

An Oracle White Paper
October 2010

Application Development with Oracle Database 11g Release 2

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.

| | |
|---|----|
| Introduction | 1 |
| Edition-Based Redefinition | 3 |
| Oracle SQL Developer | 4 |
| Creating and Editing Objects | 5 |
| SQL Worksheet | 5 |
| PLSQL Editing and Debugging | 5 |
| Hierarchical Profiler | 6 |
| Reporting | 6 |
| Migrations (Migrating to Oracle Database) | 6 |
| TimesTen | 7 |
| Oracle Application Express Support | 7 |
| SQL Tuning | 7 |
| Version Control | 7 |
| Schema Copy and Compare | 7 |
| Later Releases | 8 |
| Unit Testing | 8 |
| Version Control | 8 |
| Integrated Data Modeler Viewer | 8 |
| Oracle SQL Developer Data Modeler | 9 |
| Oracle SQL Developer Data Modeler Layers | 10 |
| Importing | 10 |
| Formatting, Subviews and Displays | 11 |
| Reporting Repository | 11 |

| | |
|--|----|
| Oracle Application Express | 11 |
| Interactive Reports | 13 |
| Flash Charts | 14 |
| Dynamic Actions | 14 |
| Plug-Ins | 14 |
| PDF Printing | 15 |
| Oracle Forms Conversion | 15 |
| Microsoft Access Migration | 15 |
| Team Development | 15 |
| Security Enhancements | 16 |
| Cross Language Features | 17 |
| Client Query Result Cache | 17 |
| Server Query Result Cache | 18 |
| Database Resident Connection Pooling | 18 |
| IPv6 Support | 19 |
| Fast Connect Time Failover for Applications | 19 |
| Secure LOBs and Prefetching of LOB Data | 19 |
| REF CURSOR Prefetching | 20 |
| .NET | 20 |
| Visual Studio 2010 and .NET Framework 4 Support | 22 |
| Visual Studio Application Performance Tuning Tools | 22 |
| Visual Studio Application Lifecycle Management and Source Control Integration | 23 |

| | |
|--|----|
| Performance Enhancements..... | 23 |
| Oracle Streams Advanced Queueing..... | 24 |
| Promotable Transactions | 24 |
| XCopy Deployment | 25 |
| Edition-Based Redefinition | 25 |
| User-Defined Types | 25 |
| Bulk Copy | 26 |
| HA Event Notification and Callback | 26 |
| PHP..... | 26 |
| Database Resident Connection Pooling | 27 |
| Real Application Clusters | 27 |
| High Availability with FAN and RAC..... | 27 |
| Client Query Result Caching..... | 27 |
| Row and REFCURSOR Prefetching | 27 |
| Edition Based Redefinition | 28 |
| Client Identifier and Session Attributes | 28 |
| PL/SQL..... | 28 |
| Real Native Compilation..... | 28 |
| Function Result Cache | 29 |
| Sequences | 29 |
| PL/Scope..... | 29 |
| Hierarchical Profiler..... | 30 |
| PL/SQL Comes to Oracle TimesTen..... | 30 |

| | |
|--|----|
| Java..... | 30 |
| OracleJVM Performance, JIT Compiler | 31 |
| Universal Java Connection Pool | 31 |
| Productivity and Portability through Java Standards..... | 31 |
| JDBC 4.0..... | 31 |
| JDBC Enhancements..... | 31 |
| OracleJVM Compliance with IP V6 | 32 |
| RowSet (JSR-114) Comes to Server-side JDBC | 32 |
| Exposing Oracle to Java | 32 |
| Prefetch in 1st Roundtrip..... | 32 |
| Native AQ Protocol..... | 32 |
| Database/Query Change Notification..... | 32 |
| Server and Client Query Result Cache | 32 |
| Security | 33 |
| Oracle JVM Ease of Use..... | 33 |
| Manageability | 33 |
| Oracle Call Interface (OCI)..... | 33 |
| Oracle TimesTen In-Memory Database Cache 11g..... | 33 |
| Database Resident Connection Pool | 34 |
| Client Query Result Cache..... | 34 |
| Secure File LOBs | 34 |
| REF CURSOR Prefetching | 34 |
| Edition Based Redefinition | 34 |

| | |
|--|----|
| 64-bit Integer Host Datatype | 34 |
| Implicit Fetching of ROWIDs | 34 |
| Binding and Defining Multiple Buffers | 35 |
| Allocating an Array of Descriptors | 35 |
| Diagnosability | 35 |
| Runtime Connection Load Balancing | 35 |
| Client and Server with Different Versions of Time Zone Files | 35 |
| Conclusion | 35 |

Introduction

The modern IT organization is faced with a multitude of application development languages and tools to choose from. In some cases multiple application development products are used within a single organization. Organizations that have grown through mergers and acquisitions are faced with the difficult problem of managing and integrating a large collection of competing application development technologies that were used in predecessor organizations. Technology enables business innovation and those organizations that are able to quickly deliver applications gain competitive advantage.

Oracle Database 11g provides a single integrated platform offering security, high performance and scalability for the most popular technologies used by application developers today. Oracle Database 11g adds significant new capabilities to all the major application development environments, enabling you to increase developer productivity and shorten time to market.

Application developers have a choice of languages for creating Oracle database-centric applications including Java, Oracle Application Express, C, C++, COBOL, PL/SQL, Visual Basic, C#, PHP, Ruby, Python and Perl. Developers can choose the language in which they are most proficient or one that is most suitable for a specific task. For example an application might use Java in middle-tier to create dynamic Web pages, PL/SQL or Java to implement stored procedures in the database, and C++ or Java to implement computationally intensive logic in the middle tier. Oracle Database 11g offers customers the power and performance of the industry's leading database delivered across a wide variety of the most popular application development technologies.

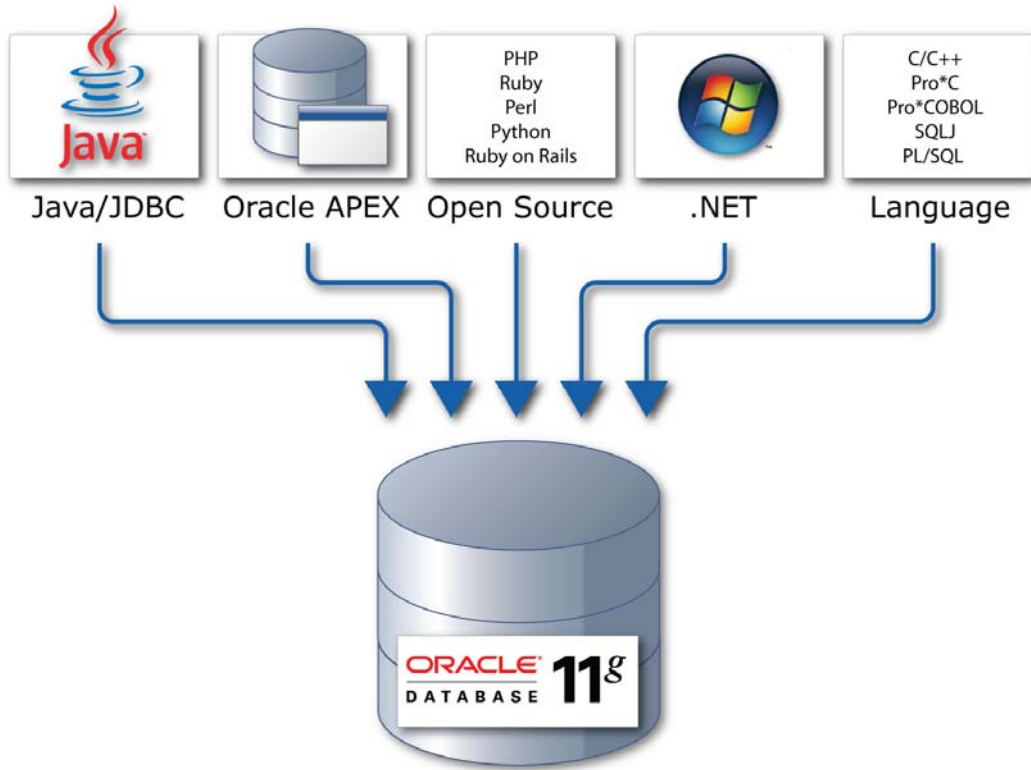


Figure 1. Oracle Database 11g Supports Popular Application Development Technologies

This paper provides an overview of application development features in Oracle Database 11g Release 2.

Edition-Based Redefinition

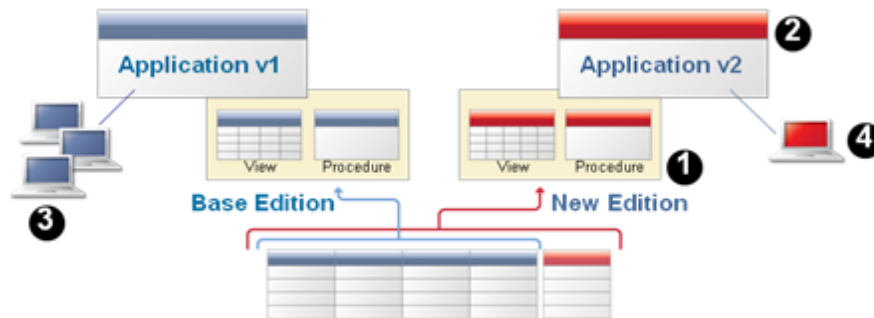
Large, mission critical applications are often unavailable for tens of hours while an upgrade to the database component is installed. This application downtime not only results in a loss of service to users but can also cost some organizations millions of dollars per year. Oracle Database 11g Release 2 introduces revolutionary new capabilities that allow online application upgrade with uninterrupted availability of the application.

Edition-based redefinition enables application developers and DBAs to upgrade an application with little or no application down time. A new database construct called an edition provides a privacy mechanism for installing new code and for making data changes so that the running production application does not see the changes. When all the required changes have been made in private, they can be made available to users. In support of edition-based redefinition, a new kind of view called an editioning view and a new kind of trigger called a crossedition trigger are introduced.

Every database has at least one edition. The DBA creates a new edition as a child of the existing one (see Figure 2). The changes are made in the child edition while you continue to use the parent edition. When needed, changes to data are made safely by writing only to new columns or new tables not seen by the old edition. Editioning views expose a different projection of each changed table into each edition to allow each to see just its own columns. Crossedition triggers propagate data changes made by the old edition into the columns of the new edition. When the installation of the changes is complete, some users start to use the new edition while other users continue using the old edition. Crossedition triggers propagate data changes made by the new edition into the columns of the old edition.

Edition-Based Redefinition

Online Application Upgrade



- ① Compile objects in new edition without affecting online users
- ② Deploy new application version referencing new edition
- ③ Current users are not effected, they reference base edition
- ④ Phase in new application version over time

Figure 2. Edition-Based Redefinition

A complete high availability database solution must address planned and unplanned changes to hardware and software. Edition-based redefinition complements other Oracle Database high availability technologies including Oracle Real Application Clusters, Oracle Database Flashback Technologies and Oracle Data Guard. When used together, these technologies protect against hardware failures and provide maximum availability through online database software upgrades, online database maintenance and online application upgrades.

For more information about Oracle Database 11g High Availability, go to:

<http://www.oracle.com/technetwork/database/features/availability/index.html>

Oracle SQL Developer

Oracle SQL Developer is a no cost graphical tool that enhances productivity and simplifies database development tasks. Designed for Oracle Database developers, Oracle SQL Developer simplifies development cycles and reduces the need to buy third-party tools for developing and debugging SQL and PL/SQL code. Using Oracle SQL Developer, users can browse, create and modify database objects, run SQL statements, edit and debug PL/SQL and can run reports from an extensive list of predefined reports or create their own. The introduction of this tool underscores Oracle's commitment to improving the productivity and supporting the needs of the database developer community.

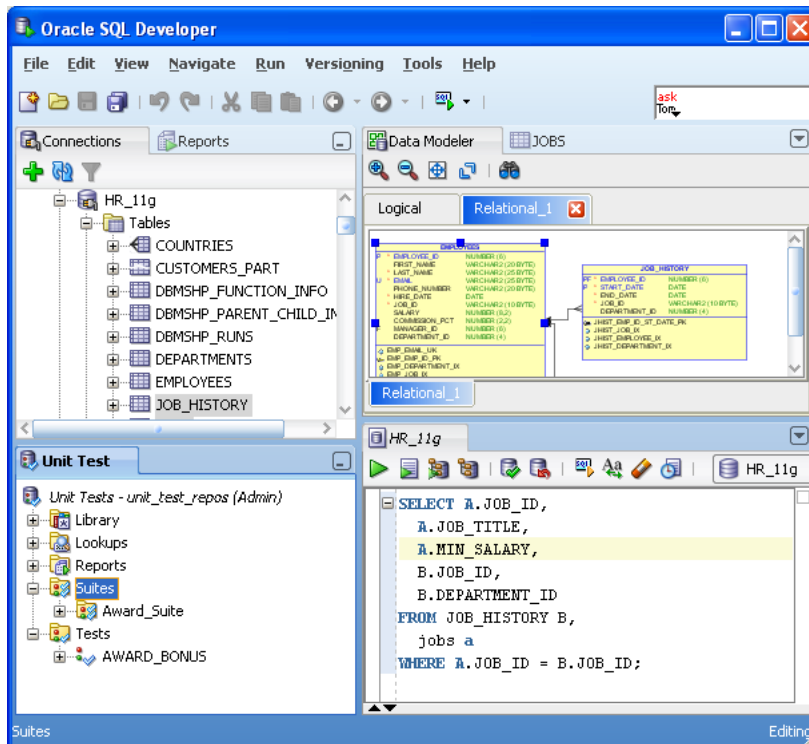


Figure 3. Oracle SQL Developer

Creating and Editing Objects

Users can connect to any target Oracle database schema using standard Oracle database authentication. Once connected, they can create, edit and update database objects. As new objects are created or existing objects edited, the DDL for those adjustments is available for review. An Export DDL option is available for users wanting to create the full DDL for one or more objects in the schema.

SQL Worksheet

The SQL Worksheet supports the creation of SQL, PL/SQL and SQL *Plus commands. These can be run individually or consecutively. A SQL History option makes recalling previous commands easy, while the Explain Plan option allows users to see the execution plan for selected statements.

PLSQL Editing and Debugging

Using this robust editing environment, users can create and edit PL/SQL, take advantage of the code formatting, add bookmarks and use code insight. When it comes to debugging PL/SQL, breakpoints, smart data, a debugger stack and watches are all available. These features allow the user to set a break point and run and test the code, supplying alternate data at runtime while

debugging. Creating PL/SQL in the editor or using the SQL Worksheet is made easier by the availability of snippets. Snippets are code fragments, such as SQL functions, optimizer hints or miscellaneous PL/SQL programming techniques, which users can drag onto the PL/SQL Editor or the SQL Worksheet.

Hierarchical Profiler

For users connecting to Oracle Database 11g, the PL/SQL Hierarchical Profiler provides the ability to identify points of contention, delays and hence the option to review and tune the code.

Reporting

Oracle SQL Developer provides a number of predefined reports about the database and its objects. Any report can be pushed over to a SQL Worksheet and manipulated further. Oracle SQL Developer allows users to create, save and share user-defined reports for repeated use. Report styles include charts and master-detail reports. Predefined data dictionary, ASH and AWR reports are supplied with Oracle SQL Developer.

Migrations (Migrating to Oracle Database)

Oracle SQL Developer includes an intuitive tool that enables users to migrate a third-party database, including the schema objects, triggers, and stored procedures, to an Oracle Database using a simple point-and-click process. Oracle SQL Developer currently supports migrations from IBM DB2 UDB, Microsoft SQL Server, Microsoft Access, MySQL, Sybase Adaptive Server and Teradata to Oracle. The supported releases of these databases are IBM DB2 UDB version 7, 8 and 9, Microsoft Access versions 97, 2000, 2002/XP and 2003, Microsoft SQL Server 7, 2000, and 2005 and MySQL versions 3, 4, 5, Sybase Adaptive Server versions 12 and 15 and Teradata 12.

The following migration features are included in Oracle SQL Developer:

- Full integration with Oracle SQL Developer means that users have one tool to browse third-party databases, migrate selected objects to Oracle and then work with them.
- Quick Migration Wizard offers least privilege migration i.e. No DBA privilege is required.
- Step driven migration offers control at each stage of the migration process.
- Fine grain migration support provides users with the ability to select specific objects for migration.
- Complex object migration supports stored procedures, triggers and views.
- Translation Scratch Editor offers single statement migration.
- Translation Difference Viewer provides side-by-side comparison by matching statements and blocks.

TimesTen

Oracle SQL Developer can also be used to manage objects in both Oracle In-Memory Database Cache and Oracle TimesTen In-Memory Database.

Oracle Application Express Support

Oracle SQL Developer supports the management of objects for Oracle Application Express 3.0.1 and above. This feature was introduced in Oracle SQL Developer 1.2 and includes the ability to:

- Connect to and browse all Oracle APEX applications associated with a database schema (at application and page level).
- Run a number of reports that allow users to report on Workspaces, Applications, Pages and Schemas.
- Export and Import Oracle APEX applications using a context menu.

SQL Tuning

Oracle SQL Developer includes the following SQL Tuning capabilities:

- Display execution plans
- Display trace information using auto trace
- Hierarchical Profiler
- Real time monitoring
- Parse SQL Trace files

Version Control

Oracle SQL Developer provides tight integration with version control systems CVS and Subversion. Both version control systems require access to a repository or repositories. You should create connections for each of the available repositories. Source code is held and maintained within the repositories. Using Oracle SQL Developer, users can place files under version control and check them out to a local file system. Until the files are checked back in, they can be worked on locally and even discarded. On check-in, Oracle SQL Developer offers a compare and merge facility.

The Oracle SQL Developer Files navigator provides users the flexibility of being able to browse and open files, whether under version control or not.

Schema Copy and Compare

Working with multiple schemas often involves copying objects and their data from one schema to another. Oracle SQL Developer provides utilities for both copying and exporting objects

from one schema to another. The Schema Diff utility identifies differences across schema is useful for managing database objects across multiple environments (e.g., development and test).

Later Releases

Oracle SQL Developer 1.5.5 is shipped with Oracle Database 11g Release 2. Later Oracle SQL Developer releases are available and can be installed alongside the shipped release. For details and to download the later releases go to:

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

The following features are new to Oracle SQL Developer 2.1.

Unit Testing

Introduced in Oracle SQL Developer 2.1, PL/SQL unit-testing brings a powerful testing infrastructure to the product, by helping users create extensive unit tests and save them to a unit test repository. A repository based testing system means that users can run and rerun tests as required, whether interactively or using batch commands run from the command line. Single tests can be grouped together in a suite of tests, thus providing a power test bed for running regression tests for PL/SQL in an application.

Version Control

Oracle SQL Developer provides tight integration with Subversion version control system. In Oracle SQL Developer 2.1 there are additional extensions available for connecting to Concurrent Versions System (CVS), Serena Dimensions and Perforce. Once a connection is made to the version control system of choice, users can open files in Oracle SQL Developer and bring them under version control. Once under version control, files can be checked out, modified and checked back in.

Integrated Data Modeler Viewer

Oracle SQL Developer 2.1 provides an integrated Data Modeler Viewer. This extension to Oracle SQL Developer allows users to open data model designs created in Oracle SQL Developer Data Modeler. In addition to browsing these designs, users can also select a collection of tables in a schema and drag them onto a Data Modeler display, to create a relational model of the database objects.

Oracle SQL Developer Data Modeler

Oracle SQL Developer Data Modeler is a no cost graphical data modeling tool that facilitates and enhances communication between data architects, database administrators, application developers and users, and simplifies the data modeling development process itself. Using Oracle SQL Developer Data Modeler users can create, browse and edit, logical, relational, physical, multi-dimensional, and data type models. The generation of DDL scripts improves productivity and promotes the use of standards. Oracle SQL Developer Data Modeler is an independent and standalone product.

The introduction of this tool illustrates Oracle's commitment to improve developer productivity and to facilitate the improved quality and standard of the work done by the database developer community and their users. Oracle SQL Developer Data Modeler is designed for all database data modelers, from business architects to DBAs and from database developers to application developers. The role of Oracle SQL Developer Data Modeler is to simplify data modeling development tasks and serves as a powerful communication tool between developers and business users.

Oracle SQL Developer Data Modeler is a complete model-to-implementation solution for data related modeling, whether for

- New operational or business intelligence related databases
- Capturing existing database implementations to provide a graphical representation and related metadata for documentation
- Adding and implementing new data requirements

The Oracle SQL Developer Data Modeler audience ranges from data architects to database administrators and developers, all users who need a clear, consistent and understandable picture of the data and the underlying data properties.

For more information about Oracle SQL Developer Data Modeler, go to:

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

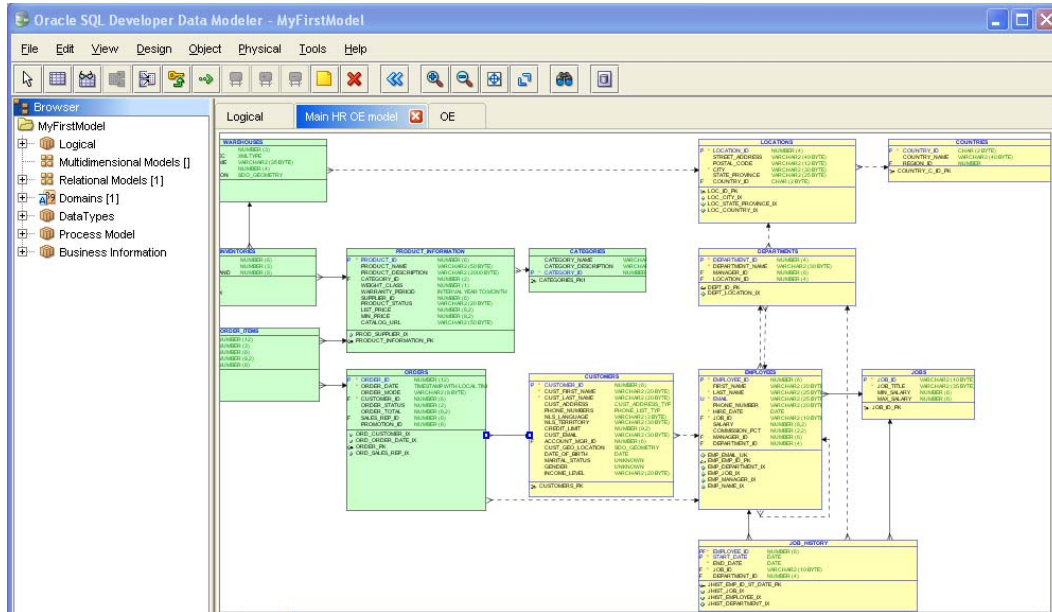


Figure 4. Oracle SQL Developer Data Modeler

Oracle SQL Developer Data Modeler Layers

Oracle SQL Developer Data Modeler model comprises three tightly synchronized layers: a logical model, relational models, and physical models. There is a one-to-many implementation when moving from the logical to the relational to the physical layer, which means that you can create multiple instances at the next, lower level in the hierarchy. Properties set at the upper layers are reflected and preserved at the lower levels. This is useful when an application has more than one physical deployment or customer-facing implementation to support.

Importing

Oracle SQL Developer Data Modeler can import models from various formats, including the metadata from a variety of sources. This allows an organization to take advantage of existing operational databases across current platforms and importing models that may have been built using other tools.

Oracle SQL Developer Data Modeler imports:

- Details from standard DDL scripts
- Schema and physical details directly from the database data dictionary
- Entity and schema model metadata from the Oracle Designer repository
- Multi-dimensional models using Cube Views or XMLA files

- CA ERwin Data Modeler models complete with diagram layouts and the name standardization and abbreviations files for naming conventions

Formatting, Subviews and Displays

Users can control colors, fonts and the dimensions of a single or collection of objects. A subview is a group of, often related, objects on a diagram. Any changes in the subview are reflected in the main model. A relational model subview is automatically created per database schema when several schemas are imported at once. Subviews make it easier to maintain larger models.

Reporting Repository

Oracle SQL Developer Data Modeler supports a reporting repository, allowing users to save and run SQL queries to gather details of the designs. A set of predefined reports is available as an extension to Oracle SQL Developer. Users can export their designs to the repository and use Oracle SQL Developer to run reports against their designs or write their own SQL query reports.

Oracle Application Express

Oracle Application Express (Oracle APEX), formerly called HTML DB, is a rapid Web application development tool for the Oracle Database. Using only a Web browser and limited programming experience, you can develop and deploy professional applications that are both fast and secure. No client software is required to develop, deploy or run Oracle Application Express applications. Oracle Application Express is built for the Web and combines the qualities of a workgroup database (i.e., productivity, ease of use and flexibility) with the qualities of an enterprise database (i.e., security, integrity, scalability and availability). Oracle Application Express is an ideal tool for quickly building opportunistic and departmental applications and is a “no cost” option of the Oracle Database.

Oracle Application Express provides four primary tools:

- Application Builder - to create dynamic database driven Web applications.
- SQL Workshop - to browse 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.
- Team Development - to manage software development projects using integrated features, todos, bugs, milestones and feedback.
- Administration – to manage the Application Express Workspace including managing the service, maintaining users, and monitoring activity.

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 11g with Oracle Application Express. Developers 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 11g 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 user interface themes and form controls to create highly professional, secure, and scalable applications.

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>

The screenshot displays the Oracle Application Express (APEX) workspace interface. At the top, the Oracle logo and 'Application Express' are visible, along with a user greeting 'Welcome DPEAKE (Logout)'. A navigation menu includes 'Home', 'Application Builder', 'SQL Workshop', 'Team Development', and 'Administration', with a search bar for the workspace. Below the menu, the 'Workspace APEX_PM' section features four main tool icons: Application Builder, SQL Workshop, Team Development, and Administration. A 'News' section is present below the workspace tools. On the right side, there are several utility sections: 'About' (describing APEX as a rapid Web application development tool), 'Site-Specific Tasks' (listing links to the discussion forum, OTN, and help), 'Available Updates' (showing the system is up-to-date), and 'Language' (with options for Deutsch, English, and Español).

Figure 5. Oracle Application Express

Interactive Reports

Interactive Reporting Regions enable end users to customize reports. Users can alter the layout of report data by choosing the columns they are interested in, applying filters, highlighting, and sorting. They can also define breaks, aggregations, different charts, group bys, and their own computations. Users can create multiple variations of the report and save them as named reports and download to various file formats including comma-delimited file (CSV) format, HTML format, Microsoft Excel (XLS) format, Adobe Portable Document Format (PDF), and Microsoft Word Rich Text Format (RTF).

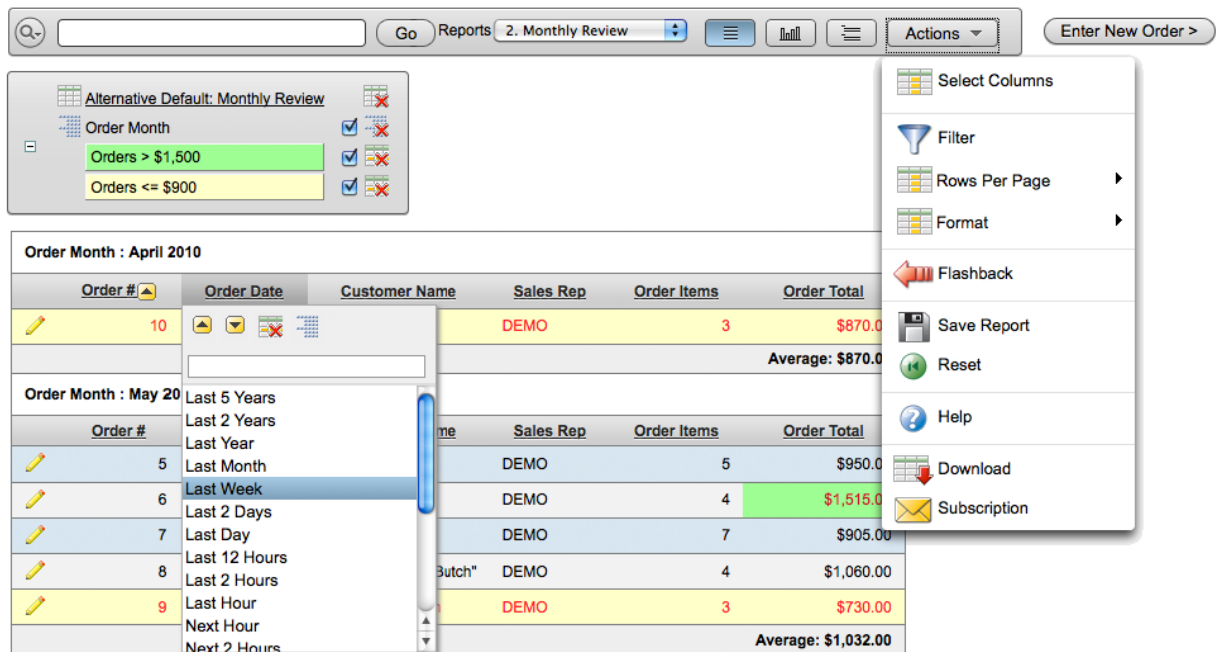


Figure 6. Oracle Application Express Interactive Reports

Flash Charts

Applications can be built to use Adobe's Flash Player browser plug-in to display charts. There are several types of Flash Charts supported including column, bar, pie, doughnut, scatter, line, candlestick, gauge, Gantt, and Flash Maps. During the creation process you can preview the selected type and set most attributes for your resulting chart. In addition to creating new Flash Charts, you can also convert your existing SVG Charts to Flash Charts. Flash Charts enable far greater controls on the user interface and also includes features such as asynchronous updates.

Dynamic Actions

Dynamic Actions allow developers to incorporate highly interactive Web 2.0 client-side processing without needing to code JavaScript and AJAX. Developers can declaratively define the client-side action and various true and false operations on various Application Express components. Simple actions like enable / disable or hide / show to complex actions such as an asynchronous call to the database to select values can be easily defined and maintained.

Plug-Ins

If you need to extend the functionality provided by Application Express, for example with a new GUI widget or specialized region, then you can utilize the Application Express Plug-Ins. Plug-Ins can be defined as item types, region types, dynamic actions, or processes and are built using PL/SQL and imported into your application. Once imported they are utilized in a very similar

manner to native components and can incorporate custom attributes. The Oracle Application Express development team and the APEX community have developed a number of plug-ins that can be downloaded and used to enhance your application.

PDF Printing

You can export a report region to PDF - essentially, printing a report. Integration with Oracle Business Intelligence (BI) Publisher enables “high fidelity” reports to be produced. Oracle BI Publisher provides a Microsoft Word plug-in to develop complex report templates that can incorporate multiple tables (e.g., master – detail), charts, and other parameters passed from Oracle Application Express.

Oracle Forms Conversion

Oracle Application Express enables you to load your Oracle Forms source into an Oracle Application Express project and generate an initial APEX application. The conversion automatically converts most user interface components but does not convert business logic implemented within triggers, program units, or PL/SQL libraries. Loading Oracle Forms definitions into the Oracle Application Express project provides analysis and tracking of the conversion project. Once the initial design has been generated, the rapid application development capabilities inherent within Oracle Application Express can be utilized to enhance and expand the generated application. Post-generation, the business logic within Oracle Forms can be re-implemented and new screens which take advantage of Web 2.0 capabilities such as interactive reports and flash charts can be added using the wizards built into the Oracle Application Express builder.

Microsoft Access Migration

Microsoft Access Migration enables you to generate an Oracle Application Express application based upon a Microsoft Access application. The migration process involves the following steps:

- Use the Exporter tool to export metadata from Microsoft Access
- Use Oracle SQL Developer Migrations (included in Oracle SQL Developer 1.2 and above) to migrate a Microsoft Access database to an Oracle database
- Create an Oracle Application Express workspace and then a migration project
- Validate and update the retrieved objects
- Generate an Oracle Application Express application

Team Development

Team development is designed to assist in managing the software development lifecycle. Features, milestones, to-dos and bugs can readily be cross linked to help manage releases and inter-dependencies. Features can be linked in a hierarchy with preferences to customize the

sections available (e.g. Documentation, Testing, Security, etc.). Features, to-dos and bugs can also be linked directly to an APEX Application and a page within that application.

You can incorporate feedback into your application to allow end-users to provide their comments. When a user enters feedback their current session state is also recorded for reference by the developer. The feedback is managed within Team Development where it can be easily transformed into a feature, to-do or bug.

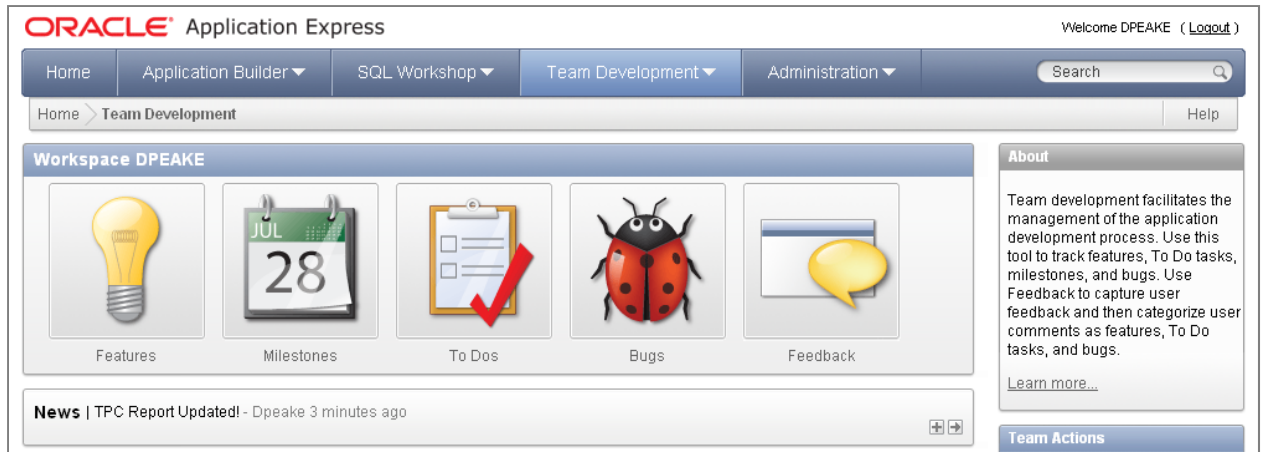


Figure 7. Oracle Application Express Team Development

Security Enhancements

Security best practices for developing Web applications specify minimizing the exposed footprint, protecting the integrity of the session state, and ensuring that data being transmitted between the browser and the server cannot be compromised. To align with these best practices, Oracle Application Express, release 3.2 has improved security measures making it more declarative and powerful.

Some key enhancements include the ability to:

- Declaratively encrypt session state
- Declaratively specify session timeouts for maximum idle time and maximum session duration
- Create new password item types that enable users to enter passwords without ever saving them to session state

Other features designed to minimize the exposed footprint include reducing the privileges required by the Oracle Application Express database account, disabling the Database Monitor feature by default, and providing administrators the ability to require HTTPS for the administration and development suite applications. In addition, administrators can now restrict new or updated account passwords to those that have not previously been used for a specified

duration and, for new installations, require that service administrator account passwords conform to a strong password policy.

Oracle Application Express documentation has also been improved to provide you with more resources to build secure Web applications. These new features complement existing Oracle Application Express security features including flexible authentication, authorization schemes, URL tampering protection, and so on.

Cross Language Features

The following application development features are available across multiple programming languages. Some of the features are implemented within the database server. Client-side features are provided by OCI, Oracle's native C programming interface. They are also available to other programming language interfaces such as .NET, PHP, Java (via JDBC-OCI), precompilers (Pro*C, Pro*Cobol, SQLJ) and OCCI all of which are built on top of OCI.

Client Query Result Cache

This feature enables caching of query results in client memory (in the OCI driver) and the result set data is kept consistent with database changes as if the query was being continuously executed. Because retrieving results locally from the client memory is much faster than making a database call, re-processing the query, and retrieving results from the disk, frequently executed queries experience a significant performance improvement when their results are cached. Client Query Result Cache (CQRC) also reduces the server CPU that would have otherwise been consumed for processing the query, thereby improving server scalability as well. Internal testing of a standard benchmark shows response time improvements of up to 22% and a 6 times reduction in server CPU consumption.

Table annotation in Oracle Database 11g Release 2, allows the caching of result sets to be completely transparent to the application. `ALTER TABLE sales RESULT_CACHE (MODE FORCE);`

When a table is annotated as cache worthy all queries run against the table become candidates for caching in the client/server cache. This eliminates the need for changing the application to add query level hints.

Client Query Result Cache can also be activated by annotating the query with the `/* + result_cache */` hint.

```
select /*+ result_cache */ * from employees
```


Server Query Result Cache

Results of queries can also be cached on the database side (within the shared pool). Users can annotate a query with the `/* + result_cache */` hint as in CQRC (see previous feature above) to indicate that result sets are to be cached.

Being on the database side, Server Query Result Cache is available to all database users irrespective of the language and API however, unlike CQRC, it requires a database roundtrip for data to be retrieved by the client.

The `RESULT_CACHE_MODE` initialization parameter controls whether the server query result cache is used for all queries (when possible), or only for annotated queries.

Database Resident Connection Pooling

Database Resident Connection Pooling (DRCP) addresses scalability requirements in environments requiring large numbers of connections with minimal database resource usage. As illustrated in Figure 8, DRCP provides a connection pool in the database server for typical usage scenarios where the application acquires a database connection, works on it for a relatively short duration, and then releases it. Multiple mid-tier machines can share the pool, allowing application servers to be added without requiring additional configuration.

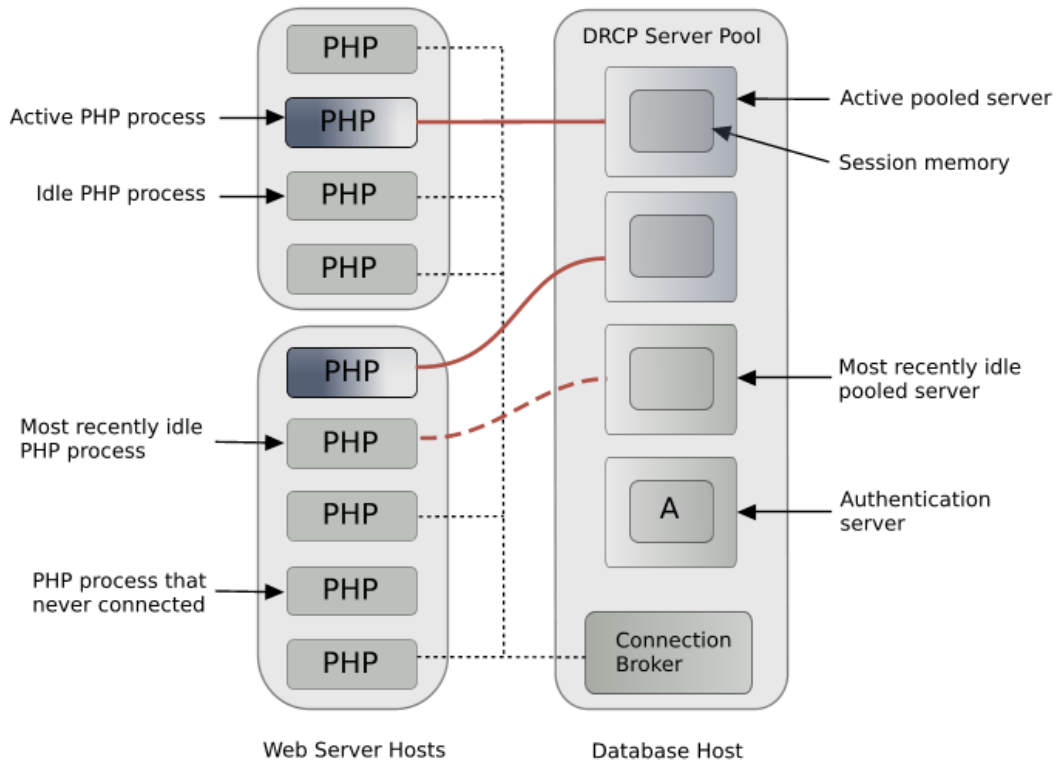


Figure 8. DRCP Architecture

DRCP is paramount for process-based systems such as PHP where there are no middle-tier connection pools to share connections among middle-tier processes. DRCP is also very useful for multiple middle-tier hosts to share a single pool of database server processes. This results in a significant reduction in key database resources needed to support a large number of client connections, thereby reducing the database tier memory footprint and boosting the scalability of both middle and database tiers. Having a pool of readily available servers also has the additional benefit of reducing the cost of creating and tearing down client connections.

With DRCP, a single database instance can scale up to tens of thousands of simultaneous connections. Internal DRCP testing shows 20K concurrent users against a single database instance on a commodity host server.

DRCP is currently exposed to C and C++ through the OCI driver; to PHP through the OCI8 adapter; to Python through the cx-oracle adapter and to Perl through DBD::Oracle.

For more details, see the PHP related white paper:

<http://www.oracle.com/technetwork/topics/php/whatsnew/php-scalability-ha-twp-128842.pdf>

IPv6 Support

Client or middle-tier applications can use IPv6 to connect to a single-instance database running on a dual-stack or IPv6-only host.

Fast Connect Time Failover for Applications

Faster failover can be achieved at connection establishment time by the use of new Net Services parameters. Rather than waiting for system level TCP timeouts, connect level timeouts can be set in seconds based on network latency. One can also set retry count for transparent retries at connection establishment time. Please see Net Services documentation for more details.

Secure LOBs and Prefetching of LOB Data

Several enhancements have been made to reduce the storage requirements for LOB data and increase performance in accessing it.

For Oracle SecureFiles (LOBs with STORE as SECUREFILE option), one can specify the SQL parameter DEDUPLICATE in CREATE TABLE and ALTER TABLE statements. This enables the application to specify that LOB data that are identical in two or more rows in LOB column will all share the same data block, thus saving disk space. Similarly, the parameter COMPRESS turns on LOB compression. The parameter ENCRYPT turns on LOB encryption and optionally selects an encryption algorithm. Each LOB column can have its own encryption specification, independent of the encryption of other LOB or non-LOB columns.

To improve query performance of smaller LOBs, LOB data and length can be prefetched and cached while also fetching the locator.

REF CURSOR Prefetching

In Oracle Database 11g Release 2 REF CURSOR prefetching is supported. It retrieves the REF CURSOR and the first rows in the same roundtrip, which improves performance for customers using stored procedures for business logic.

.NET

Oracle offers four components that simplify .NET development with Oracle Developer Tools for Visual Studio, Oracle Data Provider for .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 ADO.NET, and tools, such as Visual Studio, while also exposing powerful Oracle Database enterprise technologies, including Oracle Real Application Clusters and Oracle Advanced Security.

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 2010, Visual Studio 2008, and Visual Studio 2005.

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. This code generation makes developing ASP.NET web applications, client-server applications, or Microsoft Office applications easy with little coding required. Developers can easily modify table data, execute Oracle SQL statements, edit and debug PL/SQL code, and develop and deploy .NET stored procedures. 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>

Oracle Data Provider for .NET (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 Real Application Clusters, performance optimizations, XML DB, and advanced security features. ODP.NET gives programmers better performance, flexibility, and feature choice for their .NET applications. With it, developers can use .NET, but not sacrifice the powerful data management capabilities that Oracle provides. ODP.NET natively supports .NET Framework 2.0 to 4, including the .NET Framework for 32-bit Windows and Windows x64.

For more information about Oracle Data Provider for .NET, go to:
<http://www.oracle.com/technetwork/topics/dotnet/index-085163.html>

Oracle Database Extensions for .NET is a feature of Oracle Database 11g 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 ASP.NET providers that follow the ASP.NET provider model and use Oracle Database as the data source. These providers are a new Oracle Database 11g developer feature. Developers can store application state, such as shopping cart or user information, in a persistent Oracle data source. 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 Provider
- Role Provider
- Site Map Provider
- Session State Provider
- Profile Provider
- Web Event Provider
- Web Parts Personalization Provider
- Cache Dependency Provider

For more information about Oracle Providers for ASP.NET, go to:

<http://www.oracle.com/technetwork/topics/dotnet/index-087367.html>

Visual Studio 2010 and .NET Framework 4 Support

Visual Studio 2010 is fully supported by ODT 11.2.0.1.2 and higher, continuing Oracle's long history of supporting the latest new versions of Visual Studio soon after they become available. All of the productivity enhancing tools that Oracle developers have enjoyed in earlier releases are available from Visual Studio 2010.

ODP.NET, Oracle Providers for ASP.NET, and Oracle Database Extensions for .NET 11.2.0.1.2 and higher support .NET Framework 4, including support for the full framework and the new Client Profile.

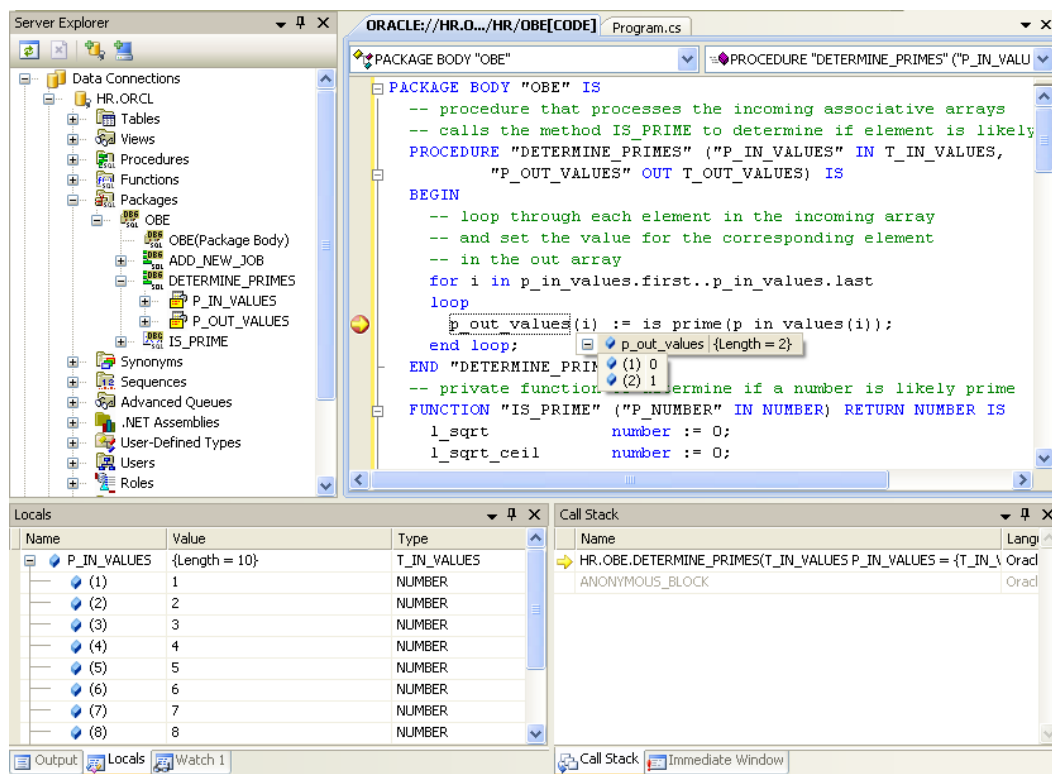


Figure 9. Oracle Developer Tools for Visual Studio

Visual Studio Application Performance Tuning Tools

One major new 11g feature focus area for ODT is application performance tuning. The new Oracle Performance Analyzer built into Visual Studio examines an application's use of an Oracle database over a specified period of time and provides recommendations to improve the performance of the application, such as modifications to SQL or adding an index on a table. In some cases, the recommendations can be automatically implemented by clicking a single button. Oracle Performance Analyzer is implemented using Automatic Workload Repository (AWR) snapshots and Automatic Database Diagnostic Monitor (ADDM) tasks.

Developers can also easily tune arbitrary SQL statements with SQL Tuning Advisor. Easily accessed via the "Tune SQL" menu item in the Oracle Query Window in Visual Studio, this tool provides instant recommendations explaining how to improve the performance of Oracle SQL statements.

Visual Studio Application Lifecycle Management and Source Control Integration

Also new in 11g, .NET developers can accomplish nearly all the tasks in a database application development lifecycle without having to leave Visual Studio to use another tool. For example, they can use graphical designers to create users and roles and grant privileges to them, import tables and table data from Oracle or other third party databases, graphically design SQL statements, generate SQL scripts for all of the Oracle schema objects that the application uses, edit and run the SQL scripts with a built in SQL*Plus engine, and check the application and SQL scripts into source control. Most source control systems are supported including Microsoft SourceSafe, Microsoft Team Foundation Server, and Subversion

Performance Enhancements

ODP.NET 11g introduces new performance enhancements that dramatically speed data access performance. Developers do not need to make .NET code changes to use these enhancements. The features include:

- A new ODP.NET self-tuning capability results in faster query execution. These improvements are a result of dynamically optimizing the statement cache size during runtime. ODP.NET monitors statement usage and load conditions to constantly tune the statement cache size. As a result, even if the load changes and query frequency changes over the course of a day or a month, ODP.NET will continue to tune the statement cache size.
- The Oracle TimesTen Data Provider for .NET, new with ODP.NET 11.2.0.2, enables fast data access for any .NET application to TimesTen in-memory databases. TimesTen Data Provider supports the classes, enumerations, interfaces, delegates and structures of the Oracle.DataAccess.Client and Oracle.DataAccess.Types namespaces.
- All ODP.NET data retrieval, not just repeatedly executed queries, will experience faster runtime performance when retrieving data with OracleDataReader or populating a DataSet. No ODP.NET code changes are required to use this feature. ODP.NET optimizations were added under the covers to speed data retrieval performance.
- ODP.NET now optimizes reuse of the fetch array buffer, which stores data for executed statements. This feature reduces client memory usage since ODP.NET requires fewer fetch array buffers for typical running applications.
- To help monitor application performance, ODP.NET 11g now publishes performance counters for connection pooling. These can be viewed using the Windows Performance Monitor.

- Using Oracle Database 11g, ODP.NET users will experience faster LOB retrieval performance (as described in the Cross Language Features section).
- Using Client Result Cache (as described in the Cross Language Features section), ODP.NET applications executing frequently run queries will experience a significant performance improvement.

Oracle Streams Advanced Queueing

The new Oracle Streams Advanced Queuing (AQ) administration tools in Visual Studio make it easy for developers who are developing applications that leverage Oracle's powerful database-integrated message queuing functionality to take care of administrative tasks. The easy to use interface greatly simplifies the task of creating and modifying queue and queue tables.

Besides administration tools, ODP.NET 11g introduces an AQ programmatic interface for messaging developers. ODP.NET can access all of AQ's operational features, such as enqueue messages, dequeue messages, listen for queue messages, and message notification.

The table below illustrates some of the key ODP.NET APIs for AQ.

TABLE 1. ODP.NET AQ FUNCTIONALITY AND IMPLEMENTATION SAMPLING

| AQ FUNCTIONALITY | ODP.NET IMPLEMENTATION |
|---------------------------------|--|
| Create Message | Create an <i>OracleAQMessage</i> object |
| Enqueue single message | Specify an <i>OracleAQMessage</i> message, an <i>OracleAQQueue</i> queue, and the <i>OracleAQQueue.enqueue</i> options. Finally, call <i>OracleAQQueue.Enqueue</i> . |
| Enqueue multiple messages | Specify an <i>OracleAQMessage</i> array of messages in <i>OracleAQQueue.EnqueueArray</i> . |
| Dequeue single message | Specify the <i>OracleAQQueue</i> dequeue options and then call <i>OracleAQQueue.Dequeue</i> . |
| Dequeue multiple messages | Call <i>OracleAQQueue.DequeueArray</i> . |
| Listen for messages on Queue(s) | Call <i>OracleAQQueue.Listen</i> . To listen on multiple queues, use static <i>OracleAQQueue.Listen</i> method. |
| Message Notification | Use <i>OracleAQQueue.MessageAvailable</i> event and the <i>OracleAQQueue.NotificationConsumers</i> property |

Promotable Transactions

Distributed transactions require orchestration among application, transaction coordinator, and multiple resource managers, which is a high overhead task. Local transactions only require an application and a single resource manager, or database. At design time, it may not be clear whether a transaction will be local or distributed during run time. Depending on run time

circumstances, the transaction may sometimes be local and sometimes require more than one database's involvement.

With ODP.NET 11g, promotable transactions allow all transactions to start local. When more than one resource manager joins, the transaction is promoted to a distributed transaction so that it can be managed by the transaction coordinator. Promotable transactions provide better run time optimization of system resources.

XCOPY Deployment

Oracle XCOPY, a new deployment feature, provides system administrators with an ODP.NET 11g client that is smaller in disk size than the standard ODP.NET client and is easily configurable. The install scripts consist of customizable text files so that administrators can easily add or remove software they need or do not need, respectively. They can embed the scripts into third-party deployment software or their own in-house installation scripts. Oracle's data access components, such as ODP.NET, Oracle Providers for ASP.NET, and Oracle Services for MTS, can be deployed via 32-bit or 64-bit xcopy.

Edition-Based Redefinition

Edition-based redefinition enables the upgrade of a database component of an application even while the application is being used. This minimizes or eliminates downtime for the application. ODP.NET 11g Release 2 supports specifying an Edition at deployment time when used with Oracle Database 11.2 or later.

User-Defined Types

Oracle user-defined types (UDTs), also known as objects and abstract data types, are used to represent complex data structures in the Oracle database. These data structures are representations of entities with multiple properties and methods in a single data structure.

Oracle includes a number of built-in object types: REFs and collections. A REF is a logical pointer to a row object, allowing easy navigation between objects. It can be moved around or change the object it points to without the overhead of moving the object being pointed to. Collections include VARRAYs, an ordered collection of elements, and nested tables, a collection that can have any arbitrary number of elements.

In ODP.NET 11g, Oracle supports both UDTs and built-in types by providing generic .NET provider type classes. The classes are generic in the sense that they can be used to represent any Oracle object type. They are relatively easy to set up and are not strongly typed.

ODP.NET provides three generic UDT classes that correspond to an object (OracleObject), a REF (OracleRef), and a collection (OracleArray). ODP.NET treats generic UDT classes as provider-specific types. Like all provider-specific types, they must be fetched from the OracleDataReader object using provider-specified type accessors, such as `GetOracleObject`.

Oracle Developer Tools for Visual Studio includes new tools to create and manage UDTs in the database. It also includes a custom class wizard, which takes an existing Oracle database UDT and auto-generates a .NET custom class in C#, Visual Basic, or C++.

ODT and ODP.NET UDT features increase developer productivity creating, managing, and developing with complex types for Oracle .NET applications.

Bulk Copy

ODP.NET provides a new bulk copy feature enabling applications to efficiently load large amounts of data from one data source into an Oracle database table.

HA Event Notification and Callback

ODP.NET can register for Oracle High Availability (HA) events. ODP.NET is then able to receive notifications on which database, service, host, or instance has gone down or come up. .NET developers can register an ODP.NET callback to notify the application when one of these events occurs and subsequently execute an event handler, which is a new feature in ODP.NET 11g. Developers can create their own application logic to be executed upon an Oracle server's up or down event. Prior to ODP.NET 11g, applications were able to request ODP.NET to register for Oracle High Availability (HA) with Real Application Clusters (RAC) so that whenever a database, service, service member, or host went down, ODP.NET processed these notifications internally to clean up connection pools. With ODP.NET 11g, these notifications are exposed to ODP.NET applications, enabling developers to implement custom HA event handling.

PHP

PHP is a popular scripting language commonly used for Web applications. PHP makes developing Web applications easy and rapid. It powers approximately 30% of Web sites on the internet [Source: <http://trends.builtwith.com/framework/PHP>]. PHP has matured and is making inroads in corporate systems and has a large user community. Oracle works with the open source community and maintains the OCI8 extension which is a stable, high-performance PHP database driver. Oracle Database 11g brings PHP developers important features and enhancements for mission-critical applications such as DRCP for connection scalability, Client Query Result Cache, REF CURSOR prefetching, RAC FAN events for fast connection failover, Edition Based Redefinition for online application upgrades and A/B testing, Client Identifiers for web auditing, authorization and monitoring, and much more.

For more information about Oracle and PHP, go to:

<http://www.oracle.com/technetwork/topics/php>

Database Resident Connection Pooling

Database Resident Connection Pooling (see the Cross Languages Features section) enables PHP applications to easily scale to tens of thousands of Oracle Database connections and can significantly reduce database server memory usage.

Real Application Clusters

PHP applications can greatly benefit from the scalability and high availability of Oracle's Real Application Clusters (RAC). RAC is a shared-disk implementation of Oracle Database that allows multiple instances to simultaneously access the same database. RAC's unique ability to run and scale real application workloads on a server pool is an essential foundation of Private Clouds.

High Availability with FAN and RAC

Clients that run in high availability configurations such as with Oracle RAC or Data Guard Physical Stand-By can take advantage of Fast Application Notification (FAN) events to allow applications to respond quickly to database node failures. FAN support in PHP may be used with or without DRCP – the two features are independent.

Without FAN, when a database instance or machine node fails unexpectedly, PHP applications may be blocked waiting for a database response until a TCP timeout expires. Errors are therefore delayed, sometimes up to several minutes, by which time the application may have exceeded PHP's maximum allowed execution time.

By leveraging FAN events, PHP applications are quickly notified of failures that affect their established database connections. Connections to a failed database instance are proactively terminated without waiting for a potentially lengthy TCP timeout. This allows PHP scripts to recover quickly from a node or network failure. The application can reconnect and continue processing without the user being aware of a problem.

Client Query Result Caching

Oracle Database 11g includes result caching. Result caches greatly speed the repeated execution of queries (see Cross Language Features section) and function calls (see PL/SQL section) that access read-only or read-mostly data.

Row and REF CURSOR Prefetching

Tuning the buffering of query rows can have a dramatic improvement on the performance of PHP applications by reducing network round-trips, and making available the appropriate number of results for an application. To complement the existing prefetching of query rows, Oracle Database 11g R2 introduces prefetching from PL/SQL REF CURSORS, improving performance of PHP applications that store business logic in the database.

Edition Based Redefinition

PHP OCI8 `oci_set_edition` allows setting editions; Oracle 11g Release 2 "editions" allow multiple versions of database objects to exist at one time in the same database schema. By setting different editions, two different versions of an application can run concurrently, making upgrading easier and faster. The use of editions makes it possible to quickly do "A/B" testing, allowing the evaluation of user responses to web site changes.

Client Identifier and Session Attributes

PHP OCI8 `oci_set_module_name`, `oci_set_action`, `oci_set_client_info`, and `oci_set_client_identifier` APIs allow setting of the module, action, client information and of client identifiers thereby enabling Oracle Database end-to-end tracing, authentication and end-user auditing. Client identifiers are particularly useful for PHP applications which generally connect to the one database schema and do their own end-user authentication. Client identifiers allow database operation to be monitored per end-user, and not solely amalgamated under the common database schema owner.

For more information about using client identifiers in PHP, go to:

<http://www.oracle.com/technetwork/articles/dsl/php-web-auditing-171451.html>

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.

For more information about PL/SQL, go to:

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

Real Native Compilation

Prior to Oracle Database 11g it was possible to speed up PL/SQL procedures by compiling them into native code residing in shared libraries. These procedures were translated into C code, then compiled with a C compiler and linked into the Oracle process. In Oracle Database 11g, the need for a C compiler vanishes and PL/SQL source is directly converted into native code. Moreover, Oracle does the linking and loading so that file system directories are no longer

needed. An individual developer can now compile program units for native execution without relying on any set-up on the part of the DBA. Compiling PL/SQL code natively is about twice as fast in Oracle Database 11g as it was in Oracle Database 10g. The run-time performance also improves noticeably.

Function Result Cache

A PL/SQL function is sometimes used to return the result of a computation whose inputs are one or several parameterized queries issued by the function. In some cases, these queries access data (for example, the catalog of wares in a shopping application) that changes very infrequently compared to the frequency of calling the function. You can include syntax in the source text of a PL/SQL function to request that its results be cached and, to ensure correctness, that the cache be purged when any of a list of tables experiences DML. The cache, like the SQL query result cache, is in the shared pool. The DBA manages and monitors both result caches using common APIs. The look-up key for the cache is the combination of actual arguments with which the function is invoked. When a particular invocation of the result-cached function is a cache hit, then the function body is not executed; instead, the cached value is returned immediately.

Sequences

The sequence generator provides a sequential series of numbers to applications. The sequence generator is especially useful in multi-user environments for generating unique sequential numbers such as an employee id without the overhead of disk I/O or transaction locking.

In the previous release of Oracle Database, when a PL/SQL program needed to get a value from a sequence, it used SQL. This is a usability irritation for PL/SQL programmers. In Oracle Database 11g, it is now possible to simply use the pseudocolumns `CURRVAL` and `NEXTVAL` in a PL/SQL expression.

PL/Scope

PL/Scope allows you to browse PL/SQL source code analogously to the way that Cscope (see <http://cscope.sourceforge.net>) allows you to browse C source code. You can search for and display all types of definitions, declarations, assignments and references in the PL/SQL source code. The PL/SQL compiler can optionally derive the metadata needed to support PL/Scope and store it in the database catalog. The metadata takes into account the nuances of the language, including scoping and overloading. You can generate reports (especially hyperlinked HTML reports) with supplied report generators. PL/Scope supports increased developer productivity, especially for those who need to maintain someone else's code.

Hierarchical Profiler

The PL/SQL hierarchical profiler reports the dynamic execution profile of your PL/SQL program, organized by subprogram calls. It accounts for SQL and PL/SQL execution times separately. Each subprogram-level summary in the dynamic execution profile includes information such as number of calls to the subprogram, time spent in the subprogram itself, time spent in the subprogram's subtree (that is, in its descendent subprograms), and detailed parent-children information. You can browse the generated HTML reports in any browser. The browser's navigational capabilities, combined with well chosen links, provide a powerful way to analyze performance of large applications, improve application performance, and lower development costs.

PL/SQL Comes to Oracle TimesTen

Oracle In-Memory Database Cache provides customers the ability to cache a subset of the performance critical Oracle database tables in the application tier to improve application response time. For PL/SQL code whose SQL processing needs improvement, customers can move the performance critical code to the application tier and run it within the application address space with the TimesTen in-memory database; resulting in improved response time and higher throughput.

For more information on TimesTen and In-Memory Database Cache, go to:

<http://www.oracle.com/technetwork/database/options/imdb-cache>

Java

Oracle Database includes an embedded Java VM, which lets you run Java directly in the database. This feature is a response to customer requirements for portability and reuse. Java can be migrated to/from middle-tier (J2EE, POJOS, JDBC) to the database (Java stored procedures) and vice versa. Other important benefits Java brings include the ability to reuse the large collection of Java class libraries that are available and to leverage the Java skills of application developers. Java in the database is being used to implement a broad range of applications; for more details, see the related white paper at:

<http://www.oracle.com/technetwork/database/enterprise-edition/ow-30820-java-stored-proc-paper.pdf>

JDBC is an industry-standard application programming interface (API) that lets you embed SQL statements in Java code. Each database vendor, such as Oracle, creates its JDBC implementation by implementing the interfaces of the standard `java.sql` package.

OracleJVM Performance, JIT Compiler

The OracleJVM furnishes a just-in-time compiler (JIT). The JIT dynamically and transparently produces native binaries from any Java classes existing in the database. The binary code are stored, avoiding recompilation. The JIT is enabled by default (out-of-the-box) and does not need a C compiler or further configuration. Internal testing using an industry standard benchmark shows an order of magnitude (10x) speed-up when compared to running the same test without the JIT. Field testing of Java in Database 11g at a customer site resulted in 15 times faster performance in a pure computational section of code and a 11 times improvement in a SQL section.

Universal Java Connection Pool

Universal Java Connection Pool (UCP) provides a single/universal connection pool supporting any database (Oracle, non-Oracle) with any type of connection (JDBC, JCA, LDAP) and any application server (Oracle, non-Oracle). It also supports stand-alone deployment (BPEL, Toplink, Tomcat) and seamless integration with both RAC and non-RAC Oracle Databases.

UCP for JDBC ships as an independent jar in Oracle Database 11g release 11.1.0.7 and up. For more details on UCP, see the UCP Developers Guide at:

http://www.oracle.com/pls/db112/to_pdf?pathname=java.112/e12265.pdf

Productivity and Portability through Java Standards

Productivity and application portability across platforms and vendors are the primary motivations for using Java. In this release, JDBC brings support for Java SE 6, JDBC 4.0 and JMX. The Java runtime embedded in the RDBMS brings support for Java SE 5, RowSet (JSR-114), and JMX.

JDBC 4.0

The Oracle Database 11g JDBC comes in two flavors: ojdbc5.jar for Java 5 (i.e., JDK 1.5) and ojdbc6.jar for Java 6 (i.e., JDK 1.6). The ojdbc6.jar supports the JDBC 4.0 specification including: Connection and Statement Enhancements, Wrapper Interface, New Standard Datatypes, SQL 2003 National Character Set types, LOB Enhancements, Exception Hierarchy, and RowSet (JSR-114) Enhancements.

JDBC Enhancements

The following enhancements were made to Oracle Database JDBC driver:

- Ease of Upgrade for Timezone and TimesTamp file versions
- JDBC LOB Pre-Fetch and Zero-Copy SecureFiles LOB
- Universal Connection Pool

- Java FAN API
- Support for Database Edition (JDBC-OCI) via property `oracle.jdbc.editionName`

OracleJVM Compliance with IP V6

IP V6 support via system properties: `java.net.preferIPv4Stack` & `java.net.preferIPv6Addresses`

RowSet (JSR-114) Comes to Server-side JDBC

The server-side JDBC driver now supports the RowSet specification. Java/JDBC code that make use of `JDBCRowSet`, `CachedRowSet`, `WebRowSet`, `JoinRowSet` and `FilteredRowSet` can now run directly in the database.

Exposing Oracle to Java

In this release the following features are exposed: `SYS.ANYDATA`, `SYS.ANYTYPE`, Secure Files, and Query change notification.

Prefetch in 1st Roundtrip

For ad-hoc SQL queries, the new pre-fetch mechanism combines parse, execute, and fetch to retrieve the first batch of the results set in the first interaction with the database. Reliable internal tests show a 50% reduction in network roundtrip; 50% reduction in response time; 30% reduction of server and client CPU consumption. To put this reduction in perspective, a typical Web retail application issuing 1 million ad-hoc queries per day (or a determined period of time) will see a reduction of 1 million roundtrips during the same period.

Native AQ Protocol

For JMS/AQ calls via JDBC, a new native AQ protocol (as opposed to a PL/SQL based interface) garners up to 300% performance improvement. In addition, server CPU consumption is also reduced as a result of shifting from PL/SQL based interface to the native AQ protocol.

Database/Query Change Notification

JDBC-Thin and JDBC-OCI support for Database and Query Notification allows a JDBC thread to subscribe to notification of changes in a query result set

Middle-tier may use this feature to invalidate and refresh data caches. See more details on the APIs in the Oracle Database 11g JDBC documentation.

Server and Client Query Result Cache

SQL result sets can be cached in the SGA on the server-side and automatically invalidated whenever related objects are changed. The JDBC-OCI driver furnishes a client-side counterpart

of server result cache. It is synchronized with changes to the server cache. See the Cross Language Features section for more details.

Security

JDBC-Thin now supports the Oracle Advanced Security option – on par with JDBC-OCI, which already supports it -- including strong authentication (i.e., Kerberos, Radius, SSL) and support for new encryption and data integrity algorithms.

Oracle JVM Ease of Use

Limited ease of use is one of the obstacles to mass adoption of Java in the database. This release brings many features to simplify and improve the user experience including: like-JDK interface, output redirect, property interface, database resident JAR, two-tier duration of Java session state.

Manageability

Manageability is a key requirement in production environments for Java applications running against (i.e., JDBC, Java EE components) or within the Oracle Database (i.e., Java in the database). This release brings the following manageability features: MBean for JDBC Logging, programmatic startup and shutdown, JMX in OracleJVM, and OracleJVM Utilities enhancements.

For more information about Oracle Java, JDBC, UCP, & Database Web Services, go to:

<http://www.oracle.com/technetwork/database/enterprise-edition/index-097123.html>

Oracle Call Interface (OCI)

Oracle Call Interface provides high performance, powerful access and fine-grained control for application design. OCI is also the foundation on which .NET, PHP, and the JDBC type 2 driver is built upon (see the Cross Language Features section for OCI features exposed through multiple languages). The following additional features are introduced in OCI with Oracle Database 11g.

Oracle TimesTen In-Memory Database Cache 11g

Oracle In-Memory Database Cache (IMDB Cache) 11g supports OCI. This release is built using the Oracle Client 11.1.0.7 and supports the contemporary OCI 8 style APIs. TimesTen OCI syntax and usage is the same as that in Oracle Database. Applications may use OCI to connect to IMDB Cache databases via direct-link (in-process) connection mode and client/server connection mode; direct-linked connections provides the most optimal response time. IMDB Cache also supports Oracle Pro*C/C++ precompilers.

For more information, refer to the TimesTen C Developer Guide:

<http://www.oracle.com/technetwork/database/timesten/documentation>

Database Resident Connection Pool

OCI session pool is integrated with the Database Resident Connection Pool (DRCP) functionality in Oracle Database 11g. DRCP allows sharing of database connections across middle tier processes and hosts, yielding extreme scalability and optimal memory usage on the database. In Oracle Database 11g Release 2, OCI also provides programmatic interfaces to get connections from DRCP for custom connection pools that might not be built using OCI session pool.

Client Query Result Cache

OCI provides the basic support for the Client Query Result Cache described in the Cross Language Features section earlier.

Secure File LOBs

To improve performance of smaller LOBs, OCI supports pre-fetching LOB data and LOB length when fetching the locator. This improves performance by reducing roundtrips to the database.

REF CURSOR Prefetching

In Oracle Database 11g Release 2, OCI can pre-fetch rows for REF CURSORS also. OCI pre-fetches a specified number of additional rows on the first and subsequent fetches on the REF CURSOR. This improves performance for customers using stored procedures for business logic.

Edition Based Redefinition

In Oracle Database 11g Release 2, OCI exposes the Edition Based Redefinition functionality programmatically by allowing users to specify the EDITION name as a connection attribute as well as an OCI session pool level attribute.

64-bit Integer Host Datatype

Starting with release 11.2, OCI supports the ability to bind and define integer values to a **NUMBER** column using a 64-bit native host variable and **SQLT_INT** or **SQLT_UIN** as the external datatype in an OCI application.

Implicit Fetching of ROWIDs

A ROWID is a globally unique identifier for a row in a database. A ROWID is the fastest way to access a single row. Implicit fetching of ROWIDs in **SELECT ... FOR UPDATE** statements means that the ROWID is retrieved at the client side even if it is not one of the columns named

in the SELECT statement. The position parameter of `OCIDefineByPos()` is set to zero (0) to access the ROWID.

Binding and Defining Multiple Buffers

You can specify non-contiguous buffers for use with a single bind or define call. Performance is improved because data stored/fetched at non-contiguous addresses does not need to be copied from/to a contiguous buffer. Also, this feature is useful for applications that cannot allocate large contiguous buffers for array bindings due to memory management concerns. With this feature, the application can split the array across a set of smaller non-contiguous buffers.

Allocating an Array of Descriptors

Array binds and defines allow an application to insert or fetch multiple rows in one round trip to the database. To facilitate binding/defining array of descriptors (such as those for LOBs, timezone etc.), OCI has added the `OCIArrayDescriptorAlloc()` call where you can allocate an array of descriptors in one OCI call.

Diagnosability

Diagnostic data of OCI and Net services is captured without user intervention in the event of fatal errors. This helps in first failure diagnosis.

Runtime Connection Load Balancing

OCI provides intelligent allocations of sessions in the session pool based on the current service level provided by the database instances when applications request a session to complete some work. The allocation is done based on RAC Load Balancing Advisories, in other words, the best instance to process the given work.

Client and Server with Different Versions of Time Zone Files

Starting with Oracle Database 11g Release 2, under certain scenarios, you can use different versions of time zone file on the client and server with OCI, PRO*C, JDBC and SQL*Plus clients; this mode of operation was not supported prior to 11.2. Both client and server must be 11.2 or higher to operate in such a mixed mode.

For more on OCI, go to <http://www.oracle.com/technetwork/database/features/oci/index-090945.html>

Conclusion

Application development organizations have a wide variety of programming languages and technologies to choose from. Oracle Database 11g delivers a highly productive and powerful set of application development tools supporting the most popular development technologies

including Java, PHP and .NET. Compelling Oracle Database features (e.g., Real Application Clusters, XML DB, etc.) are available to developers through Oracle's programming language interfaces.

Edition-based redefinition reduces the time, cost and risk of upgrading applications. Oracle Application Express is a unique Web application development tool that is ideal for quickly building departmental applications. 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. 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. PL/SQL and Java in the database provide architectural options to create database server-side code that can increase the performance, security, maintainability and scalability of your applications.

Collectively these tools simplify your development tasks, reduce costs and enable your organization to shorten application development time to market.



Application Development with
Oracle Database 11g Release 2
October 2010

Author: David Gambino
Contributing Authors: Sue Harper, Christopher
Jones, Alex Keh, Bryn Llewellyn, Kuassi
Mensah, David Peake, Christian Shay

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2010, 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 is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.