Oracle® Database
Learning Database New Features

Release 21c

F36089-26

March 2024

ORACLE®

Oracle Database Learning Database New Features, Release 21c

F36089-26

Contents

3   Big Data and Data Warehousing Solutions

4   Database Upgrade and Utilities

5   Management Solutions

6   Performance and High Availability

7   Security Solutions

8   New Features in 21c Release Updates

1

Introduction

- About the Guide
- Database Features and Licensing App
- Try It On LiveLabs
- Oracle Database 21c Interactive Architecture Diagram

About the Guide

This guide contains descriptions of all the Oracle Database Release 21c new features and links to documentation and labs.

You can listen to the presentation from Andy Mendelsohn, Executive VP of Oracle Database Server Technologies, about the core Oracle Database Breakthrough Innovations.
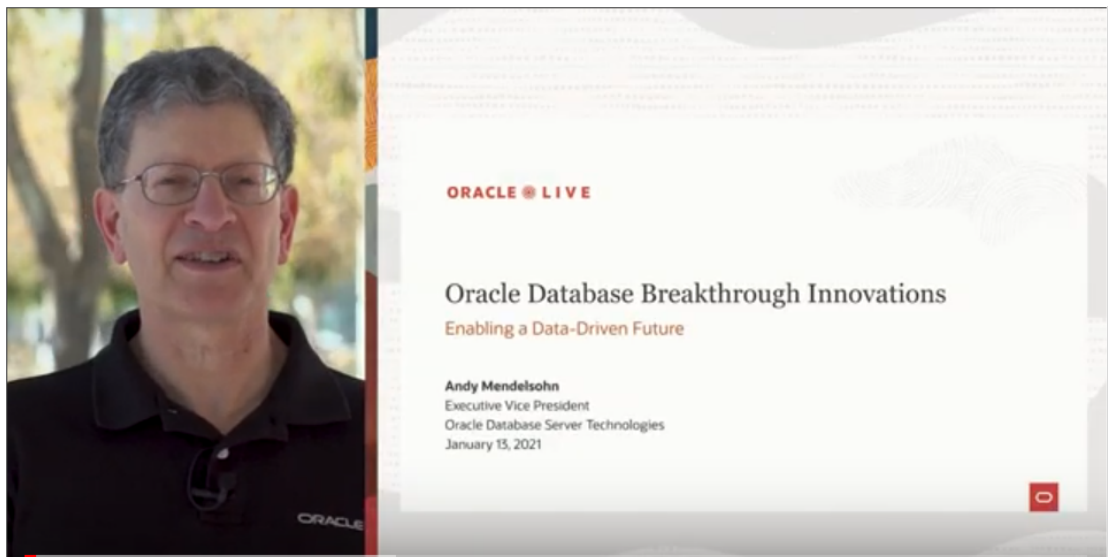
You can read VP of Database Product Management, William Hardie's blog about the core Oracle Database 21c features by visiting https://blogs.oracle.com/database/introducing-oracle-database-21c.

December 8, 2020

Introducing Oracle Database 21c

William Hardie
VICE PRESIDENT

The latest Innovation release of the world's most popular database, Oracle Database 21c, is now generally available "cloud first" in the Oracle Cloud Database Service Virtual Machine (for RAC and single instance) and Bare Metal Service (single instance). It's also available in the Autonomous Database Free Tier Service in Ashburn (IAD), Phoenix (PHX), Frankfurt (FRA) and London (LHR) regions. General availability of Oracle Database 21c for on-prem platforms (including Exadata, Linux and Windows) will follow along in 2021.

Database Features and Licensing App

Use the Database Features and Licensing app to view feature availability across Oracle Database releases and to see what features are new in Oracle Database 21c.

Try It On LiveLabs

Explore many of the new features available in the Database 21c New Features workshop in LiveLabs.

Oracle Database 21c Interactive Architecture Diagram

For more information about the technical architecture of Oracle Database 21c, visit https://docs.oracle.com/en/database/oracle/oracle-database/21/dbiad/index.html.

Oracle Database 21c Technical Architecture

This interactive diagram shows the Oracle Database 21c technical architecture.
Show PDF | Show Instructions

First Previous Next Slide 1 of 29
Database Server
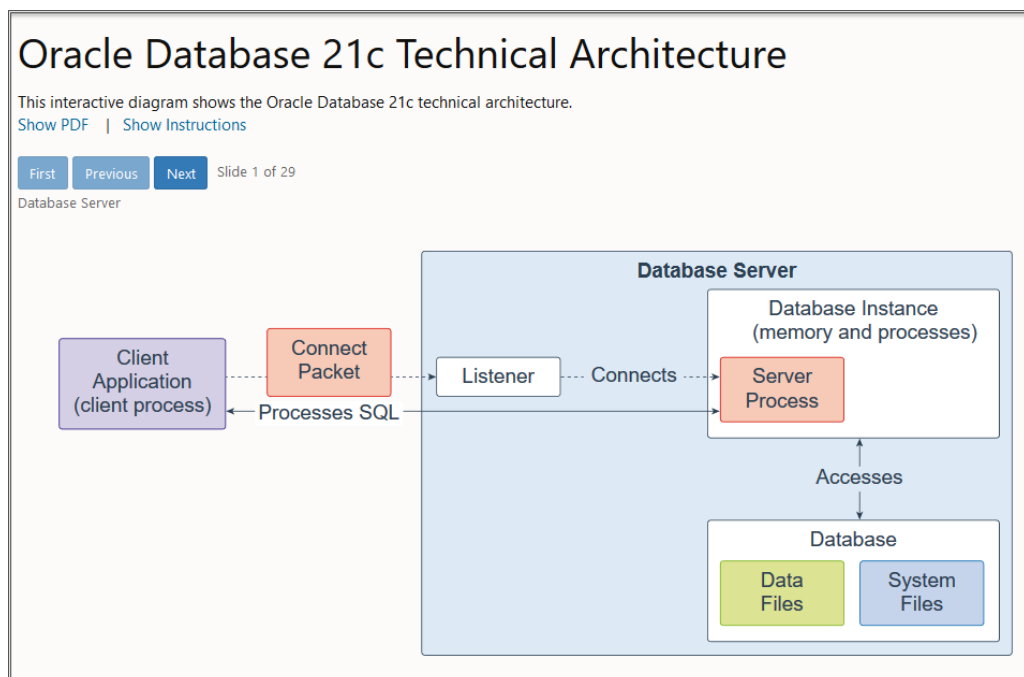
2

Application Development

- Application Express
- Globalization
- Java in Oracle Database
- JDBC
- JSON Document Store
- Net Services
- Oracle Call Interface
- Oracle Data Provider for .NET (ODP.NET)
- PL/SQL
- SQL
- Text

Application Express

- Application Backups
- Dark Mode
- Export App as Zip
- Faceted Search
- Friendly URL Syntax
- Mega Menus
- Native PDF Printing
- New Data Upload
- New Form Region
- New Team Development
- Oracle JET Charts
- Popup List of Values (LOV)
- Redwood UI
- Remote Application Deployment
- REST Read / Write Enhancements
- Shared List of Values (LOV)
- SQL Workshop Now Supports Simple Oracle Document Access (SODA)
- Universal Theme Enhancements

Application Backups

Oracle Application Express automatically backs up modified applications as a part of daily maintenance.

Related Topics

- Oracle® Application Express App Builder User's Guide

Dark Mode

Decrease eye strain by enabling Dark Mode.

The development environment can now render with a darker color scheme, which reduces eye strain and is especially helpful for developing late into the night.

Related Topics

- Oracle® Application Express App Builder User's Guide

Export App as Zip

You can now export your Application Express apps as a single zip file, which can facilitate integration with GIT / SVN and simplify incremental deployment. The zip file mirrors the directory structure of the APEXExport -split utility and splits your application, pages, and shared components into separate files for easy management.

Related Topics

- Oracle® Application Express App Builder User's Guide

Faceted Search

Faceted search, or faceted navigation, provides an easy-to-use interface to narrow the results of a report and quickly locate specific data.

Like other declarative data-driven components in Application Express, you can create a faceted search page on an existing table, an arbitrary SQL query, an ORDS REST-Enabled SQL Service, or virtually any REST endpoint. Using an intelligent auto-discovery and scoring algorithm, recommended facets are automatically created based upon data distribution, related tables, column sizes, distinct values, average data length, and more. You can customize the recommended facets and easily add new facets using a robust set of declarative attributes.

Related Topics

- Oracle® Application Express App Builder User's Guide

Friendly URL Syntax

Oracle Application Express application URL syntax has been simplified to support friendlier URLs at runtime. Friendly URLs features application or page numbers, and instead, uses the workspace path prefix, application and page aliases, and standard web parameter syntax for its URL structure.

The new syntax provides a Search Engine Optimization (SEO)-friendly URL structure which is far easier to understand and provides immediate context as to where you are within an app.

Related Topics

- Oracle® Application Express App Builder User's Guide

Mega Menus

The Universal Theme now supports Mega Menu navigation. Mega Menu navigation renders app navigation as a collapsible floating panel that displays all navigation items at once. Mega Menu navigation is especially useful for displaying the various aspects of your app and enables direct access to app sections.

Mega Menu navigation includes a number of template options that enable you to customize the layout and appearance so it uniquely works for your app. You can extend Mega Menu items to include icons, descriptions, and even badges using custom list attributes. No matter the layout for larger screens, the mega menu gracefully adjusts its appearance to fit small screen devices. Mega Menus are fully accessible and keyboard friendly.

Related Topics

- Oracle® Application Express App Builder User's Guide

Native PDF Printing

You can now print PDF files directly from interactive grids. This feature produces a PDF file which includes formatting options such as highlighting, column grouping, and column breaks. To make use of this feature, simply enable PDF as an additional download format in your region attributes.

Related Topics

- Oracle® Application Express App Builder User's Guide

New Data Upload

The data upload functionality has been modernized to support native Excel, CSV, XML and JSON documents.

Data upload in SQL Workshop features a new drag and drop user interface that provides support for uploading native Excel, CSV, XML and JSON documents into a new table or into existing tables. These same capabilities can be accessed from the Create Application Wizard by selecting the **From a File** option.

Related Topics

- Oracle® Application Express App Builder User's Guide

New Form Region

A new Form region type provides superior functionality compared to the old legacy form pages.

A new Form region type provides support for internal and external data sources (that is, Local Database, REST Enabled SQL Service, or Web Source). Forms can be created on SQL

queries, or by creating PL/SQL Functions returning SQL Queries. In addition, this new Form region type supports more than two Primary Key Columns and includes more control on Lost Update detection and Row Locking. Finally, you can migrate existing legacy form pages to the new form region with the Upgrade Application function in App Builder Utilities.

Related Topics

• Oracle® Application Express App Builder User's Guide

New Team Development

The Team Development application has been completely reimagined to provide a simple and easy way for your team to collaborate together.

Team Development provides a simple, conversational approach to team communication. Create an issue to track everything in a single place. You can classify issues using label groups and labels, create templates to provide users with starter text for issues and comments, and track progress by assigning milestones. Individual users can register their interest in specific issues and define at a general level what attributes they are interested in watching. Users can also choose whether they wish to be notified in the application, by email, or both.

Related Topics

• Oracle® Application Express App Builder User's Guide

Oracle JET Charts

With every release, Oracle Application Express upgrades Oracle JET Charts. Oracle Application Express release 20.1 includes JavaScript Extension Toolkit (JET) 8.3.0.

Related Topics

• Oracle® Application Express App Builder User's Guide

Popup List of Values (LOV)

Oracle Application Express introduces the completely redesigned Popup List of Values (LOV) which now supports multiple display columns, multiple return values, search-as-you-type, improved multiple value selection, and an all-new simplified user interface.

Redwood UI

The Oracle Application Express user interface has been refreshed to align better with Redwood, Oracle's new user experience design language.

The new design and color scheme extends across the full developer experience and provides refreshing new visuals. The appearance of Application Express can now automatically switch between the dark and light appearance modes based on your OS or platform setting.

Remote Application Deployment

Deploy an application to remote Oracle Application Express instances using REST Enabled SQL references.

One-click remote application deployment leverages the existing REST Enabled SQL references that works with a Oracle REST Data Services (ORDS) REST Enabled SQL Service.

Related Topics

- Oracle® Application Express App Builder User's Guide

REST Read / Write Enhancements

Read and write data using REST Enabled SQL references or Web Source Modules.

Define a REST Enabled SQL reference once and use it across a workspace. Create Web Source Modules to define the source within each application. Both approaches are fully declarative, using Data Discovery with parameters. Forms and interactive grids now include built-in support for the internal or external data source you specify.

Related Topics

- Oracle® Application Express SQL Workshop Guide

Shared List of Values (LOV)

Shared List of Values (LOV) in Oracle Application Express include significant enhancements and new functionality. Shared LOVs now support the full set of data sources (table, SQL Query, function returning SQL query, ORDS REST-Enabled SQL Service and Web Source Modules), declarative column mappings, multiple display columns, grouping, embedded icons, and more.

Related Topics

- Oracle® Application Express App Builder User's Guide

SQL Workshop Now Supports Simple Oracle Document Access (SODA)

SQL Workshop has now been extended to support Simple Oracle Document Access (SODA) collections. Create and store collections of documents in Oracle Database, retrieve them, and query them, without needing to know Structured Query Language (SQL) or how the documents are stored in the database. You can load data to a SODA collection as JSON, TXT, or ZIP files.

Related Topics

- Oracle® Application Express SQL Workshop Guide

Universal Theme Enhancements

The Universal Theme includes UI refinements, accessibility improvements, new template options, and Theme Roller enhancements.

Universal Theme contains many accessibility improvements including a new Skip to Main Content link. The Universal Theme also includes two additional styles for the tree-based Navigation Menu and a new Inline Popup region template. Finally, a number of components and styles have been visually tweaked and refined to provide a more streamlined user interface.

Related Topics

- Oracle® Application Express App Builder User's Guide

Globalization

- Enhanced Parallel Execution Applicability
- New German Linguistic Sorts for Capital Sharp S Support
- New Era Support for Japanese Imperial Calendar
- Unicode 12.1 Support
- Near Zero Downtime

Enhanced Parallel Execution Applicability

Parallel execution can now be used for abstract data types, character keys with linguistic sorts, and timestamp with time zone.

Broadening the applicability of parallel execution to non-native data types can increase the performance of applications using these data types.

New German Linguistic Sorts for Capital Sharp S Support

Two new linguistic sorts (XGERMAN_S and XGERMAN_DIN_S) are added to Oracle Database to support Latin Capital Letter Sharp S as the uppercase form of Latin Smaller Letter Sharp S.

This feature expands the database linguistic support to meet the requirements of the German market.

Related Topics

- Oracle® Database Globalization Support Guide

New Era Support for Japanese Imperial Calendar

The new Japanese era Reiwa, which went into effect on May 1, 2019, is now supported in Oracle Database for the Japanese Imperial Calendar.

The Japanese Imperial Calendar is used widely in Japan in addition to the Gregorian Calendar. Supporting the new era is an essential requirement for software distributed in the Japanese market.

Related Topics

- Oracle® Database Globalization Support Guide

Unicode 12.1 Support

The National Language Support (NLS) data files for AL32UTF8 and AL16UTF16 character sets are updated to match version 12.1 of the Unicode Standard character database. The support for Unicode Collation Algorithm (UCA) is also updated to conform with UCA 12.1.

This enhancement enables Oracle Database to conform to the latest version of the Unicode Standard.

Related Topics

• Oracle® Database Globalization Support Guide

Near Zero Downtime

Starting with Oracle Database 21c, you have the option of performing this method while the database is online or offline. The online version of this method requires one restart of the database at your convenience and requires minimal locks on tables that need to be upgraded. It allows applications to query all time zone data and to insert and modify time zone data for all tables that can be migrated online. The offline version of this method, which was also available in previous releases, requires the database to be in `UPGRADE` mode during part of the procedure and is more restrictive about when applications can insert and modify time zone data.

Related Topics

• Oracle® Database Database Globalization Support Guide

Java in Oracle Database

• Oracle JVM Security Enhancements

Oracle JVM Security Enhancements

Oracle Java virtual machine (JVM) security has been enhanced to make it well suited for pluggable databases (PDBs). The enhancements also include support for the new Java module system and safeguarding of Oracle JVM against security vulnerabilities.

The enhanced Oracle JVM security makes it safe to use Oracle JVM with pluggable databases (PDB) and shields it from security vulnerabilities like the Spectre attack. In addition, it simplifies developing Java in the database applications using the new module system.

Related Topics

• Oracle® Database Java Developer's Guide

JDBC

• Java Library for Reactive Streams Ingestion
• JDBC DataSource for Sharded Databases Access
• JDBC Reactive Extensions

- JDBC Support for Native JSON Data Type

Java Library for Reactive Streams Ingestion

This new Java library allows high-speed ingestion of data streams with non-blocking back pressure.

Java applications that use the provided APIs may continuously receive and ingest data from a large group of clients.

The ingestion process is non-blocking and extremely fast through the direct path load into the database tables. The library performs affinity grouping by destination including Oracle Real Application Clusters (RAC) nodes, Oracle Active Data Guard, and Sharded database. The library performs CPU optimization by decoupling record processing from database IO. Through the Universal Connection Pool (UCP), the ingestion process furnishes high availability and scalability.

This feature enables implementing high-speed ingestion of streaming data including sensors data, time series (trading), Call Detail Records (CDRs), geoSpatial activities, social media feeds, web site logs, and so on, with scalability and high availability.

Related Topics

- Oracle® Database JDBC Developer's Guide

JDBC DataSource for Sharded Databases Access

You can use the new JDBC datasource to simplify the access of Java applications to sharded databases, where the applications can connect to the sharded databases transparently, with almost no change to the application code.

This feature enables Java connectivity to a sharded database without the need for a Java application to furnish a sharding key because the sharding data source derives the sharding key from the SQL statement. Your Java applications can scale out to sharded databases transparently as there is little to no change to the application code.

The sharding data source optimizes the performance of your applications as you do not need to configure the Universal Connection Pool (UCP) or create separate connection pools for cross-shard statements and single-shard statements. Also, you do not have to check-in or check-out a physical connection for every new sharding key because the sharding data source does it automatically.

Related Topics

- Oracle® Database JDBC Developer's Guide

JDBC Reactive Extensions

The Reactive Extensions are a set of methods that extend the JDBC standard to offer asynchronous database access with back pressure support. The Reactive Extensions implement the Publisher and Subscriber types defined by `java.util.concurrent.Flow`. Flow is the JDK's standard representation of a reactive stream.

The Reactive Extensions use non-blocking mechanisms for creating Connections, executing SQL, fetching rows, committing, rolling back, closing Connections, and reading and writing BFILEs, BLOBs, and CLOBs.

The extensions bring scalability, high throughput, and Reactive Streams support to Java applications that use the Oracle Database in the Cloud and on-premises. This extension works as a Service Provider for the Reactive Streams libraries including: Reactor, RxJava, and Akka Streams.

Related Topics

- Oracle® Database JDBC Developer's Guide

JDBC Support for Native JSON Data Type

This release introduces a new JSON datatype for efficient storage of JSON data. The package `oracle.sql.json` provides classes and interfaces for working with SQL JSON type values.

This feature furnishes a simpler and richer type system (that is, support for dates, timestamps), no constraint check (IS JSON) and improves the performance of Java applications (that is, faster access to nested JSON values). A JSON parser inside the JDBC driver 21c allows optimizations during the conversion from Java String to native JSON and the other way around.

Related Topics

- Oracle® Database JDBC Java API Reference

JSON Document Store

- New JSON Data Type
- New Oracle SQL Function JSON_TRANSFORM
- SQL/JSON Syntax Improvements
- Multivalue Index for JSON
- Enhancements to View Creation and of Virtual Columns Addition
- JSON Scalar Allowed at Top Level of JSON Document (RFC 8259 Support)
- Compatibility for Old Clients With Binary JSON

New JSON Data Type

`JSON` is a new SQL and PL/SQL data type for JSON data. It provides a substantial increase in query and update performance.

The JSON data type is optimized for query and DML processing. It can yield significant database performance improvements for processing JSON data.

You can use the JSON data type in most places where a SQL data type is allowed, including:

- As the column type for table or view DDL
- With SQL/JSON functions and conditions, and with PL/SQL procedures and functions

- In Oracle dot-notation query syntax

- For creation of functional and search indexes

Oracle Call Interface (OCI) and Java Database Connectivity (JDBC) clients now provide APIs that can work directly with binary JSON data, significantly saving network costs and server CPU cycles.

Database In-Memory supports the ability to populate JSON data type columns in the In-Memory column store. JSON data defined with the JSON data type is populated in a proprietary binary format that is optimized for query processing and can significantly improve JSON processing performance.

Related Topics

- Oracle® Database JSON Developer's Guide

New Oracle SQL Function JSON_TRANSFORM

You can use SQL function `JSON_TRANSFORM` to update parts of a JSON document. You specify which parts to modify, the modifications, and any new values.

`JSON_TRANSFORM` makes it easier for an application to modify a JSON document, without having to parse and rebuild it. In most cases, it also avoids a round-trip between the server and client for the whole document.

Related Topics

- Oracle® Database JSON Developer's Guide

SQL/JSON Syntax Improvements

You can now express more complex SQL/JSON queries and express some queries more succinctly.

- New SQL function JSON_SCALAR accepts a scalar instance of a SQL data type and returns a scalar JSON value as an instance of JSON data type.

- New JSON path-language item methods support `JSON_SCALAR`: `float()`, `double()`, `binary()`, `ymInterval()`, and `dsInterval()`.

- The JSON path-language and dot-notation syntax support new, aggregate item methods: `avg()`, `count()`, `minNumber()`, `maxNumber()`, `minString()`, `maxString()`, and `sum()`.

- You can now express more complex SQL/JSON queries and express some queries more succinctly, and SQL/JSON path-expression syntax for array steps is improved.

Related Topics

- Oracle® Database JSON Developer's Guide

Multivalue Index for JSON

A new create index syntax `CREATE MULTIVALUE INDEX` allows you to create a functional index on arrays of strings or numbers within a JSON type column. Each unique value within the array will become a searchable index entry.

This avoids the need for full JSON scans to find values within arrays in JSON columns, when searched using the `JSON_EXISTS` or `JSON_QUERY` operators. It provides similar benefits to conventional functional indexes when searching JSON, but conventional functional indexes are limited to a single indexed value per row.

Related Topics

- Oracle® Database JSON Developer's Guide

Enhancements to View Creation and of Virtual Columns Addition

`DBMS_JSON.CREATE_VIEW` now gives you the option to create a materialized view instead of a standard view. It also gives you the option to specify a particular path so the view can be created on a subset of the data. Both `CREATE_VIEW` and `ADD_VIRTUAL_COLUMN` are enhanced to allow automatic resolution of column naming conflicts, to provide a prefix to be applied to column names, and to specify the case-sensitivity of column names.

Enhances development flexibility and allows for materialized views, which may improve query performance with a trade-off against DML performance.

Related Topics

- Oracle® Database JSON Developer's Guide

JSON Scalar Allowed at Top Level of JSON Document (RFC 8259 Support)

JSON documents in Oracle Database can now have a top-level JSON scalar value. Previously they had to have a JSON object or array value.

This feature helps Oracle JSON support be compliant with RFC 8259.

This feature will only be available when the database initialization parameter compatible is set to `20` or higher.

If the parameter value is at least `20`, then JSON data that is stored either textually (`VARCHAR2`, `CLOB`, `BLOB`) or as JSON data type respects RFC 8259: it allows top-level scalars in documents. For a JSON column you can, however, use an `IS JSON` check constraint with keywords `DISALLOW SCALARS` to disallow documents having top-level scalar value.

Related Topics

- Oracle® Database JSON Developer's Guide

Compatibility for Old Clients With Binary JSON

The database supports JSON in a binary format. In some situations, clients will work directly with the binary format. However, this requires that the client be a recent enough version to understand the format. The database will now identify older clients and automatically transform JSON data from binary to text (serialized) format before transmitting it to them, and convert back from text to binary format as required.

Support for older clients without being forced to upgrade.

Related Topics

- Oracle® Call Interface Programmer's Guide

Net Services

- [Migration of Oracle CMAN Sessions with High Availability](#)
- [Oracle CMAN Traffic Director Mode Support for All Types of Database Links](#)
- [REST APIs for Oracle CMAN Administration, Proxy Protocol, Enhanced Rule List, and Bandwidth Management](#)
- [Reverse Connection Support Using CMAN Tunnels](#)

Migration of Oracle CMAN Sessions with High Availability

Client/server sessions can be migrated from one Oracle Connection Manager (CMAN) instance to another Oracle CMAN instance during a planned upgrade or while patching Oracle CMAN. Live sessions can be migrated with data in-transit.

Operations that are running either on a client or on a server continue to run seamlessly during the migration with zero downtime. You can also add new client connections during the migration.

Related Topics

- [Oracle® Database Net Services Administrator's Guide](#)

Oracle CMAN Traffic Director Mode Support for All Types of Database Links

Starting with this release, Oracle Connection Manager (CMAN) Traffic Director Mode is extended to support all types of dedicated database links including Fixed User, Connected User, and Current User.

Oracle Connection Manager in Traffic Director Mode enhances application scalability, performance, security, tenant isolation, and high availability (zero downtime during planned and unplanned database outages).

Related Topics

- [Oracle® Database Net Services Administrator's Guide](#)

REST APIs for Oracle CMAN Administration, Proxy Protocol, Enhanced Rule List, and Bandwidth Management

You can use REST APIs to manage Oracle Connection Manager (Oracle CMAN) instances. Proxy protocol provides additional security and access control. Enhanced rule list allows scalability with segregation of rule_lists for each service. You can also manage distribution of bandwidth across services using Oracle CMAN.

This feature offers REST API management, security with proxy protocol, scalability with enhanced rule list, and per service bandwidth management.

Related Topics

- [Oracle® Database Net Services Administrator's Guide](#)

Reverse Connection Support Using CMAN Tunnels

Starting with Oracle Database 21c, you can use secure tunnels to connect to an Oracle Database instance, which is inside a network that supports only outbound connections.

A network may allow only outbound connections and restrict inbound connections for security reasons. However, using the Oracle Connection Manager tunnel feature, you can connect to a database inside a network that allows only outbound connections. Oracle Connection Manager creates a pool of connections, known as tunnels, that can be used to connect to a database inside the network.

Related Topics

- Oracle® Database Net Services Administrator's Guide

Oracle Call Interface

- Improvements to OCI Data Interface for LOBs
- New C Client Interface APIs for JSON Data Type
- Oracle Call Interface API to Build and Quote TNS Connection Strings
- Oracle Call Interface Session Pool Improvements
- Oracle SODA for C and PL/SQL APIs Enhancements
- Support for C99 Syntax in Pro*C/C++ Precompiler

Improvements to OCI Data Interface for LOBs

This feature makes the OCI Data Interface for LOBs more user friendly.

The OCI Data Interface for LOBs is efficient because it reduces the number of round trips between the client and the database. This feature is further improved by removing the need for NULL callbacks and usage of 'ind' variable when the LOB length exceeds the sb2 size.

Related Topics

- Oracle® Call Interface Programmer's Guide

New C Client Interface APIs for JSON Data Type

This feature introduces new Oracle Call Interface (OCI) APIs to operate with JSON descriptor. You can read and write textual JSON or binary JSON (in Oracle binary format) from/to a buffer or a stream. You can bind and define a JSON descriptor to write and read JSON data from the database.

With the new JSON APIs, you can now preserve abstraction without a need to serialize the data to text.

Related Topics

- Oracle® Call Interface Programmer's Guide

Oracle Call Interface API to Build and Quote TNS Connection Strings

A new Oracle Call Interface API can now be used to build and quote Oracle Database Transparent Network Substrate (TNS) connection strings.

The Oracle Call Interface enhancement enables validation of values, which you can use for Oracle Net Service connection string attributes.

Related Topics

• Oracle® Call Interface Programmer's Guide

Oracle Call Interface Session Pool Improvements

The Oracle Call Interface Session Pool has been enhanced.

Monitoring and administration has been improved.

When the number of sessions exceeds the minimum pool size, idle session cleanup now occurs even if there is no pool activity.

There is better OCISessionGet wait timeout accuracy.

When using Oracle Sharding, the pool has better balancing of session across shards.

You can now manage Oracle Call Interface Session Pool better with performance, auto-tuning, and administration improvements.

Related Topics

• Oracle® Call Interface Programmer's Guide

Oracle SODA for C and PL/SQL APIs Enhancements

This feature introduces new SODA APIs for saving documents to a collection, truncating a document collection, and specifying the array fetch size to read documents from a collection, which reduces network round-trips.

The enhancements improve interoperability of the SODA APIs with the database and utilities.

Related Topics

• Oracle® Call Interface Programmer's Guide

Support for C99 Syntax in Pro*C/C++ Precompiler

The Pro*C/C++ Precompiler now supports the C99 standard, the ISO/IEC 9899:1999 standards specification for C programming.

C99 syntax and semantics allow application developers to use richer functionality.

Related Topics

• Pro*C/C++ Programmer's Guide

Oracle Data Provider for .NET (ODP.NET)

- Managed ODP.NET and ODP.NET Core: Bulk Copy
- Managed ODP.NET and ODP.NET Core: Debug Tracing Redaction
- Managed ODP.NET and ODP.NET Core: WebSocket and WebSocket Secure

Managed ODP.NET and ODP.NET Core: Bulk Copy

ODP.NET Bulk Copy enables applications to efficiently load large amounts of data from a table in one database to another table in a different database. Managed ODP.NET and ODP.NET Core now support Bulk Copy and all its APIs.

ODP.NET Bulk Copy is the most optimized .NET solution when a large data set needs to loaded into a table or between database tables in different databases.

Related Topics

- Oracle® Data Provider for .NET Developer's Guide

Managed ODP.NET and ODP.NET Core: Debug Tracing Redaction

Managed ODP.NET and ODP.NET Core has introduced a new trace level, that can exclude SQL statements and network packet contents from being included in the trace file.

Related Topics

- Oracle® Data Provider for .NET Developer's Guide

Managed ODP.NET and ODP.NET Core: WebSocket and WebSocket Secure

Managed ODP.NET and ODP.NET Core now support both WebSocket and WebSocket Secure. Unmanaged ODP.NET already supports these protocols.

WebSocket has the ability to engage in two-way communication simultaneously using a single connection between web application and host unlike HyperText Transfer Protocol (HTTP). WebSocket Secure uses SSL/TLS to encrypt communications.

Related Topics

- Oracle® Data Provider for .NET Developer's Guide

PL/SQL

- DBMS_CLOUD Package
- JavaScript Execution using DBMS_MLE
- New PL/SQL Iterator Constructs
- New Pragma SUPPRESSES_WARNING_6009
- PL/SQL Qualified Expressions Enhancements

- PL/SQL Support For New JSON SQL Data Type
- PL/SQL Type Attributes in User-Defined Types

DBMS_CLOUD Package

Oracle provides two core mechanisms to work with data in object stores, as part of the new `DBMS_CLOUD` package or manually defining external tables.

Using `DBMS_CLOUD` provides benefits and additional functionality that goes beyond DDL and is fully compatible with Oracle Autonomous Database. Oracle strongly recommends leveraging the new `DBMS_CLOUD` package over manual external table creation.

Related Topics

- Oracle® Database PL/SQL Packages and Types Reference

JavaScript Execution using DBMS_MLE

The `DBMS_MLE` package allows users to execute JavaScript code inside the Oracle Database and exchange data seamlessly between PL/SQL and JavaScript. The JavaScript code itself can execute PL/SQL and SQL through built-in JavaScript modules. JavaScript data types are automatically mapped to Oracle Database data types and vice versa.

With the `DBMS_MLE` package, developers can write their data processing logic in JavaScript. JavaScript is a widely-used and popular programming language that can now also be used for writing programs that need to execute close to the data.

Related Topics

- Oracle® Database PL/SQL Packages and Types Reference

New PL/SQL Iterator Constructs

PL/SQL is enhanced to help you program iteration controls using new iterators in loops and in qualified expressions.

The new iterator constructs are clear, simple, understandable, and efficient.

Related Topics

- Oracle® Database PL/SQL Language Reference

New Pragma SUPPRESSES_WARNING_6009

The `SUPPRESSES_WARNING_6009` pragma allows more robust error handling and better encapsulation and modularization.

The PL/SQL compiler issues warning `PLW-06009` if it determines that an `OTHERS` exception handler does not, in all cases, end in either an explicit `RAISE` statement or in a call to the PL/SQL supplied procedure `RAISE_APPLICATION_ERROR`. The compiler's behavior is too aggressive for some programming styles when programmers supply their own reporting subroutines. This new pragma quiets the warning.

Related Topics

- [Oracle® Database PL/SQL Language Reference]

PL/SQL Qualified Expressions Enhancements

Starting with this release, three new types of iterator choice association are added for use in qualified expressions. The basic iterator choice association extends the current iterator choice association by allowing a full iterator as the index. The index iterator choice association provides an index expression along with the value expression. The sequence iterator choice association allows a sequence of values to be added to the end of a collection. In each case, the expressions specified may reference the iterands.

Qualified expressions improve program clarity and programmer productivity.

Related Topics

- [Oracle® Database PL/SQL Language Reference]

PL/SQL Support For New JSON SQL Data Type

You can use the new JSON SQL data type in PL/SQL.

The new JSON data type allows you to pass JSON data from SQL to PL/SQL and back to SQL (static and dynamic). PL/SQL now supports binding directly JSON data from client side interfaces, such as Oracle Call Interface (OCI) and Java Database Connectivity (JDBC), as well, as from PL/SQL to callouts. JSON can be a differentiating type in overload resolution.

Related Topics

- [Oracle® Database PL/SQL Language Reference]

PL/SQL Type Attributes in User-Defined Types

You can use attributes of PL/SQL scalar data types, such as BOOLEAN and `PLS_INTEGER`, in non-persistable object types.

You can use non-persistable object types in your PL/SQL code if you have no desire to persist instances of these types. This is useful when you are developing programs following Oracle's object oriented programming model.

Related Topics

- [Oracle® Database PL/SQL Language Reference]

SQL

- [Enhanced SQL Set Operators]
- [Expression Support for Initialization Parameters]
- [Placeholders in SQL DDL Statements]
- [SQL Macros]

Enhanced SQL Set Operators

The SQL set operators now support all keywords as defined in ANSI SQL. The new operator `EXCEPT [ALL]` is functionally equivalent to `MINUS [ALL]`. The operators `MINUS` and `INTERSECT` now support the keyword `ALL`.

Full ANSI compliance provides greater compatibility with other database vendors and makes migration to Oracle Database easier than before.

Related Topics

- Oracle® Database SQL Language Reference
- **New Set Operators** lab in the Database 21c New Features workshop in LiveLabs

Expression Support for Initialization Parameters

You can specify an expression when setting the value of an initialization parameter.

In previous releases, you were required to specify an absolute value when setting an initialization parameter. You can now specify an expression that takes into account the current system configuration and environment. This is especially useful in Oracle Autonomous Database environments.

Related Topics

- Oracle® Database Reference
- **Expressions in Init Parameters** lab in the Database 21c New Features workshop in LiveLabs

Placeholders in SQL DDL Statements

SQL DDL statements can now contain placeholders instead of hard coded values for some content. For example, placeholders may be used where a username or password are required in a `CREATE USER` statement. Oracle Call Interface programs can substitute values into the DDL statement placeholders before the statements are sent to Oracle Database. This is similar to data binding, but occurs in Oracle Client.

Application security is improved because values do not need to be hard coded in SQL DDL.

Related Topics

- Oracle® Call Interface Programmer's Guide

SQL Macros

You can create SQL Macros (SQM) to factor out common SQL expressions and statements into reusable, parameterized constructs that can be used in other SQL statements. SQL macros can either be scalar expressions, typically used in `SELECT` lists, `WHERE`, `GROUP BY` and `HAVING` clauses, to encapsulate calculations and business logic or can be table expressions, typically used in a `FROM` clause. SQL macros furthermore can improve performance as compared to PL/SQL constructs.

SQL macros increase developer productivity, simplify collaborative development, and improve code quality.

Related Topics

- Oracle® Database PL/SQL Language Reference
- Using **SQM Scalar and Table Expressions** lab in the Database 21c New Features workshop in LiveLabs

Text

- Custom Range Bucketing in Result Set Interface
- Facet Navigation Support for JSON Search Indexes
- Improved Index Synchronization and Automatic Index Optimization
- In-Memory Full Text Columns
- JSON Support in Result Set Interface
- Named Entity Recognition Improvements
- New DIRECTORY_DATASTORE Data Store Type for Oracle Text
- New NETWORK_DATASTORE Data Store Type for Oracle Text
- New Oracle Text Index Type: Search Index

Custom Range Bucketing in Result Set Interface

You can now specify range buckets for faceted navigation. For example, you can group all items costing between $10 and $20.

Range buckets increase the usability of faceted navigation and avoid the need to manually create range-value sections in the indexed table.

Related Topics

- Oracle® Text Application Developer's Guide

Facet Navigation Support for JSON Search Indexes

JSON search indexing now supports facet navigation using a JSON result set.

Related Topics

- Oracle® Text Application Developer's Guide

Improved Index Synchronization and Automatic Index Optimization

Index synchronization performance is considerably improved, particularly when lots of sessions are doing DML operations simultaneously with the sync-on-commit option. Optimize index now optimizes all index entries (including SDATA values) rather than just the word-based index entries.

Related Topics

- Oracle® Text Application Developer's Guide

In-Memory Full Text Columns

In previous releases, queries using `CONTAINS()` and `JSON_TEXTCONTAINS()` were only evaluated with a text index and JSON search index. Oracle Text technology now supports an `INMEMORY TEXT` clause during table creation, enabling fast in-memory searching rather than creating a separate on-disk text index.

An in-memory table scan can now evaluate columns directly when they are specified as `INMEMORY TEXT`. Queries using `CONTAINS()` or `JSON_TEXTCONTAINS()` can evaluate these operators in SQL predicates without requiring separate domain-specific indexes. When the In-Memory Column Store contains both scalar and non-scalar columns, On-line Transaction Processing (OLTP) applications that access both types of data can avoid accessing the row store, thereby improving performance.

Related Topics

- Oracle® Database In-Memory Guide
- **Searching Text/JSON in In-Memory** lab in the Database 21c New Features workshop in LiveLabs

JSON Support in Result Set Interface

The JSON Result Set Interface (RSI) enables you to perform queries in JSON and return results as JSON.

The RSI enables you to fetch a set of results (a "hitlist") together with summary data such as the total number of hits and facet navigation information. This feature provides easier integration with modern programming languages which support JSON.

Related Topics

- Oracle® Text Application Developer's Guide

Named Entity Recognition Improvements

Some restrictions have been removed from Named Entity Recognition (previously called Entity Extraction). Entity definitions can now be nested.

You can now incorporate existing or user-defined rules into other rules. For example, you can define rule "percentage change" as one of several words (for example, 'climbed', 'dropped', 'rose', 'increased') followed by the built-in rule for percentage.

Related Topics

- Oracle® Text Application Developer's Guide

New DIRECTORY_DATASTORE Data Store Type for Oracle Text

You can now use a new data store type called `DIRECTORY_DATASTORE` instead of the deprecated `FILE_DATASTORE` type.

`DIRECTORY_DATASTORE` provides greater security because it enables file access to be based on directory objects.

Related Topics

- [Oracle® Text Application Developer's Guide](#)

New NETWORK_DATASTORE Data Store Type for Oracle Text

You can now use a new data store type called `NETWORK_DATASTORE` instead of the deprecated `URL_DATASTORE` data type.

`NETWORK_DATASTORE` provides greater security because it enables URL access based on access control lists (ACLs), which support `HTTP` and `HTTPS` protocols.

Related Topics

- [Oracle® Text Application Developer's Guide](#)

New Oracle Text Index Type: Search Index

The search indexes for text columns support support sharding and all partition types. Index creation syntax is also simpler now.

The search index type maintains most of the features of the traditional `CONTEXT` index type, but has support for sharding and for multiple partitioning methods (the previous `CONTEXT` index type can only support range partitioning). It also has a simplified index creation syntax avoiding the semantic complexity necessary for `CONTEXT` indexes.

Related Topics

- [Oracle® Text Application Developer's Guide](#)

3

Big Data and Data Warehousing Solutions

- Analytic Views
- Analytical SQL and Statistical Functions
- Machine Learning for SQL
- Machine Learning for Python
- Query Optimization
- Spatial and Graph

Analytic Views

- Analytic View Enhancements to SQL and PL/SQL

Analytic View Enhancements to SQL and PL/SQL

SQL and PL/SQL enhancements for analytic views are the following:

- Star views can be created on fact and dimension tables
- Analytic views are created automatically for queries of the tables and views
- Level group caches are created and maintained autonomously
- Query-scoped base measures
- Support for remote sources
- Support for aggregate tables
- Star support for dimension caches

With these enhancements:

- Star view and star table queries can use the benefits of analytic views without any code changes
- Front-end tools can use analytic view calculations by inserting a new expression
- Level group caches can become more efficient over time based on usage patterns

Related Topics

- Oracle® Database Data Warehousing Guide

Analytical SQL and Statistical Functions

- Bitwise Aggregate Functions
- Enhanced Analytic Functions
- New Analytical and Statistical Aggregate Functions

Bitwise Aggregate Functions

New aggregate functions `BIT_AND_AGG`, `BIT_OR_AGG`, and `BIT_XOR_AGG` enable bitwise aggregation of integer columns and columns that can be converted or rounded to integer values.

Bitwise aggregation functions enable bitwise type processing directly in SQL. Use of these new functions improves overall query performance by eliminating unnecessary data movement and by taking full advantage of other database capabilities such as parallel processing.

Related Topics

- Oracle® Database SQL Language Reference

- **Use Bitwise Aggregate Functions** lab in the Database 21c New Features workshop in LiveLabs

Enhanced Analytic Functions

Window functions now support the `EXCLUDE` options of the SQL standard window frame clause. The query_block clause of a `SELECT` statement now supports the `window_clause`, which implements the window clause of the SQL standard table expression as defined in the SQL:2011 standard.

Supporting the full ANSI standard enables easier migration of applications that were developed against other standard-compliant database systems.

Related Topics

- Oracle® Database Data Warehousing Guide

- **Use Enhanced Analytic Functions** lab in the Database 21c New Features workshop in LiveLabs

New Analytical and Statistical Aggregate Functions

New analytical and statistical aggregate functions are available in SQL:

`CHECKSUM` computes the checksum of the input values or expression.

`KURTOSIS` functions `KURTOSIS_POP` and `KURTOSIS_SAMP` measure the tailedness of a data set where a higher value means more of the variance within the data set is the result of infrequent extreme deviations as opposed to frequent modestly sized deviations. Note that a normal distribution has a kurtosis of zero.

`SKEWNESS` functions `SKEWNESS_POP` and `SKEWNESS_SAMP` are measures of asymmetry in data. A positive skewness is means the data skews to the right of the center point. A negative skewness means the data skews to the left.

All of these new aggregate functions support the keywords `ALL`, `DISTINCT`, and `UNIQUE`.

`ANY_VALUE`, a function to simplify and optimize the performance of `GROUP BY` statements, returns a random value in a group and is optimized to return the first value in the group. It ensures that there are no comparisons for any incoming row and eliminates the necessity to specify every column as part of the `GROUP BY` clause.

With these additional SQL aggregation functions, you can write more efficient code and benefit from faster in-database processing.

Related Topics

- Oracle® Database SQL Language Reference

- **Detect Data Tampering with the CHECKSUM Function**, **Measure Asymmetry in Data with the SKEWNESS Functions**, and **Measure Tailedness of Data with the KURTOSIS Functions** labs in the Database 21c New Features workshop in LiveLabs

Machine Learning for SQL

- Adam Optimization Solver for the Neural Network Algorithm
- Oracle Machine Learning MSET-SPRT Algorithm
- Oracle Machine Learning XGBoost Algorithm

Adam Optimization Solver for the Neural Network Algorithm

Adam is an optimization solver for the Neural Network algorithm that is computationally efficient, requires little memory, and is well suited for problems that are large in terms of data or parameters or both.

Adam is a popular extension to stochastic gradient descent. It uses mini-batch optimization and can make progress faster while seeing less data than the other Neural Network optimization solver, Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) with line search.

Related Topics

- Oracle® Machine Learning for SQL Concepts

Oracle Machine Learning MSET-SPRT Algorithm

The Multivariate State Estimation Technique-Sequential Probability Ratio Test (`MSET-SPRT`) algorithm is a nonlinear, nonparametric anomaly detection technique for monitoring critical processes.

The `DBMS_DATA_MINING.ALGO_MSET_SPRT` algorithm detects subtle anomalies while producing minimal false alarms. It calibrates expected behavior from available, historical data of the normal operational sequence of monitored signals. It incorporates the learned behavior of the system into a persistent `MSET-SPRT` model. You can apply the model to new records to detect anomalous behavior.

Related Topics

- Oracle® Machine Learning for SQL Concepts

Oracle Machine Learning XGBoost Algorithm

XGBoost is a highly-efficient, scalable gradient tree boosting machine learning algorithm for regression and classification.

The `DBMS_DATA_MINING.ALGO_XBGOOST` algorithm prepares training data, builds and persists a model, and applies the model for prediction. You can use it as a stand-alone predictor or incorporate it into real-world production pipelines for a wide range of problems such as ad click-through rate prediction, hazard risk prediction, web text classification, and so on.

Related Topics

- Oracle® Machine Learning for SQL Concepts

Machine Learning for Python

- Oracle Machine Learning for Python (OML4Py)
- Oracle Machine Learning for Python Configuration in DBCA

Oracle Machine Learning for Python (OML4Py)

Oracle Machine Learning for Python (OML4Py) enables the open source Python programming language and environment to operate on database data at scale. Python users can run Python commands and scripts for statistical analysis and machine learning on data stored in Oracle Database.

With OML4Py, you can do the following:

- Use a wide range of in-database machines learning algorithms
- Minimize data movement
- Leverage Oracle Database as a high performance compute engine for data exploration and preparation
- Use AutoML for automatic algorithm selection, feature selection, and model tuning
- Execute user-defined Python functions in non-parallel, data-parallel, and task-parallel fashion

Related Topics

- Oracle® Machine Learning for Python

Oracle Machine Learning for Python Configuration in DBCA

The Database Configuration Assistant (DBCA) supports configuring Oracle Machine Learning for Python (OML4Py).

Configuration through DBCA is supported for container databases and pluggable databases.

Related Topics

- Oracle® Multitenant Administrator's Guide

Query Optimization

- In-Memory Vectorized Joins

In-Memory Vectorized Joins

The In-Memory vectorized joins feature is based on the deep vectorization framework. Using SIMD vector processing, the framework optimizes aspects of hash joins such as hashing, building, probing, and gathering. This optimization can improve the performance of join processing by 100% or more. The In-Memory vectorized joins feature is transparent to the user, requiring no plan changes.

In-Memory deep vectorization is a SIMD-based query processing framework that supports vectorization for higher-level query operators in the query plan. The framework includes optimizations such as SIMD, hardware acceleration, and pipelined execution.

Joins can account for a large percentage of SQL execution time for data warehouse workloads. Improving performance of hash joins by 100% or more can significantly improve performance.

Related Topics

- Oracle® Database In-Memory Guide

Spatial and Graph

- Property Graph: Graph Server and Client Kit
- Property Graph: Native Python Client
- Property Graph: New Features in PGQL
- Property Graph: Optimized Graph Representation for Faster Performance
- Property Graph: User-defined Graph Algorithms
- Property Graph Visualization
- RDF Graph: Native Unicode Storage and Processing
- Self-service Low-code Spatial Studio
- Spatial Network Data Model Contraction Hierarchy
- Spatial Support for Database In-Memory

Property Graph: Graph Server and Client Kit

You no longer have to copy the Property Graph in-memory analytics server (PGX) and client tools and libraries from `$ORACLE_HOME`.

With the simplified packaging and the availability of the Oracle Graph Server and Client kit on OCI Marketplace, application developers can quickly and securely install and deploy the components required to work with Property Graphs. This makes it easier to start developing applications.

Related Topics

- Oracle Spatial and Graph Property Graph Developer's Guide

Property Graph: Native Python Client

Property Graph has a native Python API that allows you to create a graph, run graph queries (using PGQL), and analyze the graph using Python and Jupyter notebooks.

This makes it easier to get started with graph and simplifies integration in data science applications where Python and Jupyter are especially popular.

Related Topics

- Oracle Spatial and Graph Property Graph Developer's Guide

Property Graph: New Features in PGQL

You can now do graph DDL and graph DML operations with the graph query language PGQL. Additionally, you can use PGQL for CHEAPEST path queries using cost functions.

This simplifies development by eliminating the need for configuration files and Java APIs when doing several common graph operations.

Related Topics

- Oracle Spatial and Graph Property Graph Developer's Guide

Property Graph: Optimized Graph Representation for Faster Performance

The In-memory analytics server (PGX, or Property Graph AnalytiX) has an optimized representation of a property graph that uses less memory. Larger graphs can be analyzed in the same amount of memory.

Graph sizes are continuously growing larger. With this optimization you can analyze larger graphs using less memory than previously required. This not only enables analysis of more data, but also reduces system costs. The optimized graph representation gives you faster performance and is transparent; existing applications will run faster with no change.

Related Topics

- Oracle Spatial and Graph Property Graph Developer's Guide

Property Graph: User-defined Graph Algorithms

You can create or extend graph algorithms using Java syntax, in addition to the dozens of pre-built graph analytics that come with the product. These user-defined algorithms will execute as fast as native algorithms in the product because they are compiled with the same optimizations.

For unique and specialized use cases, customizing graph algorithms lets you add analysis that analysts and data scientists design specifically for your applications.

Related Topics

- Oracle Spatial and Graph Property Graph Developer's Guide

Property Graph Visualization

A rich set of visualization features lets you interactively explore the graph, customize layouts, and highlight interesting relationships in your data.

Seeing graph data and relationships visually lets analysts, data scientists, and developers quickly understand and explore clusters, outliers, anomalies, patterns, communities, and critical connections in their data. This makes you more productive and helps you share and communicate results more clearly.

Related Topics

- Oracle Spatial and Graph Property Graph Developer's Guide

RDF Graph: Native Unicode Storage and Processing

When creating the RDF (Resource Description Framework) network, you can now store RDF data in a native unicode format for virtually all use cases.

This reduces the storage required and enhances query performance.

Related Topics

- Oracle® Database Graph Developer's Guide for RDF Graph

Self-service Low-code Spatial Studio

Oracle Spatial Studio is a self-service web application that makes it easy for you to create interactive maps and perform spatial analysis on your business data. You can also use Spatial Studio to publish spatial analyses as REST services and generate SQL statements for spatial analysis using low-code and UI components.

With Spatial Studio, you no longer need to write Javascript or SQL or use third-party tools to take advantage of the spatial capabilities in Oracle Database.

Related Topics

- Oracle® Spatial Developer's Guide

Spatial Network Data Model Contraction Hierarchy

Contraction hierarchy, a precomputed in-memory approach, speeds up path computations in the Oracle Spatial and Graph network data model.

By using the contraction hierarchy Java API, you can evaluate shortest path computations, drive time polygon analysis, and traveling salesperson analysis functions more efficiently. These functions will perform 10 to 100 times faster than with previous releases. You can run more network analysis functions using fewer CPUs, and support more concurrent requests using the same hardware.

Related Topics

- Oracle® Topology and Network Data Model Developer's Guide

Spatial Support for Database In-Memory

A new In-Memory Spatial index is created when the `INMEMORY SPATIAL` clause is specified for a spatial geometry column during a `CREATE` or `ALTER TABLE` statement.

This means that you no longer have to create and maintain conventional disk-based spatial indexes when spatial tables are stored using Database In-Memory. The in-memory spatial index provides much faster query performance for SDO_FILTER() operations without having to maintain a separate, on-disk spatial index.

Related Topics

- Oracle® Spatial Developer's Guide

4

Database Upgrade and Utilities

- Database Utilities
- Upgrades and Migration

Database Utilities

- Oracle Data Pump Checksums Support Cloud Migrations
- Oracle Data Pump Exports from Oracle Autonomous Database
- Oracle Data Pump Includes and Excludes in the Same Operation
- Oracle Data Pump Parallelizes Transportable Tablespace Metadata Operations
- Oracle Data Pump Provides Optional Index Compression
- Oracle Data Pump Resumes Transportable Tablespace Jobs
- Oracle Data Pump Supports Export to and Import From Cloud Object Stores
- Oracle Data Pump Supports Native JSON Datatypes
- Oracle SQL*Loader Support for Object Store Credentials
- Oracle SQL*Loader Supports Native JSON Data Type

Oracle Data Pump Checksums Support Cloud Migrations

To check Oracle Data Pump dump files for validity, you can now use checksums that are added to the dump file.

Oracle Data Pump is used for migrating application data from on-premises Oracle Database instances into the Oracle Cloud, and also for copying dump files to on-premises.

In this release, a checksum is now added to the dump file. You can use the checksum to help to confirm that the file is valid after a transfer to or from the object store and also after saving dump files on on-premises and that it has no accidental or malicious changes.

Related Topics

- Oracle® Database Database Utilities
- **Data Pump Checksums** lab in the Database 21c New Features workshop in LiveLabs

Oracle Data Pump Exports from Oracle Autonomous Database

Starting with this release, Oracle Data Pump can perform exports from Oracle Autonomous Database into dump files in a cloud object store.

Oracle Data Pump supports dump file export from cloud services to the object store. You can now migrate data from services you manage in Oracle Autonomous Database.

Related Topics

- Oracle® Database Database Utilities

Oracle Data Pump Includes and Excludes in the Same Operation

Starting with this release, Oracle Data Pump can include and exclude objects in the same export or import operation.

Oracle Data Pump provides powerful, flexible inclusion and exclusion of objects for a job. Now, Oracle Data Pump commands can include both `INCLUDE` and `EXCLUDE` parameters in the same operation. By enabling greater specificity about what is being migrated, this enhancement makes it easier to migrate to Oracle Cloud, or to another on-premises Oracle Database.

Related Topics

- Oracle® Database Database Utilities
- **Data Pump Includes & Excludes** lab in the Database 21c New Features workshop in LiveLabs

Oracle Data Pump Parallelizes Transportable Tablespace Metadata Operations

Starting with this release, Oracle Data Pump improves Transportable Tablespace metadata operations with parallelism.

Oracle Data Pump now supports parallel export and import operations for Transportable Tablespace (TTS) metadata. This is the information that associates the tablespace data files with the target database in a TTS migration. Parallelism improves TTS export and import performance, especially when there are millions of database objects in the data files, including tables, indexes, partitions, and subpartitions.

Related Topics

- Oracle® Database Database Utilities
- **Data Pump Transportable Tbsp** lab in the Database 21c New Features workshop in LiveLabs

Oracle Data Pump Provides Optional Index Compression

In this release, Oracle Data Pump supports optional index compression on imports, including for Oracle Autonomous Database.

Oracle Data Pump supports adding, changing and eliminating table compression. Oracle Database 21c supports index compression as well by introducing a new `TRANSFORM` parameter clause, `INDEX_COMPRESSION_CLAUSE`. This clause enables you to control whether index compression is performed during import. Adding this clause also enables you to specify index compression on import with the autonomous services.

Related Topics

- Oracle® Database Database Utilities
- **Data Pump Index Compression** lab in the Database 21c New Features workshop in LiveLabs

Oracle Data Pump Resumes Transportable Tablespace Jobs

Starting with this release, Oracle Data Pump resumes transportable tablespace export and import jobs that are stopped.

Oracle Data Pump has the capacity to resume transportable tablespace export and import jobs. Due to errors, or other problems, you can find that transportable tablespace export or import jobs are stopped. Oracle Data Pump's capacity to resume these stopped jobs helps to save you time, and makes the system more available.

Related Topics

- [Oracle® Database Database Utilities](#)

Oracle Data Pump Supports Export to and Import From Cloud Object Stores

Starting with Oracle Database 21c, Oracle Data Pump `expdp` and `impdp` support use of object store URIs for the `DUMPFILE` parameter.

To use this feature for exports or imports from an object store, the `CREDENTIAL` parameter must be set. This feature eases migration to and from Oracle Cloud, because it relieves you of the extra step of transferring a dumpfile to or from the object store. Note that export and import performance is slower when accessing the object store, compared to local disk access, but the process is simpler. In addition, the process should be faster than running two separate export operations from Oracle Cloud, and transferring the dumpfile from the object store to an on premises location, or transferring the dumpfile from on premises to the object store, and then importing into Oracle Cloud.

Related Topics

- [Oracle® Database PL/SQL Packages and Types Reference](#)

Oracle Data Pump Supports Native JSON Datatypes

In this release, Oracle Data Pump enables export and import of Oracle Database native JSON objects.

Oracle Data Pump supports export and import of the new native JSON datatype in Oracle Database. This includes conventional and Transportable Tablespace (TTS) export and import of tables containing the JSON Datatype in full, tablespaces and table modes.

Oracle SQL*Loader Support for Object Store Credentials

Starting with this release, Oracle SQL*Loader accesses data in an object store by presenting user-defined credentials.

Oracle SQL*Loader now enables you to to specify credentials that you define, so that you can access and read data from files in an object store into Oracle Database.

Related Topics

- [Oracle® Database Database Utilities](#)

Oracle SQL*Loader Supports Native JSON Data Type

Oracle SQL*Loader enables JSON file data loading into Oracle Database native JSON objects.

Starting with this release, SQL*Loader conventional and direct path supports loading JSON data into the new native JSON data type in Oracle Database.

Related Topics

- Oracle® Database Database Utilities

Upgrades and Migration

- AutoUpgrade Automates Data Guard Operations During Database Upgrade
- AutoUpgrade Automates Steps Required for Oracle RAC Database Upgrade
- AutoUpgrade Automates Upgrade and Conversion of Non-CDB to PDB
- AutoUpgrade Automates Upgrade of a PDB via Unplug-Plug-Upgrade
- Oracle Database Automates Database Upgrades with AutoUpgrade

AutoUpgrade Automates Data Guard Operations During Database Upgrade

Starting with Oracle Database 21c, AutoUpgrade automates the steps necessary to upgrade a database in an Oracle Data Guard configuration.

Steps that AutoUpgrade automates includes disabling and re-enabling the Data Guard Broker (if present), and deferring log file transport while the upgrade takes place. Automation of these Oracle Data Guard operations in the context of a database upgrade eliminates the need for manual or scripted steps that you otherwise would have to perform yourself. This in turn makes it easier to automate upgrades of databases in Oracle Data Guard configurations. This feature particularly helps to automate upgrades in high-end environments, where there can be many hundreds of such databases that you upgrade as part of your upgrade project.

Related Topics

- Oracle® Database Database Upgrade Guide

AutoUpgrade Automates Steps Required for Oracle RAC Database Upgrade

AutoUpgrade automates all the steps needed to upgrade an Oracle Real Application Clusters (Oracle RAC) database.

AutoUpgrade automates steps that previously had to be performed manually or scripted. Steps that AutoUpgrade automates include the `SRVCTL` commands to stop and restart instances on multiple nodes, handling of `CLUSTER_DATABASE`, updating initialization parameter files, and restarting services, where appropriate.

This feature reduces the possibility of manual errors and decreases the work needed for DBAs to perform routine maintenance and upgrade activities.

Related Topics

- Oracle® Database Database Upgrade Guide

AutoUpgrade Automates Upgrade and Conversion of Non-CDB to PDB

To simplify upgrades to Oracle Database 21c, AutoUpgrade automates steps to convert a database from a non-CDB to a PDB as part of a database upgrade.

Oracle Database 21c requires multitenant architecture. With this release, you can use AutoUpgrade for non-CDB to PDB upgrades from Oracle Database 12c Release 2 (12.2) and later releases. Using the automation features of AutoUpgrade reduces both the effort required of DBAs for upgrades, and the risk associated with typographical errors in a manual process.

Related Topics

- Oracle® Database Database Upgrade Guide

AutoUpgrade Automates Upgrade of a PDB via Unplug-Plug-Upgrade

AutoUpgrade automates unplugging a PDB from a CDB, plugging it into a higher version CDB, and upgrading the PDB to that new target version.

As part of the process to simplify upgrade, you can upgrade pluggable databases (PDBs) using an unplug/plug, where the upgrade occurs when you plug the PDB into the target release container database (CDB). This feature adds another scenario in which AutoUpgrade can automate the entire database upgrade process.

Related Topics

- Oracle® Database Database Upgrade Guide

Oracle Database Automates Database Upgrades with AutoUpgrade

Oracle Database AutoUpgrade enables you to upgrade one or many databases without human intervention, all with one command, and with a single configuration file.

AutoUpgrade enables you to upgrade one or many Oracle Database instances at the command-line, using a single command and a single configuration file. AutoUpgrade runs the preupgrade tasks, performs automated fix-ups where needed, runs the database upgrade, and finishes by completing post-upgrade tasks. It includes automatic retry and fallback, the option to schedule upgrades for future points in time, and the ability to set, change, or remove initialization parameters as desired. Using AutoUpgrade can save you time and money by enabling you to upgrade hundreds of databases with one command, and avoid the need for high-maintenance upgrade solutions.

Related Topics

- Oracle® Database Database Upgrade Guide
- Hitchhiker's Guide for Upgrading to Oracle Database 19c Workshop

5
Management Solutions

- Diagnosability
- Manageability

Diagnosability

- Enhanced Diagnosability of Oracle Database
- SQL*Net Improved Diagnosability

Enhanced Diagnosability of Oracle Database

Diagnosability of database issues is enhanced through a new attention log, as well as classification of information written to database trace files. The new attention log is written in a structured format (XML or JSON) that is much easier to process or interpret and only contains information that requires attention from an administrator. Trace files now contain information that enables easier classification of trace messages.

Enhanced diagnosability features simplify database administration and improve data security.

Related Topics

- Oracle® Database Administrator's Guide

SQL*Net Improved Diagnosability

Starting with Oracle Database 21c, a connection identifier is available for each network connection. The connection identifier uniquely identifies a connection in trace and logs of different network elements and helps in correlating diagnostic data from these elements.

When a SQL*Net connection has multiple hops, such as from a client to Oracle Connection Manager (CMAN) and then to a server, correlating diagnostic information from the existing logs and traces becomes difficult. However, with the availability of a connection identifier, you can now easily correlate diagnostics, track network data traffic, and resolve connectivity errors.

Related Topics

- Oracle® Database Net Services Administrator's Guide

Manageability

- Persistent Memory Database
- DAX-Enabled File Systems

- New Database Initialization Parameters for Database Resident Connection Pooling (DRCP)
- Multi-Mount DBFS Client
- Near Zero Brownout for Planned Maintenance
- Oracle Grid Infrastructure SwitchHome
- Read-Only Oracle Home Default

Persistent Memory Database

The Persistent Memory Database (PMEM) support feature enables you to place database files directly on non-volatile memory. The underlying file store is FsDirect, a pointer-switching PMEM file system that supports atomic writes of Oracle Database data blocks. The FsDirect PMEM file store provides the external interface for map and access the Oracle database directly in persistent memory.

Queries can read directly from PMEM without first copying data into the database buffer cache, avoiding data redundancy and unnecessary I/O.

Oracle Persistent Memory Database is ideal for providing high performance with small-scale databases such as those used in Microservices, Sharding, and Read Replicas. Oracle Persistent Memory Database delivers higher transaction rates and fast response-time, achieving unprecedented levels of performance for small-scale systems.

Related Topics

- Oracle® Database Administrator's Guide

DAX-Enabled File Systems

You must use an XFS-based Data Analytics Accelerator (DAX) file system as the file store for the Persistent Memory (PMEM) database.

You can use a PMEM file store for database data files and control files. For performance reasons, Oracle recommends that you store redo log files as independent files in the file system.

Related Topics

- Oracle® Database Installation Guide for Linux

New Database Initialization Parameters for Database Resident Connection Pooling (DRCP)

New database initialization parameters, `DRCP_DEDICATED_OPT`, `MIN_AUTH_SERVERS`, and `MAX_AUTH_SERVERS`, have been added to configure Database Resident Connection Pooling (DRCP).

`MIN_AUTH_SERVERS` and `MAX_AUTH_SERVERS` allow the number of processes used to handle session authentication for DRCP to be configured for optimal usage. Authentication server statistics can be viewed in `V$AUTHPOOL_STATS`.

With DRCP, when the number of application connections to the broker is less than the maximum pool size, a "dedicated optimization" makes DRCP behave like dedicated servers. With this optimization, DRCP tends towards a one-to-one correspondence between application connections and DRCP server processes even if those processes are not currently doing database work. Setting `DRCP_DEDICATED_OPT` to `NO` turns off the optimization and reduces the tendency of the pool to grow towards its maximum size until necessary. This helps keep the number of DRCP server processes small when statement execution concurrency is low, therefore reducing memory usage on the database host.

Related Topics

- Oracle® Database Administrator's Guide

Multi-Mount DBFS Client

DBFS (Database File System) provides a file system interface for storing files/directories in the Oracle database. `dbfs_client`, helps in exposing a DBFS in a database user as a mount point for the OS. The current version of `dbfs_client` can service only the DBFS of a single database user.

Databases can have a number of PDBs each having their own DBFS that they use to store various files (e.g trace files, import dump, user files etc). There can be about 100 PDBs and hence a 100 DBFS to be serviced concurrently. In order to cater to this environment `dbfs_client` needs be able to service multiple DBFS owned by different users across databases. Currently a `dbfs_client` instance can service only one DBFS. Hence, to service 100 DBFS the same number of `dbfs_client` instances would be required. Currently in DBFS, there does not exist a single point of control that can manage all these different `dbfs_client` instances. This could make the management and monitoring of different client instances burdensome for database administrators.

This enhanced version of `dbfs_client` is capable of servicing DBFS of multiple database users. This means that `dbfs_client` can handle multiple mount points, each mount point services DBFS under one database user. This enhanced version of `dbfs_client` is referred to as MUMV (Multi User Mount Version) and the earlier version of `dbfs_client` is referred to as SUMV (Single User Mount Version). The new version of `dbfs_client` can be started in SUMV mode or MUMV mode. If started in SUMV mode, its behavior is the same as the earlier version.

DBFS provides a file system interface for storing files/directories in the Oracle database. Existing single-mount DBFS clients (`dbfs_client`) could mount only one user's DBFS file system. DBFS therefore required multiple DBFS client processes to support multiple file systems on the same host's DBFS file system. This enhanced DBFS multi-mode client can manage different mount points within a single DBFS client process. DBFS client with multi-mount support provides better ease of use and improved performance. Multi-mount DBFS client scales seamlessly to 100s of PDBs and DBFS client can be started as either MUMV (Multi User Mount Version) or SUMV (Single User Mount Version) mode.

Related Topics

- Oracle® Database SecureFiles and Large Objects Developer's Guide

Near Zero Brownout for Planned Maintenance

Planned maintenance and unplanned outages restart the database instances, but planned maintenance allows for preparation. Near Zero Brownout for Planned Maintenance reduces reconfiguration time for an instance targeted for a planned maintenance operation.

Near Zero Brownout for Planned Maintenance increases the availability of the database during online maintenance operations.

Related Topics

- Oracle® Real Application Clusters Administration and Deployment Guide

Oracle Grid Infrastructure SwitchHome

You can use the `-switchGridHome` option with `gridSetup.sh` to switch from one Oracle Grid Infrastructure home to another.

You can use the `-switchGridHome` option for patching and upgrading Oracle Grid Infrastructure. Use the `-switchGridHome` option to switch from the source Oracle Grid Infrastructure home to the patched Oracle Grid Infrastructure home. All Oracle Clusterware and Oracle Restart services start from the patched Oracle Grid Infrastructure home automatically.

Related Topics

- Oracle® Grid Infrastructure Grid Infrastructure Installation and Upgrade Guide

Read-Only Oracle Home Default

Read-only Oracle homes, where all configuration data and log files reside outside of the read-only Oracle home, are the default option for Oracle Database installations and upgrades.

Read-only Oracle homes enable an easy, flexible, and software-image based deployment of Oracle software that can automatically and seamlessly be distributed across multiple servers. Read-only Oracle homes also enable patching and updating of Oracle Database without extended downtime, as patching simply means replacing a given set of binaries in a defined location.

Related Topics

- Oracle® Database Database Installation Guide

6
Performance and High Availability

- ACFS Cluster File System
- Application Continuity
- Automatic Operations
- Automatic Storage Manager (ASM)
- Autonomous Health Framework
- Clusterware
- Database In-Memory
- Data Guard
- Flashback
- GoldenGate
- Multitenant
- Oracle Real Application Clusters (RAC)
- SecureFiles
- Sharding
- Transactional Event Queues (TEQs)

ACFS Cluster File System

- AutoShrink for ACFS
- Mixed Sector Support
- Oracle ACFS File Based Snapshots
- Replication Unplanned Failover
- AutoShrink for ACFS
- Mixed Sector Support
- Oracle ACFS File Based Snapshots
- Replication Unplanned Failover

AutoShrink for ACFS

Oracle Advanced Cluster File System (Oracle ACFS) automatic shrinking automatically shrinks an Oracle ACFS file system based on policy, providing there is enough free storage in the volume.

The benefit is an optimization in performance and space utilization, ensuring that the data migration and associated locking do not cause delays to the running workloads or timeouts.

Related Topics

- Oracle® Advanced Cluster File System Administrator's Guide

Mixed Sector Support

Oracle ACFS mixed sector support enables the Linux primary and accelerator volumes of an Oracle ACFS file system to use a mix of different logical sector sizes, such as 512-bytes and 4096 bytes.

The benefit is flexibility for storage configuration, enabling primary and accelerator volumes to have different logical sector sizes on Linux operating systems.

Related Topics

- Oracle® Advanced Cluster File System Administrator's Guide

Oracle ACFS File Based Snapshots

Oracle Advanced Cluster File System (Oracle ACFS) file-based snapshots provide the ability to create snapshots of individual Oracle ACFS files in a space efficient manner on Linux.

The benefit is storage efficiency because you only snapshot specific files, not all files in the file system. Example use cases are for virtual machine (VM) image files and PDB snapshot copies.

Related Topics

- Oracle® Advanced Cluster File System Administrator's Guide

Replication Unplanned Failover

Oracle ACFS replication failover provides unplanned failover where the standby location assumes the role of the primary in case of failure. When a failure occurs, the standby location pursues contact with the primary and in the absence of a response, the standby assumes the primary role, and on recovery of the former primary, the former primary becomes the new standby.

The benefit is faster recovery in the event of unplanned downtime for Oracle ACFS replication.

Related Topics

- Oracle® Advanced Cluster File System Administrator's Guide

AutoShrink for ACFS

Oracle Advanced Cluster File System (Oracle ACFS) automatic shrinking automatically shrinks an Oracle ACFS file system based on policy, providing there is enough free storage in the volume.

The benefit is an optimization in performance and space utilization, ensuring that the data migration and associated locking do not cause delays to the running workloads or timeouts.

Related Topics

- Oracle® Advanced Cluster File System Administrator's Guide

Mixed Sector Support

Oracle ACFS mixed sector support enables the Linux primary and accelerator volumes of an Oracle ACFS file system to use a mix of different logical sector sizes, such as 512-bytes and 4096 bytes.

The benefit is flexibility for storage configuration, enabling primary and accelerator volumes to have different logical sector sizes on Linux operating systems.

Related Topics

- Oracle® Advanced Cluster File System Administrator's Guide

Oracle ACFS File Based Snapshots

Oracle Advanced Cluster File System (Oracle ACFS) file-based snapshots provide the ability to create snapshots of individual Oracle ACFS files in a space efficient manner on Linux.

The benefit is storage efficiency because you only snapshot specific files, not all files in the file system. Example use cases are for virtual machine (VM) image files and PDB snapshot copies.

Related Topics

- Oracle® Advanced Cluster File System Administrator's Guide

Replication Unplanned Failover

Oracle ACFS replication failover provides unplanned failover where the standby location assumes the role of the primary in case of failure. When a failure occurs, the standby location pursues contact with the primary and in the absence of a response, the standby assumes the primary role, and on recovery of the former primary, the former primary becomes the new standby.

The benefit is faster recovery in the event of unplanned downtime for Oracle ACFS replication.

Related Topics

- Oracle® Advanced Cluster File System Administrator's Guide

Application Continuity

- Application Continuity Protection Check
- Session Migration
- Reset Session State
- Transparent Application Continuity
- Transparent Application Continuity in the Oracle Cloud

Application Continuity Protection Check

Application Continuity Protection Check (ACCHK) provides guidance on the level of protection for each application that uses Application Continuity and assists you to increase protection, if required.

ACCHK identifies which application configuration is protected to help you make an informed decision about which configuration to use for maximum protection or how to increase protection level for an application configuration. ACCHK also provides diagnostics for an unsuccessful failover.

Related Topics

- Oracle® Real Application Clusters Administrator's Guide

Session Migration

Oracle Database invokes planned failover at points where the database knows that it can replay the session using Application Continuity and that the session is not expected to drain.

Session Migration is an automatic solution that Oracle Database uses for relocating sessions during planned maintenance for batch and long-running operations that are not expected to complete in the specified drain timeout period.

Related Topics

- Oracle® Real Application Clusters Administration and Deployment Guide

Reset Session State

The reset session state feature clears the session state set by the application when the request ends. The `RESET_STATE` database service attribute cleans up dirty sessions, so that the applications, which use these sessions after cleanup, cannot see the state of these sessions.

The `RESET_STATE` feature enables you to clean the session state at the end of each request so that the database developers do not have to clean the session state manually. You can reset session state of the applications that are stateless between requests.

Related Topics

- Oracle® Real Application Clusters Administration and Deployment Guide

Transparent Application Continuity

In this release, the following new features are provided for Transparent Application Continuity:

- Oracle Database clients use implicit request boundaries when connecting to the database using a service that has the attribute `FAILOVER_TYPE` set to `AUTO`.

- Planned Failover is introduced, which is failover that is forced by the Oracle database at points where the Oracle database decides that the session can be

failed over and the session is unlikely to drain. This feature is also available for Application Continuity.

- There is also a new service attribute, `RESET_STATE`. The resetting of state is an important feature that clears the session state set by the application in a request at the end of request. This feature is also available for Application Continuity.

This improvement increases coverage of Transparent Application Continuity for applications that do not use an Oracle-supplied connection pool. Planned failover is used for shedding sessions during planned maintenance. It is also used for load rebalancing. Without `RESET_STATE`, application developers need to cancel their cursors, and clear any session state that has been set.

Related Topics

- Oracle® Database Development Guide

Transparent Application Continuity in the Oracle Cloud

Transparent Application Continuity is enabled by default in an Oracle Cloud environment. Enabling Transparent Application Continuity by default in the Oracle Cloud improves runtime performance, planned failover, and provides broader application coverage.

Transparent Application Continuity reduces overhead, reduces resource consumption, and broadens replay capabilities, so that database requests can replay in Oracle Cloud. Transparent Application Continuity also ensures Continuous Availability for applications working with databases in the Oracle Cloud.

Related Topics

- Oracle® Real Application Clusters Administration and Deployment Guide

Automatic Operations

- Automatic Indexing Enhancements
- Automatic Index Optimization
- Automatic Materialized Views
- Automatic SQL Tuning Set
- Automatic Temporary Tablespace Shrink
- Automatic Undo Tablespace Shrink
- Automatic Zone Maps
- Object Activity Tracking System
- Sequence Dynamic Cache Resizing

Automatic Indexing Enhancements

Automatic indexing considers more cases for potential indexes and allows inclusion or exclusion of specific tables. An enhancement has been introduced to reduce the overhead of cursor invalidations when a new automatic index is created.

The enhancements increase the number cases where automatic indexes improve query performance.

Related Topics

• Oracle® Database Administrator's Guide

Automatic Index Optimization

ADO Policies for Indexes extends existing Automatic Data Optimization (ADO) functionality to provide compression and optimization capability on indexes. Customers of Oracle Database are interested in leveraging compression tiering and storage tiering to satisfy their Information Lifecycle Management (ILM) requirements. The existing ADO functionality enables you to set policies that enforce compression tiering and storage tiering for data tables and partitions automatically, with minimal user intervention.

In a database, indexes can also consume a significant amount of database space. Reducing the space requirement for indexes, without sacrificing performance, requires ILM actions similar to the existing Automatic Data Optimization feature for data segments. Using this new Index compression and optimization capability, the same ADO infrastructure can also automatically optimize indexes. Similar to ADO for data segments, this automatic index compression and optimization capability achieves ILM on indexes by enabling you to set policies that automatically optimize indexes through actions like compressing, shrinking, and rebuilding indexes.

Related Topics

• Oracle® Database VLDB and Partitioning Guide

Automatic Materialized Views

Materialized views offer the potential to improve query performance significantly, but considerable effort and skill is required to identify what materialized views to use. The database now incorporates workload monitoring to establish what materialized views are needed. Based on the decisions it makes, materialized views and materialized view logs are created and maintained automatically without any manual interaction.

Automatic materialized views improve application performance transparently without the need for any user action or management overhead.

Related Topics

• Oracle® Database Performance Tuning Guide

Automatic SQL Tuning Set

The automatic SQL tuning set (ASTS) is a system-maintained record of SQL execution plans and SQL statement performance metrics seen by the database. Over time, the ASTS will include examples of all queries seen on the system.

SQL plans and metrics stored in ASTS are useful for repairing SQL performance regressions quickly using SQL plan management.

ASTS enables SQL performance regressions to be resolved quickly and with minimal manual intervention. It is complementary to the automatic workload repository (AWR) and considered a similar core manageability infrastructure of the Oracle Database.

Related Topics

- Oracle® Database SQL Tuning Guide

Automatic Temporary Tablespace Shrink

A temporary tablespace can grow to a very large size due to spikes in temp usage by queries. Sorts, hash joins, and query transformations are examples that might cause high temp usage. Automatic Temp Tablespace Sizing is a feature which shrinks the temporary tablespace in the background when temp usage has subsided. In addition, if temp usage is increasing, the feature would pre-emptively grow the temporary tablespace to ensure query performance is not impacted.

This feature alleviates the need for a DBA to manually size the temporary tablespace. These are the two scenarios which would benefit:

- The DBA does not need to manually shrink the temporary tablespace to reclaim unsued space.

- The DBA does not need to manually grow the temporary tablespace in anticipation of high temp usage.

Automatic Undo Tablespace Shrink

An undo tablespace can grow to a very large size and then that space may not be needed again. Currently there is no automated way to recover that space and there have been limits placed on the size of the undo tablespace in some environments because space cannot be easily reclaimed. This can prevent large transactions from running successfully. Automatic Undo Tablespace Shrink is a feature that shrinks the undo tablespace in the background by dropping expired (i.e. manual or automatically configured) undo segments and their corresponding extents, and then performing a datafile shrink if possible. A datafile shrink is not guaranteed depending on where allocated extents are located. Even if datafile shrink is not possible, releasing undo extents back to the undo tablespace enables the reuse of space by other undo segments.

This feature reduces space requirements and the need for manual intervention:

- Automatically recovering space used by transactions that are no longer active

- Remove the need to restrict the size of the undo tablespace allowing larger transactions to run with the ability to recover that space once the transaction undo expires.

Automatic Zone Maps

Automatic zone maps are created and maintained for any user table without any customer intervention. Zone maps allow the pruning of block ranges and partitions based on the predicates in the queries. Automatic zone maps are maintained for direct loads, and are maintained and refreshed for any other DML operation incrementally and periodically in the background.

Automatic zone maps improve the performance of queries transparently and automatically without management overhead.

Related Topics

- [Oracle® Database Data Warehousing Guide](#)
- **Automatic Zone Maps** lab in the [Database 21c New Features](#) workshop in LiveLabs

Object Activity Tracking System

Object Activity Tracking System (OATS) tracks the usage of various types of database objects. Usage includes operations such as access, data manipulation, or refresh.

Automated tracking of how database objects are being used enables customers to gain a better insight into how applications are querying and manipulating the database and its objects. Internal clients such as Access Advisors or Automatic Materialized Views leverage and benefit from OATS as well.

Related Topics

- [Oracle® Database Performance Tuning Guide](#)

Sequence Dynamic Cache Resizing

With dynamic cache resizing, the sequence cache size is now auto-tuned based on the rate of consumption of sequence values. This means the cache can automatically grow and shrink over time, depending on usage, while never falling below the DDL specified cache size. By dynamically resizing the cache, performance can be improved significantly, especially for fast insert workloads on Oracle RAC, by reducing the number of trips to disk needed to replenish the cache.

Dynamic cache resizing can improve performance significantly for fast insert workloads that use sequences. This is accomplished by reducing the number of trips to disk needed to replenish the sequence cache and can be especially significant in an Oracle RAC environment.

Related Topics

- [Oracle® Database Database Administrator's Guide](#)

Automatic Storage Manager (ASM)

- [Enable ASMCA to Configure Flex ASM on an Existing NAS Configuration](#)
- [Enhanced Double Parity Protection for Flex and Extended Disk Groups](#)
- [File Group Templates](#)
- [Oracle ASM Flex Disk Group Support for Cloning a PDB in One CDB to a New PDB in a Different CDB](#)

Enable ASMCA to Configure Flex ASM on an Existing NAS Configuration

This feature enables you to install Oracle Flex ASM on a configuration previously configured on network file storage (NFS). In particular, Oracle ASM Configuration

Assistant (ASMCA) can be run in silent mode to configure Oracle ASM after an Oracle Clusterware installation has been performed using network attached storage (NAS) for Oracle Cluster Registry (OCR) and Voting disks.

The business value of this feature is that it provides an easy way for you to transition from NFS storage over to Oracle ASM managed storage. Without this feature, you would have to do a complete fresh installation and move all databases.

Related Topics

- Oracle® Automatic Storage Management Administrator's Guide

Enhanced Double Parity Protection for Flex and Extended Disk Groups

This feature provides support for double parity protection for write-once files in an Oracle Automatic Storage Management (Oracle ASM) Flex Disk Group.

With this feature you can use double parity protection for write-once files in a Oracle ASM Flex Disk Group. Double parity protection provides greater protection against multiple hardware failures. A previous release of Oracle ASM provided for simple parity protection for write-once files in a Flex Disk Group. Write-once files include files such as database backup sets and archive logs. The benefit of parity protection as compared to conventional mirroring is that it reduces storage overhead, but with a slight increase of risk of data loss after an event involving multiple hardware failures.

Related Topics

- Oracle® Automatic Storage Management Administrator's Guide

File Group Templates

With file group templates you can customize and set default file group properties for automatically created file groups, enabling you to customize file group properties that are inherited by a number of databases.

Without file group templates, if you wanted to change properties for an automatically created file group, you would have to manually change the properties after the associated files are created which triggers an unnecessary rebalance. The file group templates feature provides a much better option.

Related Topics

- Oracle® Automatic Storage Management Administrator's Guide

Oracle ASM Flex Disk Group Support for Cloning a PDB in One CDB to a New PDB in a Different CDB

Previously point-in-time database clones could only clone a pluggable database (PDB) in a multitenant container database (CDB) to a new PDB in the same CDB. The latter restriction is removed as part of this feature. Now, you can clone a PDB in a CDB to a new PDB in a different CDB.

This feature enables you to use Oracle ASM cloning for test and development cloning where the cloned PDB must be in a separate CDB.

Related Topics

- [Oracle® Automatic Storage Management Administrator's Guide](#)

Autonomous Health Framework

- [Enhanced Support for Oracle Exadata](#)
- [Oracle Cluster Health Advisor Support for Solaris](#)
- [Oracle Cluster Health Monitor Local Mode Support](#)
- [Oracle ORAchk and EXAchk Support for REST API](#)
- [Oracle Trace File Analyzer Real-Time Health Summary](#)
- [Oracle Trace File Analyzer Support for Efficient Multiple Service Request Data Collections](#)
- [Remote GIMR Support for Oracle Standalone Clusters](#)
- [Support for Automatically Enabling Oracle Database Quality of Service (QoS) Management](#)
- [Support for Deploying Grid Infrastructure Management Repository (GIMR) into a Separate Oracle Home](#)

Enhanced Support for Oracle Exadata

The ability of Oracle Cluster Health Advisor to detect performance and availability issues on Oracle Exadata systems has been improved in this release with the addition of Exadata specific models.

Oracle Cluster Health Advisor detects performance and availability issues using Oracle Database and node models that were developed using machine learning.

With the improved detection of performance and availability issues on Oracle Exadata systems, Oracle Cluster Health Advisor helps to improve Oracle Database availability and performance. New Exadata-specific models are automatically loaded when CHA runs on Exadata Engineered Systems.

Related Topics

- [Oracle® Autonomous Health Framework User's Guide](#)

Oracle Cluster Health Advisor Support for Solaris

Oracle Cluster Health Advisor supports Oracle Real Application Clusters (Oracle RAC) deployments on Oracle Solaris.

With the Oracle Cluster Health Advisor support for Oracle Solaris, you can now get early detection and prevention of performance and availability issues in your Oracle RAC database deployments.

Related Topics

- [Oracle® Autonomous Health Framework User's Guide](#)

Oracle Cluster Health Monitor Local Mode Support

You can now configure Oracle Cluster Health Monitor to operate in local mode to report the operating system metrics using the `oclumon dumpnodeview` command even if you have not deployed Grid Infrastructure Management Repository (GIMR).

In local mode, you can get only the local node data. The local mode has limited Oracle Cluster Health Monitor functionality in the deployments where you have not installed Grid Infrastructure Management Repository (GIMR). In earlier releases, Oracle Cluster Health Monitor required GIMR to report the operating system metrics using the `oclumon dumpnodeview` command.

Related Topics

- [Oracle® Autonomous Health Framework User's Guide](#)

Oracle ORAchk and EXAchk Support for REST API

In this release, support for REST API adds a remote interface to the existing Oracle ORAchk and Oracle EXAchk command-line interfaces (CLI).

You can manage Oracle software deployments remotely from centralized consoles and web interfaces. By supporting the REST interfaces, Oracle ORAchk and Oracle EXAchk integrate into these applications and help support fleet or cloud management.

Related Topics

- [Oracle® Autonomous Health Framework User's Guide](#)

Oracle Trace File Analyzer Real-Time Health Summary

Oracle Trace File Analyzer generates a real-time health summary report, which shows performance degradation due to faults and workload issues.

Similar to the status scorecard of the deployment configurations that Oracle ORAchk and Oracle EXAchk generate, Oracle Trace File Analyzer also provides a readily consumable and trackable scoring for operational status. The health summary consists of scores in the categories of availability, health, workload, and capacity broken down from cluster-wide through the database, instance, service, and hardware resource.

Related Topics

- [Oracle® Autonomous Health Framework User's Guide](#)

Oracle Trace File Analyzer Support for Efficient Multiple Service Request Data Collections

Oracle Trace File Analyzer collects multiple Service Request Data Collections into a single collection even if it detects multiple issues or errors at the same time.

Service Request Data Collection mode of operation enables you to collect only the log and trace files that are required for diagnosing a specific type of problem. Even with this optimization, Oracle Trace File Analyzer collects the same subset of files if it detects multiple

issues or errors at the same time. The enhancement further optimizes the collection of multiple Service Request Data Collections into a single collection and thus removes duplication.

It is essential to collect log and trace files upon detection of issues before the files are rotated or purged. However, collecting log and trace files involves resource overhead, which may be critically low due to these issues. The enhancement in this release reduces the resource overhead and disk space needed at a critical time.

Related Topics

- Oracle® Autonomous Health Framework User's Guide

Remote GIMR Support for Oracle Standalone Clusters

The remote Grid Infrastructure Management Repository (GIMR) feature for Oracle Standalone Cluster enables you to use a centralized GIMR. This feature does not require local cluster resources to host the GIMR.

The remote GIMR feature provides access to a persistent data store that significantly enhances the proactive diagnostic functionality of Cluster Health Monitor, Cluster Health Advisor, and Autonomous Health Framework clients. The remote GIMR feature saves cost by freeing up local resources and licensed database server resources.

Related Topics

- Oracle® Grid Infrastructure Grid Infrastructure Installation and Upgrade Guide

Support for Automatically Enabling Oracle Database Quality of Service (QoS) Management

Oracle Database Quality of Service (QoS) Management automatically configures a default policy set based upon the services it discovers and begins monitoring in measurement mode.

With this implementation, the workload performance data is always available to you and other Oracle Autonomous Health Framework components.

If you do not have Oracle Enterprise Manager deployed to monitor Oracle Database clusters, then you cannot utilize the functionality of Oracle Database QoS Management because you cannot enable it with Enterprise Manager. With automatic monitoring, you can now take advantage of the rich set of workload data provided.

In conjunction with the new REST APIs, you can integrate the advanced Oracle Database QoS Management modes into your management systems. In earlier releases, you have to configure the monitoring functionality of Oracle Database QoS Management and enable Oracle Database QoS Management with Enterprise Manager.

Related Topics

- Oracle® Autonomous Health Framework User's Guide

Support for Deploying Grid Infrastructure Management Repository (GIMR) into a Separate Oracle Home

Starting with this release of Oracle Grid Infrastructure, you must configure the Grid Infrastructure Management Repository (GIMR) in a separate Oracle home, instead of in the Grid home. This option is available when you configure GIMR during a fresh Oracle Grid Infrastructure installation or you add a GIMR to an existing deployment. It is mandatory to configure GIMR in a separate Oracle home when you upgrade Oracle Grid infrastructure with an existing GIMR deployed in it.

A separate Oracle home for the GIMR ensures faster rolling upgrades, fewer errors, and fewer rollback situations. The Oracle Grid Infrastructure installation owner user must own the GIMR home.

Related Topics

- Oracle® Grid Infrastructure Grid Infrastructure Installation and Upgrade Guide

Clusterware

- Clusterware REST API
- Common Data Model Across Fleet Patching and Provisioning Servers
- FPP Integration with AutoUpgrade
- Oracle Clusterware 21c Deprecated and Desupported Features

Clusterware REST API

The Clusterware REST APIs enable customers to remotely execute commands on their cluster and to monitor the execution, including output, error codes, and time to execute. Support is provided for existing Oracle Clusterware command line interfaces.

REST API-based management based on well-known Oracle Clusterware command line interfaces simplifies cluster management in the Oracle Cloud, at remote physical locations or locally provisioned.

Related Topics

- Oracle® Database REST APIs for Oracle Database

Common Data Model Across Fleet Patching and Provisioning Servers

The common data model across Fleet Patching and Provisioning (FPP) servers provides a unified view of fleet targets regardless of the FPP server deployment.

The unified view of the common model provides an easier and simplified operation of large estates and cloud management across multiple data centers.

Related Topics

- Oracle® Clusterware Administration and Deployment Guide

FPP Integration with AutoUpgrade

Fleet Patching and Provisioning (FPP) integration with AutoUpgrade provides a new tool for automating and simplifying Oracle Database Upgrade.

This feature makes Oracle Database AutoUpgrade more flexible, provides better control over the upgrade flow mechanism, and provides better usability by showing progress bar and additional elements. You can upgrade multiple databases in parallel.

Related Topics

- Oracle® Fleet Patching and Provisioning Administrator's Guide

Oracle Clusterware 21c Deprecated and Desupported Features

The following are deprecated and desupported features in Oracle Clusterware 21c:

- Deprecation of Policy-Managed Databases: Starting with Oracle Grid Infrastructure 21c, creation of new server pools is eliminated, and policy-managed databases are deprecated and can be desupported in a future release. Server pools will be migrated to the new Oracle Services feature that provides similar functionality.

- Deprecation of Cluster Domain - Domain Services Cluster: With the introduction of Oracle Grid Infrastructure 21c, Domain Services Cluster (DSC), which is part of the Oracle Cluster Domain architecture, are deprecated and can be desupported in a future release.

- Deprecation of Policy-Managed Databases

- Vendor Clusterware Integration with Oracle Clusterware has been desupported: Starting with Oracle Oracle Clusterware 21c , the integration of vendor or third party clusterware with Oracle Clusterware is desupported.

- Desupport of Cluster Domain - Member Clusters: Member Clusters, which are part of the Oracle Cluster Domain architecture, are desupported. However, Domain Services Clusters continue to support Members Clusters in releases previous to Oracle Grid Infrastructure 21c.

Database In-Memory

In addition to the features highlighted in the Database In-Memory section, the following features in other sections are applicable for Database In-Memory:

- In-Memory Full Text Columns

- Spatial Support for Database In-Memory

- In-Memory Deep Vectorization

- New JSON Data Type

- Database In-Memory Base Level

- Automatic In-Memory

- Database In-Memory External Table Enhancements

- In-Memory Hybrid Scans

- CellMemory Level
- Database In-Memory Additional Features

Database In-Memory Base Level

Database In-Memory is an option to Enterprise Edition and now has a new "Base Level" feature. This allows the use of Database In-Memory with up to a 16GB column store without having to license the option. The use of the Base Level features does not trigger any license tracking.

The IM column store is limited to 16GB when using the Base Level feature. This can allow customers to see the value of Database In-Memory without having to worry about licensing issues. Note that Base Level has some other restrictions; for instance, the CellMemory feature and Automatic In-Memory are not included with the base level.

Related Topics

- Oracle® Database In-Memory Guide

Automatic In-Memory

Automatic In-Memory (AIM) enables, populates, evicts, and recompresses segments without user intervention.

When `INMEMORY_AUTOMATIC_LEVEL` is set to `HIGH` the database automatically populates segments based on their usage patterns without requiring them to be marked `INMEMORY`. Combined with support for selective column level recompression, In-Memory population is largely self-managing. This automation helps maximize the number of objects that can be populated into the In-Memory column store (IM column store) at one time and maximizes overall application performance.

Related Topics

- Oracle® Database In-Memory Guide
- **Automatic In-Memory** lab in the Database 21c New Features workshop in LiveLabs

Database In-Memory External Table Enhancements

For a partitioned external table or a hybrid partitioned table (a table which has both internal and external partitions), the `INMEMORY` clause is supported at both the table and partition level. For hybrid partitioned tables, the table-level `INMEMORY` attribute applies to all partitions, whether internal or external.

This enhancement significantly broadens support for in-memory external tables by allowing partitioned external tables or hybrid partitioned tables to be selectively populated by partition thereby supporting active subsets of external data rather than having to populate the entire table.

Related Topics

- Oracle® Database In-Memory Guide

In-Memory Hybrid Scans

In-Memory hybrid scans support in-memory scans when not all columns in a table have been populated into the In-Memory column store (IM column store). This situation can occur when columns have been specified as `NO INMEMORY` to save space. In previous releases if a column that was not populated was accessed by a query then no data could be accessed from the IM column store.

The In-Memory hybrid scan feature allows a query to combine in-memory accesses with row-store accesses, improving performance by orders of magnitude over pure row store queries.

Related Topics

- Oracle® Database In-Memory Guide

- **In-Memory Hybrid Scans** lab in the Database 21c New Features workshop in LiveLabs

CellMemory Level

On Exadata systems you can use both memory in the compute servers for the In-Memory column store and the Exadata Smart Flash Cache in the storage servers to populate data in in-memory columnar format (a feature known as *CellMemory*). CellMemory is enabled by default if the IM column store is enabled on the database servers. You can now selectively enable only CellMemory without enabling the IM column store, by setting `INMEMORY_FORCE=CELLMEMORY_LEVEL` and `INMEMORY_SIZE=0`.

If you do not intend to use Database In-Memory on the Database servers, this feature allows you to use CellMemory without incurring the overhead of allocating SGA memory for the IM column store.

Related Topics

- Oracle® Database In-Memory Guide

Database In-Memory Additional Features

In addition to the features highlighted in the Database In-Memory section, the following features in other sections are applicable for Database In-Memory:

- In-Memory Full Text Columns

- Spatial Support for Database In-Memory

- In-Memory Vectorized Joins

- New JSON Data Type

Data Guard

- Active Data Guard - Standby Result Cache

- Data Guard Broker Far Sync Instance Creation

- Data Guard Far Sync instance in Maximum Performance Mode

- Fast-Start Failover Callouts

Active Data Guard - Standby Result Cache

The result cache in an Active Data Guard standby database is utilized to cache results of queries that were run on the physical standby database. In the case of a role transition to primary, the standby database result cache will now be preserved ensuring performance for offloaded reporting and other queries continue without compromising the performance benefits of the standby result cache.

Use of the result cache greatly improves query performance for recurring queries and minimizes performance impact on the primary and standby databases. By maintaining the result cache on the standby, the performance of any queries that were running on the standby will be maintained ensuring previously offloaded reporting and other read-only applications utilizing the standby will not impacted by the role transition.

Related Topics

- Oracle® Data Guard Concepts and Administration

Data Guard Broker Far Sync Instance Creation

The Data Guard Broker now enables users to create and add a Far Sync instance to a Data Guard Broker configuration using a single command.

Zero Data loss over long distance can be achieved by using the Data Guard Far Sync standby instances. To ease the setup and maintenance of these instances, the Oracle Data Guard Broker can now be utilized. This leads to easier and simplified setup, which leverages the existing maintenance solution of the overall Data Guard environment.

Related Topics

- Oracle® Data Guard Broker

Data Guard Far Sync instance in Maximum Performance Mode

The far sync instance can fully be utilized in Maximum Performance mode in both normal configurations as well as when fast-start failover (FSFO) is enabled.

This additional flexibility allows RTO/RPO objectives to be met more easily when far sync is utilized in Active Data Guard configurations. Maximum Performance mode provides for a predefined data loss, but with the benefit of a faster automated failover, which is critical in disaster recovery events.

Related Topics

- Oracle® Data Guard Broker

Fast-Start Failover Callouts

Callout scripts are now available for use with fast-start failover (FSFO). These scripts can contain user-defined commands that are run before and after a FSFO operation.

With the additional flexibility provided by callout scripts, administrators can automate manual actions that must be performed before FSFO events are initiated as well as after a FSFO event has been completed. This provides a more consistent and adaptable configuration reducing the chance for human error to be introduced these events.

Related Topics

• Oracle® Data Guard Broker

Fast-Start Failover Configuration Validation

Oracle Data Guard Broker now provides early detection of fast-start failover (FSFO) configuration issues and reports those mis-configurations allowing administrators to take action prior to a failover event.

Monitoring and validating a fast-start failover configuration helps maintain and ensure database availability. Potential configuration errors are detected early thereby preventing problems prior to a fast-start failover event that may be required to protect the configuration. This ensures that administrators can confidently maintain and validate their fast-start failover configuration in cases of new deployments as well as updates to an existing configuration.

Related Topics

• Oracle® Data Guard Broker

PDB Recovery Isolation

PDB Recovery Isolation ensures single PDB recovery while managed standby recovery is recovering other PDBs and prevents administrative PDB operations on the primary database from interfering with recovery of other PDBs on the standby database.

This preserves the PDB isolation principle in regards to Active Data Guard allowing the maintenance, protection, and query SLAs for remaining PDBs to continue unabated which is consistent with how the primary database handles PDB operations.

Related Topics

• Oracle® Data Guard Concepts and Administration

Standardized Data Guard Broker Directory Structure

Oracle Data Guard broker now utilizes a standardized directory structure to store client-side files.

Using a standardized directory structure helps keep your Oracle Data Guard environment well organized and consistent. Management is further simplified by using configuration aliases to quickly identify a specific Oracle Data Guard configuration.

Related Topics

- [Oracle® Data Guard Broker](#)

Other Features

- [Callout Configuration Scripts](#)
- [Data Guard Broker Managed Default Directory](#)
- [Oracle Database 21c Data Guard Desupported Features](#)
- [FastStartFailoverLagLimit Configuration Property](#)
- [PREPARE DATABASE FOR DATA GUARD Command](#)
- [VALIDATE FAST_START FAILOVER Command](#)

Callout Configuration Scripts

Callout configuration scripts can be used to automatically execute specified tasks before and after a fast-start failover operation. The name of the callout configuration file is `fsfocallout.ora`. You cannot use a different name for this file. This file is stored in the `$DG_ADMIN/config_ConfigurationSimpleName/callout` directory. If the `DG_ADMIN` environment variable is not defined, or the directory specified by this variable does not exist, or the directory does not have the required permissions, fast-start failover callouts will fail.

The name of the callout configuration scripts is specified in `fsfocallout.ora`. These scripts must be in the same directory as the callout configuration file. You can create two callout configuration scripts, a ***pre-callout*** configuration script and ***post-callout*** configuration script. Before a fast-start failover operation, the observer checks if a fast-start failover configuration file exists. If it exists, and it contains a pre-callout script location, this script is run before the fast-start failover is initiated. After fast-start failover succeeds, if a post-callout script is specified in the fast-start failover configuration file this script is run.

The `VALIDATE FAST_START FAILOVER` command parses the callout configuration scripts and checks for errors or misconfigurations.

To perform specified actions before or after a fast-start failover operation:

1. Create a pre-callout script, or a post-callout script, or both.

2. Create or update the fast-start failover callout configuration file and include the names of the scripts created in the previous step.

Callout Configuration Script Example

The following example displays the contents of the fast-start failover configuration file named `/home1/dataguard/config_NorthSales/callout/fsfocallout.ora`. The `fsfo_precallout` and `fsfo_postcallout` callout configuration scripts are stored in the same location as `fsfocallout.ora` with the required permissions.

```
# The pre-callout script that is run before fast-start failover is enabled.
FastStartFailoverPreCallout=fsfo_precallout

# The timeout value (in seconds) for pre-callout script.
```

```
FastStartFailoverPreCalloutTimeout=1200

# The name of the suc file created by the pre-callout script.
FastStartFailoverPreCalloutSucFileName=fsfo_precallout.suc

# The name of the error file that the pre-callout script creates.
FastStartFailoverPreCalloutErrorFileName=precallout.err

# Action taken by observer if the suc file does not exist after
FastStartFailoverPreCalloutTimeout seconds or if an error file is detected
before FastStartFailoverPreCalloutTimeout seconds passed.
FastStartFailoverActionOnPreCalloutFailur=STOP

# The post-callout script that is run after fast-start failover succeeds.
FastStartFailoverPostCallout=fsfo_postcallout
```

Related Topics

- Oracle® Data Guard Broker

Data Guard Broker Managed Default Directory

Starting with Oracle Database 21c, the `DG_ADMIN` environment variable is used to specify the default location for client-side broker files.

Client-side broker files include the following:

- Observer configuration file (`observer.ora`)

- Observer log file

- Fast-start failover log file (`fsfo.dat`)

- Fast-start failover callout configuration file

The files are stored in subdirectories created under `DG_ADMIN`. You must create the directory specified in `DG_ADMIN` and ensure that it has the required permissions.

Content for default directory for client-side files is as follows:

- **admin:** Contains the observer configuration file used by DGMGRL to manage multiple observers. This file also declares broker configurations and defines configuration groups used by multiple configuration commands. The default name of the observer configuration file is `observer.ora`. When DGMGRL starts, if the `DG_ADMIN` environment variable is set and the specified directory has the required permissions, the `admin` directory is created under `DG_ADMIN`. When commands that need access to the observer configuration file are run, such as `START OBSERVING`, `STOP OBSERVING`, `SET MASTEROBSERVER TO`, and `SET MASTEROBSERVERHOSTS`, DGMGRL reports an error if the directory does not have the required permissions.

- **config_ConfigurationSimpleName:** Stores files related to the observer and callout configuration. This directory has the same permissions as its parent directory. For each broker configuration on which one or more observers are registered, a directory named `ConfigurationSimpleName` is created. `ConfigurationSimpleName` represents an alias of the broker configuration name. Subdirectories within this directory are used to store the files related to the configuration. This directory is created when you run the `CONNECT` command. When running the `START OBSERVER` command, if this directory does not have the required permissions, DGMGRL reports an error.

- **config_ConfigurationSimpleName/log:** Contains the observer log file for the broker configuration named `ConfigurationSimpleName`. The default name of the observer log file is `observer_hostname.log`.

- **config_ConfigurationSimpleName/dat:** Contains the observer runtime data file for the broker configuration named `ConfigurationSimpleName`. The default name of the observer runtime data file is `fsfo_hostname.dat`. This file contains important information about the observer. In the event of a crash, data in this file can be used to restart the observer to the status before the crash.

- **config_ConfigurationSimpleName/callout:** Contains the callout configuration file, precallout script, post-callout script, and precallout success file for the broker configuration named `ConfigurationSimpleName`. The default name of the callout configuration file is `fsfocallout.ora`.

Permissions Required by the DG_ADMIN Directory

On Linux/Unix platforms, the directory specified by the `DG_ADMIN` environment variable must have read, write, and execute permissions for the directory owner only. The subdirectories that DGMGRL creates under this directory will also have the same permissions.

On Windows platforms, the directory specified by the `DG_ADMIN` environment variable must have exclusive permissions wherein it can be accessed only by the current operating system user who is running DGMGRL The subdirectories created under this directory by DGMGRL will also have the same permissions.

If the `DG_ADMIN` environment variable is not set or the specified directory does not exist or the permissions are different from the ones specified above, then broker does the following:

- Stores the fast-start failover log file and observer configuration file in the current working directory

- Uses standard output for displaying the observer logs

Every time DGMGRL starts, it checks if the default directories for client-side files exist. If they do not exist, the subdirectories are created.

- When you run DGMGRL commands, if a path name and file name are explicitly specified for client-side files, the specified values are used.

- If only a file name is specified, the file is stored in an appropriate directory under the broker's `DG_ADMIN` directory.

- If only a path is specified, the files are stored in the specified path using the default file names.

- If `DG_ADMIN` is not set, the then files are stored in the current working directory.

Related Topics

- Oracle® Data Guard Broker

Oracle Database 21c Data Guard Desupported Features

The following parameters that were deprectated in Oracle Database Release 19c are now desupported features in Data Guard and Data Guard Broker:

- ArchiveLagTarget
- DataGaurdSyncLatency
- LogArchiveMaxProcesses

- LogArchiveMinSucceedDest

- LogArchiveTrace

- StandbyFileManagement

- DbfileNameConvert

- LogArchiveFormat

- LogFileNameConvert

- LsbyMaxEventsRecorded

- LsbyMaxServers

- LsbyMaxSga

- LsbyPreserveCommitOrder

- LsbyRecordAppliedDdl

- LsbyRecordSkipDdl

- LsbyRecordSkipErrors

- LsbyParameters

FastStartFailoverLagLimit Configuration Property

When no synchronous standby destinations are available, a standby that uses asynchronous redo transport can be used as a fast-start failover target provided the new `FastStartFailoverLagLimit` configuration property is set. When a synchronous standby becomes available, the broker automatically switches back to the synchronous standby.

The `FastStartFailoverLagLimit` configuration property establishes an acceptable limit, in seconds, that the standby is allowed to fall behind the primary in terms of redo applied. If the limit is reached, then a fast-start failover is not allowed. The lowest possible value is 5 seconds. This property is used when fast-start failover is enabled and the configuration is operating in maximum performance mode.

The following categories are for the `FastStartFailoverLagLimit` configuration property:

- Datatype: Integer

- Valid value: Integral number of seconds. Must be greater than or equal to 5

- Broker default: 30 seconds

- Imported?: No

- Parameter class: Not applicable

- Role: Primary and standby

- Standby type: Not applicable

- Corresponds to: Not applicable

- Scope: Broker configuration. This property will be consumed by the primary database after fast-start failover has been enabled.

- Cloud control name: Lag Limit

Related Topics

- Oracle® Data Guard Broker

PREPARE DATABASE FOR DATA GUARD Command

You can configure a database for Data Guard using a single command. The `PREPARE DATABASE FOR DATA GUARD` command configures a database and sets it up to be used as a primary database in a Data Guard broker configuration. You must connect to the primary database as a user with the `SYSDG` or `SYSDBA` privilege. Database versions starting from Oracle Database 12c Release 2 are supported. For a single-instance database, if a server parameter file does not exist, it is created using the current in-memory parameter settings and stored in the default location.

The `PREPARE DATABASE FOR DATA GUARD` command is as follows:

```
PREPARE DATABASE FOR DATA GUARD [WITH DB_UNIQUE_NAME IS dbunique-name]
[DB_RECOVERY_FILE_DEST IS directory-location]
[DB_RECOVERY_FILE_DEST_SIZE is size] [BROKER_CONFIG_FILE_1 IS
brokerconfig-file-1-location] [BROKER_CONFIG_FILE_2 IS
broker-config-file-2-location]
```

Command Parameters

`db_unique_name`: The value for the `DB_UNIQUE_NAME` initialization parameter. If the initialization parameter has been set to a different value, the existing value is replaced with the value specified by `db_unique_name`. If this parameter is not specified, the `DB_UNIQUE_NAME` parameter is set to the value of the `DBNAME` parameter.

`directory-location`: The directory name for the `DB_RECOVERY_FILE_DEST` initialization parameter, which represents the fast recovery area location. The specified directory must be accessible by all instances of a RAC database. This parameter can be omitted if a local archive destination is set. However, if the `DB_RECOVERY_FILE_DEST` initialization parameter has not been set and no local archive destination has been set, specifying this parameter is mandatory. If `directory_location` is specified, a `log_archive_dest_n` initialization parameter is set to the value `USE_DB_RECOVERY_FILE_DEST`. This is done whether or not there is a local archive destination already set.

`size`: A size value for the `DB_RECOVERY_FILE_DEST` initialization parameter. This parameter is mandatory if `DB_RECOVERY_FILE_DEST` is specified.

`broker-config-file-1-location`: A file location that is used to set the `DG_BROKER_CONFIG_FILE1` initialization parameter. The file location specified must be accessible by all instances of a RAC database. This is an optional command parameter.

`broker-config-file-2-location`: A file location that is used to set the `DG_BROKER_CONFIG_FILE2` initialization parameter. The file location specified must be accessible by all instances of a RAC database. This is an optional command parameter.

Initialization Parameters

The `PREPARE DATABASE FOR DATA GUARD` command sets the following initialization parameters, as per the values recommended for the Maximum Availability Architecture (MAA):

- `DB_FILES=1024`

- `LOG_BUFFER=256M`

- `DB_BLOCK_CHECKSUM=TYPICAL`

If this value is already set to FULL, the value is left unchanged.

- DB_BLOCK_CHECKSUM=TYPICAL
  If this value is already set to FULL, the value is left unchanged.

- DB_LOST_WRITE_PROTECT=TYPICAL
  If this value is already set to FULL, the value is left unchanged.

- DB_FLASHBACK_RETENTION_TARGET=120
  If this parameter is already set to a non-default value, it is left unchanged.

- PARALLEL_THREADS_PER_CPU=1

- DG_BROKER_START=TRUE

This command enables archivelog mode, enables force logging, enables Flashback Database, and sets the RMAN archive log deletion policy to SHIPPED TO ALL STANDBY. If standby redo logs do not exist in the primary database, they are added. If the logs exist and are misconfigured, they are deleted and re-created.

Command Example

The following example prepares a database with the name boston for use as a primary database. The recovery destination is $ORACLE_BASE_HOME/dbs.

```
DGMGRL> PREPARE DATABASE FOR DATA GUARD
WITH DB_UNIQUE_NAME IS boston
DB_RECOVERY_FILE_DEST IS "$ORACLE_BASE_HOME/dbs/"
DB_RECOVERY_FILE_DEST_SIZE is "400G"
DG_BROKER_CONFIG_FILE1 IS "$ORACLE_HOME/dbs/file1.dat"
DG_BROKER_CONFIG_FILE2 IS "$ORACLE_HOME/dbs/file2.dat";
Preparing database "boston" for Data Guard.
Creating server parameter file (SPFILE)from initialization parameter
memory values.
Database must be restarted after creating the server parameter
(SPFILE).
Shutting down database "boston".
Database closed.
Database dismounted.
ORACLE instance shut down. Starting database "boston" to mounted mode.
ORACLE instance started.
Database mounted. Server parameter file (SPFILE) is
"ORACLE_BASE_HOME/dbs/spboston.ora".
Initialization parameter DB_UNIQUE_NAME set to 'boston'.
Initialization parameter DB_FILES set to 1024.
Initialization parameter LOG_BUFFER set to 268435456.
Primary database must be restarted after setting static initialization
parameters. Primary database must be restarted to enable archivelog
mode.
Shutting down database "boston".
Database dismounted.
ORACLE instance shut down.
Starting database "boston" to mounted mode.
ORACLE instance started.
Database mounted.
Initialization parameter DB_FLASHBACK_RETENTION_TARGET set to 120.
Initialization parameter DB_BLOCK_CHECKSUM set to 'TYPICAL'.
Initialization parameter DB_LOST_WRITE_PROTECT set to 'TYPICAL'.
```

```
Initialization parameter PARALLEL_THREADS_PER_CPU set to 1.
Removing RMAN archivelog deletion policy 1.
Removing RMAN archivelog deletion policy 2.
RMAN configuration archivelog deletion policy set to SHIPPED TO ALL STANDBY.
Initialization parameter DB_RECOVERY_FILE_DEST_SIZE set to '400G'.
Initialization parameter DB_RECOVERY_FILE_DEST set to 'ORACLE_BASE_HOME/
dbs/'.
Initialization parameter DG_BROKER_START set to FALSE.
Initialization parameter DG_BROKER_CONFIG_FILE1 set to 'ORACLE_HOME/dbs/
file1.dat'.
Initialization parameter DG_BROKER_CONFIG_FILE2 set to 'ORACLE_HOME/dbs/
file2.dat'.
LOG_ARCHIVE_DEST_n initialization parameter already set for local archival.
Initialization parameter LOG_ARCHIVE_DEST_2 set to
'location=use_db_recovery_file_dest valid_for=(all_logfiles,all_roles)'.
Initialization parameter LOG_ARCHIVE_DEST_STATE_2 set to 'Enable'.
Initialization parameter STANDBY_FILE_MANAGEMENT set to 'MANUAL'.
Standby log group 4 will be dropped because it was not configured correctly.
Standby log group 3 will be dropped because it was not configured correctly.
Adding standby log group size 26214400 and assigning it to thread 1.
Initialization parameter STANDBY_FILE_MANAGEMENT set to 'AUTO'.
Initialization parameter DG_BROKER_START set to TRUE.
Database set to FORCE LOGGING. Database set to ARCHIVELOG.
Database set to FLASHBACK ON.
Database opened.
```

Related Topics

- [Oracle® Data Guard Broker](#)

VALIDATE FAST_START FAILOVER Command

The `VALIDATE FAST_START FAILOVER` command enables you to validate a fast-start failover configuration. It identifies misconfiguration, either while setting up or initiating fast-start failover. This command validates the fast-start failover configuration and reports the following information:

- Incorrectly set up fast-start failover parameters. For example, the fast-start failover threshold is not set appropriately.

- Issues that prevent the enabling or initiating of fast-start failover. This includes issues that prevent the usage of fast-start failover even when the conditions required for fast-start failover are met (for example, fast-start failover is enabled in Observe-Only mode).

- Issues that affect actions taken after fast-start failover is initiated

- Issues that could impact the stability of the broker configuration

- Issues with fast-start failover callout configuration scripts. Displays if the syntax of the fast-start failover configuration file `fsfocallout.ora` is correct and if the pre-callout and post-callout scripts are accessible.

Example

```
DGMGRL> VALIDATE FAST_START FAILOVER;
Fast-Start Failover: Enabled in Potential Data Loss Mode
Protection Mode: MaxPerformance
```

```
Primary: North_Sales
Active Target: South_Sales
Fast-Start Failover Not Possible:
Fast-Start Failover observer not started
Post Fast-Start Failover Issues:
Flashback database disabled for database 'dgv1'
Other issues:
FastStartFailoverThreshold may be too low for RAC databases.
Fast-start failover callout configuration file "fsfocallout.ora" has
the following issues:
Invalid lines
The specified file "./precallout" contains a path.
```

Related Topics

• Oracle® Data Guard Broker

Flashback

• Migrate Flashback Time Travel-Enabled Tables Between Different Database Releases

• Flashback Database Support for Datafile Shrink

• PDB Point-in-Time Recovery or Flashback to Any Time in the Recent Past

Migrate Flashback Time Travel-Enabled Tables Between Different Database Releases

A new PL/SQL package called `DBMS_FLASHBACK_ARCHIVE_MIGRATE` enables the migration of Flashback Time Travel-enabled tables from a database on any release (in which the package exists) to any database on any release (that supports Flashback Time Travel).

Using the `DBMS_FLASHBACK_ARCHIVE_MIGRATE` PL/SQL package, users can export and import the Flashback Archive base tables, along with their history, to another database via the Oracle Transportable Tablespaces capability. Compression is preserved when History Tables enabled with the Advanced Compression Optimization for Flashback Time Travel History Tables capability are migrated.

Related Topics

• Oracle® Database PL/SQL Packages and Types Reference

Flashback Database Support for Datafile Shrink

The existing Flashback Database capability has some limitations with respect to permanent data file resize operations. In earlier releases, the behavior of permanent datafile resizes to a smaller size, (i.e. shrink) on Oracle Databases with Flashback Database enabled, was as follows:

• When a permanent datafile shrink operation is performed on a database, which has Flashback Database enabled, the operation is allowed to succeed. However, any subsequent flashback operations, to a SCN or timestamps across any shrink

operations fails (cannot use Flashback Database to undo or rollback a datafile shrink operation).

- When performing a permanent datafile shrink operation on a database, which has Flashback Database enabled and a guaranteed restore point created, the datafile shrink operation fails with a user error.

This new capability is an enhancement to the current Flashback Database feature, allowing Flashback Database operations to succeed with permanent datafile shrinks, and shrinks to succeed even with guaranteed flashback restore points created on the database.

When objects in a tablespace are deleted, or when blocks in objects belonging to the tablespace are defragmented, the tablespace can be shrunk. Shrinking reduces the size of a datafile and returns unused space to the operating system -- including space taken up by UNDO, and defragmenting space in tables, LOBs and etc… The existing Flashback Database capability allowed users to "rewind" the database to a point in the past. However, when a permanent datafile shrink operation was performed users could not use Flashback Database to undo or rollback a datafile shrink operation. This new Flashback Database support for datafile shrink capability enables Flashback Database operations to succeed, with permanent datafile shrinks, and shrinks to succeed even with guaranteed flashback restore points created on the database.

Related Topics

- Oracle® Database Backup and Recovery User's Guide

PDB Point-in-Time Recovery or Flashback to Any Time in the Recent Past

PDBs can be recovered to an orphan PDB incarnation within the same CDB incarnation or an ancestor incarnation.

Availability of PDBs is enhanced. Both flashback and point-in-time recovery operations are supported when recovering PDBs to orphan PDB incarnations.

Related Topics

- Oracle® Database Backup and Recovery User's Guide
- **PDB Point-In-Time Recovery** lab in the Database 21c New Features workshop in LiveLabs

GoldenGate

- Automatic CDR Enhancements
- Improved Support for Table Replication for Oracle GoldenGate
- LogMiner Views Added to Assist Replication
- Oracle GoldenGate for Oracle and XStream Support for JSON Data Type

Automatic CDR Enhancements

Automatic Conflict Detection and Resolution (CDR) was introduced in Oracle Database 12c Release 2 (and Oracle GoldenGate 12.3.0.1) to automate the conflict detection and resolution configuration in active-active Oracle GoldenGate replication setups. In Oracle Database 21c, we are enhancing automatic CDR to support earliest timestamp based resolution and site priority based resolution.

Active-active Oracle GoldenGate replication customers can use the automatic CDR feature on more types of tables simplifying their active-active replication setups.

Related Topics

- [Oracle® Database PL/SQL Packages and Types Reference](#)

Improved Support for Table Replication for Oracle GoldenGate

In earlier releases, extracting tables to Oracle GoldenGate required supplemental logging data to replicate table/schema, and required you to set `TABLE/TABLEEXCLUDE` parameters to configure which table to extract.

Starting with this release, to coordinate with the Oracle GoldenGate feature OGG `EXTRACT`, the `LOGICAL_REPLICATION` clause now provides support for automatic extract of tables.

In addition, two new views, `DBA_OGG_AUTO_CAPTURED_TABLES` and `USER_OGG_AUTO_CAPTURED_TABLES`, provide you with tools to query which tables are enabled for Oracle GoldenGate automatic capture.

Related Topics

- [Oracle® Database Utilities](#)

LogMiner Views Added to Assist Replication

The `DBMS_ROLLING` package contains a new parameter that enables you to block the replication of operations unsupported by Transient Logical Standby.

Starting with this release, in the `DBMS_ROLLING.set_parameter()` procedure, there is a new parameter called `BLOCK_UNSUPPORTED`. By default, `BLOCK_UNSUPPORTED` is set to 1 [`YES`], indicating that operations performed on tables that are unsupported by Transient Logical Standby will be blocked on the primary database. If set to 0 [`OFF`], then the `DBMS_ROLLING` package does not block operations on unsupported tables. Those tables will not be maintained by Transient Logical Standby, and will diverge from the primary database.

Related Topics

- [Oracle® Database PL/SQL Packages and Types Reference](#)

Oracle GoldenGate for Oracle and XStream Support for JSON Data Type

Oracle GoldenGate for Oracle and XStream supports JavaScript Object Notation (JSON) data type.

JSON data type represents JSON in a proprietary binary format that is optimized for query and DML processing and can yield performance improvements for JSON processing in the database. It provides strong typing of JSON values so that the data type can be propagated through SQL expressions and view columns.

Related Topics

- Oracle® Database XStream Guide

Multitenant

- DRCP Enhancements for Oracle Multitenant
- Expanded Syntax for PDB Application Synchronization
- MAX_IDLE_BLOCKER_TIME Parameter
- Namespace Integration with Database
- Support Per-PDB Capture for Oracle Autonomous Database
- Time Zone support for PDBs in DBCA
- Using Non-CDBs and CDBs

DRCP Enhancements for Oracle Multitenant

Starting with Oracle Database 21c, you can configure Database Resident Connection Pooling (DRCP) from the CDB to individual PDBs for improved PDB tenancy management.

In previous releases, the DRCP pool was used by the entire container database (CDB). With per-PDB pools, you can now configure, manage, and monitor pools on individual pluggable databases (PDBs), on the basis of tenancy. This feature also provides the ability to specify Connection Class and Purity support in connect strings for DRCP. With this change, you can leverage DRCP without application code change. This feature eases the management of DRCP pool by changing the granularity of the DRCP pool from the entire CDB to a per-PDB DRCP pool. This change enables tenant administrators to configure and manage independent tenant-specific DRCP pools. Additionally, this feature enables applications to leverage some of the DRCP benefits without making any application changes.

Related Topics

- Oracle® Database Development Guide

Expanded Syntax for PDB Application Synchronization

The `ALTER PLUGGABLE DATABASE APPLICATION ... SYNC` statement now accepts multiple application names and names to be excluded. For example, a single statement issued in an application PDB can synchronize `app1` and `app2`, or synchronize all applications except `app3`.

The expanded syntax enables you to reduce the number of synchronization statements. Also, the database replays the statements in correct order. Assume that you upgrade `ussales` from v1 to v2, and then upgrade `eusales` from v1 to v2, and then upgrade `ussales` from v2 to v3. The following statement replays the statements in sequence, upgrading `ussales` to v2, then `eusales` to v2, and then `ussales` to v3.

```
ALTER PLUGGABLE DATABASE APPLICATION ussales, eusales SYNC
```

Related Topics

- Oracle® Multitenant Administrator's Guide
- **Synchronizing Apps in App PDBs** lab in the Database 21c New Features workshop in LiveLabs

MAX_IDLE_BLOCKER_TIME Parameter

`MAX_IDLE_BLOCKER_TIME` sets the number of minutes that a session holding needed resources can be idle before it is a candidate for termination.

`MAX_IDLE_TIME` sets limits for all idle sessions, whereas `MAX_IDLE_BLOCKER_TIME` sets limits only for idle sessions consuming resources. `MAX_IDLE_TIME` can be problematic for a connection pool because it may continually try to re-create the sessions terminated by this parameter.

Related Topics

- Oracle® Multitenant Administrator's Guide
- **MAX_IDLE_BLOCKER_TIME Parameter** lab in the Database 21c New Features workshop in LiveLabs

Namespace Integration with Database

Database Nest is an infrastructure that provides operating system resource isolation and management, file system isolation, and secure computing for CDBs and PDBs. This infrastructure enables a database instance to run in a protected, virtualized environment.

Sharing instance-level and operating system resources can lead to security and isolation constraints, especially in large-scale cloud deployments. Vulnerabilities can be external, such as compromised applications, unauthorized access of resources, and shared resources. An example of an internal vulnerability is a compromised Oracle process.

Database Nest isolates a database instance from other databases and applications running on the same host, and also isolates PDBs from each other and from the CDB. The feature is implemented as a Linux-specific package that provides hierarchical containers, called nests. A CDB resides within a single parent nest, while PDBs reside within the individual child nests created within the parent.

Linux processes in a PDB nest have their own process ID (PID) number spaces and cannot access PIDs in other nests. Process isolation provides a last level of defense in a security breach if a malicious user compromises a process.

Related Topics

- Oracle® Database Security Guide

Support Per-PDB Capture for Oracle Autonomous Database

To securely capture and replicate individual pluggable database (PDB) changes to Oracle Autonomous Database, you can now use Oracle GoldenGate to provide per-PDB capture.

You can now provide local user credentials to connect to an individual PDB in a multitenant architecture Oracle Database, and replicate the data from just that PDB to an Oracle Autonomous Database. You no longer need to create a common user with access to all PDBs on the multitenant container database (CDB) to replicate a PDB to an Oracle Autonomous Database. Instead, you can now provision a local user with a

predefined set of privileges to the source PDB that you want to capture. All LogMiner and Capture processing takes place only in this PDB, and only data from this specific PDB is captured and written to the Oracle GoldenGate trail. As part of this feature, the behavior for V$LOGMNR_CONTENTS changes, depending on whether you connect to a PDB, or connect to the CDB$ROOT.

Related Topics

- [Oracle® Database Utilities](#)

Time Zone support for PDBs in DBCA

In Database Configuration Assistant (DBCA) silent mode, you can optionally use the `-pdbTimezone` parameter with the `-createPluggableDatabase` and `-configurePluggableDatabase` commands to specify a time zone for a pluggable database (PDB).

Related Topics

- [Oracle® Multitenant Administrator's Guide](#)

Using Non-CDBs and CDBs

This page provides information about the availability of CDBs only in Oracle Database 21c. The non-CDB architecture was deprecated in Oracle Database 12c. It is desupported in Oracle Database 21c which means that the Oracle Universal Installer and DBCA can no longer be used to create non-CDB Oracle Database instances. A multitenant container database is the only supported architecture in Oracle Database 21c.

Oracle Real Application Clusters (RAC)

- [Cache Fusion Hardening](#)
- [Database Management Policy change for Oracle RAC in DBCA](#)
- [Integration of PDB as a Resource in Clusterware](#)
- [Pluggable Database Cluster Resources](#)

Cache Fusion Hardening

The Global Cache Service (LMS) process is vital to the operation of an Oracle Real Application Clusters (Oracle RAC) Database. Cache Fusion Hardening helps to ensure that the critical LMS process remains running despite some discrepancies between instances that would otherwise lead to a LMS and consequently database instance failures.

Cache Fusion Hardening increases availability by reducing outages, particularly in consolidated environments in which multiple Pluggable Databases (PDBs) are operated in the same Oracle RAC Container Database.

Related Topics

- [Oracle® Real Application Clusters Administration and Deployment Guide](#)

Database Management Policy change for Oracle RAC in DBCA

DBCA supports creation of database management policy for Oracle RAC.

The following variants are supported: `Automatic` and `RANK`.

The feature allows customers to modify the PDB placement in an Oracle RAC database.

Related Topics

- Oracle® Multitenant Administrator's Guide

Integration of PDB as a Resource in Clusterware

Integration of pluggable databases (PDBs) as a resource in Oracle Clusterware completes the integration of PDBs as Oracle Clusterware resources. This feature includes support using utilities and command line tools.

With resources to be defined and mapped at the PDB level, rather than the previously supported CDB level, you are better able to manage PDBs, the workload and resources against those PDBs, and monitor those PDBs.

Related Topics

- Oracle® Real Application Clusters Administration and Deployment Guide

Pluggable Database Cluster Resources

Pluggable Database (PDB) Cluster Resources enables direct mapping and control of the PDB resources. Unlike in previous versions, in which cluster resources for Multitenant databases were mapped against the Container Database (CDB) using a control of Pluggable Databases (PDBs) using services.

Pluggable Database (PDB) Cluster Resources enable a tighter and more effective control of PDBs in an Oracle RAC Database.

Related Topics

- Oracle® Real Application Clusters Administration and Deployment Guide

SecureFiles

- SecureFiles Shrink

SecureFiles Shrink

SecureFiles Shrink provides a way to free the unused space in SecureFiles segments while allowing concurrent reads and writes to SecureFiles data.

SecureFiles Shrink also helps to reduce fragmentation and improve read and write performance. The feature supports all types of SecureFiles LOBs - compressed, deduplicated, and encrypted.

Related Topics

- Oracle® Database SecureFiles and Large Objects Developer's Guide
- **SecureFile LOBs** lab in the Database 21c New Features workshop in LiveLabs

Sharding

- Centralized Backup and Restore of a Sharded Database
- Create a Sharded Database from Multiple Existing Databases (Federated Sharding)
- Multi-Shard Query, Data Loading, and DML Enhancements
- Sharding Advisor Schema Analysis Tool

Centralized Backup and Restore of a Sharded Database

Oracle Sharding backup and recovery operations are centralized using new commands in the `GDSCTL` utility. You can define a backup policy for a sharded database as a whole and restore one or more shards, or the entire sharded database, to the same point in time. Configured backups are run automatically, and you can define a schedule to run backups during off-peak hours.

This feature streamlines backup and restore configuration and simplifies the overall management of backup policies for all of the databases in a sharded database topology. In earlier releases, you had to manually configure backup policies for each shard and the shard catalog database. Some of this work could be done with Oracle Enterprise Manager, but there was no way to orchestrate a complete restore of a sharded database such that all shards and the shard catalog are restored to the same point in time.

Related Topics

- Oracle® Database Using Oracle Sharding

Create a Sharded Database from Multiple Existing Databases (Federated Sharding)

Convert a set of existing databases running the same application into a sharded database, without modifying the database schemas or the application. The databases can be geographically distributed and can have some differences in their individual schemas.

You can more easily issue queries across multiple independent databases running the same application when they are combined into a sharded database.

Related Topics

- Oracle® Database Using Oracle Sharding

Multi-Shard Query, Data Loading, and DML Enhancements

If any shards are unavailable during query execution, then the enhanced multi-shard query attempts to find alternate shards to operate on, and the query resumes without issuing a failure condition. Bulk data loading and DML can operate on multiple shards simultaneously.

Multi-shard queries are more fault-tolerant. Bulk data loading and DML operations can occur across all shards simultaneously, making these operations much faster.

Related Topics

•   Oracle® Database Using Oracle Sharding

Sharding Advisor Schema Analysis Tool

Sharding Advisor is a standalone command-line tool that helps you redesign a database schema so that you can efficiently migrate an existing, non-sharded Oracle Database to an Oracle sharding environment. Sharding Advisor analyzes your existing database schema and produces a ranked list of possible sharded database designs.

Using the Sharding Advisor recommendations, you can experience a smoother, faster migration to Oracle Sharding. Sharding Advisor analysis provides you with the information you need to:

•   Maximize availability and scalability

•   Maximize query workload performance

•   Minimize the amount of duplicated data on each shard

Related Topics

•   Oracle® Database Using Oracle Sharding

Transactional Event Queues (TEQs)

Transactional Event Queues (TEQ) is the new name for AQ Sharded Queues which were introduced in Database 12c and are further enhanced in Database 19c. All features of AQ Sharded Queues are now in TEQ, plus some new ones.

•   Advanced Queuing Support for JSON Data Type

•   Advanced Queuing: Kafka Java Client for Transactional Event Queues

•   Advanced Queuing: PL/SQL Enqueue and Dequeue Support for JMS Payload in Transactional Event Queues

•   Advanced Queuing: PL/SQL Enqueue and Dequeue Support for non-JMS Payload in Transactional Event Queues

•   Advanced Queuing: Simplified Metadata and Schema in Transactional Event Queues

•   Advanced Queuing: Transactional Event Queues for Performance and Scalability

Advanced Queuing Support for JSON Data Type

Oracle Database Advanced Queuing now supports JSON data type.

Many client applications and microservices which use Advanced Queuing for messaging have better performance if they use the JSON data type to handle JavaScript Object Notation (JSON) messages.

Related Topics

•   Oracle® Database PL/SQL Packages and Types Reference

Advanced Queuing: Kafka Java Client for Transactional Event Queues

Kafka Java Client for Transactional Event Queues (TEQ) enables Kafka application compatibility with Oracle database. This provides easy migration of Kafka applications to TEQ.

Customers don't have to manage a separate Kafka infrastructure, and this feature simplifies the event-driven application architectures with an Oracle converged database that now includes events data.

Starting from this release, Kafka Java APIs can connect to Oracle database server and use Transactional Event Queues (TEQ) as a messaging platform. Developers can migrate an existing Java application that uses Kafka to the Oracle database. A client side library allows Kafka applications to connect to Oracle database instead of Kafka cluster and use TEQ messaging platform transparently.

Related Topics

- Oracle® Database Transactional Event Queues and Advanced Queuing User's Guide

Advanced Queuing: PL/SQL Enqueue and Dequeue Support for JMS Payload in Transactional Event Queues

PL/SQL APIs perform enqueue and dequeue operations for Java Message Service (JMS) payload in Transactional Event Queues. Similarly, the PL/SQL Array APIs are exposed to Transactional Event Queues JMS users. Since JMS supports heterogeneous messages in a single JMS destination, dequeue gets one of the five JMS message types back, but cannot predict what is the type of the next message received. Therefore, it can run into application errors with PL/SQL complaining about type mismatch. Oracle suggests that the application always dequeue from Transactional Event Queues using the generic type `AQ$_JMS_MESSAGE`.

Customers can use PL/SQL APIs to enqueue and dequeue JMS payloads in Transactional Event Queues to avoid client-server round trips.

Related Topics

- Oracle® Database Transactional Event Queues and Advanced Queuing User's Guide

Advanced Queuing: PL/SQL Enqueue and Dequeue Support for non-JMS Payload in Transactional Event Queues

PL/SQL APIs can now perform enqueue and dequeue operations for ADT and RAW payloads in Transactional Event Queues. Similarly, the PL/SQL array APIs are exposed to Transactional Event Queue users.

ADT payloads are important because they allow you to have different queue payloads required by applications with all the benefits of strong type checking.

Related Topics

- Oracle® Database Transactional Event Queues and Advanced Queuing User's Guide

Advanced Queuing: Simplified Metadata and Schema in Transactional Event Queues

Transactional Event Queues have fewer tables than AQ and implement multiple memory optimizations for higher throughput. Customers will see higher message throughput just by switching from AQ to Transactional Event Queues.

This feature provides improvement in performance, scalability, and manageability.

Related Topics

- Oracle® Database Transactional Event Queues and Advanced Queuing User's Guide

Advanced Queuing: Transactional Event Queues for Performance and Scalability

Oracle Transactional Event Queues have their queue tables partitioned into multiple Event Streams, which are distributed across multiple RAC nodes for high throughput messaging and streaming of events.

Partitioned tables form part of the foundation to scale and increase performance of Transactional Event Queues, especially on Oracle RAC or Exadata.

Related Topics

- Oracle® Database Transactional Event Queues and Advanced Queuing User's Guide

7

Security Solutions

- Advanced Security
- Database Vault
- Security

Advanced Security

- Ability to Control Heartbeats in United Mode and Isolated Mode PDBs
- Ability to Set the Default Tablespace Encryption Algorithm
- Enhanced Database Availability with Zero Downtime to Switch Over to an Updated PKCS#11 Library
- Improved Performance with Large Numbers of TDE Keys in Wallets or Oracle Key Vault
- Sharing of TDE Master Encryption Key Across Oracle Processes

Ability to Control Heartbeats in United Mode and Isolated Mode PDBs

You now can control the size of the batch of heartbeats that use Oracle Key Vault or OCI KMS (OCI Vault) for centralized key management.

The `HEARTBEAT_BATCH_SIZE` initialization parameter, new with this release, enables you to set the heartbeat batch size. The duration of the heartbeat period defaults to 3 seconds.

This enhancement benefits the situation where you have a very large deployment of PDBs (for example, 1000) that use Oracle Key Vault. By setting the heartbeat batch size, you can stagger the heartbeats across batches of PDBs to ensure that for each batch a heartbeat can be completed for each PDB within the batch during the heartbeat period, and also ensure that PDB keys can be reliably fetched from an Oracle Key Vault server and cached in the persistent state.

Related Topics

- Oracle® Database Advanced Security Guide

Ability to Set the Default Tablespace Encryption Algorithm

You now can set the `TABLESPACE_ENCRYPTION_DEFAULT_ALGORITHM` dynamic parameter to define the default encryption algorithm for tablespace creation operations.

For example, if you set `TABLESPACE_ENCRYPTION_DEFAULT_ALGORITHM` to `AES256`, then future tablespace creation operations will use AES256 as the default encryption algorithm. `TABLESPACE_ENCRYPTION_DEFAULT_ALGORITHM` applies to both offline and online tablespace encryption operations. In addition, when you create a new tablespace using Database Configuration Assistant (DBCA), you can set the default tablespace encryption algorithm by using the DBCA command line for silent installations.

Supported encryption algorithms are `AES128`, `AES192`, `AES256`, `3DES168`, `ARIA128`, `ARIA192`, `ARIA256`, `SEED128`, and `GOST256`. If you do not set `TABLESPACE_ENCRYPTION_DEFAULT_ALGORITHM`, then the default encryption algorithm is the default that was used in previous releases: AES128.

Related Topics

- [Oracle® Database Advanced Security Guide](#)
- **Tablespace Encryption Algorithm** lab in the [Database 21c New Features](#) workshop in LiveLabs

Enhanced Database Availability with Zero Downtime to Switch Over to an Updated PKCS#11 Library

Starting with this release, Oracle Database can switch over to an updated PKCS#11 library without incurring any system downtime.

This release introduces a new `ADMINISTER KEY MANAGEMENT SWITCHOVER LIBRARY FOR ALL CONTAINERS` statement, which will enable an Oracle database to switch over from the PKCS#11 library that it is currently using to the updated PKCS#11 library.

In previous releases, it was necessary to completely shut down any TDE-enabled database that used an online TDE master encryption key in Oracle Key Vault before an update to the Oracle Key Vault endpoint software could be installed. After the updated PKCS#11 library was installed, the TDE-enabled database would need to be started up again. This complete shut down followed by a start up of the database instance was necessary because long-running background processes of the database instance could not be told to unload the earlier PKCS#11 library and load the updated one.

Starting with this release, to switch over the database server to use an updated endpoint shared PKCS#11 library, you execute the following statement to initiate the switch over operation:

```
ADMINISTER KEY MANAGEMENT SWITCHOVER TO LIBRARY
'fully_qualified_file_name_of_library' FOR ALL CONTAINERS;
```

Related Topics

- [Oracle® Database Advanced Security Guide](#)

Improved Performance with Large Numbers of TDE Keys in Wallets or Oracle Key Vault

Oracle Database 21c introduces improved performance for Transparent Data Encryption (TDE).

This enhancement enables faster wallet loading and key rotations in multitenant databases. It allows for faster execution of TDE administration tasks and PDB cloning operations.

Sharing of TDE Master Encryption Key Across Oracle Processes

Starting with this release, you can enable sharing of Transparent Data Encryption (TDE) master encryption keys across Oracle processes.

This enhancement allows TDE-enabled Oracle databases to have their TDE master encryption keys managed by the Oracle Cloud Infrastructure (OCI) key management service (KMS). To control this functionality, you set the TDE_KEY_CACHE initialization parameter.

Related Topics

• Oracle® Database Advanced Security Guide

Database Vault

• Ability to Prevent Local Oracle Database Vault Policies from Blocking Common Operations

• ADMINISTER KEY MANAGEMENT Statement Now Protected by Oracle Database Vault Command Rules

• DBA_DV_SIMULATION_LOG View Columns REALM_NAME and RULE_SET_NAME Now VARCHAR2 Data Type

• No Need to Disable Oracle Database Vault Before Upgrades

• Uninstalling and Installing Oracle Label Security and Oracle Database Vault Now Supported

Ability to Prevent Local Oracle Database Vault Policies from Blocking Common Operations

Starting with this release, a `DV_OWNER` common user in the CDB root can prevent local users from creating Oracle Database Vault controls on common objects in a pluggable database (PDB).

Blocking common users from common operations can prevent the execution of SQL commands that are necessary for managing the application or CDB database. To prevent this situation, a user who has the `DV_OWNER` role in the root can execute the `DBMS_MACADM.ALLOW_COMMON_OPERATION` procedure to control whether local PDB users can create Database Vault controls on common users' objects (database or application).

In previous releases, in a multitenant environment, a local Oracle Database Vault user could create Database Vault policies that could potentially block application or common operations. Blocking common users from common operations can prevent the execution of SQL commands that are necessary for managing the application or CDB database. To prevent this situation, a user who has the `DV_OWNER` role in the root can execute the `DBMS_MACADM.ALLOW_COMMON_OPERATION` procedure to control whether local PDB users can create Database Vault controls on common users' objects (database or application).

Related Topics

• Oracle® Database Vault Administrator's Guide

• **Realms** lab in the Database 21c New Features workshop in LiveLabs

ADMINISTER KEY MANAGEMENT Statement Now Protected by Oracle Database Vault Command Rules

You now can protect the `ADMINISTER KEY MANAGEMENT` statement with Oracle Database Vault command rules.

The `ADMINISTER KEY MANAGEMENT` statement manages Transparent Data Encryption (TDE) features.

Related Topics

- Oracle® Database Vault Administrator's Guide

DBA_DV_SIMULATION_LOG View Columns REALM_NAME and RULE_SET_NAME Now VARCHAR2 Data Type

Starting with this release, the `REALM_NAME` and `RULE_SET_NAME` columns will use the `VARCHAR2` data type instead of being in nested tables.

This enhancement enables multiple realm names and rule set names to be separated by a comma in a `VARCHAR2` data type instead of using a nested table in the columns. In the unlikely situation where you may have so many realms or rule set names protecting a single object in which the `VARCHAR2` data exceeds 4000 characters, Oracle Database Vault will truncate the list of realms or rule sets at 4000 characters in the column and if the full set is needed, it can be retrieved from the nested table in the `DVSYS.SIMULATION_LOG$` base table.

Storing realm names and rule set names as a `VARCHAR2` data type makes it easier for you to read the realm name or rule set name in the simulation log. Most users only use a single realm or rule set to protect their sensitive data objects and even if they do use multiple realms or rule sets, it is easier to read data in a `VARCHAR2` data type rather than a nested table.

Related Topics

- Oracle® Database Vault Administrator's Guide

No Need to Disable Oracle Database Vault Before Upgrades

Starting with this release, you do not need to disable Oracle Database Vault in every container before upgrading from an earlier release to the current release.

You only need to grant the `DV_PATCH_ADMIN` role to `SYS` commonly before you perform the upgrade. After the upgrade is complete the Database Vault controls work as before. Then revoke the `DV_PATCH_ADMIN` role from `SYS` commonly.

Alternatively, you can explicitly disable Oracle Database Vault in all containers before the upgrade, and then after the upgrade, explicitly enable Oracle Database Vault in all the containers.

Related Topics

- Oracle® Database Vault Administrator's Guide

Uninstalling and Installing Oracle Label Security and Oracle Database Vault Now Supported

You now can install and uninstall Oracle Database Vault and Oracle Label Security in PDBs.

To install a feature into a PDB requires that the feature already be installed in the CDB root.

This enhancement enables you to configure your own databases with Oracle Label Security and Oracle Database Vault to meet your site's requirements.

Related Topics

- Oracle® Database Vault Administrator's Guide
- **Uninstalling Oracle Database Vault** and **Installing Oracle Database Vault** labs in the Database 21c New Features workshop in LiveLabs

Security

- Oracle Blockchain Table
- Immutable Tables
- Authentication and Authorization
- Encryption
- Audit

Oracle Blockchain Table

Blockchain tables are append-only tables in which only insert operations are allowed. Deleting rows is either prohibited or restricted based on time. Rows in a blockchain table are made tamper-resistant by special sequencing and chaining algorithms. Users can verify that rows have not been tampered. A hash value that is part of the row metadata is used to chain and validate rows.

Blockchain tables can be used to implement blockchain applications where the participants trust the Oracle Database provider, but want means to verify that their data hasn't been tampered with. The participants are different database users who trust the Oracle Database provider to maintain a verifiable, tamper-resistant blockchain of transactions. All participants must have privileges to insert data into the blockchain table. The contents of the blockchain table are defined and managed by the application, with a few added metadata fields maintained by Oracle Database. By leveraging a trusted provider with verifiable crypto-secure data management practices, such applications can avoid the distributed consensus requirements. This provides most of the protection of the distributed peer-to-peer blockchains, but with much higher throughput and lower transaction latency compared to peer-to-peer blockchains using distributed consensus.

Related Topics

- Oracle® Database Administrator's Guide
- **Blockchain Tables and Rows** lab in the Database 21c New Features workshop in LiveLabs

Immutable Tables

Immutable tables are insert-only tables in which existing data cannot be modified. Deleting rows is either prohibited or restricted based on the insertion time of the rows.

Immutable tables protect data against unauthorized modification by insiders. This includes database administrators or compromised users who have access to insider credentials. Immutable tables also prevent accidental data modification that may be caused by human error.

Related Topics

• Oracle® Database Administrator's Guide

Authentication and Authorization

• Ability to Specify the Location of the CMU Wallet and dsi.ora File with a Database Property

• Ability to Use Multiple Kerberos Principals with a Single Database Client

• Addition of USER_APPLICATION_ROLES Data Dictionary View

• Connect to Multiple Databases with Different Certificates from a Single Client

• Enterprise User Manager Support for Per-PDB Directory Service Connections

• Force Upgraded Password File to be Case Sensitive

• Gradual Database Password Rollover for Applications

• Minimum Password Length Enforcement for All PDBs

• New and Updated Password User Profiles for STIG and CIS

• Oracle Database Connections to Kerberos Servers Now Default to TCP

• Windows Authentication No Longer Uses NTLM by Default

Ability to Specify the Location of the CMU Wallet and dsi.ora File with a Database Property

You now can specify the location of the centrally managed users (CMU) wallet and dsi.ora files for an individual PDB by using a database property on the PDB.

This enhancement enables a PDB administrator to specify a location to store these files rather than being limited by the sqlnet.ora WALLET_LOCATION parameter or having to use the default wallet locations. This feature works almost exactly as using WALLET_LOCATION or the default wallet location except that users with administrative privileges will not be able to start the database because directory objects are part of the database. To store the CMU wallet and dsi.ora files in the location path specified by a database directory object, you must set the CMU_WALLET database property to this directory object.

After you have set the CMU_WALLET database property to a directory object for an individual PDB, you should store the CMU wallet and dsi.ora files for this PDB in the location specified by the directory object, so that they can be accessed through the corresponding file systems.

Related Topics

• Oracle® Database Advanced Security Guide

Ability to Use Multiple Kerberos Principals with a Single Database Client

Starting with this release, when you configure Kerberos authentication for an Oracle Database client, you can specify multiple Kerberos principals with a single Oracle Database client.

To enable this functionality, you will need to create a separate credential cache for each user in the client and then use the connect string to specify the user.

In previous releases, you were restricted to one Kerberos principal for each Oracle Database client.

Related Topics

• Oracle® Database Security Guide

Addition of USER_APPLICATION_ROLES Data Dictionary View

Starting with this release, the `USER_APPLICATION_ROLES` data dictionary view provides a subset of roles that are available in `DBA_APPLICATION_ROLES` that only apply to the current user.

`USER_APPLICATION_ROLES` enables the current user to see all the application roles that have been granted to the user instead of the list of all possible application roles available in `DBA_APPLICATION_ROLES`.

Related Topics

• Oracle® Database Security Guide

Connect to Multiple Databases with Different Certificates from a Single Client

Starting with this release, you can configure database clients to maintain multiple Secure Sockets Layer (SSL) sessions using different SSL certificates.

This feature enables multi-threaded clients to use multiple wallets with different certificates for simultaneous SSL sessions.

This enhancement is especially useful for database clients simultaneously connecting to multiple cloud databases, each with a different certificate.

Related Topics

• Oracle® Database Security Guide

Enterprise User Manager Support for Per-PDB Directory Service Connections

Enterprise User Security (EUS) is now able to connect a different directory service for each pluggable database (PDB). Previously in a multitenant database, all the containers connected to a single directory service.

This feature enables a large Oracle Database deployment to have different directory services. It also enables independent software vendors to allocate an individual PDB to connect to the appropriate customer's directory service.

Related Topics

• Oracle® Database Enterprise User Security Administrator's Guide

Force Upgraded Password File to be Case Sensitive

The parameter to enable or disable password file case sensitivity is removed. All passwords in new password files are case-sensitive.

Case-sensitive password files provide more security than older password files that are case insensitive. Oracle recommends that you use case-sensitive password files. However, upgraded password files from earlier Oracle Database releases can retain their original case-insensitivity. You can force your password files to be case-sensitive by migrating password files from one format to another.

Related Topics

• Oracle® Database Security Guide

• **Upgraded Password File Case Sensitivity** lab in the Database 21c New Features workshop in LiveLabs

Gradual Database Password Rollover for Applications

Starting with this release, an application can change its database passwords without an administrator having to schedule downtime.

To accomplish this, a database administrator can associate a profile having a non-zero limit for the `PASSWORD_ROLLOVER_TIME` password profile parameter, new with this release, with an application schema. This allows the database password of the application user to be altered while allowing the older password to remain valid for the time specified by the `PASSWORD_ROLLOVER_TIME` limit. During the rollover period of time, the application instance can use either the old password or the new password to connect to the database server. When the rollover time expires, only the new password is allowed.

Before this enhancement, an administrator normally took the application down when the application database password was being rotated. This is because the password update requires changes on both the database and the application side. With the gradual database password rollover enhancement, the application can continue to use the older password until the new password is configured in the application.

In addition to the new clause `PASSWORD_ROLLOVER_TIME` in the `CREATE PROFILE` and `ALTER PROFILE` statements, the `ALTER USER` statement has a new clause, `EXPIRE PASSWORD ROLLOVER PERIOD`. The `ACCOUNT_STATUS` column of the `DBA_USERS` and `USER_USERS` data dictionary views have several new statuses indicating values to indicate rollover status.

Related Topics

• Oracle® Database Security Guide

Minimum Password Length Enforcement for All PDBs

Starting with this release, you can enforce a minimum password length on all PDBs by setting a mandatory profile in the CDB root.

The mandatory profile is a generic profile and can only have the `PASSWORD_VERIFY_FUNCTION` parameter in the `CREATE MANDATORY PROFILE` statement to define the password length.

This profile adds a minimum password length to the local profiles with which the PDB user is associated. Because a common user sets the mandatory profile in the CDB, a PDB administrator cannot remove the password length requirement from local profiles and allow users to set insecure short passwords.

To use this profile, you create a password verification function to define the password length, execute the `CREATE MANDATORY PROFILE PL/SQL` procedure to use the `PASSWORD_VERIFY_FUNCTION` function in a profile, and then set the `MANDATORY_USER_PROFILE` initialization parameter in the CDB root to apply it to all containers.

Related Topics

- Oracle® Database Security Guide
- **Password Length Enforcement** lab in the Database 21c New Features workshop in LiveLabs

New and Updated Password User Profiles for STIG and CIS

This release introduces a new user profile and an updated user profile to comply with STIG and CIS standards for password management.

The `ora_stig_profile` user profile has been updated with the latest Security Technical Implementation Guide's (STIG) guidelines, and the `ora_cis_profile`, new for this release, has the latest Center for Internet Security (CIS) guidelines for passwords. You can use these user profiles directly with their database users or use them as part of your own user profiles. Oracle keeps these profiles up to date to make it easier for you to implement password policies that meet STIG and CIS guidelines.

Related Topics

- Oracle® Database Security Guide

Oracle Database Connections to Kerberos Servers Now Default to TCP

In previous releases, Oracle Database connections to Kerberos servers defaulted to start with User Datagram Protocol (UDP).

If you prefer the Oracle Database connections to start with UDP connections, then you can modify the Kerberos krb5.conf configuration file to start the connections using UDP.

This enhancement speeds the authentication process for environments that have many Kerberos Key Distribution Center (KDC) servers.

Related Topics

- Oracle® Database Security Guide

Windows Authentication No Longer Uses NTLM by Default

For Microsoft Windows installations with `AUTHENTICATION_SERVICES=NTS`, starting with this release, the `SQLNET.NO_NTLM` parameter setting in the `sqlnet.ora` file will default to `TRUE`.

In previous releases, the default for this parameter was `FALSE`. `SQLNET.NO_NTLM` controls whether NTLM can be used with NTS authentication. A `TRUE` setting means that NTLM cannot be used in NTS authentication. Because NTLM does not normally provide mutual authentication and is hence less secure, a `TRUE` setting for `SQLNET.NO_NTLM` makes the database and client more secure.

The `SQLNET.NO_NTLM` parameter is used on both the server and the client. If you have upgraded a Microsoft Windows installation of an Oracle database or a client in which `SQLNET.NO_NTLM` had not been set, then its default will be `TRUE`.

Related Topics

• Oracle® Database Security Guide

Encryption

• Updated Support for Micro Edition Suite (MES) for FIPS 140.2

• Support for DBMS_CRYPTO Asymmetric Key Operations

Updated Support for Micro Edition Suite (MES) for FIPS 140.2

Starting with this release, Oracle Database supports Micro Edition Suite (MES) version 4.5 for FIPS 140.2.

The Micro Edition Suite (MES) version 4.5 updates include four new CVEs in the RSA BSAFE MES library, support for the rules that FIPS 140.2 requires, and access to the updated NZ/ZT library from the Crypto Foundation.

This enhancement enables the Oracle Database FIPS 140.2 configuration to benefit from new features and security improvements available from the latest RSA BSAFE MES library.

Related Topics

• Oracle® Database Security Guide

Support for DBMS_CRYPTO Asymmetric Key Operations

Starting with this release, the `DBMS_CRYPTO` PL/SQL package supports asymmetric key operations, in addition to the existing support for symmetric key operations.

To implement the support for asymmetric key operations, the following procedures have been added to the `DBMS_CRYPTO` package:

• `PKENCRYPT`

• `PKDECRYPT`

• `SIGN`

- `VERIFY`

Related Topics

- [Oracle® Database PL/SQL Packages and Types Reference](#)

Audit

- [Auditing for Oracle XML DB HTTP and FTP Services](#)
- [Predefined Unified Audit Policies for Security Technical Implementation Guide (STIG) Compliance](#)
- [SYSLOG Destination for Common Unified Audit Policies](#)
- [Unified Audit Policies Enforced on the Current User](#)
- [Unified Audit Policy Configuration Changes Effective Immediately](#)
- [Unified Auditing on an Editioned Object Now Applies to All Its Editions](#)

Auditing for Oracle XML DB HTTP and FTP Services

Starting with this release, you can create unified audit policies for database connections made using the database protocol servers for HTTP, HTTPS, and FTP.

A unified audit policy can track requests that have been made to servlets, such as those used by Oracle Enterprise Manager Express, Oracle Database Native Web Services, and HTTP and WebDAV operations that use Oracle XML DB Repository.

This enhancement enables you to track and monitor access to Oracle Database provided by the HTTP, HTTPS, and FTP protocols, including access by WebDAV clients.

Related Topics

- [Oracle® Database Security Guide](#)

Predefined Unified Audit Policies for Security Technical Implementation Guide (STIG) Compliance

Starting with this release, you can audit for Security Technical Implementation Guide (STIG) compliance by using new predefined unified audit policies.

These policies are as follows:

- `ORA_STIG_RECOMMENDATIONS`
- `ORA_ALL_TOPLEVEL_ACTIONS`
- `ORA_LOGON_LOGOFF`

Related Topics

- [Oracle® Database Security Guide](#)

SYSLOG Destination for Common Unified Audit Policies

Certain predefined columns of unified audit records from common unified audit policies can be written to the UNIX SYSLOG destination.

To enable this feature, you set `UNIFIED_AUDIT_COMMON_SYSTEMLOG`, a new CDB level init.ora parameter. This enhancement enables all audit records from common unified audit policies to be consolidated into a single destination.

This feature is available only on UNIX platforms, not Windows.

Related Topics

- Oracle® Database Security Guide

Unified Audit Policies Enforced on the Current User

Starting with this release, unified audit policies are enforced on the current user who executes the SQL statement.

In previous releases, unified audit policies were enforced on the user who owned the top-level user session (that is, the login user session) in which the SQL statement is executed.

Scenarios in which the current user is different from the login user include but are not limited to the following:

- Trigger execution
- Definer rights procedure execution
- Functions and procedures that are executed during the evaluation of views

Related Topics

- Oracle® Database Security Guide

Unified Audit Policy Configuration Changes Effective Immediately

Starting with this release, changes made to a unified audit policy become effective immediately in the current session and in all other on-going active sessions.

In previous releases, users who were affected by a changed unified audit policy had to log out of and then back into the session in order for the unified audit policy to take effect.

Related Topics

- Oracle® Database Security Guide

Unified Auditing on an Editioned Object Now Applies to All Its Editions

Unified audit policies, if configured for auditing actions on an editioned object, apply to all editions of the object.

This support enables consistent audit policy definition and enforcement in all object editions, whether they are existing or new. Applications which rely on edition-based redefinition can take advantage of the availability of the audit policies to audit all editioned objects in a consistent manner.

Related Topics

- Oracle® Database Security Guide

8

New Features in 21c Release Updates

Release Update 21.4 Features

There are no new features for the 21.4 release update.

Release Update 21.5 Features

There are no new features for the 21.5 release update.

Release Update 21.6 Features

There are no new features for the 21.6 release update.

Release Update 21.7 Features

- Data Guard per Pluggable Database

Data Guard per Pluggable Database

Data Guard per Pluggable Database enables Data Guard asynchronous physical standby protection at the pluggable database level as an alternative to Data Guard at the container database level. This flexibility allows switchover and failover operations to be handled at a finer granularity, if desired, with the option to fail over a single PDB instead of all PDBs running within a container database.

While the ability to configure Oracle Data Guard at the CDB level allows a single configuration to be applied to all of the PDBs running within it, Data Guard per Pluggable Database provides a more granular level of control, allowing for better consolidation, workload distribution, and isolation.

Related Topics

- Oracle® Data Guard

Release Update 21.8 Features

There are no new features for the 21.8 release update.

Release Update 21.9 Features

There are no new features for the 21.9 release update.

Release Update 21.10 Features

There are no new features for the 21.10 release update.

Release Update 21.11 Features

There are no new features for the 21.11 release update.

Release Update 21.12 Features

- AutoUpgrade Release Update (RU) Upgrades
- AutoUpgrade Sets Parallelism Based on System Resources
- AutoUpgrade Supports Upgrades with Keystore Access to Databases Using TDE
- AutoUpgrade Unplug-Plugin Upgrades to Different Systems
- REST APIs for AutoUpgrade

AutoUpgrade Release Update (RU) Upgrades

AutoUpgrade supports the option of using AutoUpgrade to perform out-of-place Oracle home Release Update patching.

For an out-of-place patch of Oracle Database using AutoUpgrade, AutoUpgrade moves the source database that you want to patch to a new Oracle Database Oracle home, and then patches the database binaries in that target Oracle home with the Release Update that you select. With this option, you can use AutoUpgrade at any time that you want to move the database to a new Oracle home, either as part of a planned upgrade or as part of a patch plan. In a patch operation, AutoUpgrade performs the patch using the following workflow:

1. AutoUpgrade recognizes that the source database and the target Oracle Database are the same base release.

2. AutoUpgrade skips the upgrade steps.

3. AutoUpgrade patches the target database using the Release Update.

Related Topics

- AutoUpgrade Patching

AutoUpgrade Sets Parallelism Based on System Resources

AutoUpgrade automatically evaluates system resources and makes an intelligent decision as to how many upgrade jobs can run simultaneously.

AutoUpgrade uses the `CPU_COUNT` value and system process parameters to determine available system resources, and calibrates both the number of upgrades that can run at a time and the number of parallel threads for each upgrade. Upgrades that exceed a safe threshold are put in a queue so that they can be run as system resources become available.

Related Topics

• tune_setting

AutoUpgrade Supports Upgrades with Keystore Access to Databases Using TDE

AutoUpgrade enhances support for databases that use transparent data encryption (TDE) by enabling keystore generation.

AutoUpgrade now enables you to provide passwords to an external key manager generated and maintained by AutoUpgrade. With this configuration, AutoUpgrade supports unmanned or automated operations of TDE-enabled databases. AutoUpgrade can open the source database keystore without prompting for the keystore password, and enroll the target database into the TDE external keystore for key management, so that the target database can start automatically.

Related Topics

• *Upgrades and Transparent Data Encryption*

AutoUpgrade Unplug-Plugin Upgrades to Different Systems

You can now use the Oracle Database AutoUpgrade Unplug/Plug method to unplug a PDB from one system and plug into a different system and upgrade.

In earlier releases, AutoUpgrade supported unplug/plug/upgrades on the same server, but it was not possible to unplug a PDB from one server, plug it into a different system, and then upgrade the PDB. With this feature, you can now migrate and upgrade the PDB in a single operation, including migrations to the cloud.

Related Topics

• *Unplug-Plug Relocate Upgrades With AutoUpgrade*

REST APIs for AutoUpgrade

To facilitate safe and secure remote use of the AutoUpgrade for Oracle Database upgrades, AutoUpgrade now provides REST APIs (ORDS and OCI).

The Oracle REST Data Services (ORDS) database API is a database management and monitoring REST API embedded into Oracle REST Data Services. The Oracle Cloud Infrastructure (OCI) REST API is enabled by configuring the REST Adapter connection to use the OCI Signature Version 1 security policy. You can now use these features to run AutoUpgrade upgrades remotely over SSH.

Related Topics

8-4

- *C AutoUpgrade REST APIs*

Release Update 21.13 Features

There are no new features for the 21.13 release update.