A

Support for DBMS_JOB

Oracle continues to support the DBMS_JOB package. However, you must grant the CREATE JOB privilege to the database schemas that submit DBMS_JOB jobs.

Oracle Scheduler replaces the DBMS_JOB package. Although DBMS_JOB is still supported for backward compatibility, Oracle strongly recommends that you switch from DBMS_JOB to Oracle Scheduler.

In upgrades of Oracle Database 19c and later releases, if the upgrade can recreate existing DBMS_JOB jobs using DBMS_SCHEDULER, then for backward compatibility, after the upgrade, DBMS_JOB continues to act as a legacy interface to the DBMS_SCHEDULER job. If existing jobs cannot be recreated using DBMS_SCHEDULER because of issues with the metadata, then you receive a JOB_TABLE_INTEGRITY warning when you run upgrade prechecks. In that case, you have three options:

- Fix the metadata. After the upgrade continue to run after the upgrade using DBMS_JOBS as an interface, and run as DBMS_SCHEDULER jobs.
- Drop the jobs, if no longer required.
- Drop DBMS_JOBS jobs, and recreate the jobs manually using DBMS_SCHEDULER.

For existing jobs created with DBMS_JOB that are recreated during the upgrade, the legacy DBMS_JOB job is still present as an interface, but using it always creates a DBMS_SCHEDULER entry. Apart from the interface, the job is run as a DBMS_SCHEDULER job. If you subsequently disable the DBMS_JOB job created before the upgrade, then the DBMS_SCHEDULER job is also disabled. To avoid this behavior,drop the legacy job, and replace it with a DBMS_SCHEDULER job.

For all new jobs, use DBMS_SCHEDULER.

- Oracle Scheduler Replaces DBMS_JOB
  Starting with Oracle Database 11*g* Release 2 (11.2), Oracle Scheduler replaces DBMS_JOB. Oracle Scheduler is more powerful and flexible than DBMS_JOB, which is a package used to schedule jobs. Although DBMS_JOB is still supported for backward compatibility, Oracle strongly recommends that you switch from DBMS_JOB to Oracle Scheduler.
- Moving from DBMS_JOB to Oracle Scheduler
  This section illustrates some examples of how you can take jobs created with the DBMS_JOB package and rewrite them using Oracle Scheduler, which you configure and control with the DBMS_SCHEDULER package.

A.1 Oracle Scheduler Replaces DBMS_JOB

Starting with Oracle Database 11*g* Release 2 (11.2), Oracle Scheduler replaces DBMS_JOB. Oracle Scheduler is more powerful and flexible than DBMS_JOB, which is a package used to schedule jobs. Although DBMS_JOB is still supported for backward compatibility, Oracle strongly recommends that you switch from DBMS_JOB to Oracle Scheduler.

- Configuring DBMS_JOB
  The JOB_QUEUE_PROCESSES initialization parameter specifies the maximum number of processes that can be created for the execution of jobs.

- Using Both DBMS_JOB and Oracle Scheduler
  DBMS_JOB and Oracle Scheduler (the Scheduler) use the same job coordinator to start job child processes.

A.1.1 Configuring DBMS_JOB

The JOB_QUEUE_PROCESSES initialization parameter specifies the maximum number of processes that can be created for the execution of jobs.

Starting with Oracle Database Release 21c, the default value for JOB_QUEUE_PROCESSES across all containers is automatically derived from the number of sessions and CPUs configured in the system. The job coordinator process starts only as many job queue processes as are required, based on the number of jobs to run and available resources. You can set JOB_QUEUE_PROCESSES to a lower number to limit the number of job queue processes.

Setting JOB_QUEUE_PROCESSES to 0 disables DBMS_JOB jobs and DBMS_SCHEDULER jobs.

See Also:

Oracle Database Reference for more information about the JOB_QUEUE_PROCESSES initialization parameter

A.1.2 Using Both DBMS_JOB and Oracle Scheduler

DBMS_JOB and Oracle Scheduler (the Scheduler) use the same job coordinator to start job child processes.

You can use the JOB_QUEUE_PROCESSES initialization parameter to limit the number job child processes for both DBMS_JOB and the Scheduler.

If JOB_QUEUE_PROCESSES is 0, both DBMS_JOB and Oracle Scheduler jobs are disabled.

See Also:

- Scheduling Jobs with Oracle Scheduler
- "Setting Scheduler Preferences"
- *Oracle Database Reference* for more information about the JOB_QUEUE_PROCESSES initialization parameter

A.2 Moving from DBMS_JOB to Oracle Scheduler

This section illustrates some examples of how you can take jobs created with the DBMS_JOB package and rewrite them using Oracle Scheduler, which you configure and control with the DBMS_SCHEDULER package.

- Creating a Job
  An example illustrates creating a job using the DBMS_JOB package and the DBMS_SCHEDULER package.
- Altering a Job
  An example illustrates altering a job using the DBMS_JOB package and the DBMS_SCHEDULER package.
- Removing a Job from the Job Queue
  An example illustrates removing a job using the DBMS_JOB package and the DBMS_SCHEDULER package.

A.2.1 Creating a Job

An example illustrates creating a job using the DBMS_JOB package and the DBMS_SCHEDULER package.

The following example creates a job using DBMS_JOB:

```
VARIABLE jobno NUMBER;
BEGIN
 DBMS_JOB.SUBMIT(:jobno, 'INSERT INTO employees VALUES (7935, ''SALLY'',
   ''DOGAN'', ''sally.dogan@examplecorp.com'', NULL, SYSDATE, ''AD_PRES'', NULL,
    NULL, NULL, NULL);', SYSDATE, 'SYSDATE+1');
 COMMIT;
END;
/
```

The following is an equivalent statement using DBMS_SCHEDULER:

```
BEGIN
 DBMS_SCHEDULER.CREATE_JOB(
   job_name          =>  'job1',
   job_type          =>  'PLSQL_BLOCK',
   job_action        =>  'INSERT INTO employees VALUES (7935, ''SALLY'',
     ''DOGAN'', ''sally.dogan@examplecorp.com'', NULL, SYSDATE,''AD_PRES'', NULL,
      NULL, NULL, NULL);',
   start_date        =>  SYSDATE,
   repeat_interval   =>  'FREQ = DAILY; INTERVAL = 1');
END;
/
```

A.2.2 Altering a Job

An example illustrates altering a job using the DBMS_JOB package and the DBMS_SCHEDULER package.

The following example alters a job using DBMS_JOB:

```
BEGIN
 DBMS_JOB.WHAT(31, 'INSERT INTO employees VALUES (7935, ''TOM'', ''DOGAN'',
   ''tom.dogan@examplecorp.com'', NULL, SYSDATE,''AD_PRES'', NULL,
   NULL, NULL, NULL);');
 COMMIT;
END;
/
```

This changes the action for JOB1 to insert a different value.

The following is an equivalent statement using DBMS_SCHEDULER:

```
BEGIN
 DBMS_SCHEDULER.SET_ATTRIBUTE(
   name           => 'JOB1',
   attribute      => 'job_action',
   value          => 'INSERT INTO employees VALUES (7935, ''TOM'', ''DOGAN'',
      ''tom.dogan@examplecorp.com'', NULL, SYSDATE, ''AD_PRES'', NULL,
      NULL, NULL, NULL);');
END;
/
```

A.2.3 Removing a Job from the Job Queue

An example illustrates removing a job using the `DBMS_JOB` package and the `DBMS_SCHEDULER` package.

The following example removes a job using `DBMS_JOB`, where 14144 is the number of the job being run:

```
BEGIN
   DBMS_JOB.REMOVE(14144);
COMMIT;
END;
/
```

Using `DBMS_SCHEDULER`, you would issue the following statement instead:

```
BEGIN
   DBMS_SCHEDULER.DROP_JOB('myjob1');
END;
/
```

See Also:

- *Oracle Database PL/SQL Packages and Types Reference* for more information about the `DBMS_SCHEDULER` package
- Scheduling Jobs with Oracle Scheduler