

10 OLTP and Core Database

This section describes the new OLTP and core database features.

Availability

True Cache

True Cache is an in-memory, consistent, and automatically managed cache for Oracle Database. It operates similarly to an Active Data Guard readers farm, except that True Cache is mostly diskless and is designed for performance and scalability, as opposed to disaster recovery. An application can connect to True Cache directly for read-only workloads. A general read-write Java application can also simply mark some sections of code as read-only, and the 23ai JDBC Thin driver can automatically send read-only workloads to configured True Caches.

Today, many Oracle users place a cache in front of the Oracle Database to speed up query response time and improve overall scalability. True Cache is a new way to have a cache in front of the Oracle Database. True Cache has many advantages including ease of use, consistent data, more recent data, and automatically managed cache.

[View Documentation](#)

Directory-Based Sharding Method

Directory-based sharding is a type of user-defined sharding in Oracle Globally Distributed Database where the location of data records associated with a sharding key is specified dynamically at the time of insert based on user preferences. The key location information is stored in a directory which can hold a large set of key values in the hundreds of thousands. With directory-based sharding, you have the freedom to move individual key values from one location to another, or make bulk movements to scale up or down, or for data and load balancing.

Directory-based sharding method improves the user-defined sharding model and provides linear scalability, complete fault isolation, and global data distribution for the most demanding applications.

[View Documentation](#)

Oracle Globally Distributed Database Raft Replication

Raft replication provides built-in replication for Oracle Globally Distributed Database without requiring configuration of Oracle GoldenGate or Oracle Data Guard. Raft replication is logical replication with consensus-based (RAFT) commit protocol, which enables declarative replication configuration and sub-second failover.

RAFT Replication helps simplify management, improves availability and SLA delivery, as well as optimizes hardware utilization for sharded database environments.

[View Documentation](#)

Automatic Data Move on Sharding Key Update

When you update the sharding key value on a particular row of a sharded table, the data with that key value might be mapped to a different partition or shard than where it currently resides. Oracle Globally Distributed Database now handles moving the data to the new location, whether it is in a different partition on the same shard or on a different shard.

This feature makes data movement between partition or shards seamless when sharding key value update occurs due to various reasons, for example, a move to another country or change in roles.

[View Documentation](#)

Automatic Transaction Quarantine

System MONitor (SMON) is a background process responsible for transaction recovery. Transaction Quarantine can now automatically quarantine the recovery of problematic transactions while keeping the database open, allowing SMON to proceed with the recovery of the other transactions. Alerts and diagnostic information are provided to the DBA or operator so that they can review and resolve the quarantine while other database operations continue unaffected.

The benefit of transaction quarantining is increased fault tolerance and high availability of the database. The database stays up and running and continues processing transactions while the quarantine is being resolved.

[View Documentation](#)

Creating Immutable Backups Using RMAN

RMAN is now compatible with immutable OCI Object Storage using locked retention rules, which prevents deletion or modification of backups.

To help organizations meet ransomware protection or strict regulatory requirements for record management and retention, RMAN now prevents anyone, even an administrator, from deleting or modifying backups in OCI Object Storage.

[View Documentation](#)

Fine-Grained Refresh Rate Control For Duplicated Tables

Oracle Globally Distributed Database enables refresh rate control for individual duplicated tables. Each duplicated table can have a separate refresh rate which is defined either at its creation or by the `ALTER TABLE` statement.

This feature helps optimize the use of resources by customization of refresh rates for individual duplicated tables.

[View Documentation](#)

Global Partitioned Index Support on Subpartitions

Globally Distributed Database allows a global partitioned index on the sharding key when the sharded table is sub-partitioned. You can create a primary key/unique indexes on sharded tables that are composite partitioned without having to include sub-partition keys.

The benefit of this feature is that it removes the restriction on the primary key columns when the sharded table is sub-partitioned, as in the composite sharding method.

[View Documentation](#)

JDBC Support for Split Partition Set

This feature enables the Java connection pool (UCP) to receive ONS events about data in a chunk being split and moved across partition sets, and then update the sharding topology appropriately.

This feature furnishes high availability to Java applications using Sharded databases.

[View Documentation](#)

Managing Flashback Database Logs Outside the Fast Recovery Area

In previous releases, you could store flashback database logs only in the fast recovery area. Now you can optionally designate a separate location for flashback logging. For example, if you have write-intensive database workloads, then flashback database logging can slow down the database if the fast recovery area is not fast enough. In this scenario, you can now choose to write the flashback logs to faster disks. Using a separate destination also eliminates the manual administration to manage the free space in the fast recovery area.

Managing flashback database logs outside the fast recovery area lowers the operational costs related to space management and guarantees the best performance for workloads that are typically impacted by flashback logging on traditional storage.

[View Documentation](#)

New Duplicated Table Type - Synchronous Duplicated Table

Oracle Globally Distributed Database introduces a new kind of duplicated table that is synchronized on the shards ‘on-commit’ on the shard catalog. The rows in a duplicated table on the shards are synchronized with the rows in the duplicated table on the shard catalog when the active transaction performing DMLs on the duplicated tables in the shard catalog is committed.

This feature enables efficient and absolute data consistency and synchronization for duplicated tables, across all shards at all times.

[View Documentation](#)

New Partition Set Operations for Composite Sharding

For Oracle Globally Distributed Database sharded databases using the composite sharding method, two new `ALTER TABLE` operations enhance partition set maintenance. Previously, partition set operations did not support specifying tablespace sets for child and reference-partitioned tables that are affected due to add and split partition set operations. `MOVE PARTITIONSET` lets you move a whole partition set from one tablespace set to another, within the same shardspace. `MODIFY PARTITIONSET` lets you add values to the list of values of a given partition set.

These new operations enhance resharding capability. `MOVE PARTITIONSET` gives you the control to move all subpartitions of a given table to another tablespace set, within a given shardspace. You can also specify separate tablespace sets for LOBs and subpartitions. `MODIFY PARTITIONSET` extends the add list values feature of partitions to partition sets.

[View Documentation](#)

Oracle Data Pump Adds Support for Sharding Metadata

Oracle Data Pump adds support for sharding DDL in the API `dbms_metadata.get_ddl()`. A new transform parameter, `INCLUDE_SHARDING_CLAUSES`, facilitates this support. If this parameter is set to `true`, and the underlying object contains it, then the `get_ddl()` API returns sharding DDL for create table, sequence, tablespace and tablespace set. To prevent sharding attributes from being set on import, the default value for `INCLUDE_SHARDING_CLAUSES` is set to `false`.

Oracle Data Pump supports sharding migration with support for sharding DDL. You can migrate sharding objects to a target database based on source database shard objects.

[View Documentation](#)

Oracle Globally Distributed Database Coordinated Backup and Restore Enhancements

Coordinated backup and restore functionality in Oracle Globally Distributed Database has been extended to include the following:

- Enhanced error handling and diagnosis for backup jobs
- Improved automation of sharded database restore
- Support for running RMAN commands from GDSCTL
- Support for using different RMAN recovery catalogs for different shards
- Encryption of backup sets
- Support for additional backup destinations: Amazon S3, Oracle Object Storage, and ZDLRA

The benefits of this functionality are:

- Easily diagnose problems in backup jobs
- Backups sets can be encrypted so that the data is secured
- Support for additional destinations other than on-disk storage

- Support for different RMAN catalogs and destinations to abide by data residency requirements

[View Documentation](#)

PL/SQL Function Cross-Shard Query Support

PL/SQL functions are enhanced with the keyword `SHARD_ENABLE` to allow these functions to be referenced in Oracle Globally Distributed Database cross-shard queries. With the new keyword, the query optimizer takes the initiative to push the execution of the PL/SQL function to the shards.

This feature significantly improves performance for PL/SQL functions in sharded database environments.

[View Documentation](#)

Parallel Cross-Shard DML Support

The Oracle Globally Distributed Database query coordinator runs cross-shard updates and inserts in parallel on multiple shards.

This feature improves cross-shard DML performance by running updates and inserts in parallel rather than serially.

[View Documentation](#)

Pre-Deployment Diagnostic for Oracle Globally Distributed Database

While processing `GSDSCTL ADD SHARD`, `ADD GSM`, and `DEPLOY` commands, Oracle Globally Distributed Database runs a series of checks to make sure that there is no potential environmental issue.

This feature proactively avoids common pitfalls to reduce time taken to complete a sharded database deployment.

[View Documentation](#)

Priority Transactions

If a transaction does not commit or roll back for a long time while holding row locks, it can potentially block other high-priority transactions. This feature allows applications

to assign priorities to transactions and for administrators to set timeouts for each priority. The database will automatically roll back a lower priority transaction and release the row locks held if it blocks a higher priority transaction beyond the set timeout, allowing the higher priority transaction to proceed.

Priority Transactions reduces the administrative burden while also helping to maintain transaction latencies and SLAs on higher priority transactions.

[View Documentation](#)

RMAN Backup Encryption Algorithm Now Defaults to AES256

RMAN encrypted backups now default to `AES256` encryption algorithm. RMAN will continue to support restore using existing backups created with `AES128` or `AES192` encryption algorithms. You may also choose to create new backups using `AES128` by changing the default `AES256` setting. This default change applies to `BACKUP` `BACKUPSET` command and the `ALLOCATE CHANNEL` command.

To strengthen the security of encrypted backups from being decrypted by malicious users, RMAN encrypted backups now default to the `AES256` encryption standard.

[View Documentation](#)

RMAN Operational, Diagnostics, and Upgrade Enhancements

RMAN now includes easier standby database registration for Oracle Data Guard, better fault tolerance and optimization for Oracle Real Application Cluster (Oracle RAC), enhanced diagnosability which automatically gathers information to help identify issues, and updates to mitigate bottlenecks and pause sessions during recovery catalog upgrades.

RMAN operations are now easier and more resilient for highly available Oracle environments with less complex backup registration, automatic diagnostic gathering, and fewer failures when performing maintenance activities.

[View Documentation](#)

Simplified Database Migration Across Platforms Using RMAN

Using RMAN to migrate databases across different operating system platforms has been streamlined and includes support for databases encrypted with Transparent Data Encryption (TDE) and multi-section backups. New command options allow

existing RMAN backups to be used to transport tablespaces or pluggable databases to a new destination database with minimal downtime.

Migrations using RMAN are now easier, faster, and require fewer steps to execute. The new capabilities enable a simple and straightforward migration process, minimizing downtime for your applications, reducing risk, and increasing productivity.

[View Documentation](#)

Support for Oracle Database Version Specific RMAN SBT Library

The Oracle Home directory now includes the database version compatible libraries (`SBT_LIBRARY`) for Zero Data Loss Recovery Appliance, OCI Object Storage and Amazon S3. You can now configure RMAN to directly access libraries from the Oracle Home directory using an alias. For example, if the backup destination is OCI Object Storage, you only have to specify the alias name `oracle.oci` for the `SBT_LIBRARY` parameter. When RMAN attempts to backup to Object Storage, it uses the specified alias to access the SBT library used for backup cloud service from the Oracle home directory.

The RMAN storage libraries are now included with the database, eliminating the need to download and install additional software and ensuring that you have all the necessary components to immediately start backing up and restoring from Zero Data Loss Recovery Appliance, OCI Object Storage, or Amazon S3.

[View Documentation](#)

Blockchain

Blockchain Table User Chains

Earlier versions of blockchain tables supported only system chains. A system chain (one of the 32 chains per instance) is randomly chosen by Oracle for every new row inserted into a blockchain table.

A user chain is a chain of rows based on a set of up to three user-defined columns of type `NUMBER`, `CHAR`, `VARCHAR2`, and `RAW`. For example, consider a blockchain table created for tracking banking transactions (withdrawals, deposits, transfers) associated with various accounts. Assume there is a column called `ACCOUNTNO` in the blockchain table for account numbers. Each transaction inserts a new entry into this blockchain table for some account number. A user chain can be associated with every unique value in `ACCOUNTNO`. If there are a total of 100 different account numbers, there can be at most 100 user chains. You can then run a verification procedure only on a chain for a

specific `ACCOUNTNO`, providing greater data isolation. This feature allows you to create user chains for rows in blockchain tables based on version columns even if they are split across system chains.

Multiple user chains increase the flexibility of applying blockchain tables and their verification procedures to make it easier to leverage tamper-resistant tables in your applications.

[View Documentation](#)

Blockchain Table Row Versions

The blockchain table row version feature allows you to have multiple historical versions of a row that is maintained within a blockchain table corresponding to a set of user-defined columns. A view `<bctable>_last$` on top of the blockchain table allows you to see just the latest version of a row.

This feature allows you to guarantee row versioning when using tamper-resistant blockchain tables in your application.

[View Documentation](#)

Blockchain Table Log History

Flashback Data Archive History tables are now blockchain tables. This feature allows changes to one or more regular user tables to be tracked in a blockchain table maintained by the Oracle database as part of the Flashback Data Archive. Each change in a regular table will be added to the blockchain log history table as a separate row within a cryptographic hash chain maintained by the blockchain table. You can verify the data and chain integrity in a Flashback Data Archive Blockchain Log History table using the built-in verification procedures (`DBMS_BLOCKCHAIN_TABLE.verify_rows`) or through an external verification, including a continuous verification process illustrated by a sample provided in <https://github.com/oracle/blockchain-table-samples>.

This feature allows you to record changes to regular user tables in a cryptographically secure and verifiable fashion.

[View Documentation](#)

Add and Drop User Columns in Blockchain and Immutable Tables

This feature allows evolution of Blockchain and Immutable Tables, namely it allows columns to be added and dropped while maintaining the current data, including that in dropped columns for continuity of crypto-hash chains.

As applications evolve you may need to modify existing tables by adding or dropping columns. In this release, you can easily add or drop columns in previously created Blockchain or Immutable tables. Any rows prior to a column deletion will maintain the data in these columns in order to preserve the integrity of the crypto-hash chains and allow the verification procedures to work across the entire table.

[View Documentation](#)

Blockchain Table Countersignature

You can request a database countersignature at the time of signing a row. In addition to recording the countersignature and its metadata in the row, the countersignature and the `signed_bytes` are returned to the caller. The caller can then save the countersignature and `signed_bytes` in another data store, such as Oracle Blockchain Platform, for non-repudiation purposes.

A countersignature can provide user additional guarantees that data has been securely stored in the blockchain table.

[View Documentation](#)

Blockchain Table Delegate Signer

A delegate is an alternate user who's allowed to sign rows inserted by the primary user. This feature allows a delegate to sign rows in an immutable or blockchain table on behalf of another user. A delegate's signature is accepted only if the signature can be verified using the public key in the delegate's certificate, which has been added to the dictionary table.

A delegate signer can be used when users are not able to sign the rows they created and they trust their delegate.

[View Documentation](#)

New Special Privilege Required to Set Long Idle Retention Times for Blockchain and Immutable Tables

Blockchain or immutable tables with idle retention set to a sufficiently large value cannot be dropped until the newest row of the table becomes very old. This limits the ability to drop the blockchain/immutable table if necessary to prevent a disk space exhaustion attack. Hence, the operation of setting a table's idle retention to a large value is restricted to privileged users via a grant of a new `TABLE RETENTION` system privilege. The idle retention threshold, which specifies when to require the new privilege `BLOCKCHAIN_TABLE_RETENTION_THRESHOLD`, is configurable.

Ability to create blockchain or immutable tables with long retention times and inserting large amounts of data that can not be deleted could potentially be a vector for a denial of service attack via disk space exhaustion. To reduce this risk, the special privilege has been introduced. Only users granted this privilege can set idle retention above the configurable threshold level.

[View Documentation](#)

Database Architecture

Lock-Free Reservation

The Lock-Free Reservation feature enables concurrent transactions to proceed without being blocked on updates of heavily updated rows. A Lock-Free Reservation is held on the row, instead of locking the row. The Lock-Free Reservation verifies if the updates can succeed and defers the updates until the transaction commit time.

Lock-Free Reservation improves the end user experience, and concurrency, in transactions.

[View Documentation](#)

Wide Tables

The maximum number of columns allowed in a database table or view has been increased to 4096. This feature allows you to build applications that can store attributes in a single table with more than the previous 1000-column limit. Some applications, such as Machine Learning and streaming IoT application workloads, may require the use of de-normalized tables with more than 1000 columns.

You now have the ability to store a larger number of attributes in a single row which for some applications may simplify application design and implementation.

[View Documentation](#)

Consolidated Service Backgrounds for Oracle Instance

We are introducing a new set of service processes which execute database service actions.

Service actions are responsible for maintenance tasks, parallel tasks and brokered tasks, consolidated tasks and many more. These were performed by dedicated processes in the database before. The new background scheduler group processes can execute any of these service actions, thus providing consolidation of background service actions.

[View Documentation](#)

Improve Performance and Disk Utilization for Hybrid Columnar Compression

Enhancements to the compression algorithms for Hybrid Columnar Compression (HCC) include improvements for faster compression and decompression speeds, as well as better compression ratios for newly created HCC compressed tables or for existing HCC compressed tables that are rebuilt. The exact benefits can vary based on the data and the chosen compression level.

This feature improves an application's workload performance while reducing database storage utilization.

[View Documentation](#)

System Timezone Autonomy for Pluggable Databases

Oracle Multitenant enables an Oracle Database to consolidate multiple pluggable databases as self-contained databases, improving resource utilization and database management. In addition to providing a fully centrally managed database environment with identical, global time zone settings for all pluggable databases (impacting `SYSDATE` and `SYSTIMESTAMP`), pluggable databases can now control their time zone settings independently. You can control the time zone setting, including internal processes and operations, or only on a user-visible level.

The ability to control the time zone behavior for `SYSDATE` and `SYSTIMESTAMP` on a pluggable database level increases to self-containment of individual databases in a multitenant environment and enhances your consolidation capabilities of independent databases.

[View Documentation](#)

Unrestricted Direct Loads

Prior to this feature, after a direct load and prior to a commit, queries and additional DMLs were not allowed on the same table for the same session or for other database sessions. This enhancement allows the loading session to query and perform DML on the same table that was loaded. Rollback to a savepoint is also supported.

This feature removes the restrictions that you may have encountered when loading and querying data. Potentially improving the performance of your applications in areas such as Data Warehousing and complex batch processing.

[View Documentation](#)

General

Unrestricted Bulk Transactions

Oracle Database allows DML statements (`INSERT`, `UPDATE`, `DELETE`, and `MERGE`) to be executed in parallel by breaking the DML statements into mutually exclusive smaller tasks. Executing DML statements in parallel can make DSS queries, batched OLTP jobs, or any larger DML operations faster. However, parallel DML operations had a few transactional limitations.

This includes a limitation that restricted transactions with multiple per-table parallel DMLs. This means that once an object is modified by a parallel DML statement, that object cannot be read or modified by later statements of the same transaction. This enhancement removes this limitation, enabling users to run parallel DMLs, and any combination of statements like queries, serial DML, and parallel DML on the same object, within the same transaction.

For users, this simplifies and speeds up data loading and analytical processing by making full use of Oracle Database's parallel execution and parallel query capabilities.

[View Documentation](#)

ACFS Auto Resize Variable Threshold

ACFS auto resize now allows you to configure the threshold percentage for your file system automatic resize.

A more flexible threshold is now available for your file systems auto resize. Previously, the threshold was fixed to 10%. Now, you can customize to your specific use case needs.

[View Documentation](#)

ACFS Cross Version Replication

ACFS replication now allows for primary clusters to replicate to standby cluster on a previous or older release.

This feature will provide flexibility in replication configurations, providing ample time for upgrading and lifecycle maintenance.

[View Documentation](#)

ACFS Encryption Migration from OCR to OKV

ACFS Encryption now allows you to migrate from OCR to OKV.

This feature allows for a centralized point for key management using Oracle Key Vault.

[View Documentation](#)

ACFS Replication Password-less SSH Setup Tool

A new tool provides users the ability to configure SSH keys management for ACFS Replication.

Users can now avoid the repetitive, error-prone process of SSH keys management, setup, and configuration with this new tool. The tool makes the ACFS replication setup process more efficient and easier.

[View Documentation](#)

ACFS Replication Switchover

A new command, `acfsutil repl switchover`, provides a coordinated failover. However, if ACFS cannot establish contact the replication primary site, the command will fail.

Enhanced flexibility in ACFS replication management is now available with the addition of this new command.

[View Documentation](#)

ACFS SSH-less Replication

This feature provides an alternative transport choice for ACFS Replication which eliminates the need to maintain ssh-related host and user keys.

Users now have an alternative to ssh, including network data transfer, authentication between replication storage locations, encryption of the data stream, and a facility for executing remote commands.

[View Documentation](#)

ACFS Snapshots RMAN Sparse Backup and Restore

You can now back up and restore PDB snapshot copies on ACFS.

Backing up and restoring PDB snapshot copies on ACFS, provides the space-efficient storage that is inherent of ACFS Snapshots.

[View Documentation](#)

ACFS Sparse Backup and Restore of Snapshots

The `acfsutil snap duplicate` command can now generate a backup of an entire ACFS file systems and its snapshots, while preserving its sparseness.

You can now apply a full backup to another location while retaining the original sparseness. You can now replicate an entire ACFS file system and its snapshot tree with this new functionality.

[View Documentation](#)

ACFSutil plogconfig Log Files Wrapping Info

ACFSutil plogconfig offers you a way to manage persistent logging configuration settings. `acfsutil plogconfig -q` will now offer you additional information on whether the logs have wrapped or not. You can also get this information with `acfsutil plogconfig -w`, which will offer only this information and not all the comprehensive information offered by `acfsutil plogconfig -q`.

Further information regarding persistent logging is now available, hence enhancing the experience in the realm of diagnosability.

[View Documentation](#)

Automatic Parallel Direct Path Load Using SQL*Loader

The SQL*Loader client can automatically start a parallel direct path load for data without dividing the data into separate files and starting multiple SQL*Loader clients. This feature prevents fragmentation into many small data extents. The data doesn't need to be resident on the database server. Cloud users can employ this feature to load data in parallel without having to move data on to the cloud system if there is sufficient network bandwidth.

SQL*Loader can load data faster and easier into Oracle Database with automatic parallelism and more efficient data storage.

[View Documentation](#)

BIGFILE Default for SYSAUX, SYSTEM, and USER Tablespaces

Starting with Oracle Database 23ai, BIGFILE functionality is the default for `SYSAUX`, `SYSTEM`, and `USER` tablespaces.

A bigfile tablespace is a tablespace with a single, but large datafile. Traditional small file tablespaces, in contrast, typically contain multiple datafiles, but the files cannot be as large. Making `SYSAUX`, `SYSTEM` and `USER` tablespaces bigfile by default will benefit large databases by reducing the number of datafiles, thereby simplifying datafile, tablespace and overall global database management for users.

[View Documentation](#)

Bigfile Tablespace Shrink

This feature supplies the capability to reliably shrink a bigfile tablespace.

In earlier releases, organizations may have found that the datafile of a bigfile tablespace grew larger despite the actual used space being much smaller. This could happen after a user dropped segments/objects in the tablespace, but was not able to use datafile resize to recover the freed space due to the location of the data in the datafile.

By using Bigfile Tablespace Shrink, organizations can now reliably shrink a bigfile tablespace to close to the sum of the size of all objects in that tablespace, optimizing storage and reducing costs.

[View Documentation](#)

CEIL and FLOOR for DATE, TIMESTAMP, and INTERVAL Data Types

You can now pass `DATE`, `TIMESTAMP`, and `INTERVAL` values to the `CEIL` and `FLOOR` functions. These functions include an optional second argument to specify a rounding unit. You can also pass `INTERVAL` values to `ROUND` and `TRUNC` functions.

These functions make it easy to find the upper and lower bounds for date and time values for a specified unit.

[View Documentation](#)

Centralized Configuration Providers

Database clients can securely pull application configuration data from Azure or OCI Cloud. The store can contain data such as application connection descriptors and tuning parameters.

Central configuration makes application management and scaling easier. It fits well with architectures such as microservices and serverless deployments.

[View Documentation](#)

LONG_TO_LOB, a Data Pump import TRANSFORM parameter

The Oracle Data Pump `TRANSFORM` parameter now includes the `LONG_TO_LOB` option to migrate all `LONG` and `LONG RAW` data types.

Using this transform enables you to migrate the deprecated `LONG` and `LONG RAW` data types by transparently and automatically converting them to `CLOB` and `BLOB` data types. This parameter changes all `LONG` data types to `CLOB` and all `LONG RAW` data types to `BLOB`.

The syntax is `TRANSFORM = LONG_TO_LOB:[Y|N]`. The default is `N`.

[View Documentation](#)

Oracle Data Pump Filters GoldenGate ACDR Columns from Tables

The ACDR feature of Oracle GoldenGate adds hidden columns to tables to resolve conflicts when the same row is updated by different databases using active replication. GoldenGate can also create a "tombstone table," which records interesting column values for deleted rows. Oracle Data Pump can exclude the hidden columns and the tombstone tables by setting a new import transform parameter, `OMIT_ACDR_METADATA`.

Oracle Data Pump enhances migration flexibility. It can migrate data from an Oracle GoldenGate ACDR (automatic conflict detection and resolution) environment to a non-ACDR environment by excluding the GoldenGate ACDR metadata during import.

[View Documentation](#)

PDB Snapshot Carousel ACFS Support

Oracle ACFS now supports PDB Snapshot Carousel, which allows you to maintain a library of PDB Snapshots.

Oracle Database files stored on Oracle ACFS file systems can now leverage PDB Snapshot Carousel in conjunction with ACFS snapshot technology.

[View Documentation](#)

SQL*Loader Supports SODA (Simple Oracle Document Access)

SQL*Loader now supports Simple Oracle Document Access (SODA). You can insert, append, and replace external documents into SODA collections in Oracle Database applications by using the SQL*Loader utility in both control file and express modes.

SQL*Loader support for Simple Oracle Document Access (SODA) makes it easier and faster to load schema-less JSON or XML-based application data into Oracle Database.

[View Documentation](#)

Manageability and Performance

Advanced LOW IOT Compression

An index-organized table (IOT) is a table stored in a variation of a B-tree index structure where rows are ordered by primary key. IOTs are useful because they provide fast random access by primary key without duplicating primary key columns in two structures – a heap table and an index. In earlier releases, IOTs only supported Oracle's prefix compression (formerly called key compression), which required additional analysis and had the possibility of negative compression (where the overhead of compression outweighed the compression benefits).

Advanced LOW IOT Compression allows you to reduce the overall storage for Oracle Databases.

[View Documentation](#)

Automatic SecureFiles Shrink

Automatic SecureFiles Shrink selects SecureFiles LOB segments based on a set of criteria and executes the free space shrink operation in the background for the selected segments. With Automatic SecureFiles Shrink, the shrink operation happens transparently in small and gradual steps over time while allowing DDL and DML statements to execute concurrently. In the manual method, you must decide on which LOB segments to shrink using tools like Segment Advisor and use a DDL statement to execute the shrink operation. The manual method may not be feasible for very large LOB segments because it is time-consuming.

Automatic SecureFiles Shrink simplifies administrator duties and saves time due to the automation of this process.

[View Documentation](#)

Automatic Storage Compression

Organizations use Hybrid Columnar Compression for space saving and fast analytics performance. However, the compression and decompression overhead of Hybrid Columnar Compression can affect direct load performance. To improve direct load performance, Automatic Storage Compression enables Oracle Database to direct load

data into an uncompressed format initially, and then gradually move rows into Hybrid Columnar Compression format in the background.

Automatic Storage Compression improves direct load performance, while keeping the advantages of Hybrid Columnar Compression, including space savings and fast analytics performance.

[View Documentation](#)

DBCA Silent Options Changes

DBCA silent mode options changes in various functionality

DBCA silent command line options integrate smoothly with custom scripts and provide user-friendly errors.

[View Documentation](#)

Enhanced Query History Tracking and Reporting

Enhanced Query History Tracking and Reporting lets you track and report on a more complete history of user-issued queries than is available in previous releases. This feature provides you with greater capability to track user-initiated queries within a session. It includes non-parallel queries with less than five seconds of execution time, which are not tracked with Real-time SQL Monitoring unless tracking is forced by a hint. Each user can access and report on their own current session history. SYS users and DBAs can view and get query history reports for all current user sessions and can also turn this functionality on or off. Reporting is configurable, with options for selecting the reporting scope and detail level.

Enhanced Query History Tracking and Reporting allows application developers and development operations (DevOps) personas to get detailed insight into the queries that execute on your databases. This insight allows you to better manage and optimize your applications.

[View Documentation](#)

Fast Ingest (Memoptimize for Write) Enhancements

This feature adds enhancements to Memoptimize Rowstore Fast Ingest with support for partitioning, compressed tables, fast flush using direct writes, and direct In-

Memory column store population support. These enhancements make the Fast Ingest feature easier to incorporate in more situations where fast data ingest is required.

This feature helps Oracle Database provide better support for applications requiring fast data ingest capabilities. Data can be ingested and then processed all in the same database. This reduces the need for special loading environments, and thus reduces complexity and data redundancy.

[View Documentation](#)

Improved Performance of LOB Writes

You can experience improved read and write performance for LOBs due to the following enhancements:

- Multiple LOBs in a single transaction are buffered simultaneously. This improves performance when you use switch between LOBs while writing within a single transaction.
- Various enhancements, such as acceleration of compressed LOB append and compression unit caching, improve the performance of reads and writes to compressed LOBs.
- The input-output buffer is resized based on the input data for large writes to LOBs with the NOCACHE option. This improves the performance for large direct writes, such as writes to file systems on DBFS and OFS.

This feature adds a host of improvements to accelerate SecureFiles writes for JSON document-based applications, for write calls issued by a database file system, and also for LOB workloads where the underlying data is compressed for storage savings.

[View Documentation](#)

Improved System Monitor (SMON) Process Scalability

Queries can require large amounts of temporary space and some temporary space operations run in critical background processes, like the System Monitor (SMON) process. SMON is responsible for cleaning up temporary segments that are no longer in use. SMON checks regularly to see whether it is needed, and other processes can call SMON. Temporary space management can affect SMON's scalability for other critical actions. This new enhancement instead uses the Space Management Coordinator (SMCO) process so that the responsibility of managing temporary space is offloaded from SMON, thereby improving its scalability.

This feature improves the overall scalability of the SMON process, particularly in a multitenant Oracle RAC cluster.

[View Documentation](#)

Migrate BasicFile LOBs Using the SecureFiles Migration Utility

You can use the SecureFiles Migration Utility to simplify the migration and compression of BasicFile LOB segments to SecureFiles LOB segments.

Earlier it was challenging to decide which BasicFile LOBs to migrate to SecureFile LOBs, and whether or not to compress the LOBs, especially considering that organizations often have many databases, with a large numbers of schemas, tables, and segments. SecureFiles Migration Utility automates several steps that were earlier performed manually. It also generates several reports that help you decide which BasicFile LOBs you want to migrate and compress.

[View Documentation](#)

Ordered Sequence Optimizations in Oracle RAC

The processing of ordered sequences in Oracle Real Application Clusters (Oracle RAC) has been optimized to provide better performance without requiring manual changes, ensuring a strict sequence order.

Applications using ordered sequences in Oracle RAC environments will benefit from improved performance and scalability.

[View Documentation](#)

Pluggable Database Support in Oracle Data Guard Environments

There is now a pluggable database configuration in a Data Guard environment using Database Configuration Assistant (DBCA).

A command line based silent mode option is available for configuring pluggable databases (PDBs) in Data Guard environments.

[View Documentation](#)

Refreshable PDBs in DBCA

Database Configuration Assistant (DBCA) allows you to clone a remote Pluggable database (PDB) as a refreshable PDB. When a PDB is created as refreshable, the changes of the source PDB will periodically propagate to the refreshable PDB. The refreshable PDB can be configured to refresh manually or automatically during creation.

A DBCA-based graphical user interface or scripted silent mode for cloning a remote refreshable PDB reduces many commands needed to create a remote refreshable PDB clone ensuring a faster and more reliable cloning of PDBs.

[View Documentation](#)