

# Introduction to 4D for OCI

By Yvan Ayaay, Technical Support Engineer, 4D Inc.

TN 05-39

## Introduction

---

**4D for OCI** プラグインによって **4D** は **Oracle** サーバと通信できるようになります。このプラグインは、**Oracle** サーバへのネイティブアクセスを可能にする **Oracle** の **API (Application Program Interface)**である **Oracle Call Interface (OCI)**を使用しています。利用すれば、**4th Dimension** または **4D Server** によるどんなアプリケーションでも **Oracle** クライアントとして動作させることができ、**Oracle** データベースのデータにアクセスできるようになります。この **Technical Note** では、**4D for OCI** の基本的な仕組みや簡単な操作方法について解説しています。

## Overview

---

**Oracle Call Interface (OCI)**は、ひとつ以上の **Oracle Server** に対してアプリケーションから働きかけるために使用される一種の **application programming interface (API)**です。**OCI** では、標準的なデータベースアクセスやデータ取得のための機能がダイナミックルーチンライブラリの形式で提供されています。**4D for OCI** プラグインが動作するためには、まずこのライブラリをインストールする必要があります。ライブラリを使用すれば **Oracle** データベースに対するネイティブコネクティビティが実現し、**Oracle** データベースで様々なデータベースオペレーションが実行できるようになります。**OCI** ライブラリは、コンテキスト情報、接続情報、バインド情報、データ情報、エラー情報をストアするためのハンドルを使用します。**OCI** アプリケーションは、それらのハンドルによって収められている特定の情報にアクセスします。**OCI** では **SQL** を実行したり、クエリの結果を処理したりストアすることができます。

## Installation and Configuration

---

**4D for OCI** は、**Oracle 8.16**、**Oracle 9**、**Oracle 10** データベースをサポートしています。利用には、**OCI** を実行するためのライブラリファイルが必要です。**Oracle Technology Network Web** から **Oracle Database Client Software** をダウンロードすることによってファイルをインストールすることができます。<http://www.oracle.com/technology/software/>からインストールしたい製品のページを選択して下さい。例えば、使用 **OS** 用の **Oracle Database 10g** が必要であれば、そのダウンロードリンクを選んで下さい。

## OCI Library Installation for Windows

---

Windows で Oracle Client をインストールするのであれば、Oracle Universal Installer を使用すると便利です。ファイルをダウンロードして展開した後、Setup.exe ファイルをクリックして Oracle Universal Installer を実行すれば、あとは画面の指示に従ってインストールを進めることができます。本文では Administrator タイプをインストールタイプとして選択しています。以下はインストールガイドのリンクです。

[http://download-east.oracle.com/docs/cd/B19306\\_01/install.102/b14312/install.htm#BABJGGJH](http://download-east.oracle.com/docs/cd/B19306_01/install.102/b14312/install.htm#BABJGGJH)

OCI ライブラリは、Oracle Client と一緒にインストールされます。インストールに続いて、tnsnames.ora ファイルを設定して Oracle データベースに対する接続を正しく設定して下さい。このファイルは、Oracle Client と一緒にインストールされる Net Configuration Assistant プログラムを使用して設定することができます。Oracle データベースの名前、プロトコル、データベースサーバの IP アドレスを入力すると、次の場所に tnsnames.ora ファイルが作られます。

ORACLE\_BASE¥ORACLE\_HOME¥network¥admin¥tnsnames.ora

ファイルをテキストエディタなどで作成して、上記の場所に配置することもできます。それぞれのエントリーは次のフォーマットに倣っている必要があります。

```
# tnsnames.ora Network Configuration File:
C:¥oracle¥product¥10.2.0¥client_1¥network¥admin¥tnsnames.ora
# Generated by Oracle configuration tools.
ORACLE4D =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 10.64.0.20)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = oracle4d)
    )
  )
```

クライアントがインストールされると 4D for OCI が使用できるようになります。

## OCI Library Installation for Mac OS 10.3.x

---

Mac OS 10.3.9 に OCI Library をインストールする場合、Oracle 10g Instant Client for Mac

OS Xを使用することができます。**Instant Client** は次の場所からダウンロードして下さい。

<http://www.oracle.com/technology/software/tech/oci/instantclient/htdocs/macsoft.html>

ファイル名は **instantclient-basic-macosx-10.1.0.3.zip** です。

以下は **Mac OS X** におけるインストール手順の一例です。

**NOTE:**ターミナルコマンドは大文字と小文字を区別するので注意して下さい。

**1) Oracle 10g Instant Client** を正しい場所にインストールします。フォルダ名は **/Oracle**、ファイルのオーナーは **root** である必要があります。

**instantclient-basic-macosx-10.1.0.3.zip** をデスクトップに展開すると **instantclient10\_1** フォルダが作成されます。

ターミナルウィンドウを開きます。

**sudo sh** コマンドを実行します。

パスワードを要求されたら入力します。**sudo** アクセスがないとインストールは完了できません。

**mkdir /Oracle** コマンドを実行します。

**cp** とタイプし、デスクトップから **instantclient10\_1** をターミナルにドラッグします。

バックスペースを押して **/\* /Oracle** とタイプします。この時点でコマンドラインは次のようになっているはずです。

**sh-2.05b# cp <your user home>/Desktop/instantclient10\_1/\* /Oracle**

リターンキーを押します。すべての **Instant Client** ファイルが **/Oracle** フォルダにコピーされます。

**2) 必要な Oracle** ライブラリを正しい場所にコピーします。

**cp /Oracle/libclntsh.dylib.10.1 /usr/lib/libclntsh.dylib** コマンドを実行します。

**chmod 777 /usr/lib/libclntsh.dylib** コマンドを実行します。

**exit** コマンドを実行します。これによって **root** 権限ではなくなります。

**3) 4D for OCI** を使用するユーザのために **environment.plist** を作成/更新します。

**cd ~** コマンドを実行します。これによってホームディレクトリに移動します。

**mkdir .MacOSX** コマンドを実行します。ピリオドを省かないで下さい。

**environment.plist** ファイルは、**XML** テキストである **Macintosh** プロパティファイルです。**Mac OS X** のプロパティリストエディタを使用して編集することができ、テキストエディタ、あるいはターミナルの **vi** を使用しても作成することができます。いずれにしても、内容は次のようになるはずです。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>DYLD_LIBRARY_PATH</key>
<string>/Oracle</string>
<key>ORACLE_HOME</key>
<string>/Oracle/OCI/oci</string>
<key>ORA_NLS33</key>
<string>/Oracle/OCI/oci/ocommon/nls/admin/data</string>
<key>TNS_ADMIN</key>
<string>/Oracle/OCI/oci/network/admin</string>
</dict>
</plist>
```

**.MacOSX** フォルダ以外に作成した場合は、移動して下さい。**Finder** はピリオドで始まるファイル名を表示しないので、ターミナルを使用します。ターミナルでファイルをコピーするには次のようにします。

```
cp <folder that contains>/environment.plist ~/.MacOSX
```

一旦、ログアウトして再びログインします。

#### NOTE:

**Instant Client** ファイルを **/Oracle** にインストールしない場合、**4D for OCI** のコマンドである **OCISetEnv** を使用して **ORACLE\_HOME**、**ORA\_NLS33**、**TNS\_ADMIN** の値を設定する必要があります。**4D for OCI** のデフォルト設定値は次のとおりです。

```
ORACLE_HOME=/Oracle/OCI/oci
ORA_NLS33=/Oracle/OCI/oci/ocommon/nls/admin/data
TNS_ADMIN=/Oracle/OCI/oci/network/admin
```

**Windows** の場合と同じように、**TNSNAMES.ORA** ファイルを作成して **Oracle** マシンの **IP** アドレスを設定する必要があります。

1. テキストエディタで **TNSNAMES.ORA** ファイルを作成して次のエントリーを入力します。

```
oracle4d =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (ADDRESS = (PROTOCOL = TCP)(HOST = 10.96.0.61)(POST = 1521))  
    )  
    (CONNECT_DATA =  
      (SERVICE_NAME = oracle4d)  
    )  
  )  
)
```

**HOST** パラメータには **Oracle** サーバの正しい **IP** アドレスを入力して下さい。

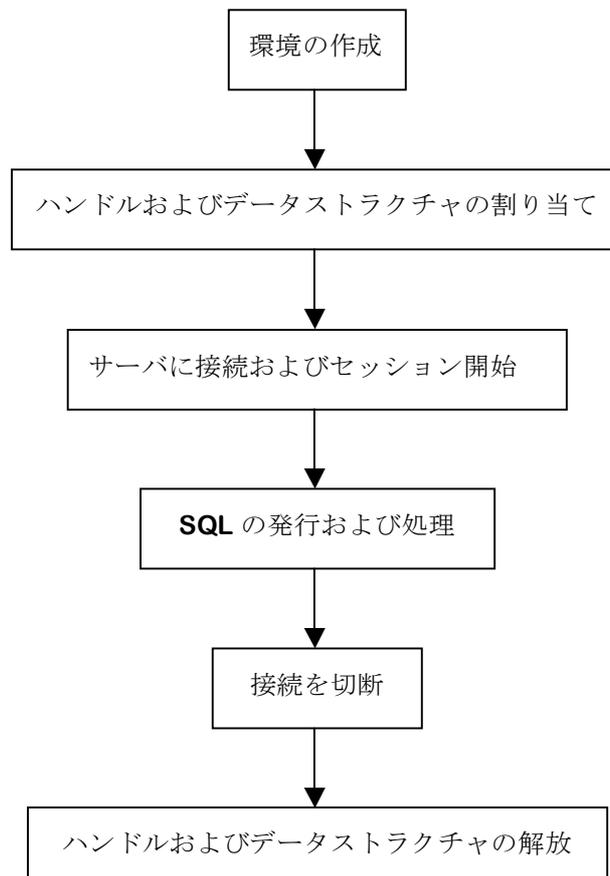
2. ターミナルウィンドウを開いてファイルを **/private/etc** ディレクトリにコピーします。

クライアントがインストールされると **4D for OCI** が使用できるようになります。

**NOTE:** デザインモードのエクスプローラで **4D for OCI** が「使用しない」になっている場合、**OCI** ライブラリが正しくインストールされていない可能性があります。

## Basic OCI Program Structure

**OCI** アプリケーションは次のようなプログラムストラクチャになっています。**OCI** アプリケーションを開発する場合もこの流れに倣うようにして下さい。



以下は **4D** で **OCI** アプリケーションを開発する際におけるそれぞれのステップです。

### Step1: OCI プログラミング環境およびスレッドの作成と初期化

**OCI** アプリケーションの冒頭で **OCI** 環境を作成する必要があります。すべての **OCI** 機能は、このコンテキストで実行されます。他の **OCI** コールをする前に **4D for OCI** コマンドの **OCIEnvCreate()** を呼び出さなくてはなりません。これにより、**OCI** 環境とそのハンドルが初期化されます。他のハンドルタイプは、このハンドルを使用して割り当てられます。以下はコマンドの使用例です。

```
$error_!:=OCIEnvCreate ($Envhp;OCI_HTYPE_ENV)
```

### Step2: ハンドルおよび記述子の割り当て

大抵の **OCI** コールはハンドルをパラメータとして使用します。ハンドルは、基本的にいって **OCI** ライブラリによって割り当てられたソトレージエリアへのポインタです。コンテキスト、接続、データ、**OCI** ファンクションコールに関する情報は、ハンドルで知ることができます。割り当てる必要があるハンドルには、サービスコンテキストハンドル、サーバハンドル、セッションハンドル、ステートメントハンドル、エラーハンドルがあります。

サービスコンテキストハンドルは、サーバに対する **OCI** コールの操作コンテキスト属性を定義するもので、サーバ、セッション、トランザクションハンドルで構成されています。サーバハンドルは、データベースサーバとの接続を特定します。セッションハンドルは、ユーザの立場と権限、操作コンテキストを特定します。トランザクションハンドルは、**SQL** が実行されるトランザクションを定義します。ステートメントハンドルは、**SQL** ステートメントとその属性を定義します。**OCI** コールのパラメータとして渡されるエラーハンドルには、**OCI** で起きたエラーに関する情報が返されます。この情報は **OCIErrorGet()** コマンドで知ることができます。以下はそれぞれのハンドルの割り当ての例です。

```
$error_!:=OCIHandleAlloc ($Envhp;$svchp;OCI_HTYPE_SVCCTX) `Service context handle  
$error_!:=OCIHandleAlloc ($Envhp;$authp;OCI_HTYPE_SESSION) `Session handle  
$error_!:=OCIHandleAlloc ($Envhp;$srvhp;OCI_HTYPE_SERVER) `Server handle  
$error_!:=OCIHandleAlloc ($Envhp;$stmthp;OCI_HTYPE_STMT) `Statement Handle  
$error_!:=OCIHandleAlloc ($Envhp;$errhp;OCI_HTYPE_ERROR) `Error handle
```

### Step3: サーバ接続およびユーザセッションの確立

**OCI** 環境が初期化されてハンドルが割り当てられたならば、次のステップはサーバ接続を確立してユーザセッションを開始することです。これにはふたつのオプションが用意されています。シングルユーザ/セッション、またはマルチプルユーザ/セッションです。

それぞれのデータベース接続に対して単一のユーザセッションだけが必要なのであれば、簡略化されたログインコール **OCILogon()**を使用することができます。次のコードに示されているように環境、エラー、サービスコンテキストハンドルがデータベースアクセスパラメータと一緒に渡されます。

```
$error_1:=OCILogon ($Envhp;$errhp;$svchp;"scott";"tiger";"oracle4d")
```

マルチプルユーザセッションの場合、個別の **OCI** コールが使用されます。次のコードに示されているように、サーバに対するアクセスパスが **OCIServerAttach()**で作成され、次いでサーバ属性が設定されます。ユーザセッションは **OCISessionBegin**,によって確立されます。

```
$error_1:=OCIServerAttach ($srvhp;$errhp;"oracle4d")  
$error_1:=OCIAttrSetVal ($svchp;$srvhp;OCI_ATTR_SERVER ;$errhp)  
$error_1:=OCIAttrSetText ($authp;"scott";22;$errhp)  
$error_1:=OCIAttrSetText ($authp;"tiger";23;$errhp)  
$error_1:=OCISessionBegin ($svchp;$errhp;$authp;1;0)  
$error_1:=OCIAttrSetVal ($svchp;$authp;OCI_ATTR_SESSION ;$errhp)
```

#### **Step 4: SQL の発行およびデータの処理**

サーバとの接続が確立し、ユーザセッションが開始したならば、**SQL** ステートメントを実行することによってデータベースサーバのデータにアクセスできるようになります。

**Oracle** データベースのテーブルに入っているデータを更新したり、データを取り出して配列、変数、フィールドなどに入れたりすることができます。**DML (Data Manipulation Language)**、たとえば **INSERT**、**UPDATE**、**DELETE** などのコマンドを使用してデータを更新する場合、ステートメントの前に入力変数のバインドを実行する必要があります。逆に **SQL** ステートメントを使用してデータを取り出す場合、出力変数の定義が必要です。どちらの場合においても、**SQL** ステートメントの準備をしなくてはなりません。バインドや定義の前には **OCIStmtPrepare()**を次のように実行します。

```
$error_1:=OCIStmtPrepare ($Stmthp;$Errhp;$sql_t;OCI_DEFAULT )
```

## **Binding: Changing data in the database.**

---

**SQL** ステートメントの一部として **Oracle** にデータを渡す場合、ステートメントのプレースホルダーと対応する **4D** の入力変数をバインドする必要があります。たとえば **INSERT** を実行する前にはデータを **4D** の変数に代入し、バインドを通して **Oracle** に渡します。**DML** ステートメントやクエリを入力変数で実行する場合、**OCIBindByPos()**や **OCIBindByName()**を使用してそれぞれの入力変数や配列をステートメントの中にあるプレースホルダーにバインドします。

たとえば **INSERT INTO DEPT (DEPTNO,DNAME, LOC) VALUES (:DEPTNO,:DName,:LOC)** というステートメントを使用する場合、**4D** の変数である **deptno**、**dname**、**location** を次のようなコードでバインドすることができます。

```
myptrvar1:=Get pointer("deptno")
myptrvar2:=Get pointer("dname")
myptrvar3:=Get pointer("location")
$error_!:=OCIBindByPos
($Stmthp;$dpp;$Errhp;1;myptrvar1;SQLT_INT ;tindp;trlenp;trcodep;OCI_DEFAULT )
$error_!:=OCIBindByPos
($Stmthp;$dpp;$Errhp;2;myptrvar2;SQLT_STR ;tindp;trlenp;trcodep;OCI_DEFAULT )
$error_!:=OCIBindByPos
($Stmthp;$dpp;$Errhp;3;myptrvar3;SQLT_STR ;tindp;trlenp;trcodep;OCI_DEFAULT )
```

バインドに続いて **OCIStmtExecute()** コマンドでステートメントを実行します。

## Defining: Retrieving data from an Oracle database.

クエリを実行すると **Oracle** から **4D** にデータが返されるので、出力変数を **4D** 側で定義する必要があります。出力変数を定義すると返された結果のストアされる場所とデータフォーマットタイプが関連づけられます。たとえば **Select job from Employee** のような **SQL** を実行する場合、出力変数としての配列を次のようなコードで定義することができます。

```
` SQL Statement: Select job from Employee
ARRAY TEXT(Returnvalue_at;0)
$error_!:=OCIDefineByPos
($Stmthp;$define;$Errhp;1;->vReturnvalue_at;SQLT_STR ;tindp;trlenp;trcodep;OCI_DEFAULT )
```

ステートメントが実行された後には、結果をテキスト変数 **Returnvalue\_at** から取る必要があります。**OCIDefineByPos** コマンドの **4** 番目のパラメータは出力変数の割り当てられた **SQL** ステートメントにおける列の番号です。結果を取るには **OCIStmtFetch()** コマンドを使用します。

```
$error_!:=OCIStmtPrepare ($Stmthp;$Errhp;$sql_t;OCI_DEFAULT ) `Length($sql_t)
```

```

`*** Prepare SQL statement
If ($error_l=0)
  $error_l:=OCIDefineByPos
($Stmthp;$define;$Errhp;1;>vReturnvalue_t;SQL_STR ;tindp;trlenp;trcodep;OCI_DEFAULT )
  `*** Bind Oracle variable by position toward 4D
  If ($error_l=0)
    $error_l:=OCIStmtExecute ($svchp;$Stmthp;$Errhp;0;0;0;0;OCI_DEFAULT )
    `*** Execute the SQL statment
    If ($error_l=0)
      Repeat
        $error_l:=OCIStmtFetch ($Stmthp;$Errhp;1) `*** Fetch the result 1 row at the time
        If ($error_l=0)
          APPEND TO ARRAY(Returnvalue_at;vReturnvalue_t)
        End if
      Until ($error_l=OCI_NO_DATA )
    End if
  End if
End if

```

### Step5: コミット

変更をデータベースに反映させるには **OCITransCommit()** コマンドでコミットを実行することが必要です。パラメータには、次のようにサービスコンテキストハンドルを渡します。コミットをすることなくデータベースが終了した場合、自動的なロールバックの結果として変更が無効になります。

```
$error_l:=OCITransCommit ($svchp;$Errhp;OCI_DEFAULT )
```

### Step6: ユーザセッションおよびサーバ接続の終了

アプリケーションを終了する前にはユーザセッションとサーバへの接続を終了する必要があります。それぞれの終了に使用するコマンドは **OCISessionEnd()** と **OCIserverDetach()** です。

### Step7: ハンドルの解放

**OCI** アプリケーションで不要になったハンドルは、すべて **OCIHandleFree()** コマンドで解放する必要があります。

## OCI Basic Operations

ここからは **OCI** でサポートされる基本的な操作を幾つか考慮します。最初の例題では **Oracle** データベースからデータを取り出して結果を配列に収めます。2 番目の例題では **Oracle** データベースにデータを挿入します。

**Example1:SQL** を使用して **Oracle** データベースからデータを取り出す

**Example2:Oracle** データベースにデータを挿入する

## Summary

**4D for OCI** プラグインは、**OCI (Oracle Call Interface) API** による **Oracle** データベースへのアクセスを可能にします。はじめに **OCI** ライブラリをインストールする必要があり、サーバ用に **tnsnames.ora** ファイルを設定しなくてはなりません。**Oracle** データベースに対するネイティブアクセスは、**OCI** ライブラリが実行します。**4D** による **OCI** アプリケーションは、**OCI** のプログラムフローに従います。**OCI** のプログラムフローは、環境の作成、ハンドルの割り当て、サーバへの接続、セッションの開始、**SQL** ステートメントの処理、接続解除、ハンドルの解放という順序になります。**4D for OCI** プラグインを使用すれば **4D** で **Oracle** データベースに通信することができます。

**Example1:**

```
`Method: QueryData
`Description: This method will retrieved data from Oracle database and place the
result
` into an array.
C_LONGINT($error_l)
C_LONGINT($Envhp;$errhp;$svchp;$authp;$srvhp)
C_LONGINT($srvhp;$svchp;$authp;$errhp)
ARRAY TEXT(Returnvalue_at;0)
C_TEXT(vReturnvalue_t>null_t)
C_POINTER(tindp;trlenp;trcodep)
tindp:=>>null_t `*** Provide a null value to the text variable
C_TEXT($sql_t)
` Select statement.
$sql_t:"Select job from emp"
` *** Create the OCI Environment
$error_l:=OCIEnvCreate ($Envhp;OCI_HTYPE_ENV )
If ($error_l=0)
`*** Allocate Handles and Data Structures
$error_l:=OCIHandleAlloc ($Envhp;$errhp;OCI_HTYPE_ERROR ) ` Allocate Error
handle.
If ($error_l=0)
$error_l:=OCIHandleAlloc ($Envhp;$svchp;OCI_HTYPE_SVCCTX ) `Allocate Service
Context Handle.
If ($error_l=0)
$error_l:=OCIHandleAlloc ($Envhp;$authp;OCI_HTYPE_SESSION )
` Allocate User Session Handle
If ($error_l=0)
$error_l:=OCIHandleAlloc ($Envhp;$srvhp;OCI_HTYPE_SERVER )
` Allocate Server session handle
If ($error_l=0)
```

```

        $loginok_b:=True
    End if
End if
End if
End if
End if
    `*** Connect to Server
$define:=0
If ($error_l=0)
    $error_l:=OCILogon ($Envhp;$errhp;$svchp;"scott";"tiger";"oracle4d") ` connect
server
using simple logon call.
    `**** Issue SQL Statement
If ($error_l=0)
    $error_l:=OCIHandleAlloc ($Envhp;$Stmthp;OCI_HTYPE_STMT) `*** Create SQL
Statement Handler
If ($error_l=0)
    $error_l:=OCIStmtPrepare ($Stmthp;$Errhp;$sql_t;Length($sql_t)) `*** Prepare
SQL
statement
If ($error_l=0)
    $error_l:=OCIDefineByPos ($Stmthp;$define;$Errhp;1;->vReturnvalue_t;SQLT_STR
;tindp;trlenp;trcodep;OCI_DEFAULT) `*** Define output variables
If ($error_l=0)
    $error_l:=OCIStmtExecute ($svchp;$Stmthp;$Errhp;0;0;0;OCI_DEFAULT) `**
*
Execute the SQL statment
If ($error_l=0)
    Repeat
        $error_l:=OCIStmtFetch ($Stmthp;$Errhp;1) `*** Fetch the result 1 row at
the time
        If ($error_l=0)
            APPEND TO ARRAY(Returnvalue_at;vReturnvalue_t) `*** Put fetch results
into array.
        End if
    Until ($error_l=OCI_NO_DATA)
    End if
End if
    $error_l:=OCIHandleFree ($Stmthp)
End if
End if
End if
End if
If ($loginok_b)
    `*** Disconnect from server
    $error_l:=OCILogoff ($svchp;$errhp)
    `**** Free Handles
    $error_l:=OCIHandleFree ($svchp)
    $error_l:=OCIHandleFree ($srvhp)
    $error_l:=OCIHandleFree ($errhp)
    $error_l:=OCIHandleFree ($authp)
End if

```

Example2:

```

`Method: InsertData
`Description: This method will insert a record in the Oracle database.
C_LONGINT($error_l)
C_LONGINT($Envhp;$errhp;$svchp;$authp;$srvhp)
C_LONGINT($srvhp;$svchp;$authp;$errhp)
C_TEXT(vReturnvalue_t;null_t)
C_POINTER(tindp;trlenp;trcodep)
C_POINTER(myptrvar1;myptrvar2;myptrvar3)
C_LONGINT(deptno1)
C_TEXT(dname1;loc1)
C_TEXT($sql_t)
deptno1:=12
dname1:="Management"
loc1:="Seattle"
myptrvar1:=Get pointer("deptno1")
myptrvar2:=Get pointer("dname1")
myptrvar3:=Get pointer("loc1")
tindp:=->null_t `*** Provide a null value to the text variable
`SQL STATEMENT to insert record to DEPT table in Oracle.
`The variables preceded by : are place holders.
$sql_t:="INSERT INTO DEPT (DEPTNO,DNAME, LOC) VALUES (:DEPTNO,:DName,:LOC)"
`*** Create the OCI Environment
$error_l:=OCIEnvCreate ($Envhp;OCI_HTYPE_ENV )
If ($error_l=0)
`*** Allocate Handles and Data Structures
$error_l:=OCIHandleAlloc ($Envhp;$errhp;OCI_HTYPE_ERROR )
`Allocate Error handle.
If ($error_l=0)
$error_l:=OCIHandleAlloc ($Envhp;$svchp;OCI_HTYPE_SVCCTX )
`Allocate Service Context Handle.
If ($error_l=0)
$error_l:=OCIHandleAlloc ($Envhp;$authp;OCI_HTYPE_SESSION )
`Allocate User Session Handle
If ($error_l=0)
$error_l:=OCIHandleAlloc ($Envhp;$srvhp;OCI_HTYPE_SERVER )
`Allocate Server session handle
If ($error_l=0)
$loginok_b:=True
End if
End if
End if
End if
End if
`*** Connect to Server and Begin Session
$define:=0
If ($error_l=0)
$error_l:=OCIServerAttach ($srvhp;$errhp;"oracle4d")
`Create access path to oracle database
If ($error_l=0)
$error_l:=OCIAttrSetVal ($svchp;$srvhp;OCI_ATTR_SERVER ;$errhp)
`Set server attributes
If ($error_l=0)
$error_l:=OCIAttrSetText ($authp;"scott";22;$errhp) `Set user name to access
server. 22 is attribute type.

```

```

If ($error_=0)
$error_:=OCIAttrSetText ($authp;"tiger";23;$errhp) ` Set password. 23 is attribute
type.
If ($error_=0)
$error_:=OCISessionBegin ($svchp;$errhp;$authp;1;0) ` Begin user session
If ($error_=0)
$error_:=OCIAttrSetVal ($svchp;$authp;OCI_ATTR_SESSION ;$errhp) `Set
session attribute
`**** Issue SQL Statement
If ($error_=0)
$error_:=OCIHandleAlloc ($Envhp;$Stmthp;OCI_HTYPE_STMT )
` *** Create SQL Statement Handler
If ($error_=0)
$error_:=OCIStmtPrepare ($Stmthp;$Errhp;$sql_t;Length($sql_t))
` *** Prepare SQL statement
If ($error_=0)
` ***Bind variables to positions in the SQL statement.
` The 4th parameter is the position in the SQL Statment, the 5th parameter is
` a pointer to the variable to bind, and the 6th parameter is the data type.
$error_:=OCIBindByPos ($Stmthp;$dpp;$Errhp;1;myptrvar1;SQLT_INT
;tindp;trlenp;trcodep;OCI_DEFAULT )
$error_:=OCIBindByPos ($Stmthp;$dpp;$Errhp;2;myptrvar2;SQLT_STR
;tindp;trlenp;trcodep;OCI_DEFAULT )
$error_:=OCIBindByPos ($Stmthp;$dpp;$Errhp;3;myptrvar3;SQLT_STR
;tindp;trlenp;trcodep;OCI_DEFAULT )
If ($error_=0)
$error_:=OCIStmtExecute ($svchp;$Stmthp;$Errhp;1;0;0;OCI_DEFAULT )
` *** Execute the SQL statment
$error_:=OCITransCommit ($svchp;$Errhp;OCI_DEFAULT )
`*** Commit changes to database.
End if
`*** Free statement handle.
$error_:=OCIHandleFree ($Stmthp)
End if
If ($loginok_b)
` ***Disconnect from server
$error_:=OCISessionEnd ($svchp;$errhp;$authp)
$error_:=OCIserverDetach ($srvhp;$errhp)
`***** Free Handles
$error_:=OCIHandleFree ($svchp)
$error_:=OCIHandleFree ($srvhp)
$error_:=OCIHandleFree ($errhp)
$error_:=OCIHandleFree ($authp)
End if

```