

A

Oracle-Supplied XML Schemas and Examples

Full listings are provided here for the Oracle XML DB-supplied XML schemas, purchase-order XML schemas and an XSLT stylesheet used in various examples, and C-language (OCI) examples for loading XML content into Oracle XML DB and initializing and terminating an XML context.

- [XDBResource.xsd: XML Schema for Oracle XML DB Resources](#)
A full listing is presented of the Oracle XML DB-supplied XML schema `XDBResource.xsd`, which is used to represent Oracle XML DB resources.
- [XDBResConfig.xsd: XML Schema for Resource Configuration](#)
A full listing is presented of the Oracle XML DB-supplied XML schema used to configure repository resources. It is accessible in Oracle XML DB Repository at path `/sys/schemas/PUBLIC/xmlns.oracle.com/xdb/XDBResConfig.xsd`.
- [acl.xsd: XML Schema for ACLs](#)
The Oracle Database-supplied XML schema used to represent access control lists (ACLs) is presented.
- [xdbconfig.xsd: XML Schema for Configuring Oracle XML DB](#)
A full listing is presented of file `xdbconfig.xsd`, which contains the XML schema used to configure Oracle XML DB.
- [xdiff.xsd: XML Schema for Comparing Schemas for In-Place Evolution](#)
A full listing is presented of `xdiff.xsd`, the Oracle XML DB-supplied XML schema to which the document specified as the `diffXML` parameter to procedure `DBMS_XMLSCHEMA.inPlaceEvolve` must conform.
- [Purchase-Order XML Schemas](#)
The full listings of purchase-order XML schemas that are used in various examples are presented.
- [XSLT Stylesheet Example, PurchaseOrder.xsl](#)
The full listing is presented of XSLT stylesheet `PurchaseOrder.xsl`, which is used in various examples.
- [Loading XML Data Using C \(OCI\)](#)
A full listing of a C program that inserts XML data into an `XMLType` table is presented.
- [Initializing and Terminating an XML Context \(OCI\)](#)
An example shows how to use OCI functions `OCIXmlDbInitXmlCtx()` and `OCIXmlDbFreeXmlCtx()` to initialize and terminate an XML context. It constructs an XML document using the C DOM API and saves it to the database.

XDBResource.xsd: XML Schema for Oracle XML DB Resources

A full listing is presented of the Oracle XML DB-supplied XML schema `XDBResource.xsd`, which is used to represent Oracle XML DB resources.

XDBResource.xsd

```
<schema xdb:schemaURL="http://xmlns.oracle.com/xdb/XDBResource.xsd"
  targetNamespace="http://xmlns.oracle.com/xdb/XDBResource.xsd" version="1.0"
  xdb:numProps="73" elementFormDefault="qualified" xdb:flags="23"
  xdb:mapStringToNCHAR="false" xdb:mapUnboundedStringToLob="false"
  xdb:storeVarrayAsTable="false" xdb:schemaOwner="XDB"
  xmlns="http://www.w3.org/2001/XMLSchema" xmlns:xdb="http://xmlns.oracle.com/xdb"
  xmlns:xdbres="http://xmlns.oracle.com/xdb/XDBResource.xsd">
  <simpleType name="OracleUserName">
    <restriction base="string">
      <minLength value="1" fixed="false"/>
      <maxLength value="4000" fixed="false"/>
    </restriction>
  </simpleType>
  <simpleType name="ResMetaStr">
    <restriction base="string">
      <minLength value="1" fixed="false"/>
      <maxLength value="128" fixed="false"/>
    </restriction>
  </simpleType>
  <simpleType name="SchElemType">
    <restriction base="string">
      <minLength value="1" fixed="false"/>
      <maxLength value="4000" fixed="false"/>
    </restriction>
  </simpleType>
  <simpleType name="GUID">
    <restriction base="hexBinary">
      <minLength value="8" fixed="false"/>
      <maxLength value="32" fixed="false"/>
    </restriction>
  </simpleType>
  <simpleType name="LocksRaw">
    <restriction base="hexBinary">
      <minLength value="0" fixed="false"/>
      <maxLength value="2000" fixed="false"/>
    </restriction>
  </simpleType>
  <simpleType name="lockModeType">
    <restriction base="string">
      <enumeration value="exclusive" fixed="false"/>
      <enumeration value="shared" fixed="false"/>
    </restriction>
  </simpleType>
  <simpleType name="lockTypeType">
    <restriction base="string">
      <enumeration value="read-write" fixed="false"/>
      <enumeration value="write" fixed="false"/>
      <enumeration value="read" fixed="false"/>
    </restriction>
  </simpleType>
  <simpleType name="lockDepthType">
    <restriction base="string">
      <enumeration value="0" fixed="false"/>
    </restriction>
  </simpleType>
```

```

    <enumeration value="infinity" fixed="false"/>
  </restriction>
</simpleType>
<complexType name="lockType" abstract="false" mixed="false">
  <sequence minOccurs="1" maxOccurs="1">
    <element xdb:propNumber="768" name="LockOwner" type="string" xdb:memType="1"
      xdb:system="false" xdb:mutable="true" xdb:JavaType="String"
      xdb:global="false" nillable="false" abstract="false" xdb:SQLInline="true"
      xdb:JavaInline="false" xdb:MemInline="true" xdb:maintainDOM="false"
      xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="1" maxOccurs="1"/>
    <element xdb:propNumber="769" name="Mode" type="xdb:lockModeType" xdb:memType="1"
      xdb:system="false" xdb:mutable="true" xdb:JavaType="String"
      xdb:global="false" nillable="false" abstract="false" xdb:SQLInline="true"
      xdb:JavaInline="false" xdb:MemInline="true" xdb:maintainDOM="false"
      xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="1" maxOccurs="1"/>
    <element xdb:propNumber="770" name="Type" type="xdb:lockTypeType" xdb:memType="1"
      xdb:system="false" xdb:mutable="true" xdb:JavaType="String"
      xdb:global="false" nillable="false" abstract="false" xdb:SQLInline="true"
      xdb:JavaInline="false" xdb:MemInline="true" xdb:maintainDOM="false"
      xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="1" maxOccurs="1"/>
    <element xdb:propNumber="771" name="Depth" type="xdb:lockDepthType" xdb:memType="1"
      xdb:system="false" xdb:mutable="true" xdb:JavaType="String"
      xdb:global="false" nillable="false" abstract="false" xdb:SQLInline="true"
      xdb:JavaInline="false" xdb:MemInline="true" xdb:maintainDOM="false"
      xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="1" maxOccurs="1"/>
    <element xdb:propNumber="772" name="Expiry" type="dateTime" xdb:memType="180"
      xdb:system="false" xdb:mutable="true" xdb:JavaType="TimeStamp"
      xdb:global="false" nillable="false" abstract="false" xdb:SQLInline="true"
      xdb:JavaInline="false" xdb:MemInline="true" xdb:maintainDOM="false"
      xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="0" maxOccurs="1"/>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element xdb:propNumber="773" name="Token" type="string" xdb:memType="1"
        xdb:system="false" xdb:mutable="true" xdb:JavaType="String"
        xdb:global="false" nillable="false" abstract="false" xdb:SQLInline="true"
        xdb:JavaInline="false" xdb:MemInline="true" xdb:maintainDOM="false"
        xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="0" maxOccurs="1"/>
      <element xdb:propNumber="774" name="NodeId" type="string" xdb:memType="1"
        xdb:system="false" xdb:mutable="true" xdb:JavaType="String"
        xdb:global="false" nillable="false" abstract="false" xdb:SQLInline="true"
        xdb:JavaInline="false" xdb:MemInline="true" xdb:maintainDOM="false"
        xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="0" maxOccurs="1"/>
    </choice>
  </sequence>
</complexType>
<complexType name="locksType" abstract="false" mixed="false">
  <sequence minOccurs="1" maxOccurs="1">
    <element xdb:propNumber="767" name="Lock" type="xdb:lockType" xdb:memType="258"
      xdb:system="false" xdb:mutable="true" xdb:JavaType="XMLType"
      xdb:global="false" nillable="false" abstract="false" xdb:SQLInline="true"
      xdb:JavaInline="false" xdb:MemInline="true" xdb:maintainDOM="false"
      xdb:defaultTableSchema="XDB" minOccurs="0" maxOccurs="2147483647"/>
  </sequence>
</complexType>
<complexType name="ResContentsType" abstract="false" mixed="false">
  <sequence minOccurs="1" maxOccurs="1">
    <any xdb:propNumber="736" name="ContentsAny" xdb:memType="258" xdb:system="false"

```

```
        xdb:mutable="false" xdb:JavaType="XMLType" minOccurs="0" maxOccurs="1"/>
    </sequence>
</complexType>
<complexType name="ResAclType" abstract="false" mixed="false">
    <sequence minOccurs="1" maxOccurs="1">
        <any xdb:propNumber="737" name="ACLAny" xdb:memType="258" xdb:system="false"
            xdb:mutable="false" xdb:JavaType="XMLType" minOccurs="0" maxOccurs="1"/>
    </sequence>
</complexType>
<complexType name="AttrCopyType" abstract="false" mixed="false">
    <sequence minOccurs="1" maxOccurs="1">
        <any xdb:propNumber="748" name="AttrCopyAny" xdb:memType="258" xdb:system="false"
            xdb:mutable="false" xdb:JavaType="XMLType" minOccurs="0" maxOccurs="65535"/>
    </sequence>
</complexType>
<complexType name="RCListType" abstract="false" mixed="false">
    <sequence minOccurs="1" maxOccurs="1">
        <element xdb:propNumber="755" name="OID" type="hexBinary" xdb:memByteLength="22"
            xdb:memType="23" xdb:system="false" xdb:mutable="false" xdb:SQLName="OID"
            xdb:SQLType="RAW" xdb:JavaType="byteArray" xdb:global="false"
            xdb:SQLCollType="XDB$OID_LIST_T" xdb:SQLCollSchema="XDB" xdb:hidden="false"
            nillable="false" abstract="false" xdb:SQLInline="true"
            xdb:JavaInline="false" xdb:MemInline="true" xdb:maintainDOM="false"
            xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="1"
            maxOccurs="65535"/>
    </sequence>
</complexType>
<complexType name="ResourceType" abstract="false" mixed="false">
    <sequence minOccurs="1" maxOccurs="1">
        <element xdb:propNumber="709" name="CreationDate" type="dateTime" xdb:memType="180"
            xdb:system="false" xdb:mutable="false" xdb:SQLName="CREATIONDATE"
            xdb:SQLType="TIMESTAMP" xdb:JavaType="TimeStamp" xdb:global="false"
            nillable="false" abstract="false" xdb:SQLInline="true"
            xdb:JavaInline="false" xdb:MemInline="true" xdb:maintainDOM="false"
            xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="1" maxOccurs="1"/>
        <element xdb:propNumber="710" name="ModificationDate" type="dateTime"
            xdb:memType="180" xdb:system="false" xdb:mutable="false"
            xdb:SQLName="MODIFICATIONDATE" xdb:SQLType="TIMESTAMP"
            xdb:JavaType="TimeStamp" xdb:global="false" nillable="false"
            abstract="false" xdb:SQLInline="true" xdb:JavaInline="false"
            xdb:MemInline="true" xdb:maintainDOM="false" xdb:defaultTableSchema="XDB"
            xdb:numCols="1" minOccurs="1" maxOccurs="1"/>
        <element xdb:propNumber="711" name="Author" type="xdb:ResMetaStr" xdb:memType="1"
            xdb:system="false" xdb:mutable="false" xdb:SQLName="AUTHOR"
            xdb:SQLType="VARCHAR2" xdb:JavaType="String" xdb:global="false"
            nillable="false" abstract="false" xdb:SQLInline="true"
            xdb:JavaInline="false" xdb:MemInline="false" xdb:maintainDOM="false"
            xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="0" maxOccurs="1"/>
        <element xdb:propNumber="712" name="DisplayName" type="xdb:ResMetaStr"
            xdb:memType="1" xdb:system="false" xdb:mutable="false"
            xdb:SQLName="DISPNAME" xdb:SQLType="VARCHAR2" xdb:JavaType="String"
            xdb:global="false" nillable="false" abstract="false" xdb:SQLInline="true"
            xdb:JavaInline="false" xdb:MemInline="false" xdb:maintainDOM="false"
            xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="1" maxOccurs="1"/>
        <element xdb:propNumber="713" name="Comment" type="xdb:ResMetaStr" xdb:memType="1"
            xdb:system="false" xdb:mutable="false" xdb:SQLName="RESCOMMENT"
```

```

xdb:SQLType="VARCHAR2" xdb:JavaType="String" xdb:global="false"
nillable="false" abstract="false" xdb:SQLInline="true"
xdb:JavaInline="false" xdb:MemInline="false" xdb:maintainDOM="false"
xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="1" maxOccurs="1"/>
<element xdb:propNumber="714" name="Language" type="xdb:ResMetaStr" xdb:memType="1"
xdb:system="false" xdb:mutable="false" xdb:SQLName="LANGUAGE"
xdb:SQLType="VARCHAR2" xdb:JavaType="String" default="en"
xdb:global="false" nillable="false" abstract="false" xdb:SQLInline="true"
xdb:JavaInline="false" xdb:MemInline="false" xdb:maintainDOM="false"
xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="1" maxOccurs="1"/>
<element xdb:propNumber="715" name="CharacterSet" type="xdb:ResMetaStr"
xdb:memType="1" xdb:system="false" xdb:mutable="false"
xdb:SQLName="CHARSET" xdb:SQLType="VARCHAR2" xdb:JavaType="String"
xdb:global="false" nillable="false" abstract="false" xdb:SQLInline="true"
xdb:JavaInline="false" xdb:MemInline="false" xdb:maintainDOM="false"
xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="0" maxOccurs="1"/>
<element xdb:propNumber="716" name="ContentType" type="xdb:ResMetaStr"
xdb:memType="1" xdb:system="false" xdb:mutable="false"
xdb:SQLName="CONTYPE" xdb:SQLType="VARCHAR2" xdb:JavaType="String"
xdb:global="false" nillable="false" abstract="false" xdb:SQLInline="true"
xdb:JavaInline="false" xdb:MemInline="false" xdb:maintainDOM="false"
xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="0" maxOccurs="1"/>
<element xdb:propNumber="717" name="RefCount" type="nonNegativeInteger"
xdb:memByteLength="4" xdb:memType="68" xdb:system="false"
xdb:mutable="true" xdb:SQLName="REFCOUNT" xdb:SQLType="RAW"
xdb:JavaType="long" xdb:global="false" nillable="false" abstract="false"
xdb:SQLInline="true" xdb:JavaInline="false" xdb:MemInline="true"
xdb:maintainDOM="false" xdb:defaultTableSchema="XDB" xdb:numCols="1"
minOccurs="1" maxOccurs="1"/>
<element xdb:propNumber="718" name="LockBuf" type="xdb:LocksRaw" xdb:memType="23"
xdb:system="false" xdb:mutable="true" xdb:SQLName="LOCKS" xdb:SQLType="RAW"
xdb:JavaType="byteArray" xdb:global="false" nillable="false"
abstract="false" xdb:SQLInline="true" xdb:JavaInline="false"
xdb:MemInline="false" xdb:maintainDOM="false" xdb:defaultTableSchema="XDB"
xdb:numCols="1" minOccurs="0" maxOccurs="1"/>
<element xdb:propNumber="732" name="ACL" type="xdb:ResAclType" xdb:memType="258"
xdb:system="false" xdb:mutable="false" xdb:JavaType="XMLType"
xdb:global="false" xdb:hidden="false" xdb:transient="generated"
xdb:baseProp="false" nillable="false" abstract="false" xdb:SQLInline="true"
xdb:JavaInline="false" xdb:MemInline="false" xdb:maintainDOM="false"
xdb:defaultTableSchema="XDB" xdb:JavaClassname="oracle.xdb.ResAclTypeBean"
xdb:beanClassname="oracle.xdb.ResAclTypeBean" xdb:numCols="0" minOccurs="0"
maxOccurs="1"/>
<element xdb:propNumber="719" name="ACLOID" type="hexBinary" xdb:memType="23"
xdb:system="false" xdb:mutable="false" xdb:SQLName="ACLOID"
xdb:SQLType="RAW" xdb:JavaType="byteArray" xdb:global="false"
xdb:hidden="true" xdb:baseProp="true" nillable="false" abstract="false"
xdb:SQLInline="true" xdb:JavaInline="false" xdb:MemInline="true"
xdb:maintainDOM="false" xdb:defaultTableSchema="XDB" xdb:numCols="1"
minOccurs="1" maxOccurs="1"/>
<element xdb:propNumber="720" name="Owner" type="xdb:OracleUserName" xdb:memType="1"
xdb:system="false" xdb:mutable="false" xdb:JavaType="String"
xdb:global="false" xdb:hidden="false" xdb:transient="generated"
xdb:baseProp="false" nillable="false" abstract="false" xdb:SQLInline="true"
xdb:JavaInline="false" xdb:MemInline="false" xdb:maintainDOM="false"
xdb:defaultTableSchema="XDB" xdb:numCols="0" minOccurs="0" maxOccurs="1"/>

```

```
<element xdb:propNumber="721" name="OwnerID" type="xdb:GUID" xdb:memType="23"
  xdb:system="false" xdb:mutable="false" xdb:SQLName="OWNERID"
  xdb:SQLType="RAW" xdb:JavaType="byteArray" xdb:global="false"
  xdb:hidden="true" xdb:baseProp="true" nillable="false" abstract="false"
  xdb:SQLInline="true" xdb:JavaInline="false" xdb:MemInline="true"
  xdb:maintainDOM="false" xdb:defaultTableSchema="XDB" xdb:numCols="1"
  minOccurs="1" maxOccurs="1"/>
<element xdb:propNumber="722" name="Creator" type="xdb:OracleUserName"
  xdb:memType="1" xdb:system="false" xdb:mutable="false"
  xdb:JavaType="String" xdb:global="false" xdb:hidden="false"
  xdb:transient="generated" xdb:baseProp="false" nillable="false"
  abstract="false" xdb:SQLInline="true" xdb:JavaInline="false"
  xdb:MemInline="false" xdb:maintainDOM="false" xdb:defaultTableSchema="XDB"
  xdb:numCols="0" minOccurs="0" maxOccurs="1"/>
<element xdb:propNumber="723" name="CreatorID" type="xdb:GUID" xdb:memType="23"
  xdb:system="false" xdb:mutable="false" xdb:SQLName="CREATORID"
  xdb:SQLType="RAW" xdb:JavaType="byteArray" xdb:global="false"
  xdb:hidden="true" xdb:baseProp="true" nillable="false" abstract="false"
  xdb:SQLInline="true" xdb:JavaInline="false" xdb:MemInline="true"
  xdb:maintainDOM="false" xdb:defaultTableSchema="XDB" xdb:numCols="1"
  minOccurs="1" maxOccurs="1"/>
<element xdb:propNumber="724" name="LastModifier" type="xdb:OracleUserName"
  xdb:memType="1" xdb:system="false" xdb:mutable="false"
  xdb:JavaType="String" xdb:global="false" xdb:hidden="false"
  xdb:transient="generated" xdb:baseProp="false" nillable="false"
  abstract="false" xdb:SQLInline="true" xdb:JavaInline="false"
  xdb:MemInline="false" xdb:maintainDOM="false" xdb:defaultTableSchema="XDB"
  xdb:numCols="0" minOccurs="0" maxOccurs="1"/>
<element xdb:propNumber="725" name="LastModifierID" type="xdb:GUID" xdb:memType="23"
  xdb:system="false" xdb:mutable="false" xdb:SQLName="LASTMODIFIERID"
  xdb:SQLType="RAW" xdb:JavaType="byteArray" xdb:global="false"
  xdb:hidden="true" xdb:baseProp="true" nillable="false" abstract="false"
  xdb:SQLInline="true" xdb:JavaInline="false" xdb:MemInline="true"
  xdb:maintainDOM="false" xdb:defaultTableSchema="XDB" xdb:numCols="1"
  minOccurs="1" maxOccurs="1"/>
<element xdb:propNumber="726" name="SchemaElement" type="xdb:SchElemType"
  xdb:memType="1" xdb:system="false" xdb:mutable="false"
  xdb:JavaType="String" xdb:global="false" xdb:hidden="false"
  xdb:transient="generated" xdb:baseProp="false" nillable="false"
  abstract="false" xdb:SQLInline="true" xdb:JavaInline="false"
  xdb:MemInline="false" xdb:maintainDOM="false" xdb:defaultTableSchema="XDB"
  xdb:numCols="0" minOccurs="0" maxOccurs="1"/>
<element xdb:propNumber="727" name="ElNum" type="nonNegativeInteger"
  xdb:memByteLength="4" xdb:memType="3" xdb:system="false"
  xdb:mutable="false" xdb:SQLName="ELNUM" xdb:SQLType="INTEGER"
  xdb:JavaType="long" xdb:global="false" xdb:hidden="true"
  xdb:baseProp="true" nillable="false" abstract="false" xdb:SQLInline="true"
  xdb:JavaInline="false" xdb:MemInline="true" xdb:maintainDOM="false"
  xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="1" maxOccurs="1"/>
<element xdb:propNumber="728" name="SchOID" type="hexBinary" xdb:memType="23"
  xdb:system="false" xdb:mutable="false" xdb:SQLName="SCHOID"
  xdb:SQLType="RAW" xdb:JavaType="byteArray" xdb:global="false"
  xdb:hidden="true" xdb:baseProp="true" nillable="false" abstract="false"
  xdb:SQLInline="true" xdb:JavaInline="false" xdb:MemInline="true"
  xdb:maintainDOM="false" xdb:defaultTableSchema="XDB" xdb:numCols="1"
  minOccurs="1" maxOccurs="1"/>
```

```
<element xdb:propNumber="733" name="Contents" type="xdb:ResContentsType"
  xdb:memType="258" xdb:system="false" xdb:mutable="false"
  xdb:JavaType="XMLType" xdb:global="false" xdb:hidden="false"
  xdb:transient="manifested" xdb:baseProp="false" nillable="false"
  abstract="false" xdb:SQLInline="true" xdb:JavaInline="false"
  xdb:MemInline="false" xdb:maintainDOM="false" xdb:defaultTableSchema="XDB"
  xdb:JavaClassname="oracle.xdb.ResContentsTypeBean"
  xdb:beanClassname="oracle.xdb.ResContentsTypeBean" xdb:numCols="0"
  minOccurs="0" maxOccurs="1"/>
<element xdb:propNumber="729" name="XMLRef" type="REF" xdb:memType="110"
  xdb:system="false" xdb:mutable="true" xdb:SQLName="XMLREF"
  xdb:SQLType="REF" xdb:JavaType="Reference" xdb:global="false"
  xdb:hidden="true" xdb:baseProp="false" nillable="false" abstract="false"
  xdb:SQLInline="true" xdb:JavaInline="false" xdb:MemInline="false"
  xdb:maintainDOM="false" xdb:defaultTableSchema="XDB" xdb:numCols="0"
  minOccurs="0" maxOccurs="1"/>
<element xdb:propNumber="730" name="XMLLob" type="hexBinary" xdb:memType="113"
  xdb:system="false" xdb:mutable="true" xdb:SQLName="XMLLOB"
  xdb:SQLType="BLOB" xdb:JavaType="String" xdb:global="false"
  xdb:hidden="true" xdb:baseProp="false" nillable="false" abstract="false"
  xdb:SQLInline="true" xdb:JavaInline="false" xdb:MemInline="false"
  xdb:maintainDOM="false" xdb:defaultTableSchema="XDB" xdb:numCols="0"
  minOccurs="0" maxOccurs="1"/>
<element xdb:propNumber="731" name="Flags" type="nonNegativeInteger"
  xdb:memByteLength="4" xdb:memType="3" xdb:system="false" xdb:mutable="true"
  xdb:SQLName="FLAGS" xdb:SQLType="RAW" xdb:JavaType="long"
  xdb:global="false" xdb:hidden="true" xdb:baseProp="true" nillable="false"
  abstract="false" xdb:SQLInline="true" xdb:JavaInline="false"
  xdb:MemInline="true" xdb:maintainDOM="false" xdb:defaultTableSchema="XDB"
  xdb:numCols="0" minOccurs="1" maxOccurs="1"/>
<element xdb:propNumber="740" name="VCRUID" type="xdb:GUID" xdb:memType="23"
  xdb:system="false" xdb:mutable="false" xdb:SQLName="VCRUID"
  xdb:SQLType="RAW" xdb:JavaType="byteArray" xdb:global="false"
  nillable="false" abstract="false" xdb:SQLInline="true"
  xdb:JavaInline="false" xdb:MemInline="false" xdb:maintainDOM="false"
  xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="1" maxOccurs="1"/>
<element xdb:propNumber="741" name="Parents" type="hexBinary" xdb:memType="23"
  xdb:system="false" xdb:mutable="false" xdb:SQLName="PARENTS"
  xdb:SQLType="RAW" xdb:JavaType="Reference" xdb:global="false"
  xdb:SQLCollType="XDB$PREDECESSOR_LIST_T" xdb:SQLCollSchema="XDB"
  nillable="false" abstract="false" xdb:SQLInline="true"
  xdb:JavaInline="false" xdb:MemInline="false" xdb:maintainDOM="false"
  xdb:defaultTableSchema="XDB" xdb:numCols="0" minOccurs="0"
  maxOccurs="1000"/>
<element xdb:propNumber="745" name="SBResExtra" type="REF" xdb:memType="110"
  xdb:system="false" xdb:mutable="true" xdb:SQLName="SBRESEXTA"
  xdb:SQLType="REF" xdb:JavaType="Reference" xdb:global="false"
  xdb:SQLCollType="XDB$XMLTYPE_REF_LIST_T" xdb:SQLCollSchema="XDB"
  xdb:hidden="true" xdb:baseProp="false" nillable="false" abstract="false"
  xdb:SQLInline="true" xdb:JavaInline="false" xdb:MemInline="false"
  xdb:maintainDOM="false" xdb:defaultTableSchema="XDB" xdb:numCols="0"
  minOccurs="0" maxOccurs="2147483647"/>
<element xdb:propNumber="746" name="Snapshot" type="hexBinary" xdb:memType="23"
  xdb:system="false" xdb:mutable="true" xdb:SQLName="SNAPSHOT"
  xdb:SQLType="RAW" xdb:JavaType="byteArray" xdb:global="false"
  xdb:hidden="true" xdb:baseProp="true" nillable="false" abstract="false"
```

```

xdb:SQLInline="true" xdb:JavaInline="false" xdb:MemInline="true"
xdb:maintainDOM="false" xdb:defaultTableSchema="XDB" xdb:numCols="0"
minOccurs="1" maxOccurs="1"/>
<element xdb:propNumber="747" name="AttrCopy" type="xdb:AttrCopyType"
xdb:memType="258" xdb:system="false" xdb:mutable="true"
xdb:SQLName="ATTRCOPY" xdb:SQLType="BLOB" xdb:JavaType="XMLType"
xdb:global="false" xdb:hidden="true" xdb:baseProp="true" nillable="false"
abstract="false" xdb:SQLInline="true" xdb:JavaInline="false"
xdb:MemInline="false" xdb:maintainDOM="false" xdb:defaultTableSchema="XDB"
xdb:numCols="0" minOccurs="0" maxOccurs="1"/>
<element xdb:propNumber="749" name="CtsCopy" type="hexBinary" xdb:memType="113"
xdb:system="false" xdb:mutable="true" xdb:SQLName="CTSCOPY"
xdb:SQLType="BLOB" xdb:JavaType="String" xdb:global="false"
xdb:hidden="true" xdb:baseProp="false" nillable="false" abstract="false"
xdb:SQLInline="true" xdb:JavaInline="false" xdb:MemInline="false"
xdb:maintainDOM="false" xdb:defaultTableSchema="XDB" xdb:numCols="1"
minOccurs="0" maxOccurs="1"/>
<element xdb:propNumber="750" name="NodeNum" type="hexBinary" xdb:memType="23"
xdb:system="false" xdb:mutable="true" xdb:SQLName="NODENUM"
xdb:SQLType="RAW" xdb:JavaType="byteArray" xdb:global="false"
xdb:hidden="true" xdb:baseProp="true" nillable="false" abstract="false"
xdb:SQLInline="true" xdb:JavaInline="false" xdb:MemInline="true"
xdb:maintainDOM="false" xdb:defaultTableSchema="XDB" xdb:numCols="0"
minOccurs="1" maxOccurs="1"/>
<element xdb:propNumber="751" name="ContentSize" type="integer"
xdb:memByteLength="8" xdb:memType="3" xdb:system="false"
xdb:mutable="false" xdb:JavaType="long" xdb:global="false"
xdb:hidden="true" xdb:transient="generated" xdb:baseProp="false"
nillable="false" abstract="false" xdb:SQLInline="true"
xdb:JavaInline="false" xdb:MemInline="true" xdb:maintainDOM="false"
xdb:defaultTableSchema="XDB" xdb:numCols="0" minOccurs="0" maxOccurs="1"/>
<element xdb:propNumber="752" name="SizeOnDisk" type="nonNegativeInteger"
xdb:memByteLength="8" xdb:memType="3" xdb:system="false"
xdb:mutable="false" xdb:SQLName="SIZEONDISK" xdb:SQLType="INTEGER"
xdb:JavaType="long" xdb:global="false" xdb:hidden="true"
xdb:baseProp="true" nillable="false" abstract="false" xdb:SQLInline="true"
xdb:JavaInline="false" xdb:MemInline="true" xdb:maintainDOM="false"
xdb:defaultTableSchema="XDB" xdb:numCols="0" minOccurs="0" maxOccurs="1"/>
<element xdb:propNumber="754" name="RCList" type="xdb:RCListType" xdb:memType="258"
xdb:system="false" xdb:mutable="false" xdb:SQLName="RCLIST"
xdb:SQLType="XDB$RCLIST_T" xdb:SQLSchema="XDB" xdb:JavaType="XMLType"
xdb:global="true" xdb:hidden="true" xdb:baseProp="true" nillable="false"
abstract="false" xdb:SQLInline="true" xdb:JavaInline="false"
xdb:MemInline="false" xdb:maintainDOM="false" xdb:defaultTable="00"
xdb:defaultTableSchema="XDB" xdb:JavaClassname="oracle.xdb.RCListBean"
xdb:beanClassname="oracle.xdb.RCListBean" xdb:numCols="1" minOccurs="0"
maxOccurs="1"/>
<element xdb:propNumber="762" name="Branch" type="string" xdb:memType="1"
xdb:system="false" xdb:mutable="false" xdb:SQLName="BRANCH"
xdb:SQLType="VARCHAR2" xdb:JavaType="String" xdb:global="false"
xdb:hidden="false" xdb:transient="generated" xdb:baseProp="false"
nillable="false" abstract="false" xdb:SQLInline="true"
xdb:JavaInline="false" xdb:MemInline="false" xdb:maintainDOM="false"
xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="0" maxOccurs="1"/>
<element xdb:propNumber="763" name="CheckedOutBy" type="xdb:OracleUserName"
xdb:memType="1" xdb:system="false" xdb:mutable="false"

```



```

xdb:JavaType="String" xdb:global="false" xdb:hidden="false"
xdb:transient="generated" xdb:baseProp="false" nillable="false"
abstract="false" xdb:SQLInline="true" xdb:JavaInline="false"
xdb:MemInline="false" xdb:maintainDOM="false" xdb:defaultTableSchema="XDB"
xdb:numCols="0" minOccurs="0" maxOccurs="1"/>
<element xdb:propNumber="764" name="CheckedOutByID" type="xdb:GUID" xdb:memType="23"
xdb:system="false" xdb:mutable="false" xdb:SQLName="CHECKEDOUTBYID"
xdb:SQLType="RAW" xdb:JavaType="byteArray" xdb:global="false"
xdb:hidden="true" xdb:baseProp="true" nillable="false" abstract="false"
xdb:SQLInline="true" xdb:JavaInline="false" xdb:MemInline="true"
xdb:maintainDOM="false" xdb:defaultTableSchema="XDB" xdb:numCols="1"
minOccurs="0" maxOccurs="1"/>
<element xdb:propNumber="765" name="BaseVersion" type="hexBinary" xdb:memType="23"
xdb:system="false" xdb:mutable="false" xdb:SQLName="BASEVERSION"
xdb:SQLType="RAW" xdb:JavaType="byteArray" xdb:global="false"
xdb:hidden="false" xdb:baseProp="false" nillable="false" abstract="false"
xdb:SQLInline="true" xdb:JavaInline="false" xdb:MemInline="true"
xdb:maintainDOM="false" xdb:defaultTableSchema="XDB" xdb:numCols="1"
minOccurs="0" maxOccurs="1"/>
<element xdb:propNumber="766" name="Locks" type="xdb:locksType" xdb:memType="258"
xdb:system="false" xdb:mutable="true" xdb:JavaType="XMLType"
xdb:global="false" xdb:hidden="true" xdb:transient="generated"
nillable="false" abstract="false" xdb:SQLInline="true"
xdb:JavaInline="false" xdb:MemInline="false" xdb:maintainDOM="false"
xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="0" maxOccurs="1"/>
<any xdb:propNumber="735" name="ResExtra" xdb:memType="258" xdb:system="false"
xdb:mutable="false" xdb:SQLName="RESEXTRA" xdb:SQLType="CLOB"
xdb:JavaType="XMLType" namespace="##other" minOccurs="0" maxOccurs="65535"/>
</sequence>
<attribute xdb:propNumber="705" name="Hidden" type="boolean" xdb:memByteLength="1"
xdb:memType="252" xdb:system="false" xdb:mutable="false"
xdb:JavaType="boolean" default="false" xdb:hidden="false"
xdb:transient="generated" xdb:baseProp="false"/>
<attribute xdb:propNumber="706" name="Invalid" type="boolean" xdb:memByteLength="1"
xdb:memType="252" xdb:system="false" xdb:mutable="false"
xdb:JavaType="boolean" default="false" xdb:hidden="false"
xdb:transient="generated" xdb:baseProp="false"/>
<attribute xdb:propNumber="707" name="VersionID" type="integer" xdb:memByteLength="4"
xdb:memType="3" xdb:system="false" xdb:mutable="false"
xdb:SQLName="VERSIONID" xdb:SQLType="INTEGER" xdb:JavaType="long"/>
<attribute xdb:propNumber="708" name="ActivityID" type="integer" xdb:memByteLength="4"
xdb:memType="3" xdb:system="false" xdb:mutable="false"
xdb:SQLName="ACTIVITYID" xdb:SQLType="INTEGER" xdb:JavaType="long"/>
<attribute xdb:propNumber="738" name="Container" type="boolean" xdb:memByteLength="1"
xdb:memType="252" xdb:system="false" xdb:mutable="true"
xdb:JavaType="boolean" default="false" xdb:hidden="false"
xdb:transient="generated" xdb:baseProp="false"/>
<attribute xdb:propNumber="739" name="CustomRslv" type="boolean" xdb:memByteLength="1"
xdb:memType="252" xdb:system="false" xdb:mutable="false"
xdb:JavaType="boolean" default="false" xdb:hidden="false"
xdb:transient="generated" xdb:baseProp="false"/>
<attribute xdb:propNumber="742" name="VersionHistory" type="boolean"
xdb:memByteLength="1" xdb:memType="252" xdb:system="false"
xdb:mutable="false" xdb:JavaType="boolean" default="false"
xdb:hidden="false" xdb:transient="generated" xdb:baseProp="false"/>
<attribute xdb:propNumber="743" name="StickyRef" type="boolean" xdb:memByteLength="1"

```

```

        xdb:memType="252" xdb:system="false" xdb:mutable="false"
        xdb:JavaType="boolean" default="false" xdb:hidden="false"
        xdb:transient="generated" xdb:baseProp="false"/>
<attribute xdb:propNumber="744" name="HierSchmResource" type="boolean"
        xdb:memByteLength="1" xdb:memType="252" xdb:system="false"
        xdb:mutable="false" xdb:JavaType="boolean" default="false"
        xdb:hidden="true" xdb:transient="generated" xdb:baseProp="false"/>
<attribute xdb:propNumber="753" name="SizeAccurate" type="boolean"
        xdb:memByteLength="1" xdb:memType="252" xdb:system="false"
        xdb:mutable="false" xdb:JavaType="boolean" default="false"
        xdb:hidden="true" xdb:transient="generated" xdb:baseProp="false"/>
<attribute xdb:propNumber="756" name="IsVersionable" type="boolean"
        xdb:memByteLength="1" xdb:memType="252" xdb:system="false"
        xdb:mutable="false" xdb:JavaType="boolean" default="false"
        xdb:hidden="true" xdb:transient="generated" xdb:baseProp="false"/>
<attribute xdb:propNumber="757" name="IsCheckedOut" type="boolean"
        xdb:memByteLength="1" xdb:memType="252" xdb:system="false"
        xdb:mutable="false" xdb:JavaType="boolean" default="false"
        xdb:hidden="true" xdb:transient="generated" xdb:baseProp="false"/>
<attribute xdb:propNumber="758" name="IsVersion" type="boolean" xdb:memByteLength="1"
        xdb:memType="252" xdb:system="false" xdb:mutable="false"
        xdb:JavaType="boolean" default="false" xdb:hidden="true"
        xdb:transient="generated" xdb:baseProp="false"/>
<attribute xdb:propNumber="759" name="IsVCR" type="boolean" xdb:memByteLength="1"
        xdb:memType="252" xdb:system="false" xdb:mutable="false"
        xdb:JavaType="boolean" default="false" xdb:hidden="true"
        xdb:transient="generated" xdb:baseProp="false"/>
<attribute xdb:propNumber="760" name="IsVersionHistory" type="boolean"
        xdb:memByteLength="1" xdb:memType="252" xdb:system="false"
        xdb:mutable="false" xdb:JavaType="boolean" default="false"
        xdb:hidden="true" xdb:transient="generated" xdb:baseProp="false"/>
<attribute xdb:propNumber="761" name="IsWorkspace" type="boolean"
        xdb:memByteLength="1" xdb:memType="252" xdb:system="false"
        xdb:mutable="false" xdb:JavaType="boolean" default="false"
        xdb:hidden="true" xdb:transient="generated" xdb:baseProp="false"/>
<attribute xdb:propNumber="776" name="HasUnresolvedLinks" type="boolean"
        xdb:memByteLength="1" xdb:memType="252" xdb:system="false"
        xdb:mutable="false" xdb:JavaType="boolean" default="false"
        xdb:hidden="false" xdb:transient="generated" xdb:baseProp="false"/>
<attribute xdb:propNumber="777" name="IsXMLIndexed" type="boolean"
        xdb:memByteLength="1" xdb:memType="252" xdb:system="false"
        xdb:mutable="true" xdb:JavaType="boolean" default="false" xdb:hidden="true"
        xdb:transient="generated" xdb:baseProp="false"/>
</complexType>
<element xdb:propNumber="734" name="Resource" type="xdb:ResourceType" xdb:memType="258"
        xdb:system="false" xdb:mutable="false" xdb:SQLName="RESOURCE"
        xdb:SQLType="XDB$RESOURCE_T" xdb:SQLSchema="XDB" xdb:JavaType="XMLType"
        xdb:global="true" nillable="false" abstract="false" xdb:SQLInline="false"
        xdb:JavaInline="false" xdb:MemInline="false" xdb:maintainDOM="false"
        xdb:defaultTable="XDB$RESOURCE" xdb:defaultTableSchema="XDB"
        xdb:JavaClassname="oracle.xdb.ResourceBean"
        xdb:beanClassname="oracle.xdb.ResourceBean" xdb:numCols="33" minOccurs="1"
        maxOccurs="1"/>
<element xdb:propNumber="775" name="Locks" type="xdb:locksType" xdb:memType="258"
        xdb:system="false" xdb:mutable="true" xdb:JavaType="XMLType" xdb:global="false"
        xdb:hidden="false" nillable="false" abstract="false" xdb:SQLInline="true"

```

```

xdb:JavaInline="false" xdb:MemInline="false" xdb:maintainDOM="false"
xdb:defaultTableSchema="XDB" xdb:numCols="1" minOccurs="0" maxOccurs="1"/>
</schema>

```

XDBResConfig.xsd: XML Schema for Resource Configuration

A full listing is presented of the Oracle XML DB-supplied XML schema used to configure repository resources. It is accessible in Oracle XML DB Repository at path `/sys/schemas/PUBLIC/xmlns.oracle.com/xdb/XDBResConfig.xsd`.

The use of repository events to trigger application actions is deprecated in Oracle Database 21c (21.3). There is no replacement.

Repository events are events that can be used to trigger application actions. Repository events include repository changes, such as creating, deleting, locking, unlocking, rendering, linking, unlinking, placing under version control, checking in, checking out, unchecking out (reverting a checked out version), opening, and updating a resource. The deprecation of the use of repository events includes deprecation of the `DBMS_XEVENT` package, and the following subprogram groups:

XDBResConfig.xsd

```

<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://xmlns.oracle.com/xdb/XDBResConfig.xsd"
  xmlns:xdb="http://xmlns.oracle.com/xdb"
  xmlns:rescfg="http://xmlns.oracle.com/xdb/XDBResConfig.xsd"
  elementFormDefault="qualified" xdb:schemaOwner="XDB" version="1.0">
  <annotation>
    <documentation>
      This XML schema declares the schema of an XDB resource configuration,
      which includes default ACL, event listeners and user configuration.
      It lists all XDB repository events that will be supported.

      Future extension can be added to support user-defined events and
      XML events.
    </documentation>
  </annotation>
  <simpleType name="language">
    <restriction base="string">
      <enumeration value="Java"/>
      <enumeration value="C"/>
      <enumeration value="PL/SQL"/>
    </restriction>
  </simpleType>
  <complexType name="existsNode">
    <all>
      <element name="XPath" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="namespace" type="string" minOccurs="0" maxOccurs="1"/>
    </all>
  </complexType>
  <!-- listener pre-condition element -->
  <complexType name="condition">
    <all>
      <element name="existsNode" type="rescfg:existsNode" minOccurs="0" maxOccurs="1"/>
    </all>
  </complexType>

```

```

<complexType name="events">
  <all>
    <element name="Render" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Pre-Create" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Post-Create" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Pre-Delete" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Post-Delete" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Pre-Update" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Post-Update" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Pre-Lock" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Post-Lock" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Pre-Unlock" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Post-Unlock" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Pre-LinkIn" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Post-LinkIn" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Pre-LinkTo" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Post-LinkTo" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Pre-UnlinkIn" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Post-UnlinkIn" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Pre-UnlinkFrom" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Post-UnlinkFrom" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Pre-CheckIn" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Post-CheckIn" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Pre-CheckOut" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Post-CheckOut" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Pre-UncheckOut" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Post-UncheckOut" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Pre-VersionControl" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Post-VersionControl" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Pre-Open" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Post-Open" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Pre-InconsistentUpdate" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="Post-InconsistentUpdate" type="string" minOccurs="0" maxOccurs="1"/>
  </all>
</complexType>
<!-- event listener element -->
<complexType name="event-listener">
  <all>
    <element name="description" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="schema" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="source" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="language" type="rescfg:language" minOccurs="0" maxOccurs="1"/>
    <element name="pre-condition" type="rescfg:condition" minOccurs="0" maxOccurs="1"/>
    <element name="events" type="rescfg:events" minOccurs="0" maxOccurs="1"/>
  </all>
</complexType>
<complexType name="event-listeners">
  <sequence>
    <element name="listener" type="rescfg:event-listener" minOccurs="1"
maxOccurs="unbounded"/>
  </sequence>
  <attribute name="default-schema" type="string" xdb:baseProp="true" use="optional"/>
  <attribute name="default-language" type="rescfg:language" xdb:baseProp="true"
use="optional"/>
  <attribute name="set-invoker" type="boolean" xdb:baseProp="true" default="false"/>
</complexType>

```

```

<complexType name="defaultPath">
  <all>
    <element name="pre-condition" type="rescfg:condition" minOccurs="0" maxOccurs="1"/>
    <element name="path" type="string" minOccurs="0" maxOccurs="1"
xdb:transient="generated"/>
    <element name="resolvedpath" type="string" minOccurs="1" maxOccurs="1"
      xdb:baseProp="true" xdb:hidden="true"/>
    <element name="oid" type="hexBinary" minOccurs="1" maxOccurs="1"
      xdb:baseProp="true" xdb:hidden="true"/>
  </all>
</complexType>
<complexType name="defaultACL">
  <sequence>
    <element name="ACL" type="rescfg:defaultPath" minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
</complexType>
<complexType name="defaultConfig">
  <sequence>
    <element name="configuration" type="rescfg:defaultPath"
      minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
</complexType>
<simpleType name="link-type">
  <restriction base="string">
    <enumeration value="None"/>
    <enumeration value="Hard"/>
    <enumeration value="Weak"/>
    <enumeration value="Symbolic"/>
  </restriction>
</simpleType>
<simpleType name="path-format">
  <restriction base="string">
    <enumeration value="OID"/>
    <enumeration value="Named"/>
  </restriction>
</simpleType>
<simpleType name="link-metadata">
  <restriction base="string">
    <enumeration value="None"/>
    <enumeration value="Attributes"/>
    <enumeration value="All"/>
  </restriction>
</simpleType>
<simpleType name="unresolved-link">
  <restriction base="string">
    <enumeration value="Error"/>
    <enumeration value="SymLink"/>
    <enumeration value="Skip"/>
  </restriction>
</simpleType>
<simpleType name="conflict-rule">
  <restriction base="string">
    <enumeration value="Error"/>
    <enumeration value="Overwrite"/>
    <enumeration value="Syspath"/>
  </restriction>

```

```

</simpleType>
<simpleType name="section-type">
  <restriction base="string">
    <enumeration value="None"/>
    <enumeration value="Fragment"/>
    <enumeration value="Document"/>
  </restriction>
</simpleType>
<!-- XLinkConfig complex type -->
<complexType name="xlink-config">
  <sequence>
    <element name="LinkType" type="rescfg:link-type"/>
    <element name="PathFormat" type="rescfg:path-format" minOccurs="0" default="OID"/>
    <element name="LinkMetadata" type="rescfg:link-metadata" minOccurs="0" default="None"/>
    <element name="pre-condition" type="rescfg:condition" minOccurs="0" maxOccurs="1"/>
  </sequence>
  <attribute name="UnresolvedLink" type="rescfg:unresolved-link" default="Error"/>
</complexType>
<!-- XIncludeConfig element -->
<complexType name="xinclude-config">
  <sequence>
    <element name="LinkType" type="rescfg:link-type"/>
    <element name="PathFormat" type="rescfg:path-format" minOccurs="0" default="OID"/>
    <element name="ConflictRule" type="rescfg:conflict-rule" minOccurs="0"
default="Error"/>
  </sequence>
  <attribute name="UnresolvedLink" type="rescfg:unresolved-link" default="Error"/>
</complexType>
<!-- SectionConfig element -->
<complexType name="section-config">
  <sequence>
    <element name="Section" maxOccurs="unbounded">
      <complexType>
        <sequence>
          <element name="sectionPath" type="string"/>
          <element name="documentPath" type="string" minOccurs="0"/>
          <element name="namespace" type="string" minOccurs="0"/>
        </sequence>
        <attribute name="type" type="rescfg:section-type" default="None"/>
      </complexType>
    </element>
  </sequence>
</complexType>
<!-- ContentFormat element -->
<simpleType name="content-format">
  <restriction base="string">
    <enumeration value="text"/>
    <enumeration value="binary"/>
  </restriction>
</simpleType>
<!-- resource configuration element -->
<complexType name="ResConfig">
  <all>
    <element name="defaultChildConfig" type="rescfg:defaultConfig" minOccurs="0"
maxOccurs="1"/>
    <element name="defaultChildACL" type="rescfg:defaultACL" minOccurs="0" maxOccurs="1"/>
  </all>
</complexType>

```

```

        <element name="event-listeners" type="rescfg:event-listeners" minOccurs="0"
maxOccurs="1"/>
        <element name="XLinkConfig" type="rescfg:xlink-config" minOccurs="0" maxOccurs="1"/>
        <element name="XIncludeConfig" type="rescfg:xinclude-config" minOccurs="0"
maxOccurs="1"/>
        <element name="SectionConfig" type="rescfg:section-config" minOccurs="0"
maxOccurs="1"/>
        <element name="ContentFormat" type="rescfg:content-format" minOccurs="0"
maxOccurs="1"/>
        <!-- application data -->
        <element name="applicationData" minOccurs="0" maxOccurs="1">
            <complexType>
                <sequence>
                    <any namespace="##other" maxOccurs="unbounded" processContents="lax"/>
                </sequence>
            </complexType>
        </element>
    </all>
    <attribute name="enable" type="boolean" xdb:baseProp="true" default="true"/>
    <attribute name="copy-on-inconsistent-update" type="boolean" use="optional"/>
</complexType>
<element name="ResConfig" type="rescfg:ResConfig" xdb:defaultTable="XDB$RESCONFIG"/>
</schema>

```

acl.xsd: XML Schema for ACLs

The Oracle Database-supplied XML schema used to represent access control lists (ACLs) is presented.

acl.xsd

```

<schema xmlns="http://www.w3.org/2001/XMLSchema"
        targetNamespace="http://xmlns.oracle.com/xdb/acl.xsd" version="1.0"
        xmlns:xdb="http://xmlns.oracle.com/xdb"
        xmlns:xdbacl="http://xmlns.oracle.com/xdb/acl.xsd"
        elementFormDefault="qualified">
    <annotation>
        <documentation>
            This XML schema describes the structure of XDB ACL documents.

            Note : The "systemPrivileges" element below lists all supported
                    system privileges and their aggregations.
                    See dav.xsd for description of DAV privileges
            Note : The elements and attributes marked "hidden" are for
                    internal use only.
        </documentation>
    </annotation>
    <appinfo>
        <xdb:systemPrivileges>
            <xdbacl:all>
                <xdbacl:read-properties/>
                <xdbacl:read-contents/>
                <xdbacl:read-acl/>
                <xdbacl:update/>
                <xdbacl:link/>
                <xdbacl:unlink/>
            </xdbacl:all>
        </xdb:systemPrivileges>
    </appinfo>

```

```

        <xdbacl:unlink-from/>
        <xdbacl:write-acl-ref/>
        <xdbacl:update-acl/>
        <xdbacl:link-to/>
        <xdbacl:resolve/>
        <xdbacl:write-config/>
    </xdbacl:all>
</xdb:systemPrivileges>
</appinfo>
</annotation>
<!-- privilegeNameType (this is an emptycontent type) -->
<complexType name = "privilegeNameType"/>
<!-- privilegeName element
    All system and user privileges are in the substitutionGroup
    of this element.
-->
<element name = "privilegeName" type="xdbacl:privilegeNameType"
    xdb:defaultTable=""/>
<!-- all system privileges in the XDB ACL namespace -->
<element name = "read-properties" type="xdbacl:privilegeNameType"
    substitutionGroup="xdbacl:privilegeName" xdb:defaultTable=""/>
<element name = "read-contents" type="xdbacl:privilegeNameType"
    substitutionGroup="xdbacl:privilegeName" xdb:defaultTable=""/>
<element name = "read-acl" type="xdbacl:privilegeNameType"
    substitutionGroup="xdbacl:privilegeName" xdb:defaultTable=""/>
<element name = "update" type="xdbacl:privilegeNameType"
    substitutionGroup="xdbacl:privilegeName" xdb:defaultTable=""/>
<element name = "link" type="xdbacl:privilegeNameType"
    substitutionGroup="xdbacl:privilegeName" xdb:defaultTable=""/>
<element name = "unlink" type="xdbacl:privilegeNameType"
    substitutionGroup="xdbacl:privilegeName" xdb:defaultTable=""/>
<element name = "unlink-from" type="xdbacl:privilegeNameType"
    substitutionGroup="xdbacl:privilegeName" xdb:defaultTable=""/>
<element name = "write-acl-ref" type="xdbacl:privilegeNameType"
    substitutionGroup="xdbacl:privilegeName" xdb:defaultTable=""/>
<element name = "update-acl" type="xdbacl:privilegeNameType"
    substitutionGroup="xdbacl:privilegeName" xdb:defaultTable=""/>
<element name = "link-to" type="xdbacl:privilegeNameType"
    substitutionGroup="xdbacl:privilegeName" xdb:defaultTable=""/>
<element name = "resolve" type="xdbacl:privilegeNameType"
    substitutionGroup="xdbacl:privilegeName" xdb:defaultTable=""/>
<element name = "all" type="xdbacl:privilegeNameType"
    substitutionGroup="xdbacl:privilegeName" xdb:defaultTable=""/>
<!-- privilege element -->
<element name = "privilege" xdb:defaultTable="">
    <complexType>
        <sequence>
            <any maxOccurs="unbounded" processContents="lax"/>
        </sequence>
    </complexType>
</element>
<!-- ace element -->
<element name = "ace" xdb:defaultTable="">
    <complexType>
        <sequence>
            <element name = "grant" type = "boolean"/>

```



```

<choice>
  <element name="invert" xdb:transient="generated">
    <complexType>
      <sequence>
        <element name="principal" type="string"
          xdb:transient="generated" />
      </sequence>
    </complexType>
  </element>
  <element name="principal" type="string" xdb:transient="generated"/>
</choice>
<element ref="xdbacl:privilege" minOccurs="1"/>
<!-- "any" contain all app info for an ACE e.g.reason for creation -->
<any minOccurs="0" maxOccurs="unbounded" namespace="##other"
  processContents="lax"/>
<!-- HIDDEN ELEMENTS -->
<choice minOccurs="0">
  <element name = "principalID" type = "hexBinary"
    xdb:baseProp="true" xdb:hidden="true"/>
  <element name = "principalString" type = "string"
    xdb:baseProp="true" xdb:hidden="true"/>
</choice>
<element name = "flags" type = "unsignedInt" minOccurs="0"
  xdb:baseProp="true" xdb:hidden="true"/>
</sequence>
<attribute name = "collection" type = "boolean"
  xdb:transient="generated" use="optional"/>
<attribute name = "principalFormat"
  xdb:transient="generated" use="optional">
  <simpleType>
    <restriction base="string">
      <enumeration value="ShortName"/>
      <enumeration value="DistinguishedName"/>
      <enumeration value="GUID"/>
      <enumeration value="XSName"/>
      <enumeration value="ApplicationName"/>
    </restriction>
  </simpleType>
</attribute>
<attribute name = "start_date" type = "dateTime" use = "optional"/>
<attribute name = "end_date" type = "dateTime" use = "optional"/>
</complexType>
</element>
<!-- acl element -->
<complexType name="inheritanceType">
  <attribute name="type" type="string" use="required"/>
  <attribute name="href" type="string" use="required"/>
</complexType>
<complexType name="aclType">
  <sequence>
    <element name = "schemaURL" type = "string" minOccurs="0"
      xdb:transient="generated"/>
    <element name = "elementName" type = "string" minOccurs="0"
      xdb:transient="generated"/>
    <element name = "security-class" type = "QName" minOccurs="0"/>
    <choice minOccurs="0">

```

```

        <element name="extends-from" type="xdbacl:inheritanceType"/>
        <element name="constrained-with" type="xdbacl:inheritanceType"/>
    </choice>
    <element ref = "xdbacl:ace" minOccurs="0" maxOccurs = "unbounded"/>
    <!-- this "any" contains all application specific info for an ACL,
         e.g., reason for creation -->
    <any minOccurs="0" maxOccurs="unbounded" namespace="##other"
        processContents="lax"/>
    <!-- HIDDEN ELEMENTS -->
    <element name = "schemaOID" type = "hexBinary" minOccurs="0"
        xdb:baseProp="true" xdb:hidden="true"/>
    <element name = "elementNum" type = "unsignedInt" minOccurs="0"
        xdb:baseProp="true" xdb:hidden="true"/>
</sequence>
<attribute name = "shared" type = "boolean" default="true"/>
<attribute name = "description" type = "string"/>
</complexType>
<complexType name="rule-based-acl">
    <complexContent>
        <extension base="xdbacl:aclType">
            <sequence>
                <element name = "param" minOccurs="0" maxOccurs="unbounded">
                    <complexType>
                        <simpleContent>
                            <extension base="string">
                                <attribute name = "name" type = "string" use = "required"/>
                            </extension>
                        </simpleContent>
                    </complexType>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<element name = "acl" type="xdbacl:aclType" xdb:defaultTable = "XDB$ACL"/>
<element name = "write-config" type="xdbacl:privilegeNameType"
    substitutionGroup="xdbacl:privilegeName" xdb:defaultTable="" />
</schema>

```

xdbconfig.xsd: XML Schema for Configuring Oracle XML DB

A full listing is presented of file `xdbconfig.xsd`, which contains the XML schema used to configure Oracle XML DB.



Note:

The value of attribute `value` of element `pattern` has been split here for documentation purposes. In reality, the value is not split (no newline characters), but is one long string.

xdbcconfig.xsd

```

<schema targetNamespace="http://xmlns.oracle.com/xdb/xdbcconfig.xsd"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xdbc="http://xmlns.oracle.com/xdb/xdbcconfig.xsd"
  xmlns:xdb="http://xmlns.oracle.com/xdb"
  version="1.0" elementFormDefault="qualified">
  <element name="xdbcconfig" xdb:defaultTable="XDB$CONFIG">
    <complexType>
      <sequence>

        <!-- predefined XDB properties - these should NOT be changed -->
        <element name="sysconfig">
          <complexType>
            <sequence>

              <!-- generic XDB properties -->
              <element name="acl-max-age" type="unsignedInt" default="15"/>
              <element name="acl-cache-size" type="unsignedInt" default="32"/>
              <element name="invalid-pathname-chars" type="string" default=""/>
              <element name="case-sensitive" type="boolean" default="true"/>
              <element name="call-timeout" type="unsignedInt" default="300"/>
              <element name="max-link-queue" type="unsignedInt" default="65536"/>
              <element name="max-session-use" type="unsignedInt" default="100"/>
              <element name="persistent-sessions" type="boolean" default="false"/>
              <element name="default-lock-timeout" type="unsignedInt"
                default="3600"/>
              <element name="xdbccore-logfile-path" type="string"
                default="/sys/log/xdblog.xml"/>
              <element name="xdbccore-log-level" type="unsignedInt"
                default="0"/>
              <element name="resource-view-cache-size" type="unsignedInt"
                default="1048576"/>
              <element name="case-sensitive-index-clause" type="string"
                minOccurs="0"/>

              <!-- protocol specific properties -->
              <element name="protocolconfig">
                <complexType>
                  <sequence>

                    <!-- these apply to all protocols -->
                    <element name="common">
                      <complexType>
                        <sequence>
                          <element name="extension-mappings">
                            <complexType>
                              <sequence>
                                <element name="mime-mappings"
                                  type="xdbc:mime-mapping-type"/>
                                <element name="lang-mappings"
                                  type="xdbc:lang-mapping-type"/>
                                <element name="charset-mappings"
                                  type="xdbc:charset-mapping-type"/>
                                <element name="encoding-mappings"

```

```

        type="xdbc:encoding-mapping-type"/>
      <element name="xml-extensions"
        type="xdbc:xml-extension-type"
        minOccurs="0"/>
    </sequence>
  </complexType>
</element>
<element name="session-pool-size" type="unsignedInt"
  default="50"/>
<element name="session-timeout" type="unsignedInt"
  default="6000"/>
</sequence>
</complexType>
</element>

<!-- FTP specific -->
<element name="ftpconfig">
  <complexType>
    <sequence>
      <element name="ftp-port" type="unsignedShort"
        default="2100"/>
      <element name="ftp-listener" type="string"/>
      <element name="ftp-protocol" type="string"/>
      <element name="logfile-path" type="string"
        default="/sys/log/ftplog.xml"/>
      <element name="log-level" type="unsignedInt"
        default="0"/>
      <element name="session-timeout" type="unsignedInt"
        default="6000"/>
      <element name="buffer-size" default="8192">
        <simpleType>
          <restriction base="unsignedInt">
            <minInclusive value="1024"/>
            <maxInclusive value="1048496"/>
          </restriction>
        </simpleType>
      </element>
      <element name="ftp-welcome-message" type="string"
        minOccurs="0" maxOccurs="1"/>
      <element name="host-name" type="string"
        minOccurs="0" maxOccurs="1"/>
    </sequence>
  </complexType>
</element>

<!-- HTTP specific -->
<element name="httpconfig">
  <complexType>
    <sequence>
      <element name="http-port" type="unsignedShort"
        default="8080"/>
      <element name="http-listener" type="string"/>
      <element name="http-protocol" type="string"/>
      <element name="max-http-headers" type="unsignedInt"
        default="64"/>
      <element name="max-header-size" type="unsignedInt"

```

```

        default="4096"/>
<element name="max-request-body" type="unsignedInt"
        default="2000000000" minOccurs="1"/>
<element name="session-timeout" type="unsignedInt"
        default="6000"/>
<element name="server-name" type="string"/>
<element name="logfile-path" type="string"
        default="/sys/log/httplog.xml"/>
<element name="log-level" type="unsignedInt"
        default="0"/>
<element name="servlet-realm" type="string"
        minOccurs="0"/>
<element name="webappconfig">
  <complexType>
    <sequence>
      <element name="welcome-file-list"
        type="xdbc:welcome-file-type"/>
      <element name="error-pages"
        type="xdbc:error-page-type"/>
      <element name="servletconfig"
        type="xdbc:servlet-config-type"/>
    </sequence>
  </complexType>
</element>
<element name="default-url-charset" type="string"
        minOccurs="0"/>
<element name="http2-port" type="unsignedShort"
        minOccurs="0"/>
<element name="http2-protocol" type="string"
        default="tcp" minOccurs="0"/>
<element name="plssql" minOccurs="0">
  <complexType>
    <sequence>
      <element name="log-level"
        type="unsignedInt" minOccurs="0"/>
      <element name="max-parameters"
        type="unsignedInt" minOccurs="0"/>
    </sequence>
  </complexType>
</element>
<element name="allow-repository-anonymous-access"
        minOccurs="0" default="false" type="boolean"/>
<element name="authentication" minOccurs="0"
        maxOccurs="1">
  <complexType>
    <sequence>
      <element name="allow-mechanism" minOccurs="1"
        maxOccurs="unbounded">
        <simpleType>
          <restriction base="string">
            <enumeration value="digest"/>
            <enumeration value="basic" />
            <enumeration value="custom"/>
            <enumeration value="digestMD5"/>
            <enumeration value="digestSHA256" />
            <enumeration value="digestSHA512"/>
          </restriction>
        </simpleType>
      </element>
    </sequence>
  </complexType>
</element>

```

```

        </restriction>
      </simpleType>
    </element>
    <element name="digest-auth" minOccurs="0"
      maxOccurs="1">
      <complexType>
        <sequence>
          <element name="nonce-timeout"
            type="unsignedInt"
            minOccurs="1" maxOccurs="1"
            default="300"/>
        </sequence>
      </complexType>
    </element>
  </sequence>
</complexType>
</element>
<element name="http-host" type="string" minOccurs="0"/>
<element name="http2-host" type="string" minOccurs="0"/>
<element name="custom-authentication"
  type="xdbc:custom-authentication-type"
  minOccurs="0"/>
<element name="realm" type="string" minOccurs="0"/>
<element name="respond-with-server-info" type="boolean"
  default="true" minOccurs="0"/>
<element name="expire" type="xdbc:expire-type"
  minOccurs="0"/>
<element name="white-list" minOccurs="0">
  <complexType>
    <sequence>
      <element name="white-list-pattern" minOccurs="0"
        maxOccurs="unbounded">
        <simpleType>
          <restriction base="string">
            <pattern value="(/[^\s/]+)* (/\s)*?" />
          </restriction>
        </simpleType>
      </element>
    </sequence>
  </complexType>
</element>
<element name="nfsconfig" minOccurs="0">
  <complexType>
    <sequence>
      <element name="nfs-port" type="unsignedShort"
        default="2049"/>
      <element name="nfs-listener" type="string"/>
      <element name="nfs-protocol" type="string"/>
      <element name="logfile-path" type="string"
        default="/sys/log/nfslog.xml"/>
      <element name="log-level" type="unsignedInt"
        default="0"/>
      <element name="nfs-exports"

```

```

        type="xdbc:nfs-exports-type"/>
    </sequence>
</complexType>
</element>
</sequence>
</complexType>
</element>
<element name="schemaLocation-mappings"
    type="xdbc:schemaLocation-mapping-type" minOccurs="0"/>
<element name="xdbc:core-xobmem-bound" type="unsignedInt"
    default="1024" minOccurs="0"/>
<element name="xdbc:core-loadableunit-size" type="unsignedInt"
    default="16" minOccurs="0"/>
<element name="folder-hard-links" type="boolean" default="false"
    minOccurs="0"/>
<element name="non-folder-hard-links" type="boolean" default="true"
    minOccurs="0"/>
<element name="copy-on-inconsistent-update" type="boolean"
    default="false" minOccurs="0"/>
<element name="rollback-on-sync-error" type="boolean"
    default="false" minOccurs="0"/>
<element name="acl-evaluation-method" default="deny-trumps-grant"
    minOccurs="0">
    <simpleType>
        <restriction base="string">
            <enumeration value="deny-trumps-grant"/>
            <enumeration value="ace-order"/>
        </restriction>
    </simpleType>
</element>
<element name="default-workspace" type="string" minOccurs="0"/>
<element name="num_job_queue_processes" type="unsignedInt"
    minOccurs="0"/>
<element name="allow-authentication-trust" type="boolean"
    default="false" minOccurs="0"/>
<element name="custom-authentication-trust"
    type="xdbc:custom-authentication-trust-type"
    minOccurs="0"/>
<element name="default-type-mappings" minOccurs="0">
    <simpleType>
        <restriction base="string">
            <enumeration value="pre-11.2"/>
            <enumeration value="post-11.2"/>
        </restriction>
    </simpleType>
</element>
<element name="localApplicationGroupStore" type="boolean"
    default="true" minOccurs="0"/>
</sequence>
</complexType>
</element>

<!-- users can add any properties they want here -->
<element name="userconfig" minOccurs="0">
    <complexType>
        <sequence>

```

```

        <any maxOccurs="unbounded" namespace="##other"/>
    </sequence>
</complexType>
</element>
</sequence>
</complexType>
</element>
<complexType name="welcome-file-type">
    <sequence>
        <element name="welcome-file" minOccurs="0" maxOccurs="unbounded">
            <simpleType>
                <restriction base="string">
                    <pattern value="[^/]*"/>
                </restriction>
            </simpleType>
        </element>
    </sequence>
</complexType>

<!-- customized error pages -->
<complexType name="error-page-type">
    <sequence>
        <element name="error-page" minOccurs="0" maxOccurs="unbounded">
            <complexType>
                <sequence>
                    <choice>
                        <element name="error-code">
                            <simpleType>
                                <restriction base="positiveInteger">
                                    <minInclusive value="100"/>
                                    <maxInclusive value="999"/>
                                </restriction>
                            </simpleType>
                        </element>

                        <!-- Fully qualified classname of a Java exception type -->
                        <element name="exception-type" type="string"/>
                        <element name="OracleError">
                            <complexType>
                                <sequence>
                                    <element name="facility" type="string" default="ORA"/>
                                    <element name="errnum" type="unsignedInt"/>
                                </sequence>
                            </complexType>
                        </element>
                    </choice>
                    <element name="location" type="anyURI"/>
                </sequence>
            </complexType>
        </element>
    </sequence>
</complexType>

<!-- parameter for a servlet: name, value pair and a description -->
<complexType name="param">
    <sequence>

```



```

    <element name="param-name" type="string"/>
    <element name="param-value" type="string"/>
    <element name="description" type="string"/>
  </sequence>
</complexType>
<complexType name="servlet-config-type">
  <sequence>
    <element name="servlet-mappings">
      <complexType>
        <sequence>
          <element name="servlet-mapping" minOccurs="0"
            maxOccurs="unbounded">
            <complexType>
              <sequence>
                <element name="servlet-pattern" type="string"/>
                <element name="servlet-name" type="string"/>
              </sequence>
            </complexType>
          </element>
        </sequence>
      </complexType>
    </element>
    <element name="servlet-list">
      <complexType>
        <sequence>
          <element name="servlet" minOccurs="0" maxOccurs="unbounded">
            <complexType>
              <sequence>
                <element name="servlet-name" type="string"/>
                <element name="servlet-language">
                  <simpleType>
                    <restriction base="string">
                      <enumeration value="C"/>
                      <enumeration value="Java"/>
                      <enumeration value="PL/SQL"/>
                    </restriction>
                  </simpleType>
                </element>
                <element name="icon" type="string" minOccurs="0"/>
                <element name="display-name" type="string"/>
                <element name="description" type="string" minOccurs="0"/>
                <choice>
                  <element name="servlet-class" type="string" minOccurs="0"/>
                  <element name="jsp-file" type="string" minOccurs="0"/>
                  <element name="plssql" type="xdbc:plssql-servlet-config"
                    minOccurs="0"/>
                </choice>
                <element name="servlet-schema" type="string" minOccurs="0"/>
                <element name="init-param" minOccurs="0"
                  maxOccurs="unbounded" type="xdbc:param"/>
                <element name="load-on-startup" type="string" minOccurs="0"/>
                <element name="security-role-ref" minOccurs="0"
                  maxOccurs="unbounded">
                  <complexType>
                    <sequence>
                      <element name="description" type="string" minOccurs="0"/>

```

```

        <element name="role-name" type="string"/>
        <element name="role-link" type="string"/>
    </sequence>
</complexType>
</element>
<!-- session-state-cache-param captures all the parameters
      of the session state cache. expiration-timeout is
      specified in centi-seconds. -->
<element name="session-state-cache-param" minOccurs="0">
    <complexType>
        <sequence>
            <element name="cache-size" type="unsignedInt"/>
            <element name="expiration-timeout" type="unsignedInt"/>
        </sequence>
    </complexType>
</element>
</sequence>
</complexType>
</element>
</sequence>
</complexType>
</element>
</sequence>
</complexType>
</element>
</sequence>
</complexType>
<complexType name="lang-mapping-type">
    <sequence>
        <element name="lang-mapping" minOccurs="0" maxOccurs="unbounded">
            <complexType>
                <sequence>
                    <element name="extension" type="xdbc:exttype"/>
                    <element name="lang" type="string"/>
                </sequence>
            </complexType>
        </element>
    </sequence>
</complexType>
<complexType name="charset-mapping-type">
    <sequence>
        <element name="charset-mapping" minOccurs="0" maxOccurs="unbounded">
            <complexType>
                <sequence>
                    <element name="extension" type="xdbc:exttype"/>
                    <element name="charset" type="string"/>
                </sequence>
            </complexType>
        </element>
    </sequence>
</complexType>
<complexType name="encoding-mapping-type">
    <sequence>
        <element name="encoding-mapping" minOccurs="0" maxOccurs="unbounded">
            <complexType>
                <sequence>
                    <element name="extension" type="xdbc:exttype"/>
                    <element name="encoding" type="string"/>
                </sequence>
            </complexType>
        </element>
    </sequence>
</complexType>

```

```

        </complexType>
      </element>
    </sequence>
  </complexType>
  <complexType name="mime-mapping-type">
    <sequence>
      <element name="mime-mapping" minOccurs="0" maxOccurs="unbounded">
        <complexType>
          <sequence>
            <element name="extension" type="xdbc:exttype"/>
            <element name="mime-type" type="string"/>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>
  <complexType name="xml-extension-type">
    <sequence>
      <element name="extension" type="xdbc:exttype"
        minOccurs="0" maxOccurs="unbounded">
      </element>
    </sequence>
  </complexType>
  <complexType name="schemaLocation-mapping-type">
    <sequence>
      <element name="schemaLocation-mapping"
        minOccurs="0" maxOccurs="unbounded">
        <complexType>
          <sequence>
            <element name="namespace" type="string"/>
            <element name="element" type="string"/>
            <element name="schemaURL" type="string"/>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>
  <complexType name="plsql-servlet-config">
    <sequence>
      <element name="database-username" type="string" minOccurs="0"/>
      <element name="authentication-mode" minOccurs="0">
        <simpleType>
          <restriction base="string">
            <enumeration value="Basic"/>
            <enumeration value="SingleSignOn"/>
            <enumeration value="GlobalOwa"/>
            <enumeration value="CustomOwa"/>
            <enumeration value="PerPackageOwa"/>
          </restriction>
        </simpleType>
      </element>
      <element name="session-cookie-name" type="string" minOccurs="0"/>
      <element name="session-state-management" minOccurs="0">
        <simpleType>
          <restriction base="string">
            <enumeration value="StatelessWithResetPackageState"/>

```

```

        <enumeration value="StatelessWithFastResetPackageState"/>
        <enumeration value="StatelessWithPreservePackageState"/>
    </restriction>
</simpleType>
</element>
<element name="max-requests-per-session" type="unsignedInt" minOccurs="0"/>
<element name="default-page" type="string" minOccurs="0"/>
<element name="document-table-name" type="string" minOccurs="0"/>
<element name="document-path" type="string" minOccurs="0"/>
<element name="document-procedure" type="string" minOccurs="0"/>
<element name="upload-as-long-raw" type="string" minOccurs="0"
    maxOccurs="unbounded"/>
<element name="path-alias" type="string" minOccurs="0"/>
<element name="path-alias-procedure" type="string" minOccurs="0"/>
<element name="exclusion-list" type="string" minOccurs="0"
    maxOccurs="unbounded"/>
<element name="cgi-environment-list" type="string" minOccurs="0"
    maxOccurs="unbounded"/>
<element name="compatibility-mode" type="unsignedInt" minOccurs="0"/>
<element name="nls-language" type="string" minOccurs="0"/>
<element name="fetch-buffer-size" type="unsignedInt" minOccurs="0"/>
<element name="error-style" minOccurs="0">
    <simpleType>
        <restriction base="string">
            <enumeration value="ApacheStyle"/>
            <enumeration value="ModplsqlStyle"/>
            <enumeration value="DebugStyle"/>
        </restriction>
    </simpleType>
</element>
<element name="transfer-mode" minOccurs="0">
    <simpleType>
        <restriction base="string">
            <enumeration value="Char"/>
            <enumeration value="Raw"/>
        </restriction>
    </simpleType>
</element>
<element name="before-procedure" type="string" minOccurs="0"/>
<element name="after-procedure" type="string" minOccurs="0"/>
<element name="bind-bucket-lengths" type="unsignedInt" minOccurs="0"
    maxOccurs="unbounded"/>
<element name="bind-bucket-widths" type="unsignedInt" minOccurs="0"
    maxOccurs="unbounded"/>
<element name="always-describe-procedure" minOccurs="0">
    <simpleType>
        <restriction base="string">
            <enumeration value="On"/>
            <enumeration value="Off"/>
        </restriction>
    </simpleType>
</element>
<element name="info-logging" minOccurs="0">
    <simpleType>
        <restriction base="string">
            <enumeration value="InfoDebug"/>

```

```

        </restriction>
      </simpleType>
    </element>
    <element name="owa-debug-enable" minOccurs="0">
      <simpleType>
        <restriction base="string">
          <enumeration value="On"/>
          <enumeration value="Off"/>
        </restriction>
      </simpleType>
    </element>
    <element name="request-validation-function" type="string" minOccurs="0"/>
    <element name="input-filter-enable" minOccurs="0">
      <simpleType>
        <restriction base="string">
          <enumeration value="On"/>
          <enumeration value="Off"/>
          <enumeration value="SecurityOn"/>
          <enumeration value="SecurityOff"/>
        </restriction>
      </simpleType>
    </element>
    <element name="database-edition" type="string" minOccurs="0"/>
  </sequence>
</complexType>
<simpleType name="exttype">
  <restriction base="string">
    <pattern value="^[^*\.\.]*"/>
  </restriction>
</simpleType>
<simpleType name="ipaddress">
  <restriction base="string">
    <maxLength value="40" />
  </restriction>
</simpleType>
<complexType name="nfs-exports-type">
  <sequence>
    <element name="nfs-export" minOccurs="1" maxOccurs="unbounded">
      <complexType>
        <sequence>
          <element name="nfs-clientgroup">
            <complexType>
              <sequence>
                <element name="nfs-client" minOccurs="1" maxOccurs="unbounded">
                  <complexType>
                    <sequence>
                      <choice>
                        <element name="nfs-client-subnet"
                          type="xdbc:ipaddress"/>
                        <element name="nfs-client-dnsname" type="string"/>
                        <element name="nfs-client-address"
                          type="xdbc:ipaddress"/>
                      </choice>
                      <element name="nfs-client-netmask"
                        type="xdbc:ipaddress"/>
                    </sequence>
                  </complexType>
                </element>
              </sequence>
            </complexType>
          </element>
        </sequence>
      </complexType>
    </element>
  </sequence>

```

```

        </complexType>
      </element>
    </sequence>
  </complexType>
</element>
<element name="nfs-export-paths">
  <complexType>
    <sequence>
      <element name="nfs-export-path" minOccurs="1"
        maxOccurs="unbounded">
        <complexType>
          <sequence>
            <element name="path" type="string"/>
            <element name="mode">
              <simpleType>
                <restriction base="string">
                  <enumeration value="read-write"/>
                  <enumeration value="read-only"/>
                </restriction>
              </simpleType>
            </element>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>
</element>
</sequence>
</complexType>
</element>
</sequence>
</complexType>
</element>
</sequence>
</complexType>
<complexType name="custom-authentication-type">
  <sequence>
    <element name="custom-authentication-mappings">
      <complexType>
        <sequence>
          <element name="custom-authentication-mapping" minOccurs="0"
            maxOccurs="unbounded">
            <complexType>
              <sequence>
                <element name="authentication-pattern" type="string"/>
                <element name="authentication-name" type="string"/>
                <element name="authentication-trust-name" type="string"
                  minOccurs="0"/>
                <element name="user-prefix" type="string" minOccurs="0"/>
                <element name="on-deny" minOccurs="0">
                  <simpleType><restriction base="string">
                    <enumeration value="next-custom"/>
                    <enumeration value="basic"/>
                  </restriction></simpleType>
                </element>
              </sequence>
            </complexType>
          </element>
        </sequence>
      </complexType>
    </element>
  </sequence>
</complexType>

```

```

    </complexType>
  </element>
  <element name="custom-authentication-list">
    <complexType>
      <sequence>
        <element name="authentication" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <sequence>
              <element name="authentication-name" type="string"/>
              <element name="authentication-description" type="string"
                minOccurs="0"/>
              <element name="authentication-implement-schema" type="string"/>
              <element name="authentication-implement-method" type="string"/>
              <element name="authentication-implement-language">
                <simpleType>
                  <restriction base="string">
                    <enumeration value="PL/SQL"/>
                  </restriction>
                </simpleType>
              </element>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </complexType>
  </element>
  <element name="custom-authentication-trust"
    type="xdbc:custom-authentication-trust-type" minOccurs="0"/>
</sequence>
</complexType>
<complexType name="custom-authentication-trust-type">
  <sequence>
    <element name="trust-scheme" minOccurs="0" maxOccurs="unbounded">
      <complexType>
        <sequence>
          <element name="trust-scheme-name" type="string"/>
          <element name="requireParsingSchema" type="boolean" default="true"
            minOccurs="0"/>
          <element name="allowRegistration" type="boolean" default="true"
            minOccurs="0"/>
          <element name="trust-scheme-description" type="string" minOccurs="0"/>
          <element name="trusted-session-user" type="string" minOccurs="1"
            maxOccurs="unbounded"/>
          <element name="trusted-parsing-schema" type="string" minOccurs="0"
            maxOccurs="unbounded"/>
        </sequence>
      </complexType>
    </element>
  </sequence>
</complexType>
<complexType name="expire-type">
  <sequence>
    <element name="expire-mapping" minOccurs="0" maxOccurs="unbounded">
      <complexType>
        <sequence>
          <element name="expire-pattern" type="string"/>

```

```

    <element name="expire-default">
      <simpleType>
        <restriction base="string">
          <pattern value=
"(now|modification) (\s(plus))?( \s(([1]\s(year)) |
([0-9]*\s(years))))? (\s(([1]\s(month)) | ([0-9]*\s(months))))? (\s(([1]\s(week)) |
([0-9]*\s(weeks))))? (\s(([1]\s(day)) | ([0-9]*\s(days))))? (\s(([1]\s(hour)) |
([0-9]*\s(hours))))? (\s(([1]\s(minute)) |
([0-9]*\s(minutes))))? (\s(([1]\s(second)) | ([0-9]*\s(seconds))))?"/>1
        </restriction>
      </simpleType>
    </element>
  </sequence>
</complexType>
</element>
</sequence>
</complexType>
</schema>

```

xdiff.xsd: XML Schema for Comparing Schemas for In-Place Evolution

A full listing is presented of `xdiff.xsd`, the Oracle XML DB-supplied XML schema to which the document specified as the `diffXML` parameter to procedure `DBMS_XMLSCHEMA.inPlaceEvolve` must conform.

xdiff.xsd

```

<schema targetNamespace="http://xmlns.oracle.com/xdb/xdiff.xsd"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xd="http://xmlns.oracle.com/xdb/xdiff.xsd"
  version="1.0" elementFormDefault="qualified"
  attributeFormDefault="qualified">
  <annotation>
    <documentation xml:lang="en">
      Defines the structure of XML documents that capture the difference
      between two XML documents. Changes that are not supported by Oracle
      XmlDiff may not be expressible in this schema.
    </documentation>
  </annotation>

```

'oracle-xmldiff' PI:

We use 'oracle-xmldiff' PI to describe certain aspects of the diff. This should be the first element of top level xdiff element.

operations-in-docorder:

Can be either 'true' or 'false'.

If true, the operations in the diff document refer to the elements of the input doc in the same order as document order. Output

of

global algorithm meets this requirement while local does not.

¹ The value of attribute value has been split here for documentation purposes. In reality, the value is one long string, with no line breaks.

output-model: output models for representing the diff. Can be either 'Snapshot' or 'Current'.

Snapshot model:

Each operation uses Xpaths as if no operations have been applied to the input document. (like UNIX diff) Default and works for both XmlDiff and XmlPatch. For XmlPatch to handle this model, "operations-in-docorder" must be true and the Xpaths must be simple. (see XmlDiff C API documentation).

Current model :

Each operation uses Xpaths as if all operations till the previous one have been applied to the input document. Not implemented for XmlDiff. Works with XmlPatch.

<!-- Example:

```

    <?oracle-xmldiff operations-in-docorder="true" output-model=
    "snapshot" diff-algorithm="global"?>
  -->
  </documentation>
</annotation>
<!-- Enumerate the supported node types -->
<simpleType name="xdiff-nodetype">
  <restriction base="string">
    <enumeration value="element"/>
    <enumeration value="attribute"/>
    <enumeration value="text"/>
    <enumeration value="cdata"/>
    <enumeration value="entity-reference"/>
    <enumeration value="entity"/>
    <enumeration value="processing-instruction"/>
    <enumeration value="notation"/>
    <enumeration value="comment"/>
  </restriction>
</simpleType>

<element name="xdiff">
  <complexType>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element name="append-node">
        <complexType>
          <sequence>
            <element name="content" type="anyType"/>
          </sequence>
          <attribute name="node-type" type="xd:xdiff-
nodetype"/>
          <attribute name="xpath" type="string"/>
          <attribute name="parent-xpath" type="string"/>
          <attribute name="attr-local" type="string"/>
          <attribute name="attr-uri" type="string"/>
        </complexType>
      </element>

      <element name="insert-node-before">
        <complexType>

```

```

        <sequence>
            <element name="content" type="anyType"/>
        </sequence>
        <attribute name="xpath" type="string"/>
        <attribute name="node-type" type="xd:xdiff-nodetype"/>

    </complexType>
</element>

<element name="delete-node">
    <complexType>
        <attribute name="node-type" type="xd:xdiff-nodetype"/>
        <attribute name="xpath" type="string"/>
        <attribute name="parent-xpath" type="string"/>
        <attribute name="attr-local" type="string"/>
        <attribute name="attr-uri" type="string"/>
    </complexType>
</element>
<element name="update-node">
    <complexType>
        <sequence>
            <element name="content" type="anyType"/>
        </sequence>
        <attribute name="node-type" type="xd:xdiff-
nodetype"/>
        <attribute name="parent-xpath" type="string"/>
        <attribute name="xpath" type="string"/>
        <attribute name="attr-local" type="string"/>
        <attribute name="attr-uri" type="string"/>
    </complexType>
</element>
<element name="rename-node">
    <complexType>
        <sequence>
            <element name="content" type="anyType"/>
        </sequence>
        <attribute name="xpath" type="string"/>
        <attribute name="node-type" type="xd:xdiff-
nodetype"/>
    </complexType>
</element>
</choice>
    <attribute name="xdiff-version" type="string"/>
</complexType>
</element>
</schema>

```

Purchase-Order XML Schemas

The full listings of purchase-order XML schemas that are used in various examples are presented.

[Example A-1](#) shows an unannotated purchase-order XML schema.

Example A-1 Unannotated Purchase-Order XML Schema

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" version="1.0">
  <xs:element name="PurchaseOrder" type="PurchaseOrderType"/>
  <xs:complexType name="PurchaseOrderType">
    <xs:sequence>
      <xs:element name="Reference" type="ReferenceType"/>
      <xs:element name="Actions" type="ActionsType"/>
      <xs:element name="Reject" type="RejectionType" minOccurs="0"/>
      <xs:element name="Requestor" type="RequestorType"/>
      <xs:element name="User" type="UserType"/>
      <xs:element name="CostCenter" type="CostCenterType"/>
      <xs:element name="ShippingInstructions" type="ShippingInstructionsType"/>
      <xs:element name="SpecialInstructions" type="SpecialInstructionsType"/>
      <xs:element name="LineItems" type="LineItemsType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="LineItemsType">
    <xs:sequence>
      <xs:element name="LineItem" type="LineItemType" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="LineItemType">
    <xs:sequence>
      <xs:element name="Description" type="DescriptionType"/>
      <xs:element name="Part" type="PartType"/>
    </xs:sequence>
    <xs:attribute name="ItemNumber" type="xs:integer"/>
  </xs:complexType>
  <xs:complexType name="PartType">
    <xs:attribute name="Id">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:minLength value="10"/>
          <xs:maxLength value="14"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="Quantity" type="moneyType"/>
    <xs:attribute name="UnitPrice" type="quantityType"/>
  </xs:complexType>
  <xs:simpleType name="ReferenceType">
    <xs:restriction base="xs:string">
      <xs:minLength value="18"/>
      <xs:maxLength value="30"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="ActionsType">
    <xs:sequence>
      <xs:element name="Action" maxOccurs="4">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="User" type="UserType"/>
            <xs:element name="Date" type="DateType" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

```
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="RejectionType">
  <xs:all>
    <xs:element name="User" type="UserType" minOccurs="0"/>
    <xs:element name="Date" type="DateType" minOccurs="0"/>
    <xs:element name="Comments" type="CommentsType" minOccurs="0"/>
  </xs:all>
</xs:complexType>
<xs:complexType name="ShippingInstructionsType">
  <xs:sequence>
    <xs:element name="name" type="NameType" minOccurs="0"/>
    <xs:element name="address" type="AddressType" minOccurs="0"/>
    <xs:element name="telephone" type="TelephoneType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="moneyType">
  <xs:restriction base="xs:decimal">
    <xs:fractionDigits value="2"/>
    <xs:totalDigits value="12"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="quantityType">
  <xs:restriction base="xs:decimal">
    <xs:fractionDigits value="4"/>
    <xs:totalDigits value="8"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="UserType">
  <xs:restriction base="xs:string">
    <xs:minLength value="0"/>
    <xs:maxLength value="10"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="RequestorType">
  <xs:restriction base="xs:string">
    <xs:minLength value="0"/>
    <xs:maxLength value="128"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="CostCenterType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="4"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="VendorType">
  <xs:restriction base="xs:string">
    <xs:minLength value="0"/>
    <xs:maxLength value="20"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="PurchaseOrderNumberType">
  <xs:restriction base="xs:integer"/>
</xs:simpleType>
```

```

<xs:simpleType name="SpecialInstructionsType">
  <xs:restriction base="xs:string">
    <xs:minLength value="0"/>
    <xs:maxLength value="2048"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="NameType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="20"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="AddressType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="256"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="TelephoneType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="24"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="DateType">
  <xs:restriction base="xs:date"/>
</xs:simpleType>
<xs:simpleType name="CommentsType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="2048"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="DescriptionType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="256"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

Example A-3 represents a modified version of [Example A-2](#). The modification is used in [XML Schema Evolution](#) to illustrate XML schema evolution. [Example A-2](#) is the complete listing of the annotated XML schema used in examples of [XML Schema Storage and Query: Basic](#).

Example A-2 Annotated Purchase-Order XML Schema

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xdb="http://xmlns.oracle.com/xdb"
  version="1.0">
  <xs:element name="PurchaseOrder" type="PurchaseOrderType" xdb:defaultTable="PURCHASEORDER"/>
  <xs:complexType name="PurchaseOrderType" xdb:SQLType="PURCHASEORDER_T">
    <xs:sequence>
      <xs:element name="Reference" type="ReferenceType" minOccurs="1" xdb:SQLName="REFERENCE"/>
      <xs:element name="Actions" type="ActionsType" xdb:SQLName="ACTIONS"/>
      <xs:element name="Reject" type="RejectionType" minOccurs="0" xdb:SQLName="REJECTION"/>
      <xs:element name="Requestor" type="RequestorType" xdb:SQLName="REQUESTOR"/>
      <xs:element name="User" type="UserType" minOccurs="1" xdb:SQLName="USERID"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

```

    <xs:element name="CostCenter" type="CostCenterType" xdb:SQLName="COST_CENTER"/>
    <xs:element name="ShippingInstructions" type="ShippingInstructionsType"
      xdb:SQLName="SHIPPING_INSTRUCTIONS"/>
    <xs:element name="SpecialInstructions" type="SpecialInstructionsType"
      xdb:SQLName="SPECIAL_INSTRUCTIONS"/>
    <xs:element name="LineItems" type="LineItemsType" xdb:SQLName="LINEITEMS"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="LineItemsType" xdb:SQLType="LINEITEMS_T">
  <xs:sequence>
    <xs:element name="LineItem" type="LineItemType" maxOccurs="unbounded"
      xdb:SQLName="LINEITEM" xdb:SQLCollType="LINEITEM_V"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="LineItemType" xdb:SQLType="LINEITEM_T">
  <xs:sequence>
    <xs:element name="Description" type="DescriptionType"
      xdb:SQLName="DESCRIPTION"/>
    <xs:element name="Part" type="PartType" xdb:SQLName="PART"/>
  </xs:sequence>
  <xs:attribute name="ItemNumber" type="xs:integer" xdb:SQLName="ITEMNUMBER"
    xdb:SQLType="NUMBER"/>
</xs:complexType>
<xs:complexType name="PartType" xdb:SQLType="PART_T">
  <xs:attribute name="Id" xdb:SQLName="PART_NUMBER" xdb:SQLType="VARCHAR2">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:minLength value="10"/>
        <xs:maxLength value="14"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="Quantity" type="moneyType" xdb:SQLName="QUANTITY"/>
  <xs:attribute name="UnitPrice" type="quantityType" xdb:SQLName="UNITPRICE"/>
</xs:complexType>
<xs:simpleType name="ReferenceType">
  <xs:restriction base="xs:string">
    <xs:minLength value="18"/>
    <xs:maxLength value="30"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="ActionsType" xdb:SQLType="ACTIONS_T">
  <xs:sequence>
    <xs:element name="Action" maxOccurs="4" xdb:SQLName="ACTION" xdb:SQLCollType="ACTION_V">
      <xs:complexType xdb:SQLType="action_t">
        <xs:sequence>
          <xs:element name="User" type="UserType" xdb:SQLName="ACTIONED_BY"/>
          <xs:element name="Date" type="DateType" minOccurs="0" xdb:SQLName="DATE_ACTIONED"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="RejectionType" xdb:SQLType="REJECTION_T">
  <xs:all>
    <xs:element name="User" type="UserType" minOccurs="0" xdb:SQLName="REJECTED_BY"/>
    <xs:element name="Date" type="DateType" minOccurs="0" xdb:SQLName="DATE_REJECTED"/>
    <xs:element name="Comments" type="CommentsType" minOccurs="0" xdb:SQLName="REASON_REJECTED"/>
  </xs:all>
</xs:complexType>
<xs:complexType name="ShippingInstructionsType" xdb:SQLType="SHIPPING_INSTRUCTIONS_T">
  <xs:sequence>
    <xs:element name="name" type="NameType" minOccurs="0" xdb:SQLName="SHIP_TO_NAME"/>
    <xs:element name="address" type="AddressType" minOccurs="0" xdb:SQLName="SHIP_TO_ADDRESS"/>
    <xs:element name="telephone" type="TelephoneType" minOccurs="0" xdb:SQLName="SHIP_TO_PHONE"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="moneyType">
  <xs:restriction base="xs:decimal">
    <xs:fractionDigits value="2"/>
  </xs:restriction>
</xs:simpleType>

```

```
<xs:totalDigits value="12"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="quantityType">
  <xs:restriction base="xs:decimal">
    <xs:fractionDigits value="4"/>
    <xs:totalDigits value="8"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="UserType">
  <xs:restriction base="xs:string">
    <xs:minLength value="0"/>
    <xs:maxLength value="10"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="RequestorType">
  <xs:restriction base="xs:string">
    <xs:minLength value="0"/>
    <xs:maxLength value="128"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="CostCenterType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="4"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="VendorType">
  <xs:restriction base="xs:string">
    <xs:minLength value="0"/>
    <xs:maxLength value="20"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="PurchaseOrderNumberType">
  <xs:restriction base="xs:integer"/>
</xs:simpleType>
<xs:simpleType name="SpecialInstructionsType">
  <xs:restriction base="xs:string">
    <xs:minLength value="0"/>
    <xs:maxLength value="2048"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="NameType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="20"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="AddressType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="256"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="TelephoneType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="24"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="DateType">
  <xs:restriction base="xs:date"/>
</xs:simpleType>
<xs:simpleType name="CommentsType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="2048"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="DescriptionType">
```

```

    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
      <xs:maxLength value="256"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

Example A-3 is the complete listing of the revised annotated XML schema presented in **Example 20-1**. Text that is in **bold face** is additional or significantly different from that in the schema of **Example A-2**.

Example A-3 Revised Annotated Purchase-Order XML Schema

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xdb="http://xmlns.oracle.com/xdb"
  version="1.0">
  <xs:element
    name="PurchaseOrder" type="PurchaseOrderType"
    xdb:defaultTable="PURCHASEORDER"
    xdb:columnProps=
      "CONSTRAINT purchaseorder_pkey PRIMARY KEY (XMLDATA.reference),
      CONSTRAINT valid_email_address FOREIGN KEY (XMLDATA.userid)
      REFERENCES hr.employees (EMAIL) "
    xdb:tableProps=
      "VARRAY XMLDATA.ACTIONS.ACTION STORE AS TABLE ACTION_TABLE
      ((CONSTRAINT action_pkey PRIMARY KEY (NESTED_TABLE_ID, SYS_NC_ARRAY_INDEX$)))
      VARRAY XMLDATA.LINEITEMS.LINEITEM STORE AS TABLE LINEITEM_TABLE
      ((constraint LINEITEM_PKEY primary key (NESTED_TABLE_ID, SYS_NC_ARRAY_INDEX$)))
      lob (XMLDATA.NOTES) STORE AS (ENABLE STORAGE IN ROW STORAGE(INITIAL 4K NEXT 32K))"/>
  <xs:complexType name="PurchaseOrderType" xdb:SQLType="PURCHASEORDER_T">
    <xs:sequence>
      <xs:element name="Actions" type="ActionsType" xdb:SQLName="ACTIONS"/>
      <xs:element name="Reject" type="RejectionType" minOccurs="0" xdb:SQLName="REJECTION"/>
      <xs:element name="Requestor" type="RequestorType" xdb:SQLName="REQUESTOR"/>
      <xs:element name="User" type="UserType" xdb:SQLName="USERID"/>
      <xs:element name="CostCenter" type="CostCenterType" xdb:SQLName="COST_CENTER"/>
      <xs:element name="BillingAddress" type="AddressType" minOccurs="0"
        xdb:SQLName="BILLING_ADDRESS"/>
      <xs:element name="ShippingInstructions" type="ShippingInstructionsType"
        xdb:SQLName="SHIPPING_INSTRUCTIONS"/>
      <xs:element name="SpecialInstructions" type="SpecialInstructionsType"
        xdb:SQLName="SPECIAL_INSTRUCTIONS"/>
      <xs:element name="LineItems" type="LineItemsType" xdb:SQLName="LINEITEMS"/>
      <xs:element name="Notes" type="NotesType" minOccurs="0" xdb:SQLType="CLOB"
        xdb:SQLName="NOTES"/>
    </xs:sequence>
    <xs:attribute name="Reference" type="ReferenceType" use="required" xdb:SQLName="REFERENCE"/>
    <xs:attribute name="DateCreated" type="xs:dateTime" use="required"
      xdb:SQLType="TIMESTAMP WITH TIME ZONE"/>
  </xs:complexType>
  <xs:complexType name="LineItemsType" xdb:SQLType="LINEITEMS_T">
    <xs:sequence>
      <xs:element name="LineItem" type="LineItemType" maxOccurs="unbounded" xdb:SQLName="LINEITEM"
        xdb:SQLCollType="LINEITEM_V"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="LineItemType" xdb:SQLType="LINEITEM_T">
    <xs:sequence>
      <xs:element name="Part" type="PartType" xdb:SQLName="PART"/>
      <xs:element name="Quantity" type="quantityType"/>
    </xs:sequence>
    <xs:attribute name="ItemNumber" type="xs:integer" xdb:SQLName="ITEMNUMBER"
      xdb:SQLType="NUMBER"/>
  </xs:complexType>
  <xs:complexType name="PartType" xdb:SQLType="PART_T">
    <xs:simpleContent>
      <xs:extension base="UPCCodeType">
        <xs:attribute name="Description" type="DescriptionType" use="required"

```



```

        xdb:SQLName="DESCRIPTION"/>
        <xs:attribute name="UnitCost" type="moneyType" use="required"/>
    </xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:simpleType name="ReferenceType">
    <xs:restriction base="xs:string">
        <xs:minLength value="18"/>
        <xs:maxLength value="30"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="ActionsType" xdb:SQLType="ACTIONS_T">
    <xs:sequence>
        <xs:element name="Action" maxOccurs="4" xdb:SQLName="ACTION" xdb:SQLCollType="ACTION_V">
            <xs:complexType xdb:SQLType="ACTION_T">
                <xs:sequence>
                    <xs:element name="User" type="UserType" xdb:SQLName="ACTIONED_BY"/>
                    <xs:element name="Date" type="DateType" minOccurs="0" xdb:SQLName="DATE_ACTIONED"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="RejectionType" xdb:SQLType="REJECTION_T">
    <xs:all>
        <xs:element name="User" type="UserType" minOccurs="0" xdb:SQLName="REJECTED_BY"/>
        <xs:element name="Date" type="DateType" minOccurs="0" xdb:SQLName="DATE_REJECTED"/>
        <xs:element name="Comments" type="CommentsType" minOccurs="0" xdb:SQLName="REASON_REJECTED"/>
    </xs:all>
</xs:complexType>
<xs:complexType name="ShippingInstructionsType" xdb:SQLType="SHIPPING_INSTRUCTIONS_T">
    <xs:sequence>
        <xs:element name="name" type="NameType" minOccurs="0" xdb:SQLName="SHIP_TO_NAME"/>
        <xs:choice>
            <xs:element name="address" type="AddressType" minOccurs="0"/>
            <xs:element name="fullAddress" type="FullAddressType" minOccurs="0"
                xdb:SQLName="SHIP_TO_ADDRESS"/>
        </xs:choice>
        <xs:element name="telephone" type="TelephoneType" minOccurs="0" xdb:SQLName="SHIP_TO_PHONE"/>
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="moneyType">
    <xs:restriction base="xs:decimal">
        <xs:fractionDigits value="2"/>
        <xs:totalDigits value="12"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="quantityType">
    <xs:restriction base="xs:decimal">
        <xs:fractionDigits value="4"/>
        <xs:totalDigits value="8"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="UserType">
    <xs:restriction base="xs:string">
        <xs:minLength value="0"/>
        <xs:maxLength value="10"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="RequestorType">
    <xs:restriction base="xs:string">
        <xs:minLength value="0"/>
        <xs:maxLength value="128"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="CostCenterType">
    <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="4"/>
    </xs:restriction>
</xs:simpleType>

```

```

</xs:simpleType>
<xs:simpleType name="VendorType">
  <xs:restriction base="xs:string">
    <xs:minLength value="0"/>
    <xs:maxLength value="20"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="PurchaseOrderNumberType">
  <xs:restriction base="xs:integer"/>
</xs:simpleType>
<xs:simpleType name="SpecialInstructionsType">
  <xs:restriction base="xs:string">
    <xs:minLength value="0"/>
    <xs:maxLength value="2048"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="NameType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="20"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="FullAddressType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="256"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="TelephoneType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="24"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="DateType">
  <xs:restriction base="xs:date"/>
</xs:simpleType>
<xs:simpleType name="CommentsType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="2048"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="DescriptionType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="256"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="AddressType" xdb:SQLType="ADDRESS_T">
  <xs:sequence>
    <xs:element name="StreetLine1" type="StreetType"/>
    <xs:element name="StreetLine2" type="StreetType" minOccurs="0"/>
    <xs:element name="City" type="CityType"/>
    <xs:choice>
      <xs:sequence>
        <xs:element name="State" type="StateType"/>
        <xs:element name="ZipCode" type="ZipCodeType"/>
      </xs:sequence>
      <xs:sequence>
        <xs:element name="Province" type="ProvinceType"/>
        <xs:element name="PostCode" type="PostCodeType"/>
      </xs:sequence>
      <xs:sequence>
        <xs:element name="County" type="CountyType"/>
        <xs:element name="Postcode" type="PostCodeType"/>
      </xs:sequence>
    </xs:choice>
    <xs:element name="Country" type="CountryType"/>
  </xs:sequence>

```

```

</xs:complexType>
<xs:simpleType name="StreetType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="128"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="CityType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="64"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="StateType">
  <xs:restriction base="xs:string">
    <xs:minLength value="2"/>
    <xs:maxLength value="2"/>
    <xs:enumeration value="AK"/>
    <xs:enumeration value="AL"/>
    <xs:enumeration value="AR"/>
    <xs:enumeration value="AS"/>
    <xs:enumeration value="AZ"/>
    <xs:enumeration value="CA"/>
    <xs:enumeration value="CO"/>
    <xs:enumeration value="CT"/>
    <xs:enumeration value="DC"/>
    <xs:enumeration value="DE"/>
    <xs:enumeration value="FL"/>
    <xs:enumeration value="FM"/>
    <xs:enumeration value="GA"/>
    <xs:enumeration value="GU"/>
    <xs:enumeration value="HI"/>
    <xs:enumeration value="IA"/>
    <xs:enumeration value="ID"/>
    <xs:enumeration value="IL"/>
    <xs:enumeration value="IN"/>
    <xs:enumeration value="KS"/>
    <xs:enumeration value="KY"/>
    <xs:enumeration value="LA"/>
    <xs:enumeration value="MA"/>
    <xs:enumeration value="MD"/>
    <xs:enumeration value="ME"/>
    <xs:enumeration value="MH"/>
    <xs:enumeration value="MI"/>
    <xs:enumeration value="MN"/>
    <xs:enumeration value="MO"/>
    <xs:enumeration value="MP"/>
    <xs:enumeration value="MQ"/>
    <xs:enumeration value="MS"/>
    <xs:enumeration value="MT"/>
    <xs:enumeration value="NC"/>
    <xs:enumeration value="ND"/>
    <xs:enumeration value="NE"/>
    <xs:enumeration value="NH"/>
    <xs:enumeration value="NJ"/>
    <xs:enumeration value="NM"/>
    <xs:enumeration value="NV"/>
    <xs:enumeration value="NY"/>
    <xs:enumeration value="OH"/>
    <xs:enumeration value="OK"/>
    <xs:enumeration value="OR"/>
    <xs:enumeration value="PA"/>
    <xs:enumeration value="PR"/>
    <xs:enumeration value="PW"/>
    <xs:enumeration value="RI"/>
    <xs:enumeration value="SC"/>
    <xs:enumeration value="SD"/>
    <xs:enumeration value="TN"/>
    <xs:enumeration value="TX"/>
    <xs:enumeration value="UM"/>
  </xs:restriction>
</xs:simpleType>

```

```

    <xs:enumeration value="UT"/>
    <xs:enumeration value="VA"/>
    <xs:enumeration value="VI"/>
    <xs:enumeration value="VT"/>
    <xs:enumeration value="WA"/>
    <xs:enumeration value="WI"/>
    <xs:enumeration value="WV"/>
    <xs:enumeration value="WY"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ZipCodeType">
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{5}"/>
    <xs:pattern value="\d{5}-\d{4}"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="CountryType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="64"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="CountyType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="32"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="PostCodeType">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="12"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ProvinceType">
  <xs:restriction base="xs:string">
    <xs:minLength value="2"/>
    <xs:maxLength value="2"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="NotesType">
  <xs:restriction base="xs:string">
    <xs:maxLength value="32767"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="UPCCodeType">
  <xs:restriction base="xs:string">
    <xs:minLength value="11"/>
    <xs:maxLength value="14"/>
    <xs:pattern value="\d{11}"/>
    <xs:pattern value="\d{12}"/>
    <xs:pattern value="\d{13}"/>
    <xs:pattern value="\d{14}"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

XSLT Stylesheet Example, PurchaseOrder.xsl

The full listing is presented of XSLT stylesheet `PurchaseOrder.xsl`, which is used in various examples.

Example A-4 PurchaseOrder.xsl XSLT Stylesheet

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"

```

```

        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:xdb="http://xmlns.oracle.com/xdb"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<xsl:template match="/">
  <html>
    <head/>
    <body bgcolor="#003333" text="#FFFFCC" link="#FFCC00"
      vlink="#66CC99" alink="#669999">
      <FONT FACE="Arial, Helvetica, sans-serif">
        <xsl:for-each select="PurchaseOrder"/>
        <xsl:for-each select="PurchaseOrder">
          <center>
            <span style="font-family:Arial; font-weight:bold">
              <FONT COLOR="#FF0000">
                <B>Purchase Order </B>
              </FONT>
            </span>
          </center>
          <br/>
          <center>
            <xsl:for-each select="Reference">
              <span style="font-family:Arial; font-weight:bold">
                <xsl:apply-templates/>
              </span>
            </xsl:for-each>
          </center>
        </xsl:for-each>
        <P>
          <xsl:for-each select="PurchaseOrder">
            <br/>
          </xsl:for-each>
        </P>
        <P>
          <xsl:for-each select="PurchaseOrder">
            <br/>
          </xsl:for-each>
        </P>
        <xsl:for-each select="PurchaseOrder"/>
        <xsl:for-each select="PurchaseOrder">
          <table border="0" width="100%" BGCOLOR="#000000">
            <tbody>
              <tr>
                <td WIDTH="296">
                  <P>
                    <B>
                      <FONT SIZE="+1" COLOR="#FF0000"
                        FACE="Arial, Helvetica, sans-serif">Internal
                      </FONT>
                    </B>
                  </P>
                  <table border="0" width="98%" BGCOLOR="#000099">
                    <tbody>
                      <tr>
                        <td WIDTH="49%">
                          <B>

```

```

        <FONT COLOR="#FFFF00">Actions</FONT>
    </B>
</td>
<td WIDTH="51%">
    <xsl:for-each select="Actions">
        <xsl:for-each select="Action">
            <table border="1" WIDTH="143">
                <xsl:if test="position()=1">
                    <thead>
                        <tr>
                            <td HEIGHT="21">
                                <FONT
                                    COLOR="#FFFF00">User</FONT>
                            </td>
                            <td HEIGHT="21">
                                <FONT
                                    COLOR="#FFFF00">Date</FONT>
                            </td>
                        </tr>
                    </thead>
                </xsl:if>
                <tbody>
                    <tr>
                        <td>
                            <xsl:for-each select="User">
                                <xsl:apply-templates/>
                            </xsl:for-each>
                        </td>
                        <td>
                            <xsl:for-each select="Date">
                                <xsl:apply-templates/>
                            </xsl:for-each>
                        </td>
                    </tr>
                </tbody>
            </table>
        </xsl:for-each>
    </xsl:for-each>
</td>
</tr>
<tr>
    <td WIDTH="49%">
        <B>
            <FONT COLOR="#FFFF00">Requestor</FONT>
        </B>
    </td>
    <td WIDTH="51%">
        <xsl:for-each select="Requestor">
            <xsl:apply-templates/>
        </xsl:for-each>
    </td>
</tr>
<tr>
    <td WIDTH="49%">
        <B>
            <FONT COLOR="#FFFF00">User</FONT>

```

```

        </B>
      </td>
      <td WIDTH="51%">
        <xsl:for-each select="User">
          <xsl:apply-templates/>
        </xsl:for-each>
      </td>
    </tr>
    <tr>
      <td WIDTH="49%">
        <B>
          <FONT COLOR="#FFFF00">Cost Center</FONT>
        </B>
      </td>
      <td WIDTH="51%">
        <xsl:for-each select="CostCenter">
          <xsl:apply-templates/>
        </xsl:for-each>
      </td>
    </tr>
  </tbody>
</table>
</td>
<td width="93"/>
<td valign="top" WIDTH="340">
  <B>
    <FONT COLOR="#FF0000">
      <FONT SIZE="+1">Ship To</FONT>
    </FONT>
  </B>
  <xsl:for-each select="ShippingInstructions">
    <xsl:if test="position()=1"/>
  </xsl:for-each>
  <xsl:for-each select="ShippingInstructions">
    <xsl:if test="position()=1">
      <table border="0" BGCOLOR="#999900">
        <tbody>
          <tr>
            <td WIDTH="126" HEIGHT="24">
              <B>Name</B>
            </td>
            <xsl:for-each
              select="../ShippingInstructions">
              <td WIDTH="218" HEIGHT="24">
                <xsl:for-each select="name">
                  <xsl:apply-templates/>
                </xsl:for-each>
              </td>
            </xsl:for-each>
          </tr>
          <tr>
            <td WIDTH="126" HEIGHT="34">
              <B>Address</B>
            </td>
            <xsl:for-each
              select="../ShippingInstructions">

```

```

        <td WIDTH="218" HEIGHT="34">
            <xsl:for-each select="address">
                <span style="white-space:pre">
                    <xsl:apply-templates/>
                </span>
            </xsl:for-each>
        </td>
    </xsl:for-each>
</tr>
<tr>
    <td WIDTH="126" HEIGHT="32">
        <B>Telephone</B>
    </td>
    <xsl:for-each
        select="../ShippingInstructions">
        <td WIDTH="218" HEIGHT="32">
            <xsl:for-each select="telephone">
                <xsl:apply-templates/>
            </xsl:for-each>
        </td>
    </xsl:for-each>
    </tr>
</tbody>
</table>
</xsl:if>
</xsl:for-each>
</td>
</tr>
</tbody>
</table>
<br/>
<B>
    <FONT COLOR="#FF0000" SIZE="+1">Items:</FONT>
</B>
<br/>
<br/>
<table border="0">
    <xsl:for-each select="LineItems">
        <xsl:for-each select="LineItem">
            <xsl:if test="position()=1">
                <thead>
                    <tr bgcolor="#C0C0C0">
                        <td>
                            <FONT COLOR="#FF0000">
                                <B>ItemNumber</B>
                            </FONT>
                        </td>
                        <td>
                            <FONT COLOR="#FF0000">
                                <B>Description</B>
                            </FONT>
                        </td>
                        <td>
                            <FONT COLOR="#FF0000">
                                <B>PartId</B>
                            </FONT>

```



```

        </td>
        <td>
            <FONT COLOR="#FF0000">
                <B>Quantity</B>
            </FONT>
        </td>
        <td>
            <FONT COLOR="#FF0000">
                <B>Unit Price</B>
            </FONT>
        </td>
        <td>
            <FONT COLOR="#FF0000">
                <B>Total Price</B>
            </FONT>
        </td>
    </tr>
</thead>
</xsl:if>
<tbody>
    <tr bgcolor="#DADADA">
        <td>
            <FONT COLOR="#000000">
                <xsl:for-each select="@ItemNumber">
                    <xsl:value-of select="."/>
                </xsl:for-each>
            </FONT>
        </td>
        <td>
            <FONT COLOR="#000000">
                <xsl:for-each select="Description">
                    <xsl:apply-templates/>
                </xsl:for-each>
            </FONT>
        </td>
        <td>
            <FONT COLOR="#000000">
                <xsl:for-each select="Part">
                    <xsl:for-each select="@Id">
                        <xsl:value-of select="."/>
                    </xsl:for-each>
                </xsl:for-each>
            </FONT>
        </td>
        <td>
            <FONT COLOR="#000000">
                <xsl:for-each select="Part">
                    <xsl:for-each select="@Quantity">
                        <xsl:value-of select="."/>
                    </xsl:for-each>
                </xsl:for-each>
            </FONT>
        </td>
        <td>
            <FONT COLOR="#000000">
                <xsl:for-each select="Part">

```

```

        <xsl:for-each select="@UnitPrice">
            <xsl:value-of select="."/>
        </xsl:for-each>
    </xsl:for-each>
</FONT>
</td>
<td>
    <FONT FACE="Arial, Helvetica, sans-serif"
        COLOR="#000000">
        <xsl:for-each select="Part">
            <xsl:value-of select="@Quantity*@UnitPrice"/>
        </xsl:for-each>
    </FONT>
</td>
</tr>
</tbody>
</xsl:for-each>
</xsl:for-each>
</table>
</xsl:for-each>
</FONT>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Loading XML Data Using C (OCI)

A full listing of a C program that inserts XML data into an `XMLType` table is presented.

The same program is partially listed in [Loading XML Content Using C](#).

Example A-5 Inserting XML Data into an `XMLType` Table Using C

```

#include "stdio.h"
#include <xml.h>
#include <stdlib.h>
#include <string.h>
#include <ocixmllob.h>
OCIEnv *envhp;
OCIError *errhp;
OCISvcCtx *svchp;
OCIStmt *stmthp;
OCIServer *srvhp;
OCIDuration dur;
OCISession *sesshp;
oratext *username = "QUINE";
oratext *password = "*****"; /* Replace with real password */
oratext *filename = "AMCEWEN-20021009123336171PDT.xml";
oratext *schemaloc = "http://localhost:8080/source/schemas/poSource/xsd/purchaseOrder.xsd";

/* Execute a SQL statement that binds XML data */
sword exec_bind_xml(OCISvcCtx *svchp, OCIError *errhp, OCIStmt *stmthp,
                    void *xml, OCIType *xmltdo, OraText *sqlstmt)
{
    OCIBind *bndhpl = (OCIBind *) 0;
    sword status = 0;
    OCIInd ind = OCI_IND_NOTNULL;
    OCIInd *indp = &ind;

```

```

if(status = OCISstmtPrepare(stmthp, errhp, (OraText *)sqlstmt,
                           (ub4)strlen((const char *)sqlstmt),
                           (ub4) OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT))

    return OCI_ERROR;
if(status = OCIBindByPos(stmthp, &bndhpl, errhp, (ub4) 1, (dvoid *) 0,
                        (sb4) 0, SQLT_NTY, (dvoid *) 0, (ub2 *)0,
                        (ub2 *)0, (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT))

    return OCI_ERROR;
if(status = OCIBindObject(bndhpl, errhp, (CONST OCIType *) xmltdo,
                        (dvoid **) &xml, (ub4 *) 0,
                        (dvoid **) &indp, (ub4 *) 0))

    return OCI_ERROR;
if(status = OCISstmtExecute(svchp, stmthp, errhp, (ub4) 1, (ub4) 0,
                        (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                        (ub4) OCI_DEFAULT))

    return OCI_ERROR;
return OCI_SUCCESS;
}

/* Initialize OCI handles, and connect */
sword init_oci_connect()
{
    sword status;
    if (OCIEnvCreate((OCIEnv **) &(envhp), (ub4) OCI_OBJECT,
                    (dvoid *) 0, (dvoid * *) (dvoid *,size_t)) 0,
                    (dvoid * *) (dvoid *, dvoid *, size_t)) 0,
                    (void *) (dvoid *, dvoid *) 0, (size_t) 0, (dvoid **) 0))

    {
        printf("FAILED: OCIEnvCreate()\n");
        return OCI_ERROR;
    }
    /* Allocate error handle */
    if (OCIHandleAlloc((dvoid *) envhp, (dvoid **) &(errhp),
                      (ub4) OCI_HTYPE_ERROR, (size_t) 0, (dvoid **) 0))

    {
        printf("FAILED: OCIHandleAlloc() on errhp\n");
        return OCI_ERROR;
    }
    /* Allocate server handle */
    if (status = OCIHandleAlloc((dvoid *) envhp, (dvoid **) &srvhp,
                              (ub4) OCI_HTYPE_SERVER, (size_t) 0, (dvoid **) 0))

    {
        printf("FAILED: OCIHandleAlloc() on srvhp\n");
        return OCI_ERROR;
    }
    /* Allocate service context handle */
    if (status = OCIHandleAlloc((dvoid *) envhp,
                              (dvoid **) &(svchp), (ub4) OCI_HTYPE_SVCCTX,
                              (size_t) 0, (dvoid **) 0))

    {
        printf("FAILED: OCIHandleAlloc() on svchp\n");
        return OCI_ERROR;
    }
    /* Allocate session handle */
    if (status = OCIHandleAlloc((dvoid *) envhp, (dvoid **) &sesshp ,
                              (ub4) OCI_HTYPE_SESSION, (size_t) 0, (dvoid **) 0))

    {
        printf("FAILED: OCIHandleAlloc() on sesshp\n");
        return OCI_ERROR;
    }
    /* Allocate statement handle */
    if (OCIHandleAlloc((dvoid *)envhp, (dvoid **) &stmthp,
                      (ub4)OCI_HTYPE_STMT, (CONST size_t) 0, (dvoid **) 0))

    {
        printf("FAILED: OCIHandleAlloc() on stmthp\n");
        return status;
    }
    if (status = OCIServerAttach((OCIServer *) srvhp, (OCIError *) errhp,
                              (CONST oratext *)"", 0, (ub4) OCI_DEFAULT))

    {

```

```

        printf("FAILED: OCIServerAttach() on srvhp\n");
        return OCI_ERROR;
    }
    /* Set server attribute to service context */
    if (status = OCIAttrSet((dvoid *) svchp, (ub4) OCI_HTYPE_SVCCTX,
                           (dvoid *) srvhp, (ub4) 0, (ub4) OCI_ATTR_SERVER,
                           (OCIError *) errhp))
    {
        printf("FAILED: OCIAttrSet() on svchp\n");
        return OCI_ERROR;
    }
    /* Set user attribute to session */
    if (status = OCIAttrSet((dvoid *) sesshp, (ub4) OCI_HTYPE_SESSION,
                           (dvoid *) username,
                           (ub4) strlen((const char *) username),
                           (ub4) OCI_ATTR_USERNAME, (OCIError *) errhp))
    {
        printf("FAILED: OCIAttrSet() on authp for user\n");
        return OCI_ERROR;
    }
    /* Set password attribute to session */
    if (status = OCIAttrSet((dvoid *) sesshp, (ub4) OCI_HTYPE_SESSION,
                           (dvoid *) password,
                           (ub4) strlen((const char *) password),
                           (ub4) OCI_ATTR_PASSWORD, (OCIError *) errhp))
    {
        printf("FAILED: OCIAttrSet() on authp for password\n");
        return OCI_ERROR;
    }
    /* Begin a session */
    if (status = OCISessionBegin((OCISvcCtx *) svchp,
                                (OCIError *) errhp,
                                (OCISession *) sesshp, (ub4) OCI_CRED_RDBMS,
                                (ub4) OCI_STMT_CACHE))
    {
        printf("FAILED: OCISessionBegin(). Make sure database is up and the username/password is valid. \n");
        return OCI_ERROR;
    }
    /* Set session attribute to service context */
    if (status = OCIAttrSet((dvoid *) svchp, (ub4) OCI_HTYPE_SVCCTX,
                           (dvoid *) sesshp, (ub4) 0, (ub4) OCI_ATTR_SESSION,
                           (OCIError *) errhp))
    {
        printf("FAILED: OCIAttrSet() on svchp\n");
        return OCI_ERROR;
    }
}

/* Free OCI handles, and disconnect */
void free_oci()
{
    sword status = 0;

    /* End the session */
    if (status = OCISessionEnd((OCISvcCtx *) svchp, (OCIError *) errhp,
                              (OCISession *) sesshp, (ub4) OCI_DEFAULT))
    {
        if (envhp)
            OCIHandleFree((dvoid *) envhp, OCI_HTYPE_ENV);
        return;
    }
    /* Detach from the server */
    if (status = OCIServerDetach((OCIServer *) srvhp, (OCIError *) errhp,
                                (ub4) OCI_DEFAULT))
    {
        if (envhp)
            OCIHandleFree((dvoid *) envhp, OCI_HTYPE_ENV);
        return;
    }
    /* Free the handles */
}

```

```

if (stmthp) OCIHandleFree((dvoid *)stmthp, (ub4) OCI_HTYPE_STMT);
if (sesshp) OCIHandleFree((dvoid *)sesshp, (ub4) OCI_HTYPE_SESSION);
if (svchp) OCIHandleFree((dvoid *)svchp, (ub4) OCI_HTYPE_SVCCTX);
if (srvhp) OCIHandleFree((dvoid *)srvhp, (ub4) OCI_HTYPE_SERVER);
if (errhp) OCIHandleFree((dvoid *)errhp, (ub4) OCI_HTYPE_ERROR);
if (envhp) OCIHandleFree((dvoid *)envhp, (ub4) OCI_HTYPE_ENV);
return;
}

void main()
{
    OCIType *xmldto;
    xmldocnode *doc;
    ocixmlldbparam params[1];
    xmlerr      err;
    xmlctx      *xctx;
    oratext      *ins_stmt;
    sword        status;
    xmlnode      *root;
    oratext      buf[10000];

    /* Initialize envhp, svchp, errhp, dur, stmthp */
    init_oci_connect();

    /* Get an XML context */
    params[0].name_ocixmlldbparam = XCTXINIT_OCIDUR;
    params[0].value_ocixmlldbparam = &dur;
    xctx = OCIXmlDbInitXmlCtx(envhp, svchp, errhp, params, 1);
    if (!(doc = XmlLoadDom(xctx, &err, "file", filename,
                          "schema_location", schemaloc, NULL)))
    {
        printf("Parse failed.\n");
        return;
    }
    else
        printf("Parse succeeded.\n");
    root = XmlDomGetDocElem(xctx, doc);
    printf("The xml document is :\n");
    XmlSaveDom(xctx, &err, (xmlnode *)doc, "buffer", buf, "buffer_length", 10000, NULL);
    printf("%s\n", buf);

    /* Insert the document into my_table */
    ins_stmt = (oratext *)"insert into purchaseorder values (:1)";
    status = OCITypeByName(envhp, errhp, svchp, (const text *) "SYS",
                          (ub4) strlen((const char *) "SYS"), (const text *) "XMLTYPE",
                          (ub4) strlen((const char *) "XMLTYPE"), (CONST text *) 0,
                          (ub4) 0, OCI_DURATION_SESSION, OCI_TYPEGET_HEADER,
                          (OCIType **) &xmldto);
    if (status == OCI_SUCCESS)
    {
        status = exec_bind_xml(svchp, errhp, stmthp, (void *)doc,
                              xmldto, ins_stmt);
    }
    if (status == OCI_SUCCESS)
        printf ("Insert successful\n");
    else
        printf ("Insert failed\n");

    /* Free XML instances */
    if (doc) XmlFreeDocument((xmlctx *)xctx, (xmldocnode *)doc);

    /* Free XML CTX */
    OCIXmlDbFreeXmlCtx(xctx);
    free_oci();
}

```

Initializing and Terminating an XML Context (OCI)

An example shows how to use OCI functions `OCIXmlDbInitXmlCtx()` and `OCIXmlDbFreeXmlCtx()` to initialize and terminate an XML context. It constructs an XML document using the C DOM API and saves it to the database.

[Example A-6](#) is partially listed in [C DOM API for XMLType](#) and [Initializing and Terminating an XML Context](#). It assumes that the following SQL code has first been executed to create table `my_table` in database schema `CAPIUSER`:

```
CONNECT CAPIUSER
Enter password: <password>

Connected.

CREATE TABLE my_table OF XMLType;
```

Example A-6 Using `OCIXmlDbInitXmlCtx()` and `OCIXmlDbFreeXmlCtx()`

```
#ifndef S_ORACLE
#endif
#ifndef ORATYPES_ORACLE
#include <oratypes.h>
#endif
#ifndef XML_ORACLE
#include <xml.h>
#endif
#ifndef OCIXML_ORACLE
#include <ocixml.h>
#endif
#ifndef OCI_ORACLE
#include <oci.h>
#endif
#include <string.h>

typedef struct test_ctx {
    OCIEnv *envhp;
    OCIError *errhp;
    OCISvcCtx *svchp;
    OCISmt *stmthp;
    OCIServer *srvhp;
    OCIDuration dur;
    OCISession *sesshp;
    oratext *username;
    oratext *password;
} test_ctx;

/* Helper function 1: execute a sql statement which binds xml data */
static sword exec_bind_xml(OCISvcCtx *svchp,
    OCIError *errhp,
    OCISmt *stmthp,
    void *xml,
    OCIType *xmltdo,
    OraText *sqlstmt);

/* Helper function 2: Initialize OCI handles and connect */
static sword init_oci_handles(test_ctx *ctx);

/* Helper function 3: Free OCI handles and disconnect */
static sword free_oci_handles(test_ctx *ctx);

void main()
```

```

{
  test_ctx temp_ctx;
  test_ctx *ctx = &temp_ctx;
  OCIText *xmltdo = (OCIText *) 0;
  xmldocnode *doc = (xmldocnode *)0;
  ocixmlldbparam params[1];
  xmlnode *quux, *foo, *foo_data, *top;
  xmlerr err;
  sword status = 0;
  xmlctx *xctx;

  oratext ins_stmt[] = "insert into my_table values (:1)";
  oratext tlpxml_test_sch[] = "<TOP/>";
  ctx->username = (oratext *)"CAPIUSER";
  ctx->password = (oratext *)"*****"; /* Replace with real password */

  /* Initialize envhp, svchp, errhp, dur, stmthp */
  init_oci_handles(ctx);

  /* Get an xml context */
  params[0].name_ocixmlldbparam = XCTXINIT_OCIDUR;
  params[0].value_ocixmlldbparam = &ctx->dur;
  xctx = OCIXmlDbInitXmlCtx(ctx->envhp, ctx->svchp, ctx->errhp, params, 1);

  /* Start processing - first, check that this DOM supports XML 1.0 */
  printf("\n\nSupports XML 1.0? : %s\n",
        XmlHasFeature(xctx, (oratext *) "xml", (oratext *) "1.0") ?
        "YES" : "NO");

  /* Parse a document */
  if (!(doc = XmlLoadDom(xctx, &err, "buffer", tlpxml_test_sch,
                        "buffer_length", sizeof(tlpxml_test_sch)-1,
                        "validate", TRUE, NULL)))
  {
    printf("Parse failed, code %d\n", err);
  }
  else
  {
    /* Get the document element */
    top = (xmlnode *)XmlDomGetDocElem(xctx, doc);

    /* Print out the top element */
    printf("\n\nOriginal top element is :\n");
    XmlSaveDom(xctx, &err, top, "stdio", stdout, NULL);

    /* Print out the document. The changes are reflected here */
    printf("\n\nOriginal document is :\n");
    XmlSaveDom(xctx, &err, (xmlnode *)doc, "stdio", stdout, NULL);

    /* Create some elements and add them to the document */
    quux = (xmlnode *) XmlDomCreateElem(xctx, doc, (oratext *) "QUUX");
    foo = (xmlnode *) XmlDomCreateElem(xctx, doc, (oratext *) "FOO");
    foo_data = (xmlnode *) XmlDomCreateText(xctx, doc, (oratext *) "data");
    foo_data = XmlDomAppendChild(xctx, (xmlnode *) foo, (xmlnode *) foo_data);
    foo = XmlDomAppendChild(xctx, quux, foo);
    quux = XmlDomAppendChild(xctx, top, quux);

    /* Print out the top element */
    printf("\n\nNow the top element is :\n");
    XmlSaveDom(xctx, &err, top, "stdio", stdout, NULL);

    /* Print out the document. The changes are reflected here */
    printf("\n\nNow the document is :\n");
    XmlSaveDom(xctx, &err, (xmlnode *)doc, "stdio", stdout, NULL);

    /* Insert the document into my_table */
    status = OCITextByName(ctx->envhp, ctx->errhp, ctx->svchp,
                          (const text *) "SYS", (ub4) strlen((char *) "SYS"),
                          (const text *) "XMLTYPE",
                          (ub4) strlen((char *) "XMLTYPE"), (CONST text *) 0,

```

```

        (ub4) 0, OCI_DURATION_SESSION, OCI_TYPEGET_HEADER,
        (OCIType **) &xmldto);
    if (status == OCI_SUCCESS)
    {
        exec_bind_xml(ctx->svchp, ctx->errhp, ctx->stmthp, (void *)doc, xmldto,
            ins_stmt);
    }
}
/* Free xml ctx */
OCIXmlDbFreeXmlCtx(xctx);

/* Free envhp, svchp, errhp, stmthp */
free_oci_handles(ctx);
}

/* Helper function 1: execute a SQL statement that binds xml data */
static sword exec_bind_xml(OCISvcCtx *svchp,
                           OCLError *errhp,
                           OCISmt *stmthp,
                           void *xml,
                           OCIType *xmldto,
                           OraText *sqlstmt)
{
    OCIBind *bndhpl = (OCIBind *) 0;
    sword status = 0;
    OCInd ind = OCI_IND_NOTNULL;
    OCInd *indp = &ind;
    if(status = OCISmtPrepare(stmthp, errhp, (OraText *)sqlstmt,
        (ub4)strlen((char *)sqlstmt),
        (ub4) OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT)) {
        printf("Failed OCISmtPrepare\n");
        return OCI_ERROR;
    }
    if(status = OCIBindByPos(stmthp, &bndhpl, errhp, (ub4) 1, (dvoid *) 0,
        (sb4) 0, SOLT_NTY, (dvoid *) 0, (ub2 *)0,
        (ub2 *)0, (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT)) {
        printf("Failed OCIBindByPos\n");
        return OCI_ERROR;
    }
    if(status = OCIBindObject(bndhpl, errhp, (CONST OCIType *) xmldto, (dvoid **)
        &xml,
        (ub4 *) 0, (dvoid **) &indp, (ub4 *) 0)) {
        printf("Failed OCIBindObject\n");
        return OCI_ERROR;
    }
    if(status = OCISmtExecute(svchp, stmthp, errhp, (ub4) 1, (ub4) 0,
        (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
        (ub4) OCI_DEFAULT)) {
        printf("Failed OCISmtExecute\n");
        return OCI_ERROR;
    }
    return OCI_SUCCESS;
}

/* Helper function 2: Initialize OCI handles and connect */
static sword init_oci_handles(test_ctx *ctx)
{
    sword status;
    ctx->dur = OCI_DURATION_SESSION;
    if (OCIEnvCreate((OCIEnv **) &(ctx->envhp), (ub4) OCI_OBJECT,
        (dvoid *) 0, (dvoid *) (dvoid *,size_t) 0,
        (dvoid *) (dvoid *, dvoid *, size_t) 0,
        (void *) (dvoid *, dvoid *) 0, (size_t) 0, (dvoid **) 0))
    {
        printf("FAILED: OCIEnvCreate()\n");
        return OCI_ERROR;
    }
    /* Allocate error handle */
    if (OCIHandleAlloc((dvoid *) ctx->envhp, (dvoid **) &(ctx->errhp),
        (ub4) OCI_HTYPE_ERROR, (size_t) 0, (dvoid **) 0))

```



```

{
    printf("FAILED: OCIHandleAlloc() on errhp\n");
    return OCI_ERROR;
}
/* Allocate server handle */
if (status = OCIHandleAlloc((dvoid *) ctx->envhp, (dvoid **) &ctx->srvhp,
                            (ub4) OCI_HTYPE_SERVER, (size_t) 0, (dvoid **) 0))
{
    printf("FAILED: OCIHandleAlloc() on srvhp\n");
    return OCI_ERROR;
}
/* Allocate service context handle */
if (status = OCIHandleAlloc((dvoid *) ctx->envhp,
                            (dvoid **) &(ctx->svchp), (ub4) OCI_HTYPE_SVCCTX,
                            (size_t) 0, (dvoid **) 0))
{
    printf("FAILED: OCIHandleAlloc() on svchp\n");
    return OCI_ERROR;
}
/* Allocate session handle */
if (status = OCIHandleAlloc((dvoid *) ctx->envhp, (dvoid **) &ctx->sesshp,
                            (ub4) OCI_HTYPE_SESSION, (size_t) 0, (dvoid **) 0))
{
    printf("FAILED: OCIHandleAlloc() on sesshp\n");
    return OCI_ERROR;
}
/* Allocate statement handle */
if (OCIHandleAlloc((dvoid *) ctx->envhp, (dvoid **) &ctx->stmthp,
                  (ub4) OCI_HTYPE_STMT, (CONST size_t) 0, (dvoid **) 0))
{
    printf("FAILED: OCIHandleAlloc() on stmthp\n");
    return status;
}
if (status = OCIServerAttach((OCIServer *) ctx->srvhp, (OCIError *) ctx->errhp,
                            (CONST oratext *) "", 0, (ub4) OCI_DEFAULT))
{
    printf("FAILED: OCIServerAttach() on srvhp\n");
    return OCI_ERROR;
}
/* Set server attribute to service context */
if (status = OCIAttrSet((dvoid *) ctx->svchp, (ub4) OCI_HTYPE_SVCCTX,
                       (dvoid *) ctx->srvhp, (ub4) 0, (ub4) OCI_ATTR_SERVER,
                       (OCIError *) ctx->errhp))
{
    printf("FAILED: OCIAttrSet() on svchp\n");
    return OCI_ERROR;
}
/* Set user attribute to session */
if (status = OCIAttrSet((dvoid *) ctx->sesshp, (ub4) OCI_HTYPE_SESSION,
                       (dvoid *) ctx->username,
                       (ub4) strlen((char *) ctx->username),
                       (ub4) OCI_ATTR_USERNAME, (OCIError *) ctx->errhp))
{
    printf("FAILED: OCIAttrSet() on authp for user\n");
    return OCI_ERROR;
}
/* Set password attribute to session */
if (status = OCIAttrSet((dvoid *) ctx->sesshp, (ub4) OCI_HTYPE_SESSION,
                       (dvoid *) ctx->password,
                       (ub4) strlen((char *) ctx->password),
                       (ub4) OCI_ATTR_PASSWORD, (OCIError *) ctx->errhp))
{
    printf("FAILED: OCIAttrSet() on authp for password\n");
    return OCI_ERROR;
}
/* Begin a session */
if (status = OCISessionBegin((OCISvcCtx *) ctx->svchp,
                             (OCIError *) ctx->errhp,
                             (OCISession *) ctx->sesshp, (ub4) OCI_CRED_RDBMS,
                             (ub4) OCI_STMT_CACHE))

```

```

{
    printf("FAILED: OCISessionBegin(). Make sure database is up and the \
        username/password is valid. \n");
    return OCI_ERROR;
}
/* Set session attribute to service context */
if (status = OCIAAttrSet((dvoid *) ctx->svchp, (ub4) OCI_HTYPE_SVCCTX,
                        (dvoid *)ctx->sesshp, (ub4) 0, (ub4) OCI_ATTR_SESSION,
                        (OCIError *) ctx->errhp))
{
    printf("FAILED: OCIAAttrSet() on svchp\n");
    return OCI_ERROR;
}
return status;
}

/* Helper function 3: Free OCI handles and disconnect */
static sword free_oci_handles(test_ctx *ctx)
{
    sword status = 0;
    /* End the session */
    if (status = OCISessionEnd((OCISvcCtx *)ctx->svchp, (OCIError *)ctx->errhp,
                              (OCISession *)ctx->sesshp, (ub4) OCI_DEFAULT))
    {
        if (ctx->envhp)
            OCIHandleFree((dvoid *)ctx->envhp, OCI_HTYPE_ENV);
        return status;
    }
    /* Detach from the server */
    if (status = OCIServerDetach((OCIServer *)ctx->srvhp, (OCIError *)ctx->errhp,
                                (ub4)OCI_DEFAULT))
    {
        if (ctx->envhp)
            OCIHandleFree((dvoid *)ctx->envhp, OCI_HTYPE_ENV);
        return status;
    }
    /* Free the handles */
    if (ctx->stmthp) OCIHandleFree((dvoid *)ctx->stmthp, (ub4) OCI_HTYPE_STMT);
    if (ctx->sesshp) OCIHandleFree((dvoid *)ctx->sesshp, (ub4) OCI_HTYPE_SESSION);
    if (ctx->svchp) OCIHandleFree((dvoid *)ctx->svchp, (ub4) OCI_HTYPE_SVCCTX);
    if (ctx->srvhp) OCIHandleFree((dvoid *)ctx->srvhp, (ub4) OCI_HTYPE_SERVER);
    if (ctx->errhp) OCIHandleFree((dvoid *)ctx->errhp, (ub4) OCI_HTYPE_ERROR);
    if (ctx->envhp) OCIHandleFree((dvoid *)ctx->envhp, (ub4) OCI_HTYPE_ENV);
    return status;
}

```