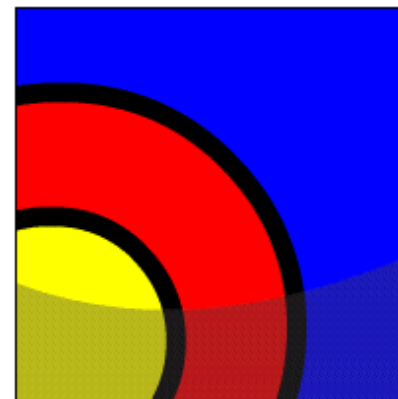


Managing BFILES

What Will I Learn?

In this lesson, you will learn to:

- Define BFILES and the BFILE column data type
- Create DIRECTORY objects and view them in the Data Dictionary
- Manage and manipulate BFILES using BFILENAME and DBMS_LOB



Why Learn It?

You have already learned about two of the Oracle LOB data types: CLOB and BLOB.

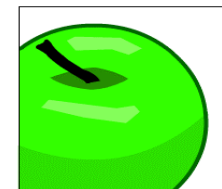
A BFILE is a third LOB data type whose value is stored outside the database.

Because a BFILE value is stored outside the database (unlike CLOBs and BLOBs), you need to learn new ways of creating and managing it.



Tell Me/Show Me

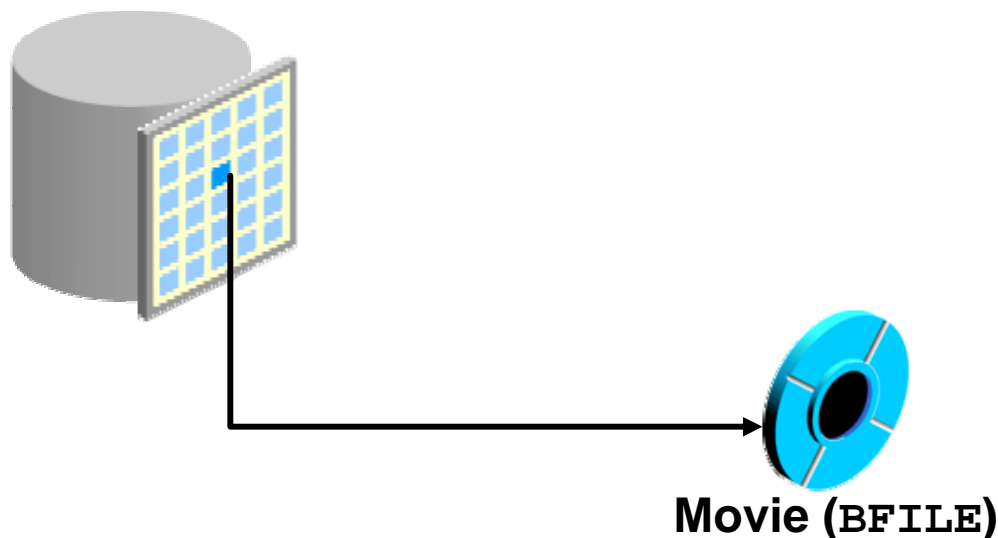
What Is a BFILE?



A BFILE is like a CLOB or BLOB, except that its value is stored outside the database in a separate file. The database holds a pointer to the external file.

The data can be text (like a CLOB) or multimedia (like a BLOB).

The external file can be on a normal computer disk, or on a CD or DVD.





Tell Me/Show Me

How is a BFILE Different From CLOBs and BLOBs?

Because the BFILE data is stored outside the database:

- It can be read but not modified; therefore, it must be created outside Oracle
- It cannot have normal database privileges granted on it
- Normal SQL statements cannot be used on it
- All data access is through the DBMS_LOB package



Tell Me/Show Me

When to Use a BFILE?

- If the media already exists (for example, CD or DVD), then why bother re-copying it into the database (into a BLOB column)?
- If the media is already read-only (for example, CD or DVD), then the BFILE being read-only is not an issue.

When *NOT* to use a BFILE?

- If it doesn't fit! For Oracle 9i and some 10g, the limit is 4 GB.
- How big is a DVD? Up to 4.7 GB.
- How big is a Blu-ray or HD-DVD? From 15 to 50 GB.



Tell Me/Show Me

A New Database Object: DIRECTORY

You need a way of specifying which operating system directories (folders) contain your BFILES, and also controlling privileges: Which Oracle users are allowed to read the BFILE data. To do this, you create a DIRECTORY.

A DIRECTORY is a pointer from the database to an operating system directory (Windows folder) where BFILES are stored.



Tell Me/Show Me

Creating and Managing Directories

Create a directory to point to an external location where BFILES are stored, then allow everyone to use it:

```
CREATE DIRECTORY movie_dir AS 'c:\mymovies';  
  
GRANT READ ON DIRECTORY movie_dir TO PUBLIC;
```

Suppose that later you move your movie files to a different location. You must update the directory pointer:

```
ALTER DIRECTORY movie_dir AS 'c:\latermovies';
```




Tell Me/Show Me

Viewing Directories in the Data Dictionary

Unlike most database objects, directories do not belong to any schema, so there is no `USER_DIRECTORIES` dictionary view. Instead, you use `ALL_DIRECTORIES`:

```
SELECT directory_name, directory_path  
FROM all_directories;
```

DIRECTORY_NAME	DIRECTORY_PATH
WF_FLAGS	/u02/webapps/oa1bprd_dir

1 rows returned in 0.05 seconds [Download](#)



Tell Me / Show Me

Adding and Populating a BFILE column for a Table

You add the column like any other column:

```
ALTER TABLE employees ADD (movie BFILE);
```

Then you populate it with a locator value that points to a specific file in a specific directory, using the `DIRECTORY` pointer you created earlier.

You do this using the inbuilt `BFILENAME` function, within a PL/SQL block. The next slide shows how to do this.



Tell Me / Show Me

Adding and Populating a BFILE Column: Example

```
DECLARE
    v_locator          BFILE;                                --1
BEGIN
    v_locator := BFILENAME('MOVIE_DIR','titanic.avi');      --2
    IF DBMS_LOB.FILEEXISTS(v_locator) = 1 THEN              --3
        DBMS_LOB.FILEOPEN(v_locator);
        UPDATE employees SET movie = v_locator              --4
            WHERE employee_id = 100;
        DBMS_LOB.FILECLOSE(v_locator);
    ELSE
        RAISE_APPLICATION_ERROR
            (-20210,'This BFILE does not exist');
    END IF;
END;
```



Tell Me / Show Me

Reading BFILE Locator and Data Values

You can read the locator value using `DBMS_LOB.FILEGETNAME`:

```
DECLARE
    v_locator          BFILE;
    v_directory         VARCHAR2(30);
    v_filename         VARCHAR2(50);
BEGIN
    SELECT movie INTO v_locator
        FROM employees WHERE employee_id = 100;
    DBMS_LOB.FILEGETNAME(v_locator,v_directory,v_filename);
    DBMS_OUTPUT.PUT_LINE(v_directory||' '||v_filename);
END;
```

You read the data value using `DBMS_LOB.READ` as for BLOBs, but the data cannot be displayed in Oracle Application Express.

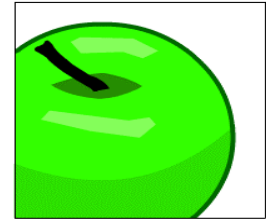
Tell Me/Show Me

Terminology

Key terms used in this lesson include:

BFILE

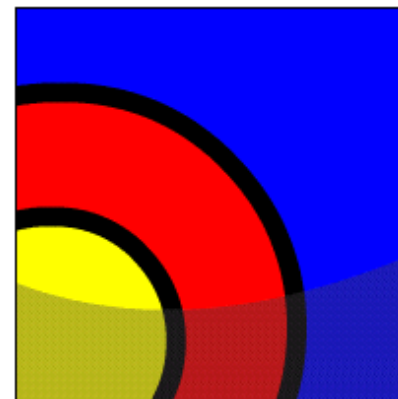
DIRECTORY



Summary

In this lesson, you learned to:

- Define BFILEs and the BFILE column data type
- Create DIRECTORY objects and view them in the Data Dictionary
- Manage and manipulate BFILEs using BFILENAME and DBMS_LOB





Try It/Solve It

This practice covers the following topics:

- Creating `DIRECTORY` objects and viewing them in the Data Dictionary
- Managing and manipulating `BFILES` using `BFILENAME` and `DBMS_LOB`

