ORACLE

Transportable binary XML in Oracle Database 23c

What - High Level Description

Transportable binary XML (TBX) is a new XMLType storage method. It is binary XML, like the existing Compact Schema-Aware XML format (CSX).

TBX offers everything that CSX offers, except for the use of an unstructured XMLIndex component (UXI), and XML search index can be used for the same use cases as UXI.

What - High Level Description (more detail)

- Unlike CSX, TBX is not dependent on a central token table or a schema registry; the XML data is self-contained.

- This allows TBX to support features such as sharding, Exadata pushdown, and distributed XML, which CSX does not support.

- You can migrate existing XMLType storage of a different format to TBX format in any of these ways:
  - Insert-as select or create-as-select
  - Online Redefinition
  - Data Pump

Why – the problem it solves, what are its benefits

- CSX was not well-suited to cloud usage, particularly Autonomous Database, and required access to the file system
- As a self-contained format with no external schema registry or token table, TBX is easily exchanged between different servers, containers and PDBs
- Can be used with advanced database features such as Sharding and XML Search Index
- Is supported by Exadata pushdown for operations on storage servers, limiting the impact on database servers
- Client-side encoding of TBX is far more efficient than CSX, as it requires no back-and-forth to look up token IDs in the central token table.

Context – how does it fit into the focus area's big picture

Application developers should not be limited to a single database.

XML should be treated as 'just another datatype' during application development, and developers should expect to be able to move data seamlessly through multiple application systems.

CSX will remain usable, and will remain the default XMLType storage in 23c. However, customers are encouraged to move to TBX, which will likely become the default XMLType storage in the next release.

Benefit To The Business

Business benefits from:

- Transportable nature of TBX, simplifying development processes
- Ability to make use of advanced features such as sharding and Exadata pushdown with XMLType data.

How to use it

Create TBX tables:

```
CREATE TABLE TAB1 OF XMLTYPE
XMLTYPE STORE AS TRANSPORTABLE BINARY XML;
```

Create tables with TBX columns:

```
CREATE TABLE TAB2 (id NUMBER, doc XMLTYPE)
XMLTYPE COLUMN doc STORE AS TRANSPORTABLE BINARY XML;
```

Add TBX columns to tables:

```
ALTER TABLE TAB3 ADD (doc XMLTYPE)
XMLTYPE COLUMN doc STORE AS TRANSPORTABLE BINARY XML;
```

Users can create old-fashioned CSX tables/columns by specifying NOT TRANSPORTABLE:

```
CREATE TABLE TAB4 OF XMLTYPE
XMLTYPE STORE AS NOT TRANSPORTABLE BINARY XML;
```


How to use it – online redefinition

Online Redefinition is the suggested option to migrate legacy storage options to TBX since it incurs no application downtime.

Setup:

```
create table p of xmltype xmltype store as binary XML;
create table int_p of xmltype xmltype store as transportable binary XML;
insert into p values (
xmltype('<PurchaseOrder xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:noNamespaceSchemaLocation="orx2.xsd">
  <Reference>ABSENT_LINES</Reference>
  <Requestor>Michael L. Allen</Requestor>
  <User>ALLEN</User>
  <CostCenter>S30</CostCenter>
</PurchaseOrder>'));
commit;
```


How to use it – online redefinition (continued)

```
-- this check should not throw an error:
exec DBMS_REDEFINITION.CAN_REDEF_TABLE('SCOTT', 'P', 'INT_P', DBMS_REDEFINITION.CONS_USE_ROWID);
begin
  DBMS_REDEFINITION.START_REDEF_TABLE( 'SCOTT', 'P', 'INT_P',
    options_flag => DBMS_REDEFINITION.CONS_USE_ROWID );
  DBMS_REDEFINITION.COPY_TABLE_DEPENDENTS('SCOTT', 'P', 'INT_P', 1,
                                          true, true, true, true, error_count, true);
  DBMS_REDEFINITION.SYNC_INTERIM_TABLE( 'SCOTT', 'P', 'INT_P' );
  DBMS_REDEFINITION.FINISH_REDEF_TABLE('SCOTT', 'P', 'INT_P');
end;
/
```