**Naming Conventions**

**Entities & Tables**

All entity names should be singular and may have spaces. These are replaced with '\_' by Oracle Designer during table creation.

APPLICATION

APPLICATION FUNCTION

APPLICATION FUNCTION ROLE

All table names should be plural. If the table name contains several words, only the last one should be plural.

APPLICATIONS

APPLICATION\_FUNCTIONS

APPLICATION\_FUNCTION\_ROLES

**Aliasing**

All entities and tables should be defined a unique alias that is used as part of the foreign key definitions. The alias should be an abbreviation of the name.

APPLICATIONS = APPL (4)

APPLICATION\_FUNCTIONS = APFU (2:2)

APPLICATION\_FUNCTION\_ROLES = APFR (2:1:1)

APPLICATION\_FUNCTION\_ROLE\_BANANAS = AFRB (1:1:1:1)

APPLICATION\_FUNCTION\_ROLE\_BANANA\_APPLES = (Do what you like!)

The bracketed numbers indicate the number of letters used from each word in the name.

Oracle has a 30 character restriction on object names, so table aliases are useful to reduce the length of object names.

**Attributes & Columns**

Attributes and columns should not be prefixed with a table alias. For the most part this is unnecessary and often very messy.

**Keys & Their Columns**

Primary Keys are named after the table or its alias with the suffix of '\_PK'.

Table : APPLICATIONS

Primary Key: APPLICATIONS\_PK

or

APPL\_PK

Unique Keys are named after the table or its alias with the suffix of '\_UK'. If more than one unique key is present you may need to add further information to make the key name unique.

Table : APPLICATIONS

Unique Key : APPLICATIONS\_UK

or

APPL\_UK

Foreign Keys are usually named using the two table aliases, from and to, with the suffix of '\_FK':

Relationship: APPLICATION\_FUNCTIONS -> APPLICATIONS

Foreign Key : APFU\_APPL\_FK

The foreign key column on the dependent table may be named using the singular form of the full table name or the alias, along with the column name.

APPLICATION\_FUNCTIONS.APPLICATION\_ID -> APPLICATIONS.ID

or

APPLICATION\_FUNCTIONS.APPL\_ID -> APPLICATIONS.ID

I prefer the singular form of the full table name, but the 30 character limit forces the use of the alias at times.

**Indexes**

Indexes are created implicitly to support PKs and UKs. These indexes have the same name as the constraint.

As a rule of thumb, all foreign key columns should be indexed. Indexes to support foreign keys should be named using the foreign key name with the suffix '\_I'.

Foreign Key : APFU\_APPL\_FK

Suporting Index: APFU\_APPL\_FK\_I

Other indexes should be created with meaningful names, usually incorporating the table alias and the column name(s) where possible, along with the suffix '\_I'.

**Triggers**

Trigger names should be made up of the table name, an acronym representing the triggering action and the suffix "\_TRG".

Table : APPLICATIONS

Action: BEFORE INSERT STATEMENT-LEVEL

Name : APPLICATION\_BIS\_TRG

Action: AFTER INSERT AND UPDATE ROW-LEVEL

Name : APPLICATION\_AIUR\_TRG

**Other Objects**

Other database objects have a suffix that identifies their object type.

Check Constraints : <name>\_CHK

Sequences : <name>\_SEQ

Views : <name>\_V

Materialized Views: <name>\_MV

Types : <name>\_T

Directories : <name>\_DIR

External Tables : <name>\_EXT

PL/SQL Packages : <name>\_API

PL/SQL Procedures : <name>\_PRC

PL/SQL Functions : <name>\_FUN

**PL/SQL Variables**

PL/SQL variables are prefixed with a single letter, if possible, to indiate their type or usage.

Package Global Variables: g\_variable\_name

Local Variables : l\_variable\_name

Types : t\_type\_name

Cursors : c\_cursor\_name

Exceptions : e\_exception\_name

Input Parameters : i\_parameter\_name

Outut Parameters : o\_parameter\_name

In/Out Parameters : io\_parameter\_name

Sometimes I flip back to using a generic "p\_" prefix for parameters, regardless of their IN/OUT usage. Old habits dies hard.

**File Extensions**

File extensions include:

.pks – Package specification.

.pkb – Package body.

.sql – Everything else.

# Naming Standards

## Overview of Naming Standards

This section provides you with information you need to define all your database and form objects. It provides naming standards and header information for all your objects and files.

The naming standards are grouped into the following sections:

* [Database objects](https://docs.oracle.com/cd/E18727_01/doc.121/e12897/T302934T458266.htm#I_devnamedb)
* [Form objects](https://docs.oracle.com/cd/E18727_01/doc.121/e12897/T302934T458266.htm#I_devnameform)
* [File standards](https://docs.oracle.com/cd/E18727_01/doc.121/e12897/T302934T458266.htm#I_filestand)
* [PL/SQL Packages and Procedures](https://docs.oracle.com/cd/E18727_01/doc.121/e12897/T302934T458266.htm#I_nsx2Dplsql)
* [Reserved Words](https://docs.oracle.com/cd/E18727_01/doc.121/e12897/T302934T458266.htm#I_nsx2Dresword)

## Naming Standards and Definitions

In general, make names meaningful and brief. Do not use generic, all-purpose phrases like "COMMON", "MISC", "OTHER", or "UTILITY" in the name.

## Database Objects

In addition to specific naming conventions for particular objects, all database objects should be named without using any special characters. Database object names should use only ASCII letters, numbers, and underscores, and they should always begin with a letter. Note that database object names are case-insensitive, so "Name" would be the same as "NAME".

Include header information when you create your objects. The header should include the following documentation

* Name
* Purpose
* Arguments

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Arg1** | Describe arg1 |
| **Arg2** | Describe arg2 |

* Notes
  1. Special usage notes
  2. Special usage notes
* History

DD-MON-YY J. Doe Created

### Tables

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *prod\_objects* *prod* is the product short name, and *objects* is the name of the objects stored in the table and should be plural. The table name should be 20 characters or less. It can be longer, but you will need to abbreviate it for the table handler package name, which must be 24 characters or less. |

**Example**

PO\_VENDORS, AS\_LOOKUPS

### Views

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *table\_*V or *table\_criteria\_*V |

*table* is the name of the root table the view is based on. The criteria is a qualifier of the objects shown by the view. Use the criteria qualifier if using table name alone is not unique, the view is based on a join of 2 or more tables, the view contains a WHERE clause, or the view is unusual.

**Example**

PO\_VENDORS\_CHICAGO\_V, AS\_LOOKUPS\_RANK\_V

### Triggers

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *table\_*Ti |

*table* is the name of the table on which the trigger is based, and *i* is a unique ID starting at 1.

**Example**

PO\_HEADERS\_T1

### Synonyms

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *table* |

Your synonym should have the same name as the table or view it is based on.

Using two different names (the synonym name and the underlying object name) is confusing. If you change object names, you should clean up your code instead of creating synonyms.

### Constraints

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Primary Key** | *table\_*PK |
| **Unique** | *table\_*Ui |
| **Foreign Key** | *table\_*Fi |
| **Check** | Use Message Dictionary message naming standards.  See: [Overview of Message Dictionary](https://docs.oracle.com/cd/E18727_01/doc.121/e12897/T302934T462354.htm#I_ox2Dmd) |

*table* is the name of the table on which the constraint is created, while *i* is a unique id starting at 1. You should name all of your constraints so that you can enable or disable them easily.

### Packages

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *prod\_module* or *prod\_description* |

*prod* is the product short name, *module* is a functional module, and *description* is a one or two word explanation of the purpose. Stored package names should be 24 characters or less. For library packages, your package should be unique within 27 characters. Wrapper packages use a three character prefix. Select a description that helps simplify the names of procedures in the package.

**Example**

OE\_SCHEDULE, AOL\_FLEXFIELD

### Packaged Procedures

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *verb\_noun* |

*verb\_noun* is a brief explanation of the purpose. Do not reuse the product short name or any part of the package name in the procedure name. Remember that you will invoke the procedure as package procedure. For example, if the package name is APP\_ORDER\_BY, then the procedures should simply be named APPEND and REVERT. Be careful you don't name your package procedure a SQL, PL/SQL, Oracle Forms, or other reserved word, or redefine an Oracle Forms built-in.

**Example**

CALCULATE\_PRICE\_VARIANCE, TERMINATE\_EMPLOYEE

### Table Handler Package and Procedures

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Package** | *table\_*PKG |

*table* is the name of the table on which the package acts (to insert, update, delete, or lock a record, or to check if a record in another table references a record in this table). The package name should be 24 characters or less.

**Example**

PO\_LINES\_PKG

### Private Packages

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *package\_*PRIVATE |

*package* is the name of the owning package.

**Example**

APP\_ITEM\_PROPERTY\_PRIVATE

### Forms PL/SQL Program Units (Stand-Alone Procedures)

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *prod\_verb\_noun* |

*prod* is the product short name, and *verb\_noun* is a brief explanation of the purpose.

**Example**

AP\_INITIALIZE\_FORM

### PL/SQL Variables

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *variable* or *X\_variable* |

*variable* should be a logical, meaningful, and concise name. Precede variable name with X when the variable is used in a SQL statement, so that there is no possibility of conflicts with database object names or confusion as to which are PL/SQL variables and which are database objects.

**Example**

X\_header\_id

### PL/SQL Global Variables

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | G*\_variable* |

*variable* should be a logical, meaningful, and concise name. Precede variable name with G to distinguish global variables from local variables.

**Example**

G\_set\_of\_books\_id

## Form Objects

In general, objects that can show multiple items (record groups, LOVs, etc.) should have plural names, while singular objects (modules, blocks) have singular names.

### Modules

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *file name* |

Your form module name should match your form file name. For example, if a form is called POXPOMPO.fmb, make sure the Module Name (visible in the Designer) is POXPOMPO. This is especially important if you reference objects from your form. ZOOM also relies on the Module Name being correct.

### Record Groups

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *object* |

*object* is the name of the object that the record group contains.

### Oracle Forms Global Variables

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *prod\_variable* |

*prod* is the product short name, and *variable* is the name you would normally give to the variable.

**Example**

PO\_SECURITY\_LEVEL, MFG\_ORGANIZATION

### Item

Use logical, meaningful, and concise names. Note that table columns based on LOOKUP\_CODES should have a "\_CODE" or "\_FLAG" suffix, and the displayed meaning item should have the same name but without the suffix.

Mirror Items use the name of the item plus a "\_mir" suffix. So if the item in the detail portion is "ename", name the mirror-item "ename\_mir".

### Blocks

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *case\_short\_name* or *object* *case\_short\_name* is the CASE block short name, and *object* is the name of the objects in the block. The block name should be 14 characters or less. |

**Example**

ORDER, LINES

### Special Blocks

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Block containing toolbar** | TOOLBAR |
| **Block containing control items** | CONTROL |
| **Block containing display-only, context info** | CONTEXT |
| **Blocks to submit concurrent requests** | *program* or *report* |
| **Non-database blocks (such as search blocks)** | *action* or *action\_object* |

If the block is shared with other forms, make the block name unique by preceding it with the name of your form.

### Canvasses

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *object* *object* is the name of the object shown on the canvas. |

### Alternative Region Stacked Canvasses

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *block\_region* The region field belong to *block*. *region* describes the fields shown in the region. For example, a block LINES has two alternative regions, one showing price information (base price, discounted price, total price) and the other showing account information. The alternative region stacked canvases are named LINES\_PRICE and LINES\_ACCOUNT. |

**Example**

LINES\_DESCRIPTION, LINES\_PRICES

### Query-Find Canvasses, Windows, and Blocks

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | QF*\_object* To distinguish windows, blocks and canvasses used for Find Windows, prefix the object name with "QF\_". |

### Windows

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *object* *object* is the name of the object shown in the window. |

### LOVs

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *object* *object* is the name of the object shown in the LOV. |

**Example**

ORDER\_SALESREPS, LINE\_SALESREPS, FREIGHT\_CODES

### LOV Record Groups

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *object* or *object\_criteria* *object* is the name of the objects in the record group, usually the same as the basic item or LOV name. *criteria* is a brief description of why specific objects are included in the record group. Use the criteria description only if using object name alone is not unique. Abbreviate the object name or criteria description if *object\_criteria* exceeds 30 characters. |

### Query LOVs and Related Record Groups

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | QF*\_object* To distinguish between LOVs and record groups used for entry from those used for querying purposes (such as Find Windows), prefix the object name with "QF\_". For example, QF\_FREIGHT\_CODES, QF\_DEMAND\_CLASSES. |

### Alternative Region LOVs and Related Record Groups

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *block\_*REGIONS |

To distinguish the LOVs and record groups invoked when pressing KEY-MENU in an alternative region, append \_REGIONS to the block name.

**Example**

LINES\_REGIONS, ORDERS\_REGIONS

### Relations

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *master\_detail* *master* is the name of the master block in the relation, and *detail* is the name of the detail block in the relation. |

### Item and Event Handler Packages and Procedures

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Package** | *block* or *form* |
| **Item Handler Procedure** | *item* |

*block* is the name of the block owning the items, *form* is the name of the form, and *item* is the name of the item on which the item handler procedure acts.

### Combination Block Parameter

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *block*\_RECORD\_COUNT *block* is the name of the combination block. |

## File Standards

All file names must be no greater than 8 chars in length plus a three character extension (FILENAME.ext). The files names must be all caps except for the extension. This will provide a consistent naming scheme across all platforms.

### Form Source File Names

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *pppggxxx.fmb, pppggxxx.fmx,* or *pppggxxx.fmt* |

*ppp* is the two or three character product short name, *g* is a two-character abbreviation for the functional group, and *xxx* is a three-character abbreviation for the explanation of the purpose.

*fmb* is the suffix for the source binary file, *fmx* is the suffix for the executable binary file, and *fmt* is the suffix for the source text file. The files reside in $prod/forms/US (or the platform equivalent).

### APPSTAND Equivalents

The APPSTAND form provides many standard settings and components that other forms reference in. Particular applications and functional groups may create their own standard form that they use for reference.

The naming convention of these APPSTAND equivalents is:

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *PPP*STAND |

*PPP* is the two or three character product short name, such as OE for Order Entry (OESTAND) or GL for General Ledger (GLSTAND).

## PL/SQL Packages, Procedures and Source Files

Note that PL/SQL packages and procedures are documented slightly differently: Packages do not have Arguments sections and procedures do not need to have History sections.

Begin all SQL and PL/SQL files with the following lines (after the copyright header):

SET VERIFY OFF

WHENEVER SQLERROR EXIT FAILURE ROLLBACK;

End all SQL and PL/SQL files with the following lines:

COMMIT;

EXIT;

### PL/SQL Stored Package Source File Names

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *pppgxxxs.pls* or *pppggxxb.pls* |

*ppp* is the two or three character product short name, *g* is a one-character abbreviation for the functional group, and *xxx* is a three-character abbreviation for the explanation of the purpose. If you do not need three characters for that purpose, you may use two characters for the functional group. *s* indicate a specification file, and *b* indicates a body file. Each file defines only one package specification or body. The files reside in $prod/install/sql (or the platform equivalent).

### Table Handler Package Source File Names

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *pppgixxs.pls* and *pppgixxb.pls* |

*i* indicates (table) "interface."

### PL/SQL Library File Names

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *pppggxxx.pll, pppggxxx.plx,* and *pppggxxx.pld* |

The files reside in $prod/plsql and, for run-time, in $au/plsql (or the platform equivalent).

### Icon File Names

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *ppxxxxxx.ico, ppxxxxxx.bmp* |

*pp* is the two-character product short name, and *xxxxxx* is an icon name up to six characters long. The files reside in $TK2\_ICON (or the platform equivalent).

### Package Creation Scripts for Patches

|  |  |
| --- | --- |
| **Variable** | **Description** |
| **Standard** | *PPPGGXXS.pls* |
| **Standard** | *PPPGGXXB.pls* |

*PPP* is the two or three character product short name, *GG* is an abbreviation for the functional group, and *XX* is an abbreviation for the specific functionality. *S* indicate a specification file, and *B* indicates a body file.

## Reserved Words

In addition to all words reserved by the Oracle database, PL/SQL, and Oracle Forms, do not use any words that begin with the following letters (including the words themselves):

* FOLDER
* CALENDAR
* APPCORE