http://www.dba-oracle.com/job\_scheduling/resource\_manager.htm

SQL> @table\_comments.sql sys dba\_rsrc

TABLE\_NAME                     COMMENTS  
------------------------------ -------------------------------------  
DBA\_RSRC\_CONSUMER\_GROUPS       all the resource consumer groups  
DBA\_RSRC\_CONSUMER\_GROUP\_PRIVS  Switch privileges for consumer groups  
DBA\_RSRC\_GROUP\_MAPPINGS        all the consumer group mappings  
DBA\_RSRC\_MANAGER\_SYSTEM\_PRIVS  system privileges for the resource   
                               manager  
DBA\_RSRC\_MAPPING\_PRIORITY      the consumer group mapping attribute   
                               priorities  
DBA\_RSRC\_PLANS                 All the resource plans  
DBA\_RSRC\_PLAN\_DIRECTIVES       all the resource plan directives

7 rows selected.

Modifications to resource management must be complete and valid before they are applied to the system.  For this reason, most operations using the dbms\_resource\_manager package are performed in a pending area where they are validated before being applied.  The following code shows the procedure calls which must enclose any modifications:

BEGIN  
  DBMS\_RESOURCE\_MANAGER.clear\_pending\_area;  
  DBMS\_RESOURCE\_MANAGER.create\_pending\_area;

  -- Do something

  DBMS\_RESOURCE\_MANAGER.validate\_pending\_area;  
  DBMS\_RESOURCE\_MANAGER.submit\_pending\_area;  
END;  
/

To illustrate the use of the resource manager, assume there is a system in which OLTP operations must take priority over batch operations during the day.  At night, the situation is reversed such that batch operations take priority over OLTP operations.

To model this scenario, create two new consumer groups for the OLTP and batch tasks using the create\_consumer\_group procedure.

PROCEDURE create\_consumer\_group(  
  consumer\_group  IN  VARCHAR2,  
  comment         IN  VARCHAR2,  
  cpu\_mth         IN  VARCHAR2 DEFAULT 'ROUND-ROBIN')

The create\_consumer\_groups.sql script uses this procedure to create the OLTP and batch consumer groups.

\* create\_consumer\_groups.sql

--

CONN sys/password AS SYSDBA  
BEGIN  
  DBMS\_RESOURCE\_MANAGER.clear\_pending\_area;  
  DBMS\_RESOURCE\_MANAGER.create\_pending\_area;

  -- Create the consumer groups  
  DBMS\_RESOURCE\_MANAGER.create\_consumer\_group(  
    consumer\_group => 'oltp\_consumer\_group',  
    comment        => 'OLTP process consumer group.');

  DBMS\_RESOURCE\_MANAGER.create\_consumer\_group(  
    consumer\_group => 'batch\_consumer\_group',  
    comment        => 'Batch process consumer group.');

  DBMS\_RESOURCE\_MANAGER.validate\_pending\_area;  
  DBMS\_RESOURCE\_MANAGER.submit\_pending\_area;  
END;  
/

The consumer\_groups.sql script listed below uses the*dba\_rsrc\_consumer\_groups*view to display information about the consumer groups that have been created.

\* consumer\_groups.sql

column comments format a60

select  
   consumer\_group,  
   comments  
from  
   dba\_rsrc\_consumer\_groups  
order by  
   consumer\_group  
;

The output from this script is displayed below.

SQL> @consumer\_groups.sql

CONSUMER\_GROUP                 COMMENTS  
------------------------------ ------------------------------------------------  
AUTO\_TASK\_CONSUMER\_GROUP       System maintenance task consumer group  
BATCH\_CONSUMER\_GROUP           Batch process consumer group.  
DEFAULT\_CONSUMER\_GROUP         consumer group for users not assigned to any   
                               group  
LOW\_GROUP                      Group of low priority sessions  
OLTP\_CONSUMER\_GROUP            OLTP process consumer group.  
OTHER\_GROUPS                   consumer group for users not included in any  
                               group in the active top-plan  
SYS\_GROUP                      Group of system sessions

The *delete\_consumer\_groups.sql* script uses the*delete\_consumer\_group*procedure to clean up the consumer groups created for the example.  The consumer groups can only be removed if they have no dependant plan directives.

\* delete\_consumer\_groups.sql

BEGIN  
  DBMS\_RESOURCE\_MANAGER.clear\_pending\_area();  
  DBMS\_RESOURCE\_MANAGER.create\_pending\_area();

  -- Delete consumer groups.  
  DBMS\_RESOURCE\_MANAGER.delete\_consumer\_group (  
    consumer\_group => 'oltp\_consumer\_group');

  DBMS\_RESOURCE\_MANAGER.delete\_consumer\_group (  
    consumer\_group => 'batch\_consumer\_group');

  DBMS\_RESOURCE\_MANAGER.validate\_pending\_area;  
  DBMS\_RESOURCE\_MANAGER.submit\_pending\_area();  
END;  
/

With the consumer groups present, a resource plan can be created using the *create\_plan* procedure, and it can be associated to the consumer groups using the*create\_plan\_directive*procedure.

PROCEDURE create\_plan (  
  plan                      IN  VARCHAR2,  
  comment                   IN  VARCHAR2,  
  cpu\_mth                   IN  VARCHAR2 DEFAULT 'EMPHASIS',  
  active\_sess\_pool\_mth      IN  VARCHAR2 DEFAULT 'ACTIVE\_SESS\_POOL\_ABSOLUTE',  
  parallel\_degree\_limit\_mth IN  VARCHAR2 DEFAULT 'PARALLEL\_DEGREE\_LIMIT\_ABSOLUTE',  
  queueing\_mth              IN  VARCHAR2 DEFAULT 'FIFO\_TIMEOUT')

PROCEDURE create\_plan\_directive (  
  plan                      IN  VARCHAR2,  
   group\_or\_subplan          IN  VARCHAR2,  
  comment                   IN  VARCHAR2,  
  cpu\_p1                    IN  NUMBER DEFAULT NULL,  
  cpu\_p2                    IN  NUMBER DEFAULT NULL,  
  cpu\_p3                    IN  NUMBER DEFAULT NULL,  
  cpu\_p4                    IN  NUMBER DEFAULT NULL,  
  cpu\_p5                    IN  NUMBER DEFAULT NULL,  
  cpu\_p6                    IN  NUMBER DEFAULT NULL,  
  cpu\_p7                    IN  NUMBER DEFAULT NULL,  
  cpu\_p8                    IN  NUMBER DEFAULT NULL,  
  active\_sess\_pool\_p1       IN  NUMBER DEFAULT NULL,  
  queueing\_p1               IN  NUMBER DEFAULT NULL,  
  parallel\_degree\_limit\_p1  IN  NUMBER DEFAULT NULL,  
   switch\_group              IN  VARCHAR2 DEFAULT NULL,  
  switch\_time               IN  NUMBER DEFAULT NULL,  
  switch\_estimate           IN  BOOLEAN DEFAULT FALSE,  
  max\_est\_exec\_time         IN  NUMBER DEFAULT NULL,  
  undo\_pool                 IN  NUMBER DEFAULT NULL,  
  max\_idle\_time             IN  NUMBER DEFAULT NULL,  
  max\_idle\_blocker\_time     IN  NUMBER DEFAULT NULL,  
  switch\_time\_in\_call       IN  NUMBER DEFAULT NULL)

The day\_plan.sql script uses these procedures to create a resource plan suitable for daytime processing.  The OLTP operations are associated 80% of the CPU on level one; while batch operations receive 100% of the remaining CPU at level two.  The switch\_group and switch\_time parameters are used in the OLTP plan directive to specify that OLTP processes lasting more than 60 seconds should be switched to the batch consumer group.  The other\_groups consumer group must be included in any valid plan as it provides resource allocation information for any processes that are not explicitly associated with the consumer groups.

\* day\_plan.sql

BEGIN  
  DBMS\_RESOURCE\_MANAGER.clear\_pending\_area;  
  DBMS\_RESOURCE\_MANAGER.create\_pending\_area;

  -- Create a new plan  
  DBMS\_RESOURCE\_MANAGER.create\_plan(  
    plan    => 'day\_plan',  
    comment => 'Plan suitable for daytime processing.');

  -- Assign consumer groups to plan and define priorities  
  DBMS\_RESOURCE\_MANAGER.create\_plan\_directive (  
    plan             => 'day\_plan',  
    group\_or\_subplan => 'oltp\_consumer\_group',  
     comment          => 'Give OLTP processes higher priority - level 1',  
    cpu\_p1           => 80,  
    switch\_group     => 'batch\_consumer\_group',  
    switch\_time      => 60);

  DBMS\_RESOURCE\_MANAGER.create\_plan\_directive (  
    plan             => 'day\_plan',  
    group\_or\_subplan => 'batch\_consumer\_group',  
    comment          => 'Give batch processes lower priority - level 2',  
    cpu\_p2           => 100);

  DBMS\_RESOURCE\_MANAGER.create\_plan\_directive(  
    plan             => 'day\_plan',  
    group\_or\_subplan => 'OTHER\_GROUPS',  
    comment          => 'all other users - level 3',  
    cpu\_p3           => 100);

  DBMS\_RESOURCE\_MANAGER.validate\_pending\_area;  
  DBMS\_RESOURCE\_MANAGER.submit\_pending\_area;  
END;  
/

The night\_plan.sql script creates a resource plan suitable for nighttime processing in which the resource allocation is the reverse of the daytime processing, such that batch processes receive 80% of the CPU at level one, and OLTP operations receive 100% of the remaining CPU at level two.  Once again, the other\_groups consumer group is specified as a catch-all.

\* night\_plan.sql

BEGIN  
  DBMS\_RESOURCE\_MANAGER.clear\_pending\_area;  
  DBMS\_RESOURCE\_MANAGER.create\_pending\_area;

  -- Create a new plan  
  DBMS\_RESOURCE\_MANAGER.create\_plan(  
    plan    => 'night\_plan',  
    comment => 'Plan suitable for daytime processing.');

  -- Assign consumer groups to plan and define priorities  
  DBMS\_RESOURCE\_MANAGER.create\_plan\_directive (  
    plan             => 'night\_plan',  
    group\_or\_subplan => 'batch\_consumer\_group',  
    comment          => 'Give batch processes lower priority - level 2',  
    cpu\_p1           => 80);

  DBMS\_RESOURCE\_MANAGER.create\_plan\_directive (  
    plan             => 'night\_plan',  
    group\_or\_subplan => 'oltp\_consumer\_group',  
    comment          => 'Give OLTP processes higher priority - level 1',  
    cpu\_p2           => 100);

  DBMS\_RESOURCE\_MANAGER.create\_plan\_directive(  
    plan             => 'night\_plan',  
    group\_or\_subplan => 'OTHER\_GROUPS',  
    comment          => 'all other users - level 3',  
    cpu\_p3           => 100);

  DBMS\_RESOURCE\_MANAGER.validate\_pending\_area;  
  DBMS\_RESOURCE\_MANAGER.submit\_pending\_area;  
END;  
/

The resource\_plan\_directives.sql script uses the dba\_rsrc\_plan\_directives view to display information about the resource plans currently defined on the system.

\* resource\_plan\_directives.sql

select  
   plan,  
   group\_or\_subplan,  
   status  
from  
   dba\_rsrc\_plan\_directives  
order by  
   plan,  
   group\_or\_subplan  
;

The output from the resource\_plan\_directives.sql script is displayed below.

SQL> @resource\_plan\_directives.sql

PLAN                           GROUP\_OR\_SUBPLAN               STATUS  
------------------------------ ------------------------------ ------  
DAY\_PLAN                       BATCH\_CONSUMER\_GROUP           ACTIVE  
DAY\_PLAN                       OLTP\_CONSUMER\_GROUP            ACTIVE  
DAY\_PLAN                       OTHER\_GROUPS                   ACTIVE  
INTERNAL\_PLAN                  OTHER\_GROUPS                   ACTIVE  
INTERNAL\_QUIESCE               OTHER\_GROUPS                   ACTIVE  
INTERNAL\_QUIESCE               SYS\_GROUP                      ACTIVE  
NIGHT\_PLAN                     BATCH\_CONSUMER\_GROUP           ACTIVE  
NIGHT\_PLAN                     OLTP\_CONSUMER\_GROUP            ACTIVE  
NIGHT\_PLAN                     OTHER\_GROUPS                   ACTIVE  
SYSTEM\_PLAN                    LOW\_GROUP                      ACTIVE  
SYSTEM\_PLAN                    OTHER\_GROUPS                   ACTIVE  
SYSTEM\_PLAN                    SYS\_GROUP                      ACTIVE