

O2 Works

Putting Oracle to Work

Oracle E-Business Specialists

Oracle Package - FND_CONC_GLOBAL

Internal Techies Presentation

December 4, 2018



Abstract

The Oracle package FND_CONC_GLOBAL allows a PLSQL program to be setup as a Master concurrent program with a parameter for the number of workers that will work like a request set. This allows a single concurrent program to be submitted with one parameter change that can submit one or more workers. The number of workers can be tuned to the available concurrent managers to easily change the configuration and get the program to finish in the minimum timeframe. This is useful when a program like a data warehouse extract program needs to finish in a defined period of time.

Background / Issue

- Client has a Data Warehouse program to extract product information From Oracle EBS. The program needed to wait for batch jobs to finish on Oracle and the close of business on the West coast. Long running programs need be run after this extract to prepare the Data Warehouse for users the next business morning.
- The client acquired another business which resulted in the Product count going from 50K to over 500K.
- The extract program was setup to run multiple instances concurrently using a request set with item id ranges set in program parameters.
- Reworking and rescheduling the request set to add more processes each time another group of Products was added to the system was proving time consuming. Often the program was running long causing downstream delays in the other processes.

Solution

- Program was tuned. A number of function call were reworked into in-line views in the SQL statement.
- The request set was eliminated. The program was recoded to be run from a single concurrent program with one parameter to specify the number of sub-programs to submit.

Oracle Package - FND_CONC_GLOBAL

- **FND_CONC_GLOBAL.SET_REQ_GLOBALS** – sets the value of the REQUEST_DATA global variable.
 - ✓ conc_status – Status of concurrent program like 'PAUSED', 'RUNNING'
 - ✓ request_data – Value of the global variable
 - ✓ conc_restart_time – Concurrent restart time
 - ✓ release_sub_request – Release subrequest
- **FND_CONC_GLOBAL.REQUEST_DATA** – retrieves the value of the REQUEST_DATA global

Master concurrent program with parameter for the number of workers

The number of worker programs can be adjusted from the single concurrent program.

The screenshot shows the Oracle Navigator 'Submit Request' dialog box. The main dialog has the following fields: Name (INV DW - OnHand/OnOrder Rebuild Submit), Operating Unit, Parameters (5:No), and Language (American English). A 'Parameters' sub-dialog box is open, showing a 'Workers' field with the value '20' highlighted by a red box, and a 'Debug?' field with the value 'No'. The sub-dialog also has 'OK', 'Cancel', 'Clear', and 'Help' buttons. The main dialog has a 'Copy' button and a 'Schedule' button. The 'Upon Completion' section has a checked 'Save all Output Files' checkbox and an unchecked 'Burst Output' checkbox. The 'Layout' and 'Notify' fields are also visible.

Submitting the program

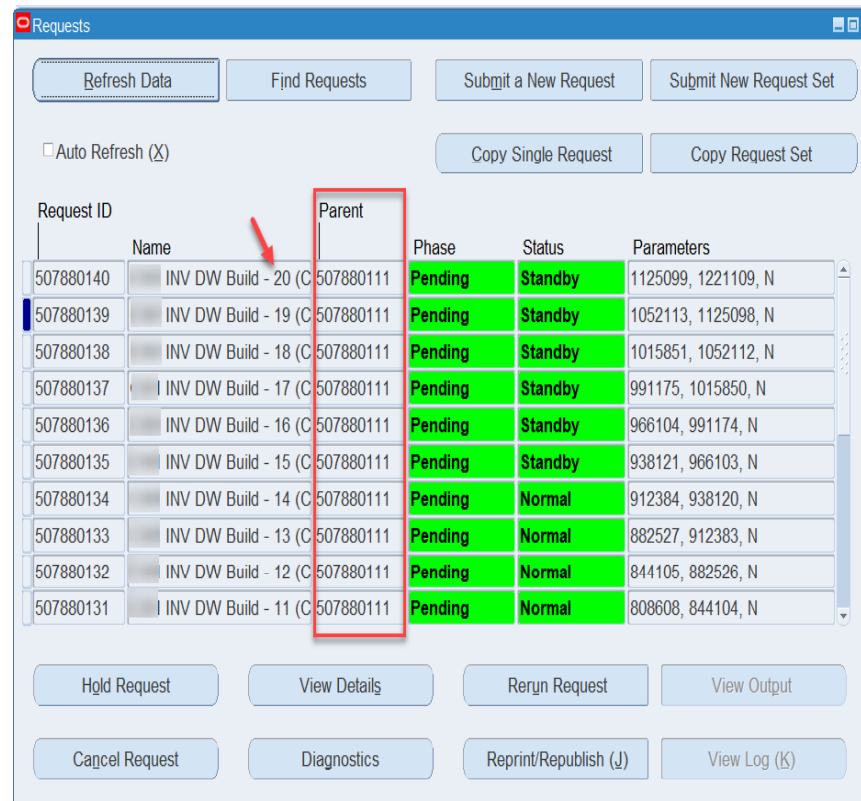
The screenshot shows a window titled 'Requests' with several buttons and a table. The buttons are: Refresh Data, Find Requests, Submit a New Request, Submit New Request Set, Copy Single Request, and Copy Request Set. There is also a checkbox for 'Auto Refresh (X)'. The table below has columns for Request ID, Name, Parent, Phase, Status, and Parameters. One row is visible with the following data:

Request ID	Name	Parent	Phase	Status	Parameters
507880111	INV DW - OnHand/Or		Running	Normal	20, N

The program then submits the worker concurrent programs

The 'Parent' concurrent program request id is set just as it would be with a request set.

The worker number is included on the Program Name for each worker.



Requests

Refresh Data Find Requests Submit a New Request Submit New Request Set

Auto Refresh (X) Copy Single Request Copy Request Set

Request ID	Name	Parent	Phase	Status	Parameters
507880140	INV DW Build - 20 (C	507880111	Pending	Standby	1125099, 1221109, N
507880139	INV DW Build - 19 (C	507880111	Pending	Standby	1052113, 1125098, N
507880138	INV DW Build - 18 (C	507880111	Pending	Standby	1015851, 1052112, N
507880137	INV DW Build - 17 (C	507880111	Pending	Standby	991175, 1015850, N
507880136	INV DW Build - 16 (C	507880111	Pending	Standby	966104, 991174, N
507880135	INV DW Build - 15 (C	507880111	Pending	Standby	938121, 966103, N
507880134	INV DW Build - 14 (C	507880111	Pending	Normal	912384, 938120, N
507880133	INV DW Build - 13 (C	507880111	Pending	Normal	882527, 912383, N
507880132	INV DW Build - 12 (C	507880111	Pending	Normal	844105, 882526, N
507880131	INV DW Build - 11 (C	507880111	Pending	Normal	808608, 844104, N

Hold Request View Details Rerun Request View Output

Cancel Request Diagnostics Reprint/Republish (J) View Log (K)

The worker programs can be found using Find Requests the same as if it was a request set.

Requests

Refresh Data Find Requests Submit a New Request Submit New Request Set

Auto Ref My Completed Requests My Requests In Progress All My Requests Specific Requests

Request ID

507880129
507880128
507880127
507880126
507880125
507880124
507880123
507880122
507880121
507880111

Request ID 507880111

Name

Date Submitted

Date Completed

Status

Phase

Requestor

Include Request Set Stages in Query

Order By Request ID

Select the Number of Days to View: 7

Submit a New Request Submit New Request Set Clear Find

Search results

Requests

Auto Refresh (X)

Request ID	Name	Parent	Phase	Status	Parameters
507880129	INV DW Build - 9 (CV	507880111	Pending	Normal	734459, 767272, N
507880128	INV DW Build - 8 (CV	507880111	Pending	Normal	694349, 734458, N
507880127	INV DW Build - 7 (CV	507880111	Pending	Normal	663350, 694348, N
507880126	INV DW Build - 6 (CV	507880111	Pending	Normal	622770, 663349, N
507880125	INV DW Build - 5 (CV	507880111	Running	Normal	596053, 622769, N
507880124	INV DW Build - 4 (CV	507880111	Running	Normal	567654, 596052, N
507880123	INV DW Build - 3 (CV	507880111	Running	Normal	537734, 567653, N
507880122	INV DW Build - 2 (CV	507880111	Running	Normal	105206, 537733, N
507880121	INV DW Build - 1 (CV	507880111	Running	Normal	1009, 105205, N
507880111	INV DW - OnHand/Or		Running	Paused	20, N

Master concurrent program shows as 'paused' while the workers are running.

507880122	INV DW Build - 2 (CV	507880111	Running	Normal	105206, 537733, N
507880121	INV DW Build - 1 (CV	507880111	Running	Normal	1009, 105205, N
507880111	INV DW - OnHand/Or		Running	Paused	20, N

Hold Request	View Details	Rerun Request	View Output
Cancel Request	Diagnostics	Reprint/Republish (J)	View Log (K)

Canceling the Master concurrent program will cancel all of the workers.

507880124	INV DW Build - 4 (CV	507880111	Running	Normal	550000, 022700, N
507880123	INV DW Build - 3 (CV	507880111	Running	Normal	
507880122	INV DW Build - 2 (CV	507880111	Running	Normal	
507880121	INV DW Build - 1 (CV	507880111	Running	Normal	
507880111	INV DW - OnHand/Or		Running	Normal	

Decision

?

Canceling a request cannot be undone. Continue?

Yes No

Hold Request View Details Rerun Request View Output

Cancel Request Diagnostics Reprint/Republish (J) View Log (K)

507880123	INV DW Build - 3 (CV	507880111	Running	Normal	537734, 567653, N
507880122	INV DW Build - 2 (CV	507880111	Running	Normal	105206, 537733, N
507880121	INV DW Build - 1 (CV	507880111	Running	Normal	1009, 105205, N
507880111	INV DW - OnHand/Or		Completed	Terminated	20, N

Hold Request View Details Rerun Request View Output

Cancel Request Diagnostics Reprint/Republish (J) View Log (K)

Canceling the Master concurrent program will cancel all of the workers.

Refresh Data Find Requests Submit a New Request Submit New Request Set

Auto Refresh (X) Copy Single Request Copy Request Set

Request ID	Name	Parent	Phase	Status	Parameters
507880129	INV DW Build - 9 (CV	507880111	Completed	Cancelled	734459, 767272, N
507880128	INV DW Build - 8 (CV	507880111	Completed	Cancelled	694349, 734458, N
507880127	INV DW Build - 7 (CV	507880111	Completed	Cancelled	663350, 694348, N
507880126	INV DW Build - 6 (CV	507880111	Completed	Cancelled	622770, 663349, N
507880125	INV DW Build - 5 (CV	507880111	Completed	Terminated	596053, 622769, N
507880124	INV DW Build - 4 (CV	507880111	Completed	Terminated	567654, 596052, N
507880123	INV DW Build - 3 (CV	507880111	Completed	Terminated	537734, 567653, N
507880122	INV DW Build - 2 (CV	507880111	Completed	Terminated	105206, 537733, N
507880121	INV DW Build - 1 (CV	507880111	Completed	Terminated	1009, 105205, N
507880111	INV DW - OnHand/Or		Completed	Terminated	20, N

Hold Request View Details Rerun Request View Output

Cancel Request Diagnostics Reprint/Republish (J) View Log (K)

Gives the control available using a request set with easier reconfiguration that allows the process to be tuned for data volume changes and availability of concurrent managers.

Code that allows this to function like a request set.

- Package: `fnd_conc_global`
- **FND_CONC_GLOBAL.SET_REQ_GLOBALS** – sets the value of the `REQUEST_DATA` global variable.
 - `conc_status` – Status of concurrent program like ‘PAUSED’, ‘RUNNING’
 - `request_data` – Value of the global variable
 - `conc_restart_time` – Concurrent restart time
 - `release_sub_request` – Release subrequest
- **FND_CONC_GLOBAL.REQUEST_DATA** – retrieves the value of the `REQUEST_DATA` global

--Get value of global variable. It is null initially.

```
l_req_data := fnd_conc_global.request_data;
```

-- If equal to 'END', exit the program with return code '0'.

```
IF l_req_data ='END' THEN
```

```
    RETURN;
```

```
END IF;
```

-- Set parent program status as 'PAUSED' and set global variable value to 'END'

```
fnd_conc_global.set_req_globals(conc_status => 'PAUSED'  
                                , request_data => 'END');
```

The global is checked before the workers are submitted. At the start the global will be set to null.

Program needs to set the global to tell what stage it is at. It can be set to any value, the program just needs to be coded to respond to whatever the value is.

Create a function to Check Completion Status of workers to determine the Return Status for the parent.

```
-----  
-- Function: check_completion_status  
-- Purpose: checks the completion status of the children.  
-----  
FUNCTION check_completion_status  
RETURN NUMBER IS  
  
    module_name VARCHAR2(30) := 'CHECK_COMPLETION_STATUS';  
    l_record_count NUMBER := 0;  
  
BEGIN  
  
    fnd_file.put_line(fnd_file.log, 'STARTING: '||module_name);  
  
    SELECT COUNT(*)  
    INTO l_record_count  
    FROM FND_CONC_REQ_SUMMARY_V csv  
    WHERE 1=1  
    and status_code <> 'C'  
    and parent_request_id = FND_GLOBAL.CONC_REQUEST_ID;  
    -- If no errors return 0  
    IF (l_record_count = 0) THEN  
        fnd_file.put_line(fnd_file.log, 'Completion Status : 0.');        RETURN 0;  
    ELSE  
        -- Look to see if any children errored  
        SELECT COUNT(*)  
        INTO l_record_count  
        FROM FND_CONC_REQ_SUMMARY_V csv  
        WHERE 1=1  
        and status_code = 'E'  
        and parent_request_id = FND_GLOBAL.CONC_REQUEST_ID;  
  
        IF (l_record_count >= 1) THEN  
            -- If error found then return 2  
            fnd_file.put_line(fnd_file.log, 'Completion Status : 2.');            RETURN 2;  
        ELSE  
            -- Otherwise return warning  
            fnd_file.put_line(fnd_file.log, 'Completion Status : 1.');            RETURN 1;  
        END IF;  
    END IF;  
  
    fnd_file.put_line(fnd_file.log, 'END: '||module_name);  
  
EXCEPTION  
    WHEN OTHERS THEN  
        RETURN 2;  
        fnd_file.put_line(fnd_file.log, module_name||' WHEN OTHERS Exception Raised in  
check_completion_status '||SUBSTR(SQLERRM, 1,100));  
        RAISE;  
  
END check_completion_status;
```


After the last worker is submitted set 'set_req_globals conc status to PAUSED and request_data to 'END''

- -----
- -- Call submit_request
- -----
- ln_RequestId := submit_request(p_application => 'XXCWINV',
- p_prog_short_name => 'CWIINV_TEST_EDW_INV_BUILD',
- p_description => 'CWI INV EDW Build - '||ln_worker_count,
- p_sub_request => TRUE,
- p_argument1 => ln_low_item_id,
- p_argument2 => ln_high_item_id,
- p_argument3 => p_debug);
- END IF;
- -- Set status as 'PAUSED' and set global variable value to 'END'
- fnd_file.put_line(fnd_file.log,'Setting parent request to PAUSED');
- fnd_conc_global.set_req_globals(conc_status => 'PAUSED',request_data => 'END');

The parent program has code at the start of the program to check the conc_status and request_data

```
-- get request_data.
l_req_data := fnd_conc_global.request_data;
fnd_file.put_line(fnd_file.log,'Starting to submit workers. '||l_req_data);

IF (l_req_data = 'END') THEN
  fnd_file.put_line(fnd_file.log,'Request subprograms have completed. '||l_req_data);
  -- set parent return status based on children return statuses.
  retcode := check_completion_status;
  RETURN;
ELSIF (l_req_data = '1') THEN
  IF ((check_completion_status) = 0) THEN

    fnd_file.put_line(fnd_file.log,'Going to submit_notification. '||l_req_data);
    submit_notification;
    fnd_conc_global.set_req_globals(conc_status => 'PAUSED',request_data => 'END');
  -- set parent return status based on children return statuses.
  retcode := check_completion_status;

  ELSE
    retcode := check_completion_status;
    RETURN;
  END IF;
ELSIF (l_req_data is NULL) THEN
  -- clear the temporary table and continue with the body of this package.
```

Conclusion

With FND_CONC_GLOBAL and a dedicated Concurrent Manager - large volume imports or exports can be easily tunable for the number of processes available in the Concurrent Manager

About the Author

Glenn Christenson has over 25 years of experience as an IT consultant for Oracle and Oracle applications serving in both functional and technical roles.

- He has designed and developed numerous interfaces into and out of Oracle applications including SOA web services using SOAP messaging and JAVA.
- Glenn has extensive understanding of Oracle Trading Community Architecture and has utilized Oracle provided API's to leverage conversions and external data to enrich TCA information.
- Recently designed and developed AME for an iExpense implementation incorporating PCard entry and approvals.

Contact information

O2Works – Knowledge Base:

<https://o2works.com/knowledge-works/>

Name: Glenn Christenson

Position: Principal Consultant -Technical Developer

E-mail: glennc@o2works.com

O2Works LLC

Oracle Applications Consulting

Dallas – Chicago - Denver

972-466-2260

<https://o2works.com>



INTERNAL DOCUMENT
©Copyright 2019 O2Works
All Rights Reserved