

# Application Development with Oracle Database 12c

ORACLE WHITE PAPER | JANUARY 2017





## Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



## Table of Contents

Disclaimer	1
Introduction	6
Database Access Infrastructure and APIs (Java, C, C++)	7
Introduction	7
Oracle Sharding	7
Oracle Multitenant	7
New SQL Features	8
Auto-Increment (IDENTITY) Columns	8
Row Limiting Clause	8
Invisible or Hidden Columns	9
PL/SQL Package Types and Boolean Types as Parameters	9
Implicit Results	9
Performance and Scalability	9
SQL Plan Management (SPM)	9
Scale with Database Resident Connection Pool	9
Global Data Services – Runtime Load-Balancing	9
Large Network Buffers and Advanced Network Compression	10
Maximum Application Availability and Reliability	10
Common HA Events & Notification	10
New Concepts	10
Transaction Guard	10
Application Continuity	11



Global Data Services	11
Usability	11
Row Count per Iteration for Array DML	11
Client-Side Deployment Descriptors and Auto-Tuning	12
Monitoring and Tracing Database Operations	12
Intelligent Connectivity and Faster Dead Client Detection	12
Longer Identifiers	12
PL/SQL Callback for Session State Fix Up	12
Security	13
Customizing Default Java Security for Java in the Database	13
JDBC Enhanced Security	13
Advanced Security Option	13
Java Standards	13
JDBC 4.1	13
JDK 1.7 and Multiple JDK	14
JNDI	14
Logging	14
Application Migration	14
SQL Translation Framework	14
Driver for MySQL Applications	15
Additional Resources	15
Oracle Database Cloud Service	16

SQL and PL/SQL	17
SQL and PL/SQL Key New Features	17
Enhanced Oracle Native LEFT OUTER JOIN Syntax	17
Define Tables with Valid-Time Support	18
Valid-Time Temporal Flashback Queries	18
PL/SQL Functions Defined in the SQL WITH Clause	18
PL/SQL-Specific Data Types Allowed Across the PL/SQL to SQL Interface	18
Restrict the Ability to Reference a PL/SQL Unit to a List of Database Objects	19
An Invoker's Rights Function Can Be Result Cached	19
Object Type LIBRARY Defined Using an Object Type DIRECTORY	19
New PL/SQL Package UTL_CALL_STACK	19
New PL/SQL Subprogram DBMS_UTILITY.EXPAND_SQL_TEXT	19
New Predefined PL/SQL Inquiry Directives	19
ACCESSIBLE BY Clause Enhancements	20
Data-Bound Collation	20
PL/SQL Expressions Enhancement	20
Support for SQL JSON operators in PL/SQL	20
Support for Longer Identifiers	20
Code Coverage Analysis	21
The DEPRECATE Pragma	21
PL/Scope Extended to SQL Statements	21
Bind PL/SQL-Only Datatypes in DBMS_SQL	21
Oracle SQL Developer	22
Oracle SQL Developer Core Areas	22



Development	22
Database Administration	22
Modeling	22
Migrations	22
Benefits	22
Oracle Database 12c Support	23
Oracle Multitenant	23
SQL Translation Framework	23
Oracle Data Redaction	23
Oracle Application Express	24
Oracle Application Express 5.0 Key Features	25
Page Designer	25
Universal Theme	25
Interactive Reports	25
Charting Capabilities	26
Dynamic Actions	26
Plug-Ins	26
Modal Dialogs	26
Calendar	26
Mobile Reporting	26
Packaged Applications	27
Websheets	27
Oracle REST Data Services	28



How ORDS Builds REST APIs	29
How ORDS Works with new SQL/JSON Functions in Oracle Database 12c	29
ORDS Architecture	30
ORDS Security	30
Conclusion	30
JSON	32
.NET	33
.NET Key New Features	34
Application Continuity	34
Sharding	34
Cloud	34
ODP.NET, Managed Driver Enhancements	34
Longer Schema Identifiers	35
Globalization Support	36
Globalization Support New Features	36
Unicode 6.1 Support	36
Unicode Collation Algorithm (UCA) Conformance	36
New Locale Coverage	36
Database Migration Assistant for Unicode (DMU)	37
Conclusion	38



## Introduction

The efficiency and capability of applications is largely governed by the capabilities of the underlying database. Oracle Database 12c delivers significant new functionality for application development and deployment. Oracle Database 12c is the first relational database designed for the Cloud. Oracle Multitenant is new in Oracle Database 12c and provides a highly secure, multitenant database architecture that makes it easier to develop for the Cloud by relieving developers of the need to code multitenancy at the application level. Oracle Sharding is introduced in Oracle Database 12c allow databases to scale out across servers.

Oracle Database 12c offers customers the power and performance of the industry's leading database delivered across a wide variety of the most popular application development technologies including Oracle APEX, SQL Developer, ORDS, PL/SQL, SQL, C, C++, Java, .NET, JSON, Node.js, PHP, Perl, Python, Ruby, Go and the R language. Oracle Database 12c is a single integrated database platform offering market-leading security, high performance, availability and scalability for on-line transaction processing, data warehousing and Big Data.

New functionality has been added to Oracle Database 12c with a focus on ease of use, security, scalability and performance. Many new features have been added to make it easier than ever to migrate from other databases to Oracle Database 12c. This paper provides an overview of key new features of interest to application developers.

## Database Access Infrastructure and APIs (Java, C, C++)

### Introduction

In this section, we'll look at new features for Oracle Database access infrastructure and APIs. The features focus on the performance, scalability, availability, security and usability of Java, C, and C++ applications. Scripting language drivers can immediately benefit from many of these features. (A few features that require client API changes depend on driver maintainers adding support.) These drivers include OCI-based drivers like node-oracledb for Node.js, PHP's OCI and PDO\_OCI extensions, Ruby's ruby-oci8 extension, Python's cx\_Oracle driver, Perl's DBD::Oracle, and the R language's ROracle driver. Languages like Jython and JRuby that use Java's JDBC can also benefit.

### Oracle Sharding

Oracle Sharding is a shared-nothing architecture that allows near-linear scaling of the database across low-cost commodity database servers located in one or more local or global data centers. Sharding's benefits include global data distribution (store particular data close to consumers) and fault containment (failure of one shard does not affect the availability of other shards). Global Data Services manages the location of data among the shards and allows application requests to be routed to the appropriate shard in this distributed database system.

### Oracle Multitenant

In 12c, Multitenant provides great efficiency and great agility, both of which are key capabilities for this use case.

With sophisticated automation, technologies associated with first-generation clouds can improve this cost model to a nearly linear function. Let's consider a development environment, for example. To support ten developers, you might need ten virtual machines (VMs). To support twenty developers, you'll need 20 VMs. There is an intrinsic cost to each VM, even if it's not carrying any great workload. Hence the linear relationship between cost and scale. However, this was still an important step in the right direction. Oracle Multitenant is the architecture for the next-generation Database Cloud. Multitenant changes the situation completely. Multitenant delivers true economies of scale. The expensive model of a VM containing a database is replaced by a pluggable database (PDB). Because there is negligible intrinsic cost to a PDB, the cost of each developer's PDB is reduced to the actual work they do. All developers' PDBs can be consolidated into a single multitenant container database (CDB) and the costs of running that CDB can be shared among those developers. In terms of compute resources that's because there's a single set of background processes and a single shared memory area (the SGA). In terms of administration, that's a single CDB to be backed up, configured for high availability, patched, etc.

Modern, innovative and highly productive development and testing organizations demand great agility. For database components of an application this requires highly efficient capabilities to provision databases. Multitenant in 12.2 supports several powerful on-line provisioning capabilities. Provisioning of PDBs is accomplished by a process known as cloning, of which there are many varieties including full clones, subset clones and even thin provisioning of snapshot clones (leveraging copy-on-write technology). Pluggable databases (PDBs) may be cloned without requiring any downtime in the source. Incremental refresh capabilities are supported to allow "development master" clones of even huge production databases to be topped up with fresh data simply and efficiently with negligible impact on the source. On-line relocation of PDBs realizes the dream of simple migration to the Cloud without requiring an application outage.

For Software as a Service (SaaS), Multitenant in 12.2 delivers a powerful new Application Container capability. This extends the benefits of managing many tenants as one from the DBA to the Application Administrator. For applications formerly architected for standalone deployment on premises, Application Container provides an instant



cloud architecture. For cloud-born applications, the SaaS characteristic of great economy of scale has typically been achieved at the expense of agility. Application Container delivers all the agility of PDBs without compromising on these great economies of scale. A master application definition can now be installed in a single container, known as Application Root. Individual tenants in this Application Container are served by this master definition. This enables application maintenance operations (such as applying an application patch) to be performed in a central location and individual tenants can simply synchronize with this new version on their own schedule. Powerful cross-container aggregation capabilities are possible, allowing for the simple creation of dashboards from which an ISV can assess the overall performance of the application using key metrics such as number of orders or sum of revenue in the current quarter. These capabilities are not just for ISVs, but apply equally well to many other business models, such as franchises or global organizations that replicate business practices across multiple subsidiaries around the globe.

JDBC, SQLJ, OCI and OCCI drivers support Pluggable Databases. From Java, C, and C++ applications perspective, PDBs feel and operate identically to traditional Oracle databases, provided a SERVICE name (Net Services parlance) is used in connect strings.

## New SQL Features

Java, C, C++ applications may leverage new SQL features including: 32K VARCHAR, NVARCHAR, and RAW, invisible/hidden columns, implicit results, auto-increment or IDENTITY columns, and PL/SQL package types and Boolean as parameters. Since scripting languages are written in these core languages they can take advantage of the new features.

### 32K VARCHAR, NVARCHAR, and RAW

The maximum size of the VARCHAR2, NVARCHAR2, and RAW data types has been increased from 4,000 to 32,767 bytes. Java, C and C++ applications using respectively JDBC, OCI, and OCCI will no longer need to switch to large objects (LOBs) less than 32K in size; indexes may also be built on top of columns declared with such data types.

Increasing the allotted size for these data types allows users to store more information before switching to large objects (LOBs). This is especially useful for brief textual data types and the capabilities to build indexes on these types of columns.

### Auto-Increment (IDENTITY) Columns

Table columns have been enhanced to support the American National Standards Institute (ANSI) IDENTITY keyword.

This provides a standards-based approach to the declaration of automatically incrementing columns simplifying application development and making the migration of DDL to Oracle simpler.

### Row Limiting Clause

Oracle Database 12c introduces the OFFSET and FETCH NEXT clauses for SELECT, allowing subsets of query records to be easily fetched without requiring any nested query syntax. This makes it easier to write correct, simple SQL queries, for example when web applications need to 'page' through data.

### **Invisible or Hidden Columns**

New columns can be created or added to table(s) then hidden, using the INVISIBLE SQL keyword. Invisible columns are not displayed during generic access such as a “SELECT \* FROM table” or “DESCRIBE table” however these may be displayed when specified explicitly in the SELECT list.

### **PL/SQL Package Types and Boolean Types as Parameters**

This feature allows database client APIs (for example, OCI and JDBC) to natively describe and bind PL/SQL package types and boolean types. Java and C-based applications can now easily bind and execute PL/SQL functions or procedures with PL/SQL package types or boolean types (except Java) as parameters.

This feature reduces the complexity of executing PL/SQL functions or procedures from client-side applications.

### **Implicit Results**

Implicit Results allows Java, C and C++ applications to retrieve the return of stored procedures (PL/SQL, Java) directly, without the need to use REF CURSOR, using the PL/SQL procedure DBMS\_SQL.RETURN\_RESULT. This feature makes it easier to migrate Java, C, C++ applications from third-party databases to Oracle Database.

### **Performance and Scalability**

Java, C and C++ applications can leverage the following performance and scalability features: SQL plan management, DRCP support for Java and other enhancements, very large network buffers, advanced network compression, and runtime load balancing across geographies with Global Data Services.

### **SQL Plan Management (SPM)**

SQL Plan Management enhancements in this release include:

- Storing execution plans for plan baselines
- Create automatic jobs that performs the evolve task

SQLJ now supports the creation of plan baselines. You can generate plan baselines at the time of translating the SQLJ files. The necessary SQL statements to create the plan baselines are generated in the .sql files.

### **Scale with Database Resident Connection Pool**

Database Resident Connection Pool (DRCP) is an RDBMS-side pool of server processes shared across client applications, programming languages and middle-tiers. Introduced in Oracle Database 11g OCI for C, C++, PHP, Python and Perl, it is indispensable for single-threaded systems (such as PHP) that cannot do client/middle-tier connection pooling. The reduction of database server processes and memory may reach orders of magnitude, which makes DRCP beneficial for multi-threaded systems as well for the large scale deployment of thousands of middle-tiers accessing the same database.

DRCP is now available for Java through Oracle JDBC drivers. A client-side connection pool e.g., Oracle's Universal Connection Pool (UCP) for Java (similar to the OCI Session Pool for C/C++) is required for tracking connections check-in and check-out on the clients/mid-tiers.

New JDBC connection properties oracle.jdbc.DRCP.name and oracle.jdbc.DRCP.purity allow Java applications to name DRCP pools and sub-partition a single pool across several applications.

### **Global Data Services – Runtime Load-Balancing**

UCP now supports Runtime Load Balancing across geographies.

## Large Network Buffers and Advanced Network Compression

Oracle Net Services now supports very large network buffers a.k.a. Session Data Unit (SDU), up to 2 MB, up from 64K in Oracle database 11g Release 2. The SDU can be configured in SQLNET.ORA on RDBMS side and in SQLNET.ORA, TNSNAMES.ORA or connect strings on clients/mid-tiers.

Advanced Network Compression reduces the size of the SDU transmitted over a network connection, thereby increasing response time, network throughput and reducing bandwidth utilization. Once configured, Advanced Network Compression works transparently to clients (applications, mid-tiers) and the server (RDBMS).

More details can be found in the “Advanced Network Compression” white paper on the Net Services portal and in the Net Services Administrator Guide.

## Maximum Application Availability and Reliability

Java, C and C++ application can achieve maximum availability and reliability through the following new features: a consolidated common HA events notification, Transaction Guard, Application Continuity and Fast Connection Failover across geographies with Global Data Services.

## Common HA Events & Notification

RAC and Data Guard emit HA events such as NODE DOWN, INSTANCE UP/DOWN, SERVICE UP/DOWN, and so on; these events are carried and notified to subscribers (drivers, applications) using ONS.

## New Concepts

**Recoverable Error:** Oracle Database 12c exposes a new error attribute that applications can use to determine if an error is recoverable or not. C and C++ applications using OCI/OCCI will now be on par with Java and will no longer need to maintain their own list of error codes (e.g., ORA-1033, ORA-1034, ORA-xxx).

**Database Request:** a unit of work submitted by the application, including SQL PL/SQL, local calls, and remote procedure calls; has typically one COMMIT but could have zero or more than one.

**Logical Transaction ID (LTXID):** a new concept to obtain the last COMMIT outcome.

Mutable Functions: non-deterministic functions that can change their results each time they are called e.g., SYSDATE, SYSTIMESTAMP, SEQUENCES, SYS\_GUID.

**Session State Consistency Model:** non-transactional session state; could be dynamic or static.

## Transaction Guard

Transaction Guard provides a generic tool to address the challenge of reliably determining the outcome of the last COMMIT operation following a break in communication between the client application and Oracle Database.

Without Transaction Guard, if a transaction has been started and commit has been issued, the commit message that is sent back to the client is not durable. The client is left not knowing whether the transaction committed or not. The transaction cannot be resubmitted if the non-transactional state is incorrect or if it already committed. In the absence of knowing the commit and completion information, resubmission can lead to transactions being applied more than once and in the incorrect state.

The benefits of Transaction Guard are:

- First database to preserve commit outcome.
- Known outcome for every transaction.

- A tool for at-most-once transaction execution.

The Transaction Guard API is available with JDBC-thin, OCI, OCCI, and ODP.NET.

For more details and code fragments, consult the “Maximum Application Availability” and “Transaction Guard” white papers on the Oracle Technology Network (OTN).

### **Application Continuity**

Application Continuity in Oracle Database 12c provides “almost non-intrusive” application availability, out-of-the-box when C or Java Applications use RAC and Active Data Guard. Application Continuity transparently:

1. Captures in-flight work, during normal runtime, within “database request” boundaries
2. Upon database instance or site failure, on recoverable errors, uses Transaction Guard under the covers and reconnects to a good Database instance (RAC) or disaster recovery site (ADG),
3. Replays captured in-flight work

Application Continuity provides the following benefits:

- Masks hardware, software, network, storage errors and outages
- Improved end user experience; i.e., a slight delay in response time, upon failures.

Application Continuity for Java is available out-of-the box through JDBC-Thin only and UCP. Third party drivers or connection pools may enable it by explicitly demarcating (beginRequest/endRequest) the units of work (a.k.a. “database requests”). The standard JDBC interface is required instead of deprecated Oracle.SQL concrete classes: BLOB, CLOB, BFILE, OPAQUE, ARRAY, STRUCT, or ORADATA; see My Oracle Support Note 1364193.1

Although mostly non-intrusive, Application Continuity design considerations and restrictions are discussed at length in Oracle “Application Continuity” and “Maximum Application Availability” white papers.

### **Global Data Services**

The Oracle Universal Connection Pool (UCP) supports Fast Connection Failover and Application Continuity across geographies. See the Oracle Database 12c Active Data Guard white paper for more details on Global Data Services.

### **Usability**

Oracle database 12c introduces a number of usability features. These include finding the row count per iteration for array DML, client-side deployment descriptor, auto-tuning, monitoring and tracing database operations, intelligent client connectivity and faster dead connection detection.

### **Row Count per Iteration for Array DML**

Java, C and C++ applications using respectively JDBC, OCI, and OCCI can now retrieve the number of rows affected by, each iteration of an array DML statement (array INSERT, UPDATE, DELETE).

For C and C++ applications, OCI furnishes:

- A new mode parameter attribute: OCI\_RETURN\_ROW\_COUNT\_ARRAY in the OCIStmtExecute() call
- And a new statement handle attribute: OCI\_ATTR\_ROW\_COUNT\_ARRAY

For Java applications, Oracle JDBC now returns the number of rows effected in each iteration. The following statement now prints the update count for each UPDATE.

```
int rcount[] = stmt.executeBatch();
```

### **Client-Side Deployment Descriptors and Auto-Tuning**

Similar to Java EE, the behavior of C or C++ applications using OCI or OCCI can now be controlled via a deployment descriptor `oraaccess.xml`. It can be used to configure selected OCI/OCCI parameters to change the behavior of C/C++ applications at deployment time, without the need to modify and recompile source code. The deployment time settings override programmatic settings.

Updates to `oraaccess.xml` will not affect already running clients; these need to be restarted to uptake the changes. The `oraaccess.xml` file is read either from the default location i.e., `$ORACLE_HOME/network/admin` on UNIX and `%ORACLE_HOME%\NETWORK\ADMIN` on Microsoft Windows, or from the directory specified by the `TNS_ADMIN` environment variable in regular and Instant Client installations.

Leveraging `oraaccess.xml` deployment descriptor, OCI now furnishes automatic and non-intrusive performance management of C/C++ applications, in the areas of Client Result Cache, Connection Cache, Statement Cache, Prefetch Memory Size, and ONS.

In addition, a new session handle attribute `OCI_ATTR_MAX_OPEN_CURSORS` is also provided.

### **Monitoring and Tracing Database Operations**

For end-to-end tracing Oracle Database furnishes a reserved namespace (OCSID) for storing tags: `MODULE`, `ACTION`, `CLIENTID`, `ExecutionContextID (ECID)`, `MODULE`, `SEQUENCE_NUMBER` and the new `DBOP`. These tags may be associated with a thread without requiring an active connection to the database or client/server. When the application makes a database call, the tags are sent along to the database, piggybacking on the application's connection.

Java applications can use `DBOP` and other OCSID through either JDBC `setClientInfo()` method or DMS APIs while C and C++ applications can use these through DMS APIs and the new OCI `OCI_ATTR_DBOP`.

### **Intelligent Connectivity and Faster Dead Client Detection**

Net Services is now smarter during connection attempts and decrease the priority of unresponsive nodes in the address string of connect descriptor, for subsequent attempts thereby increasing connectivity time and availability.

Similarly, the detection of terminated session/connection has been accelerated.

The `SQLNET.EXPIRE_TIME` parameter in the `sqlnet.ora` configuration file helps detect terminated clients faster. If the system supports TCP keepalive, then Oracle Net Services automatically uses the enhanced detection model, and tunes the TCP keepalive parameters.

For user of OCI, the Session Pool also has improved dead connection detection, cleanup and idle session reestablishment. This improves reliability of applications after network outages occur.

### **Longer Identifiers**

OCI and Java applications can take advantage of the increased identifier length in Oracle Database 12.2. Identifiers can now be 128 bytes.

### **PL/SQL Callback for Session State Fix Up**

Users of the OCI Session Pool can now 'tag' sessions with multiple properties. Applications can use these tag properties to indicate what state is set on a session.

When an application checks-out a session from the session pool, if not all requested tags are set on the obtained session, a PL/SQL callback can be invoked with the desired and actual tags. This can run any desired session state



fix-up logic. Application logic can be simplified because the fix-up is run automatically. Performance is improved because the fix-up is run on the server.

## Security

### Customizing Default Java Security for Java in the Database

Oracle Database includes enhanced permission and policy management for Java runtime. The Java policy is reloadable by the system administrator after adding third-party encryption suites. In addition, the database administrator can change the algorithm search order.

These enhancements provide tighter permission and policy management, as well as flexible and advanced security support for third-party encryption libraries.

### JDBC Enhanced Security

Java Database Connectivity (JDBC) supports the security enhancements in Oracle Database including Kerberos authentication, SHA-2, and Windows Authentication (NTS).

This feature provides advanced security for Java applications.

### Advanced Security Option

Oracle Advanced Security uses the following hashing algorithms to generate the secure message digest and includes it with each message sent across a network: MD5, SHA1, SHA256, SHA384 or SHA512.

JDBC now supports SHA-2 hashing algorithms including: SHA-224, SHA-256, SHA-384, SHA-512.

DRCP in OCI now supports encryption and strong authentication (except TCPs).

Invoking Runtime.exec with Java in the Database

For security reasons, it is advisable to run the processes forked by the Runtime.exec functionality with OS identity granted lesser rights. The following procedure associates a database user/schema DBUSER with an OS osuser account:

```
dbms_java.set_runtime_exec_credentials('DBUSER', 'osuser', 'ospass');
```

## Java Standards

Oracle Database 12c also brings the following Java standards: JDBC 4.1, JDK 1.7, Logging and JNDI.

### JDBC 4.1

Oracle Database 12c now supports the JDBC 4.1 getObject() method with two signatures.

```
getObject(int parameterIndex, java.lang.Class<T> type)
```

throws SQLException

```
getObject(java.lang.String parameterName, java.lang.Class<T> type)
```

throws SQLException

### Example

```
ResultSet rs = . . . ;
```

```
Character c = rs.getObject(1, java.lang.Character.class);
```



Other JDBC 4.1 feature include try-with-resources which will work however, Oracle JDBC does not yet have support for RowSetFactory interface.

### **JDK 1.7 and Multiple JDK**

Both Oracle JDBC and SQLJ drivers now support JDK 1.7 through ojdbc7.jar

The database-resident Java VM which allows running plain Java SE applications directly in the DATABASE now supports JDK 1.6 (the default) and JDK 1.7 with the ability to upgrade/downgrade from one to the other.

### **JNDI**

The database resident Java VM now supports a general purpose directory service for storing objects and object references including:

- A Java Directory Service Namespace Provider (OJDS)
- A namespace: similar to Unix file system structure /pub /etc
- A namespace browser (ojvmjava utility): enables browsing permissions and properties of objects stored in the OJDS

### **Logging**

The database-resident Java VM now furnishes a logging properties initialized once per session with the LogManager API that is extended with the database resource lookup.

- If \$ORACLE\_HOME/javavm/lib/logging.properties resource is set in the current user schema, this resource is used for configuring the LogManager and the java.util.logging.config.file property is set
- Otherwise the \$ORACLE\_HOME/javavm/lib/logging.properties resource in the SYS schema is used

You may configure a different properties file and load them in your schema using the loadjava utility.

### **Application Migration**

In addition to new data types covered above, the following new features are provided, to simplify and reduce the cost of migrating Java, C and C++ applications built on foreign RDBMS onto Oracle Database: a framework and mechanism for translating foreign SQL syntax for execution against Oracle Database and a driver to run MySQL applications against Oracle Database.

### **SQL Translation Framework**

A key part of migrating non-Oracle databases to Oracle Database involves the conversion of non-Oracle SQL statements to SQL statements that are acceptable to Oracle Database. The conversion of the non-Oracle SQL statements of the applications is a manual and tedious process. To minimize the effort, or to eliminate the need to convert these statements, Oracle Database 12c introduces a new feature called SQL Translation Framework which will accept SQL statements from client applications and translate them at run-time before execution.

Inside the database, SQL statements are translated by the SQL Translator, registered with the SQL Translation Profile, to handle the translation for the non-Oracle client application. If an error occurs while a SQL statement is executed, then the SQL Translator can also translate the Oracle error code and the ANSI SQLSTATE to the vendor-specific values expected by the application. The translated statements are then saved in the SQL Translation Profile, so that you can examine and edit at your discretion.

The advantages of SQL Translation Framework are the following:

- The translation of SQL statements, Oracle error codes, and ANSI SQLSTATE is automatic.

- 
- The translations are centralized and examinable.
  - You can extract the translations and insert them back to the applications at a later point.

Java, C and C++ applications using ODBC, OCI and JDBC may now use the new SQL Translation mechanism which allows the text of a SQL statement to be translated to Oracle SQL syntax before being submitted to the Oracle Database SQL compiler.

#### **Driver for MySQL Applications**

The Oracle Database driver for MySQL applications is a drop-in replacement for the client library for MySQL 5.5. It enables applications and tools built on languages based on MySQL's C API (PHP, Ruby, Perl, Python, and so on) to run against an Oracle Database using the new Oracle library that implements the MySQL C API.

The driver assumes that the translation of MySQL SQL dialect is taken care of by either the SQL Translation framework (with a MySQL profile), or a custom hand-made SQL translation mechanism.

The key benefits are the reuse of MySQL applications against both MySQL and Oracle and the reduction in the costs and complexities of migrating MySQL applications to Oracle.

#### **Additional Resources**

More information about the products and features covered in this section can be found below.

OCI: <http://www.oracle.com/technetwork/database/features/oci/index-090945.html>

Oracle Java, JDBC, UCP, & Database Web Services: <http://www.oracle.com/technetwork/database/enterprise-edition/index-097123.html>



## Oracle Database Cloud Service

The Oracle Database is the world's leading enterprise database. The Oracle Database Cloud Service is a rich and robust offering combining a rapid application development tool, instant deployment for applications, RESTful Web Service access and a set of packaged applications which install with a few mouse clicks. The Oracle Database Cloud is built on the Oracle Database and delivers the power and robustness of the Database, but also provides additional capabilities in a fully managed service.

In terms of Cloud computing, the Oracle Database Cloud Service, although based on and resident in the Oracle Database, is a Platform-as-a-Service product, rather than a Database-as-a-Service product. The Database Cloud Service comes with three main components – a rapid application development and deployment tool, Oracle Application Express, that has been proven capable of enterprise-strength application development since its release in 2004; a RESTful Web Service interface which gives secure access to SQL and PL/SQL through this popular Web Service standard; and a collection of applications that can be installed and run with just a few clicks, providing instant business value. The Database Cloud Service is currently running Oracle Database 11g and will be upgraded to Oracle Database 12c in the future.

For more on Oracle Database Cloud Service, go to: <https://cloud.oracle.com/mycloud/f?p=service:database:0>

## SQL and PL/SQL

PL/SQL is an imperative 3GL that was designed specifically for the seamless processing of SQL commands. It provides specific syntax for this purpose and supports exactly the same datatypes as SQL. While it is available in other environments, this document focuses on PL/SQL that is stored and compiled in Oracle Database and that runs within the Oracle executable where it automatically inherits the portability of Oracle Database.

A best practice used by many Oracle customers is to have client code access Oracle Database only by calling PL/SQL subprograms. This approach not only reflects generic modular programming best practices (define a clean, functional API and hide the implementation); also, it typically drastically reduces network round trips between the client and Oracle Database.

### SQL and PL/SQL Key New Features

The following sections highlight SQL and PL/SQL new features included in Oracle Database 12c .

#### Native SQL Support for Query RowLimits and RowOffsets

The FETCH FIRST and OFFSET clauses enable native SQL queries to limit the number of rows returned and to specify a starting row for the return set.

Many queries need to limit the number of rows returned or offset the starting row of the results. For example, Top-N queries sort their result set and then return only the first n rows. FETCH FIRST and OFFSET simplify syntax and comply with the ANSI SQL standard.

#### SQL CROSS APPLY, OUTERAPPLY and LATERAL

The APPLY SQL syntax allows a table-valued function to be invoked for each row returned by a query's outer table expression. The table-valued function acts as the right input; the outer table expression acts as the left input. The right input is evaluated for each row from the left input and the rows produced are combined for the final output. Therefore, one can pass left-correlations to the table-valued functions.

There are two forms of APPLY - CROSS APPLY and OUTER APPLY. CROSS APPLY returns only rows from the outer table that produce a result set from the table-valued function. OUTER APPLY returns both rows that produce a result set, and rows that do not, with NULL values in the columns produced by the table-valued function.

LATERAL, part of the ANSI standard, is an extension of the inline view syntax that provides left-correlation scoping within the inline view. These new keywords provide easier and more flexible ways to evaluate and return SQL query results.

#### Enhanced Oracle Native LEFT OUTER JOIN Syntax

This release provides an extension of the Oracle native LEFT OUTER JOIN syntax that allows multiple tables on the left hand side of an outer join.

Prior to Oracle Database 12c , having multiple tables on the left hand side of an outer join was illegal and resulted in an ORA-01417 error. The only way to execute such a query was to translate it into ANSI syntax. In Oracle Database 12c , the native syntax for a LEFT OUTER JOIN has been expanded to allow multiple tables on the left hand side. This expansion provides the following benefits:

- Merging of multiple table views on the left hand side of an outer join. Such views can originate from the user query or they may be generated during conversion from LEFT OUTER JOIN syntax.

- Merging of such views allows more join reordering and, therefore, more optimal execution plans. These views are merged in a heuristic manner without having to go through cost-based query transformation.
- It relieves the application developers from the burden of formulating queries in terms of views or LEFT OUTER JOIN syntax.

### **Define Tables with Valid-Time Support**

You can add a valid time dimension to a table using existing columns or using columns automatically created by the database. This can be done with both the CREATE TABLE and ALTER TABLE statements.

Applications often indicate the validity of a fact recorded in the database with dates or time stamps that are relevant to the underlying business they manage. Examples of such dates include the hire date of an employee in a Human Resources application, the effective date of coverage for an insurance policy, or the effective date of a change in address. This is in contrast to the date or time at which the fact was recorded in the database. The former temporal attribute is usually called valid-time (VT), while the latter is called transaction-time (TT). Valid-time is typically controlled by the user or application, while transaction-time is system managed. Flashback Data Archive, first introduced with Oracle Database 11g Release 1 (11.1) as part of Total Recall, implements transaction-time functionality. Valid-time support in Oracle Database reduces the complexity of application code by providing a simple declarative interface to allow the database to manage the validity of rows.

### **Valid-Time Temporal Flashback Queries**

Users can now execute queries with the AS OF and VERSIONS BETWEEN clauses based on valid-time. Queries that mix valid-time and transaction-time are called bi-temporal queries.

Users can now query data based on current values (that is, current in valid-time and transaction-time), what we know now (that is, AS OF in valid-time; CURRENT in transaction-time), or what we knew before (that is, AS OF in valid-time and transaction-time), giving declarative access to all possible views of data based on the two time dimensions. Bi-temporal queries in Oracle Database 12c Release 1 (12.1) provide functionality previously available only with extensive and complex application code.

### **PL/SQL Functions Defined in the SQL WITH Clause**

You can define a PL/SQL function in the WITH clause of a subquery and use it as an ordinary function beginning with this release.

The procedural logic needed to support a SQL statement is encapsulated with the SQL statement. This is particularly useful in a read-only database.

Using this construct results in better performance as compared with schema-level functions.

### **PL/SQL-Specific Data Types Allowed Across the PL/SQL to SQL Interface**

Through Oracle Database 11g Release 2 (11.2), when PL/SQL invoked SQL, only values with data types supported by SQL could be bound. This restriction applied even when the called SQL was a PL/SQL anonymous block. This restriction is removed in Oracle Database 12c Release 1 (12.1). For example, a PL/SQL subprogram with a formal parameter whose data type is BOOLEAN can now be invoked dynamically using an anonymous block.

Other restrictions are also removed. The table operator can now be used in a PL/SQL program on a collection whose data type is declared in PL/SQL. This also allows the data type to be a PL/SQL associative array. (In prior releases, the collection's data type had to be declared at the schema level.)



The removal of these restrictions increases the power of expression and the usefulness of PL/SQL. In particular, the extended flexibility of the table operator allows code written to run other vendors' stored procedure languages to be easily migrated to PL/SQL.

### **Restrict the Ability to Reference a PL/SQL Unit to a List of Database Objects**

It is now possible to mark a schema-level function, procedure, package, or type specification with a white list of allowed callers. The allowed caller may be of any object type that can invoke a PL/SQL subprogram (for example, a trigger, view, table, or index), but it must be in the same schema as the unit that has the white list. The white list is optional but, when used, only the listed objects may reference the unit in question. Cross-schema references to a unit with a white list are, therefore, disallowed even when the reference is attempted from a schema owned by SYS.

This capability supports the robust implementation of a module, consisting of a main unit and helper units, by allowing the helper units to be inaccessible from anywhere except the unit they are intended to help.

### **An Invoker's Rights Function Can Be Result Cached**

Through Oracle Database 11g Release 2 (11.2), only definer's rights PL/SQL functions could be result cached. Now, invoker's rights PL/SQL functions can also be result cached. (The identity of the invoking user is implicitly added to the key of the result.)

At times, it may be appropriate to use an invoker's rights PL/SQL function to issue one or more SELECT statements. This feature improves performance.

### **Object Type LIBRARY Defined Using an Object Type DIRECTORY**

In previous releases, an object of the LIBRARY type could only be defined by using an explicit path. However, now the DIRECTORY type can be the single point of maintenance for file system paths. Moreover, using a DIRECTORY type has security benefits. A directory object can be defined using a DIRECTORY type.

Additionally, the definition of an object of the LIBRARY type can now include a credential so that the designated external program can be run as a different operating system user than the owner of the Oracle installation.

These enhancements improve security and portability of an application that uses external procedures.

### **New PL/SQL Package UTL\_CALL\_STACK**

The UTL\_CALL\_STACK package provides subprograms to return the current call stack for a PL/SQL program.

It is functionally similar to the existing DBMS\_UTILITY.FORMAT\_CALL\_STACK which returns information as a human-readable essay. This new package makes this information available in a structured representation amenable for programmatic analysis.

### **New PL/SQL Subprogram DBMS\_UTILITY.EXPAND\_SQL\_TEXT**

The DBMS\_UTILITY.EXPAND\_SQL\_TEXT subprogram accepts a subquery that references views and returns a subquery with the identical meaning that references only tables.

This functionality can help in the analysis of SQL which depends on views with the aim of fixing application logic or resolving performance issues.

### **New Predefined PL/SQL Inquiry Directives**

The \$\$PLSQL\_OWNER and \$\$PLSQL\_TYPE predefined PL/SQL inquiry directives are now supported in this release.



Through Oracle Database 11g Release 2 (11.2), the predefined inquiry directives, \$\$PLSQL\_LINE and \$\$PLSQL\_UNIT, allowed diagnostic code to identify the current PL/SQL statement, but with a certain ambiguity. This ambiguity is now removed.

#### New SCHEMA Formal for DBMS\_SQL.PARSE()

DBMS\_SQL.PARSE() has a new formal SCHEMA. It specifies the schema in which to resolve unqualified object names.

This allows a definer's rights unit to control the name resolution for the dynamic SQL it issues.

#### **ACCESSIBLE BY Clause Enhancements**

The ACCESSIBLE BY clause specifies a list of PL/SQL units that are considered safe to invoke the subprogram, and blocks all others.

Starting with Oracle Database 12c release 2 (12.2), the accessor list can be defined on individual subprograms in a package. This list is checked in addition to the accessor list defined on the package itself (if any). This list may only restrict access to the subprogram – it cannot expand access. This code management feature is useful to prevent inadvertent use of internal subprograms. For example, it may not be convenient or feasible to reorganize a package into two packages: one for a small number of procedures requiring restricted access, and another one for the remaining units requiring public access.

#### **Data-Bound Collation**

Collation (also called sort ordering) is a set of rules that determines if a character string equals, precedes, or follows another string when the two strings are compared and sorted.

Oracle Database 12c release 2 adds a new property called default collation to tables, views, materialized views, packages, stored procedures, stored functions, triggers, and types. The default collation of a unit determines the collation for data containers, such as columns, variables, parameters, literals, and return values, that do not have their own explicit collation declaration in that unit.

#### **PL/SQL Expressions Enhancement**

Starting with Oracle Database 12c release 2 (12.2), expressions may be used in declarations where previously only literal constants were allowed. Static expressions can now also be used in subtype declarations.

Expanded and generalized expressions have two primary benefits for PL/SQL developers: (1) Programs are much more adaptable to changes in their environment; (2) Programs are more compact, clearer, and substantially easier to understand and maintain.

#### **Support for SQL JSON operators in PL/SQL**

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

Oracle Database support for storing and querying JSON documents in the database is extended by the addition of new capabilities, including the ability to declaratively generate JSON documents from relational data using SQL and the ability to manipulate JSON documents as PL/SQL objects. SQL JSON operators are supported in PL/SQL with a few exceptions.

#### **Support for Longer Identifiers**

The maximum length of all identifiers used and defined by PL/SQL is increased to 128 bytes, up from 30 bytes in previous releases.



If the COMPATIBLE parameter is set to a value of 12.2.0 or higher, the representation of the identifier in the database character set cannot exceed 128 bytes. If the COMPATIBLE parameter is set to a value of 12.1.0 or lower, the limit is 30 bytes.

### **Code Coverage Analysis**

PL/SQL will now gather data on code coverage: which lines of your program units were executed in a given run of the code. The DBMS\_PLSQL\_CODE\_COVERAGE package enables you to collect data at the basic block level. PL/SQL developers and testing engineers use code coverage testing results as part of their standard quality assurance metric.

### **The DEPRECATE Pragma**

PL/SQL developers can now use the DEPRECATE pragma to mark a program unit or individual subprogram as deprecated. The compiler warnings tell users of a deprecated element that other code may need to be changed to account for the deprecation.

### **PL/Scope Extended to SQL Statements**

PL/Scope, which gathers information on identifier usages, allowing fine-detailed impact analysis of program units, now also gathers information about SQL statements embedded in PL/SQL program units. It also identifies use of native dynamic SQL and SQL hints.

### **Bind PL/SQL-Only Datatypes in DBMS\_SQL**

The ability to bind values of PL/SQL-only datatypes in SQL (first implemented in 12.1 for EXECUTE IMMEDIATE) is now extended to DBMS\_SQL.

For more information about PL/SQL, go to:

<http://www.oracle.com/technetwork/database/features/plsql>



## Oracle SQL Developer

Oracle SQL Developer is a no cost integrated development environment that simplifies the development and management of the Oracle Database. SQL Developer offers complete end-to-end development of your PL/SQL applications, a worksheet for running queries and scripts, a DBA console for managing the database, a reports interface, a complete data modeling solution, and a migration platform for moving your 3rd party databases to Oracle.

### Oracle SQL Developer Core Areas

**Development** - Using SQL Developer, users can browse, edit and create database objects, run SQL statements, edit and debug PL/SQL statements, build PL/SQL unit tests, run reports, and place files under version control.

**Database Administration** - With the DBA Navigator, DBAs can better manage their databases and maximize productivity and performance with SQL Developer's DBA utilities such as Database Diff and Database Copy as well as its graphical user interfaces for RMAN, Data Pump, Real Time SQL Monitoring, Explain Plan and Autotrace.

**Modeling** - Oracle SQL Developer Data Modeler is a graphical tool that enhances productivity and simplifies data modeling tasks. Using SQL Developer Data Modeler, users can create, browse and edit, logical, relational, physical, multi-dimensional, and data type models. The Data Modeler provides forward and reverse engineering capabilities and supports collaborative development through integrated source code control. Users can use Oracle SQL Data Modeler as a standalone tool or within Oracle SQL Developer.

**Migrations** - SQL Developer supports migrating 3rd party databases to Oracle. The tight integration capabilities provide users with a single point to browse database objects and data in third party databases, and to migrate from these databases to Oracle. SQL Developer supports IBM DB2 UDB LUW, Microsoft SQL Server, Sybase Adaptive Server and Teradata.

For more information on migrating from another database to Oracle Database, please go to:

<http://www.oracle.com/technetwork/products/migration/index-084442.html>

### Benefits

Developed in Java, SQL Developer runs on Windows, Linux and Mac OS X. This is a great advantage to the ever-increasing numbers of developers using multiple platforms. Multiple platform support also means that users can install SQL Developer on the same system as the database and connect remotely from their desktops, thus avoiding client-server network traffic. Default connectivity to the database is through the JDBC thin driver, so no Oracle Home is required. To install SQL Developer simply unzip the downloaded file. With SQL Developer, users can connect to any supported Oracle Databases including Oracle Express Edition. Users can create database connections for non-Oracle databases for object and data browsing and migration. SQL Developer is a supported product for all customers with Oracle Database Support contracts.

SQL Developer is also available via a command-line interface. Oracle SQLcl supports everything SQL\*Plus has to offer but also includes:

- » SQL History
- » Code Assistance, Press <TAB> for help with keywords and table, view, column name completion
- » New commands such as CD, CTAS, DDL, REPEAT
- » JavaScripting

SQLcl only requires a JRE to run and is less than a 20MB download.



## Oracle Database 12c Support

Oracle SQL Developer Release 4.1 continues to support the latest Oracle Database 12c features.

**Oracle Multitenant** – Oracle Database 12c brings a new architecture to allow for many Pluggable Databases inside a single Oracle Database instance. Using SQL Developer, DBAs can connect to a 12c Database and create a Container Database to consolidate the Pluggable Databases. Full management capabilities such as creating, dropping, cloning, unplugging and syncing a Pluggable Database are available through easy to use wizards.

**SQL Translation Framework** – SQL Developer is the official Database Migration tool for Oracle. The introduction of the SQL Translation Framework in Oracle Database 12c allows for the support of migrating applications to Oracle. The SQL Translation Framework allows for minimal changes to the application code. The non-Oracle SQL will be sent unmodified to Oracle Database where the application SQL will then be recorded and translated. Users can review, tune and modify the translated SQL. The Translation Framework consists of the Translation profile to collect the translated statements and the SQL Translator, a Java compiler that translates the SQL. Using SQL Developer, users can create a SQL Translation Profile for each application and install a SQL Translator. SQL Developer supports Sybase and SQL Server translators.

**Oracle Data Redaction** – Part of the Oracle Advanced Security option in Oracle Database 12c, enables the masking of data that is returned from queries issued by low privileged users or applications. Different masking styles such as full, partial, random, and regular expressions can be applied to mask data. These masking styles are applied through a Redaction Policy. With SQL Developer, Redaction Policies are managed centrally. Policies can be created and applied to disguise any sensitive data.

You should always download the latest version of Oracle SQL Developer from the Oracle Technology Network (OTN). For more information about Oracle SQL Developer, go to:

<http://www.oracle.com/sqldeveloper>



## Oracle Application Express

Oracle Application Express (Oracle APEX) is a declarative, rapid web application development tool for the Oracle database. It is a fully supported, no cost option available with all editions of the Oracle database. Using only a web browser, you can develop, design and deploy beautiful, responsive, database-driven desktop and mobile applications that are both fast and secure.

Whether you are an experienced SQL and PL/SQL developer or a power user used to writing reports, wizards allow you to quickly build Web applications on top of your Oracle database objects. Enhancing and maintaining these applications is done using a declarative framework, all of which increases your productivity.

Oracle Application Express is database-centric and suited to building a vast array of applications. You can start with webifying a spreadsheet to facilitate collaboration or dive right into extremely complex applications with numerous external interfaces such as the Oracle Store. Because Oracle APEX resides within the Oracle Database and can easily integrate with authentication schemes (such as Oracle Access Manager, SSO, LDAP, etc.) you can build secure applications that can scale to meet your largest user communities.

Oracle Application Express provides four primary tools:

- **Application Builder** - to create dynamic database driven Web applications.
- **SQL Workshop** - to browse and maintain database objects, run ad-hoc SQL queries, as well as a graphical query builder. Utilities are provided to load and unload data, set user interface defaults, generate DDL scripts, and review various aspects of the database. RESTful Web Services provides stateless access to data and logic, through the use of SQL and PL/SQL using the Oracle APEX Listener 2.0 or later.
- **Team Development** - to manage software development projects using integrated features, to-dos, bugs, milestones and feedback.
- **Packaged Apps** – to install a broad collection of point solutions with just a few clicks.

You should always download the latest version of Oracle Application Express from the Oracle Technology Network (OTN). For more information about Oracle Application Express, go to:

<http://www.oracle.com/technetwork/developer-tools/apex/overview/index.html>

With Oracle Application Express you can easily build applications that report on database data. Reports can be hypertext linked with other reports allowing users to navigate through database data in the same way they navigate Web sites. Columns in reports can be easily linked to other reports, charts and data entry forms. An extensive charting engine allows SQL queries to be represented graphically. Oracle Application Express is also very adept at editing database data and supports a large number of declarative form controls including radio groups, checkboxes, select lists, shuttles, text editors and date pickers.

Many organizations have experienced the proliferation of workgroup databases. These databases are created primarily because users need immediate solutions that they can control. Departments continuously create decentralized databases to solve tactical issues. Costs are driven up as hundreds of inefficient, ad-hoc databases are deployed. These databases often contain business critical data. They tend to have issues with security, sharing, data accuracy, and high availability. Unfortunately, users of these databases usually end up working against their IT organization.



The Oracle Application Express consolidated model provides decentralized development with centralized management. Customers can consolidate workgroup data and the application development service into Oracle Database 12c with Oracle Application Express. Departments maintain full control over application development while their IT organization provides professional data management. End users experience dramatically improved data access via modern Web based applications. Oracle Database 12c ensures that Oracle Application Express applications are secure, reliable and scalable. The browser-based design time interface, declarative programming framework and simple wizards make Oracle Application Express a natural replacement for multi-user workgroup database applications such as Microsoft Access.

SQL-aware application developers with little or no Web development experience can easily create database applications with Oracle Application Express. You don't need to learn scripting languages or complex deployment frameworks, you simply write a few queries and choose from the provided set of form controls to create highly professional, secure, and scalable applications.

### **Oracle Application Express 5.0 Key Features**

Oracle Application Express 5.0, which is included with Oracle Database 12c Release 2, contains the following key features.

#### **Page Designer**

Page Designer is a modern, intuitive and exceedingly powerful browser-based IDE. It is a completely new IDE designed to greatly improve developer's productivity, allowing you to very quickly develop and maintain your Application Express applications. Page Designer provides better visual representation of your application pages, and provides an entirely new way to develop pages on the fly, via 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.

#### **Universal Theme**

Universal Theme is an all-new user interface for your applications that has been built from the ground up for Application Express 5. It is a simpler, yet more capable theme that does away with excessive templates and enables effortless customization using the built-in Theme Roller and Template Options. Universal Theme aims to empower developers to build modern, responsive, sophisticated applications without requiring expert knowledge of HTML, CSS, or JavaScript. You can build fully responsive applications that work flawlessly across a variety of screen sizes and form factors. Universal Theme includes key enhancements such as enabling complex form layout and using flexible grids.

#### **Interactive Reports**

Interactive Reports are very simple to write "SELECT \* FROM My\_Favorite\_Table", but provide a very powerful reporting capability. End Users can manipulate the data by selecting columns, sorting, filtering, adding computations, highlights, aggregates and more. Then they can save the reports, download them or subscribe to the report. Interoperate with Spreadsheets

Application Express developers can readily build a wizard to allow end-users to upload spreadsheet data into an existing table. Rather than relying on the IT department to upload data into an existing table, this feature allows the end users to step through a simple wizard and upload data into an existing table, allowing end users to be more self-sufficient.

Alternatively, developers and power-users can easily create a complete CRUD application using the Create Application Wizard based on a spreadsheet. So instead of sending out a spreadsheet, users can now build an



application based on that data in minutes. They can then send out a URL so there is only a single source of truth and the data is securely stored in Oracle Database 12c .

Interactive Reports have been rebuilt from the ground up in Application Express 5, to enhance both developer and end-user capabilities. Now Interactive Report regions can be restyled using CSS, in a similar manner to other regions within Application Express. The previous limitation of only allowing one Interactive Report on a page has now been lifted, allowing developers to define any number of reports on a page. Each report is a separate, customizable region. End-users can now define pivot reports directly within an Interactive Report. Instead of end-users downloading data from Application Express, and importing it back into Excel, just so they can perform a pivot on the data, they can now declaratively define a pivot on columns within their report.

### **Charting Capabilities**

Oracle Application Express includes a wide variety of “Flash-preferred” or HTML5 charts to visually show your data. When defined as “Flash-preferred”, if Flash is not available, such as on an iPad, then the chart will be rendered using HTML5. Maps can show country or state aggregates and you can also define project and resource Gantt charts.

### **Dynamic Actions**

Incorporate rich client-side interactivity declaratively using Dynamic Actions. This feature enables developers to declaratively define client-side behaviors without needing to know JavaScript and AJAX. Developers use the simple wizards and declarative constructs to specify when to fire, what to do, and what to operate on. Replacing hand crafted JavaScript and AJAX with declarative definitions greatly improves the quality, consistency, and manageability of rich client-side interactivity.

### **Plug-Ins**

This feature enables development of and the ability to share custom item types, region types, processes, dynamic actions, plus authentication and authorizations schemes. This dramatically broadens the reach of Oracle Application Express applications and provides a library of custom features for Oracle Application Express. When developers require functionality not available with native components, this architecture allows them to extend their applications in a manner that is both supported and maintained. There are a number of plug-ins available from the Application Express development team and also over 120 available on APEX community sites.

### **Modal Dialogs**

Now you can easily define modal (and non-modal) pages complete with the ability to utilize standard page processes. No longer will you need to hack a page using JavaScript and performing a lot of hand-crafting. Instead just set the display type and appropriate template and let Application Express take care of the rest.

### **Calendar**

The new calendar component comes with 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.

### **Mobile Reporting**

You have the ability to declaratively define mobile applications and mobile application components including HTML5 charts, HTML5 item types, and mobile calendars. You can develop applications with both desktop and mobile user interfaces together with automatic detection. The mobile applications are built using jQuery Mobile. Instead of building separate applications for different mobile operating systems (for example iOS, Android, Blackberry, and Windows), the same application can be run on any mobile device using a mobile browser.



Development of mobile applications uses the same builder and declarative framework as used for developing desktop applications. Therefore, mobile applications can be built very rapidly using the same SQL and PL/SQL skill set.

With Application Express 5 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 to displaying multiple lines on very small screens. Column toggle allows you to specify the most important columns, and those which will be hidden as necessary on smaller screens.

### **Packaged Applications**

Oracle Application Express includes a number of packaged business applications, which are available free of charge, and are supported by Oracle. The productivity applications enable customers to improve their business processes and immediately start utilizing their Oracle Database 12c investment. There is also a collection of sample applications that have been designed to help developers quickly understand specific capabilities of Application Express.

The productivity applications are locked by default and can't be modified by developers. If you wish to modify a packaged application or simply review its implementation and learn best practices, you can easily unlock it. Once unlocked, the application is no longer supported by Oracle and can no longer be updated. You can always return to a supported version by removing the unlocked copy and re-installing the locked version.

### **Websheets**

Websheets are a different class of application development within Oracle Application Express, lowering the bar even further to manage data in an Oracle Database from a Web browser. Websheets are designed for business users rather than developers. They combine design and runtime into a common environment and is similar in operation to WIKIs. Using only a Web browser, end users can define pages, data grids and reports and then select the community that can see and edit the content.

The major difference between a WIKI and Websheets is that once you add data to a WIKI it becomes stale and dated, whereas with Websheets you can incorporate data elements directly into your pages.

Within a Worksheet you can define data-grids ,which are spreadsheets stored in the database, define reports on tables, or write SQL directly against tables within your Oracle Database 12c schema.

Whenever a user accesses the Worksheet page the data is now queried directly from the database.

## Oracle REST Data Services

Oracle REST Data Services (ORDS) is a mid-tier technology which serves up RESTful Services for the Oracle Database. ORDS enables you to transform your Oracle Database into a RESTful API Service that today's modern application developers will find easy to use. ORDS receives REST requests and marshalls those to your Oracle Database, running SQL or PL/SQL code blocks, and returns any output as a JSON collection back to the calling application. With ORDS, any database resource can be made available via REST.

The data access APIs can fully exploit all the power of Oracle database technology to deliver the highest levels of reliability, security, and performance. Application developers can use these data access APIs to build enterprise class applications employing the same methods and techniques that have become dominant in the startup world for over a decade, at universities where new developers are trained, at the leading web companies like Google, Facebook, and many more, and the most advanced mainstream enterprise IT departments.

With ORDS and REST developers can:

### 1. Choose the programming language they want to use

Years ago, leading IT departments built everything in either Java or .Net. No more. Today's world is much more diverse. Javascript is as widely used today as is Java. Other languages such as Python, PHP, Ruby, etc. continue to be very popular. New languages such as Go from Google and Swift from Apple have gained substantial market share. Having one API that supports this entire diverse world of application development languages is extremely valuable and a necessity for the emerging polyglot and microservice architectures.

### 2. Use native language capabilities with no client drivers or complicated proprietary networking,

REST is native to all programming languages. Unlike SQL access, there are no special drivers to install and maintain, or complicated, proprietary networking to wrestle with. REST is totally based on Internet technologies, in particular HTTP, which all programming languages support today.

### 3. Access the database the same way they access all other external services

REST is by far the dominant technology used today by application developers to access external services outside of their programs. With ORDS and REST the database can be accessed in exactly the same way. IT shops can make immediate use of application developers straight out of university, or from the startup world, or from the many web companies that only use REST, or from leading enterprise IT shops with no additional, demanding training to learn about special client drivers, proprietary networking, or SQL. You also don't need to deal with the resistance you may get from developers who do not wish to acquire these skills.

### 4. Use JSON for easy integration with programming languages and agile development

Databases have been difficult for programming languages to integrate with. Commonly referred to as the "impedance mismatch", it was hard to map relations in databases to objects in programming languages using SQL. ORDS and the new SQL/JSON functions in Oracle Database 12c have eliminated this problem by enabling all data exchange between the application program and the database to be done in JSON which is readily mapped to/from programming language objects. (JSON is the Object Notation used by JavaScript and is very similar to the object notation used by other programming languages.)

With Oracle Database 12c, data can also be stored as JSON in the database. This allows developers to employ new, agile, "schemaless" or "schemalast" development methodologies that enable small teams of developers to safely operate with more independence and release new functionality faster and in smaller increments.

## 5. Build everything Cloud first

With ORDS and REST everything is built cloud first. REST uses the HTTP protocol which is at the core of the Internet and the key “stateless” technology that gives the Internet its simplicity, reliability, and scalability. No adaptors are needed. No tunneling. It is entirely native to the Internet. ORDS and REST are also entirely suitable for on premise and mixed hybrid cloud deployments as well.

### How ORDS Builds REST APIs

ORDS provides tools for automatically generating basic data access REST APIs for individual tables and views and manually generating more advanced APIs, e.g., for executing stored procedures and for accessing nested data in multiple tables.

This includes tools for defining the URI to identify the REST resource and route REST HTTP calls, e.g., GET (query), POST (insert), or PUT (Update), to the SQL or PL/SQL handler routine that processes the request. Parameters are passed in a variety of ways to/from the handler routine. These include using JSON, or other formats for data objects like images, in the body of the call, e.g., in a POST or PUT. Parameters call also be passed in the URI using query strings or route patterns. Results are returned to the calling application in JSON, or optionally in a CSV format.



Figure 1: What ORDS Does

ORDS will auto-generate handlers to perform basic CRUD operations on single tables or views. These include query all rows, query individual row, insert single row, bulk insert, update, and return metadata.

Handlers can also be manually defined using SQL or autonomous PL/SQL blocks to perform more advanced functions. This can include stored procedure calls with input and output variables. It can also include mapping nested JSON documents to/from hierarchical relations using the SQL/JSON functions in Oracle Database 12c.

### How ORDS Works with new SQL/JSON Functions in Oracle Database 12c

ORDS REST API handlers can make full use of the new SQL/JSON functions available in Oracle Database 12c. For example, handlers can use the `json_table` function to insert a JSON purchase order document containing a nested array of purchase order items into `PurchaseOrder` and `PurchaseOrderItems` tables. For the reverse operation, the `json_object` and `json_arrayagg` functions can be used to return a JSON purchase order object containing a nested array of purchase order items by querying `PurchaseOrder` and `PurchaseOrderItem` tables.

ORDS and REST can also be used to access JSON stored in Oracle Database 12c in the separate JSON Document Store facility in Oracle Database 12c using the Simple Oracle Document Access or SODA protocol or in columns within regular relations or tables. Many other useful SQL/JSON functions are also available such as the `IS JSON` function can be used as a database constraint to ensure that any JSON stored in the database is in a proper JSON format.

## ORDS Architecture

ORDS is a Java application that runs in Java application servers. It runs in a standalone mode, using Jetty, that is ideal for development and testing. It can also be used in production deployment that doesn't require strong application server management capabilities. For more demanding production deployments, ORDS runs in Weblogic, Tomcat, or Glassfish.

ORDS supports Oracle Database 10gR2 or above. It also supports the SODA protocol with the JSON Document Store in Oracle Database 12c. ORDS can also be used to provide REST API services for the Oracle NoSQL Database.

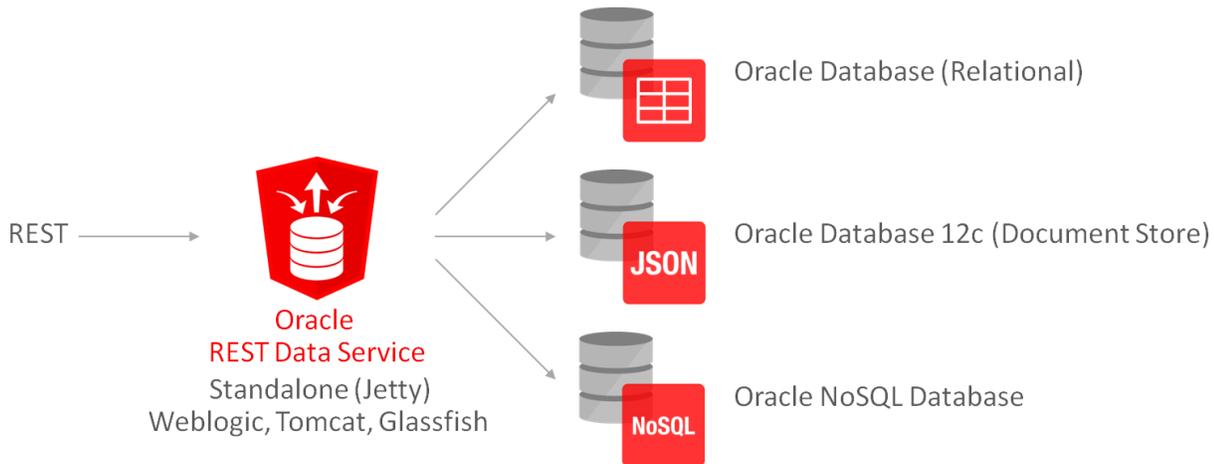


Figure 2: ORDS Architecture

## ORDS Security

For security, ORDS supports OAuth2 which is the IETF standard for securing access to REST APIs. This includes both server to server (two legged) and end-user to application (three legged) forms for third party applications. ORDS also supports first party authentication where requests originate from the same origin as the REST service. And, ORDS supports external authentication with Identify Managers, single sign on mechanisms, and other facilities.

ORDS is also secure because it runs in application servers such as Weblogic and Tomcat. It is protected by the security facilities these enterprise class application servers provide.

ORDS also does not allow direct database access from client applications. Client application can only make calls to REST APIs defined by specifically authorized Data Access Developers with high levels of database skills to ensure the APIs are secure.

## Conclusion

ORDS enables you to transform your Oracle Database into a RESTful API Service that today's modern application developers will want to use. Developers with database skills, such as SQL, can build data access REST APIs to the database that:

- » Fully exploit advanced database capabilities for maximum reliability, security, and performance

- 
- » Are generated automatically for single tables and views or easily generated manually for more advanced functions.
  - » Use JSON for data exchange between applications and the database
  - » Access JSON stored in the database to facilitate use of agile development methodologies
  - » Are protected by OAuth2 and other advanced security mechanisms
  - » Are fully Cloud ready

Application developers can use these data access REST APIs with any programming language, natively without client drivers or complex proprietary networking. Application developers don't have to acquire SQL and other advanced database skills. They call these data access APIs in the same manner that they access other external services, i.e., with REST and JSON.

For more information on ORDS see the ORDS webpages on the Oracle Technology Network [here](#) (or just Google "ORDS"). These webpages include a "Downloads" tab section where you can download and try the product, a "Documentation" tab section which has all of the ORDS documentation, a "Community" tab section which include the ORDS forum where you can get your technical questions answered and links to various blogs on ORDS related topics by the ORDS Product Manager and senior ORDS Developers, and a "Learn More" tab section that provides links to the highly recommended ORDS Hands-On Lab (HOL) and the ORDS Video Channel on YouTube. The ORDS HOL enables you to download a VM with all the software needed to try out ORDS on your laptop including exercises that walk you step-by-step through the key ORDS functionality. The ORDS YouTube Video Channel currently includes an ORDS overview video and several quick briefs (<10 minute videos) on key ORDS topics such as installation, security, auto enablement, and multiple datastore support.

## JSON

Modern applications developers are choosing to store application data as documents instead of using entity relationship models backed by relational storage. The primary driver for this switch is the flexibility offered by using JSON or XML based storage. This flexibility has enabled the application developer to be much more responsive to the needs of the business, as changes to the application data model no longer require changes to the database schema. This allows applications to be deployed and updated on a much faster cycle. The switch to document-based persistence has led to the adoption of NoSQL document stores for data persistence.

Oracle Database 12c has been engineered to provide full support for this style of application development. The Simple Oracle Document Architecture (SODA) specification, introduced as part of Oracle Database 12c describes an extremely simple API that allows the Oracle database to be used as a JSON document store. The SODA API provides support for creating and dropping document collections, create, retrieval, update and delete (CRUD) operations on documents, List and Query by Example (QBE) operations on document collections and various ancillary operations such as bulk insert and indexing. SODA allows application developers to create and deploy applications that manage data using JSON documents without any knowledge of SQL, JDBC and without requiring any assistance from an Oracle DBA. In addition to introducing SODA the database itself is now capable of enforcing JSON validity, indexing JSON content, and using these indexes to optimize operations on JSON content.

Choosing SODA allows application developers to get all the benefits of JSON based persistence without losing any of the benefits of Oracle's data management platform. It allows organizations to adopt NoSQL style development without introducing the complexity of having to manage multiple databases. They can continue to rely on the Oracle Database to provide them with high availability, scalability, security and recovery.

The other major benefit of choosing to use Oracle Database 12c as a NoSQL-style JSON document store is you still have all the power of SQL when you need it. Application developers can create and deploy their applications without any knowledge of SQL using Query-by-Example techniques to query the application data. However when it becomes time to use the data captured by the application in ways other than were envisaged by the application developer (ad-hoc queries) or perform reporting or analytics on the information contained in the JSON documents Oracle Database 12c allows SQL to be used for this purpose.

Oracle Database 12c extends the SQL language allowing JSON documents to be queried as part of SQL operations. These extensions allow the full power of SQL to be applied to the content of your JSON documents in a simple and straightforward manner. They also enable join operations between JSON documents and join operations between JSON documents and all the other kind of content managed by the Oracle Database, including relational data, XML content, Spatial Content, Semantic Content and Text Content.

Oracle database 12c Release 2 for Oracle Cloud also introduces the Oracle Data Guide for JSON, an exciting new feature that helps with understanding the structure of the JSON documents the database is managing. The Oracle Data Guide for JSON dynamically tracks the structure of JSON documents allowing you to easily generate relational views over your JSON documents that enable programmers and tools that have no understanding of JSON to work directly with your JSON documents.

Oracle Database 12c Release 2 extends the databases' support for JSON by adding support for

- » Generating JSON document directly from relational data
- » Partial update operations on JSON documents allowing a programmer to change the content of specific parts of a JSON document.
- » Oracle In-Memory database and Oracle Exadata to optimize query operations on JSON documents
- » Using Oracle Spatial to query Geo-JSON objects embedded in JSON documents

## .NET

Oracle offers four components that simplify .NET development with Oracle Developer Tools for Visual Studio, Oracle Data Provider for .NET (ODP.NET), Oracle Database Extensions for .NET, and Oracle Providers for ASP.NET. There is no charge for these components and they are easy to use for either the novice database programmer or the advanced programmer. These products fully support existing Microsoft standards, such as Entity Framework, and tools, such as Visual Studio, while also exposing powerful Oracle Database enterprise technologies, including Oracle Real Application Clusters (RAC) and Oracle Data Guard.

The Oracle Developer Tools for Visual Studio (ODT) is a tightly integrated "Add-in" for Microsoft Visual Studio. ODT is free and it is available for Visual Studio 2015, Visual Studio 2013, and Visual Studio 2012.

ODT makes developing .NET code for Oracle easy and fast, allowing developers to stay in Visual Studio for the entire development lifecycle. ODT makes it easy to browse and edit Oracle schema objects using integrated visual designers and can automatically generate .NET code via a simple drag and drop. Developers can easily modify table data, execute Oracle SQL statements, edit and debug PL/SQL code, automatically generate SQL deployment scripts using schema comparison tools and then check them into source control. Integration with Oracle Multitenant makes it easy for developers to clone databases and perform testing. The integrated context sensitive online help, including the Oracle SQL and PL/SQL Users Guides, puts Oracle Database documentation at your fingertips.

For more information about Oracle Developer Tools for Visual Studio, go to:

<http://www.oracle.com/technetwork/developer-tools/visual-studio>

ODP.NET features optimized data access to Oracle databases from a .NET environment. ODP.NET allows developers to take advantage of advanced Oracle Database functionality, including RAC and Data Guard; self-tuning and caching; XML DB and REF Cursors; and security features. ODP.NET gives programmers optimal performance, flexibility, and feature choice for .NET Framework and Oracle Database development. With it, developers bring Oracle's powerful data management capabilities to .NET. ODP.NET natively supports .NET Framework 3.5 and higher.

For more information about ODP.NET, go to:

<http://www.oracle.com/technetwork/topics/dotnet/index-085163.html>

Oracle Database Extensions for .NET is a feature of Oracle Database 12c on Windows that makes it easy to develop, deploy, and run stored procedures and functions written in a .NET managed language, such as C# or VB.NET. .NET stored procedures or functions are developed using Microsoft Visual Studio and deployed using the tightly integrated .NET Deployment Wizard, which is a feature of the Oracle Developer Tools for Visual Studio. After deployment, a .NET stored procedure can be called from within .NET application code; from SQL or PL/SQL; from another .NET, PL/SQL, or Java stored procedure; from a trigger; or from anywhere else a stored procedure or function call is allowed.

For more information about Oracle Database Extensions for .NET, go to:

<http://www.oracle.com/technetwork/topics/dotnet/index-085095.html>

Oracle Providers for ASP.NET are a collection of data services that follow the ASP.NET provider model and use Oracle Database as the data source. Developers store web application state, such as shopping cart or user information, in a persistent Oracle database. ASP.NET developers are now more productive as they build their Web applications through ASP.NET services and controls that are part of the .NET Framework. By simply configuring the Oracle Providers for ASP.NET as default providers in a configuration file, ASP.NET applications can store various



types of application states in an Oracle database. Oracle offers the following providers: Membership, Role, Site Map, Session State, Profile, Web Event, Web Parts Personalization, and Cache Dependency.

For more information about Oracle Providers for ASP.NET, go to:  
<http://www.oracle.com/technetwork/topics/dotnet/index-087367.html>

## .NET Key New Features

### Application Continuity

Oracle Application Continuity enables database requests to automatically replay transactional or non-transactional operations in a non-disruptive and rapid manner in the event of a severed database session, which results in a recoverable error. Application Continuity improves end-user experience by masking planned and unplanned related errors. Applications can be developed without complex logic to handle exceptions, while automatically replaying database operations upon a recoverable error.

ODP.NET, Unmanaged Driver 12.2 includes new support for Application Continuity, which automatically masks and recovers from many system failures, communication failures, hardware failures, and storage outages, all transparent to the end user. Applications can add ODP.NET Application Continuity to their existing applications without having to make .NET code changes, simplifying feature adoption.

### Sharding

Oracle Sharding is a shared-nothing architecture that allows near-linear scaling of the database across low-cost commodity database servers located in one or more local or global data centers. Sharding's benefits include global data distribution (store particular data close to consumers) and fault containment (failure of one shard does not affect the availability of other shards). Global Data Services manages the location of data among the shards and allows ODP.NET client requests to be routed to the appropriate shard in this distributed database system.

Starting from Release 12.2, ODP.NET, Unmanaged Driver and Oracle Database support sharding. Oracle Sharding provides the ability to horizontally partition the data across multiple independent Oracle databases (shards). Based on a key specified in the connect string, ODP.NET can route the database requests to a particular shard.

### Cloud

Managed and unmanaged ODP.NET can be deployed easily to Oracle Cloud, private clouds, 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 application's .NET configuration files. Managed and unmanaged ODP.NET now share a unified configuration file format that can be used for on-premises and the cloud to make managing these configuration files easier and uniform.

On-premises ODP.NET can be easily configured to connect to Oracle Database Clouds (PaaS), including the new Oracle Database Exadata Express Cloud Service. ODP.NET applications can be deployed to Oracle Compute Cloud Service (IaaS) on Windows using Oracle Developer Tools for Visual Studio and Web Deploy. These cloud applications can be setup with Microsoft .NET Framework and Internet Information Services. To learn more, visit the [Using .NET applications and Visual Studio with Oracle Public Cloud](#) web page.

### ODP.NET, Managed Driver Enhancements

- » Data Integrity – ODP.NET, Managed Driver supports cryptographic hash functions to better ensure data integrity between the database server and the client. The algorithms supported include MD5, SHA-1, and SHA-2 (SHA-256, SHA-384, and SHA-512).

- 
- » Transport Layer Security (TLS) – ODP.NET, Managed Driver has added support for TLS 1.1 and 1.2 in addition to existing support for TLS 1.0 and SSL 3.0. ODP.NET, Managed Driver connections using SSL/TLS can now also ensure that the distinguished name (DN) is correct for the database server that it is trying to connect to.
  - » PL/SQL Boolean – ODP.NET, Managed Driver supports the *OracleBoolean* data type when using the database's PL/SQL Boolean data type. Booleans store TRUE or FALSE values. The ODP.NET *OracleBoolean* data type eases parameter binding and data type mapping setup with Boolean values.

#### **Longer Schema Identifiers**

ODP.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 managed and unmanaged ODP.NET.

## Globalization Support

Oracle Database 12c delivers enriched globalization support by introducing a set of new features that facilitate the deployment of databases in the Unicode® character set and the development of multilingual, standards-compliant enterprise applications. Oracle recommends a Unicode-based system architecture which enables the storage, processing, and retrieval of character data in any languages. The new built-in database capabilities offer enhanced usability and industry compatibility in building a complete Unicode solution that meets your business requirements.

### Globalization Support New Features

#### Unicode 6.1 Support

Unicode Standard defines the universal character set for encoding characters used in most of the writing systems of the world. It provides a uniform representation of textual information independent of platform or programming language. Oracle has been supporting the Unicode character sets since Oracle 7. In Oracle Database 12c, this support has been updated to include version 6.1 of the Unicode Standard.

#### Unicode Collation Algorithm (UCA) Conformance

Unicode Collation Algorithm (UCA) is a Unicode standard for determining the linguistic order of Unicode strings. The UCA defines a Default Unicode Collation Element Table (DUCET) that supplies a reasonable default collation for all Unicode characters. The DUCET is also customizable to accommodate the special ordering of specific languages. The UCA is fully compatible with the international collation standard ISO 14651 but offers extended features and flexibility in the collation behavior.

Oracle Database 12c introduces UCA support in addition to the existing database monolingual and multilingual linguistic collations. Oracle's implementation of UCA is compliant with Unicode Standard 6.1. The main features include:

- » Full collation ordering based on Unicode 6.1 DUCET 23
- » Multilevel comparison algorithm up to 4 collation levels
- » Configurable options for sorting variable weighting characters (spaces, punctuations, symbols)
  - » Blanked
  - » Non-ignorable
  - » Shifted
- » 12 tailored language-specific UCA collations for Spanish, Traditional Spanish, Canadian French, Danish, Thai, Simplified

The linguistic operations involve transforming the character data into binary values called collation keys before evaluating the relative order. As the collation keys are represented in Oracle with the RAW data type, you can now sort longer text with higher precision in Oracle Database 12c since the maximum length limits of the VARCHAR2, NVARCHAR2, and RAW data types have been extended to 32767 bytes.

#### New Locale Coverage

As part of the continued effort to expand the scope of globalization support and address fast evolving customer requirements, Oracle Database 12c has introduced a set of 12 new languages and 32 new territories to supported database locales, covering additional regions of Asia, Africa, Americas, and Europe:

- » New languages – Amharic, Armenian, Dari, Divehi, Khmer, Lao, Latin Bosnian, Maltese, Nepali, Persian, Sinhala, Swahili

- 
- » New Territories – Afghanistan, Armenia, Bahamas, Belize, Bermuda, Bolivia, Bosnia and Herzegovina, Cambodia, Cameroon, Congo Brazzaville, Congo Kinshasa, Ethiopia, Gabon, Honduras, Iran, Ivory Coast, Kenya, Laos, Maldives, Malta, Montenegro, Nepal, Nigeria, Pakistan, Paraguay, Senegal, Serbia, Sri Lanka, Tanzania, Uganda, Uruguay, Zambia

Moreover, it also includes the support for Ethiopian calendar, a calendar system based on the Coptic calendar with a 13th month of either 5 or 6 days in length.

### **Database Migration Assistant for Unicode (DMU)**

Migrating to the Unicode character set is an intricate process that involves many different operational aspects which can be both time-consuming and resource intensive. Any misstep along the way can lead to data loss and serious business consequences. Oracle Database Migration Assistant for Unicode (DMU) is a next-generation migration tool that streamlines the entire migration process with an intuitive GUI to minimize the DBA's manual workload and decision-making. It helps ensure all migration issues are addressed beforehand and the conversion of data is carried out correctly and efficiently. The DMU migration workflow covers:

- » Enumeration - auto-identification of database objects containing textual data that requires conversion <sup>24</sup>
- » Oracle White Paper— Application Development with Oracle Database 12c
- » Scanning - comprehensive assessment of migration feasibility and discovery of potential data issues
- » Cleansing - sophisticated toolsets for iterative data analysis and cleansing to ensure data safety
- » Conversion - automated in-place data conversion to minimize time and space requirements

The DMU also goes beyond helping migration to the Unicode character set by providing the capability to conduct ongoing health check of your post-migration database to maintain data compliance with the Unicode Standard. Even with a database that uses the Unicode character set, incorrectly configured applications may introduce invalid character codes into the database and cause data corruption. The DMU's Validation Mode feature can help expose the source of the issues and detect data problems before these issues are noted by end-users.

The DMU was first released in April 2011 on OTN as a free downloadable product. The latest DMU version 1.2 is bundled with Oracle Database 12c and is the officially supported method for migrating databases to the Unicode character set. The DMU also supports migrating selected prior database releases of 10.2, 11.1, and 11.2. The legacy command-line utilities CSSCAN and CSALTER have been de-supported.

An added benefit of performing the migration in Oracle Database 12c is that the new extended VARCHAR2 type limit of 32767 bytes guarantees any VARCHAR2 column with the pre-12.1 limit of 4000 bytes or less can just be lengthened to accommodate longer values resulting from data expanding in conversion to Unicode. Neither truncation nor migration to the CLOB data type is now necessary for such columns.



## Conclusion

Oracle Database 12c delivers a highly productive and powerful set of application development tools supporting the most popular development technologies including Oracle APEX, PL/SQL, SQL, C, C++, Java, .NET, Node.js, PHP, Perl, Python and Ruby. Oracle Multitenant and Oracle Database Cloud Service make it easy for application developers to create applications for the Cloud. Transaction Guard and Application Continuity take application availability and reliability to an extraordinary high level.

Migrating to Oracle Database is easier than ever with the addition of 32K VARCHAR, NVARCHAR, and RAW datatypes and Auto-Increment (IDENTITY) columns. The SQL Translation Framework and the Driver for MySQL applications greatly reduce the amount of effort required to migrate applications to Oracle Database. Other migration support features include: Enhanced SQL to PL/SQL Bind Handling, and Native SQL Support for Query Row Limits and Row Offsets.

Oracle Application Express is a unique Web application development tool that is ideal for rapidly building desktop and mobile applications using Oracle Database. Oracle SQL Developer Data Modeler supports logical or conceptual modeling (including multi-dimensional modeling for Business Intelligence), relational database modeling and the final detailed physical implementation. Oracle SQL Developer provides a graphical work environment for the Oracle Database that increases database developer productivity and eases migrations from other databases to Oracle Database.

Oracle Database's .NET components support the latest .NET features including Entity Framework and Language Integrated Query as well as Oracle Database 12c features such as Oracle Multitenant and Transaction Guard.

The new globalization support features in Oracle Database 12c enable the development of database applications that conform to the Unicode Standard 6.1. The Unicode Collation Algorithm (UCA) implementation supports the industry standard multilingual collation with flexible capabilities. The additional database locale coverage further expands the application localization support to provide behaviors that match local users' cultural conventions. The Oracle Database Migration Assistant for Unicode (DMU) significantly reduces the downtime, lowers the costs, and simplifies the tasks of migrating databases to the Unicode character set.

Oracle's support for JSON as a first class data type allows the Oracle Database to be used as a JSON document store. This allows the Oracle Database to service the needs of the Next Generation Application Developers, who have chosen to use JSON as their primary tools for preserving application data.

Collectively these tools simplify your development tasks, reduce costs and enable your organization to reduce application development time. Oracle Database 12c introduces new and innovative features which allow you to create secure, highly available, high performance applications.



**Oracle Corporation, World Headquarters**

500 Oracle Parkway  
Redwood Shores, CA 94065, USA

**Worldwide Inquiries**

Phone: +1.650.506.7000  
Fax: +1.650.506.7200

CONNECT WITH US

-  [blogs.oracle.com/oracle](http://blogs.oracle.com/oracle)
-  [facebook.com/oracle](http://facebook.com/oracle)
-  [twitter.com/oracle](http://twitter.com/oracle)
-  [oracle.com](http://oracle.com)

**Hardware and Software, Engineered to Work Together**

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.0115

White Paper Title  
January 2015  
Author: [OPTIONAL]  
Contributing Authors: [OPTIONAL]