36
Export and Import of Oracle XML DB Data

You can use Oracle Data Pump to export and import `XMLType` tables for use with Oracle XML DB.

> **Note:**
>
> You can use the older export and import utilities `exp` and `imp` to migrate data to database releases that are prior to Oracle Database 11g. However, these older utilities do not support using `XMLType` data that is stored as binary XML.

- Overview of Exporting and Importing XMLType Tables
  Oracle XML DB supports export and import of `XMLType` tables and columns that store XML data, whether it is XML schema-based or not.
- Export/Import Limitations for Oracle XML DB Repository
  When you export or import tables that store data for Oracle XML DB Repository resources that are based on a registered XML schema, only that XML data is exported. The repository structure is lost during export, so that when these tables are imported they are not hierarchy-enabled.
- Export/Import Syntax and Examples
  Guidelines and examples are presented for using commands `expdp` and `impdp` with `XMLType` data.

Overview of Exporting and Importing XMLType Tables

Oracle XML DB supports export and import of `XMLType` tables and columns that store XML data, whether it is XML schema-based or not.

Oracle Data Pump enables high-speed movement of data and metadata from one database to another. There are two modes for using Oracle Data Pump: transportable tablespaces mode and non-transportable tablespaces mode.

For the transportable tablespaces mode there is this restriction regarding `XMLType` data: you cannot change the `XMLType` storage model.

As with other database objects, XML data is exported in the character set of the exporting server. During import, the data is converted to the character set of the importing server.

Oracle Data Pump has two command-line clients, `expdp` and `impdp`, that invoke Data Pump Export utility and Data Pump Import utility, respectively. The `expdp` and `impdp` clients use procedures provided in PL/SQL package `DBMS_DATAPUMP` to execute export and import commands, passing the parameters entered at the command-line. These parameters enable the exporting and importing of data and metadata for a complete database or subsets of a database.

The Data Pump Export and Import utilities (invoked with commands `expdp` and `impdp`, respectively) have a similar look and feel to the original Export (`exp`) and Import (`imp`) utilities, but they are completely separate.

Data Pump Export utility (invoked with `expdp`) unloads data and metadata into a set of operating system files called a *dump file set*. The dump file set can be imported only by the Data Pump Import utility (invoked using `impdp`).

Oracle XML DB supports export and import of `XMLType` tables and columns that store XML data, whether it is XML schema-based or not. If a table is XML schema-based, then it depends on the XML schema used to define its data. This XML schema can also have dependencies on SQL object types that are used to store the data, in the case of object-relational storage.

Therefore, exporting a user who has XML schema-based `XMLType` tables also exports the following:

- SQL objects types (if object-relational storage was used)
- XML schemas
- XML tables

You can export and import this data regardless of the `XMLType` storage format (object-relational or binary XML). However, Oracle Data Pump exports and imports XML data as text or binary XML data only. The underlying tables and columns used for object-relational storage of `XMLType` are thus not exported. Instead, they are converted to binary form and then exported as self-describing binary XML data.

> **Note:**
>
> Oracle Data Pump for Oracle Database 11g Release 1 (11.1) does not support the export of XML schemas, XML schema-based `XMLType` columns, or binary XML data to database releases prior to 11.1.

Regardless of the `XMLType` storage model, the format of the dump file is either text or self-describing binary XML with a token map preamble. By default, self-describing binary XML is used.

Since `XMLType` data is exported and imported as XML data, the source and target databases can use different `XMLType` storage models for that data. You can export data from a database that stores `XMLType` data one way and import it into a database that stores `XMLType` data a different way.

> **Note:**
>
> Do not use option `table_exists_action=append` to import more than once from the same dump file into an `XMLType` table, regardless of the `XMLType` storage model used. Doing so raises a unique-constraint violation error because rows in `XMLType` tables are always exported and imported using a unique object identifier.
>
> See *Oracle Database Utilities* for information about `table_exists_action`.

Export/Import Limitations for Oracle XML DB Repository

When you export or import tables that store data for Oracle XML DB Repository resources that are based on a registered XML schema, only that XML data is exported. The repository structure is lost during export, so that when these tables are imported they are not hierarchy-enabled.

You can export and import the `XMLType` tables that store the XML data for Oracle XML DB Repository resources that are based on a registered XML schema.

However, only the XML data is exported. The repository structure is lost during export. Relationships in the folder hierarchy, row-level security (RLS) policies, and path-index triggers are not exported for hierarchy-enabled tables. When these tables are imported, they are not hierarchy-enabled.

Export/Import Syntax and Examples

Guidelines and examples are presented for using commands `expdp` and `impdp` with `XMLType` data.

The examples presented here use the command-line commands `expdp` and `impdp`. After submitting such a command with a user name and command parameters, you are prompted for a password. The examples here do not show this prompting.

Export and import using Oracle Data Pump is described fully in *Oracle Database Utilities*.

- Performing a Table-Mode Export /Import
  Examples are presented of performing a table-mode export and a table-mode import, to and from a dump file, respectively.
- Performing a Schema-Mode Export/Import
  Examples here perform schema-mode exporting and importing. When performing a `Schema` mode export, if you have role `EXP_FULL_DATABASE`, then you can export a database schema, the database schema definition, and the system grants and privileges of that database schema.

Performing a Table-Mode Export /Import

Examples are presented of performing a table-mode export and a table-mode import, to and from a dump file, respectively.

An `XMLType` table has a dependency on the XML schema that was used to define it. Similarly, that XML schema has dependencies on the SQL object types that were created or specified for it. Importing an `XMLType` table requires the existence of the corresponding XML schema and SQL object types.

When a `TABLE` mode export is used, only the table related metadata and data are exported. To be able to import this data successfully, you must ensure that the relevant XML schema and object types have been created.

The examples here assume that you are using a database with the following features:

- A database with schema `user23`
- A table `user23.tab41` with an `XMLType` column stored as binary XML

- A directory object `dpump_dir`, for which `READ` and `WRITE` privileges have been granted to the user running `expdp` or `impdp`

Example 36-1 shows a table-mode export, specified using the `TABLES` parameter. It exports table `tab41` to dump file `tab41.dmp`.

> **Note:**
>
> In table mode, if you do not specify a schema prefix in the `expdp` command then the schema of the exporter is used by default.

Example 36-2 shows a table-mode import. It imports table `tab41` from dump file `tab41.dmp`.

If a table named `tab41` already exists at the time of the import then specifying `table_exists_action = append` causes rows to be appended to that table. Whenever you use parameter value `append` the data is loaded into new space; existing space is never reused. For this reason you might need to compress your data after the load operation.

> **See Also:**
>
> *Oracle Database Utilities*, for more information about Oracle Data Pump and its command-line clients, `expdp` and `impdp`

Example 36-1 Exporting XMLType Data in TABLE Mode

```
expdp system directory=dpump_dir dumpfile=tab41.dmp tables=user23.tab41
```

Example 36-2 Importing XMLType Data in TABLE Mode

```
impdp system tables=user23.tab41 directory=dpump_dir dumpfile=tab41.dmp
table_exists_action=append
```

Performing a Schema-Mode Export/Import

Examples here perform schema-mode exporting and importing. When performing a `Schema` mode export, if you have role `EXP_FULL_DATABASE`, then you can export a database schema, the database schema definition, and the system grants and privileges of that database schema.

The examples here assume that you are using a database with the following features:

- User `x4a` has created a table `po2`.
- User `x4a` has a registered XML schema, `ipo`, which created two ordered collection tables `item_oct2` and `sitem_nt2`.

User `x4a` creates table `po2` as shown in Example 36-3.

Table `po2` is then populated and exported, as shown in Example 36-4.

Example 36-4 exports all of the following:

- All data types that were generated during registration of XML schema `ipo`.
- XML schema `ipo`.
- Table `po2` and the ordered collection tables `item_oct2` and `sitem_nt2`, which were generated during registration of XML schema `ipo`.
- All data in all of those tables.

Example 36-5 imports all of the data in `x4a.dmp` to another database, in which the user `x4a` already exists.

Example 36-6 does the same thing as Example 36-5, but it also remaps the database schema from user `x4a` to user `quine`.

Example 36-6 imports all of the data in `x4a.dmp` (exported from the database schema of user `x4a`) into database schema `quine`. To remap the database schema, user `x4a` must have been granted role `IMP_FULL_DATABASE` on the local database and role `EXP_FULL_DATABASE` on the source database. `REMAP_SCHEMA` loads all of the objects from the source schema into the target schema.

> **Note:**
>
> If you import an XML schema into the same database that it was exported from, and if that XML schema is still registered with Oracle XML DB at the time of importing, do not use `remap_schema` unless you also specify `impdp` parameter `transform=oid:n`. See *Oracle Database Utilities* for information about parameter `transform`.

Example 36-3 Creating Table po2

```
CREATE TABLE po2 (po XMLType)
  XMLTYPE COLUMN po
  XMLSCHEMA "ipo.xsd"
  ELEMENT "purchaseOrder"
  VARRAY po.XMLDATA."items"."item"
    STORE AS TABLE item_oct2 ((PRIMARY KEY(NESTED_TABLE_ID, SYS_NC_ARRAY_INDEX$)))
      NESTED TABLE po.XMLDATA."shippedItems"."item" STORE AS sitem_nt2;
```

Example 36-4 Exporting XMLType Data in SCHEMA Mode

```
expdp x4a directory=tkxm_xmldir dumpfile=x4a.dmp
```

Example 36-5 Importing XMLType Data in SCHEMA Mode

```
impdp x4a directory=tkxm_xmldir dumpfile=x4a.dmp
```

Example 36-6 Importing XMLType Data in SCHEMA Mode, Remapping Schema

```
impdp x4a directory=tkxm_xmldir dumpfile=x4a.dmp remap_schema=x4a:quine
```