

UTL_RECOMP

The UTL_RECOMP package recompiles invalid PL/SQL modules, invalid views, Java classes, indextypes and operators in a database, either sequentially or in parallel.

This chapter contains the following topics:

- [Overview](#)
- [Operational Notes](#)
- [Examples](#)
- [Summary of UTL_RECOMP Subprograms](#)

UTL_RECOMP Overview

This script is particularly useful after a major-version upgrade that typically invalidates all PL/SQL and Java objects. Although invalid objects are recompiled automatically on use, it is useful to run this script prior to operation because this will either eliminate or minimize subsequent latencies due to on-demand automatic recompilation at runtime.

Parallel recompilation can exploit multiple CPUs to reduce the time taken to recompile invalid objects. The degree of parallelism is specified by the first argument to [RECOMP_PARALLEL Procedure](#).

In general, a parallelism setting of one thread for each available CPU provides a good initial setting. However, please note that the process of recompiling an invalid object writes a significant amount of data to system tables and is fairly I/O intensive. A slow disk system may be a significant bottleneck and limit speedups available from a higher degree of parallelism.

UTL_RECOMP Operational Notes

UTL_RECOMP has several operational notes.

- This package uses the job queue for parallel recompilation.
- This package must be run using `SQL*PLUS`.
- You must be connected `AS SYSDBA` to run this script.
- This package expects the following packages to have been created with `VALID` status:
 - `STANDARD` (`standard.sql`)
 - `DBMS_STANDARD` (`dbmsstdx.sql`)
 - `DBMS_JOB` (`dbmsjob.sql`)
 - `DBMS_RANDOM` (`dbmsrand.sql`)
- There should be no other DDL on the database while running entries in this package. Not following this recommendation may lead to deadlocks.

UTL_RECOMP Examples

These examples show various ways that UTL_RECOMP can recompile objects.

- Recompile all objects sequentially:

```
EXECUTE UTL_RECOMP.RECOMP_SERIAL();
```

- Recompile objects in schema SCOTT sequentially:

```
EXECUTE UTL_RECOMP.RECOMP_SERIAL('SCOTT');
```

- Recompile all objects using 4 parallel threads:

```
EXECUTE UTL_RECOMP.RECOMP_PARALLEL(4);
```

- Recompile objects in schema JOE using the number of threads specified in the parameter JOB_QUEUE_PROCESSES:

```
EXECUTE UTL_RECOMP.RECOMP_PARALLEL(NULL, 'JOE');
```

Summary of UTL_RECOMP Subprograms

This table lists the UTL_RECOMP subprograms and briefly describes them.

Table 299-1 UTL_RECOMP Package Subprograms

Subprogram	Description
RECOMP_PARALLEL Procedure	Recompiles invalid objects in a given schema, or all invalid objects in the database, in parallel
RECOMP_SERIAL Procedure	Recompiles invalid objects in a given schema or all invalid objects in the database

RECOMP_PARALLEL Procedure

This procedure uses the information exposed in the `DBA_Dependencies` view to recompile invalid objects in the database, or in a given schema, in parallel.

Syntax

```
UTL_RECOMP.RECOMP_PARALLEL(  
  threads IN PLS_INTEGER DEFAULT NULL,  
  schema  IN VARCHAR2     DEFAULT NULL,  
  flags   IN PLS_INTEGER DEFAULT 0);
```

Parameters

Table 299-2 RECOMP_PARALLEL Procedure Parameters

Parameter	Description
threads	The number of recompile threads to run in parallel. If NULL, use the value of 'job_queue_processes'.
schema	The schema in which to recompile invalid objects. If NULL, all invalid objects in the database are recompiled.

Table 299-2 (Cont.) RECOMP_PARALLEL Procedure Parameters

Parameter	Description
flags	Flag values are intended for internal testing and diagnosability only.

Usage Notes

The UTL_RECOMP_COMPILED table is updated with objects that RECOMP_PARALLEL has attempted to compile, or that it has compiled so far. If you monitor the progress of the recompilation, then the number of objects should increase with time. These tables are used internally by RECOMP_PARALLEL to prevent an object from being compiled more than once.

Table 299-3 Tables Generated by UTL_RECOMP.RECOMP_PARALLEL

Table	Purpose
UTL_RECOMP_ERRORS	A table of objects that are sorted into groups, based on dependencies. It is used internally by RECOMP_PARALLEL to guide parallel recompilation.
UTL_RECOMP_COMPILED	A table of objects that RECOMP_PARALLEL has attempted to compile. It is used internally by recomp_parallel to prevent an object from being compiled more than once.
UTL_RECOMP_COMP_IDX1	An index created on UTL_RECOMP_COMPILED, which uses object number as the primary key.
UTL_RECOMP_SKIP_LIST	A table of objects that RECOMP_PARALLEL will not attempt to compile. Typical usage is for RECOMP_PARALLEL to call UTL_RECOMP.POPULATE_UTL_RECOMP_SKIP_LIST before the update starts so that this table is filled with previously identified invalid objects. Afterward, when RECOMP_PARALLEL is run after upgrade, it will not attempt to compile those previously identified invalid objects.

Table 299-4 Views Generated by UTL_RECOMP.RECOMP_PARALLEL

View	Purpose
UTL_RECOMP_ALL_OBJECTS	A view that includes all objects that can be compiled.
UTL_RECOMP_INVALID_ALL	A view that selects invalid objects from UTL_RECOMP_ALL_OBJECTS that have not been compiled and are not on the skip list.
UTL_RECOMP_INVALID_SEQ	A view that selects objects from UTL_RECOMP_INVALID_ALL that must be compiled sequentially.
UTL_RECOMP_INVALID_PARALLEL	A view that selects objects from UTL_RECOMP_INVALID_ALL that can be compiled in parallel.
UTL_RECOMP_INVALID_JAVA_SYN	A view that selects from UTL_RECOMP_ALL_OBJECTS objects that are invalid JAVA synonyms.
UTL_RECOMP_CIRCULAR_MV	A view that selects from UTL_RECOMP_ALL_OBJECTS materialized view objects that have a circular dependency chain.
UTL_RECOMP_INVALID_MV	A view that selects from UTL_RECOMP_ALL_OBJECTS materialized view objects that are invalid but don't have a circular dependency chain.

Table 299-4 (Cont.) Views Generated by UTL_RECOMP.RECOMP_PARALLEL

View	Purpose
UTL_RECOMP_ALL_STUBS	A view that includes all stub objects that have not been compiled, and are not on the skip list. When this view is used in RECOMP_PARALLEL, it is restricted to show only stubs in the current edition.

RECOMP_SERIAL Procedure

This procedure recompiles invalid objects in a given schema or all invalid objects in the database.

Syntax

```
UTL_RECOMP.RECOMP_SERIAL(  
    schema    IN    VARCHAR2    DEFAULT NULL,  
    flags     IN    PLS_INTEGER DEFAULT 0);
```

Parameters

Table 299-5 RECOMP_SERIAL Procedure Parameters

Parameter	Description
schema	The schema in which to recompile invalid objects. If NULL, all invalid objects in the database are recompiled.
flags	Flag values are intended for internal testing and diagnosability only.