

Oracle XML DB 11gR1 Repository Features

An Oracle White Paper
October 2007

Introduction	2
Architecture	2
Installation	3
Server Pre-requisites	3
Oracle Software	3
Client Pre-requisites	3
Oracle Software	3
Non Oracle Software	3
Database Configuration Changes	4
Installing the Application	4
Using the demonstration framework	8
Performing the demonstration	10
0.1 Initialize Demo	12
1.1 Show XML Schema (XMLSPY)	14
1.2 View XML Schema (XFILES)	16
2.1 List Repository Events	17
2.2 Repository Event Configuration File	18
2.3 Repository Event Package	19
2.4 Register Event Manager	21
3.1 Repository Event Queue	23
3.2 Image Processor	25
4.1 Create Folder	30
4.2 Count Metadata	30
4.3 Copy Images (WebDAV)	31
4.4 Recount Metadata	33
5.1 Query Metadata	35
5.2 Expand Metadata	37
6.1 View Resource XML	42
6.2 View Images (XFILES)	43
6.3 View Image Categories (XFILES)	49
6.4 View Image Categories (WebDav)	51

Introduction

The Oracle XML DB basic demonstration high-lights some of the advanced features of the Oracle XML repository DB including

- Extracting Metadata from content stored in the Oracle XML DB repository using Oracle Database 11g's advanced multimedia capabilities.
- Managing XML Schema-based metadata using the Oracle XML DB repository
- Using Oracle XMD event to automate the processing of resources stored in the Oracle XML DB repository
- Metadata extraction using Oracle Database 11g Release 1's advance multimedia capabilities.
- Asynchronous process of resources using Oracle Database 11g Advanced Queuing
- Managing documents with embedded HTML content using Oracle XML DB.
- Interactive editing and partial update of XML documents using AJAX and Web Services.

Architecture

Installation

Server Pre-requisites

The following software is required to run the Oracle XML DB basic demonstration

Oracle Software

- Oracle Database 11g release 11.1.0.6.0 or later, with the XML DB, Oracle Text and Oracle JVM features installed.

Client Pre-requisites

The installation process uses an HTML application, VB Scripting and the HTTP protocol to upload the source code into the Oracle XML DB repository, SQL*PLUS scripts are used to re-configure the Oracle XML DB repository to support the demonstration.

The following software is required to install the Oracle XML DB Repository features demonstration.

Oracle Software

- Oracle Client (SQL*PLUS and Oracle Net Services) 11.1.0.6.0 (Production) or later. The application can be installed into a remote database, however both SQL*PLUS and Oracle Net Services must be installed on the client machine in order to perform a remote install. Currently remote installs are only supported on the Windows platform.
- Oracle XML DB X-Files application. The basic demonstration runs inside an AJAX-framework that is included as part of the Oracle XML DB X-Files application. Starting with Oracle Database 11g the X-Files demonstration must be downloaded and installed before installing the basic features demonstration.

Non Oracle Software

- Microsoft Internet Explorer 7.0 with the latest service packs. The X-Files application has not been tested with any release of other browsers including Firefox, Mozilla, Netscape, Safari or Opera.
- Microsoft Windows Scripting Technologies version 5. Windows Scripting is used by the installation process. You can verify the version of Windows Scripting installed on your machine by opening a command prompt and typing the command cscript.
- Microsoft Core XML Services (MSXML) versions 4.0sp2 and 6.0 are required in-order install the demonstration. It is also required by the AJAX based framework that is used to run the demonstration.

At the time of writing the latest version of this software can be downloaded from <http://www.microsoft.com/downloads/Search.aspx?displaylang=en>

- XMLSPY: XMLSPY is an IDE for XML from Altova Corporation. An evaluation copy of this product can be obtained from Altova's website at <http://www.altova.com>.
- Microsoft Windows XP Professional with Service Pack 2.

Database Configuration Changes

Installing the Repository Features demonstration application will make the following changes to the configuration of the target database.

- **Oracle XML DB HTTP and FTP Server:** Installing the basic demonstration will enable the database's native HTTP and FTP Servers. Ensure that you have read the XML DB documentation regarding the use of the XML DB HTTP Server before installing this application into a database that contains production data. This information can be found in the Oracle XML DB Developers guide.
- **Database Native Web Services:** The basic demonstration is executed using an AJAX-based framework. Ensure that you have read the XML DB documentation regarding the use of the Database Native Web Services before installing this application into a database that contains production data. This information can be found in the Oracle XML DB Developers guide.
- **Database Schema XDBMETADATA:** This database owns the objects used by the demonstration. This includes the global XML Schema <http://xmlns.oracle.com/demo/imageMetadata.xsd>, which defines the metadata managed by the demonstration, as well as the tables and packages required to create and manage the metadata. XDBMETADATA is a locked account with XDBADMIN role. It also has read access to some catalog tables owned by XDB and SYS. This account should remain locked.

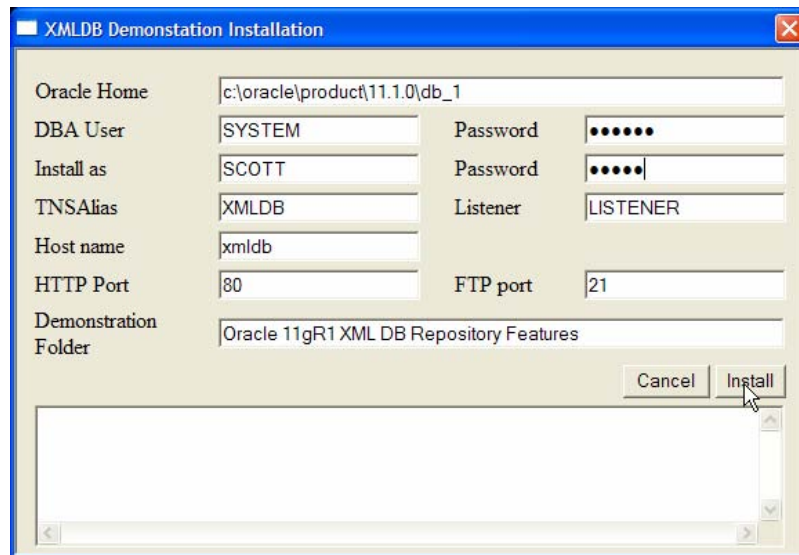
Installing the Application

To install the Oracle XML DB 11g Release 1 Repository Features demonstration unzip the file XMLDB_11GR1_Repos.zip into a folder of your choice. Ensure that there are no spaces in any of the parent folder names. After unzipping this file the target folder should contain a subfolder called repositoryFeatures. This folder will contain subfolders SQL, setup and Install.

To start the installation, execute the file install.hta found in the install subfolder.

```
C:\basicFeatures\install>install.hta
```

This will launch the installer dialog. The installation process is an HTML application.



The default values for the dialog are obtained from the file InstallationParameters.xml. The contents of this file are as follows:

```
<installationParameters>
  <shortCutFolderName>
    Oracle 11gR1 XML DB Repository Features Demonstration
  </shortCutFolderName>
  <oracleHome>c:\oracle\product\11.1.0\db_1</oracleHome>
  <dba>SYSTEM</dba>
  <oracleUser>SCOTT</oracleUser>
  <oraclePassword/>
  <tnsAlias>XMLDB</tnsAlias>
  <listener>LISTENER</listener>
  <sqlPort/>
  <hostName>xmldb</hostName>
  <httpPort>80</httpPort>
  <ftpPort>21</ftpPort>
  <parameter name="%BASEFOLDER%" value="/publishedContent/imageMetadata" />
</installationParameters>
```

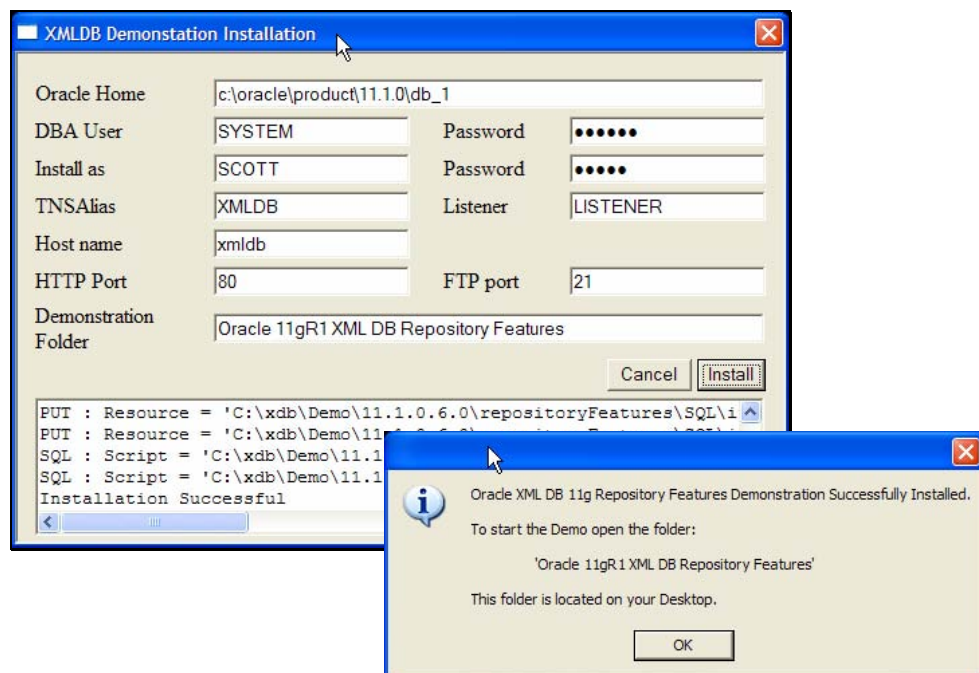
You can modify the default values by modifying the contents of InstallationParameters.xml before starting the installation. To install the demonstration, start the installer, modify any values that are not correct for your environment, enter the DBA and installation user's passwords and click install. Remember that in Oracle Database 11g passwords are case-sensitive. To cancel the installation click cancel.

The meaning of each parameter / field is given in the following table

< shortCutFolderName >	The name of the folder that will contain the set of Icons used to run the demo. This folder will be placed on the user's desktop.
<oracleHome>	The location of the Oracle Home on the local computer. An Oracle client installation is required to run the installation process.
<dba>	The name of a user with DBA capabilities which can be used to install the demonstration. Normally this will be SYSTEM, but any DBA is acceptable. The password for the DBA user can only be entered using the installation dialog.
<oracleUser>	The database user that will be used to run the demo. This user should already exist and be able to connect to the database. The installation process will grant this user the following privileges: session, unlimited tablespace, create table, create view, create any directory, drop any directory. These privileges are required to run the demonstration,
<oraclePassword>	The password for the demonstration user.
<tnsAlias>	The tnsAlias that can be used to connect to the target database instance
<listener>	The name of the listener associated with the database instance. A listener should not service more than one database when Oracle XML DB protocols are in use.
<hostname>	The name of the machine running the Listener.
httpPort	The port used by the Oracle XML DB HTTP Service. The port must not already be in use by any other service. The installation process will configure the database to use this HTTP port. If this port is a privileged port on a unix system the listener.ora must be configured appropriately.
ftpPort	The port used by the Oracle XML DB FTP Service. The port must not already be in use by any other service. The installation process will configure the database to use this HTTP port. If this port is a privileged port on a unix system the listener.ora must be configured appropriately.

Clicking Install will start the installation. The installation will verify that it can connect as the DBA, and as the demonstration user using SQL and HTTP. Once connectivity has been verified the demonstration will be installed. If the connectivity tests fail the installation will not proceed.

The progress of the installation will be shown in the status window at the bottom of the installer dialog. When the installation is complete the following message will be displayed.



Click OK to dismiss the dialog and then cancel to exit the installer.

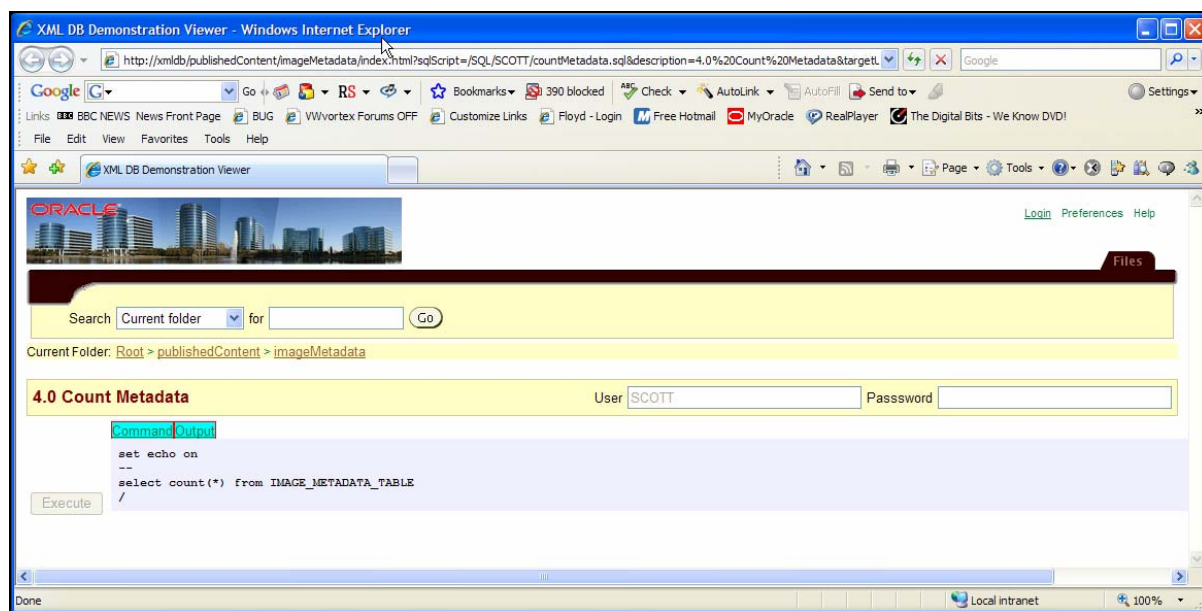
Using the demonstration framework

The SQL based portions of the demo are presented using the Oracle XML DB demonstration framework. This is an AJAX-based application uses the Database Native Web Services (DBNWS) feature of Oracle Database 11g to execute SQL scripts. The demonstration framework is installed as part of the X-Files application, which can be downloaded from the XML DB page on OTN. Please make sure you have the latest version of the X-Files application installed before running this demonstration.

The XML DB demonstration framework is launched by clicking the icons in the demonstration folder. If the browser currently owns an authenticated HTTP connection to the Oracle XML DB Database, the framework will automatically execute the SQL script. If the browser does not own an authenticated HTTP connection to the Oracle Database, or the current session does not belong to the correct user, the framework will prompt for a password before running the script. Entering the correct password will execute the script.

If the demonstration framework encounters a pause command, the CONTINUE button will be enabled. Click the CONTINUE button to continue executing the script. When the script is complete the CLOSE button will be enabled. Click the close button will close the framework session. If at least one framework session is left open, subsequent windows will be able to inherit the HTTP connection, avoiding the need to enter a password each time a new framework session is opened.

The following screen shot shows the demonstration framework ready to run a script. The framework is waiting for the password to be entered. The first SQL command in the script is displayed in the command area. The cursor is positioned in the Password field and the EXECUTE button is disabled.



Entering a valid password will automatically execute the SQL statement.

The framework consists of interleaved command and output areas.

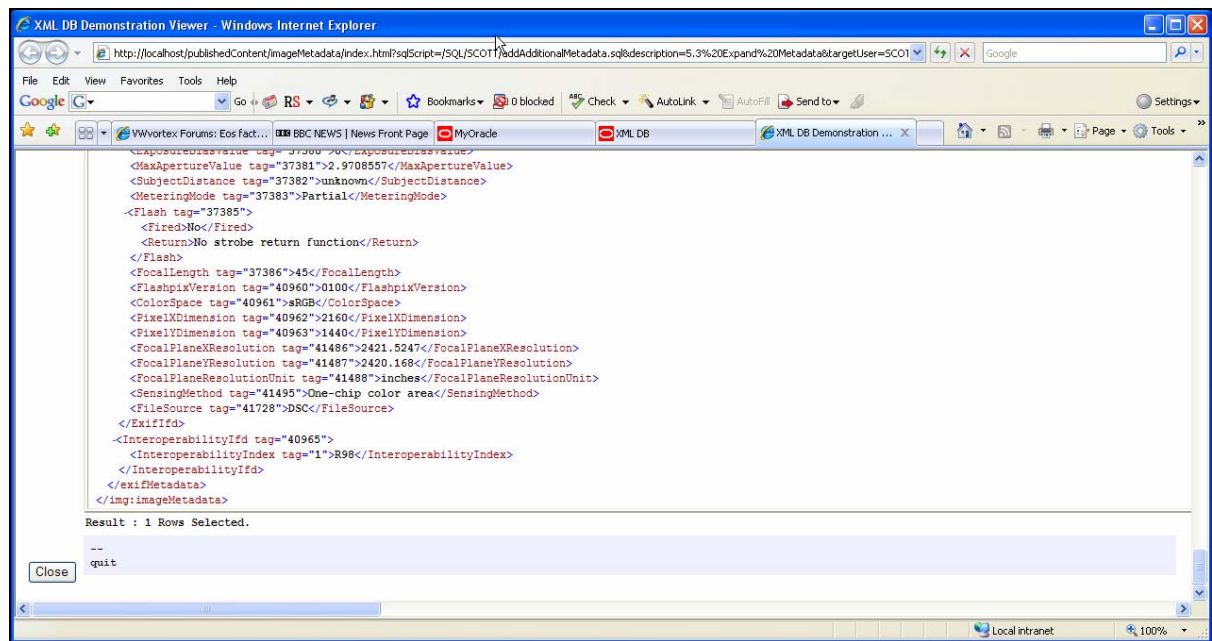
The command area shows the current SQL command. If the SQL command contains more lines than can be displayed in the default command area a vertical scroll bar will appear. The scroll bar can be used to scroll the contents of the command area. When the scroll bar is present clicking the Command tab will expand the command area to show the entire command. Once the command area is expanded clicking the Command tab again will revert to the default size.

The output area shows the results of the query. If the script includes a **set autotrace on explain** command the output area can also show the query plan for the current query. When the query plan is available two additional tabs, Result and Plan will be displayed. Click Result to see the query output, Click Plan to see the query plan.

If the command generates more output than can be displayed by the default output area a vertical scroll bar will appear. The scroll bar can be used to scroll the contents of the output area. When the scroll bar is present clicking the Output tab will expand the output area to show as much of the output as possible. Once the output area has been expanded clicking the Output tab again will revert to the default size. This behavior also occurs when the query plan is displayed in the output area.

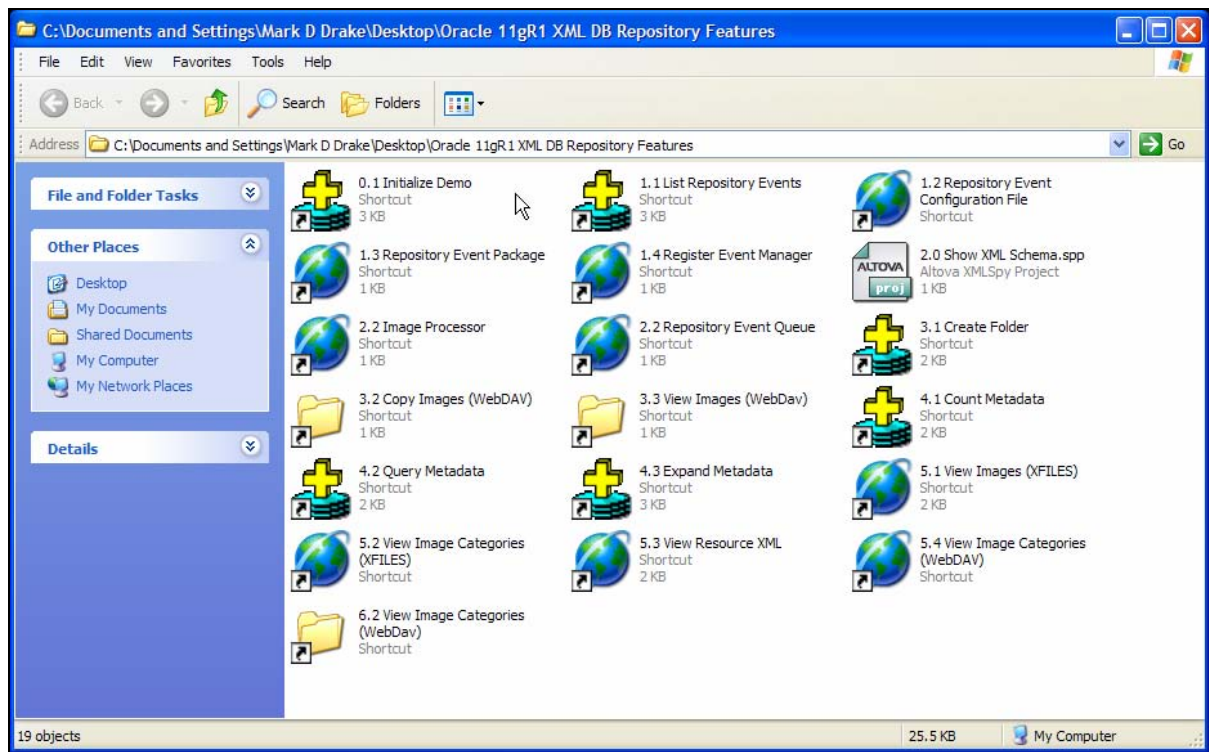
One major advantage of the demonstration framework is that it is completely XML aware. When a query executes XML, the XML will be displayed complete with all of its tag information intact. Elements with complex content will be marked with a - icon. Clicking the - icon will close up the children of the element, and replace the - icon with a + icon. Clicking the + icon will expand the subtree for the element.

The following screen shows the demonstration framework after completion of a script. The output area contains an XML document. Since the script is complete the CLOSE button is enabled.



Performing the demonstration

The installation creates the folder “Oracle 11gR1 XML DB Basic Features Demonstration”. This folder is located on the Desktop. The folder contains the following Icons.



The icons are numbered. To run the demonstration, click each icon in turn.

The first part of this demonstration shows how to define additional categories of metadata for Oracle XML DB resources. Extended metadata makes it easy to categorize content stored in the Oracle XML DB repository and to search and process content based on extended metadata. Oracle XML DB allows arbitrary metadata to be attached to a resource. It also allows categories of metadata to be defined using an XML Schema.

The demonstration uses an XML Schema to define a class of metadata appropriate for use with digital photographs. The metadata is based on the EXIF standard that is used by most camera manufacturers to record information about the camera and settings used to take a picture. The EXIF standard provides an XML Schema that defines the XML representation of the metadata captured when a picture is taken.

The second part of the demonstration shows how to automate the processing of resources using Oracle XML DB repository events. Repository events allow PL/SQL or Java code to be automatically executed when an actions occurs in the Oracle XML DB repository. Events can be defined for CRUD (Create, Retrieve, Update or Delete) operations on files and folders. Oracle XML DB events enable an

intelligent programmable repository in the same way that database triggers enables an intelligent, programmable database.

The demonstration will define a set of events that will automatically extract metadata when a digital photograph is loaded in the Oracle XML DB repository. The metadata will be extracted using Oracle Database 11g's Multimedia features. The metadata will be stored as schema-based metadata in the Oracle XML DB repository.

Processing performed using a repository event is like processing performed in a database triggers. It is executed as part of the transaction that triggers the event. The operation that raised the event will not complete until the processing defined using events has completed. This means that care should be taken when defining events to ensure that the processing cannot lead to delays that will force protocol time-out conditions to occur.

Since the processing associated with metadata extraction is significant, this demonstration uses Oracle Advanced Queuing to provide an asynchronous infrastructure for performing the extraction process. The overhead incurred as part of the event is the minimal overhead required to create and queue a message. The metadata extraction is performed once the message has been de-queued, in a separate transaction which takes place after the resource has been created. This will ensure that the overhead associated with metadata extraction does not result in protocol time-out errors.

The metadata extraction process reads the content of the new resource and extracts the embedded metadata using function GETMETADATA in package ORDIMAGE. This function returns the metadata as an XML document compliant with the EXIF XML Schema. The metadata is stored as a document in table IMAGE_METADATA_TABLE. The metadata is then associated with the resource.

The metadata extraction process also performs some simple categorization of the resource based on the metadata. The Images is published under two separate folder trees in the Oracle XML DB repository. The first category is based on the camera that was used to take the picture. Images are categorized under manufacturer and model. The second category is based on the data the picture was taken. Images are categorized under Year and Month. Categorizing files this way makes it possible to locate a file based on its content, rather than its name.

Installing the demonstration registers the XML Schema and creates the repository events that are used by the demonstration. It also creates the Oracle Database 11g Advanced Queue that will be used to provide asynchronous extraction of the metadata contained in the images.

0.1 Initialize Demo

This step initializes the demonstration.

Some of the tasks performed by this step required XDBADMIN or DBA privilege, so this step is run as the DBA used to install the demonstration. The DBA must have permission to use Database Native Web Services (DBNWS). The permissions required to use DBNWS can be granted by calling procedure enableWebServicesDemo in the package XDBPM.XDB_DEMO_HELPER_11100. The procedure expects a single argument which is the name of the user to grant the permissions to. Since a DBA cannot grant themselves privileges this procedure must be invoked by another DBA.

Click the icon to launch the XML DB demonstration framework and run the SQL script. The username field will be pre-filled with the name of the DBA user and the password field will be empty. Type the password and hit enter or click outside of the password field. If the password is entered correctly the form will refresh and the script will execute. If an HTTP authentication dialog appears after entering the correct password then Database Native Web Services have not been correctly configured for the user.

The script undoes all of the actions performed while running the demonstration.

```
Command Output
--
set echo on
--
declare
res boolean;
begin
    if (dbms_xdb.existsResource('/home/SCOTT/ImageLibrary')) then
        dbms_xdb.deleteResource('/home/SCOTT/ImageLibrary', dbms_xdb.DELETE_RECURSIVE_FORCE);
    end if;

    if (dbms_xdb.existsResource('/publishedContent/allImages/camera')) then
        dbms_xdb.deleteResource('/publishedContent/allImages/camera', dbms_xdb.DELETE_RECURSIVE_FORCE);
    end if;

    if (dbms_xdb.existsResource('/publishedContent/allImages/dateTaken')) then
        dbms_xdb.deleteResource('/publishedContent/allImages/dateTaken', dbms_xdb.DELETE_RECURSIVE_FORCE);
    end if;

    if not dbms_xdb.existsResource('/publishedContent/allImages') then
        res := dbms_xdb.createFolder('/publishedContent/allImages');
    end if;

    dbms_xdb.setAcl('/publishedContent/allImages', '/sys/acls/bootstrap_acl.xml');

    res := dbms_xdb.createFolder('/publishedContent/allImages/camera');
    res := dbms_xdb.createFolder('/publishedContent/allImages/dateTaken');

end;
/
PL/SQL procedure successfully completed.
```

- Remove folder ImageLibrary from the demonstration user's home folder.
- Remove folders camera and dateTaken from folder allImages. Any content in these folders will also be removed.
- Re-create folder allImages if it does not exist. Recreate folders camera and dateTaken.

- Secure folder allImages with the bootstrap ACL. The bootstrap ACL is one of the pre-defined ACLs shipped with the Oracle XML DB repository. It provides read-only access to the resource it is applied to. The bootstrap ACL called bootstrap_acl.xml and is located in the folder /sys/acls.
- Create folders camera and dateTaken in folder allImages. These folders will inherit the use of the bootstrap ACL from allImages.

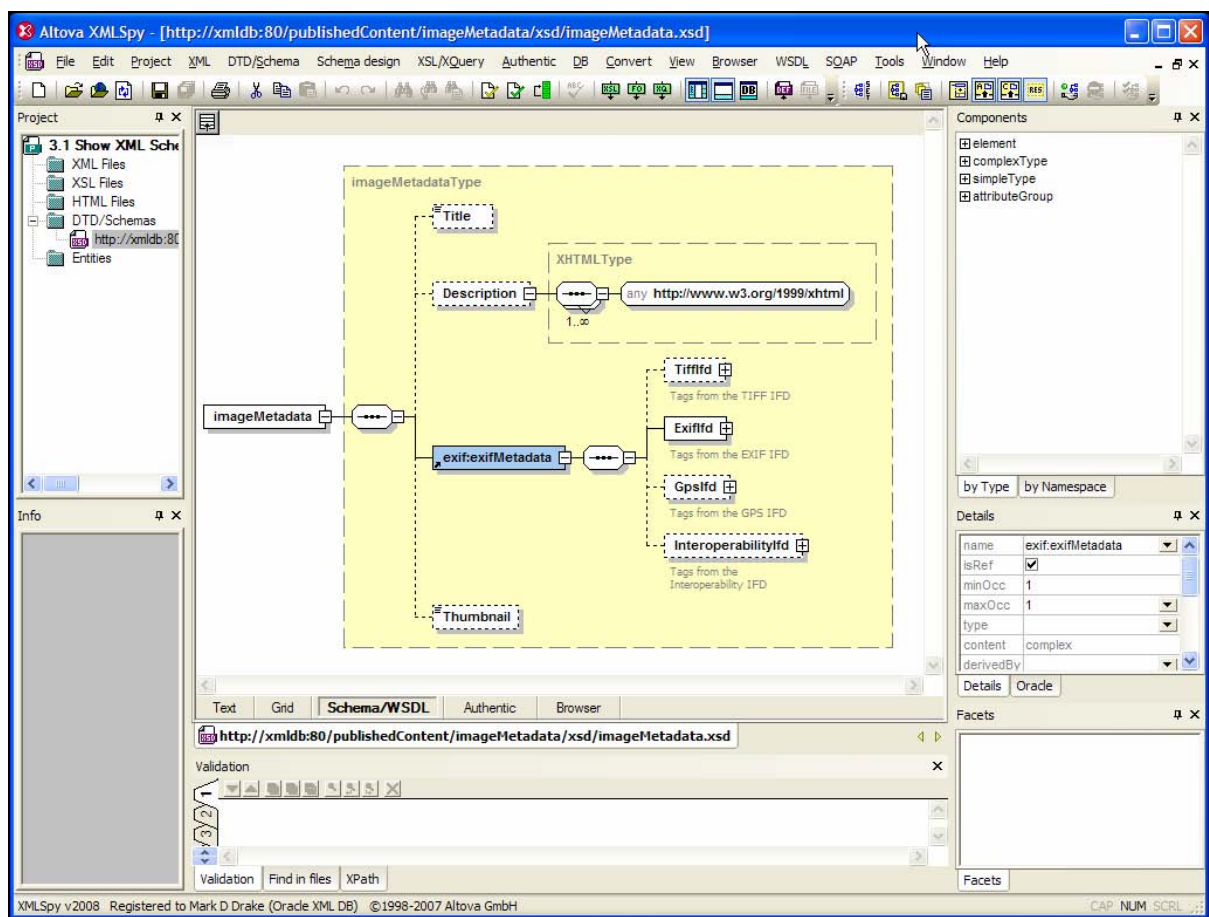
1.1 Show XML Schema (XMLSPY)

This step shows the XML Schema that defines the class of metadata that will be captured for digital photographs. The XML Schema is loaded into the repository as part of the demonstration installation.


Click the icon to open XMLSPY. When the application opens click the + sign next to the DTD/Schemas entry in the Project Panel. This branch will contain a single XML Schema, accessible via the following URL:

<http://hostname:httpPort/publishedContent/imageMetadata/imageMetadata.xsd>.

Double-click this item. XMLSPY will prompt for a username and password. Enter the demonstration user's username password and click OK. XML Spy will open the XML Schema using HTTP and display a list of the elements and types defined by the XML Schema.



XMLSPY provides a powerful, graphical, easy to use interface for creating and editing XML Schemas. XMLSPY supports both the WebDAV and FTP protocols allowing it direct access to content stored in Oracle XML DB.

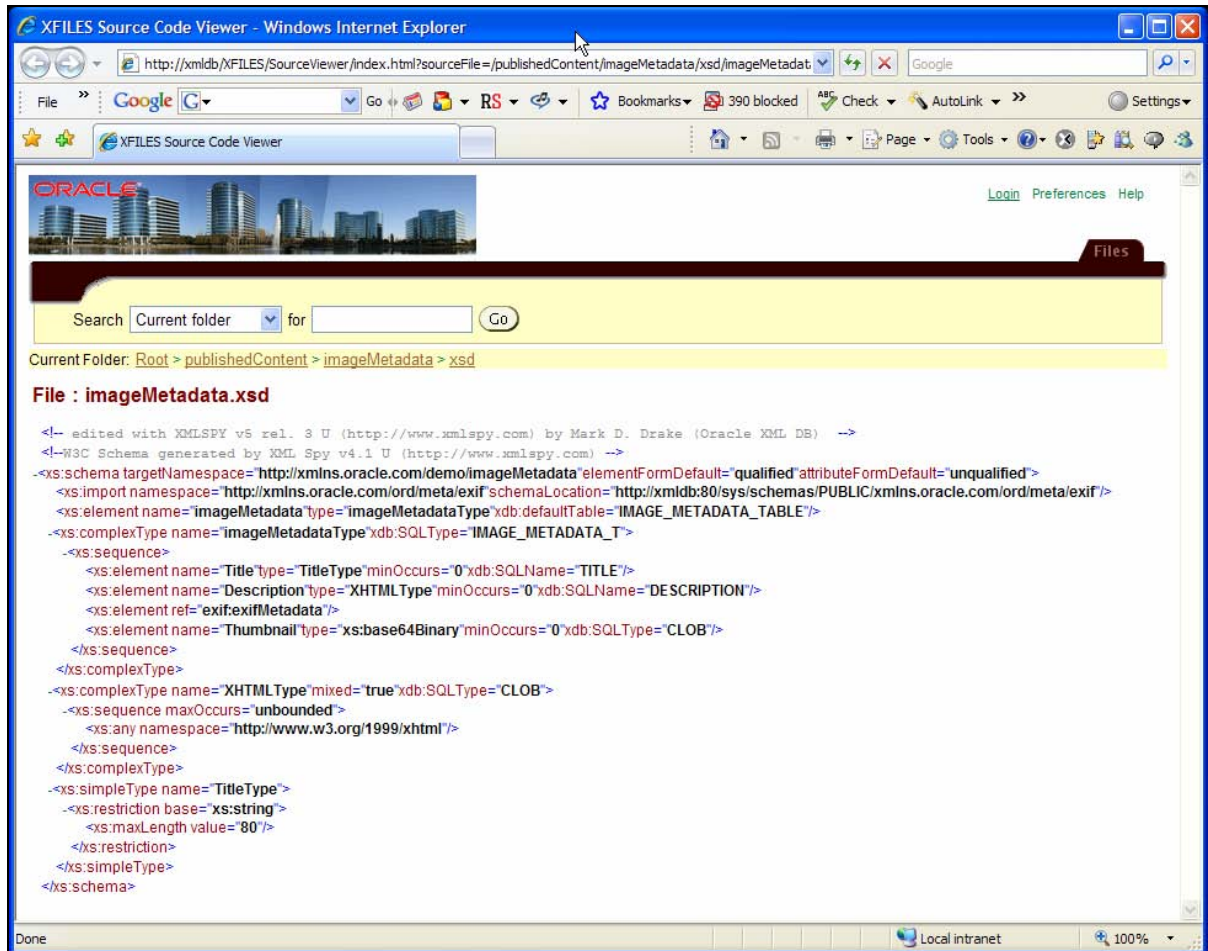
Click the  control next to element imageMetadata. Click the + sign next to element Descripton and the + sign next to element exifMetadata. XMLSpy displays a graphical representation of the XML Schema.

The XML Schema is a very simple XML Schema. It defines four elements, Title, Description, Metadata and Thumbnail.

- The target namespace for all of the elements are defined by this XML schema is <http://xmlns.oracle.com/demo/imageMetadata>
- Title is an 80 character string.
- Description is defined as instance of the complexType XHTMLType.
- XHTMLType is defined as a collection of XML Schema any type elements, restricted to elements in the XHTML namespace, <http://www.w3.org/1999/xhtml>. This allows description to consist of a block of XHTML mark-up.
- The entire fragment is stored as a single CLOB. This avoids the overhead of converting between the XHTML and the internal object format is avoided when storing or retrieving this type of content.
- There is no benefit to storing XHTML content using the internal object representation. It is very unlikely that the content of an XHTML fragment will be subjected to path based searches or fragment level extraction or update operations. Full text indexes can be created on the fragment to assist with searching the content. Updates to the content will typically involve replacing the entire fragment, not some piece within it.
- exifMetadata is defined as an reference to global element exifMetadata. This element is defined by XML Schema [http:// hostname:httpPort /xmlns.oracle.com/ord/meta/exif](http://hostname:httpPort/xmlns.oracle.com/ord/meta/exif). The EXIF XML schema is imported by this XML Schema. The EXIF XML Schema is pre-registered with Oracle XML DB when the Oracle Multimedia feature is installed. It defines the format of the EXIF metadata that can be obtained using class ordsys.ordimage.
- Thumbnail is defined as a single block of base64binary content stored in a CLOB. It is not currently used by this version of the demonstration.

1.2 View XML Schema (XFILES)

This step uses the source code viewer component of the X-Files application to show the XML Schema. The XML Schema is registered with Oracle XML DB as part of the demonstration installation. Click the icon to launch the XML DB Source Code Viewer and display the contents of the file.



- The default table for element imageMetadata is defined as IMAGE_METADATA_TABLE.
- The XML Schema is registered as a global XML Schema. This allows any user of the XML DB repository to create an instance of the document defined by the XML Schema. The user registering the XML Schema must have been granted XDBADMIN.
- Since the XML Schema defines metadata rather than content, parameter ENABLEHIERARCHY is set to DBMS_XMLSCHEMA.ENABLE_HIERARCHY_RESMETADATA when registering the

XML Schema. The code used to register the XML Schema can be found in the Oracle XML DB repository at `/publishedContent/imageMetadata/SQL/install/registerSchema.sql`.

2.1 List Repository Events

Events are defined using a Resource Configuration document. Resource Configuration documents are registered with the Oracle XML DB using package `DBMS_RESCONFIG`. Events are configured on a repository wide basis, or on a specific folder. Multiple Resource Configuration documents can be attached to a single target.

This step shows the set of repository-wide Resource Configuration documents for the repository. Click the icon to launch the XML DB demonstration framework and run the SQL script.

Command Output	
<pre>set echo on -- select rownum config_id, column_value config_path from table(dbms_resConfig.getRepositoryResConfigPaths()) /</pre>	
CONFIG_ID	CONFIG_PATH
1	/sys/xs/userrc.xml
2	/sys/xs/rolesetrc.xml
3	/sys/xs/drolerc.xml
4	/sys/xs/rolerc.xml
5	/sys/xs/frolerc.xml
6	/sys/xs/xdserc.xml
7	/sys/xs/scrc.xml
8	/publishedContent/imageMetadata/xml/imageEventConfiguration.xml
Result : 8 Rows Selected.	

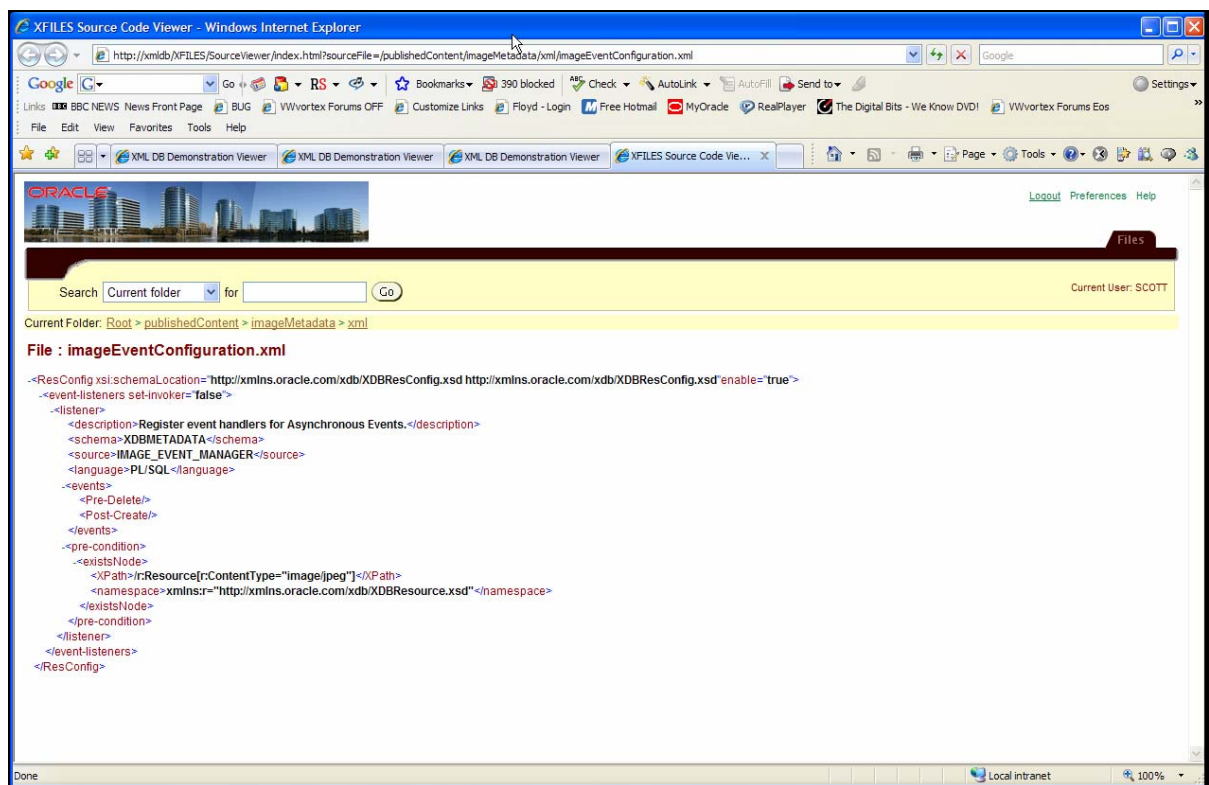
Out of the box Oracle XML DB ships with 7 pre-configured repository-wide Resource Configuration documents. Installing the demonstration adds number 8 in the above list. This document configures special handling for JPEG images.

2.2 Repository Event Configuration File

A Resource Configuration document is an XML document that defines

- The name of the PL/SQL package that will handle the event. Methods on this package will be called when the event occurs
- Which events the package provides handlers for. There are 32 possible events that allow handlers to be associated with repository operation.
- Any conditions that must exist before the handler is invoked. This allows events to be associated with particular types of document.

This step uses the source code viewer component of the X-Files application to show the Repository Configuration document that configures events for digital photographs. The file is loaded into the repository as part of the demonstration installation. Click the icon to launch the XML DB Source Code Viewer and display the contents of the file.

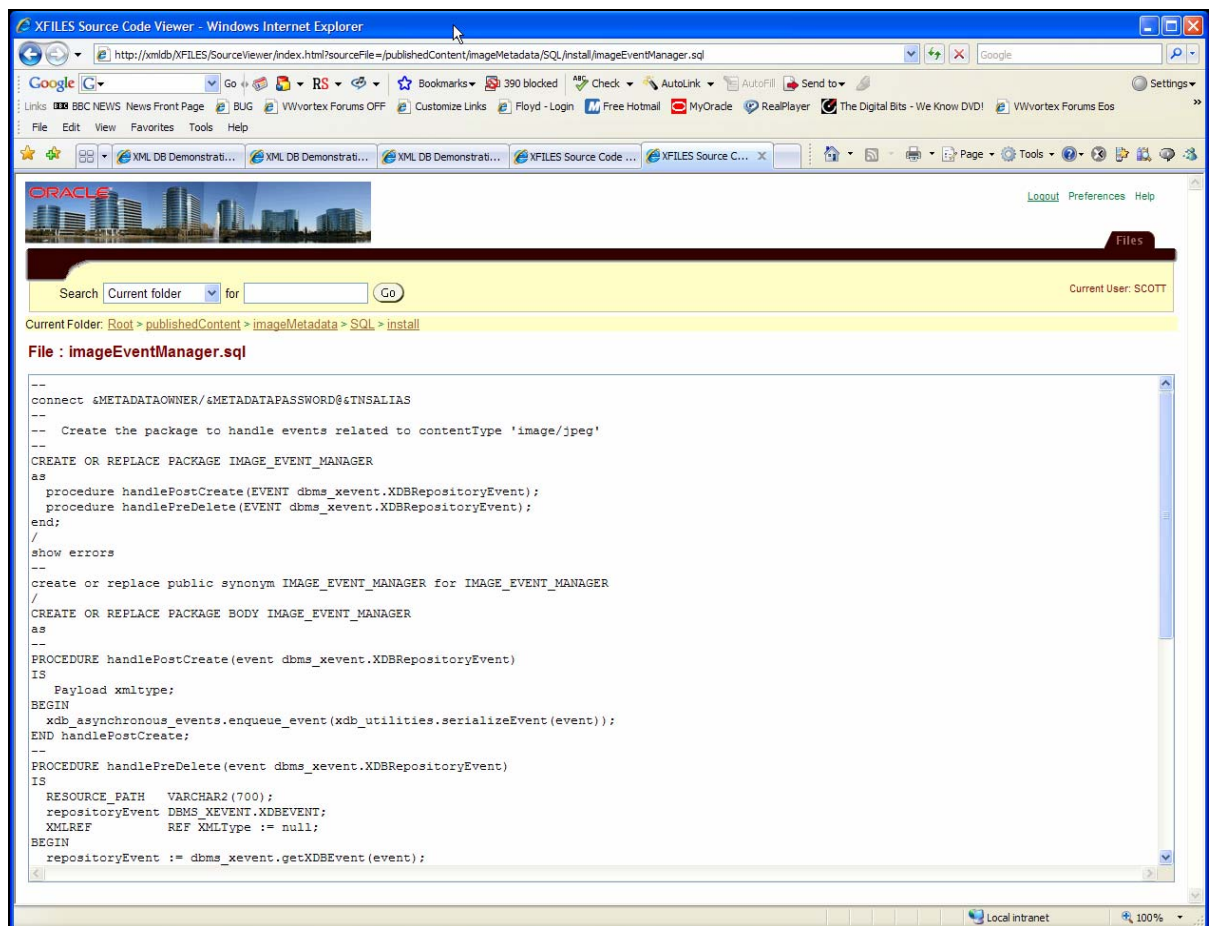


The Repository Configuration file configures the following repository event:

- Events will only be raised for resources where element ContentType in the XDB Resource namespace contains image/jpeg.
- Only Post-Create and Pre-Delete events will be raised.
- The events will be handled by PL/SQL package IMAGE_EVENT_MANAGER in database schema XDBMETADATA.

2.3 Repository Event Package

This step uses the source code viewer component of the X-Files application to show the PL/SQL package that provides the event implementation for JPEG files. This package is created as part of the demonstration installation. Click the icon to launch the XML DB Source Code Viewer and display the contents of the file.



The PL/SQL package implements two methods, `handlePostCreate` and `handlePreDelete`. The methods each take parameter `EVENT` of type `XDBRepositoryEvent`, defined by package

DBMS_XEVENT. The signature of these methods is predefined by the Oracle XML DB repository events framework.

Parameter Event provides access to the full details of the event, including information about the path to the files and folders that that event applies to. The event can also be used to access the contents of the document.

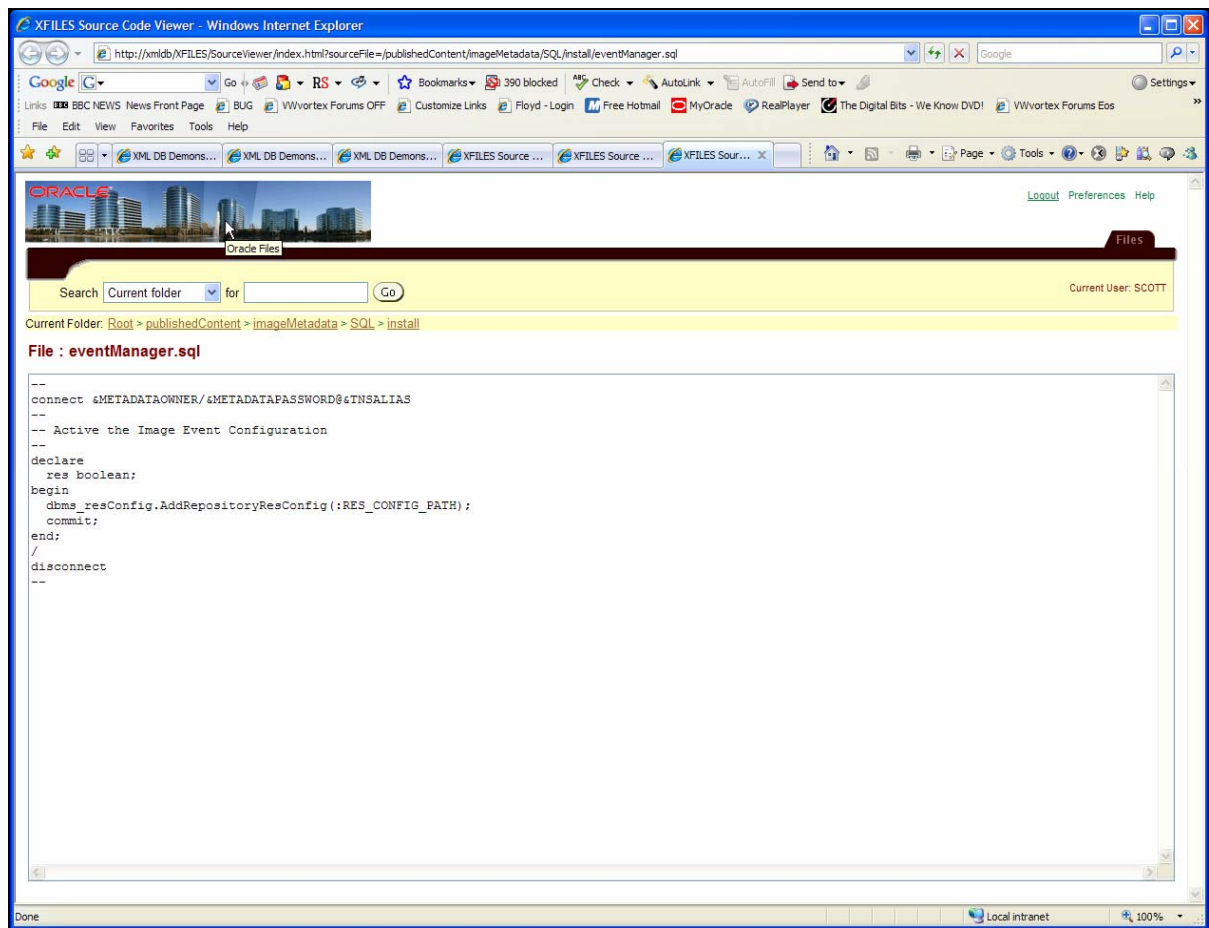
Procedure handlePostCreate converts the XDBRepositoryEvent event object into and invokes procedure enqueue_event in package xdb_asynchronous_events. This procedure creates a message and adds it to a message queue managed by Oracle Advanced Queuing. The payload of the message is the XML representation of the XDBRepositoryEvent. The use of Oracle Advanced Queuing makes it possible for the content of the new resource to be processed in an asynchronous manner. Using Oracle Advanced Queuing in this manner is recommended when performing processing as part of the event may cause protocol operations to timeout.

Procedure handlePreDelete ensures that any metadata associated with the resource is deleted when the resource is deleted.

2.4 Register Event Manager

The Resource Configuration document is installed as a repository wide configuration. This means that regardless of where in the Oracle XML DB repository a JPEG file is created or deleted the corresponding methods in the PL/SQL package will be invoked. The procedures will be invoked regardless of how the operation that raised the event is performed. The event will be raised by operations initiated using the Oracle XML DB protocol and by operations initiated using SQL and PL/SQL or other programmatic APIs.

This step uses the source code viewer component of the X-Files application to show the PL/SQL code that registers the repository configuration document. The script is run as part of the demonstration installation. Click the icon to launch the XML DB Source Code Viewer and display the contents of the file.



- The installation process registers the repository configuration document with Oracle XML DB.
- The repository configuration document is registered using procedure addRepositoryResConfig in package DBMS_RES_CONFIG. This procedure is used to

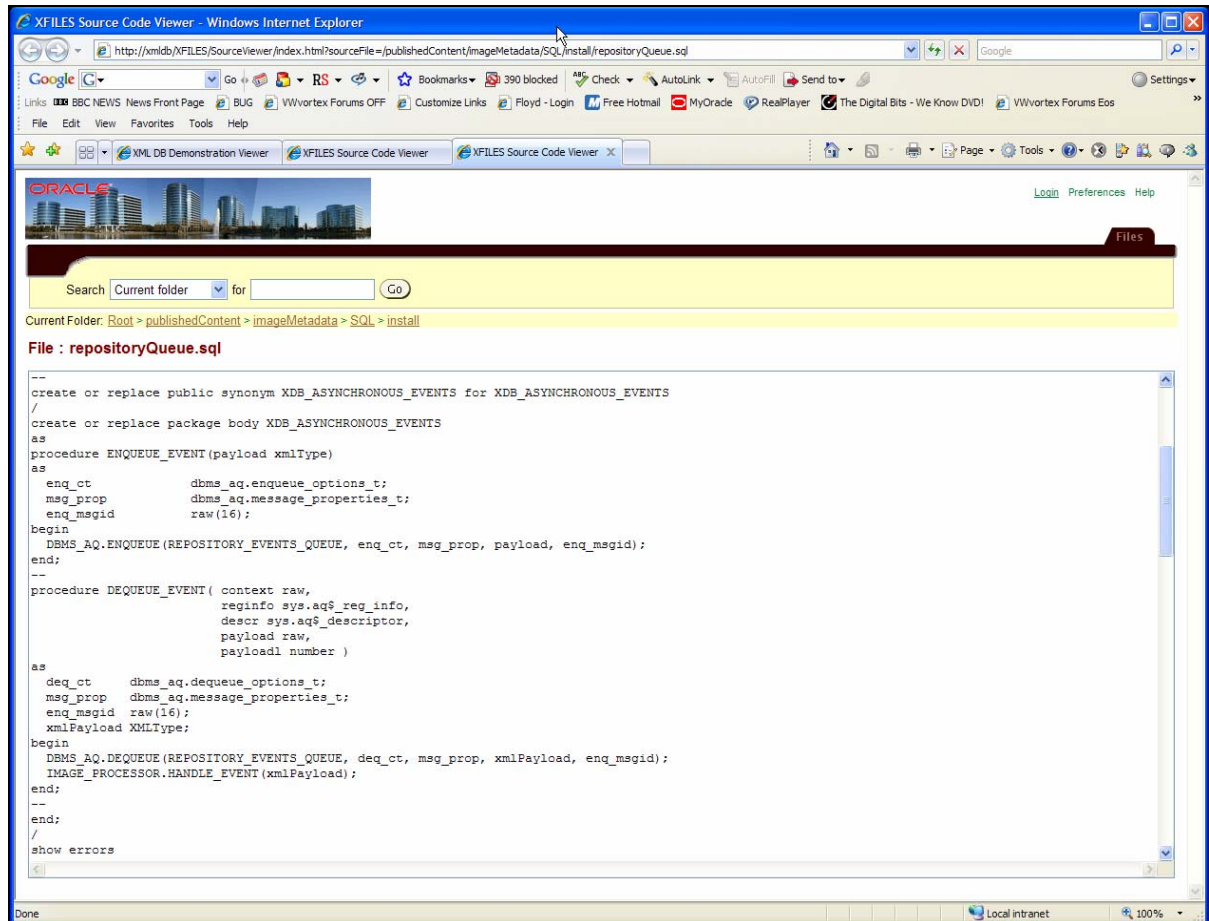
register repository -wide resource configuration documents. XDBADMIN is required to register a repository-wide resource configuration document.

- Other procedures in this package allow repository configuration documents to be associated with a particular resource, rather than the repository as a whole.

3.1 Repository Event Queue

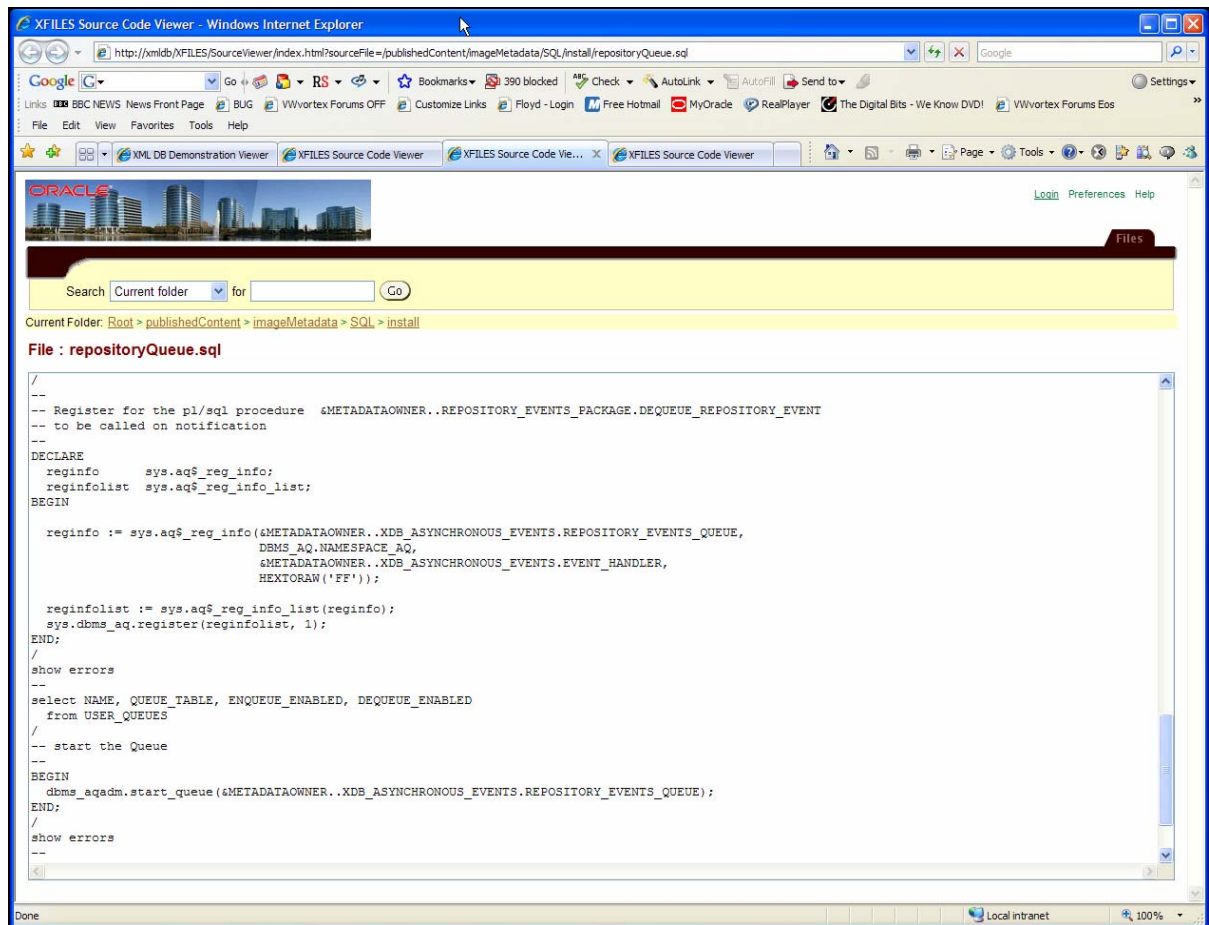
Oracle's Advanced Queuing infrastructure manages the creation and processing of the message generated by the event associated with creating a resource from a digital photograph. Oracle Advanced Queuing makes it possible to perform the metadata extraction in an asynchronous manner.

This step uses the source code viewer component of the X-Files application to show the script that creates the package and creates and starts the queue. The script is run as part of the demonstration installation. Click the icon to launch the XML DB Source Code Viewer and display the contents of the file.



- Package XDB_ASYNCHRONOUS_EVENTS provides the methods used to queue and dequeue the messages used to manage the process.
- Procedure ENQUEUE_EVENT creates a message and places it into the queue using procedure ENQUEUE in package DBMS_AQ. The payload of the message contains the type of resource, the operation that took place (creation) and the path to the newly created resource.

- Procedure `DEQUEUE_EVENT` processes a message. It uses procedure `DEQUEUE` in package `DBMS_AQ` to read the message from the queue and then passes the payload to procedure `HANDLE_EVENT` in package `IMAGE_PROCESSOR`. This procedure performs the metadata extraction and processing.



```

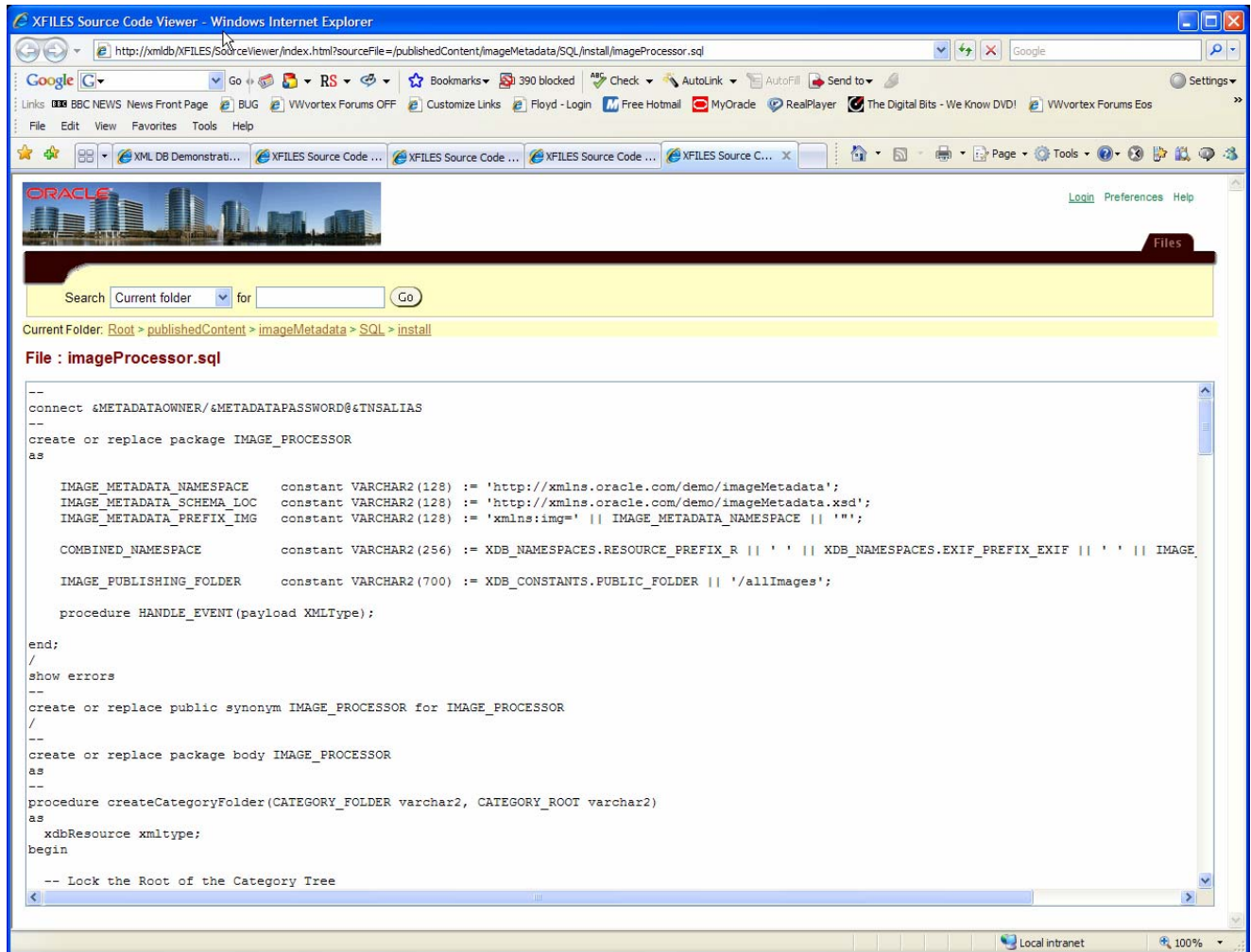
/
-- Register for the pl/sql procedure &METADATAOWNER..REPOSITORY_EVENTS_PACKAGE.DEQUEUE_REPOSITORY_EVENT
-- to be called on notification
--
DECLARE
  reginfo      sys.aq$reg_info;
  reginfo_list sys.aq$reg_info_list;
BEGIN
  reginfo := sys.aq$reg_info(&METADATAOWNER..XDB_ASYNCHRONOUS_EVENTS.REPOSITORY_EVENTS_QUEUE,
                           DBMS_AQ.NAMESPACE_AQ,
                           &METADATAOWNER..XDB_ASYNCHRONOUS_EVENTS.EVENT_HANDLER,
                           HEXTORAW('FF'));
  reginfo_list := sys.aq$reg_info_list(reginfo);
  sys.dbms_aq.register(reginfo_list, 1);
END;
/
show errors
--
select NAME, QUEUE_TABLE, ENQUEUE_ENABLED, DEQUEUE_ENABLED
from USER_QUEUES
/
-- start the Queue
--
BEGIN
  dbms_aqadm.start_queue(&METADATAOWNER..XDB_ASYNCHRONOUS_EVENTS.REPOSITORY_EVENTS_QUEUE);
END;
/
show errors
--
/

```

- Queue `REPOSITORY_EVENTS_QUEUE` is queue used to manage the process. The queue, and the associated queue table are created using methods provided by package `DBMS_AQ`.
- Procedure `DEQUEUE_EVENT` is defined as the notification procedure for the queue. This means that the procedure will be invoked by Oracle Advanced Queuing whenever a new message becomes available for processing.
- The queue is started by calling method `START_QUEUE` in package `DBMS_AQ`.

3.2 Image Processor

This step uses the source code viewer component of the X-Files application look at the PL/SQL package that performs the metadata extraction and subsequent categorization of the image. The package is created as part of the demonstration installation. Click the icon to launch the XML DB Source Code Viewer and display the contents of the file



- The package defines a single public procedure, HANDLE_EVENT.

The source code for the package is show below.

```
procedure HANDLE_EVENT(payload XMLType)
as
  RESID raw(16);
  TARGET_RESOURCE varchar2(700);
  EVENT_TYPE pls_integer;

  STACK_TRACE xmltype;
  res boolean;
begin
  EVENT_TYPE := payload.extract
    (
      '/re:ResourceEvent/re:EventType/text()',
      XDB_NAMESPACES.RESOURCE_EVENT_PREFIX_RE
    ).getNumberVal();

  TARGET_RESOURCE := payload.extract
    (
      '/re:ResourceEvent/re:resourcePath/text()',
      XDB_NAMESPACES.RESOURCE_EVENT_PREFIX_RE
    ).getStringVal();

  if (EVENT_TYPE = DBMS_XEVENT.POST_CREATE_EVENT) then
    insertImageMetdata(TARGET_RESOURCE);
  end if;

  commit;

exception

  when others then
    select xmlElement
      (
        "ImageProcessingError",
        xmlAttributes(USER as "User",SYSTIMESTAMP as "Timestamp"),
        xmlElement
          (
            "ErrorStack",
            xmlCDATA(DBMS_UTILITY.FORMAT_ERROR_STACK())
          ),
        xmlElement
          (
            "BackTrace",
            xmlCDATA(DBMS_UTILITY.FORMAT_ERROR_BACKTRACE())
          ),
        payload
      )
    into STACK_TRACE
    from DUAL;

  rollback;

  insert into IMAGE_METADATA_ERROR_TABLE values (STACK_TRACE);
  commit;

end;
```

Procedure `HANDLE_EVENT` takes the payload and extracts the type of event and the path to the resource that the event occurred on. If the event type is a `CREATE` it calls the private procedure `INSERTIMAGEMETADATA` to process the metadata contained in the image.

```
procedure insertImageMetdata(TARGET_RESOURCE VARCHAR2)
as
  XMLREF ref XMLType;
  SQL_ERROR_CODE NUMBER;
  SQL_ERROR_MSG VARCHAR2(4000);
  imageMetadata XMLType;
begin
  --
  -- Insert ORDIMAGE metadata into IMAGE_METADATA_TABLE and return REF.
  --
  imageMetadata := extractImageMetadata(TARGET_RESOURCE);
  if (imageMetadata is not null) then
    XDB.XDBPM_HELPER.APPENDRESOURCEMETADATA(TARGET_RESOURCE,imageMetadata);
    createWeakLinks(TARGET_RESOURCE,imageMetadata);
  end if;
end;
```

Procedure `INSERTIMAGEMETADATA` takes the path to the target resource and calls the private function `EXTRACTIMAGEMETADATA` to get the metadata for this resource. The metadata is attached to the resource using procedure `APPENDRESOURCEMETADATA`. Once the metadata has been attached to the resource the private procedure `CREATEWEAKLINKS` is used to create the categorization entries for this document.

```

function extractImageMetadata(TARGET_RESOURCE VARCHAR2)
return xmltype
as
    EXIF_METADATA          xmlSequenceType;
    EXIF_METADATA_COUNT    number;
    IMAGE_METADATA          xmlType;
    CONTENT                 BLOB;
    CSID                    BINARY_INTEGER;
begin
    --
    -- Use ORDIMAGE to extract Meta data from XMLLOB
    --

    SAVEPOINT READ_CONTENT;

    LOCK TABLE IMAGE_METADATA_TABLE IN EXCLUSIVE MODE;

    CONTENT := XDB.XDBPM_HELPER.GETCONTENTBLOB(TARGET_RESOURCE);

    if DBMS_LOB.GETLENGTH(CONTENT) > 0 then
        EXIF_METADATA := ordsys.ordimage.getMetadata(CONTENT, 'EXIF');

        if EXIF_METADATA.COUNT() > 0 then
            select xmlElement
            (
                "img:imageMetadata",
                xmlAttributes
                (
                    IMAGE_METADATA_NAMESPACE as "xmlns:img",
                    XDB_NAMESPACES.EXIF_NAMESPACE as "xmlns",
                    XDB_NAMESPACES.XMLINSTANCE_NAMESPACE as "xmlns:xsi",
                    IMAGE_METADATA_NAMESPACE || ' ' || IMAGE_METADATA_SCHEMA_LOC
                    as "xsi:schemaLocation"
                ),
                EXIF_METADATA(1)
            )
            into IMAGE_METADATA
            from dual;
        end if;
    end if;
    ROLLBACK TO SAVEPOINT READ_CONTENT;
    return IMAGE_METADATA;
end;

```

Procedure `EXTRACTIMAGEMETADATA` calls procedure `GETCONTENTBLOB` to get the content of the resource. It passes the content to function `GETMETADATA` in package `ORDIMAGE`. This function returns the metadata in the image as an XML document that complies with element `exifMetadata` defined by the EXIF XML Schema.

The metadata extracted from the image is then wrapped up inside an `imageMetadata` element, making it compliant with the Image Metadata XML Schema used by the demonstration.

```

procedure createWeakLinks(TARGET_RESOURCE VARCHAR2, IMAGE_METADATA XMLTYPE)
as
  FILENAME          VARCHAR2(700);
  TARGET_FOLDER     VARCHAR2(700);

  CAMERA_MAKE       VARCHAR2(2000);
  CAMERA_MODEL       VARCHAR2(2000);
  DATE_TAKEN        VARCHAR2(64);

  NODECHECK NUMBER;
begin
  select CAMERA_MAKE, CAMERA_MODEL, DATE_TAKEN
    into CAMERA_MAKE, CAMERA_MODEL, DATE_TAKEN
    from xmltable
      (
        xmlnamespaces
        (
          'http://xmlns.oracle.com/demo/imageMetadata' as "img",
          'http://xmlns.oracle.com/ord/meta/exif' as "exif"
        ),
        '/img:imageMetadata/exif:exifMetadata'
      )
    passing IMAGE_METADATA
    columns
      CAMERA_MAKE   varchar2(128)
                    path 'exif:TiffIfd/exif:Make/text()',
      CAMERA_MODEL  varchar2(128)
                    path 'exif:TiffIfd/exif:Model/text()',
      DATE_TAKEN    varchar2(128)
                    path 'exif:ExifIfd/exif:DateTimeOriginal/text()'
      );

  FILENAME := SUBSTR(TARGET_RESOURCE, INSTR(TARGET_RESOURCE, '/', -1) + 1);

  if (CAMERA_MAKE is not NULL and CAMERA_MODEL is not NULL) then
    TARGET_FOLDER := IMAGE_PUBLISHING_FOLDER || '/camera/'
                  || CAMERA_MAKE || '/'
                  || CAMERA_MODEL;
    createCategoryFolder(TARGET_FOLDER, IMAGE_PUBLISHING_FOLDER);
    createWeakLink(TARGET_RESOURCE, TARGET_FOLDER, FILENAME);
  end if;

  if (DATE_TAKEN is not NULL) then
    TARGET_FOLDER := IMAGE_PUBLISHING_FOLDER || '/dateTaken/'
                  || SUBSTR(DATE_TAKEN, 1, 4) || '/'
                  || SUBSTR(DATE_TAKEN, 6, 2);
    createCategoryFolder(TARGET_FOLDER, IMAGE_PUBLISHING_FOLDER);
    createWeakLink(TARGET_RESOURCE, TARGET_FOLDER, FILENAME);
  end if;

  XDB.XDBPM_HELPER.SETACL(TARGET_RESOURCE, '/sys/acls/published_image_acl.xml');

exception
  when no_data_found then
    null;
end;

```

Procedure CREATEWEAKLINKS extracts the camera manufacturer and model information and the date the picture was taken from the XML document containing the metadata. It ensures that a

categorization folder exists for the manufacturer / model combination and then creates a weak link between the folder and the resource being processed.

The categorization process uses weak links to ensure that if the owner of the resource deletes the resource the all references to the resource are removed from the categorization directory trees.

4.1 Create Folder

This step uses SQL to create a folder in the Oracle XML DB repository that will be used to store images. Click the icon to launch the XML DB demonstration framework and run the SQL script.

```
Command Output
set echo on
declare
  result boolean;
begin
  if (not dbms_xdb.existsResource('/home/SCOTT/ImageLibrary')) then
    result := dbms_xdb.createFolder('/home/SCOTT/ImageLibrary');
  end if;
end;
/
PL/SQL procedure successfully completed.
```

The script uses procedure createFolder, defined by package DBMS_XDB, to create a folders in the demonstration user's home folder called ImageLibrary.

4.2 Count Metadata

This step uses PL/SQL to count the number of rows in IMAGE_METADATA_TABLE. Click the icon to launch the XML DB demonstration framework and run the SQL script.

```
Command Output
set echo on
--
select count(*) from IMAGE_METADATA_TABLE
/

```

COUNT(*)
0

Result : 1 Rows Selected.

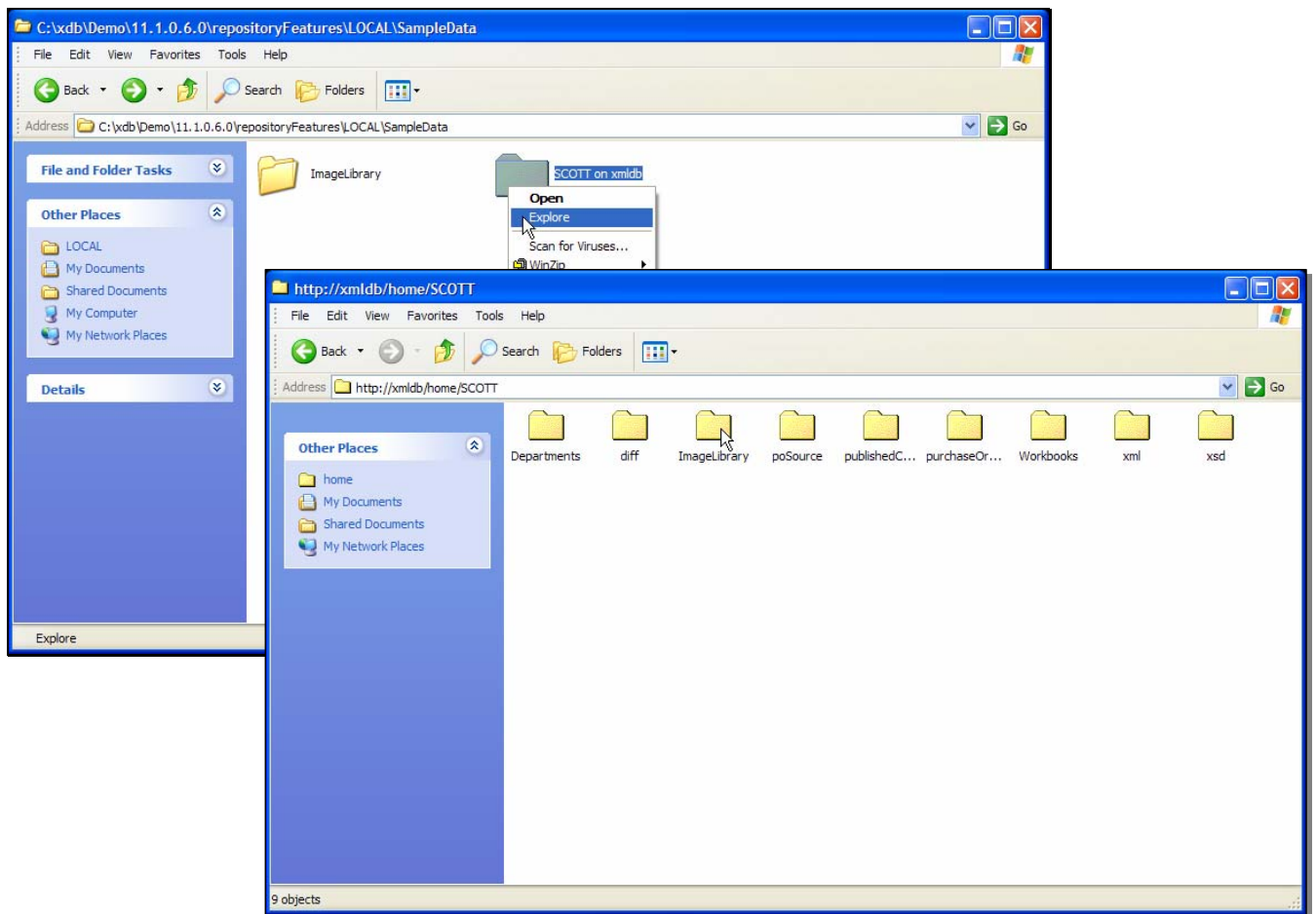
As long as the demonstration user is not the owner of any digital photographs stored in the Oracle XML DB repository the result of the query should be zero.

4.3 Copy Images (WebDAV)

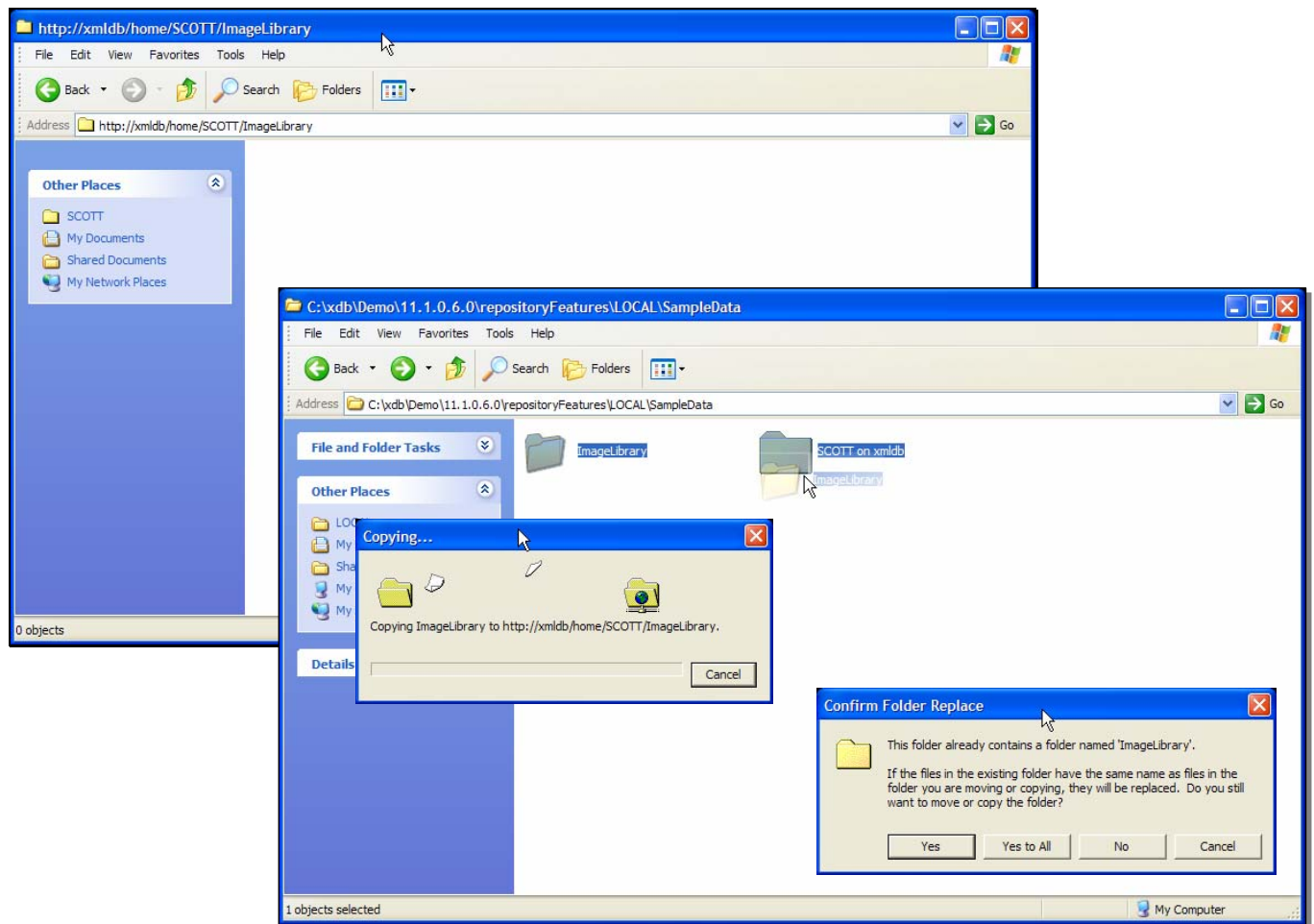
This step uses Microsoft Windows Explorer to copy digital photographs into the XML DB repository. Most of the photographs contain EXIF encoded metadata. Right click the icon and select explore. This will open a new window containing the local folder SampleData.

The window contains two items, a folder called ImageLibrary and a shortcut called SCOTT on xmldb. The folder ImageLibrary contains a number of digital photographs taken with a variety of different digital camera. This shortcut is a link to the demonstration user's home folder in the Oracle XML DB repository. The demonstration should work with any digital photograph that contains EXIF metadata. However, some of the later steps assume that this set of images has been loaded.

Right click the SCOTT on xmldb shortcut and select explore. This will open a new window containing the Oracle XML DB folder /home/SCOTT. Since the repository requires an authenticated connection, Windows Explorer will prompt for a username and password. Enter the demonstration user's username and password and click OK.

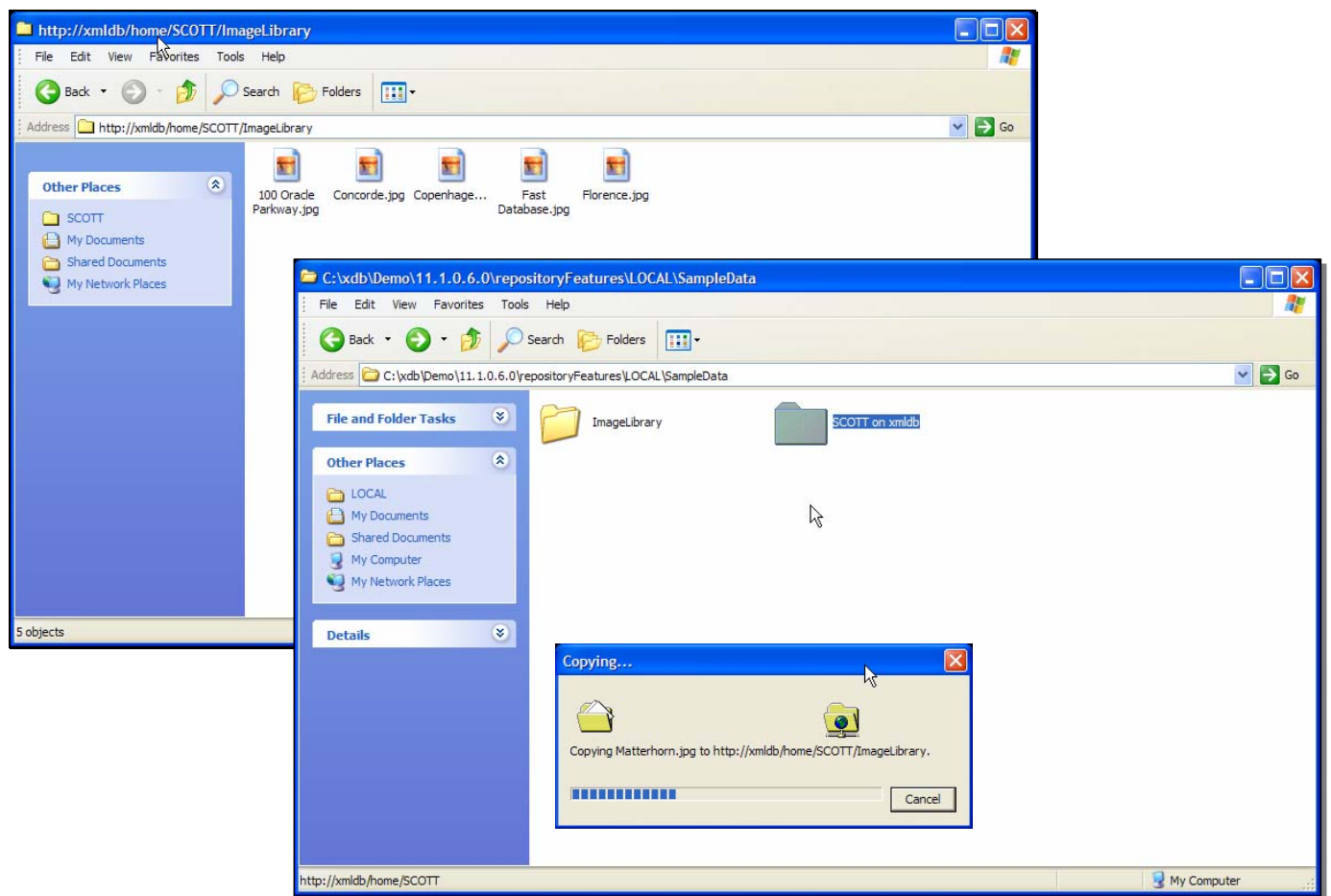


The window will contain a number of folders. Double click the ImageLibrary shortcut. This will open the ImageLibrary folder. This folder should be empty at this point. Reposition the windows so the window containing the remote folder ImageLibrary is clearly visible behind the window containing the local folder SampleData. Drag the local folder ImageLibrary and drop it on the link to the remote folder.



Since folder ImageLibrary already exists in the Oracle XML DB repository windows explorer will open a dialog box requesting confirmation that the content of the remote folder should be replaced with the content of the local folder before starting the copy. Click Yes.

The contents of the local folder will be copied into the Oracle XML DB repository. As each copy operation completes, the file should be come visible as an icon in the window containing the remote folder.



4.4 Recount Metadata

This step re-executes the SQL that counts the number of rows in `IMAGE_METADATA_TABLE`. Click the icon to launch the XML DB demonstration framework and run the SQL script.

```

Command Output
set echo on
--
select count(*) from IMAGE_METADATA_TABLE
/

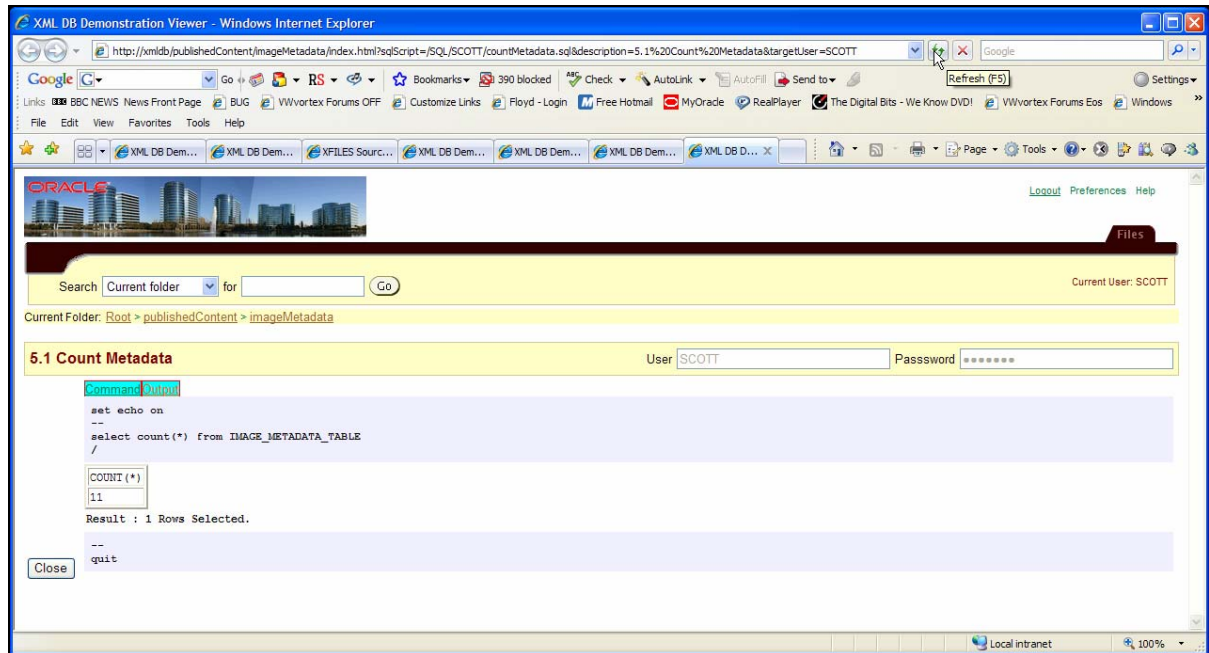
COUNT (*)
6

Result : 1 Rows Selected.

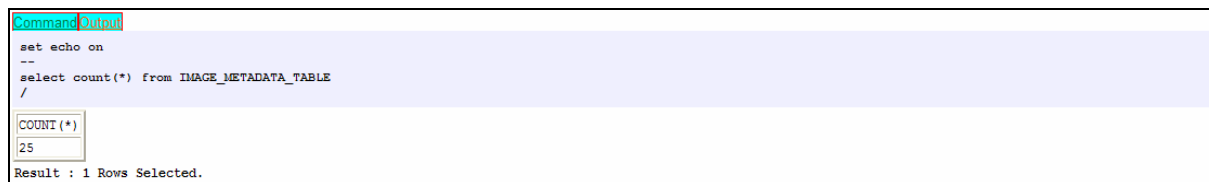
```

Since the metadata extraction is performed in an asynchronous manner the number of rows in `IMAGE_METADATA_TABLE` may initially be less than the number of images that have been copied into the Oracle XML DB repository.

Click the refresh button or press the F5 key to reload the page.



Repeat the process until all the images have been processed. With the supplied set of photographs, 25 of the 26 images contain EXIF formatted metadata. The metadata extraction process will result in 25 rows being created in IMAGE_METADATA_TABLE. One of the images, Florence.jpg, was taken with an Olympus D600 camera which recorded metadata using a proprietary Olympus format that pre-dated the EXIF standard.



Once IMAGE_METADATA_TABLE contains 25 rows it is safe to proceed with the rest of the demonstration.

5.1 Query Metadata

This step shows some simple queries on IMAGE_METADATA_TABLE. Click the icon to launch the XML DB demonstration framework and run the SQL script.

```
Command Output
set linesize 132
set long 10240
--
column "Filename" format A32
column "Camera" format A20
column "Height" format 99999
column "Width" format 99999
column "Exposure" format A20
--
set echo on
--
select object_value
from IMAGE_METADATA_TABLE m, RESOURCE_VIEW r
where m.RESID = r.RESID
and equals_path(res, '/home/SCOTT/ImageLibrary/Concorde.jpg') = 1
/

OBJECT_VALUE
<img:imageMetadata xsi:schemaLocation="http://xmlns.oracle.com/demo/imageMetadata http://xmlns.oracle.com/demo/imageMetadata.xsd">
  <exifMetadata xsi:schemaLocation="http://xmlns.oracle.com/ord/meta/exif http://xmlns.oracle.com/ord/meta/exif">
    <TiffId>
      <Make tag="271">Canon</Make>
      <Model tag="272">Canon EOS D30</Model>
      <Orientation tag="274">top left</Orientation>
      <XResolution tag="282">180</XResolution>
      <YResolution tag="283">180</YResolution>
      <ResolutionUnit tag="296">inches</ResolutionUnit>
      <DateTime tag="306">2003-05-05T09:42:11.000000</DateTime>
      <YCbCrPositioning tag="531">centered</YCbCrPositioning>
    </TiffId>
    <ExifId tag="34665">
      <ExposureTime tag="33434">.0028571428</ExposureTime>
      <FNumber tag="33437">6.7</FNumber>
      <ExifVersion tag="36864">0210</ExifVersion>
      <DateTimeOriginal tag="36867">2003-05-05T09:42:11.000000</DateTimeOriginal>
      <DateTimeDigitized tag="36868">2003-05-05T09:42:11.000000</DateTimeDigitized>
      <ComponentsConfiguration tag="37121">YCbCr</ComponentsConfiguration>
      <CompressedBitsPerPixel tag="37122">3</CompressedBitsPerPixel>
      <ShutterSpeedValue tag="37377">8.451218</ShutterSpeedValue>
      <ApertureValue tag="37378">5.488327</ApertureValue>
      <ExposureBiasValue tag="37380">0</ExposureBiasValue>
      <MaxApertureValue tag="37381">2.9708557</MaxApertureValue>
      <SubjectDistance tag="37382">unknown</SubjectDistance>
      <MeteringMode tag="37383">Partial</MeteringMode>
    </ExifId>
    <Flash tag="37385">
      <Fired>No</Fired>
      <Return>No strobe return function</Return>
    </Flash>
    <FocalLength tag="37386">45</FocalLength>
    <FlashpixVersion tag="40960">0100</FlashpixVersion>
    <ColorSpace tag="40961">sRGB</ColorSpace>
    <PixelXDimension tag="40962">2160</PixelXDimension>
    <PixelYDimension tag="40963">1440</PixelYDimension>
    <FocalPlaneXResolution tag="41486">2421.5247</FocalPlaneXResolution>
    <FocalPlaneYResolution tag="41487">2420.168</FocalPlaneYResolution>
    <FocalPlaneResolutionUnit tag="41488">inches</FocalPlaneResolutionUnit>
    <SensingMethod tag="41495">One-chip color area</SensingMethod>
    <FileSource tag="41728">DSC</FileSource>
  </exifMetadata>
</img:imageMetadata>

Result : 1 Rows Selected.
```

The first query performs a join between RESOURCE_VIEW and IMAGE_METADATA_TABLE to get the metadata for the image /home/SCOTT/ImageLibrary/Concorde.jpg. The join is performed by comparing with the value of column RESID in RESOURCE_VIEW with column RESID in IMAGE_METADATA_TABLE.

When procedure APPENDRESOURCEMETA is used to attach metadata to a resource, the metadata is stored in the default table defined by the metadata schema, and the hidden column RESID is set to the value of the RESID column associated with the resource. This allows XML DB to access schema

The second query shows a summary of the schema Based Metadata for each resource.

The query shows the filename, manufacturer and model of camera used, dimensions of the image in pixels, and exposure information for each picture in the folder /home/SCOTT/ImageLibrary. The results are order by the size of the image.

5.2 Expand Metadata

This step uses the Oracle XML DB DML operators to add additional information to the metadata extracted from photographs.. Click the icon to launch the XML DB demonstration framework and run the SQL script.

The first operation shows the metadata for resource `/home/SCOTT/ImageLibrary/Norwich (3).jpg`.

Command	Output
<pre>set long 100000 set echo on -- select object_value from IMAGE_METADATA_TABLE m, RESOURCE_VIEW r where m.RESID = r.RESID and equals_path(res, '/home/SCOTT/ImageLibrary/Norwich (3).jpg') = 1 /</pre>	<pre>OBJECT_VALUE <img:imageMetadata xsi:schemaLocation="http://xmlns.oracle.com/demo/imageMetadata http://xmlns.oracle.com/demo/imageMetadata.xsd"> <exifMetadata xsi:schemaLocation="http://xmlns.oracle.com/ord/meta/exif http://xmlns.oracle.com/ord/meta/exif"> <TiffIfd> <Make tag="271">Canon</Make> <Model tag="272">Canon EOS D30</Model> <Orientation tag="274">top left</Orientation> <XResolution tag="282">180</XResolution> <YResolution tag="283">180</YResolution> <ResolutionUnit tag="296">inches</ResolutionUnit> <DateTime tag="306">2002-03-16T06:19:53.000000</DateTime> <YCbCrPositioning tag="531">centered</YCbCrPositioning> </TiffIfd> <ExifIfd tag="34665"> <ExposureTime tag="33434">.0028571428</ExposureTime> <FNumber tag="33437">9.5</FNumber> <ExifVersion tag="36864">0210</ExifVersion> <DateTimeOriginal tag="36867">2002-03-16T06:19:53.000000</DateTimeOriginal> <DateTimeDigitized tag="36868">2002-03-16T06:19:53.000000</DateTimeDigitized> <ComponentsConfiguration tag="37121">XCbCr</ComponentsConfiguration> <CompressedBitsPerPixel tag="37122">3</CompressedBitsPerPixel> <ShutterSpeedValue tag="37377">8.451218</ShutterSpeedValue> <ApertureValue tag="37378">6.4958496</ApertureValue> <ExposureBiasValue tag="37380">0</ExposureBiasValue> <MaxApertureValue tag="37381">2.9708557</MaxApertureValue> <SubjectDistance tag="37382">unknown</SubjectDistance> <MeteringMode tag="37383">Pattern</MeteringMode> </ExifIfd> <Flash tag="37385"> <FiredNo</Fired> <Return>No strobe return function</Return> </Flash> <FocalLength tag="37386">28</FocalLength> <FlashpixVersion tag="40960">0100</FlashpixVersion> <ColorSpace tag="40961">sRGB</ColorSpace> <PixelYDimension tag="40963">2160</PixelYDimension> <PixelXDimension tag="40962">1440</PixelXDimension> <FocalPlaneXResolution tag="41486">2421.5247</FocalPlaneXResolution> <FocalPlaneYResolution tag="41487">2420.168</FocalPlaneYResolution> <FocalPlaneResolutionUnit tag="41488">inches</FocalPlaneResolutionUnit> <SensingMethod tag="41495">One-chip color area</SensingMethod> <FileSource tag="41728">DSC</FileSource> </ExifIfd> <InteroperabilityIfd tag="40965"> <InteroperabilityIndex tag="1">R98</InteroperabilityIndex> </InteroperabilityIfd> </exifMetadata> </img:imageMetadata></pre>
Result : 1 Rows Selected.	

The second operation is an update that uses operator insertChildXML to add elements Title and Description to the row in IMAGE_METADATA_TABLE that contains the metadata for /home/SCOTT/ImageLibrary/Norwich (3).jpg .

```

Command Output
update IMAGE_METADATA_TABLE m
set object_value = insertChildXML
(
    object_value,
    '/imageMetadata',
    'Title',
    xmltype('<Title xmlns="http://xmlns.oracle.com/demo/imageMetadata">Norwich Cathedral from Ethelbert Gate</Title>'),
    'xmlns="http://xmlns.oracle.com/demo/imageMetadata"'
)
where m.RESID =
(
    select RESID
    from RESOURCE_VIEW
    where equals_path(res, '/home/SCOTT/ImageLibrary/Norwich (3).jpg') = 1
)
/
1 row updated.
Command Output
declare
narrative XMLTYPE := xmltype(
'<img:Description xmlns:img="http://xmlns.oracle.com/demo/imageMetadata" xmlns="http://www.w3.org/1999/xhtml">
<p>In the heart of Norwich stands the Cathedral, separated from the busy streets
by flint walls and entrance gates, but still a living part of the city. At least
three services are held in the Cathedral every day, often sung by the choir. The
choristers are pupils of King Edward VI School, which has its daily assembly in
the Cathedral. Concerts, lectures and exhibitions also frequently take place
here.<br/>
<br/>
The Cathedral was begun in 1096, the vision of Herbert de Losinga, first bishop
of Norwich. Building work on the Cathedral, a bishop's palace and the
associated Benedictine monastery continued throughout his life, but the
Cathedral was not finally consecrated until 1278. The building is mainly of Caen
stone, a pale, honey-coloured limestone brought over from Normandy, but Norfolk
flints form the core of the Cathedral, and stone from Northamptonshire was used
for medieval additions.<br/>
<br/>
This great church has a Norman ground plan and walls, and a Perpendicular roof
and spire, added after a fire caused by lightning destroyed the wooden roof and
spire in 1463. The Cathedral spire is 315 ft (96m) high - second only in height
to that of Salisbury.</p>
</img:Description>');
begin
update IMAGE_METADATA_TABLE m
set object_value = insertChildXML
(
    object_value,
    '/img:imageMetadata',
    'img:Description',
    narrative,
    'xmlns:img="http://xmlns.oracle.com/demo/imageMetadata"'
)
where m.RESID =
(
    select RESID
    from RESOURCE_VIEW
    where equals_path(res, '/home/SCOTT/ImageLibrary/Norwich (3).jpg') = 1
);
end;
/
PL/SQL procedure successfully completed.

```

- The first update statement inserts element Title. The content of element Title is simple string containing the title of the image.
- The second update statements adds element Description. The content of element Description is a fragment containing a description of the image. The fragment consists of complex, XHTML formatted markup.
- Operator insertChildXML can only add one element at a time, so a separate statement is required for each element. It would be possible to stack the insertChildXML operations into a single update statement.

The third operation shows the updated metadata for /home/SCOTT/ImageLibrary/Norwich (3).jpg.

```
Command Output
select object_value
from IMAGE_METADATA_TABLE m, RESOURCE_VIEW r
where m.RESID = r.RESID
and equals_path(res,'/home/SCOTT/ImageLibrary/Norwich (3).jpg') = 1
/

OBJECT_VALUE
<img:imageMetadata xsi:schemaLocation="http://xmlns.oracle.com/demo/imageMetadata http://xmlns.oracle.com/demo/imageMetadata.xsd">
  <img:title>Norwich Cathedral from Ethelbert Gate</img:title>
  <img:description>
    <p>
      In the heart of Norwich stands the Cathedral, separated from the busy streets by flint walls and entrance gates, but still a living part of the city. At least thr
    <br/>
    <br/>
      The Cathedral was begun in 1096, the vision of Herbert de Losinga, first bishop of Norwich. Building work on the Cathedral, a bishop's palace and the associated F
    <br/>
    <br/>
      This great church has a Norman ground plan and walls, and a Perpendicular roof and spire, added after a fire caused by lightning destroyed the wooden roof and spi
    <p>
  </img:description>
  <exif:Metadata xsi:schemaLocation="http://xmlns.oracle.com/ord/meta/exif http://xmlns.oracle.com/ord/meta/exif">
    <TiffIfd>
      <Make tag="271">Canon</Make>
      <Model tag="272">Canon EOS D30</Model>
      <Orientation tag="274">top left</Orientation>
      <XResolution tag="282">180</XResolution>
      <YResolution tag="283">180</YResolution>
      <ResolutionUnit tag="296">inches</ResolutionUnit>
      <DateTime tag="306">2002-03-16T06:19:53.000000</DateTime>
      <YCbCrPositioning tag="531">centered</YCbCrPositioning>
    </TiffIfd>
    <ExifIfd tag="34665">
      <ExposureTime tag="33434">.0028571428</ExposureTime>
      <FNumber tag="33437">9.5</FNumber>
      <ExifVersion tag="36864">0210</ExifVersion>
      <DateTimeOriginal tag="36867">2002-03-16T06:19:53.000000</DateTimeOriginal>
      <DateTimeDigitized tag="36868">2002-03-16T06:19:53.000000</DateTimeDigitized>
      <ComponentsConfiguration tag="37121">YCbCr</ComponentsConfiguration>
      <CompressedBitsPerPixel tag="37122">3</CompressedBitsPerPixel>
      <ShutterSpeedValue tag="37377">8.451218</ShutterSpeedValue>
      <ApertureValue tag="37378">6.4958496</ApertureValue>
      <ExposureBiasValue tag="37380">0</ExposureBiasValue>
      <MaxApertureValue tag="37381">2.9708557</MaxApertureValue>
      <SubjectDistance tag="37382">unknown</SubjectDistance>
      <MeteringMode tag="37383">Pattern</MeteringMode>
    </ExifIfd>
    <Flash tag="37385">
      <FiredNo</Fired>
      <Return>No strobe return function</Return>
    </Flash>
    <FocalLength tag="37386">28</FocalLength>
    <FlashpixVersion tag="40960">0100</FlashpixVersion>
    <ColorSpace tag="40961">sRGB</ColorSpace>
    <PixelYDimension tag="40963">2160</PixelYDimension>
    <PixelXDimension tag="40962">1440</PixelXDimension>
    <FocalPlaneXResolution tag="41486">2421.5247</FocalPlaneXResolution>
    <FocalPlaneYResolution tag="41487">2420.168</FocalPlaneYResolution>
    <FocalPlaneResolutionUnit tag="41488">inches</FocalPlaneResolutionUnit>
    <SensingMethod tag="41495">One-chip color area</SensingMethod>
    <FileSource tag="41728">DSC</FileSource>
  </ExifIfd>
  <InteroperabilityIfd tag="40965">
    <InteroperabilityIndex tag="1">R98</InteroperabilityIndex>
  </InteroperabilityIfd>
</exif:Metadata>
</img:imageMetadata>

Result : 1 Rows Selected.
```

- The document now contains elements Title and Description. Description contains elements from the XHTML specification. Since the definition of description allows an unbounded number of child elements in the XHTML namespace this perfectly valid

The fourth operation updates metadata for /home/SCOTT/ImageLibrary/Concorde.jpg.

```
Command Output
update IMAGE_METADATA_TABLE m
set object_value = insertChildXML
(
    object_value,
    '/imageMetadata',
    'Title',
    xmltype('<Title xmlns="http://xmlns.oracle.com/demo/imageMetadata">Concorde From the Concorde Lounge at British Airways Terminal 1</Title
    xmlns="http://xmlns.oracle.com/demo/imageMetadata"'
)
where m.RESID =
(
    select RESID
    from RESOURCE_VIEW
    where equals_path(res,'/home/SCOTT/ImageLibrary/Concorde.jpg') = 1
)
/
1 row updated.

Command Output
--
declare
narrative XMLTYPE := xmlType(
'<img:Description xmlns:img="http://xmlns.oracle.com/demo/imageMetadata" xmlns="http://www.w3.org/1999/xhtml">
<p>Concorde measures 204ft in length - stretching between six and ten inches
in-flight due to heating of the airframe. She is painted in a specially
developed white paint to accommodate these changes and to dissipate the heat
generated by supersonic flight. The wingspan is 83ft 8ins - much less than
conventional subsonic aircraft as Concorde flies in totally a different way
using &apos;Vortex Lift&apos; to achieve her exceptional performance. The height is 37ft
1ins. The characteristic droop nose is lowered to improve pilots&apos; visibility
for take-off and landing.<br/>
Power: Concorde&apos;s four engines - specially designed Rolls-Royce/ Snecma Olympus
593s - give more than 38,000lbs of thrust each, with &apos;reheat&apos;. This adds fuel
to the final stage of the engine to produce the extra power required for
take-off and the transition to supersonic flight. They are the most powerful
pure jet engines flying commercially.<br/>
Speed: Concorde takes off at 220 knots (250mph) (compared with 165 knots for
most subsonic aircraft). She cruises at around 1350mph - more than twice the
speed of sound - and at an altitude of up to 60,000 ft (over 11 miles high). A
typical London to New York crossing takes a little less than three and a half
hours as opposed to about eight hours for a subsonic flight. Travelling
Westwards, the five-hour time difference means Concorde effectively arrives
before she has taken off. She travels &apos;faster than the sun&apos;.<br />
Service: More than 2.5 million passengers have flown supersonically on British
Airways&apos; Concorde since she entered commercial service in 1976. The most
frequent passenger, an oil company executive, has clocked up almost 70 round
trip transatlantic crossings a year.<br/>
Safety Enhancements: A team of about 250 British Airways&apos; engineers worked
tirelessly, together with the relevant authorities, to further improve safety on
board. The engineers and manufacturers have worked closely together to create
and install a new Kevlar-rubber lining for Concorde&apos;s wing-based fuel tanks and
the addition of specially developed Michelin tires, as well as strengthened
winging in the undercarriage.<br/>
History: Supersonic airline research in Europe began in 1956 and resulted in the
British and French Governments signing an international treaty for the joint
design, development and manufacture of a supersonic airliner six years later.
The first prototype was rolled out at Toulouse in 1967 and since then, there
have been a number of notable dates in the history of Concorde.<br/>
</p>
<ul>
<li>2 March 1969: First flight of Concorde 001 from Toulouse France.</li>
<li>9 April 1969: First flight of Concorde 002 from Filton, Bristol, UK to its test centre at Fairford.</li>
<li>1 October 1969: Concorde&apos;s first supersonic flight.</li>
<li>28 June 1972: British Airways (BOAC) orders five Concorde.</li>
<li>20 September 1973: Concorde 002 lands at Dallas/ Fort Worth on first visit to the USA.</li>
<li>7 June 1974: Concorde makes its first double Atlantic crossing in one day.</li>
<li>5 December 1975: UK Civil Aviation Authority awards Concorde its Certificate of Airworthiness.</li>
<li>21 January 1976: British Airways commences commercial supersonic travel from London to Bahrain.</li>
<li>22 November 1977: British Airways Concorde&apos;s first London - New York commercial flight.</li>
<li>8 November 1986: First round the world flight by a British Airways Concorde - covering 28,238 miles in 29 hours 59 minutes.</li>
<li>7 February 1996: Concorde G-BOAD crosses the Atlantic between New York and London in a new record flight time of 2 hours, 52 minutes and 59 seconds.</li>
<li>11 August 1999: Two British Airways Concorde fly in supersonic formation to chase the total eclipse of the sun.</li>
</ul>
</img:Description>');
begin
update IMAGE_METADATA_TABLE m
set object_value = insertChildXML
(
    object_value,
    '/img:imageMetadata',
    'img:Description',
    narrative,
    'xmlns:img="http://xmlns.oracle.com/demo/imageMetadata"'
)
where m.RESID =
(
    select RESID
    from RESOURCE_VIEW
    where equals_path(res,'/home/SCOTT/ImageLibrary/Concorde.jpg') = 1
);
end;
/
PL/SQL procedure successfully completed.
```

The fifth operation shows the updated metadata for /home/SCOTT/ImageLibrary/Concorde.jpg.

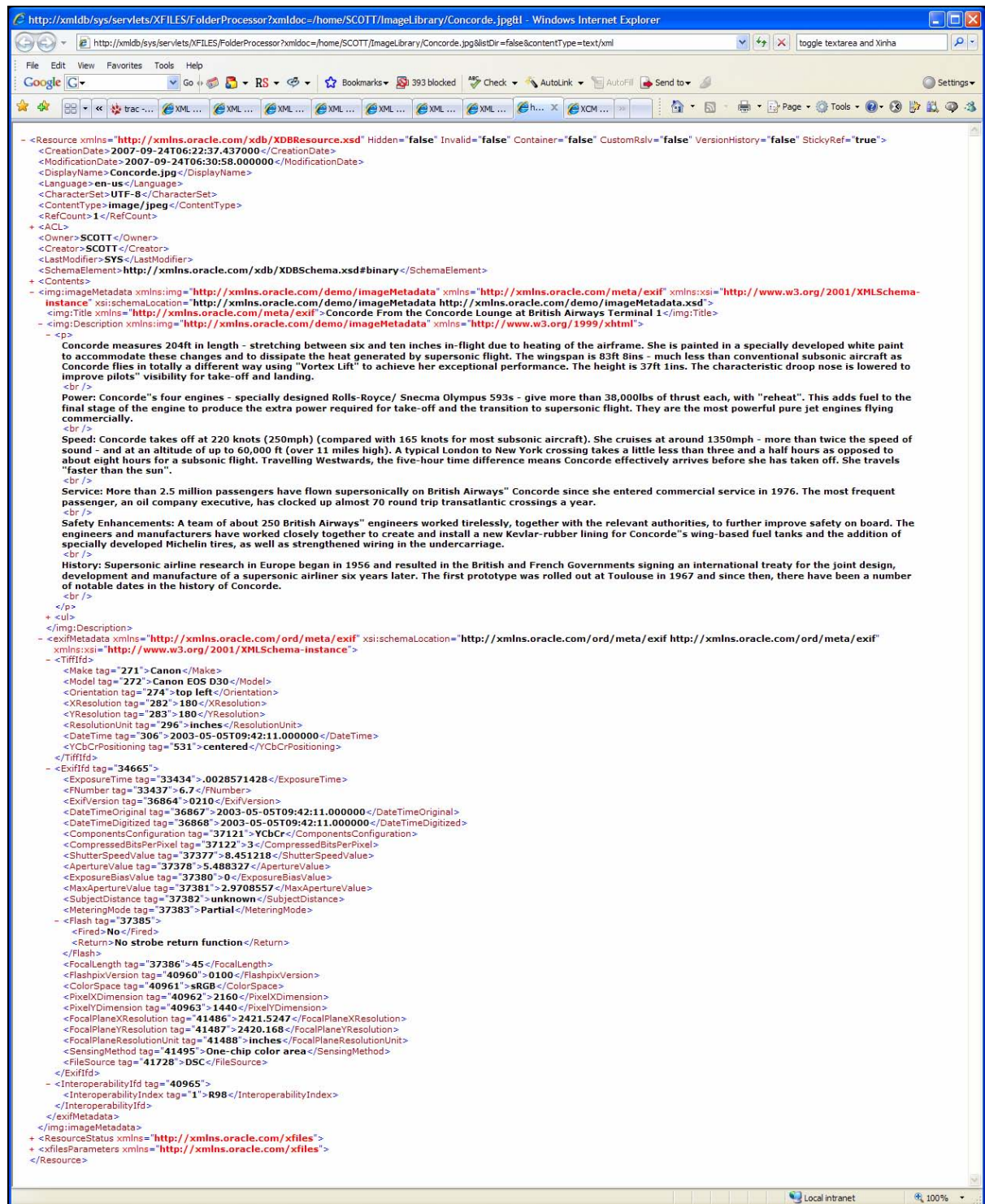
```
Command Output
select object_value
  from IMAGE_METADATA_TABLE i, RESOURCE_VIEW r
 where i.RESID = r.RESID
    and equals_path(res,'/home/SCOTT/ImageLibrary/Concorde.jpg') = 1
/

OBJECT_VALUE
<img:imageMetadata xsi:schemaLocation="http://xmlns.oracle.com/demo/imageMetadata http://xmlns.oracle.com/demo/imageMetadata.xsd">
  <img:Title>Concorde From the Concorde Lounge at British Airways Terminal 1</img:Title>
  <img:Description>
    <p>
      Concorde measures 204ft in length - stretching between six and ten inches in-flight due to heating of the airframe. She is painted in a specially developed white
    <br/>
      Power: Concorde's four engines - specially designed Rolls-Royce/ Snecma Olympus 593s - give more than 38,000lbs of thrust each, with 'reheat'. This adds fuel to t
    <br/>
      Speed: Concorde takes off at 220 knots (250mph) (compared with 165 knots for most subsonic aircraft). She cruises at around 1350mph - more than twice the speed of
    <br/>
      Service: More than 2.5 million passengers have flown supersonically on British Airways' Concorde since she entered commercial service in 1976. The most frequent p
    <br/>
      Safety Enhancements: A team of about 250 British Airways' engineers worked tirelessly, together with the relevant authorities, to further improve safety on board.
    <br/>
      History: Supersonic airline research in Europe began in 1956 and resulted in the British and French Governments signing an international treaty for the joint desig
    <br/>
    </p>
  </img:Description>
  <ul>
    <li>2 March 1969: First flight of Concorde 001 from Toulouse France.</li>
    <li>9 April 1969: First flight of Concorde 002 from Filton, Bristol, UK to its test centre at Fairford.</li>
    <li>1 October 1969: Concorde's first supersonic flight.</li>
    <li>28 June 1972: British Airways (BOAC) orders five Concorde.</li>
    <li>20 September 1973: Concorde 002 lands at Dallas/ Fort Worth on first visit to the USA.</li>
    <li>7 June 1974: Concorde makes its first double Atlantic crossing in one day.</li>
    <li>5 December 1975: UK Civil Aviation Authority awards Concorde its Certificate of Airworthiness.</li>
    <li>21 January 1976: British Airways commences commercial supersonic travel from London to Bahrain.</li>
    <li>22 November 1977: British Airways Concorde's first London - New York commercial flight.</li>
    <li>8 November 1986: First round the world flight by a British Airways Concorde - covering 28,238 miles in 29 hours 59 minutes.</li>
    <li>7 February 1996: Concorde G-BOAD crosses the Atlantic between New York and London in a new record flight time of 2 hours, 52 minutes and 59 seconds.</li>
    <li>11 August 1999: Two British Airways Concorde fly in supersonic formation to chase the total eclipse of the sun.</li>
  </ul>
  </img:Description>
  <exif:Metadata xsi:schemaLocation="http://xmlns.oracle.com/ord/meta/exif http://xmlns.oracle.com/ord/meta/exif">
    <TiffIfd>
      <Make tag="271">Canon</Make>
      <Model tag="272">Canon EOS D30</Model>
      <Orientation tag="274">top left</Orientation>
      <XResolution tag="282">180</XResolution>
      <YResolution tag="283">180</YResolution>
      <ResolutionUnit tag="296">inches</ResolutionUnit>
      <DateTime tag="306">2003-05-05T09:42:11.000000</DateTime>
      <YCbCrPositioning tag="531">centered</YCbCrPositioning>
    </TiffIfd>
    <ExifIfd tag="34665">
      <ExposureTime tag="33434">.0028571428</ExposureTime>
      <FNumber tag="33437">6.7</FNumber>
      <ExifVersion tag="36864">0210</ExifVersion>
      <DateTimeOriginal tag="36867">2003-05-05T09:42:11.000000</DateTimeOriginal>
      <DateTimeDigitized tag="36868">2003-05-05T09:42:11.000000</DateTimeDigitized>
      <ComponentsConfiguration tag="37121">YCbCr</ComponentsConfiguration>
      <CompressedBitsPerPixel tag="37122">3</CompressedBitsPerPixel>
      <ShutterSpeedValue tag="37377">8.451218</ShutterSpeedValue>
      <ApertureValue tag="37378">5.488327</ApertureValue>
      <ExposureBiasValue tag="37380">0</ExposureBiasValue>
      <MaxApertureValue tag="37381">2.9708557</MaxApertureValue>
      <SubjectDistance tag="37382">unknown</SubjectDistance>
      <MeteringMode tag="37383">Partial</MeteringMode>
      <Flash tag="37385">
        <Fired>No</Fired>
        <Return>No strobe return function</Return>
      </Flash>
      <FocalLength tag="37386">45</FocalLength>
      <FlashpixVersion tag="40960">0100</FlashpixVersion>
      <ColorSpace tag="40961">sRGB</ColorSpace>
      <PixelXDimension tag="40962">2160</PixelXDimension>
      <PixelYDimension tag="40963">1440</PixelYDimension>
      <FocalPlaneXResolution tag="41486">2421.5247</FocalPlaneXResolution>
      <FocalPlaneYResolution tag="41487">2420.168</FocalPlaneYResolution>
      <FocalPlaneResolutionUnit tag="41488">inches</FocalPlaneResolutionUnit>
      <SensingMethod tag="41495">One-chip color area</SensingMethod>
      <FileSource tag="41728">DSC</FileSource>
    </ExifIfd>
    <InteroperabilityIfd tag="40965">
      <InteroperabilityIndex tag="1">R98</InteroperabilityIndex>
    </InteroperabilityIfd>
  </exif:Metadata>
</img:imageMetadata>

Result : 1 Rows Selected.
```

6.1 View Resource XML

This step shows how the both the basic metadata data and the extended user-defined metadata can be accessed as part of the resource. Click the icon to view the resource document in the browser.



6.2 View Images (XFILES)

This step uses the X-Files application to view the folder containing the images. Click the icon to launch the X-Files application and open the folder /home/SCOTT/ImageLibrary.

The X-Files application will be launched showing the FolderBrowser view of the target folder.

XCM - Browse Files: ImageLibrary - Windows Internet Explorer

http://xmldb/sys/servlets/XFILES/FolderProcessor?xmlDoc=/home/SCOTT/ImageLibrary&xslDoc=/cabo/XSL/FolderBrowser.xsl&listDir=true&contentType=text/html&doAuth...

File Google Go RS Bookmarks 393 blocked Check AutoLink AutoFill Send to Settings

ORACLE Logout Preferences Help

Search Current folder for Go Current User: SCOTT

Current Folder: Root > home > SCOTT > ImageLibrary

Browse Files : ImageLibrary

Select and ... New Folder Upload Go

Select All | Select None

Select	Name	Actions	Status	Size	Owner	Last Modified	Description
<input type="checkbox"/>	100 Oracle Parkway.jpg			1703931	SCOTT	23 Sep 2007 17:09	
<input type="checkbox"/>	Concorde.jpg			955499	SCOTT	23 Sep 2007 17:18	
<input type="checkbox"/>	Copenhagen.jpg			1010881	SCOTT	23 Sep 2007 17:09	
<input type="checkbox"/>	Fast Database.jpg			1692879	SCOTT	23 Sep 2007 17:09	
<input type="checkbox"/>	Florence.jpg			943898	SCOTT	23 Sep 2007 17:08	
<input type="checkbox"/>	Half Dome.jpg			1670528	SCOTT	23 Sep 2007 17:09	
<input type="checkbox"/>	Half Moon Bay (Bhushan).jpg			1889798	SCOTT	23 Sep 2007 17:09	
<input type="checkbox"/>	Matterhorn.jpg			977033	SCOTT	23 Sep 2007 17:09	
<input type="checkbox"/>	Milan.jpg			1386811	SCOTT	23 Sep 2007 17:09	
<input type="checkbox"/>	Moonrise over the Alps.jpg			633799	SCOTT	23 Sep 2007 17:09	
<input type="checkbox"/>	Norwich (1).jpg			1220667	SCOTT	23 Sep 2007 17:09	
<input type="checkbox"/>	Norwich (2).jpg			1624456	SCOTT	23 Sep 2007 17:09	
<input type="checkbox"/>	Norwich (3).jpg			1649698	SCOTT	23 Sep 2007 17:11	
<input type="checkbox"/>	Norwich (4).jpg			1280614	SCOTT	23 Sep 2007 17:10	
<input type="checkbox"/>	Nurnberg.jpg			1439867	SCOTT	23 Sep 2007 17:10	
<input type="checkbox"/>	Ravi_Japan.jpg			1106590	SCOTT	23 Sep 2007 17:10	
<input type="checkbox"/>	Redwood Shores.jpg			2870711	SCOTT	23 Sep 2007 17:10	
<input type="checkbox"/>	Shanghai.jpg			847036	SCOTT	23 Sep 2007 17:10	
<input type="checkbox"/>	Skiddaw.jpg			1331161	SCOTT	23 Sep 2007 17:10	
<input type="checkbox"/>	Solvang.jpg			808750	SCOTT	23 Sep 2007 17:10	
<input type="checkbox"/>	Sunrise over Twickenham.jpg			1095339	SCOTT	23 Sep 2007 17:10	
<input type="checkbox"/>	Sunset Over the Langdales.jpg			718969	SCOTT	23 Sep 2007 17:10	
<input type="checkbox"/>	Sunset on Buttermere.jpg			931457	SCOTT	23 Sep 2007 17:10	
<input type="checkbox"/>	Swarthmoor Hall.jpg			1406142	SCOTT	23 Sep 2007 17:10	
<input type="checkbox"/>	Tahoe Moonlight.JPG			826425	SCOTT	23 Sep 2007 17:10	
<input type="checkbox"/>	Thumbs.db			82944	SCOTT	23 Sep 2007 17:09	
<input type="checkbox"/>	Tokyo.jpg			1954002	SCOTT	23 Sep 2007 17:10	

Logout Preferences Help

Copyright © 2007 Oracle Corporation. All rights reserved.

View XML View XSL

Done Local intranet 100%

Click Concorde.jpg. The image will be displayed using a custom XSL stylesheet.


Oracle XML DB 11gR1 Repository Features

Click Concorde.jpg. The image will be displayed using a custom XSL stylesheet.

Current User: SCOTT

Current Folder: Root > home > SCOTT > ImageLibrary

Review and edit extended meta data for: Concorde From the Concorde Lounge at British Airways Terminal 1



Display Name: Concorde.jpg
Created By: SCOTT
Created: 01 Oct 2007 06:19
Last Modified By: SYS
Last Modified: 01 Oct 2007 14:26
Content Type: image/jpeg
Manufacturer: Canon
Model: Canon EOS D30
Resolution: 2160 x 1440
Exposure: .0028571428 at F6.7
Date Taken: 05 May 2003 09:42
Subject Distance: unknown
Metering Mode: Partial
Focal Length: 45

Description:

Concorde measures 204ft in length - stretching between six and ten inches in-flight due to heating of the airframe. She is painted in a specially developed white paint to accommodate these changes and to dissipate the heat generated by supersonic flight. The wingspan is 83ft 8ins - much less than conventional subsonic aircraft as Concorde flies in totally a different way using Vortex Lift to achieve her exceptional performance. The height is 37ft 1ins. The characteristic droop nose is lowered to improve pilots' visibility for take-off and landing.

Power: Concorde's four engines - specially designed Rolls-Royce/Snecma Olympus 593s - give more than 38,000lbs of thrust each, with 'reheat'. This adds fuel to the final stage of the engine to produce the extra power required for take-off and the transition to supersonic flight. They are the most powerful pure jet engines flying commercially.

Speed: Concorde takes off at 220 knots (250mph) (compared with 165 knots for most subsonic aircraft). She cruises at around 1350mph - more than twice the speed of sound - and at an altitude of up to 60,000 ft (over 11 miles high). A typical London to New York crossing takes a little less than three and a half hours as opposed to about eight hours for a subsonic flight. Travelling Westwards, the five-hour time difference means Concorde effectively arrives before she has taken off. She travels 'faster than the sun'.

Service: More than 2.5 million passengers have flown supersonically on British Airways' Concorde since she entered commercial service in 1976. The most frequent passenger, an oil company executive, has clocked up almost 70 round trip transatlantic crossings a year.

Safety Enhancements: A team of about 250 British Airways' engineers worked tirelessly, together with the relevant authorities, to further improve safety on board. The engineers and manufacturers have worked closely together to create and install a new Kevlar-rubber lining for Concorde's wing-based fuel tanks and the addition of specially developed Michelin tires, as well as strengthened wiring in the undercarriage.


History: Supersonic airline research in Europe began in 1956 and resulted in the British and French Governments signing an international treaty for the joint design, development and manufacture of a supersonic airliner six years later. The first prototype was rolled out at Toulouse in 1967 and since then, there have been a number of notable dates in the history of Concorde.

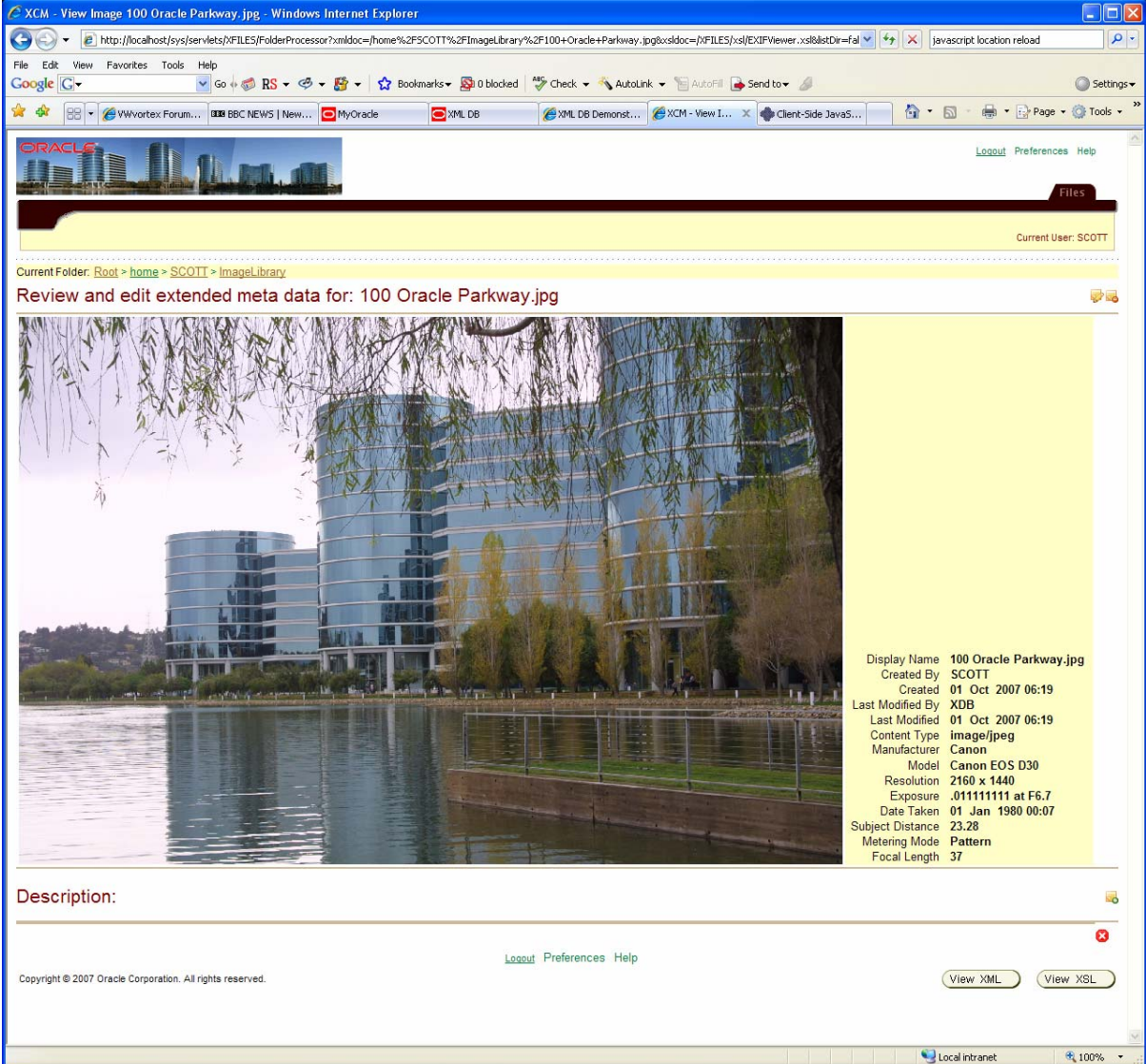
- 2 March 1969: First flight of Concorde 001 from Toulouse France.
- 9 April 1969: First flight of Concorde 002 from Filton, Bristol, UK to its test centre at Fairford.
- 1 October 1969: Concorde's first supersonic flight.
- 28 June 1972: British Airways (BOAC) orders five Concorde's.
- 20 September 1973: Concorde 002 lands at Dallas/ Fort Worth on first visit to the USA.
- 7 June 1974: Concorde makes its first double Atlantic crossing in one day.
- 5 December 1975: UK Civil Aviation Authority awards Concorde its Certificate of Airworthiness.
- 21 January 1976: British Airways commences commercial supersonic travel from London to Bahrain.
- 22 November 1977: British Airways Concorde's first London - New York commercial flight.
- 8 November 1986: First round the world flight by a British Airways Concorde - covering 28,238 miles in 29 hours 59 minutes.
- 7 February 1996: Concorde G-BOAD crosses the Atlantic between New York and London in a new record flight time of 2 hours, 52 minutes and 59 seconds.
- 11 August 1999: Two British Airways Concorde's fly in supersonic formation to chase the total eclipse of the sun.

Copyright © 2007 Oracle Corporation. All rights reserved.

View XML View XSL

- The HTML page contains content derived from the standard metadata maintained for each resource in the repository and the custom metadata maintained for digital photographs.
- Note that the title of the page is obtained from element Title and the contents of element Description provides the description of the photograph. Since element Description contains an XHTML fragment, the description automatically appears as properly formatted text when viewed in a browser.

Click the  (close) icon to return to the folder browser. Click 100 Oracle Parkway.jpg. The image will be displayed using the EXIF viewer. Since the metadata for this photograph does not contain elements Title and Description the name of the resource is used as the title and the description is empty.




The screenshot shows a web browser window titled "XCM - View Image 100 Oracle Parkway.jpg - Windows Internet Explorer". The address bar shows a local URL. The browser has several tabs open, including "VWvortex Forum...", "BBC NEWS | New...", "MyOracle", "XML DB", "XML DB Demon...", "XCM - View I...", and "Client-Side JavaS...".

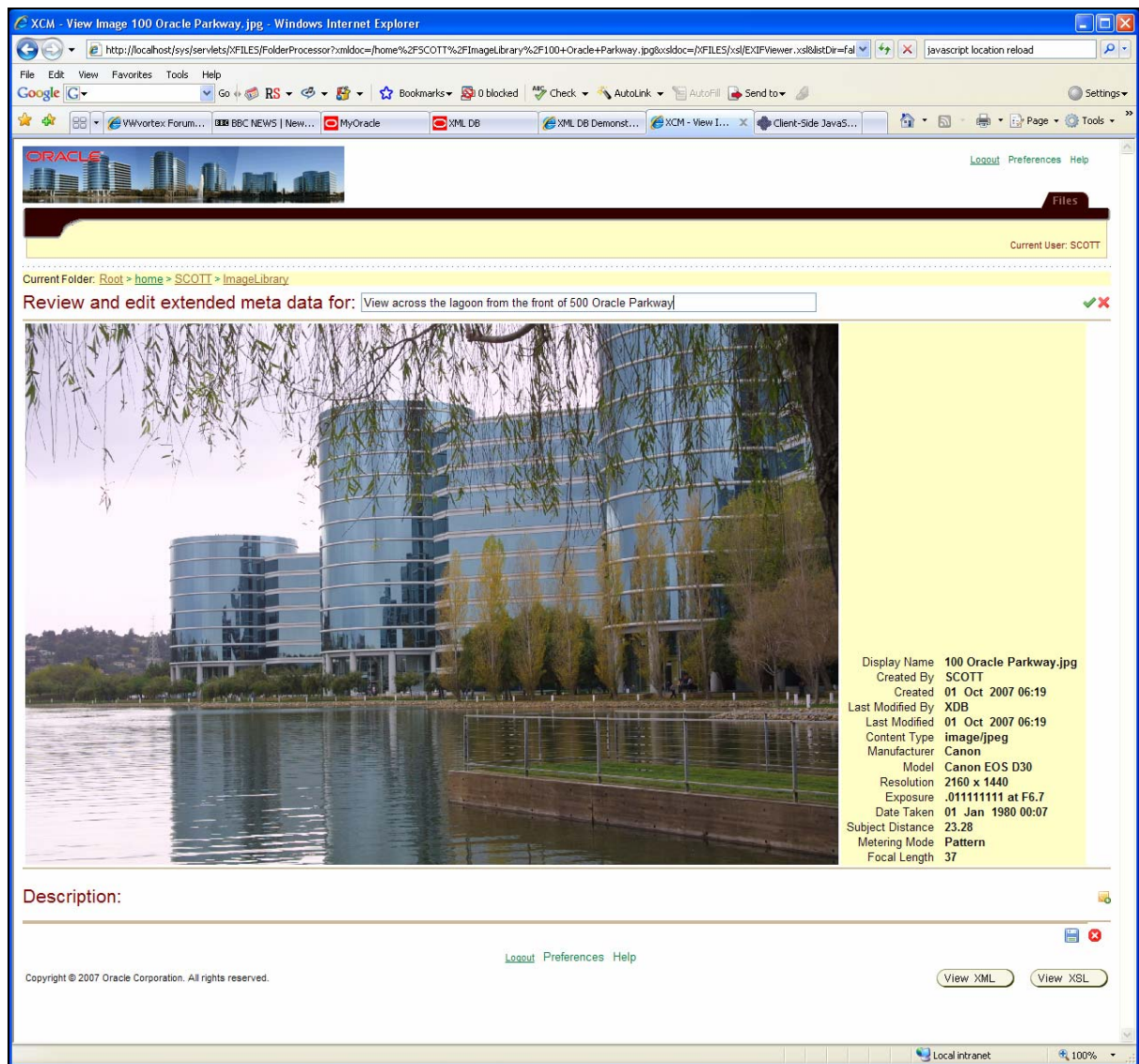
The web application interface includes a header with the Oracle logo and a navigation bar with "Logout", "Preferences", and "Help" links. Below the header, a yellow banner displays "Current User: SCOTT". The main content area shows the current folder path: "Root > home > SCOTT > ImageLibrary".



The title of the page is "Review and edit extended meta data for: 100 Oracle Parkway.jpg". The main image is a photograph of a modern glass building (100 Oracle Parkway) reflected in a body of water, with a willow tree in the foreground. To the right of the image, the EXIF metadata is displayed:



Display Name	100 Oracle Parkway.jpg
Created By	SCOTT
Created	01 Oct 2007 06:19
Last Modified By	XDB
Last Modified	01 Oct 2007 06:19
Content Type	image/jpeg
Manufacturer	Canon
Model	Canon EOS D30
Resolution	2160 x 1440
Exposure	.011111111 at F6.7
Date Taken	01 Jan 1980 00:07
Subject Distance	23.28
Metering Mode	Pattern
Focal Length	37

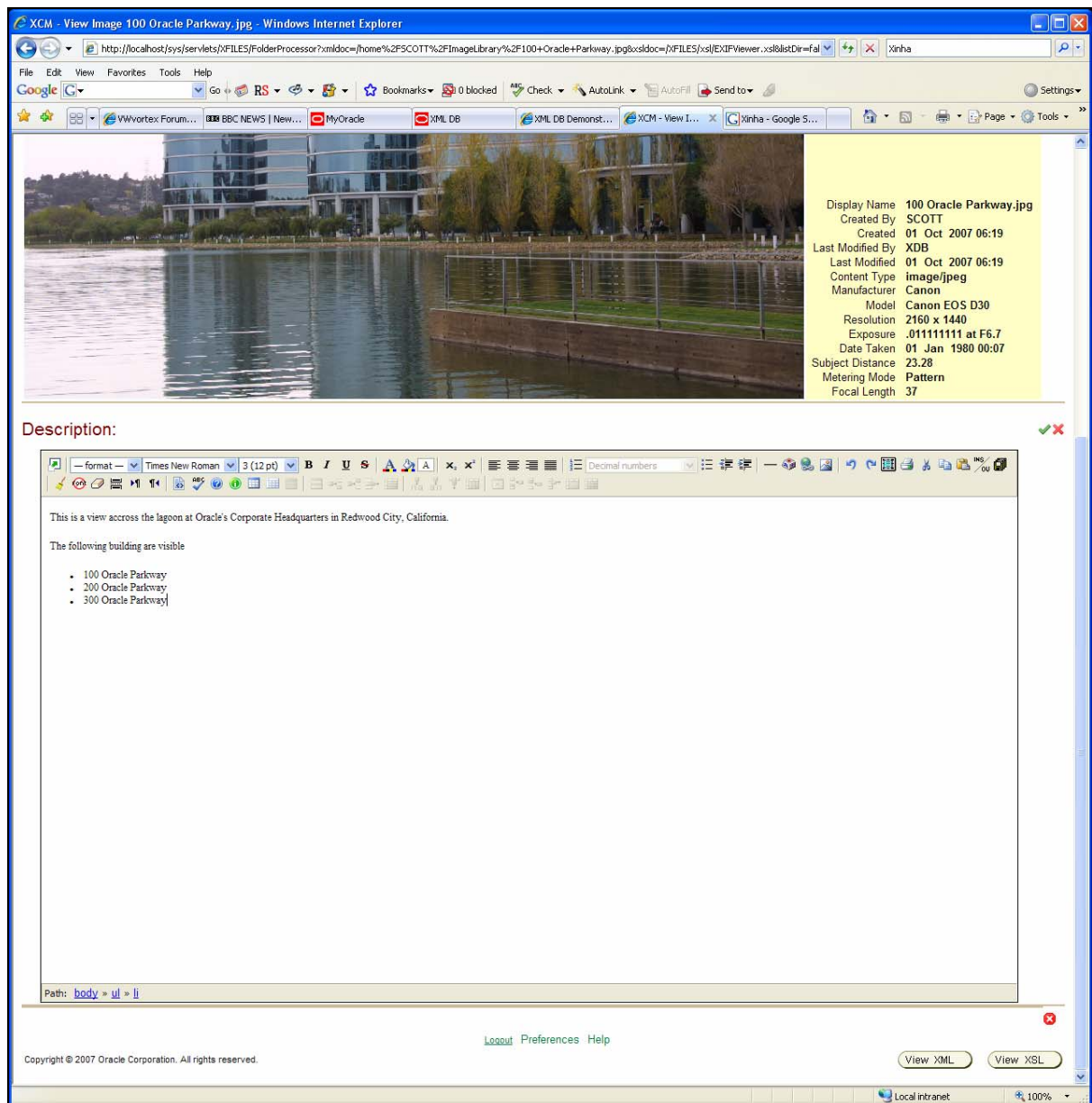
Below the image, the "Description:" field is empty. At the bottom of the page, there is a copyright notice: "Copyright © 2007 Oracle Corporation. All rights reserved." and two buttons: "View XML" and "View XSL".

Click the  (Edit) icon. The title is replaced with an Input box that allows the title to be edited.



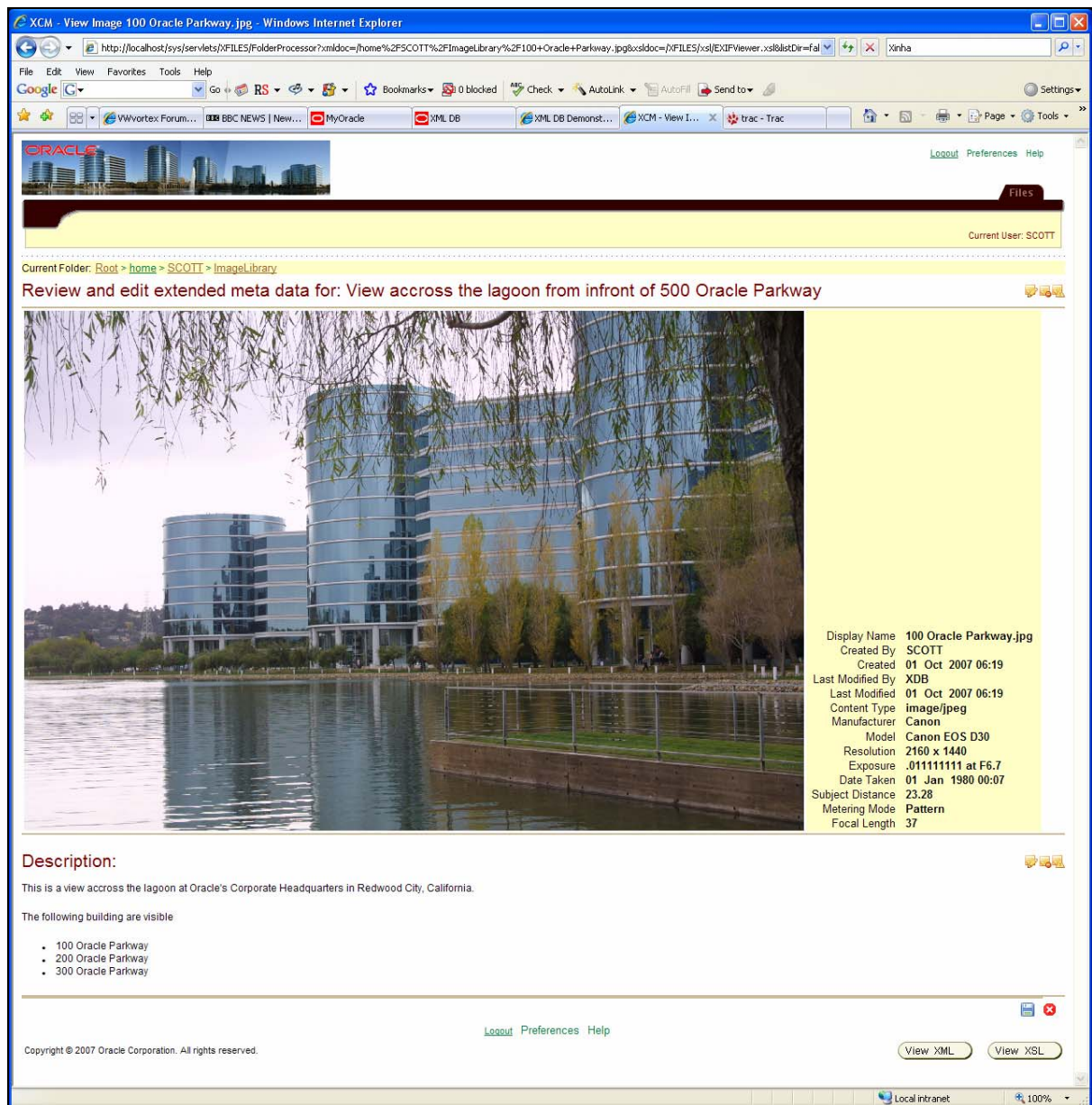
Enter new text for the photograph title and then click the  (Save) icon. The content of the Title area will be updated with the value entered and the  (Save Changes) will appear at the bottom of the page.


Now click the  (Add Description) icon. A WYSIWYG editor will appear that allows descriptive text to be entered. Enter some formatted text and click the  icon to save the changes.



- The editor used in this example is Xinha.
- Xinha is an open source Rich Text editing environment, implemented in JavaScript.
- Xinha provides a familiar WISIWIG editing environment for XHTML content.
- The Xinha project is maintained at the following URL <http://xinha.webfactional.com/>

The description will be updated with the content that was entered using Xinha.

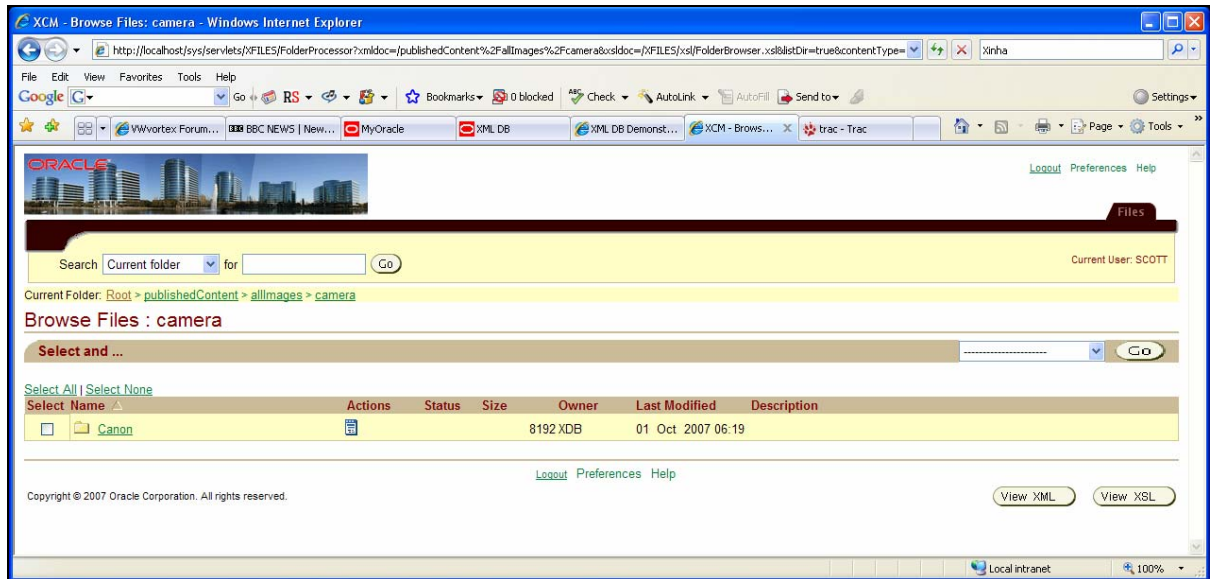


Click the save  icon to update the metadata for this image. The changes will be posted to the database by calling a simple PL/SQL procedure that updates elements Title and Description. The PL/SQL procedure is invoked using Database Native Web Services. The Update Complete dialog will indicate that the operation has completed.

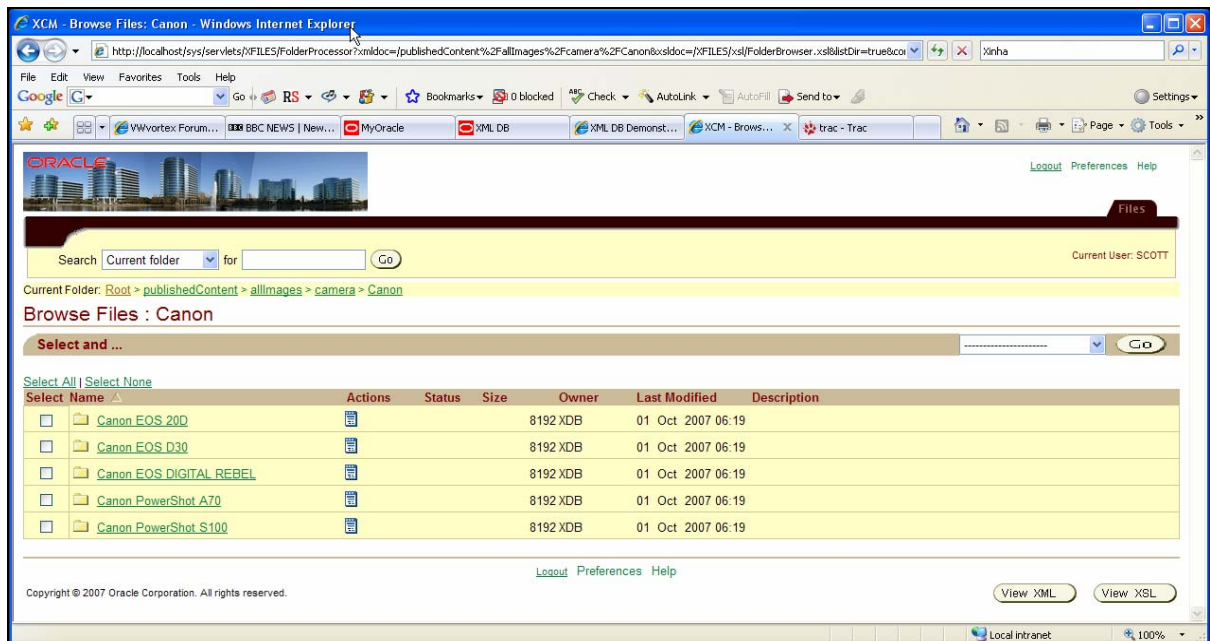


6.3 View Image Categories (XFILES)

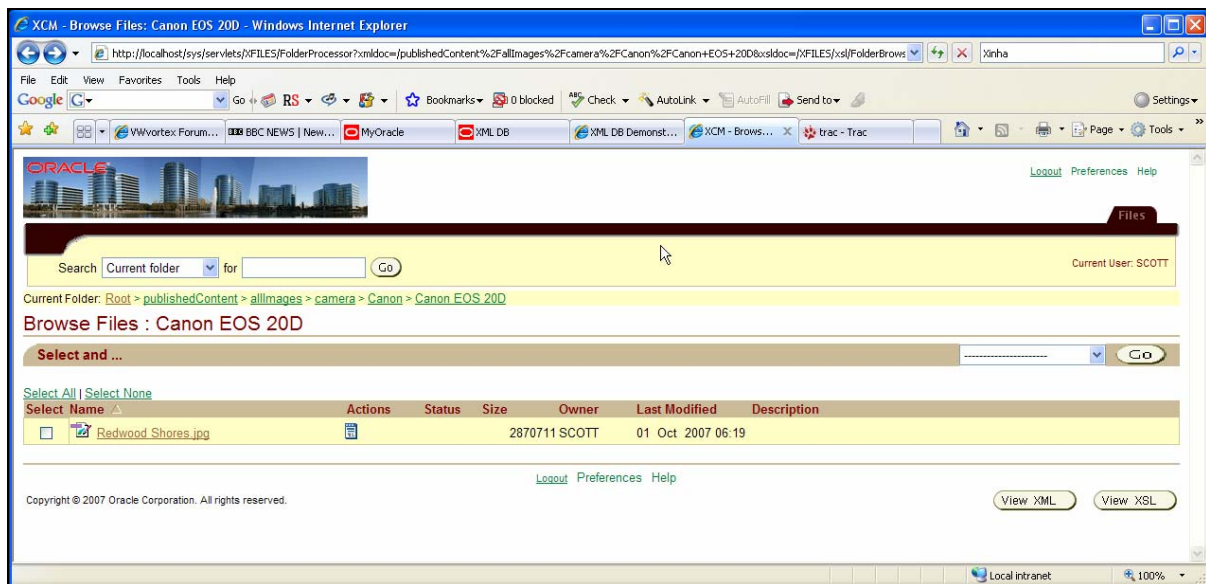
This step uses the X-Files application to show how the photographs were automatically categorized by Camera make and model as part of the metadata extraction process. Click the icon to launch the X-Files application and open the folder /publishedContent/allImages/camera



This folder contains a subfolder for each camera manufacturer that was used to take the photographs loaded in the repository. All the photographs supplied with the demonstration were taken with canon cameras, so only one subfolder appears in folder camera. Click Canon to open the folder.



Folder canon contains one subfolder for each model of Canon camera that was used to take the photographs loaded in the repository. A number of different models of Canon camera were used to take the photographs supplied with the demonstration. Click the Canon EOS 20D link to see the photographs that were taken using this model.

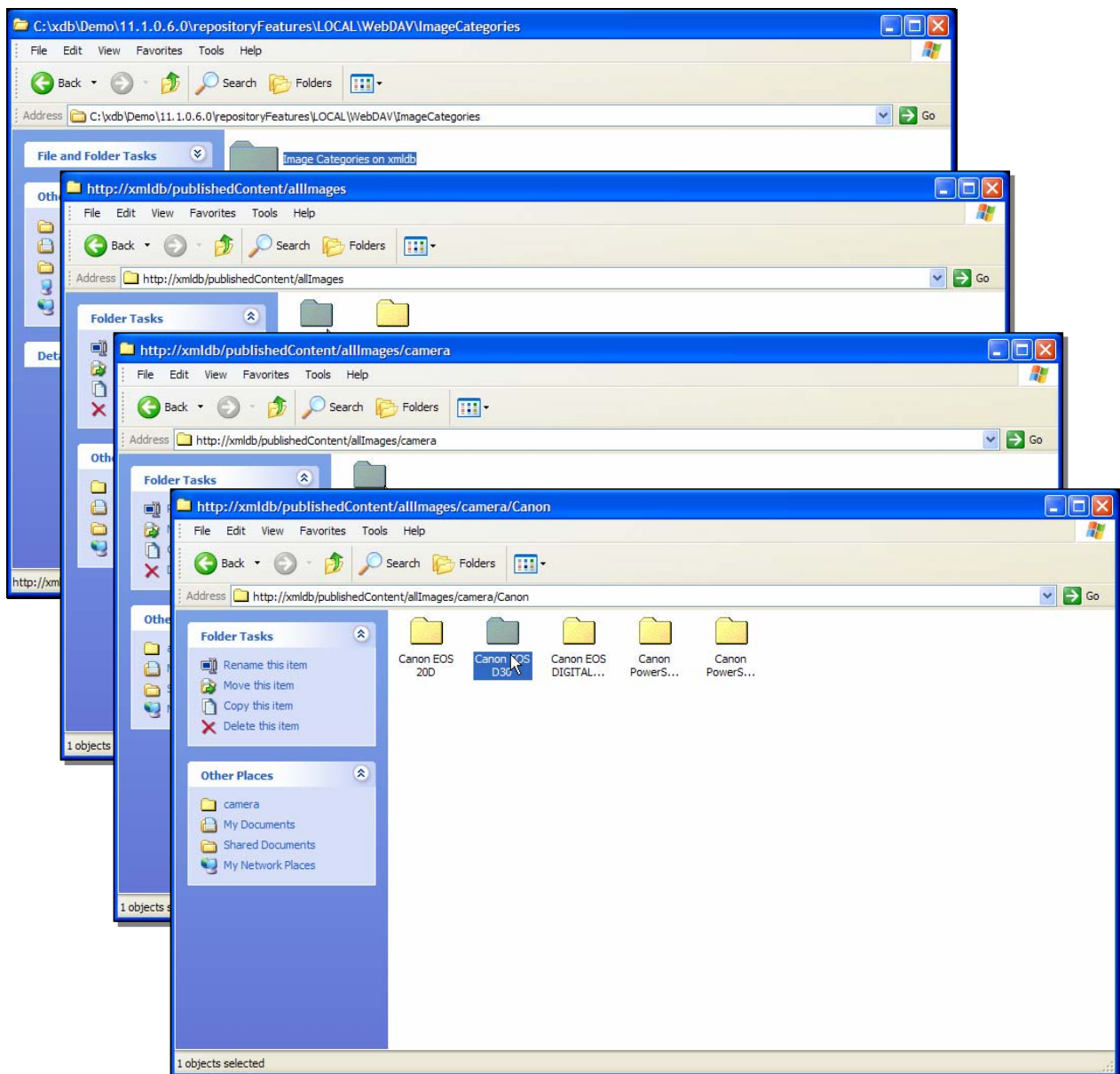


The only photograph taken with a Canon EOS 20D was Redwood Shores.jpg.

6.4 View Image Categories (WebDav)

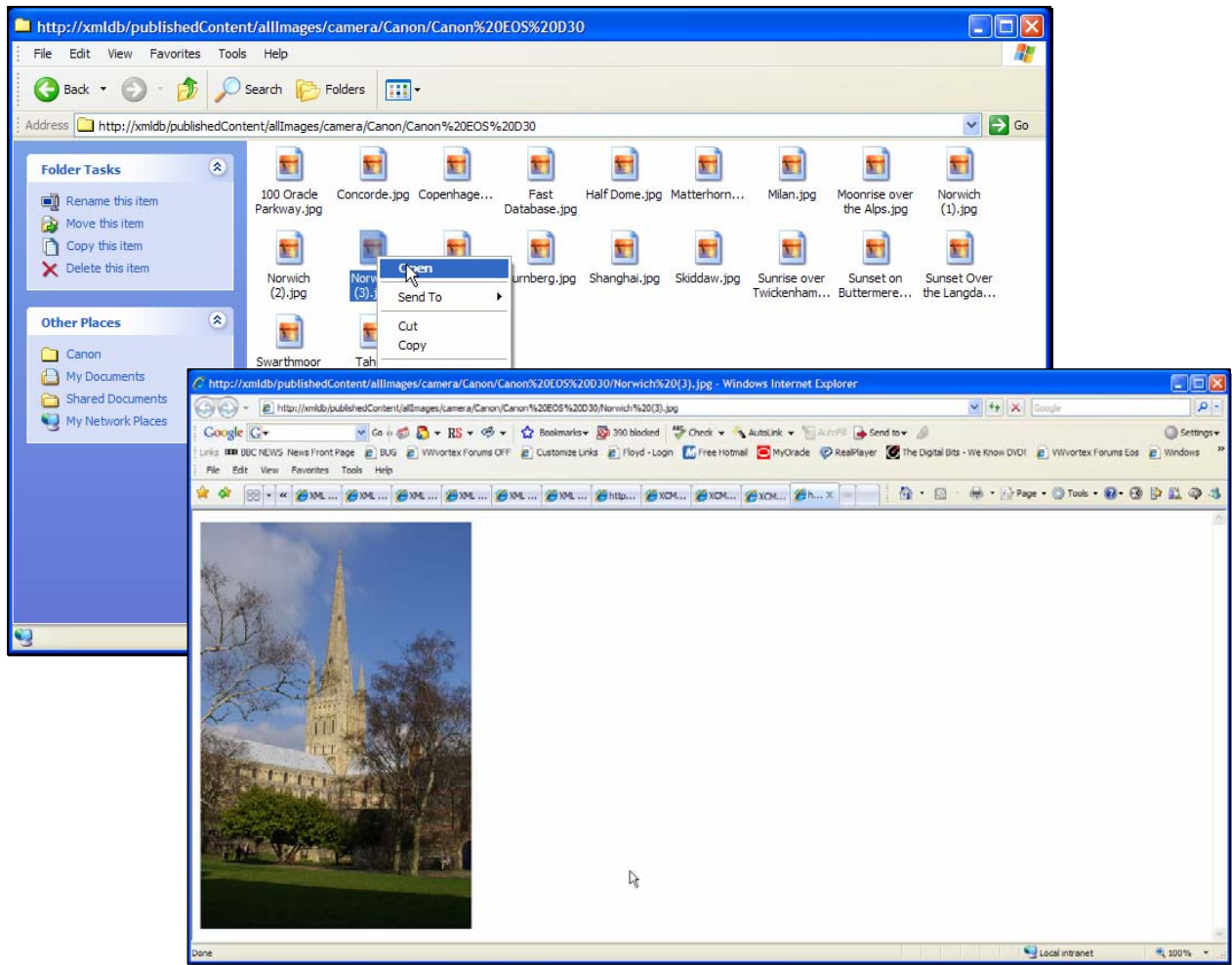
Using folders to categorize the photographs makes it possible to locate content by simply navigating the categorization folder tree using Window Explorer. This step shows how to use Windows Explorer to find all the photographs that were taken using a Canon EOS D30 camera.

Right click the icon and select explore. This will open a new window containing the local folder ImageCategories. The window contains shortcut called ImageCategories on xmldb. This shortcut is a link to the image categorization folder in the Oracle XML DB repository. Right click on the shortcut and select explore. This will open a new window containing folders camera and dateTaken. Right click on folder camera and select explore. This will open a new window containing folder Canon. Right click on folder Canon and click explore. This will open a new window containing the folders for each model of Canon camera.



Oracle XML DB 11gR1 Repository Features

Right click on Canon EOS 30D and click explore. This will open a folder containing all the pictures taken using a Canon EOS 30D camera. Right click one of the images and select open. This will open the image for viewing



The metadata extraction and categorization process has made it possible to locate photographs based on content.



Oracle Database 11g Oracle XML DB

October 2007

Author: Mark D Drake

Contributing Authors: Oracle XML DB Development Team

Oracle Corporation

World Headquarters

500 Oracle Parkway

Redwood Shores, CA 94065

U.S.A.

Worldwide Inquiries:

Phone: +1.650.506.7000

Fax: +1.650.506.7200

oracle.com

Copyright © 2007, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.