Oracle® Database

Database New Features Guide

12*c* Release 2 (12.2)
E85871-02
December 2017

Oracle Database Database New Features Guide, 12*c* Release 2 (12.2)

E85871-02

Contents

Preface

1 Oracle Database 12c Release 2 (12.2) New Features

Preface

This document describes new features implemented in Oracle Database 12*c* Release 2 (12.2).

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

Oracle Database New Features Guide is addressed to people familiar with previous releases of Oracle Database who would like to become familiar with features, options, and enhancements that are new in this release of the database.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Database 12*c* Release 2 (12.2) documentation set:

- *Oracle Database Error Messages*
- *Oracle Database Administrator’s Guide*
- *Oracle Database Concepts*
- *Oracle Database Reference*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Oracle Database 12c Release 2 (12.2) New Features

This chapter contains descriptions of all of the features that are new to Oracle Database 12c Release 2 (12.2).

- Application Development
- Availability
- Big Data and Data Warehousing
- Compression and Archiving
- Database Lifecycle Management
- Database Overall
- Diagnosability
- Exadata
- Manageability
- Miscellaneous
- Performance
- Public Cloud
- RAC and Grid
- Security
- Spatial and Graph
- Windows

Application Development

- Database Development Productivity Tools Enhancements
- General
- Globalization Support Enhancements
- JSON Support
- Oracle SQL and PL/SQL Improvements
- Reducing Costs and Complexities of Migration to Oracle Database
- Support for the .NET and MS Development Community
- Support for the Java Development Community
- XML

Database Development Productivity Tools Enhancements

- Application Express 5.0: Packaged Applications
- Application Express 5.0: Page Designer
- Application Express 5.0: Universal Theme
- Application Express 5.0: User Interface Enhancements
- Application Express 5.0: Improved Application Builder Design
- Application Express 5.0: Interactive Reporting
- Application Express 5.0: Mobile Enhancements
- Application Express 5.0: Modal Dialog Pages
- Application Express 5.0: All New Calendar

Application Express 5.0: Packaged Applications

Oracle Application Express release 5.0 includes a broad collection of point solutions called packaged applications. These are Application Express applications that you can use out-of-the-box and that are supported by Oracle Database. Examples include Project Tracking, Survey Builder, Meeting Minutes, and Group Calendar. There are 19 productivity applications in all. Additionally, there are 16 sample applications that are used to showcase the features of Oracle Application Express, from Sample Charts to Sample Data Loading. Release 5.0 even includes a sample application that demonstrates the powerful spatial capabilities that are present in every Oracle Database.

Oracle Application Express release 5.0 packaged applications are very popular as point solutions and learning examples.

Related Topics

- *Oracle® Application Express Application Builder User's Guide*

Application Express 5.0: Page Designer

Page Designer is a modern, intuitive, and exceedingly powerful browser-based Integrated Development Environment (IDE). As a completely new IDE that is designed to greatly improve developer productivity, Page Designer enables you to very quickly develop and maintain your Application Express applications. Page Designer features a better visual representation of your application pages and provides an entirely new way of quickly developing pages by using intuitive drag and drop. The enhanced code editor provides SQL and PL/SQL validation with inline errors, auto completion, syntax highlighting, search and replace with regex support, complete with undo and redo support.

Oracle Application Express Page Designer greatly improves developer productivity, provides a cohesive user experience, better visual representation, intuitive drag and drop, and an enhanced code editor.

Related Topics

- *Oracle® Application Express Application Builder User's Guide*

Application Express 5.0: Universal Theme

Universal Theme is a new user interface for your applications for Oracle Application Express release 5.0. It is a simpler, yet more capable theme that eliminates excessive templates and enables customization using the built-in Theme Roller and Template Options. The Universal Theme enables developers to build modern, responsive, and sophisticated applications without requiring expert knowledge of HTML, Cascading Style Sheet (CSS), or JavaScript.

Oracle Application Express Universal Theme provides a number of new capabilities including Theme Roller, Template Options, responsive design, and accessibility.

Related Topics

- *Oracle® Application Express Application Builder User's Guide*

Application Express 5.0: User Interface Enhancements

With the development of Universal Theme, there are also several enhancements to the handling of themes and templates. Oracle Application Express release 5.0 includes features such as Theme Subscriptions, Template Options, and Theme Styles. These features give you more granular control over your templates and the HTML that the Application Express engine produces. It is easier to fully control your application user interface.

Oracle Application Express release 5.0 enables developers to easily build modern, responsive, and beautiful applications out-of-the-box.

Related Topics

- *Oracle® Application Express Application Builder User's Guide*

Application Express 5.0: Improved Application Builder Design

Oracle Application Express release 5.0 introduces a new interface that focuses on improving the user experience through simplicity and the removal of clutter. The new design uses a new color palette, carefully crafted icons, and improved menus and navigation. The new design also provides improved accessibility and keyboard support, more intuitive page layouts, and many other enhancements.

Oracle Application Express release 5.0 Application Builder is more intuitive and productive for developers.

Related Topics

- *Oracle® Application Express Application Builder User's Guide*

Application Express 5.0: Interactive Reporting

Interactive reports are completely rebuilt in Oracle Application Express release 5.0 to enhance both developer and end-user capabilities. New capabilities include the ability to define multiple reports on a single page, column pivot, fixed headers, and modernized actions. You can also restyle interactive report regions using Cascading Style Sheets (CSS) in a similar manner to other regions within Oracle Application Express.

Oracle Application Express release 5.0 interactive reports enhances both developer and end-user capabilities.

Related Topics

- *Oracle® Application Express Application Builder User's Guide*

Application Express 5.0: Mobile Enhancements

You can now build reports that display all of your data on any mobile device by using reflow table or column toggle. Reflow table wraps each column or changes the display to allow multiple lines on very small screens. Column toggle enables you to specify the most important columns to view and those columns that should be hidden, as necessary, on smaller screens. Panels are now incorporated into mobile applications and are used to display navigation menus.

Oracle Application Express release 5.0 mobile improvements enable the development of improved mobile-first applications.

Related Topics

- *Oracle® Application Express Application Builder User's Guide*

Application Express 5.0: Modal Dialog Pages

Now you can easily define modal and non-modal pages, complete with the ability to use standard page processes. You no longer need to manually edit a page using JavaScript. Instead, set the display type and the appropriate template and let Oracle Application Express take care of the rest.

Oracle Application Express release 5.0 modal dialog pages make it easy for developers to display pages modally, as opposed to writing significant amounts of custom code.

Related Topics

- *Oracle® Application Express Application Builder User's Guide*

Application Express 5.0: All New Calendar

The new calendar component includes built-in support for Month, Week, Day, and Agenda views, and is much easier to customize. The calendar is based on the popular FullCalendar library and supports drag and drop, time-based events, and is even responsive. The ability to easily define duration-based events and restyle the calendars, makes the new calendar very popular with both developers and end-users.

Oracle Application Express release 5.0 calendars now support duration-based events and are significantly easier to style.

Related Topics

- *Oracle® Application Express Application Builder User's Guide*

General

- SQL*Plus Command History
- SQL*Plus Tuning Options

- Extensible Indexing Enhancements

SQL*Plus Command History

SQL*Plus now provides the ability to reissue the previously executed commands.

This functionality is similar to the shell `history` command available on the UNIX platform command line shells.

Related Topics

- *SQL*Plus® User's Guide and Reference*

SQL*Plus Tuning Options

This feature introduces new SQL*Plus commands `SET PREFETCH`, `SET LOBPREFETCH`, and `SET STATEMENTCACHE`. SQL*Plus can now use Oracle Database techniques such as pre-fetching, array fetching, and statement caching.

These new `SET` commands can be used to tune query performance.

Related Topics

- *SQL*Plus® User's Guide and Reference*

Extensible Indexing Enhancements

This feature enhances the extensible indexing framework in Oracle Database to support system-managed composite (multi-column) domain indexes. The object identifier is returned to extensible indexing call out functions to support parallelism.

DBAs no longer have to manually manage composite domain indexes which is especially beneficial for Oracle Text users. Parallel load and parallel insert is now supported for domain indexes which is especially beneficial for Oracle XML DB users.

Related Topics

- *Oracle® Database Data Cartridge Developer's Guide*

Globalization Support Enhancements

- Column-Level Collation
- Case-Insensitive Database
- AL32UTF8 As the Default Database Character Set
- Unicode 7.0 Support

Column-Level Collation

Collation (also called sort ordering) determines if a character string equals, precedes or follows another string when the two strings are compared and sorted. Numerous collations supported by Oracle Database can order strings according to conventions used in dictionaries, names lists and other sorted text written in many languages of the world. The column-level collation allows you to declare the sort ordering on a column level, using syntax and semantics compatible with the ISO, IEC and ANSI SQL standard. The declared collation is automatically applied by all collation-sensitive SQL operations referencing the column.

The column-level collation capability simplifies application development and provides greater flexibility for creating applications. It enables developers to optimally address applications where only some of the data needs language or region-specific sorting rules. Support for ISO, IEC and ANSI SQL standard syntax simplifies application migration to Oracle Database from non-Oracle Database systems that support column-level collation declarations.

Related Topics

- *Oracle® Database Globalization Support Guide*

Case-Insensitive Database

Oracle Database supports case-insensitive collations, such as `BINARY_CI` or `GENERIC_M_CI`. By applying such collations to SQL operations, an application can perform string comparisons and matching in a case-insensitive way, independent of the language of the data. With the new Oracle Database 12*c* Release 2 (12.2) ability to declare collations for columns, you can declare a column to always be compared in a case-insensitive way. The column collation, if not specified explicitly, is inherited from a table default collation, which in turn is inherited from a schema default collation. This way, you can easily declare all character columns of an application in a database as case-insensitive.

With this feature, developers can declare data as case-insensitive and do not have to add explicit uppercasing operations to SQL statements. This feature simplifies application migration to Oracle Database from non-Oracle Database systems that allow such declarations.

Related Topics

- *Oracle® Database Globalization Support Guide*

AL32UTF8 As the Default Database Character Set

The AL32UTF8 character set is now the default for the database character set in Oracle Database installs and in the Database Configuration Assistant (DBCA). Previously, the default database character set was based on the operating system locale setting. That is now offered as a secondary choice.

The AL32UTF8 character set is Oracle Corporation's implementation of the industry standard UTF-8 encoding which supports all of the written languages of the world. Making AL32UTF8 the default for new database deployments enables the database to support multilingual globalized applications and avoids the potential cost of later migrating to Unicode.

Related Topics

- *Oracle® Database Globalization Support Guide*

Unicode 7.0 Support

The National Language Support (NLS) data files for AL32UTF8 and AL16UTF16 character sets are updated to match version 7.0 of the Unicode Standard character database. The support for Unicode Collation Algorithm (UCA) is also updated to conform with UCA 7.0.

With this enhancement, Oracle Database conforms to the latest version of the Unicode Standard.

Related Topics

- *Oracle® Database Globalization Support Guide*

JSON Support

- JSON Improvements

JSON Improvements

This release incorporates significant support enhancements for storing and querying JavaScript Object Notation (JSON) documents for Oracle Database. These improvements include:

- Improvements to JSON searching:
  - The JSON path expressions used with simplified syntax for querying JSON now support navigating to specific members of an array.
  - JSON path expressions used with `JSON_EXISTS` condition now support predicates.
- Improvements to JSON search index:
  - A new, simplified syntax makes it easier to create a JSON search index.
  - The JSON search index supports `RANGE` and `LIST` partitioned tables.
  - The JSON search index support range-based searching on numeric values.
  - The JSON search index can now deal with large keys.
- New capabilities for generating JSON documents directly from SQL queries and PL/SQL data.
- JSON operators are supported in PL/SQL.
- Support for manipulating JSON documents using PL/SQL. This includes the ability to make incremental modifications to JSON documents.
- Support for optimizing the performance for JSON query operations using Oracle Database In-Memory.
- Support for performing spatial-based queries on JSON documents containing GeoJSON.
- A new data guide feature that facilitates understanding of the structure and content of your JSON documents.
- Support for using JSON in a sharded database configuration.

This feature makes it easier to work with JSON documents stored in an Oracle database and to generate JSON documents from relational data.

Related Topics

- *Oracle® Database JSON Developer's Guide*

Oracle SQL and PL/SQL Improvements

- PL/Scope Reports on Static SQL Statements and Call Sites for Dynamic SQL
- Enhancing CAST Function With Error Handling
- New SQL and PL/SQL Function VALIDATE_CONVERSION
- Binding PL/SQL-Only Data Types to SQL Statements Using DBMS_SQL
- Improving the PL/SQL Debugger
- New PL/SQL Pragma to Mark an Item as Deprecated
- Materialized Views: Real-Time Materialized Views
- Materialized Views: Statement-Level Refresh
- Enhancing LISTAGG Functionality
- DBMS_PLSQL_CODE_COVERAGE Package
- Approximate Query Processing
- White Lists (ACCESSIBLE BY) Enhancements

PL/Scope Reports on Static SQL Statements and Call Sites for Dynamic SQL

The new view, `DBA_STATEMENTS`, reports on the occurrences of static SQL in PL/SQL units; listing, for example, the statement text, the type (`SELECT, INSERT, UPDATE,` or `DELETE`) and the `SQL_ID`. Dynamic SQL call sites (`EXECUTE IMMEDIATE, OPEN` *`cursor`* `FOR` *`dynamic text=""`*) are also listed in this view family. The `DBA_IDENTIFIERS` view family now reports on identifiers used in static SQL and notes their type (table, column, materialized view, sequence, and so on).

The purpose of PL/SQL is to issue SQL statements. Therefore, it is useful that PL/Scope now knows about the uses of SQL in PL/SQL source code. For example, if performance investigation reports the `SQL_ID` of a slow statement, its call sites in the PL/SQL programs can be found immediately. When PL/SQL source is under quality control for SQL injection risks, where dynamic SQL is used, the sites to look at can also be found immediately.

Related Topics

- *Oracle® Database Development Guide*

Enhancing CAST Function With Error Handling

The existing `CAST` function is enhanced to return a user-specified value in the case of a conversion error instead of raising an error.

This new functionality provides more robust and simplified code development.

Related Topics

- *Oracle® Database SQL Language Reference*

New SQL and PL/SQL Function VALIDATE_CONVERSION

The new function, `VALIDATE_CONVERSION`, determines whether a given input value can be converted to the requested data type.

The `VALIDATE_CONVERSION` function provides more robust and simplified code development.

Related Topics

- *Oracle® Database SQL Language Reference*

Binding PL/SQL-Only Data Types to SQL Statements Using DBMS_SQL

Oracle Database 12*c* Release 1 (12.1) introduced the ability to bind values of PL/SQL-only data types, most notably `PLS_INTEGER` tables of records to SQL statements. However, there were some restrictions which are lifted in this release. The PL/SQL-only data types can now be bound using the `DBMS_SQL` API and by invoking a C external procedure.

This improvement brings the `DBMS_SQL` API in parity with the native dynamic SQL.

Related Topics

- *Oracle® Database PL/SQL Packages and Types Reference*

Improving the PL/SQL Debugger

In prior releases, it was necessary to change the application to include calls to start and stop debugging. With this improvement, one database session can start debugging, and stop debugging a different session.

Also, when stopped at a breakpoint, it is possible for the debugging user to issue ad hoc SQL commands implying, therefore, the possibility to run PL/SQL code invoking stored PL/SQL subprograms in an anonymous block.

With these improvements, if a problem occurs in a long running test or in a production environment, it is possible to investigate the problem from another session. In addition to inspecting the state of in-scope variables, it is now possible to examine the database state as the session being debugged sees it during an uncommitted transaction.

Related Topics

- *Oracle® Database Java Developer's Guide*

New PL/SQL Pragma to Mark an Item as Deprecated

You can apply this new pragma to a whole unit, to a subprogram within a unit, or to any item defined in the unit to indicate that the unit or item is deprecated. When a unit that makes a reference to a deprecated element is compiled, a warning is displayed.

Sometimes, functionality caused existing code benefits to be reimplemented in incompatible ways. For example, the `UTL_CALL_STACK` package, which was new in Oracle Database 12*c* Release 1 (12.1), provided the functionality that earlier was provided by the `FORMAT_CALL_STACK(), FORMAT_ERROR_STACK(),` and `FORMAT_ERROR_BACKTRACE()` procedures in the `DBMS_UTILITY` package. The same thing happens in PL/SQL code that is developed by customers. The old API cannot be removed as soon as the new API is available because of the quantity of its existing uses; but new code that requires the functionality uses the new API. The new pragma provides a formal way to communicate this message with a power that ordinary external documentation cannot convey.

Related Topics

- *Oracle® Database PL/SQL Language Reference*

Materialized Views: Real-Time Materialized Views

Materialized views can be used for query rewrite even if they are not fully synchronized with the base tables and are considered stale. Using materialized view logs for delta computation together with the stale materialized view, the database can compute the query and return correct results in real time.

For materialized views that can be used for query rewrite all of the time, with the accurate result being computed in real time, the result is optimized and fast query processing for best performance. This alleviates the stringent requirement of always having to have fresh materialized views for the best performance.

Related Topics

- *Oracle® Database Data Warehousing Guide*

Materialized Views: Statement-Level Refresh

In addition to `ON COMMIT` and `ON DEMAND` refresh, the materialized join views can be refreshed when a DML operation takes place, without the need to commit such a transaction. This is predominantly relevant for star schema deployments.

The new `ON STATEMENT` refresh capability provides more flexibility to the application developers to take advantage of the materialized view rewrite, especially for complex transactions involving multiple DML statements. It offers built-in refresh capabilities that can replace customer-written trigger-based solutions, simplifying an application while offering higher performance.

Related Topics

- *Oracle® Database Data Warehousing Guide*

Enhancing LISTAGG Functionality

`LISTAGG` aggregates the values of a column by concatenating them into a single string. New functionality is added for managing situations where the length of the concatenated string is too long.

Developers can now control the process for managing overflowing `LISTAGG` aggregates. This increases the productivity and flexibility of this aggregation function.

Related Topics

- *Oracle® Database Data Warehousing Guide*

DBMS_PLSQL_CODE_COVERAGE Package

This feature introduces the new DBMS_PLSQL_CODE_COVERAGE package that enables you to collect data at the basic block level.

A basic block has a single entry point (no code within a basic block is the destination of a jump instruction) and a single exit point (only the last instruction, or an exception, can move the point of execution to a different basic block). This notion is independent of programming language, and is widely used in the presentation of the results of code coverage tests. PL/SQL developers can now use this standard quality metric.

Related Topics

- *Oracle® Database Development Guide*

Approximate Query Processing

This release extends the area of approximate query processing by adding approximate percentile aggregation. With this feature, the processing of large volumes of data is significantly faster than the exact aggregation. This is especially true for data sets that have a large number of distinct values with a negligible deviation from the exact result.

Approximate query aggregation is a common requirement in today's data analysis. It optimizes the processing time and resource consumption by orders of magnitude while providing almost exact results. Approximate query aggregation can be used to speed up existing processing.

Related Topics

- *Oracle® Database Data Warehousing Guide*

White Lists (ACCESSIBLE BY) Enhancements

In this release, you can define a white list on individual subprograms in a package. The `ACCESSIBLE BY` clause specifies a list of PL/SQL units that are considered safe to invoke the subprogram, and blocks all others.

Lowering the granularity increases the usefulness of the `ACCESSIBLE BY` clause for a package.

Reducing Costs and Complexities of Migration to Oracle Database

- Long Identifiers
- Oracle Java Virtual Machine Support for Long Identifiers
- Static PL/SQL Expressions Now Allowed Where Previously Literals Were Required

Long Identifiers

The maximum length of identifiers is increased to 128 bytes for most identifiers, up from 30 bytes in previous releases.

Providing longer identifiers gives customers greater flexibility in defining their naming schemes, such as longer and more expressive table names. Having longer identifiers also enables object name migration between databases with different character sets, such as Thai to Unicode.

Related Topics

- *Oracle® Database SQL Language Reference*

Oracle Java Virtual Machine Support for Long Identifiers

The maximum length of a SQL identifier is now 128 characters for Oracle Java Virtual Machine (Oracle JVM). In previous releases, the maximum length of a SQL identifier or a database schema object name was 31 characters, and all characters must be legal and convertible to the target database character set.

The full name of Java classes and methods usually exceeds the 31 character limit. To work around this issue, Oracle Database uses abbreviated names (short names) internally, but maintains a correspondence between short names and full names (long names).

Oracle JVM has been rearchitected to support long identifiers through any RDBMS command or utility which supports long names including: `SYNONYM, DDL, GRANT, REVOKE, [CREATE | ALTER | DROP] JAVA, DBMS_JAVA, DBMS_JAVA_DEFINERS, DBMS_JAVA_DUMP, DBMS_JAVA_MISC, DBMS_JAVA_TEST`, and OJVM utilities (`ojvmtc, loadjava, dropjava, ojvmjava,` and `ojds`).

A longer maximum length for SQL identifiers provides compatibility and integration with other RDBMS components. It also improves performance because there is little or no conversion needed between short names and long names in Oracle JVM.

Related Topics

- *Oracle® Database Java Developer's Guide*

Static PL/SQL Expressions Now Allowed Where Previously Literals Were Required

Some examples of places where, in earlier releases, a literal was required are:

- The length of the constraint in a `VARCHAR2` declaration.
- The precision and scale in a `NUMBER` declaration.

Now you can use expressions, but the values must allow computation at compile time.

You can now write PL/SQL programs so that the intention is self-evident without comments. You can also change PL/SQL programs to reflect changed requirements by making changes at far fewer sites. The canonical example is the `VARCHAR2` that holds the text of a simple SQL identifier. This needs to be 128 bytes plus 2 additional bytes; 128 bytes for the name, and 2 bytes to enable double quoting.

Related Topics

- *Oracle® Database PL/SQL Language Reference*

Support for the .NET and MS Development Community

- Longer Schema Identifiers for Oracle Data Provider for .NET
- Real Application Clusters Connection Node Affinity
- Entity Framework Code First Enhancements
- .NET Cloud Development and Deployment
- ODP.NET Managed Driver
- Longer Schema Identifiers for Oracle Provider for OLE DB

Longer Schema Identifiers for Oracle Data Provider for .NET

Oracle Data Provider for .NET now supports schema object identifier names, such as tables, columns, views, stored procedures, and functions, up to 128 characters in length. This feature is available in both the managed and unmanaged drivers.

The ability to use longer schema object identifier names increases database development flexibility and makes database migration to Oracle Database easier.

Real Application Clusters Connection Node Affinity

Oracle Data Provider for .NET (ODP.NET) connects to a specific Oracle RAC node when required in the case of a distributed transaction, Oracle RAC load balancing advisory, or Oracle RAC load balancing gravitation to an under-utilized node.

This feature enables better control over which Oracle RAC load balancing nodes the ODP.NET connections are dispensed from to ensure better reliability or performance.

Related Topics

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

Entity Framework Code First Enhancements

With Entity Framework Code First, developers define their object-relational map using .NET classes. These classes can then be mapped to an existing database or to generate a database schema. Managed and unmanaged ODP.NET support new Entity Framework Code First features with Oracle databases.

Developers that want to use the latest Entity Framework Code First features to develop Entity Framework applications can now do so with ODP.NET.

Related Topics

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

.NET Cloud Development and Deployment

With ODP.NET, Managed and Unmanaged Drivers can be deployed easily to common hosted and third-party cloud environments through Web deploy. All ODP.NET specific settings no longer require any operating system level configuration. These settings can be made in the .NET configuration files. Managed and Unmanaged ODP.NET Drivers now share a unified configuration file format.

Oracle .NET developers can easily deploy or migrate their applications to any cloud environment with a minimum of effort.

Related Topics

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

ODP.NET Managed Driver

Oracle Data Provider for Microsoft .NET (ODP.NET) Managed Driver adds new feature support matching the same functionality available in ODP.NET Unmanaged Driver.

.NET developers requiring these new features can now use ODP.NET Managed Driver for their applications.

Related Topics

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

Longer Schema Identifiers for Oracle Provider for OLE DB

Oracle Provider for OLE DB now supports schema object names, such as tables, columns, views, stored procedures, and functions up to 128 characters in length.

This feature increases the database development flexibility and makes the database migration to Oracle Database easier through the use of longer schema object identifier names.

Support for the Java Development Community

- Multi-Property Labeling of DRCP Servers
- PL/SQL Callback Facility for Session State Fix Up
- New MAX_TXN_THINK_TIME Setting for DRCP Pooled Servers With Transactions in Progress
- Proxy Session Sharing in DRCP
- DRCP Statistics Views and AWR Reports for Performance Monitoring and Tuning
- JDBC Support for Binding PLSQL_BOOLEAN
- Oracle JDBC Support for JDK 8 and JDBC 4.2
- Oracle Universal Connection Pool (UCP) Configuration Using XML
- Universal Connection Pool Health Check Frequency
- JDBC Support for Deprioritization of Database Node
- Oracle Java Virtual Machine (OJVM) Web Services Callout
- OJVM Support for Java SE 8
- JavaScript Stored Procedures Using Java 8 Nashorn

Multi-Property Labeling of DRCP Servers

Database Resident Connection Pool (DRCP) connections used by Oracle Call Interface (OCI) clients can retain session state such as language settings and date formats. Multi-property labeling allows connections to be tagged to record each of these properties for later matching.

When getting a DRCP connection, being able to specify the required properties of that connection with finer granularity avoids applications having to re-create session state unnecessarily. This can improve scalability and performance.

Related Topics

- *Oracle® Call Interface Programmer's Guide*

PL/SQL Callback Facility for Session State Fix Up

This feature provides a PL/SQL callback facility to enable applications to fix up the state in the Database Resident Connection Pooling (DRCP) pooled servers to the desired state.

Applications do not need to handle the state fix up in the application and have separate code paths for DRCP and non-DRCP deployments. This feature provides a consistent and transparent implementation of session state fix up.

New MAX_TXN_THINK_TIME Setting for DRCP Pooled Servers With Transactions in Progress

`MAX_TXN_THINK_TIME` is a new Database Resident Connection Pooling (DRCP) parameter that specifies the think timeout for pooled servers with transactions in progress. When the timeout occurs, sessions rollback and terminate. The new parameter is separate from the existing `MAX_THINK_TIME` that applies to connections that have no transactions in progress.

Customers can now set a higher think time for connections with transactions in progress so that these connections timeout later than connections that do not have transactions. This allows efficient pool reuse while giving incomplete transactions a longer time to be concluded.

Related Topics

- *Oracle® Database Administrator's Guide*

Proxy Session Sharing in DRCP

This new feature enables sharing of pooled servers in Database Resident Connection Pooling (DRCP) among applications that connect as different proxy users using the same schema.

Proxy session sharing increases the sharing of DRCP pooled servers and enhances the performance of applications.

Related Topics

- *Oracle® Database Development Guide*

DRCP Statistics Views and AWR Reports for Performance Monitoring and Tuning

This feature furnishes views for Database Resident Connection Pooling (DRCP) statistics and additional Automatic Workload Repository (AWR) snapshots and reports for monitoring DRCP.

This enhancement provides better manageability and diagnosability of DRCP performance and scalability.

Related Topics

- *Oracle® Database Administrator's Guide*

JDBC Support for Binding PLSQL_BOOLEAN

A SQL boolean is of type `NUMBER` where as PL/SQL boolean is a true `BOOLEAN` much like in Java. With this feature, JDBC supports the ability to `BIND PLSQL_BOOLEAN` type into any PL/SQL block from Java.

The feature provides seamless mapping of `PLSQL_BOOLEAN` to Java `BOOLEAN`. It also provides the ability to use `PLSQL_BOOLEAN` as `IN, OUT, IN OUT` parameters in Java or JDBC programs.

Related Topics

- *Oracle® Database JDBC Developer's Guide*

Oracle JDBC Support for JDK 8 and JDBC 4.2

In Oracle Database 12*c* Release 1 (12.1), the Oracle JDBC drivers furnish complete support for the JDBC 4.2 standard through JDK 8 including the `setObject()` method which complements the `getObject()` method introduced in the last release of Oracle Database.

This feature provides portability enabled by Java SE 8 and JDBC 4.2 standards.

Related Topics

- *Oracle® Database JDBC Developer's Guide*

Oracle Universal Connection Pool (UCP) Configuration Using XML

Users can configure Oracle Universal Connection Pool (UCP) using XML to specify pool properties, such as `minPoolSize` and `maxPoolSize`. The location of the XML file is specified either in the `oracle.ucp.jdbc.oracle.xmlConfigFile` system property or in the `setXMLConfigFile()` API. The corresponding XML schema (`ucp-configuration.xsd`) is included in the `ucp.jar` file.

The non-intrusive configuration of Oracle Universal Connection Pool (UCP) simplifies manageability for custom and packaged Java applications for which source codes are not available.

Related Topics

- *Oracle® Universal Connection Pool Developer's Guide*

Universal Connection Pool Health Check Frequency

This feature improves performance by reducing the frequency of the connection health validation check, using the following methods:

- `public void setSecondsToTrustIdleConnection()`

When you use this in conjunction with `ValidateConnectionOnBorrow()`, the connection is assumed valid for the next number of specified seconds to trust an idle connection. For example, if the value is set to `30`, then the connection pool trusts the validity of the connection for 30 seconds.

- `public int getSecondsToTrustIdleConnection()`

This property gets the value, in seconds, of the `SecondsToTrustIdleConnection()property`.

The Universal Connection Pool feature which increases Java applications performance by reducing the overhead and frequency of connection health validation during heavy traffic.

Related Topics

- *Oracle® Universal Connection Pool Developer's Guide*

JDBC Support for Deprioritization of Database Node

When a node fails, JDBC deprioritizes it and does not attempt to allocate connections from that host for the next 10 minutes (the default expiry time). For example, if there are three nodes A, B, C, and A is down at some point in time, then connections are allocated from nodes B and C, first, and then finally to node A. After the expiry time, node A is no longer de-prioritized. Users can specify the default expiry time for deprioritization using the `SQLNET.DOWN_HOSTS_TIMEOUT` system property.

This feature improves high-availability, performance and availability by reducing the time required to connect to a valid host.

Related Topics

- *Oracle® Database JDBC Developer's Guide*

Oracle Java Virtual Machine (OJVM) Web Services Callout

Web services callout is a new utility that enables the invocation of external web services from within a database session (SQL, PL/SQL or Java). The tool can do the following:

- Obtain the Web Services Description Language (WSDL) of a published Web service server furnished as a parameter.
- Recuperate and load the Web service client (for example, Java proxy) in an Oracle Java Virtual Machine (OJVM).
- Generate the PL/SQL wrapper and grant the proper permissions.

The Web services callout supports basic and advanced Web services authentication (for example, SSL).

This feature integrates the database in a service-based architecture. It augments database processing with dynamically produced external data. A typical use case is credit card or financial securities validation.

Related Topics

- *Oracle® Database Java Developer's Guide*

OJVM Support for Java SE 8

The database resident Oracle Java Virtual Machine (OJVM) now supports Java SE 8. Some JDK 8 features, including JavaFX, command line tools, Java tools, GUI interfaces, HotSpot, Mission Control, and so on are not supported in OJVM.

Support for Java SE 8 provides portability of Java applications and libraries and compliance with the latest Java standards. Reuse Java SE 8 applications directly in the database for data-bound processing, thereby avoiding data shipping to an external infrastructure.

Related Topics

- *Oracle® Database Java Developer's Guide*

JavaScript Stored Procedures Using Java 8 Nashorn

This feature leverages Nashorn (the JavaScript engine on the JVM) to support the execution of JavaScript modules directly on Oracle Java Virtual Machine (OJVM) in an Oracle database session. The process consists of loading your JavaScript code as Java resource in a user schema using the `loadjava` utility and then invoke it through a SQL wrapper. Using Nashorn, you can invoke Java APIs from JavaScript; for example, data access from within JavaScript using the server-side Java Database Connectivity (JDBC) driver in OJVM.

Using Nashorn provides an alternative to PL/SQL and Java for implementing stored procedures. The ability to run JavaScript in the database provides the following benefits:

- Reuse data-bound JavaScript code in the database.
- Use JavaScript developers to implement database modules.
- Avoid data shipping; for example, in-place processing of billions of data or JavaScript Object Notation (JSON) documents.
- In a service-based architecture, JavaScript stored procedures can be invoked through RESTful frameworks (Oracle REST Data Services (ORDS) and JAX-RS) to implement Cloud Data Services.

Related Topics

- *Oracle® Database Java Developer's Guide*

XML

- Enhancing the Oracle XML Developers Kit for Java (XDK/J)
- Loading Sub-Documents from XML Documents Using ORACLE_LOADER

Enhancing the Oracle XML Developers Kit for Java (XDK/J)

In this release, the Java version of the Oracle XSL processor is extended to provide complete support for the W3C XSL 2.0 Standard. The Oracle XQuery Java engine is extended to provide limited support for the forthcoming XQuery 3.0 recommendation, the XSLT 2.0 completion, and the XQuery 3.0 subset.

Developers can take advantage of the features and power of the XSL 2.0 specification when developing XML-based applications. They are also able to start making use of the feature set of XQuery 3.0.

Related Topics

- *Oracle® XML Developer's Kit Programmer's Guide*

Loading Sub-Documents from XML Documents Using ORACLE_LOADER

Many XML documents are a concatenation of multiple documents of the same type. For example, a small document describes an article in a technical journal and an XML document contains an array of those documents. This feature enables you to make the `ORACLE_LOADER` access driver extract the smaller documents that describe an article and load each one as a separate row in a table. You specify the tag that delimits the smaller document as part of the access parameters.

This feature enables easier and faster loading of XML documents that are concatenated into larger XML documents.

Related Topics

- *Oracle® Database Utilities*

Availability

- Data Guard
- End-to-End Application Availability
- General
- Logical Replication
- Online Operations
- Recovery Server and RMAN Improvements
- Sharding
- Simplifying Upgrades

Data Guard

- Distributed Operations on CLOB, BLOB and XMLType
- OCI Support for Distributed LOBs
- Minimizing Impact on Primary Database When Using Multiple SYNC Standby Databases
- Oracle Data Guard Database Compare
- Subset Standby
- Oracle Data Guard Broker Support for Multiple Automatic Failover Targets
- Oracle Data Guard Broker Support for Multiple Observers
- Simplifying Observer Management for Multiple Fast-Start Failover Configurations
- Oracle Data Guard Broker Support for Transport Destinations of Different Endianess Than the Primary
- Oracle Data Guard Broker Support for Oracle Data Guard Multiple Instance Apply
- Oracle Data Guard Broker Support for Enhanced Alternate Destination
- Fast-Start Failover in Maximum Protection Mode

- Block Comparison Tool Support in Oracle Data Guard Broker DGMGRL Interface
- Oracle Data Guard Broker Support for Executing DGMGRL Command Scripts
- Broker ConfigurationWideServiceName Configuration Property
- Enhancing Support for Alternate Destinations
- Automatically Synchronize Password Files in Oracle Data Guard Configurations
- Preserving Application Connections to An Active Data Guard Standby During Role Changes

Distributed Operations on CLOB, BLOB and XMLType

In this release, support for operations over database links for LOB-based data types, such as `CLOB`, `BLOB` and `XMLType`, is available.

This support enables operations on LOB-based data types across pluggable databases (PDBs) in an Oracle Multitenant environment.

Related Topics

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

OCI Support for Distributed LOBs

You can now query and update LOB data using a LOB locator selected from a database link to a remote database.

This feature extends the usefulness of databases that are accessed using database links.

Minimizing Impact on Primary Database When Using Multiple SYNC Standby Databases

The new parameter `DATA_GUARD_SYNC_LATENCY` allows you to determine how long the primary database should wait for a response from multiple synchronous standby databases during redo transport. The existing `NET_TIMEOUT` redo transport attribute allows you to specify how long the primary database should wait for a response from each synchronous standby database. When multiple synchronous standby database destinations are configured, the primary database must wait for all of the synchronous standby databases to acknowledge receipt of the redo or exceed their individual `NET_TIMEOUT` period before continuing. This new parameter is global for all synchronous standby database destinations and defines the maximum amount of time (in seconds) that the primary database must wait before disconnecting subsequent destinations after at least one synchronous standby database has acknowledged receipt of the redo. For example, let's say there are three synchronous standby database destinations configured and this parameter is set to a value of 2. If the first standby database acknowledges receipt of the redo immediately, then the primary database waits no longer than 2 seconds for the other two standby databases to respond. If one or both standby databases respond within 2 seconds, they are maintained as active destinations. If one or both fail to respond within the required 2 seconds, they are marked as failed. In both cases, the primary database continues on, remaining in its zero data loss protection mode because one synchronous standby database has acknowledged receipt of the redo. Any failed synchronous standby databases are reconnected as normal after the `REOPEN` seconds have passed.

Users with more than one synchronous standby database destination can reduce the impact of transient network failures by allowing the primary database to continue as long as one synchronous standby database has all the redo.

Related Topics

- *Oracle® Data Guard Concepts and Administration*

Oracle Data Guard Database Compare

This new tool compares data blocks stored in an Oracle Data Guard primary database and its physical standby databases. Use this tool to find disk errors (such as lost write) that cannot be detected by other tools like the `DBVERIFY` utility.

An administrator can validate that a standby database does not contain silent corruptions introduced independently by the I/O stack at a standby database. Oracle Data Guard already performs validation on hot data (data being read or changed) at either the primary or standby database, but this new tool provides comprehensive validation, including cold data that has not been read or changed by Oracle Data Guard. This ability provides the administrator with complete confidence that the standby database is free of physical corruption.

Related Topics

- *Oracle® Data Guard Concepts and Administration*

Subset Standby

A subset standby enables users of Oracle Multitenant to designate a subset of the pluggable databases (PDBs) in a multitenant container database (CDB) for replication to a standby database. The standard practice is to have all PDBs in a container to be subject to the same service level with regard to high availability (HA). This results in a standard practice where if one PDB requires a standby database, then all PDBs that reside in the same container would also require a standby database. This is not always the case, however, in an Oracle Active Data Guard environment where only a subset of PDBs are required to support read-only operations (a production offload feature versus an HA service level). A subset standby provides a simple way for Oracle Data Guard to limit synchronization to the subset of PDBs that are of interest at the standby database. This feature must be carefully implemented because it changes the usual semantics of Oracle Data Guard switchover and failover operations since not every PDB is present in both primary and standby databases.

A subset standby reduces storage and processing requirements when only a portion of the PDBs in a primary database are desired to be replicated to the standby database.

Related Topics

- *Oracle® Data Guard Concepts and Administration*

Oracle Data Guard Broker Support for Multiple Automatic Failover Targets

Oracle Data Guard now supports multiple failover targets in a fast-start failover configuration. Previous functionality allowed for only a single fast-start failover target. If the failover target was unable to meet the requirements for fast-start failover at the time of primary failure, then an automatic failure would not occur. Designating multiple failover targets significantly improves the likelihood that there is always a standby suitable for automatic failover when needed.

Multiple failover targets increase high availability by making an automatic failover more likely to occur if there is a primary outage.

Related Topics

- *Oracle® Data Guard Broker*

Oracle Data Guard Broker Support for Multiple Observers

It is now possible to configure multiple observers within a single Oracle Data Guard broker configuration.

Multiple observers provide an immediate benefit for high availability. If one observer fails, there are additional observers that can continue to monitor the status of the configuration. These additional observers eliminate the possibility of fast-start failover becoming disabled because the configuration is in an unobserved state.

Related Topics

- *Oracle® Data Guard Broker*

Simplifying Observer Management for Multiple Fast-Start Failover Configurations

This feature enables a single Oracle Data Guard broker DGMGRL session to manage and monitor observers from multiple fast-start failover configurations. This simplifies the configuration and management of fast-start failover in cases where automatic failover is desired for more than one database.

This allows the observer management for many fast-start failover configurations to be as one from the perspective of the broker thereby reducing operational complexity and cost.

Related Topics

- *Oracle® Data Guard Broker*

Oracle Data Guard Broker Support for Transport Destinations of Different Endianess Than the Primary

Oracle Data Guard broker can now manage a remote redo destination that is on a platform of different endianess than the primary database. This allows the Oracle Data Guard broker to manage and configure Oracle Data Guard transport services to Recovery Appliance with different endianess than the primary database.

This feature increases the flexibility of the Oracle Data Guard broker when Oracle Data Guard transport services are used in a heterogeneous configuration where the primary and a Recovery Appliance are of different endianess. Users benefit from broker automation when managing such a configuration.

Oracle Data Guard Broker Support for Oracle Data Guard Multiple Instance Apply

The Oracle Data Guard broker DGMGRL command-line interface allows you to configure and manage on which instances in an Oracle RAC standby the apply processes should be executed to use the new Oracle Active Data Guard multiple instance Redo Apply feature.

Oracle Data Guard broker support for the new multi-instance Recovery Apply feature of Oracle Data Guard allows users who employ the broker to continue to manage standby databases from the DGMGRL interface and allow the broker to fully automate the Redo Apply at all physical standby databases.

Oracle Data Guard Broker Support for Enhanced Alternate Destination

Oracle Data Guard broker users can now use the DGMGRL command-line interface to configure and implement the enhanced `ALTERNATE` redo destinations capability.

This enhancement supports the new `ALTERNATE` syntax of Oracle Data Guard in the broker.

Related Topics

- *Oracle® Data Guard Broker*

Fast-Start Failover in Maximum Protection Mode

You can now configure fast-start failover when Oracle Data Guard is running in maximum protection mode.

Users are now able to use automatic failover in the guaranteed zero data loss mode utilizing multiple fast-start failover targets when there are multiple synchronous destinations.

Related Topics

- *Oracle® Data Guard Broker*

Block Comparison Tool Support in Oracle Data Guard Broker DGMGRL Interface

The new database comparison tool, introduced in this release, is also supported by Oracle Data Guard broker. The PL/SQL package that compares blocks across primary and standby databases can also be executed from the Oracle Data Guard broker DGMGRL command-line interface.

This support enables database administrators, who use the standard interface to Oracle Data Guard (DGMGRL) in their broker configurations, to perform block comparisons between the primary database and any physical standby database.

Related Topics

- *Oracle® Data Guard Broker*

Oracle Data Guard Broker Support for Executing DGMGRL Command Scripts

This feature enables command scripts to be executed through the Oracle Data Guard broker DGMGRL command-line interface; much like in SQL*Plus. DGMGRL commands, SQL commands using the broker SQL command, and operating system commands using the new `HOST` (or `!`) capability can be put into a file and executed on the DGMGRL command line.

This feature enables customers who use the standard interface to Oracle Data Guard (DGMGRL) in their broker configurations to use scripts for Oracle Data Guard operations.

Related Topics

- *Oracle® Data Guard Broker*

Broker ConfigurationWideServiceName Configuration Property

In previous releases, the Oracle Data Guard broker lacked a `ConfigurationWideServiceName` configuration property shared by all members. In particular, there was no guarantee that you could use a single connect identifier to connect to any database in the configuration because each database had its own services running, which may or may not have had the same service name. In Oracle Database 12*c* Release 2 (12.2), Oracle Data Guard broker can publish a service on each member in a configuration with a unified service name. This capability also enables the new capabilities of the Oracle Data Guard broker observer for automatic failover control.

Having a `ConfigurationWideServiceName` configuration property allows users to easily connect to any database in the configuration and enables the new capabilities of the fast-start failover observer in Oracle Database 12*c* Release 2 (12.2).

Related Topics

- *Oracle® Data Guard Broker*

Enhancing Support for Alternate Destinations

The concept of an alternate destination for redo transport was first introduced in Oracle Database release 9.2 and was designed to work as a single pair of destinations with simple failover when the initial destination was no longer available. In Oracle Database 12*c* Release 1 (12.1), this was enhanced to use one preferred destination, and the second as an alternate destination, with automatic reconfiguration and simple failback when the initial redo destination was once again available. The new Oracle Active Data Guard far sync and real-time cascade features introduced a number of additional use cases for alternate destinations that the Oracle Database 12*c* Release 1 (12.1) model was unable to support. This enhanced support for alternate destinations greatly increases the number of different use cases that can now be supported with complete user configuration control and completely automatic management.

This feature greatly expands the number of alternate destinations that you can define. It provides greater flexibility to designate rules that define priority, and it provides numerous options for defining policies in a failure state. The collective abilities of these enhancements make this feature much more robust in its ability to be configured to meet various data protection and availability goals, regardless of the configuration used.

Related Topics

- *Oracle® Data Guard Concepts and Administration*

Automatically Synchronize Password Files in Oracle Data Guard Configurations

This feature automatically synchronizes password files across Oracle Data Guard configurations. When the passwords of `SYS`, `SYSDG`, and so on, are changed, the password file at the primary database is updated and then the changes are propagated to all standby databases in the configuration.

This feature provides additional automation that further simplifies management of Oracle Data Guard configurations.

Related Topics

- *Oracle® Data Guard Concepts and Administration*

Preserving Application Connections to An Active Data Guard Standby During Role Changes

Currently, when a role change occurs and an Active Data Guard standby becomes the primary, all read-only user connections are disconnected and must reconnect, losing their state information. This feature enables a role change to occur without disconnecting the read-only user connections. Instead, the read-only user connections experience a pause while the state of the standby database is changed to primary. Read-only user connections that use a service designed to run in both the primary and physical standby roles are maintained. Users connected through a physical standby only role continue to be disconnected.

This feature improves the user experience and facilitates improved reporting and query capabilities on an Active Data Guard standby during role transitions.

End-to-End Application Availability

Application Continuity for OCI Applications

Application developers were required to deal explicitly with outages of the underlying software, hardware, and communications layers if they wanted to mask outages from users. Since Oracle Database 10*g*, Fast Application Notification (FAN) delivered exception conditions to applications fast. However, neither FAN nor earlier Oracle Database technology reported the outcome of the last transaction to the application or recovered the in-progress request from an application perspective. As a result, outages were exposed leading to user inconvenience and lost revenue. Users could unintentionally make duplicate purchases and submit multiple payments for the same invoice. In the problematic cases, the administrator needed to reboot the mid-tier to deal with the incoming problems this caused. Application Continuity is an application-independent feature that attempts to recover incomplete requests from an application perspective and masks many system, communication and hardware failures, and storage outages from the user.

The protocol ensures that user transactions are executed no more than once. When successful, the only time that a user should see an interruption in service is when there is no point in continuing. When replayed, the execution appears to the application and client as if the request was slightly delayed. The effect is similar to a loaded system where the database runs the request slightly slower so that the response to the client is delayed. Most failures should be masked. This results in fewer calls to the error handling logic of the application. For example, less often, the application raises an error leaving the user not knowing what happened or forces the user to reenter data. Or, more problematic, the administrators must restart the mid-tier servers to cope with the failure. Other benefits include:

- Improved user experience.
- Higher application availability.
- Improved application developer productivity.

Related Topics

- *Oracle® Call Interface Programmer's Guide*

Application Continuity

Application Continuity recovers incomplete requests from an ODP.NET, unmanaged driver perspective and masks many system failures, communication failures, hardware failures, and storage outages from the user.

Application Continuity ensures that transactions are executed no more than once. When failures do occur, they are generally masked from the user. This feature leads to improved user experience, higher application availability, and improved ODP.NET developer productivity.

Related Topics

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

Transaction Guard for Oracle XA Transactions

This feature completes Transaction Guard support for Oracle transaction types with the addition of XA-based transactions. The feature supports an unambiguous commit outcome for WebLogic Server and Oracle Tuxedo, promotable XA transactions for `ODP.NET`, and promotable transactions for WebLogic Server. Transaction Guard with XA provides the missing requirement that has prevented safe replay following recoverable outages for XA transactions. With the addition of XA support, WebLogic Server and others can more easily provide replay with idempotence enforced using Transaction Guard.

The Transaction Guard with XA feature removes end-user uncertainty about the fate of a one-phase XA transaction that was in-flight and could commit when the application received a recoverable error. Using Transaction Guard, a transaction manager can now return reliable outcome for XA transactions to the user. Transaction Guard with XA provides the missing requirement that has prevented safe replay following recoverable outages for Application Continuity when used with promotable XA, and for WebLogic Server replay.

Related Topics

- *Oracle® Real Application Clusters Administration and Deployment Guide*

Java Support for FAN APIs (UP, DOWN and LoadAdvisory Events)

The SimpleFAN library (`simplefan.jar`) is enhanced with a new capability to identify `UP` events. The `oracle.simplefan.FanUpEventListener` interface is enhanced with two new methods, `NodeUpEvent()` and `ServiceUpEvent()`. Note that `NodeUpEvent()` is not supported at the server side, but the forward compatible client API is included in `simplefan.jar`. `ServiceDown`, `LoadAdvisory`, and `NodeDown` events are already supported in a previous release.

Java containers, frameworks, and applications looking to handle FAN events can use these APIs to subscribe to Oracle Database RAC FAN events for building high availability solutions.

Related Topics

- *Oracle® Database JDBC Developer's Guide*

JDBC Driver Support for Fast Application Notification (FAN)

The Oracle JDBC drivers now support Oracle Database RAC FAN events for enhanced support for planned maintenance and unplanned down times:

- `oracle.jdbc.fanEnabled` - A new system property to enable or disable the FAN support in the driver. If Oracle Universal Connection Pool (UCP) is used as a client-side pool, then UCP takes precedence.
- `oracle.jdbc.fanONSConfig` - A new connection property to be used by the driver as a remote Oracle Notification Services (ONS) subscription. This property is required only for database releases prior to 12*c*.

Java applications, containers, and frameworks get the benefits of FAN events; for example, processing `UP` and `DOWN` through the JDBC driver without the need for Oracle Universal Connection Pool (UCP). The benefits include better support for planned maintenance and unplanned outages with less integration work.

Related Topics

- *Oracle® Database JDBC Developer's Guide*

Support for Planned Outages

The Oracle Call Interface (OCI) session pool currently supports a scheme to drain work away from an instance prior to a planned outage. Applications that do not use the OCI session pool do not have the ability to easily drain work from a target instance. This feature introduces APIs that can be used by applications that deploy a custom pool, or use no pool at all, to communicate to OCI a suitable point at which to terminate connections prior to a planned outage. It is anticipated that these APIs are able to respond to both planned and unplanned outages. In addition to the new APIs, this feature also provides better support to failover at suitable points in time. By implicitly restoring a session state that is already synchronized to the client side to the post-failover session, OCI eliminates the requirement to write a failover callback.

Planned outages, for example to apply a patch or to conduct routine maintenance, occur much more frequently than unplanned outages. In a high availability (HA) configuration, the database tier can tolerate a planned outage by offering services on a different instance or a standby replica. Applications, on the other hand, are often written to expect near continuous connectivity to the database. By enhancing key client drivers, Oracle Database delivers a solution to:

- Reduce the frequency of customer-visible errors that often result during a planned outage.
- Limit the side effects of the error when an error is unavoidable.

Related Topics

- *Oracle® Call Interface Programmer's Guide*

Planned Outage for Oracle Data Guard Switchover and Oracle RAC One Node

ODP.NET already supports planned outage, which allows a database being brought offline to automatically alert ODP.NET applications of the impending downtime. ODP.NET then stops allocating new connections and close connections returned to the pool from that particular instance. To enhance planned outage functionality for an Oracle Data Guard switchover and Oracle Real Application Clusters One Node, new connection requests are paused until the transition to the new instance occurs. The pause allows new connections to wait until the new instance can accept connections without the end user experiencing a timeout.

This feature enables better end user quality of service. Oracle Data Guard switchovers and Oracle RAC One Node transitions are more seamless during planned outages without requiring additional application logic.

Related Topics

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

Application Continuity for Planned Database Maintenance

There is no reason for scheduled maintenance to be burdensome for application users. Following very simple configuration and operational practices, you can completely hide the adverse effects of scheduled maintenance from all application types. Using the correct setup, no errors are reported for either new, incoming work, or for existing work while maintenance is in progress.

For scheduled maintenance in Oracle Database 12*c* Release 2 (12.2), Fast Application Notification (FAN) is embedded in the Java Database Connectivity (JDBC) thin driver to support all Oracle Java solutions for planned and unplanned outages. Additionally, Transparent Application Failover (TAF) is extended with Transaction Guard to fail over more OCI-based applications transparently. As well, Oracle Data Provider for Microsoft .NET (ODP.NET) adds support for Application Continuity. This feature set provides a simple to use and fool-proof way for most applications to continue operation during scheduled maintenance. All of these features come with nil application changes required in almost all cases. A one command interface is also introduced for Server Control Utility (SRVCTL), Global Data Services Control Utility (GDSCTL), and Oracle Data Guard broker to enable database administrators to control the scheduled maintenance for all applications with a single command from the database.

For scheduled maintenance at the database server, Oracle Database 12*c* Release 2 (12.2) provides:

- Nil errors reported to applications for new, incoming work and existing work while maintenance is in progress.
- Transparent redirection of the active database sessions to another functional service.
- A one command orchestration focuses on the best possible application experience, as opposed to the maintenance applied, as quickly as possible, with no consideration for the applications.
- Re-balancing of database sessions, as needed, during and after the maintenance completes.

This feature hides scheduled maintenance operations that are required for the underlying infrastructure (Oracle Database, Oracle Grid Infrastructure, operating system, and hardware) without placing a burden on application developers. The solution differentiates Oracle Database by enabling you to conduct scheduled maintenance without interruption to most application work.

Related Topics

- *Oracle® Real Application Clusters Administration and Deployment Guide*

General

- Multi-Instance Redo Apply
- Oracle Data Guard for Data Warehouses
- Data Guard Broker PDB Migration or Failover

Multi-Instance Redo Apply

Releases prior to Oracle Database 12*c* Release 2 (12.2) limited Redo Apply (physical standby database) to a single instance on an Oracle RAC standby database. Redo Apply can now run on all or some standby instances as configured by the user. This enables Redo Apply performance to scale, if needed, by adding additional standby instances.

With this new feature, recovery time objectives can be achieved for any primary workload. This is particularly important for Oracle Exadata customers and customers with large Oracle RAC clusters. Oracle Active Data Guard users also have real-time access to current information. Scaling apply performance in this manner means that the standby database is always up-to-date even at very high volumes on the largest Oracle RAC clusters.

Related Topics

- *Oracle® Data Guard Concepts and Administration*

Oracle Data Guard for Data Warehouses

The use of `NOLOGGING` for direct loads on a primary database has always been difficult to correct on an associated standby database. On a physical standby database the data blocks were marked unrecoverable and any SQL operation that tried to read them would return an error. Or, for a logical standby database, SQL apply would stop upon encountering the invalidation redo.

For a physical standby database, these blocks are now recorded in the control file of the standby database and can be repaired using a new RMAN `RECOVER DATABASE NONLOGGED BLOCK` command in a simple and expedient manner.

Related Topics

- *Oracle® Data Guard Concepts and Administration*

Data Guard Broker PDB Migration or Failover

In multitenant broker configurations, you may need to move a Production PDB from one container database to another container database that resides on the same system. Users may also need to failover a PDB from a Data Guard Standby database to a new production container database when the production PDB has failed but the container database and all other PDBs function normally. The new Data Guard Broker DGMGRL command, `MIGRATE PLUGGABLE DATABASE`, enables you to unplug a PDB from the primary database and plug it into another container database on the primary system as well as failover a single PDB by unplugging the PDB from the Data Guard standby container database and plugging it into an existing new primary container database on the standby system. In both cases, the data files for the PDB being moved must reside on storage that is visible to the current container database and the target container database. When a PDB is migrated in this manner, the data files must be copied from the new container database to its Data Guard standby and recovery must be enabled for the new PDB. When failing over a PDB from a Data Guard standby to a new primary container database, the database version must be the same on both the Data Guard standby and the target container database. When moving a PDB from one container to another on the primary system, the version can be equal or higher than the current container but the PDB must be upgraded before it can be used.

Using the new Data Guard Broker command, `MIGRATE PLUGGABLE DATABASE`, you can easily move a single PDB from one container database to another, or failover a single PDB from a Data Guard standby to a new production container database.

Related Topics

- *Oracle® Data Guard Broker*

Logical Replication

- Error Handling Improvements for Oracle GoldenGate Integrated Replicat
- Abstract Data Types Support
- Oracle GoldenGate Annotations for DDL
- XStream Inbound Server Performance Optimizations

Error Handling Improvements for Oracle GoldenGate Integrated Replicat

Database views can manage and display details about Oracle GoldenGate conflict resolution methods, including DBA_APPLY_REPERROR_HANDLERS, DBA_APPLY_HANDLECOLLISIONS, DBA_APPLY_DML_CONF_HANDLERS, and DBA_APPLY_DML_CONF_COLUMNS.

With these improvements, you can view the configuration of conflict management within the database.

Related Topics

- *Oracle® Database XStream Guide*

Abstract Data Types Support

XStream data type support is extended to include tables with Abstract Data Types (user-defined types). Support is provided for both XStream outbound and inbound servers.

XStream outbound and inbound servers are extended to include tables with SDO_GEOMETRY and other simple user-defined types.

Related Topics

- *Oracle® Database XStream Guide*

Oracle GoldenGate Annotations for DDL

A new capture parameter, ANNOTATE_DDL_SIZE, is available for Oracle GoldenGate integrated capture to specify the size of the buffer that describes the DDL more fully.

Descriptive information for DDL is provided to Oracle GoldenGate integrated extract to simplify filtering and transformation of DDL within Oracle GoldenGate.

Related Topics

- *Oracle® Database XStream Guide*

XStream Inbound Server Performance Optimizations

Use the apply parameter, BATCHSQL_MODE, to control the parallel apply scheduling of batched transactions. In addition, dependency computation now includes object level dependencies.

Dependency computation and scheduling optimizations enable greater apply parallelism and faster apply throughput.

Related Topics

- *Oracle® Database XStream Guide*

Online Operations

- Restarting Redefinition From Failure
- Rolling Back Redefinition

- Redefinition Progress Monitoring
- Optimizing Batch Update During Redefinition
- Materialized Views: Fast Dependent Materialized View Refresh During Redefinition
- Redefinition Supports Tables With BFILE Columns
- Formalizing Retirement of an Edition With Automatic Garbage Collection
- Online Conversion of a Nonpartitioned Table to a Partitioned Table
- Online SPLIT Partition and Subpartition
- Online Table Move
- Partitioning: Table Creation for Partition Exchange
- Partitioning: Filtered Partition Maintenance Operations

Restarting Redefinition From Failure

Online redefinition can restart from most points of failure, eliminating the need to redo the entire operation.

Restart reduces the overall redefinition time.

Related Topics

- *Oracle® Database Administrator's Guide*

Rolling Back Redefinition

There is a new `ROLLBACK` parameter for the `FINISH_REDEF_TABLE` procedure that tracks DML on a newly redefined table so that changes can be easily synchronized with the original table using the `SYNC_INTERIM_TABLE` procedure.

There is also a new `ROLLBACK` procedure for the `DBMS_REDEFINITION` package that initiates the swapping of the redefined table with the original table, therefore effectively rolling back the redefinition changes and returning the table to its original state.

If the results of a redefinition are not acceptable (for example, a performance slow down accessing a newly redefined table), then the redefinition changes can be easily rolled back, therefore saving the DBA time in performing another redefinition to undo the table changes.

Related Topics

- *Oracle® Database Administrator's Guide*

Redefinition Progress Monitoring

The new `V$ONLINE_REDEF` view displays runtime information related to the current redefinition procedure being executed based on a redefinition session identifier.

This new view enables DBAs the ability to monitor the progress of their redefinition session and take corrective action if needed.

Related Topics

- *Oracle® Database Administrator's Guide*

Optimizing Batch Update During Redefinition

Redefinition requires ARCHIVELOG mode to be enabled. However, a new DBMS_REDEFINITION.EXECUTE_UPDATE procedure enables the execution of UPDATE statements in direct insert mode, eliminating redo logging and the associated overhead during that time period. Since redo is not logged during that period, the user must be aware that the redefinition and UPDATE changes cannot be recovered using media recovery. A database or tablespace backup must be taken after the redefinition completes to preserve recoverability.

For customers who run predictable, batch updates (for example, data warehouse) in direct insert mode to avoid redo overhead and the resulting impact to transaction throughput, they can now run batch updates in the same manner during a redefinition session, therefore reducing the overall time to complete the redefinition.

Related Topics

- *Oracle® Database Administrator's Guide*

Materialized Views: Fast Dependent Materialized View Refresh During Redefinition

The new SYNC_INTERIM_TABLE procedure for the DBMS_REDEFINITION package enables incremental refresh of dependent materialized views on the interim table. The SYNC_INTERIM_TABLE procedure refreshes the materialized view as part of its execution.

All dependent materialized views on the source table must be fully refreshed after redefinition completes, increasing the time when the table is fully usable. Refreshing the materialized views during the redefinition process eliminates the time to perform a complete refresh at the end.

Related Topics

- *Oracle® Database Administrator's Guide*

Redefinition Supports Tables With BFILE Columns

Tables with BFILE columns are now supported by redefinition.

This support increases the flexibility of online redefinition by expanding data type support.

Formalizing Retirement of an Edition With Automatic Garbage Collection

In earlier releases, the intention that an edition should never again be used could be expressed only by removing the use privilege from all users. However, doing this does not prevent SYS from using the edition; and nor does it prevent SYS from later regranting the use privilege to ordinary users. This made the discussion of garbage collection (the dropping of editioned objects in retired editions that were not the source of inherited representation in non-retired editions (hereinafter covered objects)) difficult. There was no intrinsic support, so users had to implement a mechanical scheme to discover the covered status, using relatively complex dictionary query, and then to drop each of them. And because of the shine-through effect, the present root edition could be dropped (assuming no covered objects remained in any non-retired edition) only when it contained no editioned objects.

Oracle Database 12*c* Release 2 changes this by allowing the root edition to be dropped without considering its population of objects. This is done after reasoning in the domain of the semantic model, and it is the clear and irrevocable way to assert that the root edition is retired. The `DROP EDITION` command completes instantaneously and thereafter not even SYS can use it. The user understands this in terms of the mental model of the implementation. The only immediate effect of `DROP EDITION` is to mark it unusable. However, background processes are still able to access all dropped editions to discover, and to drop covered objects. If, by virtue of the history of successive edition-based redefinition exercises, and the dropping of the covered objects that these exercises cause, the root edition ends up with no objects at all, then the background process removes it completely from the database.

Users are relieved of a noticeable burden of comprehension and programming. A database automatically achieves the state that, no matter how many edition-based redefinition exercises are done, the total number of editioned objects is identical to what it would be if the database were flattened to just a single edition.

Related Topics

- *Oracle® Database Development Guide*

Online Conversion of a Nonpartitioned Table to a Partitioned Table

Nonpartitioned tables can be converted to partitioned tables online. Indexes are maintained as part of this operation and can be partitioned as well. The conversion has no impact on the ongoing DML operations.

The online conversion of a nonpartitioned table to a partitioned table enables any application to adopt partitioning without application downtime. Customers can adopt partitioning for any system and evolve tables as needed to benefit from the partitioning for large tables.

Related Topics

- *Oracle® Database VLDB and Partitioning Guide*

Online SPLIT Partition and Subpartition

The partition maintenance operations `SPLIT PARTITION` and `SPLIT SUBPARTITION` can now be executed as online operations for heap organized tables, allowing the concurrent DML operations with the ongoing partition maintenance operation.

Allowing any partition maintenance operation to become truly online enables the customers to schedule and execute all of these operations as needed, without having to plan around periods of query-only windows. This functionality both increases application availability and simplifies application development.

Related Topics

- *Oracle® Database VLDB and Partitioning Guide*

Online Table Move

Nonpartitioned tables can be moved as an online operation without blocking any concurrent DML operations. A table move operation now also supports automatic index maintenance as part of the move.

Data maintenance on nonpartitioned tables does not require any maintenance window since it does not impact any DML or query operation.

Related Topics

- *Oracle® Database Administrator's Guide*

Partitioning: Table Creation for Partition Exchange

A new DDL command allows the creation of a table that exactly matches the shape of a partitioned table and is therefore eligible for a partition or subpartition exchange of a partitioned table. Note that indexes are not created as part of this command.

Creating a table that qualifies for a partition or subpartition exchange can be a tedious task for older tables that have had various structural changes and reorganizations. With this new DDL, the task becomes very simple and straightforward to implement. It also adds some implicit business context to such an operation, as compared to a `CREATE TABLE AS SELECT` command.

Related Topics

- *Oracle® Database VLDB and Partitioning Guide*

Partitioning: Filtered Partition Maintenance Operations

Partition maintenance operations can now be combined with data filtering. For example, a partition can be compressed and moved to a different tablespace, but only the data satisfying the specific filter criteria is actually moved.

Partition maintenance operations with data filtering combine two of the most common data maintenance operations. This combination not only makes the partition maintenance operation more flexible and powerful, but also makes it more performant and less resource intensive compared to the two separate data management operations.

Related Topics

- *Oracle® Database VLDB and Partitioning Guide*

Recovery Server and RMAN Improvements

- RMAN: Syntax Enhancements
- SCAN Listener Supports HTTP Protocol
- Oracle Recovery Manager - Enhanced Table Recoveries Across Schemas Using REMAP SCHEMA
- Disk Space Check During RECOVER TABLE Operation
- Upgrading the Incremental Transportable Scripts

- Cross-Platform Import of a Pluggable Database into a Multitenant Container Database
- Cross-Platform Migration Support for Encrypted Tablespaces
- Cross-Platform Support Over The Network
- Data Guard DUPLICATE Command Enhancements
- DUPLICATE Command Support for Non Auto-Login Wallet Based Encrypted Backups

RMAN: Syntax Enhancements

You can use the enhanced `SET NEWNAME` command for the entire tablespace or database instead of on a per file basis. The new `MOVE` command enables easier movement of files to other locations instead of using backup and delete. The new `RESTORE+RECOVER` command for data file, tablespace and database performs restore and recovery in one step versus having to issue offline, restore, recover, and online operations.

These enhancements simplify and improve ease-of-use for the `SET NEWNAME`, `MOVE` and `RESTORE+RECOVER` commands.

Related Topics

- *Oracle® Database Backup and Recovery Reference*

SCAN Listener Supports HTTP Protocol

SCAN listener now enables connections for the recovery server coming over HTTP to be redirected to different machines based on the load on the recovery server machines.

This feature enables connections to be load balanced across different recovery server machines.

Related Topics

- *Oracle® Real Application Clusters Installation Guide for Linux and UNIX*

Oracle Recovery Manager - Enhanced Table Recoveries Across Schemas Using REMAP SCHEMA

The Oracle Recovery Manager (RMAN) table recovery feature offers `REMAP TABLE` and `REMAP TABLESPACE` options. The `REMAP TABLE` option is followed by a list of logical objects to be remapped with their new names. It can be used to rename entire tables or to rename table partitions. The `REMAP TABLESPACE` option, similarly, remaps all of the specified objects in the existing tablespace into a new one. These options, however, do not offer the flexibility of cross-schema table recovery. This feature adds the `REMAP SCHEMA` option to the `RECOVER TABLE` command, which is then passed to the Oracle Data Pump import operation. This feature allows a list of tables and table partitions to be recovered to an alternative schema other than the original schema of the table.

This feature adds more flexibility to import or recover tables across different schemas, which provides more freedom to users and may also provide a better understanding of table-related indexes, triggers, constraints, and so on, if these object names are already existing under the same schema.

Related Topics

- *Oracle® Database Backup and Recovery User's Guide*

Disk Space Check During RECOVER TABLE Operation

The Oracle Recovery Manager (RMAN) table recovery feature implicitly creates auxiliary instance, restores, and recovers tablespaces or data files to perform the table recovery option. However, if there is not enough space to create this instance, an operating system level error is returned. This enhancement provides an up front check on the available disk space for the auxiliary instance, before RMAN executes table recovery.

`RECOVER TABLE` is a database operation that prevents operating system level error messages due to disk space issues. When this feature is implemented, the user is notified upfront if there is not enough space to perform the operation, and the `RECOVER TABLE` operation is aborted.

Related Topics

- *Oracle® Database Backup and Recovery User's Guide*

Upgrading the Incremental Transportable Scripts

In Oracle Database 12*c* Release 1 (12.1), Oracle Recovery Manager introduced cross-platform transport capability. This feature addresses the seamless migration process using scripts that perform tasks for the entire process. It includes:

- Preparation of source data
- Roll forward phase for the destination data state to catch up with the source data
- Transport phase

Customers have a migration path to Oracle Exadata using the My Oracle Support (MOS) Note 1389592.1. However, there are plenty of manual steps that are prone to errors. This feature helps in the seamless migrations of Oracle Database 12*c* to Oracle Exadata, thus expanding the Oracle Exadata adoption.

Cross-Platform Import of a Pluggable Database into a Multitenant Container Database

The concept of pluggable databases (PDBs) was introduced with Oracle Database 12*c* (multiple databases sharing a single database instance). This feature is the solution for multitenancy on Oracle Database. This feature addresses:

- Plugging in a remote PDB through cross-platform transportable tablespace within this release.
- Plugging in a remote non-multitenant container database (CDB) through cross-platform transportable tablespace (converting a non-CDB into a PDB with plans to support versions earlier than Oracle Database 12*c*).

This feature increases customer adoption of consolidating their databases deployed with different architecture into the multitenant database architecture.

Related Topics

- *Oracle® Database Backup and Recovery User's Guide*

Cross-Platform Migration Support for Encrypted Tablespaces

The concept of pluggable databases (PDBs) was introduced in Oracle Database 12*c* Release 1 (multiple databases sharing a single database instance). This feature addresses support for encrypted tablespaces which are also to be migrated using cross-platform transport.

This feature increases customer adoption of consolidating their databases which have encrypted tablespaces with the multitenant architecture.

Related Topics

- *Oracle® Database Backup and Recovery User's Guide*

Cross-Platform Support Over The Network

This feature addresses three capabilities of Oracle Recovery Manager (RMAN) in a cross-platform migration:

- RMAN support for Data Recovery Advisor on a standby database. Basically, when a data file is lost on a primary or standby database, RMAN generates repair scripts to fetch the file from the primary or standby database.
- Support for active duplicate scripts with cross-platform transport. The active duplicate defaults to backup set method only when the number of auxiliary channels is equal to or greater than the target channels and there is a TNS alias for the target database. This is required so that we do not break the existing active duplicate scripts.
- Active restore needs the added capability to support backup and restore cross-endian. However, this is limited to the tablespaces and not the entire database. This needs an additional `FROM SERVICE` syntax in the `RESTORE FROM PLATFORM` <*`platform`*> command such that it starts off an active restore session.

Transporting the data across platforms helps in migration, ease of use, and more adoption of Oracle databases.

Related Topics

- *Oracle® Database Backup and Recovery User's Guide*

Data Guard DUPLICATE Command Enhancements

Far Sync Standby Support

Starting with Oracle Database 12*c* Release 1 (12.1), a new variant of a standby database was introduced called far sync standby. The goal of this enhancement is to allow users to create this type of standby database using the `DUPLICATE` command.

Normal Duplicate Database From Physical Standby Database

Currently, it is not possible to create a copy of a database when the target database is a physical standby database. Only a physical standby database can be created when connected to a physical standby database as the target database. The goal here is to remove this and the other possible restrictions, and allow the same set of operations when the target database is a primary or a physical standby database. This enhancement leverages the existing physical standby database for more uses by taking load off the primary database.

Oracle Data Guard is a high availability (HA) or Maximum Availability Architecture (MAA) feature critical for disaster recovery deployments. By enabling easier and more efficient methods to enhance the `DUPLICATE` command to create far sync standby support and by allowing creation of a regular database, this feature increases the benefits for deploying Oracle Data Guard by offloading processes to the standby database.

Related Topics

- *Oracle® Data Guard Concepts and Administration*

DUPLICATE Command Support for Non Auto-Login Wallet Based Encrypted Backups

Currently, it is not possible to run the `DUPLICATE` command using encrypted backups where the backup keys are stored in a password-based Oracle Wallet or a key store. Even if a user opens the wallet by starting an auxiliary instance, the `DUPLICATE` command restarts the auxiliary instance many times during the command execution. Hence, the restores from the encrypted backups after the instance bounce fail because of the inability to decrypt the backups (because Oracle Wallet or the key store is not opened). This restriction is removed with this feature.

This feature enables complete support for encrypted backups with encrypted key storage being auto-login wallet, non-auto-login wallet, or user-specified passwords.

Related Topics

- *Oracle® Database Backup and Recovery User's Guide*

Sharding

- Oracle Database Sharding
- Data-Dependent Routing
- Automatic Deployment of Oracle Data Guard
- Creating Data Guard Standbys With Database Configuration Assistant
- Statement-Level Routing and Cross-Shard Queries
- Enhancing SSL and Kerberos Authentication for Administrative Users

Oracle Database Sharding

Sharding with Oracle Database 12*c* Release 2 (12.2) is an architecture for suitable online transaction processing (OLTP) applications where data is horizontally partitioned across multiple discrete Oracle databases, called shards, which share no hardware or software. The collection of shards is presented to an application as a single logical Oracle database.

Oracle sharding supports automated deployment, high performance routing, and complete life-cycle management. High availability for individual shards is enabled by automatic deployment of either Oracle Data Guard or Oracle GoldenGate replication, at the discretion of the administrator. Each shard is an Oracle Database that has the same capabilities, with very few exceptions, as an Oracle Database in a non-sharded deployment.

Oracle sharding is intended for custom OLTP applications that are explicitly designed for a sharded database architecture. Unlike an architecture based on Oracle Real Application Clusters (Oracle RAC), applications that use sharding must have a well-defined data model and data distribution strategy (consistent hash, range, list, or composite) that primarily accesses data using a sharding key. Examples of keys include `customer_id`, `account_no`, `country_id`, and so on. Oracle sharding also supports data placement policies (rack and geo awareness) and all deployment models (for example, on-premises and public or hybrid clouds).

Sharding with Oracle Database 12*c* Release 2 (12.2) provides a number of benefits:

1. Linear scalability with complete fault isolation. OLTP applications designed for Oracle sharding can elastically scale (data, transactions and users) to any level, on any platform, simply by deploying new shards on additional stand-alone servers. The unavailability of a shard due to either an unplanned outage or planned maintenance affects only the users of that shard; it does not affect the availability or performance of the application for users of other shards. Each shard can run a different release of the Oracle Database as long as the application is backward compatible with the oldest running version making it simple to maintain availability of an application while performing database maintenance.
2. Simplicity using automation of many lifecycle management tasks including system-managed partitioning, single-command deployment, and fine-grained rebalancing.
3. Superior runtime performance using intelligent, data-dependent routing.
4. Enterprise quality. Each shard is an Oracle Database rendering strict consistency, the full power of SQL, developer agility with JSON, and proven enterprise qualities for security, availability, backup and recovery, and lifecycle management.

Related Topics

- *Oracle® Database Administrator's Guide*

Data-Dependent Routing

Java Database Connectivity (JDBC), Oracle Call Interface (OCI), and Oracle Data Provider for .NET (ODP.NET) clients are able to recognize shard keys specified in a connection string. A shard topology cache in the connection layer is used to route the request directly to the shard where the data resides.

The Oracle Universal Connection Pool (UCP) for JDBC client is also able to recognize shard keys specified in the connection URL. A shard topology cache is used to route the connection directly to the shard where the data resides. Oracle UCP also enables non-Oracle application clients such as Apache Tomcat, WebSphere, and so on, to work with Oracle sharding.

Data-dependent routing provides high performance data access and availability for applications designed for a sharded database architecture.

Automatic Deployment of Oracle Data Guard

Deployment is automatic for Oracle Data Guard physical replication between shards with Oracle Data Guard fast-start failover (automatic database failover). This is the default Oracle High Availability (HA) configuration for individual shards.

Business benefits include the following:

1. Automated deployment reduces administrative overhead.
2. Physical database replication provides the strongest data protection using real time Oracle data validation.
3. Supports either asynchronous or synchronous replication for near-zero or zero data loss protection.
4. Automatic database failover provides high availability for server, database, network, and site outages.
5. Read-only workloads may be load-balanced across all copies of shards used for HA for additional scalability when using Oracle Active Data Guard.

Creating Data Guard Standbys With Database Configuration Assistant

Database Configuration Assistant (DBCA) is enhanced to support creation of an Oracle Data Guard standby database from an existing primary database using the DBCA command-line interface. This ability reduces the manual steps you must execute to create a standby database outside of Oracle Enterprise Manager. In addition, DBCA allows custom scripts to be run at the end of the standby database creation.

This feature enables users to script the creation of standby databases in a very simple manner from a command-line interface.

Related Topics

- *Oracle® Data Guard Concepts and Administration*

Statement-Level Routing and Cross-Shard Queries

This feature enables routing and processing of queries and transactions that access data that is stored on multiple shards. This feature also enables routing for queries that do not specify a shard key.

This feature enables simple aggregation of data and reporting across shards. It also enables the flexibility for any database application to execute SQL statements (including `SELECT` and DML) in systems where tables are sharded or replicated. For this feature, you do not need to specify the shards on which the query runs. In both use cases, you accept a reduced level of performance compared to OLTP transactions that specify the sharding key.

Related Topics

- *Oracle® Database Administrator's Guide*

Enhancing SSL and Kerberos Authentication for Administrative Users

This feature enables administrative privileged users, such as `SYSDBA`, `SYSDG`, `SYSKM`, `SYSBACKUP`, to be authenticated to Oracle Database using SSL and Kerberos authentication schemes. Administrative users can connect to Oracle Database using SSL and Kerberos schemes, even when an instance is not yet started.

Administrative privileged users, such as `SYSDBA`, `SYSDG`, `SYSKM`, `SYSBACKUP`, can connect to Oracle Database, even when a database instance is not started or if the database is not open. This is done using SSL and Kerberos authentication schemes.

Related Topics

- *Oracle® Database Security Guide*

Simplifying Upgrades

- Oracle Label Security Support for Oracle Data Guard Database Rolling Upgrade
- Oracle Database Vault Support for Oracle Data Guard Database Rolling Upgrade
- Oracle Database Vault Support for Flashback Technology and ILM
- Oracle Data Guard Broker Support for Oracle Active Data Guard Rolling Upgrades

Oracle Label Security Support for Oracle Data Guard Database Rolling Upgrade

Databases that use Oracle Label Security (OLS) can be upgraded to new Oracle Database releases and patchsets using Oracle Data Guard database rolling upgrades (transient logical standby database only). This support is for rolling upgrades from Oracle Database 12*c* Release 2 (12.2) onward.

Oracle Data Guard database rolling upgrades reduce planned downtime by enabling the upgrade to new database releases or patchsets in rolling fashion. Total database downtime for a rolling upgrade is limited to the small amount of time required to execute an Oracle Data Guard switchover.

Related Topics

- *Oracle® Data Guard Concepts and Administration*

Oracle Database Vault Support for Oracle Data Guard Database Rolling Upgrade

Databases that use Oracle Database Vault can be upgraded to new Oracle Database releases and patchsets using Oracle Data Guard database rolling upgrades (transient logical standby database only). This support is for rolling upgrades from Oracle Database 12*c* Release 2 (12.2) onward.

Oracle Data Guard database rolling upgrades reduce planned downtime by enabling the upgrade to new database releases or patchsets in rolling fashion. Total database downtime for a rolling upgrade is limited to the small amount of time required to execute an Oracle Data Guard switchover.

Related Topics

- *Oracle® Data Guard Concepts and Administration*

Oracle Database Vault Support for Flashback Technology and ILM

A number of Oracle Database Vault command rules are added to support flashback operations such as `PURGE TABLE`, `PURGE TABLESPACE`, `PURGE RECYCLEBIN`, and `PURGE DBA_RECYCLEBIN`. This also extends Oracle Database Vault support to the database recycle bin feature. In addition, an Oracle Database Vault authorization for Information Lifecycle Management (ILM) users is added.

This support enables Oracle Database Vault customers to take advantage of the flashback and ILM technologies features without compromising the security protections that Oracle Database Vault provides.

Related Topics

- *Oracle® Database Vault Administrator's Guide*

Oracle Data Guard Broker Support for Oracle Active Data Guard Rolling Upgrades

Oracle Data Guard broker now supports Oracle Active Data Guard rolling upgrade. Oracle Active Data Guard rolling upgrade was introduced in Oracle Database 12.1. It simplifies the execution of the transient logical database rolling upgrade process by automating many manual steps in a simple PL/SQL package (`DBMS_ROLLING`). In addition to making database rolling upgrades simpler, the automated process is much more reliable. Oracle Data Guard broker can now direct Oracle Active Data Guard rolling upgrades from the DGMGRL command-line interface. Broker support also adds substantial simplification to the rolling upgrade process by transparently handling redo transport destination settings and other tasks.

Broker-managed database rolling upgrades provide the ultimate method of simplifying the upgrade process and minimizing downtime and risk when introducing change to production environments.

Related Topics

- *Oracle® Data Guard Concepts and Administration*

Big Data and Data Warehousing

- Big Data Management System Infrastructure
- Dimensional In-Database Analysis
- Enhancing Query Processing and Optimization
- General
- High Performance Oracle Advanced Analytics, Data Mining, and Predictive Analytics
- Semi-Structured Big Data

Big Data Management System Infrastructure

- Partitioning: External Tables
- Enhancing Declarative Constraint Support
- Oracle Parallel Query Services on Oracle RAC Read-Only Nodes
- External Tables Can Access Data Stored in Hadoop Data Sources Including HDFS and Hive

Partitioning: External Tables

Partitioned external tables provide both the functionality to map partitioned Hive tables into the Oracle Database ecosystem as well as providing declarative partitioning on top of any Hadoop Distributed File System (HDFS) based data store.

Exposing the power of Oracle Database partitioning for external HDFS-based data stores enables orders of magnitude faster query performance and enhanced data maintenance.

Related Topics

- *Oracle® Database Administrator's Guide*

Enhancing Declarative Constraint Support

Constraints defined as declarative and non-enforced on both internal and external tables are used more comprehensively for query processing optimizations. Additional improvements include declarative `NOT NULL` constraints for both internal and external tables and the support of all declarative constraints for external tables (uniqueness, primary key, and primary key-foreign key relationship).

Allowing declarative constraint definitions on external data helps increase query performance while optimizing resource consumption for any complex SQL operation, irrespective of whether the data resides inside the database or not.

Related Topics

- *Oracle® Database SQL Language Reference*

Oracle Parallel Query Services on Oracle RAC Read-Only Nodes

Oracle parallel query services on Oracle RAC read-only nodes represents a scalable parallel data processing architecture. The architecture allows for the distribution of a high number of processing engines dedicated to parallel execution of queries.

Oracle parallel processing farm allows users to deploy a scalable processing architecture on large clustered systems specifically for parallel query operations.

Related Topics

- *Oracle® Database VLDB and Partitioning Guide*

External Tables Can Access Data Stored in Hadoop Data Sources Including HDFS and Hive

External tables can access data stored on Hadoop. The most important data sources are the Hadoop Distributed File System (HDFS) and Apache Hive. External tables are used by both SQL*Loader and Oracle Data Pump, and thus also by the `ORACLE_LOADER` and `ORACLE_DATAPUMP` access drivers.

This feature allows more users of Hadoop to combine map-reduce processing with the essential database qualities that many applications require.

Related Topics

- *Oracle® Database Utilities*

Dimensional In-Database Analysis

- Analytic Views

Analytic Views

Analytic views provide a business intelligence layer over a star schema, making it easy to extend the data set with hierarchies, levels, aggregate data, and calculated measures. The analytic view feature includes the new DDL statements `CREATE ATTRIBUTE DIMENSION`, `CREATE HIERARCHY` and `CREATE ANALYTIC VIEW` and their related `ALTER` and `DROP` statements, new calculated measure expression syntax, and new data dictionary views.

Analytic views allow data warehouse and business intelligence application developers to extend the star schema with time series and other calculations, making data more valuable to business users and eliminating the need to define calculations within the application.

Analytic views can be queried with simple SQL queries, simplifying application development by eliminating the need for complex SQL generators. Calculations can be defined in the analytic view and can be selected by including the measure name in the SQL select list.

Analytic views promote consistency across applications. By defining aggregation and calculation rules centrally in the database, the risk of inconsistent results in different reporting tools is reduced or eliminated.

Related Topics

- *Oracle® Database Data Warehousing Guide*

Enhancing Query Processing and Optimization

- Optimizer Statistics Advisor
- Enhancing SQL Plan Management
- Scan Rate and In-Memory Columnar Statistics
- Band Join Enhancements
- Parallel Recursive WITH Enhancements
- Cursor-Duration Temporary Tables Cached in Memory

Optimizer Statistics Advisor

The database provides a built-in mechanism to track and analyze how statistics are collected. Based on the tracked information and the current best practices, the database reports problems with the current statistics collection and suggests changes to the statistics collection.

Enabling the database to collect historical information about the statistics collection and to suggest changes enables a customer to fine tune the statistics collection and to adopt best practices for optimizer statistics collection.

Related Topics

- *Oracle® Database SQL Tuning Guide*

Enhancing SQL Plan Management

SQL Plan Management (SPM) leverages a larger pool of plan information (including Automatic Workload Repository (AWR)) as a source for SQL plan baselines. SPM is also enhanced to provide a more selective plan capturing and easier verification for customers.

Customers can leverage SQL Plan Management for their specific application requirements.

Related Topics

- *Oracle® Database SQL Tuning Guide*

Scan Rate and In-Memory Columnar Statistics

`DBMS_STATS` now supports external table scan rates and In-Memory Column Store (IM column store) statistics.

If the database uses the In-Memory Column Store, you can set `im_imcu_count` to the number of In-Memory Compression Units (IMCU) in the table or partition, and set `im_block_count` to the number of blocks in the table or partition. For an external table, scan rate specifies the rate at which data is scanned in MB/second.

Related Topics

- *Oracle® Database SQL Tuning Guide*

Band Join Enhancements

A band join is a special type of nonequijoin in which key values in one data set must fall within the specified range (“band”) of the second data set. When the database detects a band join, the database evaluates the costs of band joins more efficiently, avoiding unnecessary scans of rows that fall outside the defined bands.

In most cases, optimized performance is comparable to an equijoin.

Related Topics

- *Oracle® Database SQL Tuning Guide*

Parallel Recursive WITH Enhancements

Oracle Database supports recursive queries through the use of a proprietary `CONNECT BY` clause, and an ANSI compliant resursive `WITH` clause. The parallel recursive `WITH` clause enables this type of query to run in parallel mode.

Recursive `WITH` type queries are typical with graph data found in social graphs, such as Twitter graphs or call records. Recursive `WITH` type queries are also commonly used in transportation networks (for example, for flight paths, roadways, and so on). Graph data is inherently cyclic in nature and thus, recursive queries are a natural candidate for the analytical processing of graph data. Recursive `WITH` ensures the efficient computation of the shortest path from a single source node to single or multiple destination nodes in a graph. Bi-directional searching is used to ensure the efficient computation of the shortest path from a single source node to single or multiple destination nodes in a graph. A bi-directional search starts from both source and destination nodes, and then advancing the search in both directions. Techniques such as greedy search, pruning of unpromising intermediate paths, and bi-directional searches are implemented to optimize performance.

Related Topics

- *Oracle® Database SQL Language Reference*

Cursor-Duration Temporary Tables Cached in Memory

Complex queries often process the same SQL fragment (query block) multiple times to answer a business question. The results of these queries can be stored internally to avoid the multiple processing of the same query fragment, implemented as cursor-duration temporary tables. With this new functionality, these temporary tables can reside completely in cache without the need to write them to disk.

Caching intermediate, partial results in memory increases the performance of complex, mission-critical operations and optimizes I/O resource consumption.

General

- Local TEMP Tablespaces
- Oracle Database Can Contain Both Read/Write and Read-Only Instances

Local TEMP Tablespaces

Individual leaf nodes in a hub-leaf cluster environment can spill to local `TEMP` tablespaces.

This feature reduces the amount of data to be transferred to the hub nodes and the common infrastructure, enabling better scalability for large number of nodes.

Related Topics

- *Oracle® Real Application Clusters Administration and Deployment Guide*

Oracle Database Can Contain Both Read/Write and Read-Only Instances

Oracle Database 12*c* Release 2 (12.2) provides two types of instances within the same database: read/write and read-only. The read/write instances are regular Oracle Database instances and can process updates to the data (for example, DML statements `UPDATE, DELETE, INSERT`, and `MERGE`), partition maintenance operations, and so on. You can directly connect to read/write instances.

The read-only instances can process only queries and cannot directly update data. You cannot directly connect to read-only instances. Note that there are parallel SQL statements that contain both updates and querying of the data (for example, `INSERT INTO <select query>`). In this case the `<select query>` part of the statement is processed on both read/write and read-only instances while the INSERT part is processed only on read/write instances.

To designate an instance as read-only, set the `INSTANCE_MODE` parameter to `READ_ONLY`. (The default value of the parameter is `READ_WRITE`.)

The introduction of read-only instances significantly improves scalability of parallel queries for data warehousing workloads and allows Oracle Database to run on hundreds of physical nodes.

Related Topics

- *Oracle® Clusterware Administration and Deployment Guide*

High Performance Oracle Advanced Analytics, Data Mining, and Predictive Analytics

- Oracle Advanced Analytics Extensibility for R Models
- Oracle Advanced Analytics Association Rules Enhancements
- Significant Performance Improvements and Support for Partitioned Models
- Explicit Semantic Analysis Algorithm Enhancement

Oracle Advanced Analytics Extensibility for R Models

The open source Oracle R Enterprise statistical programming language and environment is widely popular with over 7,000 open source R packages available in the Comprehensive R Archive Network (CRAN) repository. In this release, R developers can run R scripts and model based algorithms within the Oracle Advanced Analytics algorithm architecture for in-database model build, model apply, model parameter settings, and model viewing. Data with nested attributes (for example, text) and aggregated transactional data can also be handled.

The primary benefit is to extend the ease of advanced analytics development from the R domain to Oracle Database. Customers require an environment in which they can roll out new analytics and take advantage of existing R packages swiftly in crucial areas such as fraud detection where businesses need to adapt constantly to an ever changing criminal landscape.

Related Topics

- *Oracle® Data Mining Concepts*

Oracle Advanced Analytics Association Rules Enhancements

There are numerous enhancements to the Oracle Advanced Analytics Association Rules algorithm, including the calculation of values associated with rules, such as sales amount.

It is valuable to collect supplementary information for Association Rules to help identify the value of each rule to the business. This feature enables Oracle Advanced Analytics to perform such computation while building the model, which is simpler and more efficient than requiring you to perform this calculation as a post-processing step.

Related Topics

- *Oracle® Data Mining User's Guide*

Significant Performance Improvements and Support for Partitioned Models

Oracle Advanced Analytics algorithms were improved to maximally exploit in-database parallel execution for both model building and model application, provide improved extensibility, and take advantage of new computing techniques.

Models can take a partitioning key from the user, which results in building separate models that are tailored to each partition. These models are still bundled as a single model to simplify management and deployment as well as to improve performance.

This feature provides significant performance improvements involving parallelism and optimization enabling all algorithms to scale to data volumes found in big data and cloud use cases.

Data scientists often have knowledge about important attributes that lead to different data distributions, such as regional differences. This feature enables Oracle Advanced Analytics to take that knowledge as input and tailor the models to the different segments. Without this feature, data scientists would need to build and manage hundreds or thousands of models to achieve this functionality, resulting in significant usability and performance issues.

Related Topics

- *Oracle® Data Mining User's Guide*

Explicit Semantic Analysis Algorithm Enhancement

Explicit Semantic Analysis (ESA) is a useful technique for extracting meaningful, interpretable features. You can use ESA for document similarity and topic identification.

As compared to latent techniques, ESA enables immediate user understanding of the features. Use of Wikipedia provides a large collection of existing documents to provide sensible features and topics.

Related Topics

- *Oracle® Data Mining User's Guide*

Semi-Structured Big Data

Support for Range-Based XML Queries with Text-Based XML Index

This feature enables an XMLIndex based on Oracle Database full text indexing technology to support range-based searches for numeric and date values.

This feature extends the scope of the text-based XMLIndex by enabling the index to optimize range-based searching. This increases the range of queries that can be optimized with the text-based index.

Improving Token Management for Binary XML

This feature improves the efficiency of client-based XML encoding and decoding by enabling token tables to be managed at a lower level of granularity than the database. This makes operations such as transportable tablespace and exchange partition more efficient.

This feature improves the efficiency of client-side encoding and decoding of binary XML by reducing the changes of the tokens that need to be reloaded during encoding and decoding operations. This feature also enables exchange partition operations to be performed using transportable tablespaces without requiring that the XML being moved be recoded.

Related Topics

- *Oracle® XML DB Developer's Guide*

Partitioning: Improving Support for XMLIndex

This feature improves the scalability of database operations involving XMLType by completing the work required during parallel operations, including parallel query (PQ), parallel DML operations (PDML), and tighter integration with Exadata storage services. This feature also enables parallel loading of XML Data with partitioning and XMLIndex support for inter- and intra-partition parallelism.

This feature enables large volumes of XML to be ingested, queried, and retrieved by making full use of the power of parallel hardware configurations such as Oracle Exadata.

Related Topics

- *Oracle® XML DB Developer's Guide*

SMALL_R_ROW Support

A new Oracle Text storage attribute, `SMALL_R_ROW`, enables you to reduce the size of a row in the `$R` index table.

This feature reduces contention on the `$R` table when many processes are updating rows at the same time.

Related Topics

- *Oracle® Text Application Developer's Guide*

Text: Reverse Token Index for Left-Truncated Queries

A new `WORDLIST` attribute, `REVERSE_INDEX`, creates a new reversed index on the `TOKEN_TEXT` column of the `$I` token table.

The left-truncated searches are now as fast as the right-truncated searches, without the need to use the `WORDLIST` attribute, `SUBSTRING_INDEX`.

Related Topics

- *Oracle® Text Reference*

Support for 30-Character Index Names

An Oracle Text index name now has an increased maximum limit of 30 characters.

The ability to use longer index names results in less restrictive rules for application developers.

Related Topics

- *Oracle® Text Reference*

Compression and Archiving

- DBFS, Oracle Exadata and SecureFiles Enhancements
- Hybrid Columnar Compression Improvements
- Index Compression Enhancements

DBFS, Oracle Exadata and SecureFiles Enhancements

- NFS Server in Database

NFS Server in Database

An Oracle database can now serve as a network file system (NFS) server. The database now responds to NFS requests from any NFS client and stores both the files and their metadata within the database.

Files associated with a primary database (for example, SQL scripts) can be automatically replicated on a standby database.

Related Topics

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

Hybrid Columnar Compression Improvements

- HCC Compression for Array Inserts
- HCC Compress Data With ADO Row-Level Policy

HCC Compression for Array Inserts

Hybrid Columnar Compression (HCC) can now be used during array inserts into tables. This means that the SQL `INSERT SELECT` statement without the `APPEND` hint can use HCC, and array inserts from programmatic interfaces such as PL/SQL and the Oracle Call Interface (OCI) can use HCC.

This feature makes it easier to gain the benefits of HCC by enabling HCC when data is inserted without using direct path operations. In addition, the impact on concurrency is minimal, because direct path operations require a segment lock, whereas array inserts do not use a segment lock.

Related Topics

- *Oracle® Database Administrator's Guide*

HCC Compress Data With ADO Row-Level Policy

Automatic Data Optimization (ADO) now supports Hybrid Columnar Compression (HCC) for row-level policies. Rows from cold blocks can be HCC-compressed even if there is DML activity on other parts of the table or partition.

ADO policies for HCC are now effective even if changes are constantly being made to a small subset of the table or partition. This means that HCC space saving and performance benefits can be transparently extended to more data.

Related Topics

- *Oracle® Database VLDB and Partitioning Guide*

Index Compression Enhancements

- Advanced Index Compression

Advanced Index Compression

Prior to this release, the only form of advanced index compression was low compression. Now you can also specify high compression. High compression provides even more space savings than low compression.

Indexes consume a large amount of storage within many databases. The high level of advanced index compression provides significant space savings while also improving performance for queries that are executed using indexes. High compression offers the following advantages over low compression:

- Gives higher compression ratios in most cases.
- Employs more complex compression algorithms than advanced low compression.
- Stores data in a compression unit, which is a special on-disk format.

Related Topics

- *Oracle® Database Administrator's Guide*

Database Lifecycle Management

- Support for Automation of Large Scale Cloud Deployments

Support for Automation of Large Scale Cloud Deployments

- Oracle Grid Infrastructure Installation Using Zip Images

Oracle Grid Infrastructure Installation Using Zip Images

In Oracle Grid Infrastructure 12*c* Release 2 (12.2), the installation media is replaced with a zip file for the Oracle Grid Infrastructure installer. The user can launch the installer wizard after unzipping the zip file into the target Grid home path.

This feature simplifies the installation experience for the user by directly delivering the Oracle home software image to the users in a zip format. Until now, users had to download the installation media from Automated Release Update (ARU) or Oracle Technology Network (OTN), unzip the files, and then run the installer, either in the setup.exe or runInstaller format, to copy the software to their systems.

Related Topics

- *Oracle® Grid Infrastructure Installation and Upgrade Guide for Linux*

Database Overall

- Core Database Improvements
- General
- Pluggable Databases Ease-of-Adoption
- Pluggable Databases Multitenancy
- Pluggable Databases Overall
- Upgrades
- Utilities

Core Database Improvements

- Text: Partition-Specific Near Real-Time Indexes
- Text: Improving Management of Near Real-Time Indexes
- Partitioning: Auto-List Partitioning
- Fine-Grained Cursor Invalidation
- Text: Adding New Document Formats
- Text: Structured/Sort Data (SDATA) Section Improvements
- Text: Availability of Updated Documents in Index
- Text: Read-Only MDATA Sections
- Text: Sentiment Analysis and Collocates
- Text: Join Character Support for JAPANESE_VGRAM_LEXER and WORLD_LEXER
- Text: Extracting Synonyms of Words in Documents
- Text: Changes to NDATA, NEAR2, and NESTED NEAR Query Operators
- Materialized Views: Refresh Statistics History
- Process Management

- Partitioning: Read-Only Partitions
- Partitioning: Multi-Column List Partitioning

Text: Partition-Specific Near Real-Time Indexes

The option to use near real-time indexes (the `STAGE_ITAB` setting) can now be specified at a partition level.

Partitions, which are frequently updated, can have near real-time indexing switched on, while less frequently updated tables can have it turned off.

Related Topics

- *Oracle® Text Reference*

Text: Improving Management of Near Real-Time Indexes

A maximum size for the near real-time `$G` index table can now be specified. After the size is filled, further updates are written to the main `$I` table instead.

Specifying a maximum size prevents the near real-time `$G` index table from getting too large to fit into memory.

Related Topics

- *Oracle® Text Reference*

Partitioning: Auto-List Partitioning

The database automatically creates a separate (new) partition for every distinct partition key value of the table.

Auto-list partitioning removes the management burden from the DBAs to manually maintain a list of partitioned tables for a large number of distinct key values that require individual partitions. It also automatically copes with the unplanned partition key values without the need of a `DEFAULT` partition.

Related Topics

- *Oracle® Database VLDB and Partitioning Guide*

Fine-Grained Cursor Invalidation

Cursor invalidations are more fine-grained, reducing the need for cursor invalidation and recompilation. It also defers any cursor recompilation, if possible, to avoid spikes in hard parses.

With a more fine-grained cursor invalidation and a slow recompilation, applications become more available and do not incur any sudden increase in the CPU utilization for cursor recompilation.

Related Topics

- *Oracle® Database SQL Tuning Guide*

Text: Adding New Document Formats

In this release, formats supported by Oracle Outside-In Technology can be indexed using `AUTO_FILTER`.

Newer formats can also be indexed.

Related Topics

- *Oracle® Text Reference*

Text: Structured/Sort Data (SDATA) Section Improvements

The following improvements are made for `SDATA` sections:

- B-tree backed `SDATA` sections are made range-searchable for better performance.
- `SDATA` can be optimized for `SEARCH` or `SORT` operations
- Multi-value `SDATA` sections - Group counting available on `SDATA` sections in the result set interface.

These `SDATA` improvements provide better performance and more flexible mixed queries within the Oracle Text index environment.

Related Topics

- *Oracle® Text Reference*

Text: Availability of Updated Documents in Index

In the earlier releases, when documents were updated, they could not be searched on until the index was synchronized. Starting Oracle Database 12*c* Release 2 (12.2), you have the option of keeping the old (possibly, outdated) index entries, so that the document can still be found from its original contents.

Documents are still searchable after minor updates, without having to immediately perform an index synchronization.

Related Topics

- *Oracle® Text Application Developer's Guide*

Text: Read-Only MDATA Sections

Normal `MDATA` sections can be updated without reindexing the entire document, but there is a performance cost for doing so. Now you have the option of specifying `MDATA` sections as read-only, which means they can only be changed when the document is updated and the index is synchronized.

This feature provides better performance for queries because an extra cursor is not required to handle read-only `MDATA` sections. Reducing the number of cursors required can also prevent exceeding the limit of the `OPEN_CURSORS` system parameter.

Related Topics

- *Oracle® Text Application Developer's Guide*

Text: Sentiment Analysis and Collocates

Oracle Text supports sentiment analysis and collocates. Sentiment analysis provides identification of positive and negative trends associated with search terms.

The identification of positive or negative trends associated with search terms allows the building of richer search applications.

Related Topics

- *Oracle® Text Application Developer's Guide*

Text: Join Character Support for JAPANESE_VGRAM_LEXER and WORLD_LEXER

Support is added for `printjoins` and `skipjoins` characters in the `JAPANESE_VGRAM_LEXER` and `WORLD_LEXER` types.

The benefit is a more consistent treatment of Latin strings when they are handled by the `JAPANESE_VGRAM_LEXER` type for some documents, and by `BASIC_LEXER` for other documents.

Text: Extracting Synonyms of Words in Documents

The functions `TOKENS` and `POLICY_TOKENS` in the `CTX_DOC` PL/SQL package now allow a thesaurus name to be specified. When this is done, the tokens returned include the actual words in the document and also all synonyms of those words as defined by the specified thesaurus.

This feature provides advantages for data analysis and data mining software because all variants of the words found in the document can be processed.

Related Topics

- *Oracle® Text Reference*

Text: Changes to NDATA, NEAR2, and NESTED NEAR Query Operators

The new Oracle Text `BESTMATCH` query operator can find a collection of terms within a document even when all terms are not present or are present in variant forms.

This feature increases recall when exact query terms are not known.

Related Topics

- *Oracle® Text Application Developer's Guide*

Materialized Views: Refresh Statistics History

Materialized views refresh statistics can be collected in varying degrees of granularity to provide historical data for analysis and reporting.

Storing historical materialized view refresh statistics provides insight into how the materialized view ecosystem (or a single, specific materialized view) has evolved. This data provides unique insight, both for historical analysis as well as for diagnosis purposes.

Related Topics

- *Oracle® Database Data Warehousing Guide*

Process Management

This feature improves connection times and parallel processing times by pre-creating process resources. An administrator can control how many processes to pre-create so that new connections do not have to wait for milliseconds or seconds. Instead, processes can use pre-created process resources, thus reducing connect and parallel processing times.

The feature improves connect and failover times for user connections and process spawn times for background processes.

Related Topics

- *Oracle® Database Administrator's Guide*

Partitioning: Read-Only Partitions

Partitions and sub-partitions can be individually set to a read-only state. This then disables DML operations on these read-only partitions and sub-partitions. This is an extension to the existing read-only table functionality.

Read-only partitions and subpartitions enable fine-grained control over DML activity. This enhances the data management capabilities of partitioned tables.

Related Topics

- *Oracle® Database VLDB and Partitioning Guide*

Partitioning: Multi-Column List Partitioning

List partitioning functionality is expanded to enable multiple partition key columns.

Using multiple columns to define the partitioning criteria for list partitioned tables enables new classes of applications to benefit from partitioning.

Related Topics

- *Oracle® Database VLDB and Partitioning Guide*

General

- JDBC Support for BigSCN
- Parameter Default and File Location Changes for Read-Only Oracle Home for Oracle Text

JDBC Support for BigSCN

A system change number (SCN) is an internal number that is generated for every transaction. The original size of an SCN is 6 bytes. Now, the JDBC drivers support BigSCN which is 8 bytes.

Customers with a very large number of transactions and using 12.1 or earlier drivers may experience the limitation due to 6 bytes of SCN. With this feature, JDBC drivers support a BigSCN of 8 bytes. This helps when you have a very large number of transactions to run smoothly.

Parameter Default and File Location Changes for Read-Only Oracle Home for Oracle Text

This feature changes some parameter defaults as well as the locations of some files as follows:

- Default value of `LOG_DIRECTORY`
  The default value of parameter `LOG_DIRECTORY` changes from `$ORACLE_HOME` to `$ORACLE_BASE_HOME`. For Oracle Text, the default log directory is now `$ORACLE_BASE_HOME/ctx/log/`.
- `CTX_OUTPUT`
  The default path of `CTX_OUTPUT.START_LOG` and `CTX_OUTPUT.START_QUERY_LOG` is now `$ORACLE_BASE_HOME` because, in some configurations, `$ORACLE_HOME` is now read only.
- `ctxload`
  The default path of `ctxload` changes to the current directory, which can be any directory with write privileges. The exception is `$ORACLE_HOME` because `$ORACLE_HOME` is read-only from this point forward.

  Log file name: The default log directory is no longer the current directory. The default directory is now `$ORACLE_BASE_HOME/ctx/log`.
- `ctxkbtc`
  Log file name: The default log directory is no longer the current directory. The default directory is now `$ORACLE_BASE_HOME/ctx/log`.

Related Topics

- *Oracle® Text Reference*

Pluggable Databases Ease-of-Adoption

- I/O Rate Limits for PDBs
- Heat Map and Automatic Data Optimization Support for CDBs
- PDB Character Set
- PDB Refresh
- Hints in CONTAINERS Query
- Parallel PDB Creation Clause
- PDB Archive Files (.pdb Files)
- Default Tablespace Clause
- Cloning a PDB
- Near Zero Downtime PDB Relocation
- Logical Standby Database to Support CDBs with PDBs with Different Character Sets
- LogMiner to Support CDBs with PDBs with Different Character Sets

- Support for PDBs with Different Character Sets, Time Zone File Versions, and Database Time Zones in a CDB

I/O Rate Limits for PDBs

This feature limits the rate of physical I/O issued by a pluggable database (PDB). You can specify the limit as either I/O requests per second or as Mbps (megabytes of I/O per second). This limit can only be applied to a PDB, not to the multitenant container database (CDB) or a non-CDB.

Background I/O, such as redo log writes or dirty buffer cache writes, are not limited. The limits are specified with the new PDB parameters, `MAX_IOPS` and `MAX_MBPS`.

Related Topics

- *Oracle® Database Administrator's Guide*

Heat Map and Automatic Data Optimization Support for CDBs

Heat Map and Automatic Data Optimization (ADO) now support multitenant container databases (CDBs).

Heat Map and ADO are now usable in databases that are consolidated using Oracle Multitenant, providing the benefits of automatic tracking of access patterns at the block and segment level, and automatic storage and compression tiering of data based on user-defined policies.

Related Topics

- *Oracle® Database VLDB and Partitioning Guide*

PDB Character Set

You can now have a different character set for each pluggable database (PDB) in the same multitenant container database (CDB).

In Oracle Database 12*c* Release 1 (12.1.0.1), the character set of the PDB had to be binary-compatible with that of the CDB. In practice, this is a barrier to consolidation. By allowing PDBs with diverse character sets to be plugged into a single CDB, it is easier to realize the benefits of consolidation.

Related Topics

- *Oracle® Database Administrator's Guide*

PDB Refresh

Customers want to periodically propagate changes from a source pluggable database (PDB) to its cloned copy. In such cases, we say that the cloned PDB is a refreshable copy of the source PDB. A refreshable cloned PDB can only be opened in read-only mode and propagating changes from the source PDB can be performed manually (on demand) or automatically.

This feature eliminates the need to re-create a cloned environment from scratch and also ensures that a cloned PDB can be periodically synchronized with source changes. This feature significantly reduces the time to provision a cloned copy of PDBs.

Related Topics

- *Oracle® Database Administrator's Guide*

Hints in CONTAINERS Query

The CONTAINERS hint introduced in 12.1.0.2 is a great way to aggregate data across multiple pluggable databases (PDBs) in the multitenant container database (CDB). When a CONTAINERS () query is submitted, recursive SQL statements are generated and executed in each PDB. Hints can be passed to these recursive SQL statements by using the CONTAINERS statement-level hint. For example:

```
SELECT /*+ CONTAINERS (DEFAULT_PDB_HINT=<hint_string>) */ ...
FROM CONTAINERS (<object>)
```

In the above example, the string <*hint_string*> is used as the hint for each recursive SQL statement executed under the covers of CONTAINERS (<*object*>).

With this new feature, you can control the SQL execution plans of recursive SQL statements generated from the CONTAINERS () query.

Related Topics

- *Oracle® Database SQL Language Reference*

Parallel PDB Creation Clause

This new clause parallelizes the creation of a pluggable database (PDB) when creating it from the seed PDB. If no number is specified, the degree of parallelism is chosen by Oracle Database. However, when a integer is specified with the PARALLEL clause, then file copy during PDB creation is parallelized with a degree of parallelism equal to the integer specified. Oracle Database has the option of not honor the specified degree of parallelism depending on the current database load.

Parallel PDB creation speeds up the provisioning of PDBs.

Related Topics

- *Oracle® Database Administrator's Guide*

PDB Archive Files (.pdb Files)

When a pluggable database (PDB) is unplugged, all data files associated with a PDB along with the PDB manifest need to be copied or moved individually over to the remote server where it is plugged into another multitenant container database (CDB). With this new feature, a you can choose to create a single PDB archive file, a compressed file with the .pdb extension, that contains the PDB manifest and all of the data files when unplugging a PDB. When plugging in a PDB, the presence of a .pdb file is interpreted and the PDB is plugged into the CDB. You can choose to run the PDB plug-in compatibility test directly on the PDB archive without extracting the PDB manifest file from the archive.

This feature provides ease of managing the unplugging and plugging of PDBs across multitenant container databases.

Related Topics

- *Oracle® Database Administrator's Guide*

Default Tablespace Clause

In Oracle Database 12*c* Release 1 (12.1), the default tablespace was allowed when creating a pluggable database (PDB) using the seed PDB. In Oracle Database 12*c* Release 2 (12.2), this clause enables the remaining types of the `CREATE PLUGGABLE DATABASE` statement. The specified tablespace needs to already exist in the PDB.

This features provides the flexibility to choose a default tablespace during any PDB creation operation.

Related Topics

- *Oracle® Database Administrator's Guide*

Cloning a PDB

Cloning of a pluggable database (PDB) resolves the issue of setting the source system to read-only mode before creating a full or snapshot clone of a PDB. With this feature, you can now clone your production PDBs and create point-in-time copies for development or testing without any application outage.

This feature eliminates the need to create application outages for the purpose of cloning.

Related Topics

- *Oracle® Database Administrator's Guide*

Near Zero Downtime PDB Relocation

This new feature significantly reduces downtime by leveraging the clone functionality to relocate a pluggable database (PDB) from one multitenant container database (CDB) to another CDB. The source PDB is still open and fully functional while the actual cloning operation is taking place. The application outage is reduced to a very small window while the source PDB is quiesced and the destination PDB is brought online after applying the incremental redo. The source PDB is subsequently dropped.

With this feature, you can now adjust to changing workloads and comply with Service Level Agreement (SLA) requirements in near real time.

Related Topics

- *Oracle® Database Administrator's Guide*

Logical Standby Database to Support CDBs with PDBs with Different Character Sets

In Oracle Database 12*c* Release 2 (12.2), multitenant container databases (CDBs) allow pluggable databases (PDBs) to have different character sets as long as the root container has a character set that is the superset of all of the PDBs character sets. Logical standby databases support such a primary database.

This feature enables the rolling upgrade of a CDB with PDBs that use different character sets.

Related Topics

- *Oracle® Data Guard Concepts and Administration*

LogMiner to Support CDBs with PDBs with Different Character Sets

In Oracle Database 12*c* Release 2 (12.2), multitenant container databases (CDBs) allow pluggable databases (PDBs) to have different character sets as long as the root container has a character set that is the superset of all of the PDB character sets. LogMiner supports such a primary database.

This feature enables the rolling upgrade of a CDB with PDBs that use different character sets.

Related Topics

- *Oracle® Database Utilities*

Support for PDBs with Different Character Sets, Time Zone File Versions, and Database Time Zones in a CDB

This feature enables pluggable databases (PDB) in a consolidated database (CDB) to have different character sets, time zone file versions, and database time zones.

CDB enables multiple departmental databases to be consolidated into one consolidated database with each of these databases as pluggable databases (PDBs). This feature enables the consolidation of databases that have different character sets, time zone file versions, and database time zones into a CDB.

Related Topics

- *Oracle® Call Interface Programmer's Guide*

Pluggable Databases Multitenancy

- Memory Resource Management
- Per-Process PGA Limits
- Performance Profiles and Mandatory PDB Profiles
- CDB-Level PDB Lockdown
- Application Root
- Proxy PDB
- Forwarding Connections to A New Address Based on Service
- Service-Level ACLs for TCP Protocol

Memory Resource Management

Oracle Database Resource Manager (the Resource Manager) can now manage memory usage between pluggable databases (PDBs) within a multitenant container database (CDB).

This feature prevents one PDB from using too much buffer cache, library cache, or program global area (PGA), which can result in poor performance for another PDB. At the same time, this feature helps to maintain good overall database performance.

Related Topics

- *Oracle® Database Administrator's Guide*

Per-Process PGA Limits

Program global area (PGA) now restricts users from using large amounts of PGA, thus handling runaway PGA because of badly written SQL statements. In a data warehouse, the PGA can be used by well-defined reports over ad-hoc users. This feature also limits temporary space usage.

The limits are configured through the Resource Manager using the `CREATE_PLAN_DIRECTIVE()` procedure of the `DBMS_RESOURCE_MANAGER` package.

DBAs can also manually handle runaway queries using the `CANCEL_SQL` group name of the `SWITCH_GROUP` parameter of the `CREATE_PLAN_DIRECTIVE()` procedure or by using the `DEQUEUE_PARALLEL_STATEMENT` procedure both of the `DBMS_RESOURCE_MANAGER` package.

A runaway query using excessive amounts of PGA or temporary space can result in serious problems on a database. In a multitenant container database (CDB) or consolidated environment, this type of query can affect other pluggable databases and even other databases.

Related Topics

- *Oracle® Database Administrator's Guide*

Performance Profiles and Mandatory PDB Profiles

For multitenant container databases (CDBs) with thousands of pluggable databases (PDBs), it is cumbersome to configure memory and other Resource Manager directives. In addition, CDBs need a way to require certain PDB-level directives (like program global area (PGA) memory limits) to be enforced across all PDBs.

The new database parameter, `DB_PERFORMANCE_PROFILE`, can be used to specify a class of database or pluggable database (for example, gold, silver, or bronze). The CDB administrator can then create CDB plan directives for all gold, silver or bronze PDBs using the PL/SQL procedure `DBMS_RESOURCE_MANAGER.CREATE_CDB_PROFILE_DIRECTIVE()`.

This feature is required for managing CDBs that have many pluggable databases in a public cloud context.

Related Topics

- *Oracle® Database Administrator's Guide*

CDB-Level PDB Lockdown

With the pluggable database (PDB) lockdown profile, you can specify a simple lockdown level for a given multitenant container database (CDB). A PDB lockdown profile is a security mechanism to restrict operations that are available to local users connected to a specified PDB. The lockdown levels can be as simple as high, medium and low. The high level restricts access to operating system files, Oracle XML Database, and external tables. The low level enables these features.

The PDB lockdown profile provides enhanced security in a multitenant environment.

Related Topics

- *Oracle® Database Security Guide*

Application Root

Federation of pluggable databases (PDBs) enables you to create a common application data model that can be shared across multiple tenants participating in the federation. You can also create and maintain a common data source that can be referenced by individual tenants.

The federation of PDBs improves operational efficiency to maintain multiple application tenants from a single master.

Related Topics

- *Oracle® Database Administrator's Guide*

Proxy PDB

A proxy pluggable database (PDB) provides fully functional access to another PDB in a remote multitenant container database (CDB). This feature enables you to build location-transparent applications that can aggregate data from multiple sources that are in the same data center or distributed across data centers.

A proxy PDB provides the capability to aggregate data from multiple application backends.

Related Topics

- *Oracle® Database Administrator's Guide*

Forwarding Connections to A New Address Based on Service

Starting with Oracle Database 12*c* Release 2 (12.2), Oracle Net listener supports multiple redirects with the introduction of a new listener parameter, `ALLOW_MULTIPLE_REDIRECTS_listener_name`. You can still use the old listener and you do not need to change the client application. The old listener forwards connections based on service to the relocated address.

The new listener supports seamless migration of a pluggable database (PDB) from a local database to the Oracle Public Cloud.

Related Topics

- *Oracle® Database Net Services Administrator's Guide*

Service-Level ACLs for TCP Protocol

With this feature, every database service can have its own access control list (ACL) and the ACL is based on IPs. Because each pluggable database is a different service, this feature enables different pluggable databases to have different ACLs. These ACLs are enforced by the listener. Access to a pluggable database service is enabled only for IPs that are permitted through an ACL.

This feature improves pluggable database security.

Related Topics

- *Oracle® Database Net Services Administrator's Guide*

Pluggable Databases Overall

- Flashback Pluggable Database
- Upgrading a CDB With One or More PDBs in a Single Operation
- Support for Thousands of Pluggable Databases for Each Multitenant Container Database
- Pluggable Database Lockdown Profiles Enhancements
- Pluggable Database Operating System Credentials

Flashback Pluggable Database

You can now set up restore points specific for a pluggable database (PDB) to flash back to that restore point without affecting other PDBs in a multitenant container database (CDB). Both normal restore points (which are an assignment of a system change number (SCN) to an alias) and guaranteed restore points (which guarantee the database to be flashed back to a point in time) can be performed at the PDB level.

Restore points provide an ease-of-use method to assign an alias to SCN. Using the alias, you can rewind the database to that point in time using flashback pluggable database. By enabling this option, you can perform point-in-time recovery (PITR) for a specific PDB instead of the entire CDB.

Related Topics

- *Oracle® Database Backup and Recovery User's Guide*

Upgrading a CDB With One or More PDBs in a Single Operation

You can now upgrade a multitenant container database (CDB) with one or more pluggable databases (PDBs) plugged into it in a single operation.

Lower management overhead is now possible because a database administrator (DBA) can upgrade a CDB with multiple PDBs in a single operation.

Related Topics

- *Oracle® Database Upgrade Guide*

Support for Thousands of Pluggable Databases for Each Multitenant Container Database

Oracle Database 12*c* Release 2 (12.2) now supports having thousands of pluggable databases (PDBs) in a single multitenant container database (CDB).

Oracle Database 12*c* Release 1 (12.1.0.1) supported up to 252 PDBs for each CDB. By supporting higher numbers of PDBs for each CDB, you have fewer CDBs to manage and thus benefit from further reductions in operating expenses.

Related Topics

- *Oracle® Database Administrator's Guide*

Pluggable Database Lockdown Profiles Enhancements

In a multitenant environment, a pluggable database (PDB) lockdown profile is a mechanism used to restrict operations that can be performed by connections to a given PDB for both cloud and non-cloud environments. By default, there are three available profiles. For example, `SOFTWARE_AS_A_SERVICE`, `PRIVATE_DB_AS_A_SERVICE` and `PUBLIC_DB_AS_A_SERVICE`. These profiles are defined in increasing order of restriction. You can alter these default profiles or create new profiles that are appropriate to their security requirements.

PDB lockdown profiles provide the flexibility to define custom security policies according to the security requirements of the application. This feature also alleviates security concerns with public Database as a Service (DBaaS) and this feature helps in cloud adoption.

Related Topics

- *Oracle® Database Security Guide*

Pluggable Database Operating System Credentials

The Oracle Database operating system user is usually a highly privileged user. Using that user during operating system interactions can expose vulnerabilities for security exploits. Furthermore, using the same operating system user for operating system interactions from different pluggable databases (PDB) can compromise data that belongs to a given PDB. Using the operating system that is described by a credential (refer to `DBMS_CREDENTIAL` on how to create operating system user credentials) whose name is specified as a value of the `PDB_OS_CREDENTIAL` parameter, ensures that operating system interactions are performed as a less powerful user. This also provides the ability to protect data that belongs to one PDB from being accessed by users connected to another PDB.

This feature provides enhanced security when interacting with the operating system from a PDB.

Related Topics

- *Oracle® Database Security Guide*

Upgrades

- Enhancing Ease-of-Use and Reporting From Pre-Upgrade Information Tool
- Automatically Set User Tablespaces to Read-Only During Upgrade

Enhancing Ease-of-Use and Reporting From Pre-Upgrade Information Tool

The Pre-Upgrade Information Tool is enhanced in several ways. Messages are reformatted and rewritten to improve clarity and consistency. Fix up routines are expanded and enhanced, and can self-validate to detect cases where the fix up is no longer needed. The Pre-Upgrade Information Tool is now delivered as a single .jar file for easier porting, copying, and migrating of the tool between systems.

These improvements to upgrade automation reduce the efforts needed to upgrade a database.

Related Topics

- *Oracle® Database Upgrade Guide*

Automatically Set User Tablespaces to Read-Only During Upgrade

The new `-T` option for the parallel upgrade utility (`catctl.pl`) can be used to automatically set user tablespaces to read-only during an upgrade, and then back to read/write after the upgrade.

Use this new functionality for a faster fallback strategy if a problem is encountered during the upgrade. Take the following steps.

1. Copy the system tablespaces to local disks, for potential fallback, if needed.
2. Use the `-T` option in the upgrade to set user tablespaces to read-only for the duration of the upgrade.
3. If there is a problem during the upgrade, then the database can be quickly restarted by restoring the copies of the system tablespaces and opening the database in the original (old) Oracle home.

The Database Upgrade Assistant (DBUA) also supports this functionality.

Related Topics

- *Oracle® Database Upgrade Guide*

Utilities

- Support for Tables With LONG Data Types in Data Pump Network Mode Operations
- Metadata Transforms Available for Oracle Data Pump in New Views
- Adding Oracle Data Pump and SQL*Loader Utilities to Instant Client
- SDF_PREFIX Parameter for SQL*Loader
- Allowing Multi-Byte Strings for SQL*Loader Parameters Used for Express Mode
- Database Migrations: Support LLS Files Generated by DB2 Export Utility

Scheduler: Job Incompatibilities

You can now specify when one or more jobs cannot execute at the same time.

There are cases when a job should not execute when another job is already running. If two jobs are using the same resource, you can specify that the two jobs cannot execute at the same time.

Related Topics

- *Oracle® Database Administrator's Guide*

Scheduler: Resource Queues

You can now specify how many defined resources are required to execute a job. A resource can be anything specified by the user and only has two attributes, `name` and `count.` At execution time, Oracle Scheduler ensures that running jobs are not going to exceed the available resources. If there are resource limitations, then you can define a resource and set its properties. In the job definition, you can then specify which resources are required to run a job.

Jobs cannot execute until the required resources are available.

Related Topics

- *Oracle® Database Administrator's Guide*

Scheduler: In-Memory Jobs

Beginning with this release, you can create in-memory jobs. For in-memory jobs, there is minimal data written to disk as compared to regular jobs. There are two types of in-memory jobs, repeating in-memory jobs and one-time in-memory jobs. For one-time in-memory jobs, nothing is written to disk. For repeating in-memory jobs, job metadata is written to disk but there is no run information.

In-memory jobs have better performance and scalability.

Related Topics

- *Oracle® Database Administrator's Guide*

Oracle Data Pump Parallel Import of Metadata

The `PARALLEL` parameter for Oracle Data Pump, which previously applied only to data, has been extended to include metadata import operations. The performance of Oracle Data Pump import jobs has improved by enabling the use of multiple processes working in parallel to import metadata.

Oracle Data Pump jobs now take less time because parallel import of metadata is added to this release.

Related Topics

- *Oracle® Database Utilities*

Oracle Data Pump Parallel Export of Metadata

The PARALLEL parameter for Oracle Data Pump, which previously applied only to data, is extended to include metadata export operations. The performance of Oracle Data Pump export jobs is improved by enabling the use of multiple processes working in parallel to export metadata.

Oracle Data Pump jobs now require shorter down time during migration and shorter elapsed time for export operations.

Related Topics

- *Oracle® Database Utilities*

New Options for Substitution Variables in Oracle Data Pump File Names

Choices for substitution (wildcard) variables are now available for Oracle Data Pump dump file names. The new choices include date or time values, a larger range for numeric values, and system generated unique file names.

Substitution variables improve file management for Oracle Data Pump dump files and enables you to take advantage of higher degrees of parallel processing without manually specifying individual file names.

Related Topics

- *Oracle® Database Utilities*

Renaming Data Files During Import

New syntax is added to let users specify new file names, or file name transforms, for the data files in a transportable tablespace job.

Renaming data files during import lowers management overhead by eliminating the need to execute ALTER TABLESPACE statements and rename or move files manually after the import is complete.

Related Topics

- *Oracle® Database Utilities*

TRUST_EXISTING_TABLE_PARTITIONS Flag for DATA_OPTIONS Parameter of Import

A new TRUST_EXISTING_TABLE_PARTITIONS flag is added to the DATA_OPTIONS parameter for import. This option tells the Data Pump to load partition data in parallel into existing tables. This is done as part of a migration when the metadata is static and can be moved before the databases are taken offline to migrate the data. Moving the metadata separately minimizes downtime. If the DBA uses this mechanism and if other attributes of the database are the same (for example, character set), then the data from the export database goes to the same partitions in the import database.

This new TRUST_EXISTING_TABLE_PARTITIONS flag enables data from multiple partitions to be loaded in parallel into a preexisting table which reduces the import time.

Related Topics

- *Oracle® Database Utilities*

GROUP_PARTITION_TABLE_DATA Flag for DATA_OPTIONS Parameter of Export

A new GROUP_PARTITION_TABLE_DATA flag is added to the DATA_OPTIONS parameter for export. This option tells the Data Pump to unload all table data in one operation rather than unload each table partition as a separate operation. Then, the definition of the table does not matter at import time. Import sees the partition of data that is loaded into the entire table.

The new GROUP_PARTITION_TABLE_DATA flag enables data for all partitions to be loaded at once and in parallel. This reduces the time to import the table data.

Related Topics

- *Oracle® Database Utilities*

Data Verification Option to Import, ORACLE_LOADER Access Driver and OCIDirPath

New options are added to verify that data for date and date fields in tables is valid. Use this option when importing a dump file from an untrusted source to prevent issues that can occur because data is corrupt in the dump file. Because of the overhead involved in validating data, the default is that data is no longer validated on import.

This verification protects the database from SQL injection bugs from bad data. Oracle Corporation recommends using this option when importing a dump file from an untrusted source.

Related Topics

- *Oracle® Database Utilities*

ORACLE_DATAPUMP and ORACLE_LOADER Access Driver Supports New File Format

Oracle Loader for Hadoop (OLH) uses files that are written and read by the ORACLE_DATAPUMP access driver. The problem with the current file format is that the first two blocks of the file have to be updated after the rest of the file is written. The Hadoop file system used by OLH is write-once, so the header blocks cannot be updated. This feature adds a new file format that can be written and read by the ORACLE_DATAPUMP access driver in the Hadoop file system. This file format adds two trailer blocks to the file with the header information so that the beginning of the file does not need to be updated after the rest of the file is written. The new Hadoop trailer format can be enabled for a dump file using the new HADOOP_TRAILERS access parameter.

The new file format enables OLH to run faster since it no longer needs to write file data multiple times to avoid updating the first two header blocks in the file.

Support for Tables With LONG Data Types in Data Pump Network Mode Operations

Data Pump normally moves table data in a network import by using the `INSERT AS SELECT` SQL statement. However, `INSERT AS SELECT` cannot be used to move `LONG` columns, therefore, data for those tables are not moved. For cases where it is possible, the Data Pump uses `OCIDirPathUnload` to unload the data from the remote database and then uses `OCIDirPathLoad` to load the table data into the target database. `OCIDirPath` allows the Data Pump to move tables with most data types including `LONG`s. For tables that contain data types that cannot be moved with `OCIDirPath`, the Data Pump continues to use `INSERT AS SELECT`. If a table has both a `LONG` column and one of the data types that `OCIDirPath` cannot move, then network import cannot move that table.

This support also makes network import available to more users.

Related Topics

- *Oracle® Database Utilities*

Metadata Transforms Available for Oracle Data Pump in New Views

This feature adds views to Oracle Database that provide information about what metadata transforms are available for different modes of Oracle Data Pump and for the metadata API.

Currently, transforms are documented with Oracle Data Pump and the metadata API, but DBAs prefer to query the database to get that information. This feature ensures that documentation for a feature is maintained as part of creating a transform, and that this information is available using interfaces such as SQL*Plus.

Related Topics

- *Oracle® Database Utilities*

Adding Oracle Data Pump and SQL*Loader Utilities to Instant Client

This feature adds SQL*Loader, `expdp, impdp, exp,` and `imp` to the tools for instant client.

Now you can run these utilities on machines that do not have a complete Oracle Database installation.

Related Topics

- *Oracle® Database Utilities*

SDF_PREFIX Parameter for SQL*Loader

This feature provides a new parameter `SDF_PREFIX` to the SQL*Loader command line. The value of this parameter is appended to the names of `LOBFILES` and secondary data files used in the load. This enables you to create a data file that you can use on any system without hard coding the complete file specification in a data file. Instead, the file names that are in the data file can all be relative to the path specified in the SDF_PREFIX parameter. You can also specify this parameter in the OPTIONS clause inside the SQL*Loader control file.

This feature simplifies the distribution of data files that can be loaded from different directory paths on different machines.

Related Topics

- *Oracle® Database Utilities*

Allowing Multi-Byte Strings for SQL*Loader Parameters Used for Express Mode

Currently, the ENCLOSED_BY, OPTIONALLY_ENCLOSED_BY, and TERMINATED_BY command line parameters for SQL*Loader only accept a single character as a value. This feature changes these parameters to accept strings as a value.

This feature gives you more flexibility in the types of files that you can load without creating a SQL*Loader control file.

Related Topics

- *Oracle® Database Utilities*

Database Migrations: Support LLS Files Generated by DB2 Export Utility

The DB2 export utility unloads table data into text files with the option to unload LOB data, either character or binary, into a separate file. When LOB data is unloaded into a separate file, DB2 writes a Lob Locator Specifier (LLS) into the data file. The LLS contains the file name, offset and length of the LOB data in the data file. This feature adds a clause for SQL*Loader control file and ORACLE_LOADER access parameters to enable you to indicate that a field in the data file is an LLS field. SQL*Loader and ORACLE_LOADER use the information in the field to read the data for the LOB column.

This feature aids in migrating data from DB2 to Oracle Database.

Related Topics

- *Oracle® Database Utilities*

Diagnosability

Bug Resolution Process Improvements

Trace File Analyzer: Web-Based Visualization

The Trace File Analyzer (TFA) utility has become the standard for collecting diagnostic information in Oracle Database, including Oracle Database single-instance and Oracle

Real Application Clusters (Oracle RAC). The TFA web-based visualization feature adds easy-to-navigate web-based visualization to TFA. The TFA web-based visualization feature is installed as part of Oracle Grid Infrastructure.

Using TFA web sourcing, reviewing, and analyzing diagnostic information gathered as part of a TFA collection becomes easier and more efficient, leading to reduced recovery time.

Related Topics

- *Oracle® Autonomous Health Framework User's Guide*

Failure Analytics

- Trace File Analyzer: Optimizing Diagnostic Collection
- Trace File Analyzer: Automating Diagnostic Collection
- Trace File Analyzer: Managing Diagnostic Collection
- Trace File Analyzer: Feedback to Auxiliary Systems

Trace File Analyzer: Optimizing Diagnostic Collection

The Oracle Trace File Analyzer (TFA) Collector is a diagnostic collection utility to simplify diagnostic data collection for Oracle Clusterware, Oracle Grid Infrastructure, and Oracle Real Application Clusters (Oracle RAC) systems.

While similar to the diagnostic collection feature, the Oracle Trace File Analyzer (TFA) Collector allows centralization and automatic collection of diagnostic information. TFA encapsulates diagnostic data collection for all clusters and Oracle Database components on all servers into a single command executed on one server only. The result can be stored on a central server and is trimmed to reduce data upload size. TFA can also be instructed to collect data for a particular product only.

Related Topics

- *Oracle® Autonomous Health Framework User's Guide*

Trace File Analyzer: Automating Diagnostic Collection

The Oracle Trace File Analyzer (TFA) Collector provides the option to automatically collect diagnostic information when TFA detects an incident.

Incident-based diagnostic collection eases the diagnostic collection burden by instructing the Oracle Trace File Analyzer (TFA) Collector to take action only when certain messages are seen. By default, TFA considers Oracle Database alert logs, Oracle ASM alert logs, and Oracle Clusterware alert logs for this feature.

Related Topics

- *Oracle® Autonomous Health Framework User's Guide*

Trace File Analyzer: Managing Diagnostic Collection

The Oracle Trace File Analyzer (TFA) Collector allows you to choose between automated and manual collection of diagnostic data. TFA, by default, uses a maximum repository size of either 10 GB or 50% of the file system size on which the TFA repository resides, whichever is greater.

The ability to have the Oracle Trace File Analyzer (TFA) Collector manage the data collected not only simplifies the management of the system, but also ensures that relevant data is provided when needed.

Related Topics

- *Oracle® Autonomous Health Framework User's Guide*

Trace File Analyzer: Feedback to Auxiliary Systems

Oracle Trace File Analyzer (TFA) collects diagnostic data that can either be analyzed directly or serve as a data stream to auxiliary systems, such as Oracle Support Services, to be visualized or analyzed in a certain context.

The ability to feedback data to auxiliary systems makes Oracle Trace File Analyzer (TFA) the preferred collection tool for all Oracle Database related diagnostic data.

Related Topics

- *Oracle® Autonomous Health Framework User's Guide*

First Failure Capture

- XStream: Replication Event View
- XStream and GoldenGate Inbound Server Message Tracking
- Trace File Analyzer: Enabling Time-Trimmed Diagnostic Collection

XStream: Replication Event View

The `DBA_REPLICATION_PROCESS_EVENTS` view records process events, such as configuration changes, start up, and shut down in a database view.

The `DBA_REPLICATION_PROCESS_EVENTS` view provides visibility to replication activity for both database and replication administrators.

Related Topics

- *Oracle® Database XStream Guide*

XStream and GoldenGate Inbound Server Message Tracking

Users can enable message tracking functionality as an apply process parameter. The `MESSAGE_TRACKING_FREQUENCY` parameter specifies the frequency at which messages received by the apply process are automatically tracked.

When enabled, this feature provides the database and replication administrators with more detail about the logical change records being processed.

Related Topics

- *Oracle® Database XStream Guide*

Trace File Analyzer: Enabling Time-Trimmed Diagnostic Collection

The trimming of files collected by the automatic diagnostic collection feature of the Oracle Trace File Analyzer can be controlled by the `trimfiles` parameter.

When enabled (which is the default), files are trimmed to only include data from around the time of the event.

Related Topics

- *Oracle® Autonomous Health Framework User's Guide*

Exadata

- General

General

- RMAN: Backing Up Sparse Databases in the Backup-Set Format
- RMAN: Backing Up Sparse Databases in the Image Copy Format
- RMAN: Restoring From Sparse Backups
- RMAN: Recovering the Sparse Database
- RMAN: Purging Sparse Backups

RMAN: Backing Up Sparse Databases in the Backup-Set Format

To support RMAN backup with sparse files, this feature introduces a new command `BACKUP AS SPARSE BACKUPSET <object>` where the object can be a data file, tablespace, or database. Running this command allows RMAN to take a backup of a sparse database or data file (this can be a pluggable database (PDB)). When this command is used at a level higher than the data file level, the command ultimately gets translated into individual data file level backup commands (for example, `BACKUP AS SPARSE DATAFILE`). Sparse backup supports full backup of databases, data files, and level 0 incremental and level 1 incremental backups.

By doing backups for a sparse database, the entire base database is not backed up. Only the delta storage file (changes happened using the sparse database) is backed up. This dramatically reduces the overall backup time and the space required to store the backups.

Related Topics

- *Oracle® Database Backup and Recovery User's Guide*

RMAN: Backing Up Sparse Databases in the Image Copy Format

RMAN now allows image copies to be created from sparse databases by leveraging the underlying sparse database mechanism to perform the image copy backup. There is no new command required for backup in the image copy format.

The new feature allows duplication or backup of a sparse database for other purposes.

Related Topics

- *Oracle® Database Backup and Recovery User's Guide*

RMAN: Restoring From Sparse Backups

The RMAN `RESTORE` command determines the most appropriate (regular or sparse) backup set, and restores the data files from the backup set. Recovery behavior changes depending on whether or not RMAN chooses a sparse backup set in the previous `RESTORE` command.

This feature allows a sparse database to be restored without affecting the base data files.

Related Topics

- *Oracle® Database Backup and Recovery User's Guide*

RMAN: Recovering the Sparse Database

An RMAN recovery operation is performed to bring the sparse database to the current time or to a certain point-in-time in the past. This usually follows a restore operation.

This feature allows complete or incomplete (point-in-time) recovery of a sparse database without affecting the base data files.

Related Topics

- *Oracle® Database Backup and Recovery User's Guide*

RMAN: Purging Sparse Backups

A purge operation is required to delete obsolete backups of sparse databases.

This is a housekeeping activity to clean up unwanted or obsolete backups of sparse databases which are no longer required. Retention policies can also influence the categorization status of backups to mark as obsolete.

Related Topics

- *Oracle® Database Backup and Recovery Reference*

Manageability

- Automatic Performance Management
- General

Automatic Performance Management

- Oracle Data Guard Support for Oracle Diagnostics Pack
- Active Data Guard Support for SQL Tuning Advisor
- SQL Performance Analyzer and SQL Tuning Sets Enhancements

Oracle Data Guard Support for Oracle Diagnostics Pack

The Oracle Diagnostics Pack can be used with an Active Data Guard standby database that is open for read-only access. This enables you to capture the performance data to the Automatic Workload Repository (AWR) for an Active Data Guard standby database and to run Automatic Database Diagnostic Monitor (ADDM) analysis on the AWR data.

This feature enables performance tuning for read-only workloads executing on an Active Data Guard standby database.

Related Topics

- *Oracle® Database Performance Tuning Guide*

Active Data Guard Support for SQL Tuning Advisor

SQL Tuning Advisor has been enhanced so that tuning can be initiated on one database but the actual tuning process is executed remotely on a different database. This allows DBAs to offload tuning of the primary database workload to an Active Data Guard standby and tuning of Active Data Guard SQL workloads on Active Data Guard itself. When offloading SQL tuning of primary database workloads to an Active Data Guard standby, the SQL tuning process is initiated from the primary database, but the expensive tuning process is executed remotely on the Active Data Guard standby and results are written back to the primary database using database links. When tuning Active Data Guard workloads, the entire SQL tuning process is executed locally at the Active Data Guard standby while maintaining the read-only nature of the database. This is accomplished by gathering the required information over database links from the primary database and writing back any database state changes (such as SQL profile implementation) over to the primary database. SQL profile recommendations implemented on the primary database are applied to the Active Data Guard standby using the redo apply mechanism.

SQL Tuning Advisor's remote SQL tuning capability supports tuning of Active Data Guard workloads and offloading tuning of primary database workloads to an Active Data Guard standby. The business benefits of this feature are as follows:

- Improved manageability of Active Data Guard databases through support of application tuning.
- Better return on investment through utilization of available resources on Active Data Guard for primary database workload tuning.
- Improved application performance of Active Data Guard and primary database workloads.
- Improved read performance of demanding applications such as in reader-farm environments.
- Flexibility in initiating SQL Tuning Advisor and storing results remotely in a different database.
- Support for end-to-end tuning and testing of workflows when used with SQL Performance Analyzer, or otherwise.

Related Topics

- *Oracle® Data Guard Concepts and Administration*

SQL Performance Analyzer and SQL Tuning Sets Enhancements

SQL Performance Analyzer (SPA) is enhanced to provide three new task parameters:

- `EXECUTE_FULLDML_TRIGGERS`: Use this parameter to enable or disable database triggers that are fired recursively when a SPA trial is run in `FULLDML` mode.
- `EXECUTE_WITH_FIXED_DATE`: Use this parameter in SPA trials to set a fixed date for SQL that references the `SYSDATE` function.
- `NUM_ROWS_TO_FETCH`: This parameter allows you to restrict the number of rows that a SQL statement fetches based on the optimizer mode setting.

SQL Tuning Sets (STS) are enhanced as follows:

- Start and stop STS capture by explicitly using the `start_capture` and `stop_capture` API of `DBMS_SQLTUNE` PL/SQL package.
- STS capture is enhanced to support SQL workload capture across all instances of Oracle Real Application Cluster (Oracle RAC).
- STS capture now includes additional statistics that are specific to Exadata, SQL Monitoring, and In-Memory Database.
- STS capture support is provided for reporting workloads that are running on standby or an active Oracle Data Guard database.

These enhancements lead to accurate diagnosis of performance problems, improved Oracle Quality of Service Management, and better quality testing with the lowest risk and effort. This also enhances system performance and reliability and lowers your overall management costs.

Related Topics

- *Oracle® Database Testing Guide*

General

- Database Replay Enhanced PL/SQL Support
- Tracking Index Usage
- Oracle Enterprise Manager Database Express (EM Express) Resource Manager Support
- EM Express: SQL Performance Analyzer (SPA) Support
- EM Express: Simplifying Configuration for Multitenant Through Single Port Access
- EM Express: Performance Hub Enhancements
- Oracle Enterprise Manager Database Express (EM Express) Support for Simplified Database Resource Management
- Automatic Workload Repository (AWR) Support for a Pluggable Database (PDB)
- Monitoring Real-Time Database Operations

Database Replay Enhanced PL/SQL Support

Database Replay has always had the ability to capture and replay PL/SQL. This is done by capturing and replaying the top level PL/SQL calls. As of Oracle Database 12*c* Release 2 (12.2), Database Replay gives you the choice of replaying the top level PL/SQL calls or the recursive SQL called within the PL/SQL procedures. Depending on the workload, the new replay mode can perform replays with more accuracy and less divergence.

Enhanced PL/SQL support gives DBAs the ability to perform database replays with less divergence, allowing faster, easier, and more complete testing of workloads that have a lot of PL/SQL.

Related Topics

- *Oracle® Database Testing Guide*

Tracking Index Usage

A new view, `V$INDEX_USAGE_INFO`, provides information about the use of indexes in the database. The information in the view can be used to determine whether indexes are used and the frequency of usage, and this information can be used to evaluate indexing strategies in the database.

A fine-grained information about which indexes are used and how frequently they are used allows you to eliminate infrequently used indexes. This results in better database performance and more effective system utilization.

Related Topics

- *Oracle® Database Reference*

Oracle Enterprise Manager Database Express (EM Express) Resource Manager Support

The EM Express now has the ability to create, delete, edit, activate, and deactivate resource manager plans for both multitenant container databases (CDBs) and non-CDB databases. A database administrator (DBA) can create resource manager plans with the appropriate level of detail for the environment. In a CDB, the individual pluggable database (PDB) can be assigned the appropriate level of resources for their workload.

This feature provides the DBAs the ability to assure proper Quality of Service (QOS) for their databases.

Related Topics

- *Oracle® Database 2 Day DBA*

EM Express: SQL Performance Analyzer (SPA) Support

Starting with Oracle Database 12*c* Release 2 (12.2), DBAs can execute the SQL Performance Analyzer (SPA) and SPA Quick Check from Oracle Enterprise Manager Database Express (EM Express). SPA and SPA Quick Check allows the DBA to rapidly evaluate changes to the database environment that might affect database performance and to remediate any potential performance regressions assuring continuous good performance in their database.

With SPA and SPA Quick Check available in EM Express, the DBA has an additional way to validate and verify changes to the database to ensure continuous good performance and to take immediate corrective action if there is any performance regression.

Related Topics

- *Oracle® Database 2 Day DBA*

EM Express: Simplifying Configuration for Multitenant Through Single Port Access

Oracle Enterprise Manager Database Express (EM Express) is enhanced for multitenant container database (CDB) to support a single port or URL that is used to access any of the pluggable databases (PDBs) within the CDB. When logging into EM Express, you can specify a pluggable database name allowing you to log directly into that PDB.

Security in a database environment is very important. By allowing access to all PDBs in a multitenant container database (CDB) from a single port, database security is enhanced by reducing open ports and possible attack vectors. Additionally, using a single port and URL simplifies management of PDBs in a multitenant architecture.

Related Topics

- *Oracle® Database 2 Day DBA*

EM Express: Performance Hub Enhancements

Oracle Enterprise Manager Database Express (EM Express) is enhanced to support multitenant container database (CDB) architecture to show Performance Hub information in the context of an individual pluggable database (PDB). Additional performance tabs show information similar in detail to a non-CDB database.

In environments when DBAs are administering individual PDBs, it is important that they are able to view all of the necessary information to correctly tune the workload in the PDB. This enhancement allows the DBA to provide the required quality of service and meet the required service level agreements.

Related Topics

- *Oracle® Database 2 Day DBA*

Oracle Enterprise Manager Database Express (EM Express) Support for Simplified Database Resource Management

This feature significantly improves the manageability of database resources through the following enhancements:

- Monitoring the database resource consumption without a resource manager plan is enabled through the Resource Manager Dashboard
- Guidance in creating a new resource manager plan through the following:

  - Quick setup when the most common type of resource manager plan must be created
  - Full setup and how to use the charts to set directives for the plan
- Monitoring effectiveness of the currently active plan (Resource Manager Dashboard page and plan home page of EM Express)
- Full support for multitenant environment using multitenant container database (CDB) level plans (directives are on pluggable databases (PDB)) as well as PDB level plans (directives are on consumer groups)

The EM Express support for simplified management of database resources significantly reduces the burden on the database administrator (DBA) by helping to create and manage resource manager plans.

Automatic Workload Repository (AWR) Support for a Pluggable Database (PDB)

The AWR can be used in a PDB. This enables the capture and storage of performance data in the SYSAUX tablespace of the PDB.

This feature enables performance tuning and troubleshooting for workloads executing on a PDB.

Related Topics

- *Oracle® Database Performance Tuning Guide*

Monitoring Real-Time Database Operations

Real-time database operations (DBOP) monitoring functionality has been significantly enhanced as follows:

- Dynamic and external start and stop of DBOP that enables you to start and stop a DBOP from any session in the database by specifying the session identifier and serial number for a particular session.
- DBOP Persistency and Comparison: DBOP monitored executions persist to the Automatic Workload Repository (AWR) automatically and you can use them for offline and historical performance analysis. You can now compare multiple runs of the same DBOP to understand why a particular run took longer.
- DBOP Reporting and Trend Analysis: Enhanced reporting now includes SQL and PL/SQL execution details under the same DBOP. Analysis of multiple runs of a DBOP over time provides better insight into how a specific workload is behaving over time.
- Multi-Session DBOP Support: This is useful for monitoring SQL or PL/SQL that runs over multiple sessions, which is common in extraction, transformation, and loading (ETL) tasks and batch jobs.

These DBOP enhancements increase DBA efficiency by aligning business operations monitoring with end-user needs, resulting in better quality of service for business applications.

Related Topics

- *Oracle® Database SQL Tuning Guide*

Miscellaneous

- General

General

- Selective PDB Upgrades
- AWR_PDB_AUTOFLUSH_ENABLED Initialization Parameter
- ENABLE_AUTOMATIC_MAINTENANCE_PDB Initialization Parameter
- AUTOTASK_MAX_ACTIVE_PDBS Initialization Parameter

Selective PDB Upgrades

Customers can prioritize pluggable databases (PDBs) for upgrade, and also for plugged in PDBs, and then can select a subset of PDBs for upgrade.

You can set priority for PDB upgrades, so that higher priority PDBs are upgraded first.

Related Topics

- *Oracle® Database Upgrade Guide*

AWR_PDB_AUTOFLUSH_ENABLED Initialization Parameter

The `AWR_PDB_AUTOFLUSH_ENABLED` initialization parameter enables you to specify whether to enable or disable Automatic Workload Repository (AWR) snapshots for all of the pluggable databases (PDBs) in a multitenant container database (CDB) or for individual PDBs in a CDB.

ENABLE_AUTOMATIC_MAINTENANCE_PDB Initialization Parameter

The `ENABLE_AUTOMATIC_MAINTENANCE_PDB` initialization parameter can be used to enable or disable the running of automated maintenance tasks for all the pluggable databases (PDBs) in a multitenant container database (CDB) or for individual PDBs in a CDB.

AUTOTASK_MAX_ACTIVE_PDBS Initialization Parameter

The `AUTOTASK_MAX_ACTIVE_PDBS` initialization parameter enables you to specify the maximum number of pluggable databases (PDBs) that can schedule automated maintenance tasks at the same time (during a maintenance window).

Performance

- General Database Performance
- In-Memory

- Unstructured Data

General Database Performance

- Advanced Queuing: PL/SQL Enqueue and Dequeue Support for JMS Payload in Sharded Queues
- Advanced Queuing: PL/SQL Enqueue and Dequeue Support for Non-JMS Payload in Sharded Queues
- ZFS Analytics
- ExaDirect SQL*Net Adapter
- Direct NFS Client Supports Parallel NFS
- Direct NFS Client Dispatcher Support
- Global and Shared Connection Pool for Oracle Cloud and Multitenant Oracle Databases
- Oracle Database Java Virtual Machine Performance Enhancements

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

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

There are important high end customers who make use of PL/SQL APIs to enqueue and dequeue JMS payload in unsharded queues because of ease-of-use and to avoid client-server round trip time.

Related Topics

- *Oracle® Database PL/SQL Packages and Types Reference*

Advanced Queuing: PL/SQL Enqueue and Dequeue Support for Non-JMS Payload in Sharded Queues

PL/SQL APIs can now perform enqueue and dequeue operations for `ADT` and `RAW` payload in sharded queues. Similarly, the PL/SQL array APIs are exposed to sharded queue users.

There are important high end customers who make use of PL/SQL APIs to enqueue and dequeue `ADT` or `RAW` payload in unsharded queues. `ADT` payloads are important as it is a way to have different queue payloads needed by the applications.

Related Topics

- *Oracle® Database PL/SQL Packages and Types Reference*

ZFS Analytics

Database-specific drill down capability is added to ZFS analytics. You can now filter statistics on Oracle ZFS Storage Appliance (ZFSSA) based on database IDs (DBID) and pluggable database IDs (PDBID). This capability is built on top of Oracle Direct NFS Client using the NFSv4 and NFSv4.1 protocols.

With this feature, customers using Oracle Database with ZFSSA can see more details on how each database, including each pluggable database in a multitenant container database, is interacting with the storage using ZFSSA monitoring tools.

ExaDirect SQL*Net Adapter

Use the ExaDirect SQL*Net adapter for low overhead database access. In Oracle Database 12*c* Release 1 (12.1.0.2), this support is added for Exadata and Exalogic environments.

Use the new message queue transport to improve latency and throughput by leveraging Remote Direct Memory Access (RDMA) in an InfiniBand environment.

Related Topics

- *Oracle® Database Net Services Administrator's Guide*

Direct NFS Client Supports Parallel NFS

The Direct NFS client now supports Parallel NFS. Parallel NFS is a part of the NFS 4.1 protocol.

Parallel NFS provides significant performance improvements for NFS workloads. Enhancing Direct NFS client to support Parallel NFS extends those benefits to Oracle Database workload when using Direct NFS client with NFS servers that support Parallel NFS.

Related Topics

- *Oracle® Grid Infrastructure Installation and Upgrade Guide for Linux*

Direct NFS Client Dispatcher Support

Dispatcher infrastructure for Direct NFS Client is added in this release. Dispatcher enables database processes to use I/O slave processes to perform I/O operations, thus limiting the number of sockets and Transmission Control Protocol (TCP) connections required by Direct NFS Client to connect to the network file system (NFS) server.

For very large database deployments running Direct NFS Client, dispatcher supports scaling of sockets and TCP connections to multipath and clustered network attached storage (NAS).

Related Topics

- *Oracle® Grid Infrastructure Installation and Upgrade Guide for Linux*

Global and Shared Connection Pool for Oracle Cloud and Multitenant Oracle Databases

This feature introduces a shared Java connections pool for multitenant data sources. This feature leverages the new switch service functionality to reuse pooled connections across multitenant (pluggable) databases.

This feature improves scalability, Oracle Cloud deployment, multitenant deployment, diagnosability, and manageability of Oracle Database connections through a global and shared connection pool.

Related Topics

- *Oracle® Universal Connection Pool Developer's Guide*

Oracle Database Java Virtual Machine Performance Enhancements

This feature improves the performance of Java, Hadoop or JavaScript modules running in Oracle Java Virtual Machine by means of implementing additional loop optimizations to the JIT (just-in-time) compiler. The following optimizations were implemented:

- Bounds checks are eliminated from frequently executed loops by making copies of the loop body and inserting appropriate index checks to isolate the bounds checking to pre- and post-loop passes.
- Loop unrolling is performed on small, frequently executed loops.
- Vectored (SIMD) instructions are used when possible on unrolled loop bodies.

Improved performance for Java and JavaScript stored procedures as well as Oracle Database components that use Java in the database such as InterMedia, Spatial, Text, and In-Database Container for Hadoop.

In-Memory

- In-Memory Expressions
- In-Memory Virtual Columns
- In-Memory FastStart
- Automatic Data Optimization Support for In-Memory Column Store
- Join Groups
- Expression Tracking
- Oracle Database In-Memory Support on Oracle Active Data Guard
- In-Memory Column Store Dynamic Resizing

In-Memory Expressions

The In-Memory Column Store allows objects (tables, partitions, and subpartitions) to be populated in memory in a compressed columnar format. In-memory expressions enable frequently evaluated query expressions to be materialized in the In-Memory Column Store for subsequent reuse.

Populating the materialized values of frequently used query expressions into the In-Memory Column Store greatly reduces the system resources required to execute queries and allows for better scalability.

Related Topics

- *Oracle® Database In-Memory Guide*

In-Memory Virtual Columns

In-Memory virtual columns enable some or all of the user-defined virtual columns on a table to have their values materialized (precalculated) and populated into the In-Memory Column Store along with all of the non-virtual columns for that table.

Materializing the values of user-defined virtual columns into the In-Memory Column Store can greatly improve query performance by enabling the virtual column values to be scanned and filtered using In-Memory techniques such as SIMD (single instruction, multiple data) vector processing, just like a non-virtual column.

Related Topics

- *Oracle® Database In-Memory Guide*

In-Memory FastStart

In-Memory Column Store allows objects (for example, tables, partitions, and subpartitions) to be populated in-memory in a compressed columnar format. Until now, the columnar format has only been available in-memory. That meant that after a database restart, the In-Memory Column Store would have to be populated from scratch using a multiple step process that converts traditional row formatted data into the compressed columnar format and placed in-memory. In-Memory FastStart enables data to be repopulated into the In-Memory Column Store at a much faster rate than previously possible by saving a copy of the data currently populated in the In-Memory Column Store on disk in its compressed columnar format.

In-Memory FastStart significantly reduces the time it takes to repopulate data into the In-Memory Column Store after a system restart. This allows businesses to start taking advantage of the analytic query performance benefits of accessing data in a columnar format much sooner than before.

Related Topics

- *Oracle® Database In-Memory Guide*

Automatic Data Optimization Support for In-Memory Column Store

Automatic Data Optimization (ADO) enables the automation of Information Lifecycle Management (ILM) tasks. The automated capability of ADO depends on the Heat Map feature that tracks access at the row level (aggregated to block-level statistics) and at the segment level. Originally, ADO supported both compression tiering and storage tiering using policies defined at the segment or tablespace level. ADO support has now been extended to encompass the In-Memory Column Store. ADO manages the In-Memory Column Store by moving objects (tables, partitions or subpartitions) in and out of the memory based on Heat Map statistics.

ADO support for the In-Memory Column Store ensures optimal use of the In-Memory Column Store based on Heat Map statistics. Since the In-Memory Column Store is allocated from the System Global Area (SGA), it uses main memory which is a limited and expensive resource. ADO ensures that only the best candidate objects are populated in the In-Memory Column Store using user defined policies. This provides optimal performance without requiring regular intervention by the DBA to manually manage the content of the In-Memory Column Store.

Related Topics

- *Oracle® Database In-Memory Guide*

Join Groups

Data populated into the In-Memory Column Store is compressed using a number of different encoding techniques. If two columns used together in a join are encoded using different techniques, then both columns must be decompressed to conduct the join. A join group allows the user to specify which columns are used for joins across tables so those columns can always be compressed using the same encoding techniques.

Having columns that are used together in joins encoded using the same technique enables the join to be conducted without having to uncompress the columns greatly improving the efficiency of the join.

Related Topics

- *Oracle® Database In-Memory Guide*

Expression Tracking

SQL statements commonly include expressions such as "+" or "-". More complicated examples include PL/SQL functions or SQL functions like `LTRIM` and `TO_NUMBER`. A repository maintains usage information about expressions identified during compilation and captured during execution. Complicated expressions involving multiple columns or functions make it more difficult to accurately estimate selectiveness in the optimizer, resulting in suboptimal plans. Processing more information about expressions and their usage is useful for establishing better query execution plans.

Customers benefit from improved query performance for statements that include expressions.

Oracle Database In-Memory Support on Oracle Active Data Guard

Oracle Active Data Guard allows a standby database to be opened in read-only mode. This capability enables enterprises to off-load production reporting workloads from their primary databases to synchronized standby databases. Thus, you can now use the in-memory column store on an Oracle Active Data Guard standby database. This enables the reporting workload that is processed on the standby database to take advantage of accessing data in a compressed columnar format, in memory. This also enables scans, joins and aggregates to perform much faster than the traditional on-disk formats performed.

It is also possible to populate a completely different set of data in the in-memory column store on the primary and standby databases, effectively doubling the size of the in-memory column store that is available to the application.

By enabling the reporting workload that is running on an Oracle Active Data Guard standby database to use the in-memory column store, you can greatly improve the execution performance of the workload. This is because the processing can take full advantage of accessing data in a compressed columnar format, in memory.

Related Topics

- *Oracle® Database In-Memory Guide*

In-Memory Column Store Dynamic Resizing

You can now dynamically increase the size of the in-memory area while the database is open, assuming that enough memory is available within the SGA.

The in-memory column store can be resized without restarting the database.

Related Topics

- *Oracle® Database In-Memory Guide*

Unstructured Data

- Oracle Database File System File Locking
- Oracle Multimedia PL/SQL API

Oracle Database File System File Locking

This feature extends the capabilities of Oracle Database File System (DBFS) to provide file locking support for POSIX-style applications that use `dbfs_client` as a front-end interface, and for applications that use the PL/SQL interfaces to DBFS. Specifically, support is added for lock, unlock, and test lock, at the full file level.

DBFS file locking enables applications to use familiar file locking mechanisms to coordinate and synchronize file access on files stored in DBFS file systems. This extends the usability of DBFS to include applications that use file locks to coordinate application behavior across multiple connections or users.

Related Topics

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

Oracle Multimedia PL/SQL API

Oracle Multimedia now provides a simplified API for managing images, audio, and video data stored in Oracle Database BLOBs. This API allows metadata extraction and image processing operations to be more spontaneously included in the SQL and PL/SQL applications.

The spontaneous and easy-to-use PL/SQL API for the multimedia data stored in BLOBs allows developers to easily include common operations, such as create an image thumbnail, crop an image, convert an image to web-friendly formats, extract metadata, and so on in their applications. A comprehensive set of database tools and features can work with this API. The multimedia data can be consolidated with other types of data in the database for easy display in reports and web UIs. These advantages enable the rapid development and deployment of In-Memory databases and other applications that include multimedia data.

Related Topics

- *Oracle® Multimedia Reference*

Public Cloud

- General

General

- Direct SQL*Net Access Over Oracle Cloud
- Controlling Outbound Database Link Options

Direct SQL*Net Access Over Oracle Cloud

This feature enables Oracle Cloud support for Oracle Database over SQL*Net.

Existing applications can now use Oracle Cloud without any code changes.

Related Topics

- *Oracle® Database Net Services Administrator's Guide*

Controlling Outbound Database Link Options

There are two new initialization parameters. Use the `OUTBOUND_DBLINK_PROTOCOLS` initialization parameter to specify the allowed network protocols for outbound database link connections. This can be used to restrict database links to use secure protocols. Use the `ALL_GLOBAL_DBLINKS` initialization parameter to allow or disallow global database links, which look up LDAP by default.

Man-in-the-middle attacks can only be addressed by securing the network protocol (choosing encryption). Database links can carry sensitive information (depending on the connect options) and should be secured. The `OUTBOUND_DBLINK_PROTOCOLS` initialization parameter specifies whether the database links communication using non-SSL protocols are allowed in the database. The `ALL_GLOBAL_DBLINKS` initialization parameter specifies whether LDAP lookup for database links are allowed for the database.

Related Topics

- *Oracle® Database Security Guide*

RAC and Grid

- Automatic Storage Management
- Ease-of-Use Improvements
- Fault-Tolerant Database Services
- High Availability and Reliability
- Oracle ASM Cluster File System (ACFS)

- Rapid Home Provisioning and Patch Management
- Super Scalable Cluster
- Super Scalable Oracle RAC

Automatic Storage Management

- Oracle Flex ASM Disk Group Quota Management
- Prioritized Rebalancing
- Extending Oracle Real Application Clusters (Oracle RAC) Configuration Support
- I/O Server

Oracle Flex ASM Disk Group Quota Management

Oracle Flex ASM disk groups provide a powerful set of capabilities that increase opportunities for consolidation across distinct databases that use ASM for storage management. Quota management allows ASM administrators to control storage consumption.

Increased consolidation at the storage level requires that the storage administrator, that is the ASM administrator, constrain storage consumption by a particular database.

Related Topics

- *Oracle® Automatic Storage Management Administrator's Guide*

Prioritized Rebalancing

Prioritized rebalancing is a feature of Oracle Flex ASM disk groups in which some file sets are rebalanced before other file sets. A file set can be considered as a collection of ASM files belonging to a database or a pluggable database (PDB). Oracle Flex ASM disk groups allow splitting of mirror copies of files in a file set and then new mirror copies are made during a rebalancing operation. Prioritized rebalancing enables you favor more critical databases or PDBs over less critical ones.

Oracle Flex ASM disk groups enable a greater level of consolidation, which enhances the need for prioritization of certain database operations. This prioritized rebalancing feature allows for that prioritization to take place.

Related Topics

- *Oracle® Automatic Storage Management Administrator's Guide*

Extending Oracle Real Application Clusters (Oracle RAC) Configuration Support

Using an Oracle RAC database across geographically dispersed data centers, or so that the data is mirrored between storage arrays in different locations within the same building or campus, qualifies an Oracle RAC installation as an extended Oracle RAC configuration. In prior releases, such a configuration had to be established manually and required postinstallation setup steps. Starting with Oracle Database 12*c* Release 2 (12.2), an understanding of sites is introduced, which allows the correct configuration of the stack as a whole to be set up during installation and optimizes the extended Oracle RAC operations.

A distinguishable extended Oracle RAC configuration allows for better reliability of extended Oracle RAC setups, either between sites or in smaller settings that require protection against storage array failures.

Related Topics

- *Oracle® Clusterware Administration and Deployment Guide*

I/O Server

This feature provides Oracle databases access to data in Oracle ASM disk groups without requiring physical “storage connectivity” to the underlying disks as currently required. Access to the data by databases is provided over a network similar to the way a network file system (NFS) server provides data to NFS clients.

This feature enables client clusters to access a disk group without requiring shared storage.

Related Topics

- *Oracle® Automatic Storage Management Administrator's Guide*

Ease-of-Use Improvements

- Cluster Resource Activity Log
- Cluster Verification Utility - Second Generation Command Line Output

Cluster Resource Activity Log

In addition to existing logs, which are mainly focused on reporting error situations, the cluster resource activity log contains information about resource activity across the cluster.

The cluster resource activity log can be used to track resource behavior in the cluster as well as to understand chain reactions triggered by the planned or unplanned relocation of individual resources.

Related Topics

- *Oracle® Autonomous Health Framework User's Guide*

Cluster Verification Utility - Second Generation Command Line Output

Cluster Verification Utility (CVU) assists in the installation and configuration of Oracle Clusterware and Oracle Real Application Clusters (Oracle RAC). CVU performs a range of tests, covering all intermediate stages during the installation and configuration of a complete Oracle RAC stack. Using the second generation command line output feature, CVU, for example, informs the user about the progress of each check and allows the user to specify an output format such as XML or HTML on request.

Report of detailed error information at various levels of installation and configuration of Oracle Clusterware and Oracle Real Application Clusters (Oracle RAC), and consolidated summary of verification checks enhances the user experience.

Related Topics

- *Oracle® Clusterware Administration and Deployment Guide*

Fault-Tolerant Database Services

- Switch Service Enhancement

Switch Service Enhancement

This feature separates the physical connection from the logical service to enable the service on a connection to be switched at request boundaries to another connection for services that are published at that database and instance. The following two consolidation use cases depend on the ability to switch services on a physical connection:

- Service and container switching for multitenant container database access when borrowing connections from pools serving multiple tenants
- Shared connection pools to scale database access, Oracle Database and third parties

These two use cases intersect and can also be used in standalone mode.

This feature supports Oracle Multitenant and custom consolidated databases, as well as reduces session overhead for all database usage. This feature eliminates an obstacle to consolidating to multitenant container databases due to loss of service features. It also eliminates an obstacle to consolidating to a few databases due to too many connection pools and connections serving these tenants. Lastly, this feature enables applications to use more database services for workload control without consuming additional connection pools. Customers are able to identify and prioritize workloads using services without over sizing the database connections.

Related Topics

- *Oracle® Database Administrator's Guide*

High Availability and Reliability

- VM Manager Agent for GI
- Shared Grid Naming Service High Availability
- Oracle Clusterware Resource Groups
- Reasoned What-If Command Evaluation (Why-If)
- Server Weight-Based Node Eviction
- Load-Aware Resource Placement
- Oracle Clusterware Application Clusters
- Oracle Grid Infrastructure Installation Support for Application Cluster Installation Type

VM Manager Agent for GI

Development of virtual machine (VM) management capabilities are using a dedicated VM Manager Agent for GI that interfaces with the Oracle Virtual Machine (Oracle VM) Manager, thus enabling the management of non-GI VM's as clusterware resources.

This feature helps to manage black box VM's as clusterware resources, without impacting their deployment or what is running within those VM's. The VM's do not require GI to be installed or configured within them.

Related Topics

- *Oracle® Clusterware Administration and Deployment Guide*

Shared Grid Naming Service High Availability

The Grid Naming Service (GNS) is linked to the corporate Domain Name Service (DNS) so that clients can resolve dynamic addresses and transparently connect to the cluster and the databases in the data center. Shared GNS provides dynamic name resolution for one cluster, or to advertise resolution for multiple clusters, so that a single GNS instance can perform name resolution for multiple registered clusters. Shared GNS high availability (HA) provides failure protection for the shared GNS instance using a secondary GNS instance.

Shared GNS high availability provides high availability of lookup and other services to the clients by running multiple instances of GNS with primary and secondary roles. All updates from the clients are serviced by the primary instance. The lookup queries are processed by both primary and secondary instances. Secondary instances act as backup for the primary instance. Secondary instances can be promoted to the primary role whenever an existing primary instance fails or is removed by a cluster administrator. In addition, shared GNS high availability provides fault tolerance by taking data backup on secondary instances using the zone transfer mechanism. Secondary instances receive a copy of data from the primary instance during installation. Thereafter, any update on the primary instance is replicated to the secondary instances.

Related Topics

- *Oracle® Clusterware Administration and Deployment Guide*

Oracle Clusterware Resource Groups

A resource group is a fundamental entity in application high availability modeling. It represents a container for a logically related group of resources.

Using resource groups, Oracle Clusterware provides an intuitive model to manage and monitor an application composed of various resources as a single composite entity.

Related Topics

- *Oracle® Clusterware Administration and Deployment Guide*

Reasoned What-If Command Evaluation (Why-If)

Oracle Clusterware 12*c* Release 1 (12.1) provided a set of evaluation commands and APIs to determine the impact of a certain operation before executing the operation. In this release, the reasoned What-If command evaluation feature provides the rationale behind the policy decisions and explains the entities involved, their attributes, and the criteria used to arrive at each of the potential actions.

Why-If command evaluations help applications, cluster, and system administrators involved in capacity planning and configuration management to set up and test resource management policies.

Related Topics

- *Oracle® Clusterware Administration and Deployment Guide*

Server Weight-Based Node Eviction

Server weight-based node eviction acts as a tie-breaker mechanism in situations where Oracle Clusterware needs to evict a particular node or a group of nodes from a cluster, in which all nodes represent an equal choice for eviction. In such cases, the server weight-based node eviction mechanism helps to identify the node or the group of nodes to be evicted based on additional information about the load on those servers. Two principle mechanisms, a system inherent automatic mechanism and a user input-based mechanism exist to provide respective guidance.

Using server weight-based node eviction allows for aligning the choice of which node gets evicted in case of certain failures in the cluster with business requirements, ensuring that the most important workload is kept alive for as long as possible, assuming an equal choice between servers.

Related Topics

- *Oracle® Clusterware Administration and Deployment Guide*

Load-Aware Resource Placement

Load-aware resource placement prevents overloading a server with more applications than the server is capable of running. The metrics used to determine whether an application can be started on a given server, either as part of the startup or as a result of a failover, are based on the anticipated resource consumption of the application as well as the capacity of the server in terms of CPU and memory.

Preventing overloading of a server by restricting new applications from starting, eases management in highly consolidated environments and prevents cascading failures.

Related Topics

- *Oracle® Clusterware Administration and Deployment Guide*

Oracle Clusterware Application Clusters

To support application-centric Oracle Clusterware deployments, Oracle Clusterware application clusters enables the deployment of Oracle Clusterware with only the components needed for operating non-database applications.

Oracle Clusterware application clusters provides an alternative solution to other light-weight clustering solutions to ease the adoption and configuration of Oracle Clusterware for the purpose of managing and deploying an Oracle Clusterware-based cluster by requesting user input only for those components that are essential for managing non-database applications.

Related Topics

- *Oracle® Grid Infrastructure Installation and Upgrade Guide for Linux*

Oracle Grid Infrastructure Installation Support for Application Cluster Installation Type

In Oracle Database 12*c* Release 2 (12.2), installing Oracle Grid Infrastructure includes the option of deploying Application Cluster, which is a generic form of Oracle Grid Infrastructure that makes applications, including non-Oracle applications, highly available.

Related Topics

- *Oracle® Grid Infrastructure Installation and Upgrade Guide for Linux*

Oracle ASM Cluster File System (ACFS)

Oracle Automatic Storage Management Cluster File System (Oracle ACFS) Snapshot Enhancements

The Oracle ACFS snapshot enhancements comprise the following:

- An Oracle ACFS snapshot quota enables individual Oracle ACFS RW snapshots in Oracle Database 12*c* Release 2 (12.2) to limit the new storage that can be allocated for a given snapshot. This provides a quota on space usage for each snapshot.
- Oracle ACFS snapshot remaster converts a given snapshot in an Oracle ACFS snapshot hierarchy into the Oracle ACFS primary file system. The previous Oracle ACFS primary file system and all previous snapshots are deleted.
- Oracle ACFS snapshot rename: This renames an existing ACFS snapshot.

- Oracle ACFS snapshot delete with open files: It is a force snapshot delete that enables a snapshot with open file references to be removed from an Oracle ACFS.
- Snap dup: Snap dup(licate) creates commands to encode the differences between two point-in-time snapshots into a data stream. This snap dup operation is offered in Oracle Database 12*c* Release 2 (12.2) as an independent Oracle ACFS snapshot command.
- Snap dup apply command: Use the apply command to apply the encoded snapshot to a standby Oracle ACFS snapshot.

These Oracle ACFS Snapshot features significantly enhance the snapshot functionality and simplify file system snapshot operations.

Related Topics

- *Oracle® Automatic Storage Management Administrator's Guide*

Oracle ACFS System Defragger

Oracle ACFS provides a file defragging tool in this release. You can run the defragging tool on a live Oracle ACFS file system using the `acfsutil` command. In addition, the automatic database file defragger is updated for scalability in Oracle Database 12*c* Release 2 (12.2) with support for concurrent, multiple node database file defragmentation.

Oracle ACFS rarely requires defragging of non-database or general purpose application files due to careful allocation and coalescing of free space. However, for rare occasions when you must defrag, Oracle ACFS provides a defragger tool to help you achieve optimal file system performance.

Related Topics

- *Oracle® Automatic Storage Management Administrator's Guide*

4K Sectors and Metadata Enhancements for Oracle ACFS

Oracle ACFS supports I/O requests in multiples of 4096-byte logical sector sizes as well as providing continued support for 512-byte logical sector size I/O requests. Oracle ACFS also adds support for creating Oracle ACFS file systems with 4096-byte (4KB) sized metadata structures.

A new Oracle ACFS make file system switch enables you to format an Oracle ACFS file system with 4096-byte metadata structures composed of multiple 512-byte logical sectors on Oracle Automatic Storage Management Dynamic Volume Manager (Oracle ADVM) volumes. Oracle ACFS metadata I/O is also in 4K multiples.

This feature provides better alignment of Oracle ACFS metadata when an Oracle Automatic Storage Management disk group is composed of all new advanced format drives emulating 512-byte sectors.

Related Topics

- *Oracle® Automatic Storage Management Administrator's Guide*

Oracle ACFS Metadata Collection Enhancements

Oracle ACFS Metadata Collector tool can be used to copy select Oracle ACFS metadata structures into a new, separate output file that you can use for subsequent analysis and diagnostics.

This is a useful tool at customer sites, because it enables you to collect Oracle ACFS metadata for diagnosis without requiring the Oracle ACFS file system to be offline. The Metadata Collector tool is not a replacement for the File System Checker (`fsck`) tool.

The Oracle ACFS Metadata Collector tool can be a useful tool at customer sites, because it enables you to collect Oracle ACFS metadata for diagnosis without requiring you to take the Oracle ACFS file system off-line. The Oracle ACFS Metadata Collector tool is not a replacement for the file system checker (`fsck`) tool.

Related Topics

- *Oracle® Automatic Storage Management Administrator's Guide*

Oracle ACFS Plug-ins for File Content

This feature provides support for an additional metric type, `Metric2`, for Oracle ACFS plug-ins. This enhancement enables you to retrieve file content metrics.

This feature extends the Oracle ACFS plug-in solution to support file content collection and supports both polling and interval-based capture. In the polling model, the application polls for file content changes and Oracle ACFS returns summary messages that identify which files have changed and the location of the changes within each file. In the interval-based model, Oracle ACFS posts messages containing file content changes on a selected time interval basis. The Oracle ACFS file system alerts a waiting application API call, and the application then collects a set of records that identify which files changed and the location of the changes within each file.

This feature enables you to extend Oracle ACFS to provide additional tracking data for application use.

Related Topics

- *Oracle® Automatic Storage Management Administrator's Guide*

Oracle ACFS Loopback Device

An Oracle ACFS loopback device is an operating system pseudo-device that enables an Oracle ACFS file access as a block device. This functionality works with Oracle Virtual Machines in support of Oracle Virtual Machine images, templates, and virtual disks that are created in Oracle ACFS file systems and presented through Oracle ACFS loopback devices.

The Oracle ACFS loopback device enables high performance block storage I/O between the client and the server components.

Related Topics

- *Oracle® Automatic Storage Management Administrator's Guide*

Oracle ACFS API Access for Snapshots, File Tags, Plug-in Metrics

This feature creates a full-featured C library that supports the external clients of Oracle ACFS functionality including Oracle ACFS snapshots, Oracle ACFS file tags, and Oracle ACFS plug-in metrics. Third party applications and Oracle Fusion Middleware are an example of target consumers of this library.

In addition, a generic Oracle ACFS library is created in Oracle Database that effectively serves as a wrapper for the platform-specific Oracle ACFS C library. This library supports generic programmatic access to Oracle ACFS, avoiding platform-specific details and dynamic library issues. An example of a consumer of this library is Oracle Database PDB cloning using Oracle ACFS snapshots.

The feature allows programmatic access to Oracle ACFS, avoiding platform-specific details and dynamic library issues for third party applications.

Related Topics

- *Oracle® Automatic Storage Management Administrator's Guide*

Oracle ACFS Compression Enhancements

The Oracle ACFS compression project improves both Oracle Database and general purpose file storage. However, a specific focus of this project for Oracle Database 12*c* Release 2 (12.2) is to deliver data compression that is optimized for data files, archive logs, data pump files, and Oracle Recovery Manager (RMAN).

This feature provides a cost-effective way to significantly reduce the disk storage required in a variety of customer environments which use Oracle ACFS files.

Related Topics

- *Oracle® Automatic Storage Management Administrator's Guide*

Oracle Snapshot-Based Replication Enhancements

This feature captures Oracle ACFS file changes for a given interval of time as a set of differences between successive snapshots of the primary Oracle ACFS file system. This is done instead of populating and transporting multiple, node-local replication logs that contain records of all Oracle ACFS file changes for a given interval of time. Oracle ACFS snapshot-based replication relies on the following components:

- A new Oracle Database 12*c* Release 2 (12.2) snap clone create command to encode the differences between two point-in-time snapshots into a data stream.
- Oracle ACFS snapshot-based replication uses SSH to transmit the encoded snapshot clone stream to a standby site. It uses the `SNAP CLONE APPLY` command to apply the encoded snapshot clone stream on a given standby site.

Snapshot-based replication significantly improves replication performance.

Related Topics

- *Oracle® Automatic Storage Management Administrator's Guide*

Oracle ACFS Auto-Resize Enhancements

The auto-resize feature enables you to specify an increment by which an Oracle ACFS file system automatically increases in size if the amount of available file system free space falls below this amount. The file system expands by an amount equal to this increment. You can also specify a maximum size to which a file system automatically resizes itself. This enables you to limit the file system expansions and to prevent runaway storage consumption. This feature effectively serves as an Oracle ACFS file system quota.

This feature enables you to specify file system size quotas and have the file system automatically increase in size, if required.

Related Topics

- *Oracle® Automatic Storage Management Administrator's Guide*

Oracle ACFS Sparse Files Enhancements

A sparse file uses file system space more efficiently when the file is mostly empty. This is achieved by writing brief information describing the representation of empty blocks on the storage device instead of the actual insertion of zeros into empty (unwritten) but allocated space on the storage device. The full block size is written to the storage device as the actual size only when the block contains actual (non-empty) data. The storage space savings associated with sparse files is obvious, but an even more motivating factor is improved performance.

Oracle ACFS sparse files greatly benefit NFS client write operations which are commonly received out of order by the NFS server and the associated Oracle ACFS file system. Sparse files also benefit the creation and management of Oracle Virtual Machine images. The sparse file approach reduces the overhead for the time required to create an image file because there is no requirement to allocate and zero fill unused file storage. This feature also reduces the overhead for space consumption because there is no requirement to allocate storage for unused, empty space.

Related Topics

- *Oracle® Automatic Storage Management Administrator's Guide*

Oracle ACFS Metadata Acceleration

The Oracle ACFS metadata acceleration feature enables you to optionally specify a storage accelerator volume for association with the file system during file system creation. This storage accelerator volume comprises of solid-state drives (SSDs) or other high-performance disks. Oracle ACFS metadata is stored on the accelerator volume. Examples of this metadata include the Oracle ACFS volume log, indirect nodes, also known as extent nodes, space management structures, and other Oracle ACFS metadata. The accelerator feature significantly improves file access performance for Oracle ACFS files that are stored on the Oracle ACFS primary file system or an Oracle ACFS snapshot.

This feature significantly improves Oracle ACFS performance.

Related Topics

- *Oracle® Automatic Storage Management Administrator's Guide*

Oracle ACFS NAS Maximum Availability eXtensions

Provides Oracle ACFS Server Message Block (SMB) protocol clients with a transparent service for failover and other NAS enhancements.

This feature extends the existing Oracle ACFS highly available network file access feature to SMB clients. This feature also provides enhancements to NFSv4 support.

Related Topics

- *Oracle® Automatic Storage Management Administrator's Guide*

Rapid Home Provisioning and Patch Management

- Rapid Home Provisioning

Rapid Home Provisioning

Rapid Home Provisioning provides the following new features:

- Provisioning, patching, and upgrading of Oracle Grid Infrastructure releases 12.2, 12.1.0.2, and 11.2.0.4.
- Provisioning of single-instance Oracle databases on various versions of client clusters.
- Upgrade of Oracle Database.

Rapid Home Provisioning facilitates automatic remote provisioning, patching, and upgrade of Oracle Database and Oracle Grid Infrastructure using gold images. This supports deployment of Oracle Grid Infrastructure clusters and Oracle Database in a cloud environment.

Related Topics

- *Oracle® Clusterware Administration and Deployment Guide*

Super Scalable Cluster

- Support for IPv6 Based IP Addresses for the Oracle Cluster Interconnect

Support for IPv6 Based IP Addresses for the Oracle Cluster Interconnect

You can configure cluster nodes to use either IPv4 or IPv6 based IP addresses on a private network, and more than one private network can be used for a cluster.

IPv6 support for the private interconnect completes the IPv6 enhancement effort for Oracle Real Application Clusters (Oracle RAC). Oracle RAC has supported IPv6 for the public network since Oracle Database 12*c* Release 1 (12.1). Unlike for the public network, the IPv6 support for the private interconnect is restricted to either support IPv4 or IPv6 based IP addresses, but not both versions simultaneously.

Related Topics

- *Oracle® Clusterware Administration and Deployment Guide*

Super Scalable Oracle RAC

- Oracle Real Application Clusters (Oracle RAC) Reader Nodes
- Service-Oriented Buffer Cache Access Optimization

Oracle Real Application Clusters (Oracle RAC) Reader Nodes

This feature facilitates Oracle Flex Cluster architecture in such a way that it allocates a set of read/write instances running OLTP workloads and a set of read-only database instances across the Hub Nodes and Leaf Nodes in the cluster. In this architecture, the updates made on the read-write instances are immediately propagated to the read-only instances on the Leaf Nodes, where they can be used for online reporting or instant queries.

The separation of OLTP and read operations in an Oracle Flex Cluster architecture allows for fast reconfiguration of read-only instances joining and leaving the cluster, as well as efficient updates to the buffer cache on those instances.

Related Topics

- *Oracle® Clusterware Administration and Deployment Guide*

Service-Oriented Buffer Cache Access Optimization

Cluster Managed Services are used to allocate workloads across various Oracle Real Application Clusters (Oracle RAC) database instances running in a cluster. The database objects cached in the buffer caches of the respective database instances are accessed through these services. This feature enables Oracle RAC to cache the instances with data blocks for objects accessed through a service, and thus improves the access time.

Data-dependent caching leads to consistent response times when accessing data across Oracle RAC database instances running in the cluster.

Related Topics

- *Oracle® Database Performance Tuning Guide*

Security

- Encryption
- Enforcing Application Security in the Database
- Improving Security Manageability, Administration, and Integration
- Improving Security Posture of the Database
- Improving User Authentication and Management
- Modernizing Network Authentication and Encryption

Encryption

- TDE Tablespace Live Conversion
- Fully Encrypted Database
- Support for ARIA, SEED, and GOST Encryption Algorithms in TDE
- TDE Tablespace Offline Conversion

TDE Tablespace Live Conversion

You can now encrypt, decrypt, and rekey existing tablespaces with Transparent Data Encryption (TDE) tablespace live conversion. A TDE tablespace can be easily deployed, performing the initial encryption that migrates to an encrypted tablespace with zero downtime. This feature also enables automated deep rotation of data encryption keys used by TDE tablespace encryption in the background with zero downtime.

This feature performs initial cryptographic migration for TDE tablespace encryption on the tablespace data in the background so that the tablespace can continue servicing SQL statements that insert, delete, select, modify, and so on.

Related Topics

- *Oracle® Database Advanced Security Guide*

Fully Encrypted Database

Transparent Data Encryption (TDE) tablespace encryption is applied to database internals including `SYSTEM`, `SYSAUX`, and `UNDO`.

Expanding internal usage of TDE tablespace encryption to include `SYSTEM`, `SYSAUX`, and `UNDO` further strengthens the security of Oracle Database.

Related Topics

- *Oracle® Database Advanced Security Guide*

Support for ARIA, SEED, and GOST Encryption Algorithms in TDE

This feature provides advanced security Transparent Data Encryption (TDE) support for these encryption algorithms:

- SEED (Korea Information Security Agency (KISA) for South Korea
- ARIA (Academia, Research Institute, and Agency) for South Korea
- GOST (GOsudarstvennyy STandart) for Russia

This feature includes support for both encryption and hashing algorithms and is available for use with data-at-rest encryption.

Certain countries require the use of their specific national and government standards for encryption. Deployment of TDE database encryption in these countries can proceed now that these national and government algorithms are supported.

Related Topics

- *Oracle® Database Advanced Security Guide*

TDE Tablespace Offline Conversion

This release introduces new SQL commands to encrypt tablespace files in place with no storage overhead. You can do this on multiple instances across multiple cores. Using this feature requires downtime, because you must take the tablespace temporarily offline. With Data Guard configurations, you can either encrypt the physical standby first and switchover, or encrypt the primary database, one tablespace at a time.

This feature provides fast offline conversion of existing clear data to TDE encrypted tablespaces.

Related Topics

- *Oracle® Database Advanced Security Guide*

Enforcing Application Security in the Database

RAS Session Privilege Scoping

This new capability limits the scope of Real Application Security (RAS) users to execute privileged user commands for a limited group of RAS users instead of the entire database.

In cloud or consolidated environments, this new feature limits the ability to execute privileged user commands of RAS users to their own group.

Related Topics

- *Oracle® Database Real Application Security Administrator's and Developer's Guide*

RAS Column Privilege Enhancements

Real Application Security (RAS) users can now be authorized specifically to insert, update or delete data in protected columns. Additional performance enhancements were made for `SELECT` statements using RAS.

A much finer degree of data management is possible through this new feature in addition to the `SELECT` performance enhancements.

Related Topics

- *Oracle® Database Real Application Security Administrator's and Developer's Guide*

RAS Schema Level Policy Administration

A schema-based Real Application Security (RAS) policy administrator can now create, modify and delete security policies within the schema instead of requiring a system-wide RAS policy administrator to make the changes. This allows a clear separation of duties between different applications.

In a cloud or consolidated database environment, RAS administration is required on a schema by schema basis without requiring a system-wide RAS administrator to make security policy changes on any schema. This improves the security profile of each application.

Related Topics

- *Oracle® Database Real Application Security Administrator's and Developer's Guide*

RAS Integration with OLS

Oracle Label Security (OLS) which provides an explicit data label and user security label is integrated with Oracle Real Application Security in administration, policy enforcement and user context.

Combining RAS implicit data realms with the Oracle Label Security explicit data labeled security provides strong security capabilities to allow formerly separate databases to consolidate into a single database and yet retain the separation of data that was available before the consolidation.

Related Topics

- *Oracle® Database Real Application Security Administrator's and Developer's Guide*

Improving Security Manageability, Administration, and Integration

- Oracle Virtual Private Database Predicate Audit
- Oracle Database Vault Policy
- Oracle Database Vault Simulation Mode Protection
- Oracle Database Vault Common Realms and Command Rules for Oracle Multitenant
- Privilege Analysis Enhancements
- Privilege Analysis Results Comparison
- Redaction: Different Data Redaction Policy Expressions
- Redaction: New Functions Allowed in Data Redaction Policy Expressions
- Redaction: Additional Data Redaction Transformations
- Automatic KDC Discovery When Configuring OCI Clients
- Automatic Provisioning of Kerberos Keytab for Oracle Databases
- Role-Based Conditional Auditing
- Inherit Remote Privileges

Oracle Virtual Private Database Predicate Audit

Oracle Database audit trail now includes the predicate or `WHERE` clause added by Oracle Virtual Private Database (VPD) policies.

By providing the VPD predicate in the audit trail, both unified and traditional auditing provide richer forensic information.

Related Topics

- *Oracle® Database Security Guide*

Oracle Database Vault Policy

Oracle Database Vault policy enables customers to group multiple Oracle Database Vault protections into a single entity called policy. Specifically, this allows the grouping of realms and command rules under a single Oracle Database Vault policy that can be managed as a single unit for enabling, disabling and authorizing. This ensures that managing Oracle Database Vault protections for applications are simple and more effective. It also enables customers to create and easily manage multiple Oracle Database Vault policies for the same application to accommodate different operational modes such as production mode and patching mode. Oracle Database Vault policy authorization delegation is also allowed.

Improved manageability is critical as additional security controls are created and managed within Oracle Database Vault. Separation of duty is enhanced by the delegation of policy management. This feature makes it easier for application security owners to manage Oracle Database Vault policies.

Related Topics

- *Oracle® Database Vault Administrator's Guide*

Oracle Database Vault Simulation Mode Protection

Oracle Database Vault simulation mode protection allows customers to test their Oracle Database Vault realms and command rules protections using soft enforcement. This only records Oracle Database Vault violations in a training log file instead of blocking the operation and potentially disrupting business processes.

Simulation mode allows adding testing to the production environment to increase confidence of new and updated security controls without disrupting operations. Application certification time on Oracle Database Vault is drastically reduced since the application can run continuously while the simulation mode log records any violations. New security controls can be rapidly rolled out into production with minimal impact to production environments.

Related Topics

- *Oracle® Database Vault Administrator's Guide*

Oracle Database Vault Common Realms and Command Rules for Oracle Multitenant

Oracle Database Vault common realms and command rules for Oracle Multitenant databases allows common objects of a multitenant container database to be protected by common realms and command rules. These common Oracle Database Vault controls protect common objects in all pluggable databases (PDB) where Oracle Database Vault is enabled. Since mixed mode is allowed where some PDBs have Oracle Database Vault enabled and some do not, the common realms and command rules protections are effective only in those PDBs where Oracle Database Vault is enabled. There is a way to enforce common realms and command rules protections across all PDBs by turning on Oracle Database Vault strict mode. Oracle Database Vault strict mode requires that every PDB in the multitenant container database (CDB) has Oracle Database Vault enabled.

Multitenant container database architectures deploying the same applications across multiple PDBs can centrally manage their Oracle Database Vault protections for the application common objects instead of locally in each PDB. This improved management capability of common objects not only saves time, but also lowers the risk of having different security profiles on similar PDBs.

Related Topics

- *Oracle® Database Vault Administrator's Guide*

Privilege Analysis Enhancements

This feature enhances privilege analysis to capture privileges for the following:

- Compile PL/SQL packages, procedures, and functions in addition to the current capability of capturing execution privileges.
- Run Oracle Java programs.
- Code-based access control (CBAC) roles and secure application roles.

New tables are introduced to help customers easily identify unused granted privileges.

These enhancements help customers make more informed decisions on what privileges to keep and what privileges to revoke from a user or a role.

Related Topics

- *Oracle® Database Vault Administrator's Guide*

Privilege Analysis Results Comparison

Privilege analysis policies can be run multiple times and generate results for each run separately. Results can be compared to see if there are any differences in privilege usage.

This feature closely tracks the real-life usage of privilege analysis and increases usability.

Related Topics

- *Oracle® Database Vault Administrator's Guide*

Redaction: Different Data Redaction Policy Expressions

Users can now associate different Oracle Data Redaction policy expressions for different columns within the same table or view.

This feature provides greater flexibility for Oracle Data Redaction customers. Now each column can be redacted according to different runtime conditions.

Related Topics

- *Oracle® Database Advanced Security Guide*

Redaction: New Functions Allowed in Data Redaction Policy Expressions

Additional database functions can be used in the Data Redaction policy expressions. Standard database substring functions now are supported.

Supporting new functions, such as substring, in the Data Redaction policy expressions enables more comprehensive policies to incorporate further runtime conditions.

Related Topics

- *Oracle® Database Advanced Security Guide*

Redaction: Additional Data Redaction Transformations

New transformations are supported including the ability to use `REGEXP` redaction on `CLOB`s and `NCLOB`s and the ability to redact to `NULL` values.

These new transformations expand capabilities and potential use cases of Data Redaction. In earlier database versions, you could not redact to `NULL`, and for `LOB` data, you only could redact the entire value to a fixed constant.

Related Topics

- *Oracle® Database Advanced Security Guide*

Automatic KDC Discovery When Configuring OCI Clients

Oracle Call Interface (OCI) clients can now be configured to automatically look up the Kerberos Key Distribution Center (KDC) and realm using Domain Name Service (DNS).

These enhancements result in fewer manual steps required by an administrator to provision Oracle Database clients, reducing setup time and decreasing the risk of errors.

Related Topics

- *Oracle® Database Security Guide*

Automatic Provisioning of Kerberos Keytab for Oracle Databases

The new `okcreate` utility automates the registering of an Oracle database as a Kerberos service principal, creating a keytab for it, and securely copying the keytab to the database for use in Kerberos authentication.

This enhancement decreases the number of manual steps required by administrators to provision Kerberos authentication for Oracle databases, reducing setup time and decreasing the risk for errors.

Related Topics

- *Oracle® Database Security Guide*

Role-Based Conditional Auditing

Role-based conditional auditing provides the ability to define unified audit policies that conditionally audit users based on a role in addition to the current capability to audit by users.

This feature enables more powerful policy-based conditional auditing by using database roles as the condition for auditing. For example, auditing for new users with the DBA role would begin automatically when they are granted the role.

Related Topics

- *Oracle® Database Security Guide*

Inherit Remote Privileges

A new privilege `INHERIT REMOTE PRIVILEGES` enables you to use a connected user database link from within a definer's rights procedure. Without this privilege, the definer's rights procedure cannot connect by way of the connected user database link.

A connected user on a database unknowingly should not allow a definer's rights procedure owner on that database to perform database operations on a remote database, logging in as that connected user. This new privilege must be explicitly granted to enable such operations.

Related Topics

- *Oracle® Database Security Guide*

Improving Security Posture of the Database

- SYSRAC - Separation of Duty for Administering Real Application Clusters
- Transparent Sensitive Data Protection Feature Integration
- Requiring Strong Password Verifiers by Default

SYSRAC - Separation of Duty for Administering Real Application Clusters

`SYSRAC` is a new role for Oracle Real Application Clusters (Oracle RAC) management. `SYSRAC`, like `SYSDG`, `SYSBACKUP,` and `SYSKM`, helps enforce separation of duties and reduce reliance on the use of `SYSDBA` on production systems. This administrative privilege is the default mode for connecting to the database by the clusterware agent on behalf of the Oracle RAC utilities such as `srvctl`.

Separation of duty for administering Oracle RAC (`SYSRAC`) improves the security of Oracle Database and reduces reliance on `SYSDBA` privilege to do Oracle RAC administration. `SYSRAC` comes after the introduction of `SYSKM`, `SYSDG`, and `SYSBACKUP` in Oracle Database 12*c* Release 1 (12.1) to provide separation of duty for key management, Oracle Data Guard and RMAN backup, respectively. With these administrative privileges in place, customers have a choice to implement separation of duty in their environment in a manner that fits their business needs. For example, customers can create a named administrative account and grant only the administrative privileges needed such as `SYSRAC` and `SYSDG` to manage both Oracle RAC and Oracle Data Guard configurations.

Related Topics

- *Oracle® Database Security Guide*

Transparent Sensitive Data Protection Feature Integration

This enhancement simplifies and expands transparent sensitive data protection (TSDP) by supporting additional database features such as Oracle Advanced Security Transparent Data Encryption and unified auditing.

This feature streamlines the application of additional controls such as encryption and auditing to sensitive data stored inside applications.

Related Topics

- *Oracle® Database Security Guide*

Requiring Strong Password Verifiers by Default

Oracle Database 11*g* and Oracle Database 12*c* introduced stronger password verifiers. These newer verifiers use salted hashes, modern SHA-1 and SHA-2 hashing algorithms, and mixed-case passwords. The `ALLOWED_LOGON_VERSION_SERVER` parameter is set by default to require strong verifiers for user logins.

Using strong password verifiers improves user authentication security by making it more difficult for attackers to find user passwords by exhaustive searches.

Related Topics

- *Oracle® Database Security Guide*

Improving User Authentication and Management

- Automatic Locking of Inactive User Accounts

Automatic Locking of Inactive User Accounts

Within a user profile, the `INACTIVE_ACCOUNT_TIME` parameter controls the maximum time that an account can remain unused. The account is automatically locked if a log in does not occur in the specified number of days.

Locking inactive user accounts prevents attackers from using them to gain access to the database.

Related Topics

- *Oracle® Database Security Guide*

Modernizing Network Authentication and Encryption

- Kerberos-Based Authentication for Direct NFS

Kerberos-Based Authentication for Direct NFS

Oracle Database now supports Kerberos implementation with Direct NFS communication.

This feature solves the problem of authentication, message integrity, and optional encryption over unsecured networks for data exchange between Oracle Database and NFS servers using Direct NFS protocols.

Related Topics

- *Oracle® Grid Infrastructure Installation and Upgrade Guide for Linux*

Spatial and Graph

- Graph Features
- Spatial Features

Graph Features

- RDF Semantic Graph Support for W3C Standards
- Property Graph Support

RDF Semantic Graph Support for W3C Standards

The RDF Semantic Graph feature in Oracle Spatial and Graph supports a new World Wide Web Consortium (W3C) standard. Oracle SQL now has W3C SPARQL Update support. This gives Oracle Database full W3C SPARQL 1.1 query, update, and federated query support. SPARQL Update is an update language for RDF graphs. SPARQL Update enables a single SPARQL request to perform a number of operations:

- Create and drop graphs
- Copy, move, or add data from one graph to another, with or without removing the data from the source or destination graph
- Insert, delete, and update (a delete plus an insert) data in one or more RDF graphs

Application developers can now be more productive with a single SPARQL request to the Oracle Database RDF graph store.

Related Topics

- *Oracle® Spatial and Graph RDF Semantic Graph Developer's Guide*

Property Graph Support

New property graph support in Oracle Spatial and Graph exploits Oracle Database to deliver industry leading graph performance, scalability, and security. Graphs of any size, up to the limits of Oracle Database, are supported with unlimited numbers of vertices (nodes), edges (relationships), properties and relationship types. Data loading and graph operations are parallelized. Application development is facilitated with Java APIs based on the open TinkerPop graph interfaces and support for scripting languages, including Groovy and Python. Text pattern and faceted search capabilities are enabled with Oracle Text and Apache Lucene. Users can optionally enable support for SOLRcloud, if desired. A suite of 35 built-in graph analytics includes graph traversal, connectivity, path, page ranking, and clustering coefficient analysis to evaluate graphs for cohesion, centrality, ranking, and roles.

Property graphs enable discovery and analysis of data relationships and properties. Oracle Spatial and Graph property graph provides a unique combination of limitless scalability, secure storage, flexible deployment, ease of application development, and rich property graph analytics for a range of social media, knowledge network, and national security applications.

Related Topics

- *Oracle® Spatial and Graph Property Graph Developer's Guide*

Spatial Features

- Extending JSON Support in Oracle Database With Spatial Operations
- Location Data Enrichment Services
- In-Database Location Tracking Data Model and Services
- Network Data Model Support for Oracle Coherence
- Network Data Model Feature Editing APIs
- Time-Based Routing Engine Enhancements
- Support for Hash and List Partitioning of Spatial Indexes
- GeoRaster Map Algebra and Image Processing Enhancements
- Spatial Map Visualization
- Open Geospatial Consortium Web Map Service and Web Map Tile Service Support
- Geospatial Consortium Web Coverage Service 2.0 Support
- Oracle Workspace Manager Enhancements

Extending JSON Support in Oracle Database With Spatial Operations

Oracle Spatial and Graph includes support for the GeoJSON format allowing users to perform spatial operations. This includes the ability to retrieve `SDO_GEOMETRY` data as a JSON object, to get the results of a spatial query as a JSON document, and to store, index and query GeoJSON documents.

JSON and GeoJSON are among the most common data exchange formats for web and mobile applications. Native support for these formats simplifies application development and the creation of REST data services.

Related Topics

- *Oracle® Spatial and Graph Developer's Guide*

Location Data Enrichment Services

Oracle Spatial and Graph includes a place name data set and hierarchy and service APIs to enrich commonly used textual location data such as place names, postal codes, addresses and partial addresses, and latitude and longitude information. Location information can be associated with information in the GeoNames geographical database now included with the product. In addition, users can add hierarchy data to GeoNames or use a custom geographic database of their choosing.

Location tags are extracted from text data and are matched with well known place names using Oracle Text and enhanced with other geographic information associated with the well know place names.

The results can be stored as additional attributes with the original data.

This feature allows users to automatically process less structured geographic and location data so that the information can be categorized, compared, filtered, and associated with other data. For example, data with postal codes only can be enriched to include city, county, state, and country allowing it to be joined or analyzed with other data sets that can have state level information. This is especially useful when comparing Big Data results with structured information in operational systems and data warehouses.

In-Database Location Tracking Data Model and Services

Oracle Spatial and Graph includes PL/SQL and Java APIs to enable users to define geographic zones, paths, and associated objects. Oracle Advance Queuing is used to manage alerts as objects enter or leave predefined zones or deviate from planned or approved paths.

Every second, Oracle Database can track tens of thousands to millions of moving objects interacting with thousands to hundreds of thousands of regions. This scale enables analysis of mobile assets such as cell phones, GPS devices on fleets, automobiles, and individuals, at a massive scale.

Related Topics

- *Oracle® Spatial and Graph Developer's Guide*

Network Data Model Support for Oracle Coherence

A distributed cache framework that supports Oracle Coherence distributed cache system is added in this release to Oracle Spatial and Graph Network Data Model.

Support for distributed caches, in conjunction with improved path computation heuristics, improves the performance of network and route calculations. This distributed cache support also allows larger network segments to be analyzed in memory.

Network Data Model Feature Editing APIs

Java and PL/SQL network feature editing APIs are added to Oracle Spatial and Graph Network Data Model that allows you to edit features, define and enforce rules regarding what editing can be performed, and create both in-memory and in-database networks.

These APIs allow PL/SQL, Java- and HTML5-based applications to edit network features, simplifying network-based operations and application development.

Time-Based Routing Engine Enhancements

The routing engine in Oracle Spatial and Graph can now use time-based data including time zones, traffic patterns, and time-based constraints (for example, no turns from 6 AM to 9 AM).

This means the routes generated by the routing engine can be more accurate and detailed, and can permit the use of the routing engine in applications that previously required custom development.

Support for Hash and List Partitioning of Spatial Indexes

Spatial indexes can now be partitioned based on hash and list mechanisms in addition to the existing support for range partitioning.

This feature enables the use of spatial analysis in applications that currently use or require hash and list partitioned indexes. Customers are no longer limited to range partitioning of tables including spatial data.

Related Topics

- *Oracle® Spatial and Graph Developer's Guide*

GeoRaster Map Algebra and Image Processing Enhancements

Enhancements to Oracle Spatial and Graph GeoRaster allow the use of an unlimited number of image layers (sets of imagery, sets of sensor readings, multi- and hyper-spectral data) when performing analysis and processing operations. They also support finer-grained statistical analysis, below the cell or pixel level, required for precise calculations used in climate modeling, digital elevation model (DEM) processing, and analysis of low-resolution imagery. A new image processing package provides more image transformations and processing including linear stretching, piece-wise stretching, normalization, equalization, color ramping, and advanced warping and affine transformations. Also added, syntax and algorithms for color balancing when appending layers and building physical and virtual mosaics.

These enhancements enable powerful in-database raster analysis and image processing on extremely large (terabyte and petabyte scale) sets of raster imagery without requiring movement of data to mid-tier servers or client applications.

Related Topics

- *Oracle® Spatial and Graph GeoRaster Developer's Guide*

Spatial Map Visualization

The map visualization feature is an HTML5-based mash up component of Oracle Spatial and Graph that allows developers who are familiar with SQL and JavaScript to incorporate a wide variety of map styles and spatial analysis into business applications.

Map visualization is a Java EE (Java Platform, Enterprise Edition) application that can be deployed to supported Java EE containers, such as WebLogic Server (WLS), Apache Tomcat, and JBoss. It includes:

- A core rendering engine (Java library) named SDOVIS that performs cartographic rendering. A servlet is provided to expose the rendering functions to web applications.
- A suite of application programming interfaces (APIs) that allow programmable access to the map visualization features. These APIs include both REST and an AJAX-based JavaScript API.

- A graphical map builder tool that enables you to create map symbols, define spatial data rendering rules, and create and edit map objects.
- A map cache and FOI (feature of interest) servers that facilitate the development of interactive geospatial web applications.

The map visualization feature supports two-dimensional vector geometries stored in Oracle Spatial and Graph, as well as GeoRaster data and data in the Oracle Spatial and Graph topology and network data models. It is also an Open Geospatial Consortium (OGC) compliant Web Map Service (WMS) and Web Map Tile Service (WMTS) server.

The map visualization feature in Oracle Spatial and Graph enhances the spatial data management capabilities by providing a generic web-based means of delivering and viewing any spatial and geographic data in the database. This creates enormous potential for understanding and capturing the geographic components of any business, by unlocking the enterprise information in many corporate warehouses and making it available to basic mapping applications. For instance, business applications such as Field Service, Transportation and Logistics, Asset Lifecycle Management, Human Resources, and Real Estate can now render and visualize the massive amount of data they control if there is a geographic component such as an address tied to the data.

Developers of location-based services, data publishers in state and local government, and architects of web services and more traditional applications can all easily integrate these mapping features into their web-based solutions.

Open Geospatial Consortium Web Map Service and Web Map Tile Service Support

Oracle Spatial and Graph includes support for the Open Geospatial Consortium (OGC) Web Map Service (WMS 1.3.0) and Web Map Tile Service (WMTS 1.0.0). These services provide simple HTTP REST interfaces to online map services and tile servers, respectively.

Through these standards-based interfaces, applications using Oracle Spatial and Graph map visualization can make map data in Oracle Database available to other applications through OGC Web Map Services and Map Tile Services. In addition, they enable geographic data and query results from Oracle Spatial and Graph to be displayed with maps and map tiles from any OGC-compliant service.

Related Topics

- *Oracle® Spatial and Graph Developer's Guide*

Geospatial Consortium Web Coverage Service 2.0 Support

The Open Geospatial Consortium, Inc. (OGC) Web Coverage Service (WCS) defines web services for sharing satellite imagery, sensor data, digital elevation models, climate/ocean data, and other raster data sets, grids and meshes. These data are referred to as geospatial coverages. Oracle Spatial and Graph WCS support implements the operations described in the OGC WCS 2.0 specification to enable interoperable access to all geography markup language (GML) and ISO coverage types.

Through the implementation of this standard, raster data stored in Oracle Spatial and Graph can be made accessible to a wide range of applications on the web. In addition, other coverages published on the Web using this WCS standard can be easily discovered and incorporated into Oracle Spatial and Graph applications. This feature offers a wide range of raster data sets to complete and enrich applications.

Related Topics

- *Oracle® Spatial and Graph Developer's Guide*

Oracle Workspace Manager Enhancements

Oracle Workspace Manager manageability and usability are enhanced:

- The DBA can now defer deletion of rows for a deleted workspace to another time.
- Shared locking is enabled for the create workspace procedure.
- A workspace can be configured so that changes to data cannot be propagated to other workspaces.
- The workspace that a session uses is tracked.

Delete workspace procedures are more flexible and complete faster and users are no longer blocked from simultaneously creating workspaces in the same parent workspace. A completely isolated test-only sandbox workspace can be created. DBAs can easily find the workspace used by a session.

Related Topics

- *Oracle® Database Workspace Manager Developer's Guide*

Windows

General

Windows Group Managed Service Accounts

In Windows 8 and Windows Server 2012, Microsoft introduced a new services account, Group Managed Services Account (gMSA). This is a domain level account and can be used by multiple servers in that domain to run their services under this account. gMSA is different from a standalone MSA because a gMSA can be associated with multiple servers in the domain, whereas an MSA can only be associated with a single server in the domain. A gMSA can be the Oracle home user for Oracle Database Real Application Clusters (Oracle RAC), single instance, and client installations.

Using a gMSA simplifies Oracle database installation and management because a password is not required during service creation and service password maintenance is not needed once the services are created.

Related Topics

- *Oracle® Database Platform Guide for Microsoft Windows*

Windows Virtual Accounts

In Windows 7 and Windows Server 2008 R2, Microsoft introduced virtual accounts, which are managed local accounts that use computer credentials to access network resources. A virtual account can be the Oracle home user for Oracle Database single instance and client installations.

Using Windows, virtual accounts simplifies Oracle Database installation and management because a password is not required during service creation and service password maintenance is not needed once the services are created.

Related Topics

- *Oracle® Database Platform Guide for Microsoft Windows*

Integrating With the Latest Windows Platforms and Services

- Windows Resilient File System
- Windows Install: Support Virtual Accounts and Group Managed Accounts

Windows Resilient File System

Resilient File System (ReFS) is a new local Windows file system that is more reliable and scalable than NTFS. Since Oracle Database 11*g* Release 2 (11.2.0.4), Oracle Database has supported installations on ReFS. In the latest Oracle Database release, I/O changes take advantage of ReFS scalability features.

Applications that use ReFS experience better scalability using Oracle Database.

Related Topics

- *Oracle® Database Platform Guide for Microsoft Windows*

Windows Install: Support Virtual Accounts and Group Managed Accounts

This release adds support for virtual accounts and group managed accounts for creating and managing Windows services.

Virtual accounts and group managed accounts eliminate the need to manage passwords for Windows services accounts simplifying service administration.

Related Topics

- *Oracle® Database Platform Guide for Microsoft Windows*

Oracle Database Manageability on Windows

- Microsoft Management Console Can Manage Oracle Database
- Windows Direct NFS Client Supports All Widely Accepted NFS Path Formats

Microsoft Management Console Can Manage Oracle Database

Administrators can now execute basic Oracle Database administrative tasks, such as create, edit, delete, start, and shutdown, from within Microsoft Management Console (MMC). This release introduces a new MMC snap-in so that these functions can be performed.

Windows administrators that need to perform basic Oracle Database administration tasks can do so within the familiar graphical user interface of MMC.

Related Topics

- *Oracle® Database Platform Guide for Microsoft Windows*

Windows Direct NFS Client Supports All Widely Accepted NFS Path Formats

Previously, Oracle's Windows Direct NFS (dNFS) Client could only use Windows specific network file system (NFS) paths. This new feature enhances the dNFS Client so that it can support all widely accepted NFS path formats, including both Windows style and UNIX style NFS paths.

Administrators now have the flexibility to use their preferred NFS path format.

Related Topics

- *Oracle® Database Installation Guide for Microsoft Windows*