

## PeopleSoft DDL & DDL Management

by David Kurtz, Go-Faster Consultancy Ltd.

***Since their takeover of PeopleSoft, Oracle has announced project Fusion, an initiative for a new generation of Oracle Applications building on the best of existing Oracle and PeopleSoft products. This is the third in a series of articles in which I will examine some aspects of PeopleTools, the proprietary technology developed by PeopleSoft, and speculate about Fusion.***

In my first article<sup>1</sup> about PeopleSoft and PeopleTools I described the PeopleTools tables. These hold most of the on-line application as meta-data that is interpreted as the application executes. Some of the PeopleTools tables correspond to the database catalogue. PeopleTools uses these tables both to specify data structures and later to build DDL statements.

When developing a PeopleTools component, the first requirement is to specify the data structures that you will be accessing. These are called records in PeopleTools, and they may correspond to tables or views in the database, or they may also be working storing structures that only exist in memory whilst the application executes. The PeopleTools Application Designer also specifies the indexes that are to be built. **The concept is similar to Oracle forms, where you put fields into blocks, except that you have to define the database structures first.**

One of the useful features of the PeopleTools Application Designer is that it maintains its own data dictionary, and will then generate the DDL required to implement the differences between that and the database catalogue. Thus when you customise an existing table or view, or more significantly when you apply a patch or upgrade to an object that you have already customised, you don't have to manually craft an ALTER TABLE statement, it will be generated for you.

An implication of this is that all DDL changes must be made via the Application Designer. If the DBA were simply to make a change directly, then the next time that Application Designer is called upon to rebuild that object it would see the change as a difference to be resolved and the change would be lost. Worse still, this may reintroduce the problem that caused the DBA to make the change in the

---

<sup>1</sup> Oracle Scene, Issue 23, Autumn 2005, p32 – also available in the UKOUG library on [www.ukoug.org](http://www.ukoug.org)

first place. For example, a DBA may add an index to resolve a performance problem, but Application Designer will drop it when it builds a DDL script for that table.

### DDL Models

I want to look next at how Application Designer builds DDL statements. PeopleTools is delivered with a set of DDL models for each database platform for each of up to 5 DDL statements: CREATE TABLE, CREATE INDEX, CREATE TABLESPACE, ANALYZE TABLE COMPUTE STATISTICS, ANALYZE TABLE ESTIMATE STATISTICS (the ANALYZE models are only available for Oracle and DB2). Each model consists of a string with variables that are substituted to make the DDL command.

#### DDL Model Defaults

**Platform ID:** 2 Oracle

**Sizing Set:** 0

Copy...

DDL
Find | View All
First 1 of 5 Last

**Statement Type:** Table + -

**\*Model SQL:**

CREATE TABLE [TBNAME] ([TBCOLLIST]) TABLESPACE [TBSPCNAME]  
 STORAGE (INITIAL \*\*INIT\*\* NEXT \*\*NEXT\*\* MAXEXTENTS \*\*MAXEXT\*\*  
 PCTINCREASE \*\*PCT\*\* PCTFREE \*\*PCTFREE\*\* PCTUSED

**Parameter Count:** 6

Parameters
Customize | Find | View All
First 1-3 of 6 Last

DDL Parm	DDL Parameter Value	+ -
INIT	40000	+ -
MAXEXT	UNLIMITED	+ -
NEXT	100000	+ -

Variables delimited by square brackets (e.g. [TBNAME]) are generated by PeopleTools code. The user has no control over these. When you create a table, the entire column list is generated by PeopleTools and cannot be intercepted.

Additional variables delimited by two asterisks (e.g. \*\*PCTFREE\*\*) can be defined in the DDL model. Default values are also specified, and can then be overridden in the record definition in Application Designer. PeopleSoft delivers some additional variables, the user can choose to add others.

CREATE TABLE and CREATE INDEX are mainly used by Application Designer when building or altering a table or index. DDL models are also embedded in the header of a Data Mover export, and are then used during import.

CREATE TABLESPACE can be used by Data Mover when importing data. However, as a DBA I create tablespaces myself. Even the delivered PeopleSoft scripts disable this feature, and so I have never used it!

ANALYZE models were introduced in PeopleTools 8.1. Batch programs that truncate and populate tables used for working storage, can cause the optimiser can make a poor choices because the statistics do not match reality. To address this, PeopleSoft introduce two ANALYZE DDL models in PeopleTools 8.1 that either fully compute statistics or estimate them using the default sample size. A new PeopleTools meta-code command *%UpdateStats* was added to Application Engine that uses these DDL models to build an ANALYZE command. Application Engine batch programs were amended to update statistics when appropriate.

### ***Other DDL***

Not all DDL generated by the Application Designer is based on DDL models. There is no model for CREATE VIEW, the code is generated entirely within Application Designer. A table can be altered either with an ALTER TABLE command, or renamed, recreated and repopulated. The whole of the ALTER table command is generated within Application Designer, and only the CREATE INDEX model is used when a table is recreated. PeopleSoft uses database triggers to assist with synchronisation with mobile devices. These triggers are also generated by Application Designer. Triggers for auditing are generated by an SQR program.

### ***Space Management***

PeopleSoft created a fairly flexible system for generating platform specific DDL. There are various opportunities for the DBA to use this to improve management and performance of the system.

Oracle DBAs have to worry about physical storage attributes for objects. For comparison, on SQL Server (as far as I can tell) there are neither the concepts of tablespaces or physical storage attributes. You just create the table.

It is possible to set values for INITIAL and NEXT in the storage clause via Application Designer. However, those of us managing PeopleSoft on Oracle databases will almost certainly be using locally managed tablespaces and will have learnt that the number of extents in an object is not a significant performance issue.

Instead it is now much more important to allocate a table to a tablespace with an appropriate extent size, and also to set PCTFREE and PCTUSED to appropriate values. PeopleSoft defaults PCTFREE to 10, and PCTUSED to 40. This value is almost certainly not appropriate for every table in the system. At best, setting these parameters inappropriately for a table will waste space and increase both physical and logical I/Os, at worst may lead to rows chained across blocks, and have to do an extra I/O in order to retrieve a row. Different tables will require different settings depending upon how they are used.

Unfortunately PeopleSoft doesn't deliver many overrides for these DDL Model variables, so the DBA needs to learn how the application behaves and set appropriate values. The values must be set in Application Designer, not just in the database.

There are other parameters that can, if necessary, easily be added to the DDL Model and maintained within Application Designer, such as CACHE and BUFFER POOL.

### **Statistics**

When PeopleSoft introduced the two ANALYZE DDL models, Oracle was already recommending that the *dbms\_stats* PL/SQL package should be used instead. These DDL models can be altered to call PL/SQL, but you need to create your own PL/SQL function<sup>2</sup> to split the schema and table names that must be passed to different parameters in *dbms\_stats*.

### **Limitations**

There are a number of Oracle RDBMS features that is often useful to use in PeopleSoft systems, but which cannot be implemented via DDL Models and so maintained with Application Designer.

---

<sup>2</sup> available from [www.psftdba.com](http://www.psftdba.com)

- Application Designer cannot generate the DDL for partitions. Although it is possible to create additional variables to hold the PARTITION keyword and clause, it would be very hard to manage, especially if different partitions go into different tablespaces and/or have different storage options. It would be better to manage the few tables that are to be partitioned manually outside Application Designer.
- Function-based indexes are useful for case-insensitive searches in PeopleSoft (e.g. *WHERE UPPER(NAME) LIKE...*). However, Application Designer will not let you put the corresponding function into the CREATE INDEX DDL. The list of columns is generated by an internal variable and cannot be intercepted.
- Index Organised Tables require a primary key constraint be created, and constraints cannot be added to the column list because this is created by a single internal variable. Also, the list of columns in the primary key are in another internal variable using in CREATE INDEX that is not available when the CREATE TABLE is processed.
- It is possible to customise the create table model so that global temporary tables can be created. The GLOBAL TEMPORARY keyword can be added, and the physical attributes can be masked out as comments. However, the result is messy, and I prefer to manage them manually.

The result is that these features are never used in a PeopleSoft system as delivered. It is then only reasonable to introduce them where there is an overriding business case, when the resultant performance benefit outweighs the overhead on manual management. That management overhead includes making sure that the feature is not accidentally lost because Application Designer should not be used to manage that table. For example, it is easy to introduce a function-based index, but you have to make sure that it doesn't get accidentally dropped by an Application Designer script.

### **Conclusion**

Automating the process of DDL management removes an opportunity for manual error. You can be confident when applying patches or upgrades to environments that at least the database structures will be correct.

There is some flexibility to permit the DBA to apply appropriate attributes to tables, but this does not go far enough. DDL Models are not sufficiently flexible as to permit some very useful database features.

Sometimes database features cannot be introduced via PeopleTools, and a decision has to be made whether their benefits justify the overhead of manual management.

### ***Fusion***

DBAs often complain when they are told that DDL changes for a particular system must be managed via a proprietary tool, such as Application Designer. They want to use the same tools and techniques for all their databases on each platform. PeopleSoft is certified to run on a variety of RDBMS, not just Oracle. Presumably the same will apply to Fusion (we still don't know for certain), and all of these databases have their own management tools.

The challenge is how to make this functionality available to both developer and DBA on all certified platforms. PeopleSoft chose a lowest common denominator approach, and did not use any features that were not common to all certified platforms.

On Oracle RDBMS, an option would be to make this functionality available to the DBA via OEM. But then what about other database platforms? I don't have an answer to that.

Application Designer has evolved to be a complete development environment for PeopleSoft, and is one of the reasons that PeopleSoft developers are acknowledged as been several times more productive than Oracle Applications developers. I would hope to see an integrated development tool for Fusion that will include DDL management, and would also have support for at least the database features discussed in this article.

*David Kurtz is a performance specialist working Enterprise PeopleSoft applications and Oracle ([www.go-faster.co.uk](http://www.go-faster.co.uk)). He is the chair of UKOUG UNIX SIG. He is also the author of 'PeopleSoft for the Oracle DBA', published by Apress ([www.psftdba.com](http://www.psftdba.com)).*