

Manejo de Oracle Multimedia sobre Oracle Database 12c

Por Francisco Riccio 

Introducción

Oracle Multimedia es un conjunto de servicios que provee la base de datos Oracle para el manejo de contenido multimedia (imagen, audio y video) a través de los esquemas: ORDSYS, ORDPLUGINS, SI_INFORMTN_SCHEMA y ORDDATA.

Funciona integrando los tipos de datos BLOB y BFILE con información adicional (metadata), además de ofrecer métodos que permiten realizar manipulaciones tales como: copiado, escalado, recorte, compresión, etc.

Este producto estuvo disponible desde la versión Oracle 8.0.3 bajo el nombre Oracle Image Cartridge, luego con el nombre Oracle Visual Information Retrieval, en la versión Oracle 9.0 se liberó con el nombre Oracle Intermedia y desde la versión Oracle 11g el producto se llama como lo conocemos hoy (Oracle Multimedia).

Las diferentes funcionalidades presentadas en cada versión están disponible en: My Oracle Support (MOS) Nota: 223443.1 (Features per Version of Oracle Multimedia/Multimedia).

Las ventajas de Oracle Multimedia respecto a BLOB & BFILE son:

- El contenido multimedia es reconocido por la base de datos con todas sus propiedades mientras un dato BLOB o BFILE no se sabe que información contiene.
- Los datos son almacenados mediante objetos dentro de Oracle Multimedia. Los objetos permiten extraer metadata o realizar modificaciones sobre los datos directamente desde sus métodos.
- Oracle Multimedia tiene la posibilidad de almacenar los datos como BLOB o BFILE y realizar migraciones entre ellos.

Este producto es gratuito y se encuentra disponible en todas las ediciones a excepción de la Express Edition.

Antes de iniciar con el desarrollo de las demos, recomiendo revisar la siguiente lectura con la finalidad de disponer una mayor comprensión sobre los tipos de datos BLOB y BFILE:

<http://www.oracle.com/technetwork/es/articles/sql/manejo-de-oracle-large-objects-1937051-esa.html>

Instalación, Validación y Consideraciones

Instalación

Mediante la herramienta Database Configuration Assistant (DBCA) debemos simplemente activar la opción de “Oracle Multimedia” durante el proceso de creación de la base de datos como se aprecia en el siguiente gráfico:



También es posible realizar la instalación vía scripts, como a continuación se detalla:

```
Export PATH=$ORACLE_BASE/product/12.1.0/dbhome_1/perl/bin:$PATH
cd $ORACLE_HOME/rdbms/admin
perl catcon.pl -u SYS -d $ORACLE_HOME/ord/admin -b ordinst ordinst.sql '--pSYSaux' '--
pSYSaux'
perl catcon.pl -u SYS -d $ORACLE_HOME/ord/im/admin -b catim catim.sql
```

Nota: Antes de ejecutar los scripts previamente mencionados, debe estar instalados los siguientes componentes en la base de datos: Oracle JVM, Oracle XML Database y Oracle XDK.

Validación del Componente Oracle Multimedia

Si deseamos validar la correcta instalación de Oracle Multimedia podemos ejecutar las siguientes consultas en las versiones de Oracle Database 11g/12c.

```
SQL> set serveroutput on
SQL> execute validate ordim;

PL/SQL procedure successfully completed.

SQL> select version, status from dba_registry where comp_id='ORDIM';

VERSION                                STATUS
-----                                -
12.1.0.2.0                              VALID
```

Debemos validar que el estado del componente se encuentre con el valor de VALID.

El procedimiento de validación para versiones inferiores a Oracle Database 11g se encuentra disponible en: My Oracle Support (MOS) Nota: 221337.1 (How to check if Multimedia Audio/Image/Video is Installed Correctly?).

Consideraciones

Es importante indicar que Oracle Multimedia trabaja sobre una gama de formatos multimedia comerciales y en cada versión se amplía este número.

El detalle de los formatos soportados por cada rubro en una base de datos Oracle 12c se encuentra en los siguientes url:

- Imagen: https://docs.oracle.com/database/121/AIVUG/ap_imgfmts.htm#AIVUG30000
- Audio: https://docs.oracle.com/database/121/AIVUG/ap_audfmts.htm#AIVUG20000
- Video: https://docs.oracle.com/database/121/AIVUG/ap_vidfmts.htm#AIVUG40000

En caso ingresemos un formato no soportado por la versión de base de datos conseguiremos el siguiente error:

```
ORA-29400: error de cartucho de datos
IMG-00580: specified format does not support output
ORA-06512: en "ORDSYS.ORDIMG_PKG", línea 1061
ORA-06512: en "ORDSYS.ORDIMAGE", línea 175
```

A continuación se desarrollarán 3 demos las cuales fueron implementadas sobre una base de datos Oracle 12.1.0.2 en una plataforma Oracle Linux 5.10 x64 bits.

Demo #1 - Manejando imágenes con ORDImage

En la siguiente demo se implementará una aplicación sencilla en .NET utilizando ODP.NET 12c, el cual permitirá almacenar imágenes de artistas musicales y a la vez mostrarlas junto con sus propiedades. Por cada imagen insertada, deberá registrarse en nuevos campos los siguientes cambios:

- Reducción del 50% sobre las dimensiones originales.
- Tamaño de la imagen original en un formato de: 50x50 píxeles.

Paso 1 - Creación de la tabla.

```
SQL> create table ARTISTA (id number generated as identity, nombre varchar(25),
                             foto1 ORDImage, foto2 ORDImage, foto3 ORDIMAGE);

Table created.

SQL> alter table ARTISTA add constraint PK_ARTISTA primary key(id);

Table altered.
```

Paso 2 – Creamos un procedimiento almacenado para insertar una foto.

```
SQL> create or replace procedure spu_insertar_artista(pnombre ARTISTA.nombre%type, pfoto blob)
is
    vid number;
    vimg ORDSYS.ORDImage;
    vfoto1 ORDSYS.ORDImage;
    vfoto2 ORDSYS.ORDImage;
    vfoto3 ORDSYS.ORDImage;
begin
    insert into ARTISTA (nombre,foto1,foto2,foto3)
        values (pnombre,ORDSYS.ORDImage.Init(),ORDSYS.ORDImage.Init(),ORDSYS.ORDImage.Init())
        returning id,foto1,foto2,foto3 into vid,vfoto1,vfoto2,vfoto3;
    vfoto1:=ORDImage(pfoto);
    vfoto1.processCopy('scale=1',vfoto1);
    vfoto1.processCopy('scale=0.5',vfoto2);
    vfoto1.processCopy('fixedScale=50 50',vfoto3);
    update ARTISTA set foto1=vfoto1,foto2=vfoto2,foto3=vfoto3 where id=vid;
    commit;
end;
/

Procedure created.
```

Nota:

- El tipo de dato ORDImage nos permite almacenar imágenes; asimismo existen los tipos de datos: ORDAudio, ORDVideo y ORDDoc (Mix de imagen, audio y video).
- El método INIT inicializa el objeto de tipo ORDImage.

- ProcessCopy (método del objeto ORDImage) permite copiar una imagen modificada a medida a otra variable de tipo ORDImage. Este método permitirá cumplir el requerimiento de mantener una copia de la imagen original con las reducciones de tamaño previamente mencionadas.

Mayor información sobre las opciones del método se encuentra disponible en el siguiente url:

https://docs.oracle.com/cd/E11882_01/appdev.112/e10776/ap_imgproc.htm#AIVUG50000

Paso 3 - Creación de la Aplicación.

Operación – INSERT

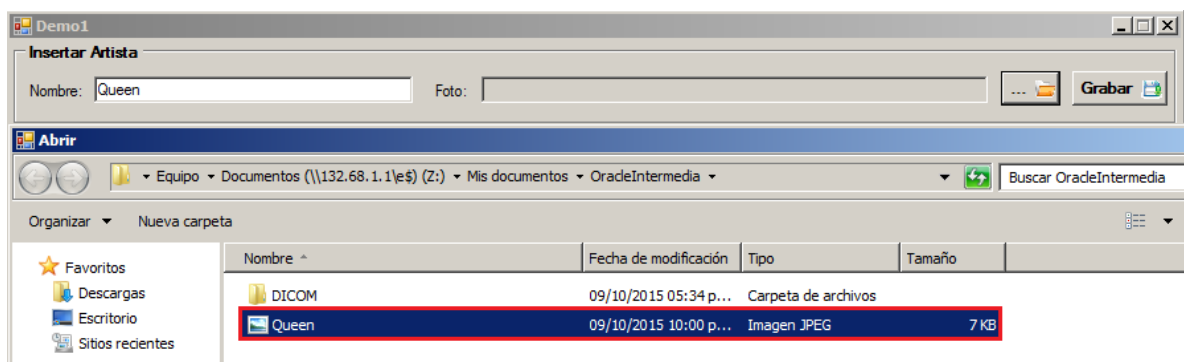
```
private void btngrabar_Click(object sender, EventArgs e)
{
    if ((txtabrirarchivo.Text != "") && (txtnombreartista.Text != ""))
    {
        Demo1DAO objdemo1dao = new Demo1DAO();
        if (objdemo1dao.insertarArtista(txtnombreartista.Text, txtabrirarchivo.Text) == 0)
        {
            MessageBox.Show("El artista fue registrado.", "Registro satisfactorio",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            MessageBox.Show("Hubo problemas para registrar el artista en la BD.", "Problemas al registrar",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        objdemo1dao = null;
    }
    else
    {
        MessageBox.Show("Debe ingresar el nombre del artista y su foto.", "Datos Incompletos",
            MessageBoxButtons.OK, MessageBoxIcon.Stop);
    }
}
```

```

class Demo1DAO
{
    private String _cadenacn = "User ID=friccio;Password=oracle;Data Source=PRD";
    private OracleConnection _cn = null;
    private OracleCommand _cmd = null;

    public int insertarArtista(String pnombre, String prutaArchivo)
    {
        int resultado = 0;
        _cn = new OracleConnection(_cadenacn);
        try
        {
            _cn.Open();
            _cmd = _cn.CreateCommand();
            _cmd.CommandType = CommandType.StoredProcedure;
            _cmd.CommandText = "spu_insertar_artista";
            _cmd.Parameters.Add("PNOMBRE", OracleDbType.Varchar2).Value = pnombre;
            FileStream archivo = new FileStream(prutaArchivo, FileMode.Open, FileAccess.Read);
            byte[] foto = new byte[archivo.Length];
            archivo.Read(foto, 0, System.Convert.ToInt32(archivo.Length));
            archivo.Close();
            _cmd.Parameters.Add("PFOTO", OracleDbType.Blob).Value = foto;
            _cmd.ExecuteNonQuery();
        }
        catch
        {
            resultado = -1;
        }
        finally
        {
            if ((_cn != null) && (_cn.State == ConnectionState.Open))
            {
                _cn.Close();
                _cmd.Dispose();
                _cn.Dispose();
            }
        }
        return resultado;
    }
}

```



Paso 4 - Operación Lectura.

```
private void btnmostrar_Click(object sender, EventArgs e)
{
    if (txtmostrarid.Text != "")
    {
        Demo1DAO objdemo1dao = new Demo1DAO();
        EArtista objArtista = objdemo1dao.getArtista(Convert.ToInt32(txtmostrarid.Text));
        if (objArtista != null)
        {
            txtmostrarnombre.Text = objArtista.Nombre;
            MemoryStream memimg = new MemoryStream(objArtista.Fotos[0].Img);
            pbfoto1.Image = Image.FromStream(memimg);
            txtmostrarfoto1Ancho.Text = objArtista.Fotos[0].Ancho.ToString();
            txtmostrarfoto1Alto.Text = objArtista.Fotos[0].Alto.ToString();
            txtmostrarfoto1formato.Text = objArtista.Fotos[0].Formato;
            txtmostrarmime.Text = objArtista.Fotos[0].Mime;
            txtmostrarfoto1compresion.Text = objArtista.Fotos[0].Compresion;
            txtmostrarfoto1tamano.Text = objArtista.Fotos[0].Tamano.ToString();
            memimg = new MemoryStream(objArtista.Fotos[1].Img);
            pbfoto2.Image = Image.FromStream(memimg);
            memimg = new MemoryStream(objArtista.Fotos[2].Img);
            pbfoto3.Image = Image.FromStream(memimg);
        }
        else
        {
            MessageBox.Show("Hubo problemas para mostrar el artista desde la BD.", "Problemas al mostrar",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        objArtista = null;
        objdemo1dao = null;
    }
    else
    {
        MessageBox.Show("Debe ingresar el ID del artista.", "Datos Incompletos",
            MessageBoxButtons.OK, MessageBoxIcon.Stop);
    }
}

class EArtista
{
    public String Nombre { get; set; }
    public List<EFoto> Fotos { get; set; }

    public EArtista()
    {
        Fotos = new List<EFoto>();
    }
}
}
```

```

class EFoto
{
    public int Ancho { get; set; }
    public int Alto { get; set; }
    public String Formato { get; set; }
    public String Mime { get; set; }
    public String Compresion { get; set; }
    public int Tamano { get; set; }
    public byte[] Img { get; set; }

    public EFoto()
    {
    }
}

public EArtista getArtista(int pid)
{
    EArtista objArtista = new EArtista();
    StringBuilder SQL = new StringBuilder();
    _cn = new OracleConnection(_cadenacn);
    try
    {
        _cn.Open();
        _cmd = _cn.CreateCommand();
        _cmd.CommandType = CommandType.Text;
        SQL.Append(" select nombre, a.foto1.getContent() as foto1, a.foto2.getContent() as foto2,");
        SQL.Append(" a.foto3.getContent() as foto3, a.foto1.getWidth() as ancho,");
        SQL.Append(" a.foto1.getHeight() as alto,a.foto1.getFileFormat() as formato,");
        SQL.Append(" a.foto1.getMimeType() as mime,a.foto1.getCompressionFormat() as compresion,");
        SQL.Append(" b.foto1.getContentLength() as tamano");
        SQL.Append(" from ARTISTA a where id=:PID");
        _cmd.CommandText = SQL.ToString();
        _cmd.Parameters.Add("PID", OracleDbType.Int32).Value = pid;
        OracleDataReader dr = _cmd.ExecuteReader();
        dr.Read();
        objArtista.Nombre = dr["nombre"].ToString();

        //Foto1
        EFoto objFoto1 = new EFoto();
        OracleBlob objBlob = dr.GetOracleBlob(1);
        objFoto1.Img = new Byte[objBlob.Length];
        objBlob.Read(objFoto1.Img, 0, System.Convert.ToInt32(objBlob.Length));
        objFoto1.Ancho = Convert.ToInt32(dr["ancho"]);
        objFoto1.Alto = Convert.ToInt32(dr["alto"]);
        objFoto1.Formato = dr["formato"].ToString();
        objFoto1.Mime = dr["mime"].ToString();
        objFoto1.Compresion = dr["compresion"].ToString();
        objFoto1.Tamano = Convert.ToInt32(dr["tamano"]);
        objArtista.Fotos.Add(objFoto1);
        //Foto2
        EFoto objFoto2 = new EFoto();
        objBlob = dr.GetOracleBlob(2);
        objFoto2.Img = new Byte[objBlob.Length];
        objBlob.Read(objFoto2.Img, 0, System.Convert.ToInt32(objBlob.Length));
        objArtista.Fotos.Add(objFoto2);
        //Foto3
        EFoto objFoto3 = new EFoto();
        objBlob = dr.GetOracleBlob(3);
        objFoto3.Img = new Byte[objBlob.Length];
        objBlob.Read(objFoto3.Img, 0, System.Convert.ToInt32(objBlob.Length));
        objArtista.Fotos.Add(objFoto3);
    }
}

```

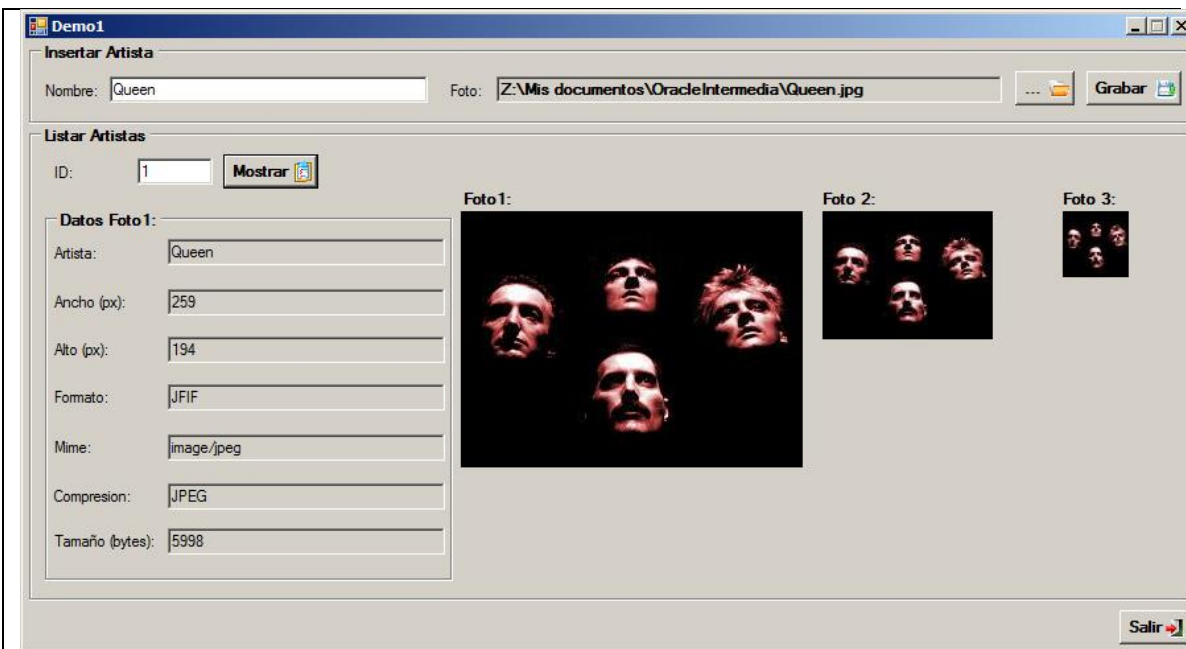


```

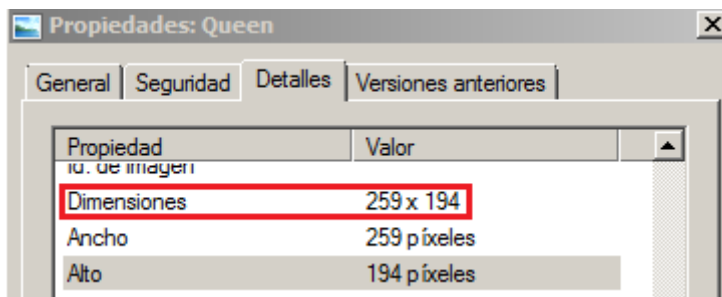
catch
{
    objArtista = null;
}
finally
{
    if ((_cn != null) && (_cn.State == ConnectionState.Open))
    {
        _cn.Close();
        _cmd.Dispose();
        _cn.Dispose();
    }
}
return objArtista;
}
}

```

A continuación se muestra el resultado final de la aplicación.



Validamos la información de la aplicación con los datos obtenidos desde el explorador de Windows:



Podemos apreciar que ambos se obtienen la misma información sobre las dimensiones del archivo.

Demo #2 - Implementando Streaming con Oracle Multimedia

En la siguiente demo se implementará un servidor de Streaming el cual utilizará los videos almacenados con Oracle Multimedia sobre la base de datos Oracle 12c.

En esta implementación el servidor de Streaming se encontrará en un servidor Windows mientras la base de datos Oracle 12c seguirá ejecutándose sobre un sistema operativo Oracle Linux 5.10.

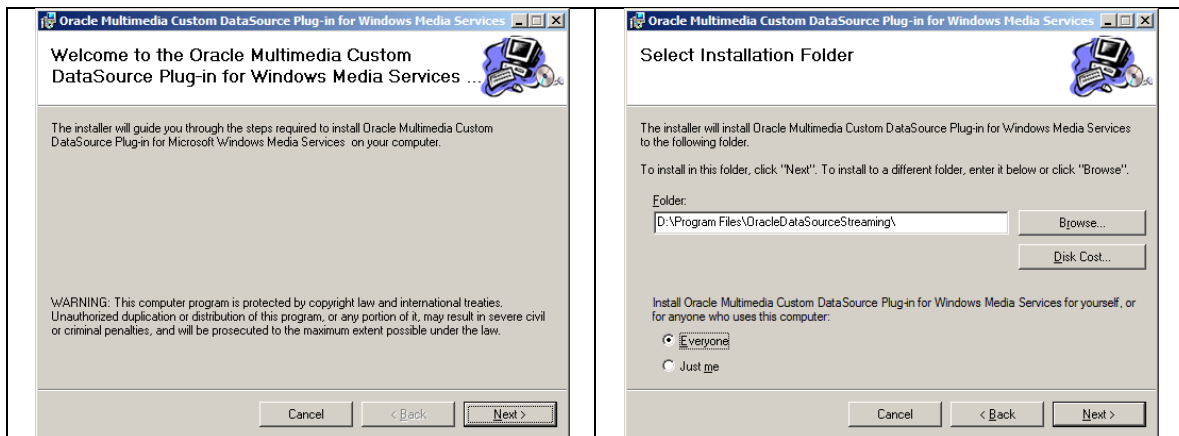
Las especificaciones técnicas que fueron instaladas en el servidor Streaming fueron:

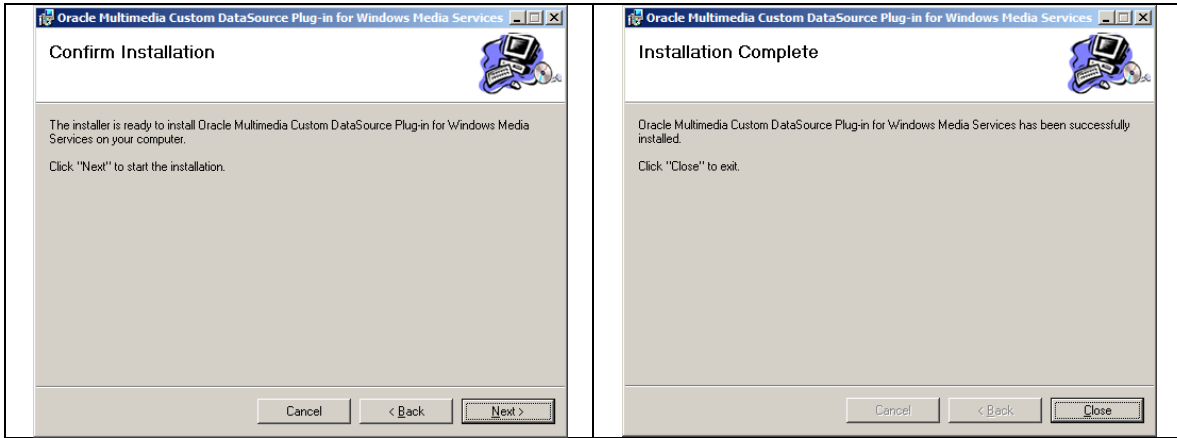
- Windows Server 2008 R2 Enterprise x32 bits.
- Microsoft Windows Media Services 9.6.
- Oracle Client 11.2.
- Oracle Plugin for Microsoft Windows Media Streaming Services, disponible en:
<http://www.oracle.com/technetwork/database/database-technologies/multimedia/downloads/wms-093032.html>

Nota:

- El plugin requiere como requisitos mínimos lo siguiente:
 - Windows Server 2003 Enterprise x32 bits.
 - Microsoft Windows Media Services 9.
 - Oracle Client 10.2.
- A la fecha de ser escrito este artículo, el Plugin únicamente es compatible sobre un sistema operativo de x32 bits. Debido a esta limitación, es posible desarrollar un Plugin a partir del Kit de Desarrollo de Microsoft, mayor detalle en:
[https://msdn.microsoft.com/en-us/library/windows/desktop/dd892588\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dd892588(v=vs.85).aspx)

Paso 1 - Instalación del Plugin.

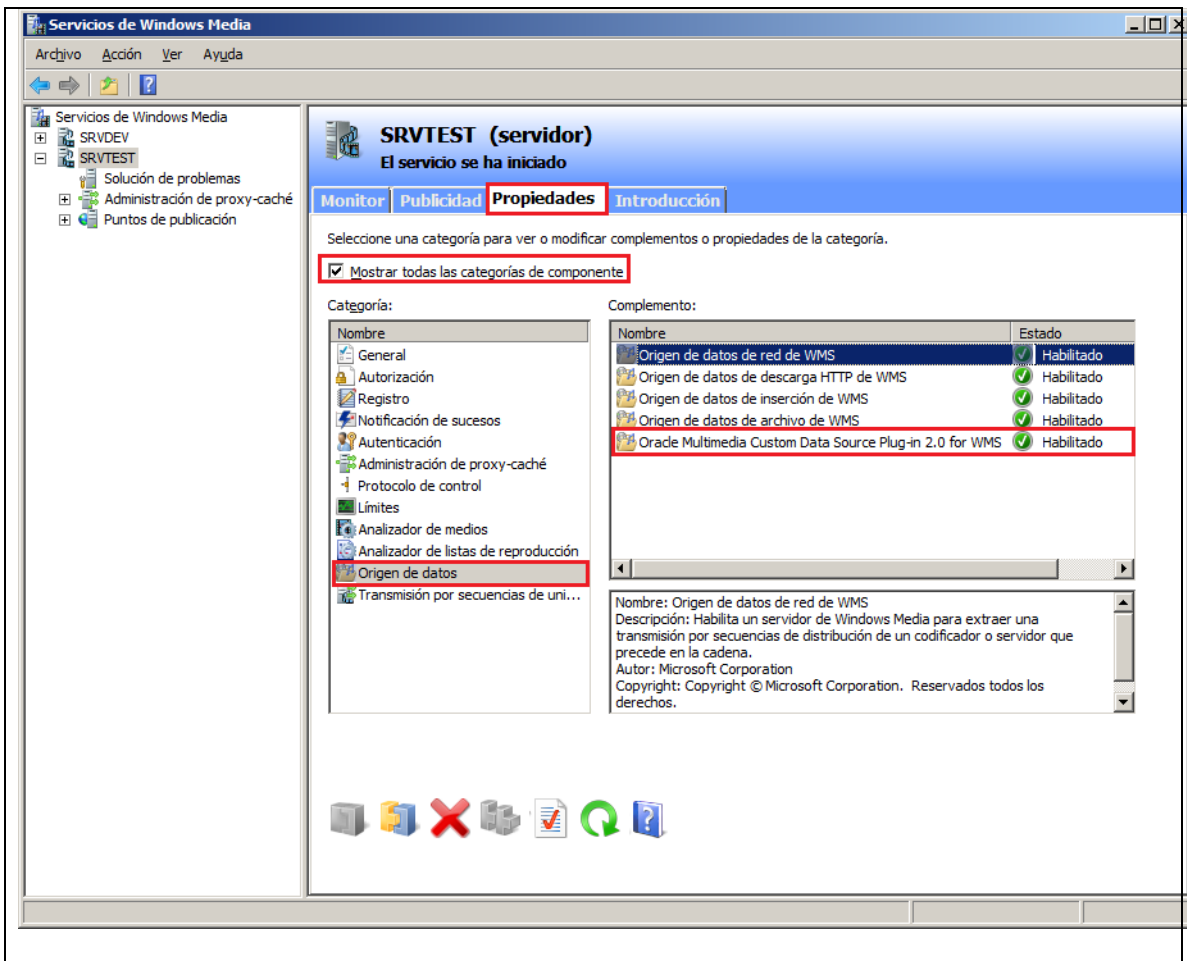




Instalado el Plugin debemos reiniciar los servicios de Windows Media Services.

Paso 2 - Habilitación del Plugin.

Por defecto el Plugin se encontrará en estado deshabilitado. Para modificar el estado debemos hacer clic derecho y seleccionamos la opción Habilitar.



Paso 3 - Creamos una columna de tipo ORDVideo en nuestra tabla Artista.

```
SQL> alter table ARTISTA add video ORDVideo;

Table altered.

SQL> desc ARTISTA;
Name                               Null?    Type
-----
ID                                   NOT NULL NUMBER
NOMBRE                              VARCHAR2 (25)
FOTO1                                PUBLIC.ORDIMAGE
FOTO2                                PUBLIC.ORDIMAGE
FOTO3                                PUBLIC.ORDIMAGE
VIDEO                                PUBLIC.ORDVIDEO
```

Actualizamos la columna VIDEO con un video en formato wmv.

```
SQL> create or replace directory DIR_VIDEOS as '/u01/videos';

Directory created.

SQL> declare
  v_video ordsys.ORDVideo;
  v_ctx blob:=null;
begin
  update artista
  set video = ORDVideo.init()
  where id = 1 returning video into v_video;
  v_video.importFrom(v_ctx,'file','DIR_VIDEOS','queen.wmv');
  update artista set video=v_video where id=1;
  commit;
end;
/

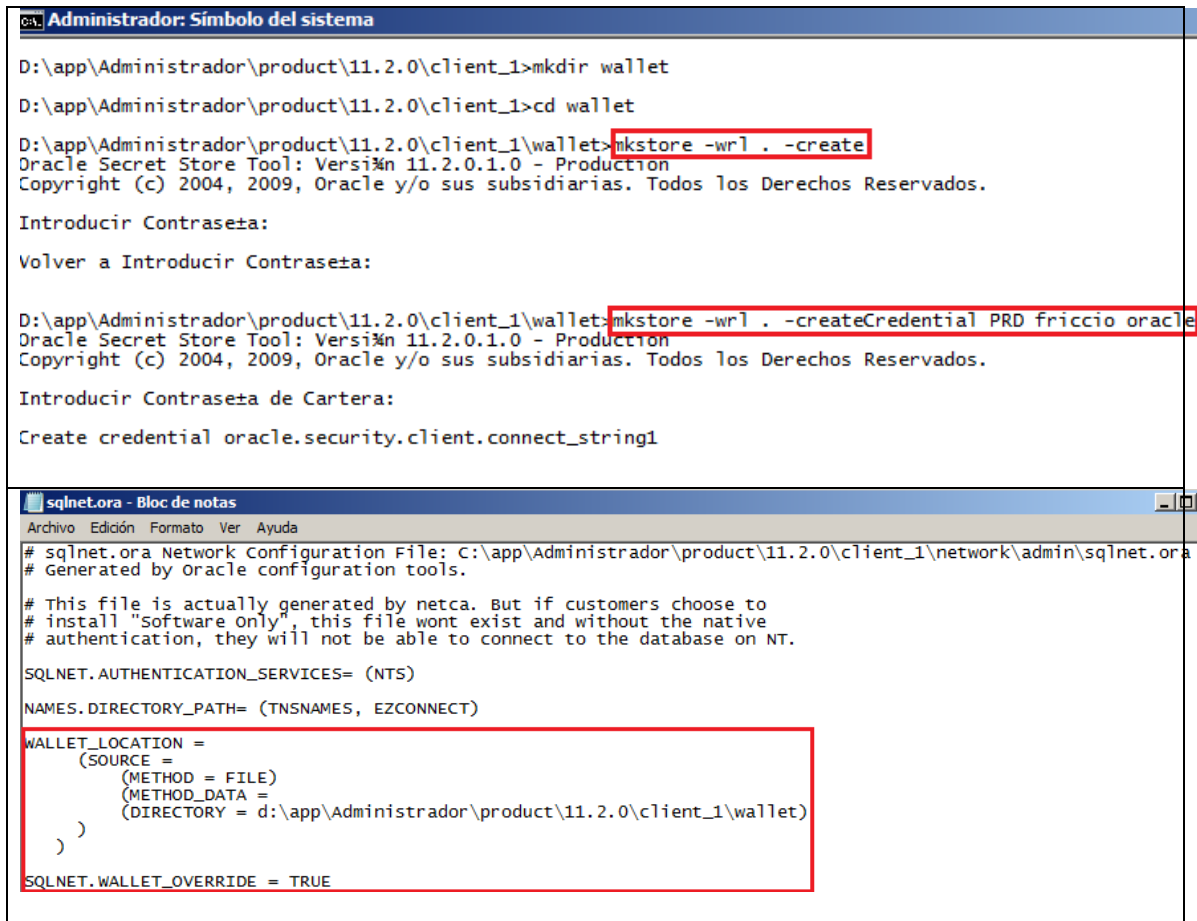
PL/SQL procedure successfully completed.

SQL> select a.video.getContentLength() from Artista a where id=1;

A.VIDEO.GETCONTENTLENGTH()
-----
                          76622985
```

Paso 4 - Creamos las credenciales de Base de Datos con Oracle Wallet.

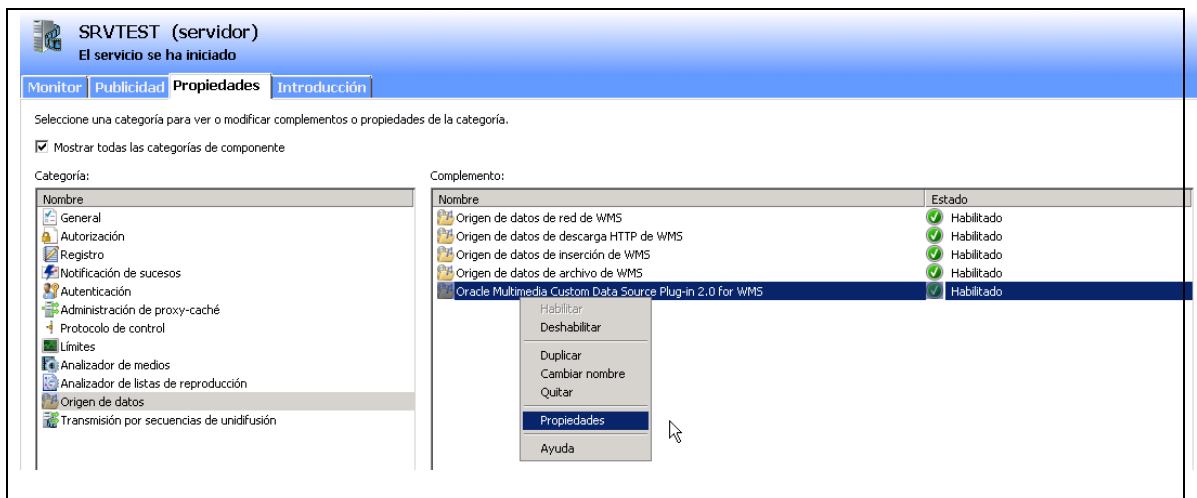
Los siguientes pasos deben ser ejecutados en el servidor Microsoft Windows Media Services utilizando el Oracle Client instalado.



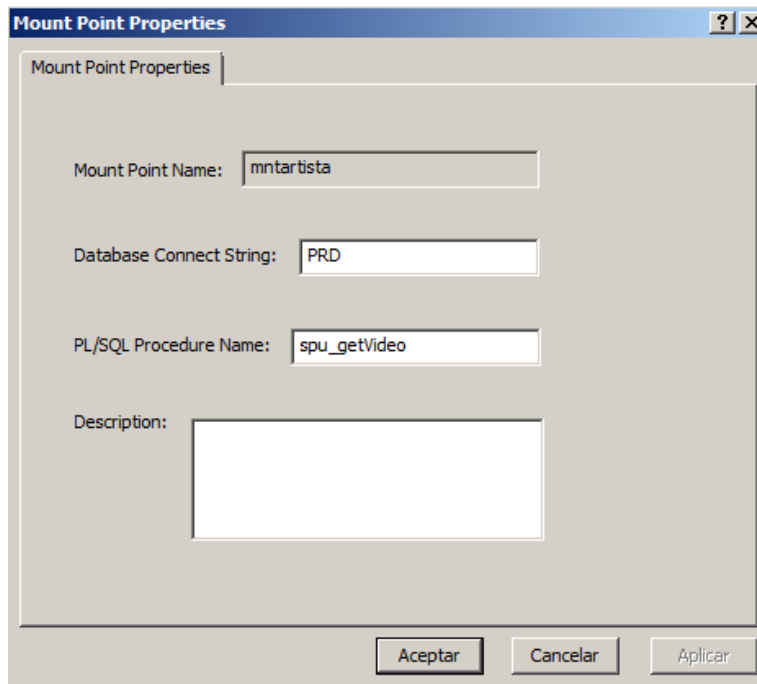
```
Administrador: Símbolo del sistema
D:\app\Administrador\product\11.2.0\client_1>mkdir wallet
D:\app\Administrador\product\11.2.0\client_1>cd wallet
D:\app\Administrador\product\11.2.0\client_1\wallet>mkstore -wr1 . -create
Oracle Secret Store Tool: Versión 11.2.0.1.0 - Production
Copyright (c) 2004, 2009, Oracle y/o sus subsidiarias. Todos los Derechos Reservados.
Introducir Contraseña:
Volver a Introducir Contraseña:
D:\app\Administrador\product\11.2.0\client_1\wallet>mkstore -wr1 . -createCredential PRD friccio oracle
Oracle Secret Store Tool: Versión 11.2.0.1.0 - Production
Copyright (c) 2004, 2009, Oracle y/o sus subsidiarias. Todos los Derechos Reservados.
Introducir Contraseña de Cartera:
Create credential oracle.security.client.connect_string1

sqlnet.ora - Bloc de notas
Archivo Edición Formato Ver Ayuda
# sqlnet.ora Network Configuration File: C:\app\Administrador\product\11.2.0\client_1\network\admin\sqlnet.ora
# Generated by oracle configuration tools.
# This file is actually generated by netca. But if customers choose to
# install "software only", this file wont exist and without the native
# authentication, they will not be able to connect to the database on NT.
SQLNET.AUTHENTICATION_SERVICES= (NTS)
NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)
WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = d:\app\Administrador\product\11.2.0\client_1\wallet)
    )
  )
SQLNET.WALLET_OVERRIDE = TRUE
```

Paso 5 - Creando un Punto de Montaje.



Definimos la cadena de conexión de la base de datos y el procedimiento almacenado que devolverá el contenido de los videos almacenados.



Mount Point Properties

Mount Point Name: mntartista

Database Connect String: PRD

PL/SQL Procedure Name: spu_getVideo

Description:

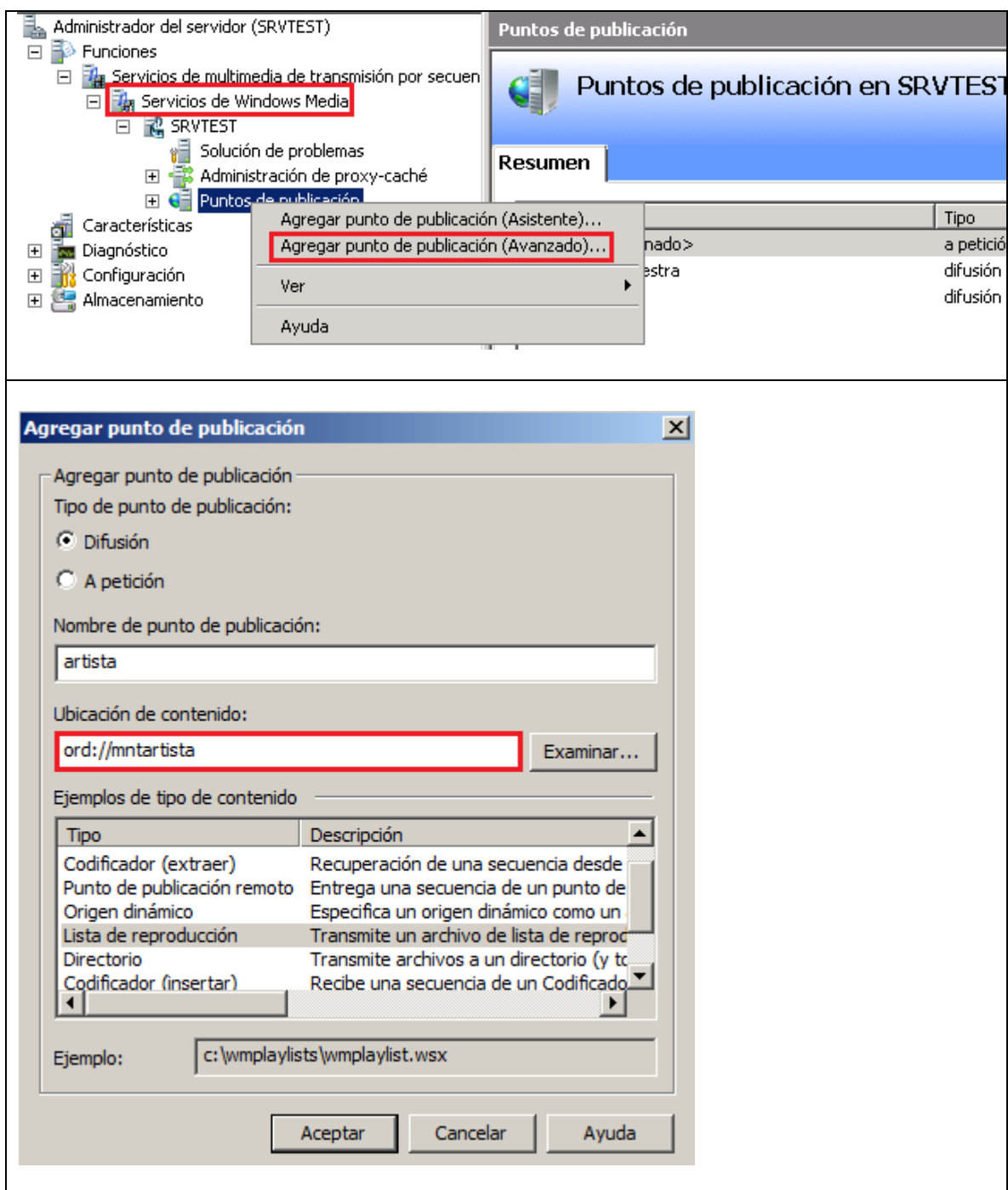
Aceptar Cancelar Aplicar

La definición del procedimiento spu_getVideo a continuación se detalla:

```
SQL> create or replace procedure spu_getVideo (DATA out blob)
is
begin
  select a.video.getContent ()
  into DATA
  from artista a;
end;
/
Procedure created.
```

El procedimiento almacenado debe recibir un parámetro de salida de tipo BLOB o BFILE llamado DATA y opcionalmente un parámetro de salida de tipo VARCHAR llamado MIMETYPE el cual nos brindará la información del tipo MIME de cada video obtenido. Asimismo puede recibir cualquier parámetro de entrada que se requiera implementar.

Paso 6 - Configurando un Punto de Publicación.



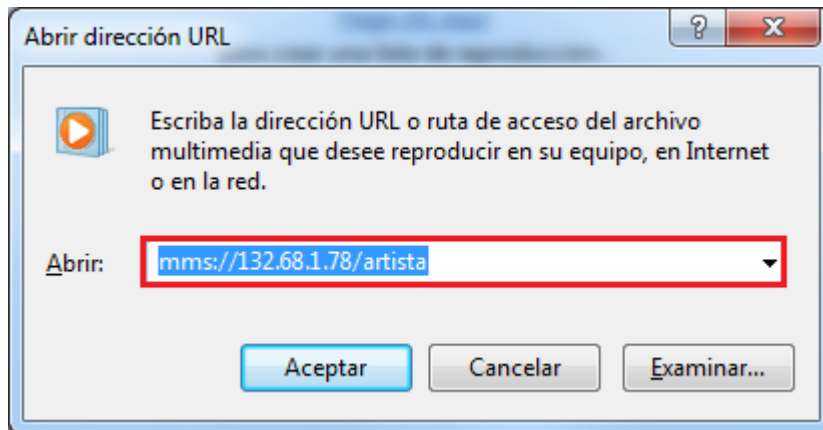
La ubicación del contenido será: `ord://<nombre_punto_montaje>`

Si el procedimiento almacenado hubiera recibido parámetros de entrada, estos deberán ser ingresados de la siguiente manera: `ord://<nombre_punto_montaje>/valor_parámetro_1, etc.`

El resto de parámetros listados al agregar un punto de publicación quedan a disposición para configurarlo acorde a la necesidad requerida.

Una vez configurado el punto de publicación, procedemos a realizar la prueba con el reproductor de Windows Media:

Debemos realizar un clic en el menú del reproductor e ingresamos el nombre del punto de publicación.



Como se puede apreciar, el reproductor reproduce los videos mediante streaming.



Paso 1 - Insertar la imagen

```
SQL> create table IMG_MEDICA
(
  id number generated as identity primary key,
  img ORDDICOM
);

Table created.

SQL> create or replace directory DIR_DICOM as '/u01/dicom';

Directory created.

SQL> insert into IMG_MEDICA (img)
values (ORDSYS.ORDDICOM('FILE', 'DIR_DICOM', 'img1.dcm'));

1 row created.

SQL> commit;

Commit complete.
```

Paso 2 - Revisar metadata de la Imagen

```
SQL> execute ordsys.ord_dicom.setDataModel();

PL/SQL procedure successfully completed.

SQL> set linesize 100
SQL> column IdPaciente format a10
SQL> column FechaEstudio format a10
SQL> column Institucion format a30
SQL> select m.img.getAttributeByName('Patient ID') as IdPaciente,
           m.img.getAttributeByName('Study Date') as FechaEstudio,
           m.img.getAttributeByName('Institution Name') as Institucion
from IMG_MEDICA m;

IDPACIENTE FECHAESTUD INSTITUCION
-----
5555      2005-05-30 Clinique Sainte Marie
```

Como podemos apreciar la metada mostrada desde la base de datos coincide con los valores indicados en la imagen DICOM.

Nota 1: Por cada sesión nueva en la base de datos que desee revisar la metadata de una imagen DCOM deberá ejecutar previamente el procedimiento: ordsys.ord_dicom.setDataModel().

Nota 2: Si deseamos revisar todos los atributos de la imagen DICOM podemos ejecutar el siguiente procedimiento almacenado: extractMetadata, ejemplo:

```
select m.img.extractMetadata('standard') from IMG_MEDICA m
```

Conclusión

Durante todos los ejemplos revisados se ha visto que Oracle Multimedia ofrece una gran flexibilidad y simplicidad en su implementación. También mantiene diversas integraciones con lenguajes de programación, servicios (Streaming, WebDAV, etc) y estándares internacionales (DICOM).

Asimismo, conocemos que contenido multimedia puede ser almacenado en tipos de datos LOB (BLOB & BFILE), pero si requerimos implementar aplicaciones eficientes que requieran manejar con mayor precisión y desempeño los datos multimedia es indispensable el uso de Oracle Multimedia.

Publicado por Ing. Francisco Riccio. Es un IT Architect en IBM Perú e instructor de cursos oficiales de certificación Oracle. Está reconocido por Oracle como un Oracle ACE y certificado en productos de Oracle Application & Base de Datos.

e-mail: francisco@friccio.com

web: www.friccio.com