

Delivering Oracle Compatibility In Postgres Plus[®] Advanced Server

An EnterpriseDB White Paper

for DBAs and Application Developers

June, 2009

Executive Summary

Enterprises running Oracle® are generally interested in alternative databases for at least three reasons:

- Lower Total Cost of Ownership (TCO) in the face of rising prices and tough economic environments,
- Greater licensing flexibility to become more agile within the company and in the larger market, and
- Vendors who will provide superior technical support and a richer customer experience

However, fear of the costs of changing databases, including costs related to performance degradation, application re-coding and personnel re-training have outweighed the expected savings, contributing to vendor lock-in.

EnterpriseDB's open source based Postgres Plus Advanced Server, is an enhanced PostgreSQL that matches Oracle performance while also being compatible with critical Oracle features like: PL/SQL, Oracle syntax, Oracle Catalog Views, OCI support, packages, Oracle-like tools, Replication, data types and much more.

An evaluation of Postgres Plus Advanced Server as a compliment or replacement to Oracle would be time well spent by Oracle shops who want to address lower, TCO, licensing flexibility, and true Oracle compatibility in an alternative database.

An in depth discussion targeted specifically to your organization's Oracle requirements can be scheduled with an EnterpriseDB domain expert by sending an email to sales@enterprisedb.com.

The Need for Oracle Compatibility

Enterprises running Oracle® are generally interested in alternative databases for at least three reasons. First, these enterprises are experiencing budget constraints and need to lower their database Total Cost of Ownership (TCO). Second, they are trying to gain greater licensing flexibility to become more agile within the company and in the larger market. Finally, they are actively pursuing vendors who will provide superior technical support and a richer customer experience. And, subsequently, enterprises are looking for a solution that will complement their existing infrastructure and skills.

The traditional database vendors have been unable to provide the combination of all three benefits. While Microsoft® SQL Server™ and IBM® DB2™ may provide the flexibility and rich customer experience, they cannot significantly reduce TCO. Open source databases, on the other hand, can provide the TCO benefits and the flexibility.

However, these open source databases either lack the enterprise-class features that today's mission-critical applications require, or they are not equipped to provide the enterprise-class support required by these organizations.

Finally, none of the databases mentioned above provide the database compatibility and interoperability that complements their existing applications and staff. The fear of the costs associated with changing databases, including costs related to application re-coding and personnel re-training, outweigh the expected savings. Therefore, these enterprises remain paralyzed and locked into Oracle.

To meet the needs of these enterprises, EnterpriseDB has substantially enhanced PostgreSQL, the world's most advanced open source database, to create Postgres Plus™ Advanced Server. Postgres Plus Advanced Server is an enterprise-class relational database management system (RDBMS) that is suitable for high-volume, mission-critical applications.

EnterpriseDB's enhancements to PostgreSQL fall primarily in two categories: performance and Oracle compatibility.

The most distinguishing feature of Postgres Plus Advanced Server is its ability to run applications written for Oracle databases without changes to the applications' code. EnterpriseDB's compatibility with Oracle makes it easy for enterprises with existing Oracle investments to enjoy the benefits of open source databases.

The ability to run existing Oracle applications on Postgres Plus Advanced Server eliminates the costly, time-consuming, and risky re-coding typically required to migrate applications from Oracle to any other database. In addition, an enterprise's Oracle DBA and database developer skill sets are completely re-usable in a Postgres Plus Advanced Server environment, eliminating the need for costly re-training or re-hiring.

At the Enterprise Performance Center near Oxford, England, EnterpriseDB database performance specialists have tuned the performance of Postgres Plus Advanced Server to enable it to run up to 50% faster than native PostgreSQL. The result is a highly scalable database solution with unmatched cost-effectiveness. In fact, after moving their existing Oracle applications to Postgres Plus Advanced Server, many customers cite performance equal to or faster than the same applications running against Oracle.

EnterpriseDB has augmented PostgreSQL in nine critical areas to create an enterprise-class database that can *replace or supplement* the Oracle databases in an enterprise's IT infrastructure:

- **Oracle SQL Compatibility.** Postgres Plus Advanced Server executes Oracle-specific SQL syntax.
- **PL/SQL Compatibility.** Postgres Plus Advanced Server executes PL/SQL, Oracle's unique language for triggers, stored procedures, packages and functions.
- **Data Dictionary Views.** Postgres Plus Advanced Server contains the most common Oracle catalog views.
- **Programming Flexibility and Drivers.** Postgres Plus Advanced Server supports the most common programming languages used to create database applications for Oracle including compatibility and interoperability with the Oracle Call Interface (OCI) TM.
- **Migration Tools.** Postgres Plus Advanced Server provides a suite of automated tools to move Oracle schema, data, packages, triggers, stored procedures, and functions to a Postgres Plus Advanced Server database in one simple step.
- **Replication.** Postgres Plus Replication Server can replicate Oracle databases in near real-time to improve database performance, offload reporting from OLTP systems, and as the foundation for other applications at a small fraction of Oracle's cost.
- **Enterprise-Class Reliability and Scalability.** Postgres Plus Advanced Server is a suitable replacement for Oracle in many high-volume, mission-critical applications.
- **Security.** Postgres Plus Advanced Server provides source code obfuscation to hide sensitive or confidential information in addition to traditional role-based authentication and authorization.

- **Oracle-Like Tools.** Postgres Plus Advanced Server includes a robust set of integrated tools that will be familiar to professional Oracle DBAs and developers.

Each of these nine areas is discussed in greater detail below. This white paper is not a technical reference, but it is intended to provide database administrators (DBAs) and developers with an understanding of the depth of Postgres Plus Advanced Server's compatibility with Oracle. Complete EnterpriseDB documentation and free downloads of Postgres Plus Advanced Server are available at www.enterprisedb.com.

Introducing Postgres Plus Advanced Server

Postgres Plus Advanced Server is a stable, secure and scalable enterprise-class relational database management system (RDBMS) that is built on the open source PostgreSQL database and compatible with many Oracle applications. While priced for open source environments and deployments, Postgres Plus Advanced Server is a proprietary, reliable RDBMS suitable for high-volume, mission-critical enterprise use.

The EnterpriseDB product family consists of a comprehensive relational database management system suite that includes all the elements of an enterprise-class software solution. The EnterpriseDB product family consists of:

- **Postgres Plus Advanced Server.** Based on the well-tested and industry-accepted PostgreSQL database, Postgres Plus Advanced Server delivers solid Oracle compatibility, stability, and scalability.
- **DBA Management Server.** A single web-based interface for monitoring and administering Postgres Plus Advanced Server databases.
- **DBA Monitoring Console.** A web-based interface for real-time resource usage monitoring.
- **Postgres Studio.** A utility that enables developers and DBAs to manage Postgres Plus Advanced Server databases.
- **Migration Studio.** An automated tool that moves Oracle schema, data, packages, functions, and procedures to Postgres Plus Advanced Server.
- **Postgres Plus Replication Server.** Replicates Oracle and Postgres Plus Advanced Server databases across the enterprise in near real-time to meet a wide array of business challenges.
- **EnterpriseDB Connectors.** Enables Postgres Plus Advanced Server to be used with all the most popular programming languages and platforms.

SQL Compatibility

A fundamental building block of Postgres Plus Advanced Server's Oracle database compatibility is its ability to recognize and appropriately execute database queries expressed in Oracle's SQL language. In addition, Postgres Plus Advanced Server supports the same data types, functions, and variables as Oracle and fixes incompatibilities between Oracle's and PostgreSQL's treatment of column aliasing, public synonyms, sequences, and the DUAL table.

Oracle-Compatible Data Types

While the International Standards Organization defines a set of standard SQL data types, Oracle has extended the set to provide additional flexibility and functionality for developers. All Oracle data types have been mapped to those native in PostgreSQL to provide Oracle compatibility.

Because Oracle has had to maintain backward binary data type compatibility for so long, most data type mappings in Postgres Plus Advanced Server provide additional options and flexibility.

Data Type	Oracle Compatible	Additional Functionality
CHAR	✓	See Note 1
VARCHAR	✓	See Note 2
VARCHAR2	✓	See Note 2
NUMBER	✓	See Note 3
BLOB	✓	
CLOB	✓	
DATE	✓	

Note 1: In Oracle, the CHAR data type has an upper limit of 2000 bytes or characters, while the maximum size of CHAR in Postgres Plus Advanced Server is 1GB (the actual number of characters which can be stored is based solely on database character set encoding).

Note 2: In Oracle, the VARCHAR and VARCHAR2 data types have an upper limit of 4000 bytes or characters, while the maximum size of these data types in Postgres Plus Advanced Server is 1GB (the actual number of characters which can be stored is based solely on database character set encoding).

Note 3: In Oracle, the NUMBER data type is limited to 38 digits of precision, while the Postgres Plus Advanced Server NUMBER implementation provides up to 1000 digits of precision.

Oracle-Compatible Functions and Variables

Postgres Plus Advanced Server supports the following SQL-callable functions and variables which are popular with Oracle developers and DBAs:

Oracle-Compatible Functions/Variable Names		
ADD_MONTHS	CEIL	CHR
CONCAT	DECODE	FLOOR
GREATEST	INITCAP	INSTR
LAST_DAY	LEAST	LENGTH
LTRIM	LOWER	LPAD
MONTHS_BETWEEN	NEW_TIME	NEXT_DAY
NVL	NVL2	REPLACE
ROUND	RPAD	RTRIM
SUBSTR	SYSDATE	SYSTIMESTAMP
TO_CHAR	TO_DATE	TO_NUMBER
TRUNC	UPPER	USER

Oracle-Compatible Column Aliasing

Column aliasing is a SQL feature commonly used by both Oracle developers and report writers. Because PostgreSQL has long required the keyword "AS" when assigning column aliases in a select-list and Oracle does not, PostgreSQL has had incompatibilities with many SQL statements written for Oracle. Postgres Plus Advanced Server has removed the requirement of the "AS" keyword, allowing Oracle SQL queries to run without error.

Oracle-Compatible Public Synonyms

Postgres Plus Advanced Server supports PUBLIC SYNONYMS; alternate database-wide names for objects such as tables, views, and sequences.

Oracle-Compatible Sequences

While PostgreSQL implements sequences, it uses the functions CURRVAL(seq_name) and NEXTVAL(seq_name) to perform manipulations. This is incompatible with Oracle's seq_name.CURRVAL and seq_name.NEXTVAL syntax. To remedy this incompatibility, EnterpriseDB has extended PostgreSQL to support Oracle-compatible sequence manipulation syntax while keeping the current syntactical benefits of PostgreSQL.

With the exception of the ORDER clause, the CREATE SEQUENCE syntax used for defining sequences in Oracle will also work in Postgres Plus Advanced Server. For example:

```
CREATE SEQUENCE empno_seq START WITH 1 INCREMENT by 1;
```

An example of Postgres Plus Advanced Server's added benefits is the use of Oracle-compatible sequence manipulation syntax within the DEFAULT clause of a table definition for auto-numbering. By taking advantage of this feature, the need to manually create a BEFORE UPDATE trigger for auto-numbering (as is required by Oracle) has been eliminated. For example:

```
CREATE TABLE emp (
  empno          NUMBER (10) NOT NULL
                DEFAULT empno_seq.NEXTVAL,
  ename         VARCHAR2(32) NOT NULL,
  ...
  PRIMARY KEY (empno));
```

Oracle-Compatible Hierarchical Queries

Postgres Plus Advanced Server supports hierarchical queries as in Oracle. Hierarchical queries allow for querying parent/child relationships within a single table. Postgres Plus Advanced Server supports the most commonly used operators for hierarchical queries. (what are some example of the operators?)

Oracle-Compatible Optimizer Hints

Optimizer hints are directives that can be embedded in certain SQL statements. These directives force the query optimizer to use (or not to use) a given type of query plan to retrieve data. Postgres Plus Advanced Server supports over 15 of the most commonly used directives.

Oracle-Compatible Database Links

Database links are named database objects containing connection information to a remote database. This allows a SQL statement to access a table in the remote database by simply adding the "@dblink" syntax after the table name. Postgres Plus Advanced Server supports database links to both Oracle databases and to other remote Postgres Plus Advanced Server databases.

The DUAL Table

As Oracle has always required a FROM clause to be present in SELECT statements, they created a table named DUAL which contains a single column named DUMMY and has been populated with a single row of value x. Postgres Plus Advanced Server includes this table for SQL statement compatibility with Oracle.

ROWNUM

ROWNUM is a pseudo column and it is called in many applications. ROWNUM numbers the records in a result set. The first record that meets the where criteria in a select statement is given ROWNUM=1, and every subsequent record meeting that same criteria increases ROWNUM. Postgres Plus Advanced Server supports ROWNUM.

PL/SQL Compatibility

Postgres Plus Advanced Server includes a procedural language called EnterpriseDB SPL that closely matches Oracle's PL/SQL procedural language. Like PL/SQL, SPL is a highly productive, block-structured procedural programming language for writing custom procedures, functions, and triggers. The close similarity between EnterpriseDB's SPL and Oracle's PL/SQL also enables Postgres Plus Advanced Server to support Oracle-style packages of procedures, functions and variables.

Block Structure

As in PL/SQL, SPL procedures, functions, and triggers have the same block structure. A block consists of up to three sections – an optional declaration section, a mandatory executable section, and an optional exception section. Minimally, a block has an executable section that consists of one or more procedural statements within the keywords BEGIN and END.

Oracle Built-In Packages

Postgres Plus Advanced Server supports the most used packages that are built into the Oracle database. A package is a construct for building reusable code and employing object-oriented design techniques. A package is a collection of a related database objects identified by a

common package name in the database. These database objects include procedures, functions, and variables. Oracle has built-in packages which are often called by existing applications and used by developers to simplify programming. The following list shows how SPL program functionality is enhanced by the built-in packages supported by Postgres Plus Advanced Server.

- **DBMS_ALERT.** Programs can register and receive alerts from a signaling program in which some predefined event has occurred.
- **DBMS_JOB.** Programs can be scheduled to run regularly at certain time intervals.
- **DBMS_LOB.** Large objects (BLOBs and CLOBs) can be manipulated within programs.
- **DBMS_OUTPUT.** Messages can be written to, or received from a session's local message buffer.
- **DBMS_PIPE.** Messages can be exchanged between programs running in different sessions.
- **DBMS_SQL.** Dynamic SQL can be executed by programs.
- **DBMS_UTILITY.** Programs can run any of various, useful utilities.
- **UTL_FILE.** Programs can manipulate operating system files.
- **UTL_MAIL.** Programs can send emails.
- **UTL_SMTP.** Programs can send emails by directly issuing Simple Mail Transfer Protocol (SMTP) commands.

User Defined Packages

In addition to the built-in packages, Postgres Plus Advanced Server also supports Oracle-style custom packages. As in Oracle, a Postgres Plus Advanced Server package consists of two main components:

- **Package Specification.** This is the public interface containing the public procedures, functions, and variables that can be referenced outside the package by other programs and applications.
- **Package Body.** This contains the implementation logic of procedures and functions declared in the package specification as well as the declaration and logic of private variables, procedures, and functions that cannot be accessed by other programs and applications external to the package.

Procedures

Procedures in EnterpriseDB SPL work as they do in Oracle PL/SQL. Procedures are programs that are invoked or called as an individual program statement. When called, procedures may optionally receive values from the caller in the form of input parameters and optionally return

values to the caller in the form of output parameters. A procedure is stored in the database by executing a script containing the procedure definition. Then, the procedure may be invoked from another program.

Functions

Functions in EnterpriseDB SPL work as they do in Oracle PL/SQL. Functions are programs that are invoked as expressions. When evaluated, a function returns a value that is substituted in the expression in which the function is embedded. Functions may optionally take values from the calling program in the form of input parameters. In addition to a function returning a value, it may optionally return additional values to the caller in the form of output parameters. The use of output parameters in functions, however, is not an encouraged programming practice. A function can be used anywhere that an expression can appear within a statement.

Triggers

Triggers in EnterpriseDB SPL work as they do in Oracle PL/SQL. A trigger is a block of code that is given a name, associated with a table, and stored in the database. When certain events occur on the table, the code block is executed. The trigger is said to be “fired” when the code block is executed.

Like Oracle, Postgres Plus Advanced Server supports both row-level and statement-level triggers. A row-level trigger fires once for each row that is affected by a triggering event. In contrast, a statement-level trigger fires once per triggering statement, regardless of the number of rows affected by the triggering event.

As in Oracle, trigger code blocks may be executed by the Postgres Plus Advanced Server before or after the triggering statement in the case of statement-level triggers and before or after each row is affected in the case of row-level triggers.

Anonymous Blocks

As in Oracle, a block of procedural code can simply be executed in Postgres Plus Advanced Server. A code block of this type is called an anonymous block. An anonymous block is unnamed and is not stored in the database. Once the block has been executed and erased from the application buffer, it cannot be re-executed unless the block code is re-entered into the application. Anonymous blocks are useful for quick, one-time programs, such as testing programs.

Collections

An Oracle collection is a set of ordered data of the same data type. An array is the most commonly thought of example of a collection. Postgres Plus Advanced Server supports the following collection types:

- **Nested Table.** List of data elements ordered by positive integers.
- **Associative Array.** Collection of data elements that can be ordered by negative and non-negative integers, or by character keys as well.

Both the nested table and associative array can be defined and used locally within a program. In addition, the CREATE TYPE statement can be used to create a globally available nested table definition that can be used by multiple programs without each program having to recreate the definition.

Manipulation of collections is facilitated by special functions called collection methods. The following are the supported collection methods:

- **COUNT.** Returns the number of non-empty elements in the collection.
- **FIRST.** Returns the index of the first element in the collection.
- **LAST.** Returns the index of the last element in the collection.

Bulk Binding and Bulk Collecting

Bulk binding and bulk collecting are techniques commonly used in Oracle where collections are used to improve the efficiency of SQL statements. These same techniques can be applied in Postgres Plus Advanced Server.

Bulk binding is used when a SQL statement is required to change a set of rows with a different set of values for each row. This technique is applied by the FORALL statement. The FORALL statement passes sets of values to the database server in collections. This eliminates the need to re-iteratively invoke the same SQL statement with the different values.

The following is an example of using the FORALL statement with the UPDATE SQL statement to change the salaries of a list of employees:

```

DECLARE
    TYPE empno_tbl IS TABLE OF emp.empno%TYPE;
    TYPE sal_tbl IS TABLE OF emp.ename%TYPE;
    t_empno EMPNO_TBL;
    t_sal SAL_TBL;
BEGIN
    ...
    FORALL i IN t_empno.FIRST..t_empno.LAST
        UPDATE emp SET sal = t_sal(i)
        WHERE empno = t_empno(i);

```

Bulk collecting is used when a SQL statement returns multiple rows such as in a SELECT statement, FETCH statement, or when the RETURNING clause is used with insertions, updates, or deletions. The BULK COLLECT clause employs bulk collecting to return the result set in collections as shown in the following example:

```

DECLARE
    TYPE empno_tbl IS TABLE OF emp.empno%TYPE;
    TYPE ename_tbl IS TABLE OF emp.ename%TYPE;
    TYPE sal_tbl IS TABLE OF emp.sal%TYPE;
    t_empno EMPNO_TBL;
    t_ename ENAME_TBL;
    t_sal SAL_TBL;
BEGIN
    ...
    UPDATE clerkemp SET sal = sal * 1.5
        RETURNING empno, ename, sal
        BULK COLLECT INTO t_empno, t_ename, t_sal;

```

Object Types

As in Oracle, an object type specification and its attributes can be created using the CREATE TYPE statement. The object type attributes can be built-in data types or user-defined types including nested tables. Object types can be used as parameters of procedures and functions as well as the return type of functions. The introduction of object types gives application developers more power and flexibility in their application designs and implementations.

Transaction Control

In an EnterpriseDB SPL program, the default transactional behavior has been automatic commits of database updates made during the transaction if the program terminates without error (no exceptions have been thrown

in the program). On the other hand, if any error occurred in the program, then all prior database updates that occurred within the transaction were unconditionally rolled back. The programmer had no control over this behavior.

Two new features are introduced in Postgres Plus Advanced Server that gives the programmer more fine-grained transactional control as well as compatibility with Oracle's PL/SQL transactional model:

- **Transaction Error Recovery Support Extensions (TERSE).** Control over whether or not an error causes an immediate abort of database updates.
- **COMMIT WORK and ROLLBACK WORK.** SPL program statements that can explicitly force a commit or rollback of database updates from anywhere within the procedural logic of an SPL procedure or function.

TERSE is controlled by a single configuration parameter. When set to "off", all database changes made during the course of an aborted transaction are automatically rolled back.

When set to "on", only the database changes made by the offending SQL statement are rolled back (this is called statement-level rollback). All other database updates made during the transaction are in a pending state, which can now be either explicitly committed or rolled back in the SPL program code. (Note that if an exception is thrown by a non-SQL statement such as division by zero, database updates within the transaction are still kept in a pending state.)

The COMMIT WORK and ROLLBACK WORK statements now give the programmer full control within the SPL program's procedural logic as to whether pending database changes within the transaction should be committed or rolled back.

Data Dictionary Views

Postgres Plus Advanced Server provides views into its data dictionary that resemble the most commonly used views into Oracle's data dictionary. These views allow database administrators and developers familiar with Oracle to look up dictionary information quickly without having to re-learn a new data dictionary format.

By providing Oracle-compatible DBA, ALL, and USER catalog views, developers and DBAs can quickly and easily access database metadata through a familiar interface that requires no additional training to use. The following table is a sampling of some of the 46 supported views:

View Name	
ALL_OBJECTS	DBA_ROLE_PRIVS
ALL_SOURCE	USER_OBJECTS
ALL_SYNONYMS	USER_SOURCE
ALL_TABLES	USER_SYNONYMS
ALL_TAB_COLUMNS	USER_TABLES
ALL_USERS	USER_TAB_COLUMNS
ALL_VIEWS	USER_VIEWS
ALL_VIEW_COLUMNS	USER_VIEW_COLUMNS
DBA_ROLES	

Programming Flexibility and Drivers

Postgres Plus Advanced Server supports the most popular programming languages used to develop Oracle database applications. In addition to the publicly available database connectors listed below, Postgres Plus Advanced Server also provides interoperability and compatibility with the Oracle Call Interface (OCI). And it includes the Open Client Library (OCL) enabling applications based on OCI to run unchanged.

Programming Language	Supported Drivers
C	✓
C++	✓
JDBC	✓
ODBC	✓
Perl	✓
PHP	✓
Ruby	✓
.NET	✓

Migration From Oracle

The time, cost, and effort required to migrate any software system from one technology to another are often major deterrents, which prevent the undertaking of the task even though there may be clear, positive benefits for doing so.

EnterpriseDB has minimized the risks of the migration path from Oracle to Postgres Plus Advanced Server in the following ways:

- **Flexible and Easy-to-Use Migration Tools.** Postgres Plus Advanced Server contains both graphical and command line oriented migration tools. The graphical tool (Migration Studio) can migrate a database object, from a single table or PL/SQL program, up to an entire schema with just a few mouse clicks. The command line tool (Migration Toolkit) provides the capability to migrate more complex combinations of database objects. The following is a list of database objects that can be migrated:
 - Schemas
 - Data
 - Constraints
 - Sequences
 - Synonyms
 - Indexes
 - Users
 - Roles
 - Database Links
 - Packages
 - Stored Procedures
 - Triggers
 - Functions
 - Views
 - Range Partitioned Tables
 - List Partitioned Tables
- **Object Dependency Resolution.** One of the difficulties encountered is migrating objects that have a dependency upon each other such as tables with complex referential relationships.

Postgres Plus Advanced Server migration tools perform a dependency resolution pass to ensure that dependent objects are properly migrated.

- **Comprehensive Logging.** A comprehensive history log is produced listing the database objects that were migrated, the number of rows loaded into tables, and the volume of data that was transferred.
- **Comprehensive Feature and Function Compatibility.** As pointed out throughout this paper, Postgres Plus Advanced Server matches the most common features of Oracle. The functionality is supported by the same syntactic constructs as Oracle SQL and PL/SQL.

Once the data and business logic have been transferred, Postgres Plus Advanced Server is ready to run an enterprise's existing Oracle application. The net result is minimal time spent on porting objects, rewriting code, and re-testing functionality.

Note that this is a major differentiator between EnterpriseDB Postgres Plus Advanced Server and native PostgreSQL.

For customers looking to standardize on Oracle-based technology, Postgres Plus Advanced Server also provides browsing and one-click migration of the data and schema for Microsoft SQL Server, MySQL™, and Sybase™.

Database Replication

Postgres Plus Replication Server replicates data in Oracle and Postgres Plus Advanced Server databases across an enterprise in near real-time to meet a wide array of business challenges. Data can be replicated across distant geographies, complex enterprise data infrastructures, and heterogeneous operating platforms, including Linux, Solaris, Windows, and HP-UX.

Postgres Plus Replication Server enables enterprises to:

- Run reporting and other applications in Oracle environments at a fraction of Oracle's cost,
- Ensure availability and seamless disaster recovery of critical database services,
- Speed performance of Oracle and Postgres Plus Advanced Server databases,
- Transfer data across heterogeneous data sources while ensuring transactional integrity,

- Migrate to updated or upgraded versions of Postgres Plus Advanced Server with virtually no database downtime or risk to data.

Enterprise-Class Reliability and Scalability

Oracle databases are routinely used for high-volume, mission-critical applications. As Postgres Plus Advanced Server has inherited the legendary reliability of the open source PostgreSQL database and significantly enhanced the performance of PostgreSQL, even these applications may be safely transferred to Postgres Plus Advanced Server. EnterpriseDB has added the following features to PostgreSQL to enhance its reliability and scalability.

Point In Time Recovery (PITR)

Postgres Plus Advanced Server continuously maintains a Write Ahead Log (WAL), which, similar to Oracle REDO logs, contains every change made to the database's data files. In the event of a crash, Postgres Plus Advanced Server can replay these changes in the form of roll-forward database recovery.

Consequently, Postgres Plus Advanced Server can perform the following types of PITR:

- **Recovery to the point in time of a backup.** Postgres Plus Advanced Server can recover all WAL changes up to the last committed transaction found in the backup logs.
- **Recovery to an arbitrary point in time.** Postgres Plus Advanced Server can recover the WAL changes up to an arbitrary point in time from the last backup, as defined by a transaction identifier or timestamp.
- **Log-shipping and Disaster Recovery.** By transferring WAL files to remote systems, Postgres Plus Advanced Server can be brought back up and running in the event of a hardware crash or disaster situation.

Performance

A database that is suitable for enterprise applications must be highly scalable. Simply put, it needs to be really fast, both in high-volume transactional environments and when working with massive amounts of

data. EnterpriseDB has made significant enhancements to PostgreSQL to accommodate high-volume enterprise environments.

Performance enhancements include:

- **Online Transaction Processing (OLTP).** In addition to supporting row-level locking and table partitioning, Postgres Plus Advanced Server contains several internal optimizations for scalability in high-volume OLTP environments.
- **Dynamic Tuning.** By analyzing the hardware it's running on, Postgres Plus Advanced Server can determine its own configuration settings, reducing the need for both initial and ongoing tuning.
- **Bulk Loading.** When moving large amounts of data from mainframes or other databases into Postgres Plus Advanced Server, the time it takes to insert hundreds, thousands, or millions of rows is essential. Postgres Plus Advanced Server bulk loads are up to 12% faster than PostgreSQL's, and more importantly, provide error handling to prevent the loads from aborting so errors can be fixed after completion.
- **Asynchronous Pre-Fetch.** On Linux systems employing Redundant Array of Inexpensive Disks (RAID), Asynchronous Pre-Fetch schedules multiple, concurrent I/O requests for regular index scans and bitmap index scans. This is especially beneficial for data warehouse queries and extract, transform, and load (ETL) queries whereby multiple drives in the RAID array can be used to work on a small number of queries.
- **Infinite Cache.** Allows networked, commodity hardware to serve as cache servers providing an expandable, in-memory cache in between the Postgres Plus Advanced Server shared buffer cache and the disk buffer cache. Thus, it is effectively possible to read an entire database into memory across multiple cache servers. Infinite Cache can increase performance up to three orders of magnitude faster than disk caching alone. Infinite Cache is available on Linux systems. Finally, Infinite Cache is completely transparent to the client application which requires no special cache coding!

Security

Postgres Plus Advanced Server provides all of the security features expected of an enterprise-class RDBMS.

- **Role-Based Authentication.** Authentication based on login roles (user names).
- **Database Authorization.** Permitted access to any database can be controlled by a wide range of attributes including login role, role membership, originating client IP address, login connections only with password, login connections only with SSL encryption, etc.
- **Privilege-Based Authorization.** Access to database objects based on privileges granted to roles by the standard SQL GRANT and REVOKE statements.
- **Privileges Assignable by Role Membership.** Users can inherit privileges based upon the roles (groups) in which they are members.
- **Oracle-Style Definer's Rights and Invoker's Rights.** Access rights on program resources can be based on the privileges of who created the program or who is attempting to run the program.

Lightweight Directory Access Protocol (LDAP)

In addition Postgres Plus Advanced Server supports authentication based on standard LDAP implementations such as OpenLDAP, MS Active Directory, eDirectory, etc.

Pluggable Authentication Module (PAM)

Authentication is supported for standard PAM enabled servers. PAM allows system administrators to change the underlying server authentication mechanism without requiring PAM aware applications to recompile their authentication code.

Source Code Obfuscation (EDB*Wrap)

Program source code can sometimes contain proprietary, confidential or sensitive information that must not be viewed by unauthorized personnel. EDB*Wrap is a command line utility program similar to Oracle's wrap Utility. EDB*Wrap "scrambles" the source code of an EnterpriseDB SPL program so that it is no longer humanly readable. The obfuscated source code file is still executable, but remains unreadable within the system catalogs.

Oracle-Like Tools

Postgres Plus Advanced Server includes a robust set of integrated tools that enable developers and database administrators to create, deploy, monitor and tune Postgres Plus Advanced Server databases and applications. These tools have a look, feel, and operation that will be immediately familiar to Oracle DBAs and developers, enabling them to work comfortably with the tools with no additional training. A comprehensive set of tools is provided to facilitate the professional management of a Postgres Plus Advanced Server database environment.

Postgres Studio

Postgres Studio is an enterprise-class, cross-platform tool that serves as the main workbench for DBAs and application developers. Postgres Studio provides the following features:

- **Database Creation and Maintenance.** DBAs can create databases and database objects such as schemas, tables, indexes, constraints, views, tablespaces, and sequences. Various maintenance functions can be performed such as adding new columns, indexes, or constraints to tables, enabling or disabling triggers, backing up and restoring tables, rebuilding indexes, reclaiming unused table space, obtaining statistics, etc.
- **User and Role Maintenance.** DBAs can create users and define the roles to which they belong. Object privileges can be assigned to, or revoked from users and roles.
- **Data Browser and Editor.** Table data can be browsed and edited either through a graphical user interface or by entering SQL commands.
- **SQL Editor.** Application developers can create, edit, and run their SQL statements, SPL procedures, functions, triggers, and packages.
- **PL/SQL Debugger.** EnterpriseDB's Procedural Language Debugger facilitates the development and analysis of stored procedures, functions, and triggers in applications written for Oracle and Postgres Plus Advanced Server databases. The Procedural Language Debugger uniquely allows users to monitor and step through stored procedures and functions as they are called from applications.

Migration Studio

Migration Studio is the graphical tool for migrating from Oracle. It provides one-click transfer of all tables, views, packages, constraints, triggers, procedures, functions, views, sequences, indexes, and data stored in an Oracle database to Postgres Plus Advanced Server.

It also features the Oracle Database Browser, which enables a user to browse Oracle databases at the same time as browsing Postgres Plus databases.

EDB*Plus

EDB*Plus is a command line utility with a look and feel identical to Oracle SQL*Plus. EDB*Plus provides an environment to run SQL queries and updates, and to create and run EnterpriseDB SPL programs that would immediately be familiar to experienced Oracle DBAs and developers

EDB*Loader

EDB*Loader is a high performance bulk data loader for Postgres Plus Advanced Server databases. EDB*Loader runs from the command line and uses directives compatible with Oracle SQL*Loader.

Embedded SQL Pre-Compiler (ECPG)

Embedded SQL is a method of combining the computing power of a high-level language, such as C or C++, with the database manipulation capabilities of SQL by allowing the execution of SQL statements from within an application program. In Postgres Plus Advanced Server, an embedded SQL program is compiled in two steps.

The first step is to use ECPG, the embedded SQL pre-compiler for Postgres Plus Advanced Server, to translate the SQL statements embedded in the program into appropriate calls to the database driver library. The output of ECPG is C or C++ code with all the application portions intact. This C or C++ code can then be compiled and used in a standard C or C++ compiler.

EnterpriseDB Network

The EnterpriseDB Network provides customers with the confidence that their EnterpriseDB product suite contains the most up-to-date components available via Product Updates. EnterpriseDB Network subscribers

automatically receive real-time notification and delivery of product releases, updates, and patches. This not only saves DBAs the time and effort to search for updates, but also eliminates potential problems before they occur. All delivered software can be installed with only a few clicks, saving even more time and effort.

DBA Management Server

The EnterpriseDB DBA Management Server provides powerful database monitoring, profiling, reporting, and querying capabilities that enable DBAs and developers to analyze, manage, and tune multiple Postgres Plus Advanced Server and PostgreSQL databases from a single Web interface. The tool includes the ability to monitor overall database performance, the ability to monitor database queries for speed and efficiency, and integration with the EnterpriseDB Network.

DBA Monitoring Console

DBA Monitoring Console provides a real-time dashboard into the system CPU usage, memory usage, caching, and disk I/O of one or more database servers. The DBA Monitoring Console provides another aide for the monitoring and tuning of Postgres Plus Advanced Server databases.

Dynamic Runtime Instrumentation Tools Architecture (DRITA)

DRITA consists of a set of catalog views and functions that can be used to determine what events are causing particular database sessions, or the system as a whole to wait for execution to begin. A series of detailed and comprehensive reports can be produced in much the same manner as with the Oracle Statspack and Automatic Workload Repository (AWR) reports. DRITA allows DBAs and developers to discover bottlenecks impeding performance, improve code to increase performance, and verify their code adjustments.

Conclusion

Enterprises considering alternatives to Oracle have real concerns about the database system they are considering moving to, not the least of which is fear of the costs of changing databases, including costs related to performance degradation, application re-coding and personnel re-training, having outweighed the expected savings.

EnterpriseDB has included many Oracle compatible features in Postgres Plus Advanced Server that enables it to run most applications written for Oracle databases unchanged nearly eliminating application re-coding with performance close to or equaling Oracle.

Postgres Plus Advanced Server's detailed Oracle support includes the ability to execute Oracle's SQL and PL/SQL languages, the most common Oracle data dictionary views, and extensive suite of programming language drivers and connectors including the Oracle Call Interface (OCI), automated data and business logic transfer tools, an Oracle database replication server, enterprise-class reliability and scalability, and a robust set of integrated tools with a look, feel, and operation immediately familiar to Oracle professionals.

Any Oracle shop looking to reduce their license costs and vendor lock would be well served to investigate Postgres Plus Advanced Server as a compliment or replacement to their existing systems.

About EnterpriseDB

EnterpriseDB is the leading provider of enterprise class products and services based on PostgreSQL, the world's most advanced open source database. The company's Postgres Plus products are ideally suited for transaction-intensive and mixed load applications requiring superior performance, massive scalability, and compatibility with proprietary database products. Postgres Plus also provides an economical open source alternative or complement to proprietary databases without sacrificing features or quality. EnterpriseDB has offices in North America, Europe, and Asia. The company was founded in 2004 and is headquartered in Westford, MA. For more information, please call +1-732-331-1300 or visit <http://www.enterprisedb.com> .