Oracle *inter*Media Image Quick Start

Object Interface

Introduction

Oracle *inter*Media ("*inter*Media") is a feature that enables Oracle Database to store, manage, and retrieve images, audio, video, or other heterogeneous media data in an integrated fashion with other enterprise information. Oracle *inter*Media extends Oracle Database reliability, availability, and data management to multimedia content in traditional, Internet, electronic commerce, and media-rich applications.

This article provides simple PL/SQL examples that upload, store, manipulate, and export image data inside a database using *inter*Media. Some common pitfalls are also highlighted. We assume only Oracle Database release 9*i* or later with Oracle *inter*Media installed (the default configuration provided by Oracle Universal Installer).

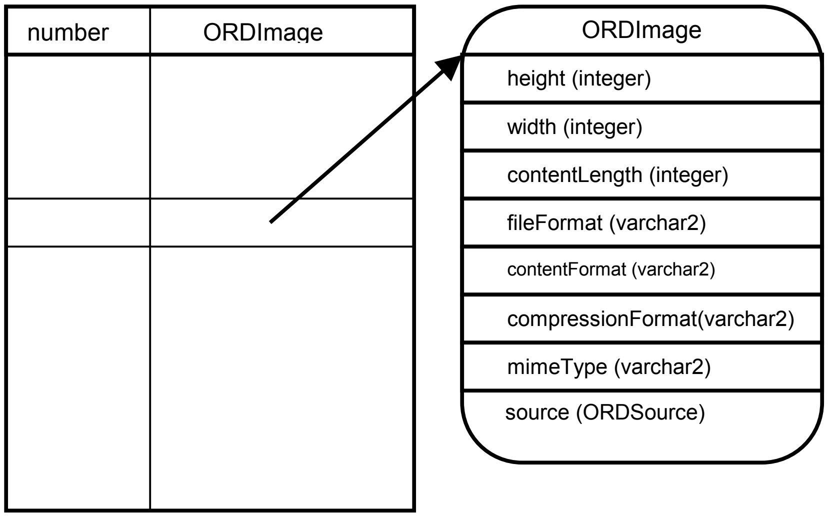
NOTE: Access to an administrative account is required in order to grant the necessary file system privileges. In the following examples, you should change the command `connect / as sysdba` to the appropriate `connect username/password as sysdba` for your site. The following examples also connect to the database using `connect scott/tiger`, which you should change to an actual user name and password on your system. You should also modify the definition of `IMGDIR` to point at the directory where you have downloaded the three sample image files `goats.gif`, `flowers.jpg`, and `dummy.dcm`.

Overview of ORDImage

Oracle can store image data in either a Binary Large OBject ("BLOB") column or an ORDImage object column. Some of the advantages of using the ORDImage object are:

1. Tight integration with Oracle's development tools such as Oracle JDeveloper, Oracle Content Management SDK, Oracle Application Server Portal, and so on.
2. Information such as height and width is automatically determined and stored with the image data.

An example of an ORDImage object in a database table is illustrated in the following diagram.



Creating a Table with an ORDImage Column

First, we create a simple table with two columns: a numeric identifier (`id`) and an ORDSYS.ORDImage object (`image`). Note that all *inter*Media objects and procedures are defined in the ORDSYS schema.

```
connect scott/tiger
create table image_table (id number primary key, image ordsys.ordimage);
```

Importing Images

This section shows how to bring images from the file system into the newly created table named `image_table`.

1. Create a *directory* object within the database that points to the file system directory that contains the sample image files. This is the directory where you saved the image files included with this quickstart.

```
connect / as sysdba
create or replace directory imagedir as '/home/alamb/quickstart/';
-- For windows: create or replace directory imagedir as 'c:\quickstart';
grant read on directory imagedir to scott;
```

2. Create a PL/SQL procedure `image_import()` that inserts a new row into `image_table` and then imports the image data in `filename` into the newly created `ORDImage` object.

```
create or replace procedure image_import(dest_id number, filename varchar2) is
  img ordsys.ordimage;
  ctx raw(64) := null;
begin
   delete from image_table where id = dest_id;
   insert into image_table (id, image)
     values (dest_id, ordsys.ordimage.init())
     returning image into img;

   img.importFrom(ctx, 'file', 'IMAGEDIR', filename);
   update image_table set image=img where id=dest_id;
end;
/
```

3. Call the newly created procedure to import 2 sample image files.

```
call image_import(1,’flowers.jpg’);
call image_import(2,’goats.gif’);
```

NOTE: The directory object is named `IMAGEDIR` (in uppercase letters) even if it was created with upper or lower case letters. Thus the command `img.importFrom(ctx, 'file', 'imagedir', filename);` **will not work** and the following error is returned.

```
ORA-22285: non-existent directory or file for FILEOPEN operation error.
```

NOTE: If the image you are importing is not one of *inter*Media’s supported formats (for example JPEG2000) the following error is returned.

```
ORA-29400: data cartridge error
IMG-00705: unsupported or corrupted input format
```

For information on the use of *inter*Media to manage non-natively supported image formats, see the appendix.

Selecting and Viewing Image Properties

Once `image_table` has been populated, you can access image metadata using SQL queries. In the following example, we demonstrate how to select some information from the imported images:

1. Height and width.
2. File format and compression format.
3. Content format (RGB, grayscale, and so on) and content length (bytes of image data).

```
connect scott/tiger
select id,
       t.image.getheight(),
       t.image.getwidth()
from image_table t;

select id,
       t.image.getfileformat(),
       t.image.getcompressionformat()
from image_table t;

select id,
       t.image.getcontentformat(),
       t.image.getcontentlength()
from image_table t;
```

The resulting output looks like the following (the formatting commands are in the included script files).

```
        id     height      width
---------- ---------- ----------
         1        600        800
         2        375        500


        id fileformat                     compression
---------- ------------------------------ ------------------------------
         1 JFIF                           JPEG
         2 GIFF                           GIFLZW


        id contentformat                      length
---------- ------------------------------ ----------
         1 24BITRGB                            66580
         2 8BITLUTRGBT                        189337
```

Creating Thumbnails and Changing Formats

We next illustrate some image processing operations that can be invoked within the database. To generate a new ORDImage object from an existing one, the programmer describes the desired properties of the new image. For example, the following description generates a JPEG thumbnail image of size 75x100 pixels:
`‘fileformat=jfif fixedscale=75 100’`.

NOTE: Some three-letter image file extensions and the corresponding *inter*Media `fileformat` are as follows.

Extension	`fileformat`
.jpg	JFIF (9*i*, 10*g*), JPEG (10*g*)
.gif	GIFF(9*i*, 10*g*), GIF (10*g*)
.tif, .tiff	TIFF
.png	PNGF

The following example defines `image_processCopy()` that adds a new row to `image_table` with identifier `dest_id` and generates an ORDImage in the new row by `processCopy`'ing of the ORDImage in the source row.

```
connect scott/tiger
create or replace procedure image_processCopy(source_id number, dest_id number, verb varchar2) is
    imgSrc    ordsys.ordimage;
    imgDst    ordsys.ordimage;
begin
  delete from image_table where id = dest_id;
  insert into image_table (id, image)
    values (dest_id, ordsys.ordimage.init());
  select image into imgSrc from image_table where id = source_id;
  select image into imgDst from image_table where id = dest_id for update;
  imgSrc.processCopy(verb, imgDst);
  update image_table set image = imgDst where id = dest_id;
end;
/

-- Scale flowers.jpg to 10% into row with id=3
call image_processcopy(1,3,'scale=.1');

-- convert goats.gif to grayscale jpeg thumbnail into row with id=4
call image_processcopy(2,4,'fileformat=jfif contentformat=8bitgray maxscale=100 100');

-- look at our handiwork
column t.image.getfileformat() format A20;
select id, t.image.getWidth(), t.image.getHeight(), t.image.getFileFormat()
       from image_table t;
```

The preceding example generates the following output.

```
        ID T.IMAGE.GETWIDTH() T.IMAGE.GETHEIGHT() T.IMAGE.GETFILEFORMA
---------- ------------------ ------------------- --------------------
         1                800                 600 JFIF
         2                500                 375 GIFF
         3                 80                  60 JFIF
         4                100                  75 JFIF
```

NOTE: The following error might be returned from `ORDImage.processCopy()`.

```
ORA-29400: data cartridge error
IMG-00703: unable to read image data
ORA-28575: unable to open RPC connection to external procedure agent
```

In Oracle Database release 9*i*, JPEG (and some other less common formats) encoding and decoding requires the external procedure agent (extproc). To fix the preceding error, the Oracle Listener needs to be configured to use extproc. See technical note 198099.1, *Configuration of the External Procedure Call for interMedia* at http://metalink.oracle.com for detailed instructions on setting up extproc. Oracle Database release 10*g* does not require extproc for JPEG encoding and decoding.

If you do not want to change your Oracle Net configuration, try changing the file format to pngf as follows.

```
-- convert goats.gif to grayscale png thumnbail
call image_processcopy(2,5,'fileformat=pngf contentformat=8bitgray maxscale=100 100');
```

Exporting Images with ORDImage.export()

Exporting image data from the database with *inter*Media's export method requires that the database write to the file system. Writing to the file system requires granting JAVA permissions to your user (scott in the examples) **and** to the ORDSYS* schema as shown in the following example.

```
connect / as sysdba
create or replace directory imagedir as '/home/alamb/quickstart';
-- For windows:
--create or replace directory imagedir as 'c:\quickstart';
grant read on directory imagedir to scott;

call dbms_java.grant_permission('SCOTT','java.io.FilePermission',
                                '/home/alamb/quickstart/*','WRITE');
call dbms_java.grant_permission('ORDSYS','java.io.FilePermission',
                                '/home/alamb/quickstart/*','WRITE');
-- For windows:
--call dbms_java.grant_permission('SCOTT','java.io.FilePermission','c:\quickstart\*','WRITE');
--call dbms_java.grant_permission('ORDSYS','java.io.FilePermission','c:\quickstart\*','WRITE');

connect scott/tiger
-- Writes the image data from ORDImage with id=<source_id> in image_table
-- to the file named <filename> in the IMAGEDIR directory
create or replace procedure image_export (source_id number, filename varchar2) as
  imgSrc ordsys.ordimage;
  ctx raw(64) := null;
begin
  select image into imgSrc from image_table where id = source_id;
  imgSrc.export(ctx, 'FILE', 'IMAGEDIR', filename);
end;
/

call image_export(3, 'flowers_thumbnail.jpg');
call image_export(4, 'goats_grayscale.jpg');
```

***NOTE**: For Oracle Database releases 9.2.0.1, 9.2.0.2, and 9.2.0.3 you **must** change ORDSYS in the preceding export example to ORDPLUGINS.

Cleaning Up

To restore your database to its original state, you need to remove all of the objects that were created in this quickstart as shown in the following example.

```
connect / as sysdba
drop directory imagedir;
call dbms_java.revoke_permission('SCOTT','java.io.FilePermission',
                                 '/home/alamb/quickstart/*','WRITE');
call dbms_java.revoke_permission('ORDSYS','java.io.FilePermission',
                                 '/home/alamb/quickstart/*','WRITE');
-- For windows:
-- call dbms_java.revoke_permission('SCOTT','java.io.FilePermission', 'c:\quickstart\*','WRITE');
-- call dbms_java.revoke_permission('ORDSYS','java.io.FilePermission','c:\quickstart\*','WRITE');

connect scott/tiger
drop procedure image_import;
drop procedure image_processcopy;
drop procedure image_export;
drop table image_table;
```

Conclusion

Using the provided ORDImage object type, we have shown how to import images into the database, write SQL queries based on image metadata (width, height, and so on), perform basic image processing, and export images to the file system.

Oracle *inter*Media provides more functionality than is covered in this quickstart. Refer to the following documentation for more information: *Oracle interMedia User's Guide and Reference, Release 9.0.1*, *Oracle interMedia Reference, 10g Release 1 (10.1)*, and *Oracle interMedia User's Guide, 10g Release 1 (10.1).* Additional examples and articles are available on the *inter*Media web page on the Oracle Technology Network at http://www.oracle.com/technology/products/intermedia/index.html.

APPENDIX: Using *inter*Media with Non-Natively Supported Image Formats

For image formats that *inter*Media understands, image properties such as height and width are automatically set when `ORDImage.importFrom()` is called. For image formats such as JPEG2000 that are not natively supported by *inter*Media, the metadata fields must be populated by the user. Metadata can be set by updating the fields of an ORDImage object either via a SQL `UPDATE` statement or with an object field assignment. The following example shows the procedure `image_import_other()` that imports an image in an unsupported format and sets the ORDImage fields to the values supplied in the arguments.

```
connect scott/tiger
create or replace procedure image_import_other
  (dest_id number, filename varchar2, imgFileFormat varchar2,
   imgHeight number, imgWidth number, imgMimeType varchar2) is
  img ordsys.ordimage;
  ctx raw(64) := null;
begin
  delete from image_table where id = dest_id;
  insert into image_table (id, image)
    values (dest_id, ordsys.ordimage.init());

  -- import the actual image data into the database and set the
  -- other ORDImage fields manually. Note that setting the filetype
  -- to 'OTHER:filetype' causes no auto-setting of fields
  select image into img from image_table where id=dest_id for update;
  img.fileFormat := 'OTHER:' || imgFileFormat;
  img.height   := imgHeight;
  img.width    := imgWidth;
  img.mimeType := imgMimeType;
  img.importFrom(ctx, 'file', 'IMAGEDIR', filename);
  img.contentLength := dbms_lob.getlength(img.source.localdata);
  update image_table set image=img where id=dest_id;
end;
/

call image_import_other(6,'dummy.dcm', 'DICOM', 100, 200, 'application/dicom');

-- view the properties we just created (nicely formatted)
column "mimetype" format A20;
select id                    "id",
       t.image.getWidth()          "width",
       t.image.getHeight()         "height",
       t.image.getMimeType()       "mimetype",
       t.image.getContentLength() "length"
from image_table t;
```

This example produces output similar to the following.

```
        id      width     height mimetype                 length
---------- ---------- ---------- -------------------- ----------
         1        800        600 image/jpeg                66580
         2        500        375 image/gif                189337
         3         80         60 image/jpeg                 1918
         4        100         75 image/jpeg                 2156
         5        100         75 image/png                  5624
         6        200        100 application/dicom           183
```

To cleanup this example, run the following.

```
connect scott/tiger
drop procedure image_import_other;
```