, 5-2

Microsoft Word Java デモのインストール, 5-2

Microsoft Word デモの使用方法, 5-3

O

OLE-COM オブジェクト ビューア, 3-7

ORA-28575 エラー・メッセージ, 2-6

Oracle COM Automation

Java デモ, 5-2

アーキテクチャ, 1-4

インストール, 2-2

概要, 1-2

基本機能, 1-2

構成, 2-4

コンポーネント, 2-2

デモ, 4-2

利点, 1-3

Oracle8i から Oracle9i への移行, 2-3

orawcom.dll, 2-2

orawpcom.dll, 1-6, 2-2

ORDEExcel

PL/SQL パッケージ, 4-7

ORDMAPI

PL/SQL パッケージ, 4-15

ORDPPT

PL/SQL パッケージ, 4-11

ORDWord

PL/SQL パッケージ, 4-2

Outlook クライアント, 4-15

P

PL/SQL

ORDEExcel パッケージ, 4-7

ORDMAPI パッケージ, 4-15

ORDPPT パッケージ, 4-11

ORDWord パッケージ, 4-2

構成, 2-4

PL/SQL API

CreateObject, 3-10

DestroyObject, 3-11

GetArg, 3-17

GetLastError, 3-12

GetProperty, 3-13

InitArg, 3-16

InitOutArg, 3-16

Invoke, 3-20

SetArg, 3-18

SetProperty, 3-14

PL/SQL MAPI デモのインストール

準備, 4-16
PL/SQL コンポーネント, 2-2
PL/SQL のアーキテクチャ, 1-5
progID
COM オブジェクト, 3-6

S

SERVEROUTPUT, 5-3
SetArg
Java API, 3-29
PL/SQL API, 3-18
SetProperty
Java API, 3-27
PL/SQL API, 3-14

T

TestWORD, 5-3
TestWORD.class, 5-2
TestWORD.java, 5-2
TestWORD.sql, 5-2, 5-3

W

wordemoj.doc, 5-3

あ

アーキテクチャ
Java, 1-6
Oracle COM Automation, 1-4
PL/SQL, 1-5

い

インストール
Oracle COM Automation, 2-2
インストールの準備, 4-16
インタフェース定義言語, 3-6

え

エラー
HRESULT, 3-4
Microsoft COM Automation, A-4
Oracle COM Automation, A-2

コード, 3-4
メッセージ, A-4

き

基本機能
Oracle COM Automation, 1-2

こ

構成
DCOM, 2-7
Java, 2-4
Oracle COM Automation, 2-4
PL/SQL, 2-4
PL/SQL 用の Oracle COM Automation のリスナー,
2-5
コール仕様, 5-3
コンストラクタ, 3-23, 3-32

さ

サンプル・スキーマ, 4-2, 5-2

し

システム要件
Oracle COM Automation, 2-3

て

データ型
Java から COM オートメーション・データ型へ,
3-3
PL/SQL から COM オートメーション・データ型
へ, 3-2
変換, 3-2, 3-3
デモ
Java 用の Oracle COM Automation, 5-2
MAPI, 4-15
MAPI デモのインストール, 4-17
Microsoft Excel, 4-7
Microsoft Excel デモのインストール, 4-7
Microsoft PowerPoint, 4-11
Microsoft PowerPoint デモのインストール, 4-11
Microsoft Word, 4-2
Microsoft Word デモのインストール, 4-3

Oracle COM Automation, 4-2
PL/SQL, 4-2

と

トラブルシューティング
Oracle COM Automation PL/SQL エラー, A-2

ふ

プログラム ID
COM オブジェクト, 3-6

り

リスナー
Oracle COM Automation の構成, 2-5
リターン・コード
HRESULT, 3-4
利点
Oracle COM Automation, 1-3

れ

例
MAPI, 4-15
Microsoft Excel, 4-7
Microsoft PowerPoint, 4-11
Microsoft Word, 4-2
Microsoft Word Java, 5-2