

# SELECT<sup>®</sup> Online

The Journal of the International Oracle Users Group

Select Journal  
Home

Regular Columns

Past Issues

General Information

IOUG Home

**IOUG**  
Independent Oracle Users Group

## Select Magazine - April 2001

Volume 7, No. 3

### Managing Concurrent Requests in Oracle Applications

By Chitti Babu

*In Oracle Applications, managing concurrent requests based upon user requirements can be a big task. Some users may submit a group of concurrent requests or programs sequentially or in periodic intervals. Oracle Applications has the capability of handling this type of requirement for submitting a group of requests in a "concurrent request set."*

A concurrent request set consists of two or more concurrent requests whose parameters can be shared. Individual programs within the concurrent request set can be submitted either sequentially or in parallel.

Processes or programs that kick off child programs may be designed to allow the parent program to wait for any child programs to complete. Creating a concurrent request set using the sequential option will execute each member of the concurrent request set, one after another. One major problem with a parent-child request occurs when the parent program exits without waiting for the child program to complete. In such cases, defining a concurrent request set may not solve the problem. The programs in the concurrent request set will be executed based upon the sequence without waiting for the completion of the preceding member, which may produce erroneous results. An example of when this type of problem occurs can be found in the planning processes within Oracle Applications. When MPS or MRP planning processes are launched, the request kicks off too many child processes and the parent completes without even waiting for all child processes to complete. If these processes are included in a request set and that request set is scheduled, the end result cannot be anticipated.

In general, planning processes are often scheduled every weekend. The Oracle Applications administrator (or other person in an equivalent role) has to create a request set incorporating all of the member processes required for the planning process. Due to the problem described above, a concurrent request set may not work. A custom program may need to be introduced in between each member process of the request set.

This article will use the planning processes available in the Oracle Applications as an example to describe one method of solving this problem. Design Strategy

All planning processes need to run as a request set. This cannot be achieved by defining the standard request set functionality of Oracle Applications because the MPS/MRP Planning Processes submit various independent concurrent requests that may complete without waiting for the completion of others.

As a solution, the administrator can generate an SQL\*Plus program to monitor the successful completion of the Memory-Based Planner for MPS Processes. Upon successful completion of the Memory Based Planner, the program will wait for an additional six minutes before starting the MRP Planning Process requests. This additional time should allow for all of the child processes to complete. The SQL\*Plus program is basically designed for monitoring successful completion of MPS Planning Processes and needs to be inserted between the MPS Planning Processes in the concurrent request set and the MRP Planning Processes.

## Name &amp; Logic

```

=====
-- FILENAME
--
-- ACMRPCHK.sql
--
-- DESCRIPTION
--
-- SQL script to increase delay between
--
-- MPS and MRP Planning Processes
--
-- ARGUMENTS
--
-- None
--
-- NOTES
--
--
-- Nov 26, 1997    chitti, Kalyan Iyer    Creation
--
--
--
-- =====

```

## Declare

```

v_req_id          number;
v_oth_req_id      number;
v_phase           varchar2(30);
v_status          varchar2(30);
v_dev_phase       varchar2(100);
v_dev_status      varchar2(100);
v_message         varchar2(240);
v_return          boolean;

```

## Begin

```

Select  NVL(max(request_id),0)
Into    v_req_id
From    fnd_concurrent_requests fcr,

```

```

                fnd_concurrent_programs fcp
Where   fcp.concurrent_program_name   = 'MRPSLPPR1'
And     fcr.concurrent_program_id     =
fcp.concurrent_program_id;

Select  NVL(max(request_id),0)
Into    v_oth_req_id
From    fnd_concurrent_requests fcr,
        fnd_concurrent_programs fcp
Where   fcp.concurrent_program_name   = 'MRCNEW'
And     fcr.concurrent_program_id     =
fcp.concurrent_program_id;

Loop
If      v_oth_req_id <= v_req_id
Then
        Select  NVL(max(request_id),0)
        Into    v_oth_req_id
        From    fnd_concurrent_requests fcr,
                fnd_concurrent_programs fcp
        Where   fcp.concurrent_program_name   =
'MRCNEW'
                And     fcr.concurrent_program_id     =
fcp.concurrent_program_id;
        Else
                Exit;
        End if;
End loop;

v_return :=
Fnd_concurrent.wait_for_request(v_oth_req_id,60,3600,v_phase,v_status,v_
dev_phase,v_dev_status,v_
message);

```

```

        If (v_dev_phase = 'COMPLETE' and v_dev_status = 'NORMAL')

        Then

                v_return := fnd_concurrent.set_completion_status('NORMAL','');

        Else

                v_return := fnd_concurrent.set_completion_status
('NORMAL','Program Failed Because Of
-'||v_message);

                End if;

        End;

/

```

The delay of 3600 in the Fnd\_concurrent.wait\_for\_request command may need to be increased based upon your specific hardware and the configuration of the ERP system in use at your site. The script shown above must be registered as a concurrent program, and this program has to be inserted between the MPS planning processes and the MRP planning processes in a request set. This request set can be scheduled to run in intervals based upon business needs. The advantage of this script is to schedule planning processes with proper parameters set up front for execution. Example of Program in the Concurrent Request Set

The above program is registered as "MRP Check Process" under custom application top as SQL Executable with Parameters None.

**The Planning Request would consist of :**

1. Launch MPS Planning Processes
2. MRP Check Process (Custom)
3. Launch MRP Planning Processes

The above set can be scheduled at periodic intervals based on user needs.

**About the Author**

Chitti Babu is vice president of NCC Softech, India, a software development company in ERP, e-Commerce and e-Business, and GIS.

[Download Acrobat Reader](#)

Copyright 2003 by the International Oracle  
Users Group