Oracle BIG Tablespace, BIGFILE

https://digitalksystems.com
https://www.youtube.com/channel/UCCGTnI9vvF_ETMhGUXGdFWw

What is a BIGFILE Tablespace?

- A **BIGFILE tablespace** is a type of Oracle tablespace that **contains only one single datafile**
- However, that single file can be **extremely large** — up to **128 TB (terabytes)** depending on your block size and Oracle version

Key Difference:

Type	Number of Datafiles	Max Size per Datafile	Typical Use
Smallfile Tablespace (Default)	Multiple datafiles	Usually up to 32 GB per file	Traditional usage
Bigfile Tablespace	Only 1 datafile	Can go up to 128 TB	Very large databases, simplifies management

Why Use BIGFILE Tablespaces

- **Simplified management -** Only one datafile per tablespace (easier for backup, restore, or resizing).
- **Useful with ASM or OMF -** Automatic Storage Management and Oracle Managed Files handle large single files efficiently.
- **Ideal for very large databases (VLDBs) -** Especially with large segment space and fewer datafiles.
  Note:
  - Don't use BIGFILE tablespaces if you prefer to manage multiple smaller datafiles manually.
  - Works best with **locally managed tablespaces** and **automatic segment space management (ASSM)**.

Creating a BIGFILE Tablespace

Example 1 – Basic creation

```
CREATE BIGFILE TABLESPACE app_data
DATAFILE '/u01/app/oracle/oradata/ORCL/app_data01.dbf'
SIZE 10G
AUTOEXTEND ON NEXT 1G MAXSIZE 100G
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO;
```

Explanation:

- BIGFILE → explicitly declares it as a bigfile tablespace.
- AUTOEXTEND → automatically increases size by 1G until 100G.
- EXTENT MANAGEMENT LOCAL and SEGMENT SPACE MANAGEMENT AUTO → recommended defaults for modern Oracle versions.

Example 2 – Create using OMF (Oracle Managed Files)

If OMF is enabled:

```
CREATE BIGFILE TABLESPACE app_data
DATAFILE SIZE 20G AUTOEXTEND ON NEXT 2G MAXSIZE UNLIMITED;
```

Oracle will automatically create the datafile in the default location defined by:

```
SHOW PARAMETER db_create_file_dest;
```

Altering a BIGFILE Tablespace

Make tablespace read-only:

```
ALTER TABLESPACE app_data READ ONLY;
```

Make tablespace read-write again:

```
ALTER TABLESPACE app_data READ WRITE;
```

Resize the datafile:

```
ALTER DATABASE DATAFILE '/u01/app/oracle/oradata/ORCL/app_data01.dbf' RESIZE 80G;
```

Add space (if AUTOEXTEND is off):

```
ALTER DATABASE DATAFILE '/u01/app/oracle/oradata/ORCL/app_data01.dbf' AUTOEXTEND ON NEXT 2G MAXSIZE 200G;
```

Rename datafile (when database is in MOUNT mode):

```
ALTER DATABASE RENAME FILE '/oldpath/app_data01.dbf' TO '/newpath/app_data01.dbf';
```

Dropping a BIGFILE Tablespace

Drop only tablespace (keep datafile)

```
DROP TABLESPACE app_data;
```

Drop tablespace and its datafile

```
DROP TABLESPACE app_data INCLUDING CONTENTS AND DATAFILES;
```

Drop tablespace and all its dependent objects (like users, segments, etc.)

```
DROP TABLESPACE app_data INCLUDING CONTENTS AND DATAFILES CASCADE CONSTRAINTS;
```

Check Tablespace Type (Bigfile or Smallfile)

```
SELECT tablespace_name, bigfile FROM dba_tablespaces;
```

If BIGFILE column = YES in the output, it's a Bigfile tablespace.

Check Tablespace Usage (Size, Used, Free in GB)

```
SELECT
   df.tablespace_name,
   ROUND(df.totalspace_gb, 2) AS total_gb,
   ROUND((df.totalspace_gb - fs.freespace_gb), 2) AS used_gb,
   ROUND(fs.freespace_gb, 2) AS free_gb
FROM
   (SELECT tablespace_name, SUM(bytes)/1024/1024/1024 AS totalspace_gb
    FROM dba_data_files
    GROUP BY tablespace_name) df,
   (SELECT tablespace_name, SUM(bytes)/1024/1024/1024 AS freespace_gb
    FROM dba_free_space
    GROUP BY tablespace_name) fs
WHERE df.tablespace_name = fs.tablespace_name
ORDER BY df.tablespace_name;
```