

Creating a concurrent processing expressway in Oracle EBS 11i

The Oracle E-Business Suite (EBS) is a complex Enterprise Resource Planning (ERP) program that consists of many components. The primary components that comprise the E-Business Suite are the Client, Forms Server, Web Server, Concurrent Processor, and Oracle Database.

User interactions with Oracle Applications data are conducted via HTML-based Applications or the more traditional Forms-based Applications. However, there are also reporting programs and data updating programs that need to run either periodically, or on an ad hoc basis. These programs may contain a very large number of data-intensive computations, and are run using the *concurrent processing* architecture. To ensure that resource-intensive concurrent processing operations do not interfere with interactive operations, they are run on a specialized server, the *Concurrent Processing server*. Processes that run on the concurrent processing server are called *concurrent requests*. When you submit such a request a row is inserted into a database table specifying the program to be run. A *concurrent manager* then reads the applicable requests in the table, and starts the associated concurrent program. The *Internal Concurrent Manager* controls the other concurrent managers, and determines when a request should be processed and which concurrent manager should carry it out.

The majority of *concurrent requests*, unless otherwise specified, run under the *Standard Manager* concurrent manager. Poorly tuned or long-running programs have the ability to cause bottlenecks that affect all other concurrent programs running under this concurrent manager. To quickly reduce the possibility of these bottlenecks, it may be necessary to migrate the faster programs to their own concurrent manager creating an “expressway” that allows them to proceed separate of the normal traffic in the *Standard Manager* concurrent manager.

First, these programs must be identified based upon a pre-agreed execution time threshold. Use the following query to get the average runtimes of all concurrent programs that have been executed in the past week for the *Standard Manager* concurrent manager (executed as the APPS user):

```
select /*+ ALL_ROWS */
  --fcptl.user_concurrent_queue_name "Concurrent Manager",
    fatl.APPLICATION_NAME "Application",
  fcptl.user_concurrent_program_name "Program",
  --round(sum(greatest(actual_completion_date - actual_start_date,0))*60*24, 2) "Total Duration (min)",
  round(avg(greatest(actual_completion_date - actual_start_date,0))*60*24, 2) "Avg Duration (min)",
  round(min(greatest(actual_completion_date - actual_start_date,0))*60*24, 2) "Min Duration (min)",
  round(max(greatest(actual_completion_date - actual_start_date,0))*60*24, 2) "Max Duration (min)",
  round(avg(greatest(actual_completion_date - crm_release_date,0))*60*24,2) "Avg CRM Wait (min)",
  count(*) "Times Run"
from
  fnd_application_tl fatl,
  fnd_concurrent_programs fcp,
  fnd_concurrent_programs_tl fcptl,
  fnd_concurrent_processes fcproc,
  fnd_concurrent_queues_tl fcqtl,
  fnd_concurrent_queues fcq,
  fnd_concurrent_requests fcr
```

```

where fcr.phase_code = 'C'
  and fcqtl.USER_CONCURRENT_QUEUE_NAME like 'Standard Manager%'
  and fcr.actual_completion_date is not null
  and actual_start_date is not null
  and fcq.concurrent_queue_id = fcproc.concurrent_queue_id
  and fcq.application_id = fcproc.queue_application_id
  and fcq.manager_type = 1
  and fcr.controlling_manager = fcproc.concurrent_process_id
  and fcr.program_application_id = fcp.application_id
  and fcr.concurrent_program_id = fcp.concurrent_program_id
  and fcp.concurrent_program_name not in ('ACTIVATE','ABORT','DEACTIVATE','VERIFY')
  and fcr.concurrent_program_id = fcptl.concurrent_program_id
  and fcr.program_application_id = fcptl.application_id
    and fcr.program_application_id = fatl.application_id
  and fcptl.language=userenv('LANG')
    and fatl.language=userenv('LANG')
  and fcproc.queue_application_id = fcqtl.application_id
  and fcproc.concurrent_queue_id = fcqtl.concurrent_queue_id
  and fcqtl.language=userenv('LANG')
  and actual_start_date >= trunc(sysdate - 6) and actual_start_date <= trunc(sysdate+1)
group by
  fcptl.user_concurrent_program_name,
  fatl.application_name
order by
  "Avg Duration (min)" desc--

```

Sample Output:

Application	Program	Avg Duration (min)	Min Duration (min)	Max Duration (min)	Avg CRM Wait (min)	Times Run
XX Custom	FLS ETL Update Requisitions	1.4	1.4	1.4	1.4	1
Projects	ADM: Rebuild Project Search	0.08	0.04	0.03	0.05	2
Bills of Material	Material cost transaction worker	0.03	0.03	0.03	0.18	1
Bills of Material	Actual Cost Worker	0.12	0.02	0	0.05	6

After the fast-running programs have been identified, it is a good idea to send this information to the business owners informing them that these programs have been labeled as fast-running and will be relegated to a separate concurrent manager to run without interference from slower programs. This will serve the purpose of informing them of your intentions and enabling others to investigate possible performance problems that they may not have been aware of if their processes were not identified as fast-running.

Next, determine the names and the amount of processes already in use for the existing concurrent managers with the following query (executed as the APPS user):

```

select Application_Id Apld, concurrent_queue_id Q_Id,
  User_Concurrent_Queue_Name Manager, Target_Node Node,
  Running_Processes Running,
  Max_Processes Max, Cache_Size Buf, Diagnostic_Level D
from fnd_concurrent_queues_vl
where running_processes > 0
order by manager

```

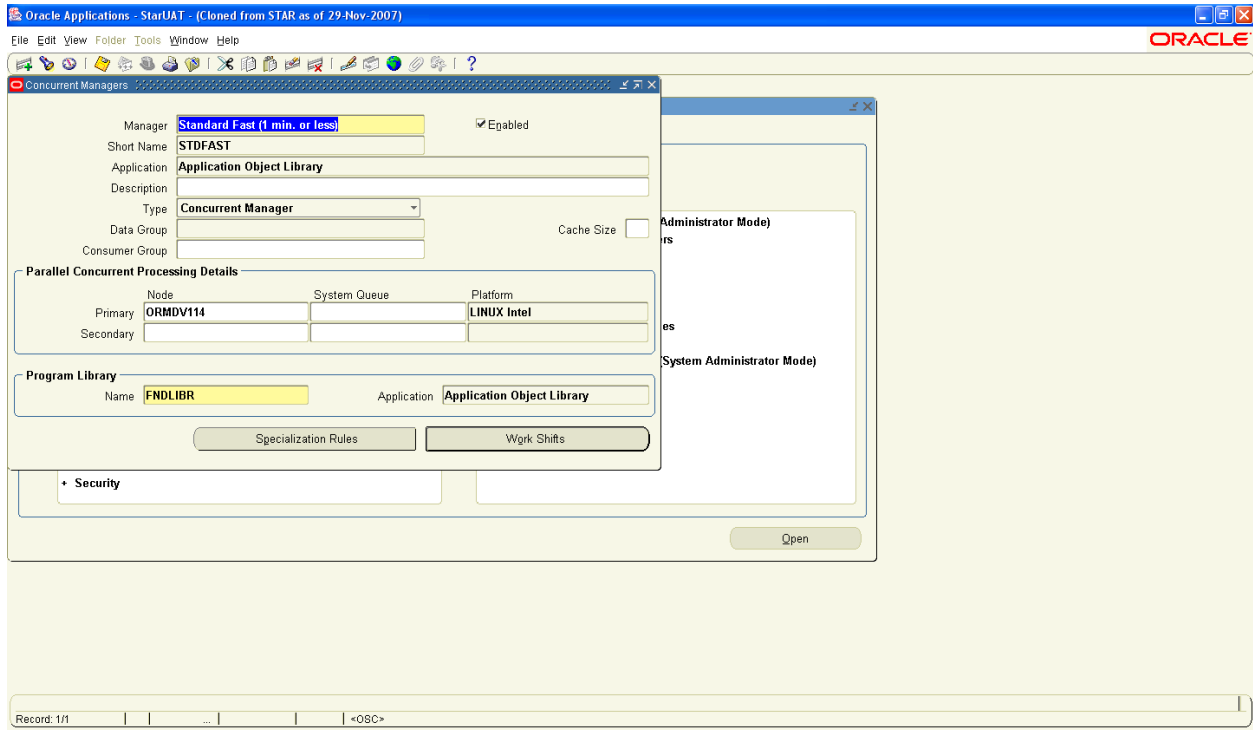
Sample Output:

APID	Q_ID	MANAGER	NODE	RUNNING	MAX	BUF	D
705	12	CRP Inquiry Manager	ORMDV114	1	1		N
0	4	Conflict Resolution Manager	ORMDV114	1	1		N
401	11	INV Remote Procedure Manager	ORMDV114	2	2		N
401	10	Inventory Manager	ORMDV114	1	1	5	N
0	29339	Long Running	ORMDV114	3	3	2	N
704	39	MRP Manager	ORMDV114	1	1	5	N
704	24339	MRP Standard Manager	ORMDV114	22	22	10	N
0	1063	OAM Metrics Collection Manager	ORMDV114	1	1	1	N
0	5288	Output Post Processor	ORMDV114	2	2		4
275	40	PA Streamline Manager	ORMDV114	1	1	5	N
201	15	PO Document Approval Manager	ORMDV114	2	2		N
201	222	Receiving Transaction Manager	ORMDV114	1	1		N
0	5	Scheduler/Prereleaser Manager	ORMDV114	1	1		N
0	28341	Service Manager: ORMDV114	ORMDV114	1	1		
696	1024	Session History Cleanup	ORMDV114	1	1	0	N
0	0	Standard Manager	ORMDV114	10	10	5	N
696	1025	UWQ Worklist Items Release	ORMDV114	1	1	0	N
0	1139	Workflow Agent Listener Service	ORMDV114	1	1		
0	1259	Workflow Document Web Services Service	ORMDV114	1	1		
0	1140	Workflow Mailer Service	ORMDV114	1	1		

In this example, the *Standard Manager* concurrent manager has 10 processes allocated to it. We could subtract the amount of processes we want for the fast-running concurrent manager from these processes, but for the sake of expedience, we will say that we will allocate 2 new processes to our new concurrent manager. Unless we are resource-constrained, we don't want to further penalize the "slower" programs left behind in the *Standard Manager* concurrent manager.

To create the fast-running concurrent manager, follow these steps:

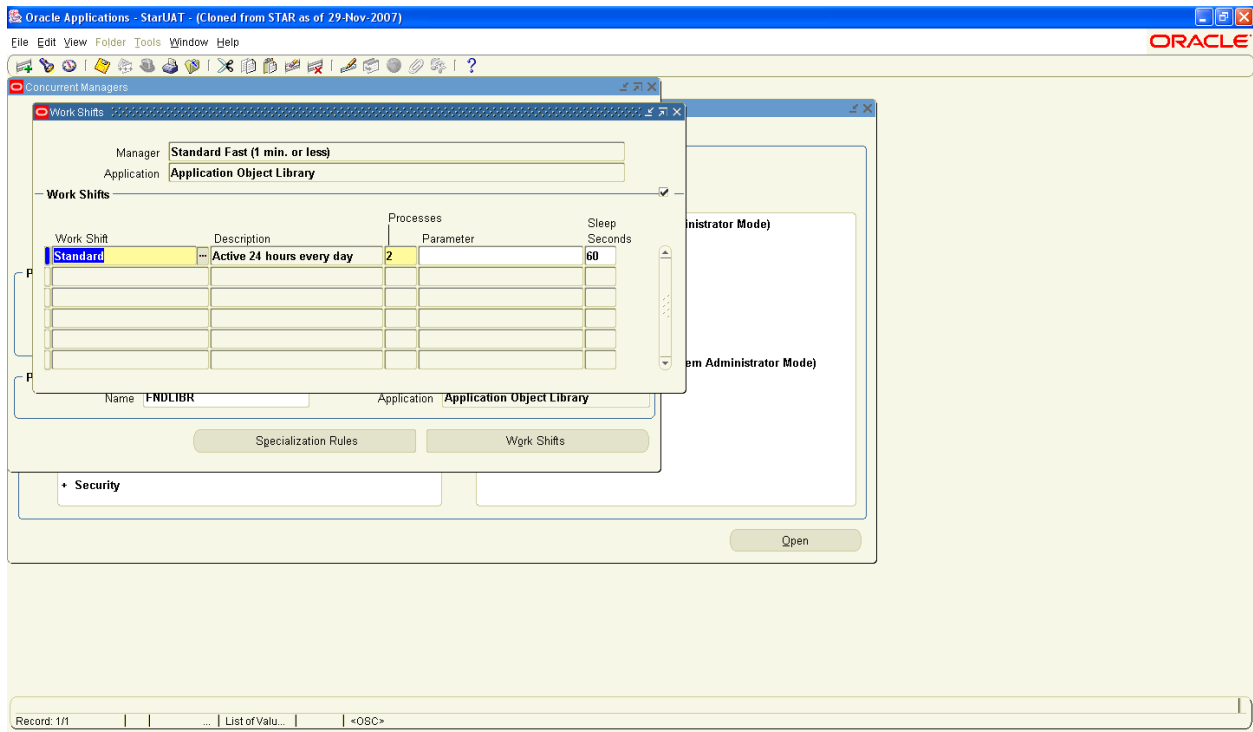
1. Log into the Oracle Applications program as a user whom has the System Administrator responsibility assigned to them.
2. Select the Define link under the Concurrent:Manager section
3. Use the screenshot to fill in the yellow areas:



These areas are:

- Manager – The name you are going to call the new concurrent manager. Be descriptive
- Enabled – Make sure there is a check in this check box to enable the new manager
- Short Name – An abbreviated concurrent manager name (i.e. STDFAST)
- Application – Use the standard value “Application Object Library”
- Type – Use the value “Concurrent Manager”
- Node – Make sure the node where this concurrent manager will run is selected. For a clustered environment, a searchable list of values will be available.
- Name (under Program Library) - Use the value “FNDLIBR”

4. Once these values are filled in, click the Work Shifts button on the bottom right of the screen.



Select a Standard work shift for this new concurrent manager so that it will run constantly. For the number of processes, use the previously mentioned value of 2.

5. Once these values are filled in, save the records by pulling down the file menu at the top left of the forms window and selecting Save.
6. Now that your new concurrent manager has been created, it needs to be started. Use <F4> to get back to the main Navigator screen and then go to Concurrent -> Manager -> Administer. Scroll down the list of concurrent managers until you see the new concurrent manager you created listed.

Oracle Applications - StarJAT - (Cloned from STAR as of 29-Nov-2007)

File Edit View Folder Tools Window Help

ORACLE

Administer Concurrent Managers

Name	Node	Processes		Requests		Status
		Actual	Target	Running	Pending	
Service Manager: ORMDV110	ORMDV110	1	0			
Service Manager: ORMDV112	ORMDV112	1	0			Deactivated
Service Manager: ORMDV113	ORMDV113	1	0			
Service Manager: ORMDV163	ORMDV163	1	0			
Collections Manager	ORMDV114	0	0	0	0	
OAM Generic Collection Si		0	0			Deactivated
OAM Generic Collection Si		0	0			Deactivated
OAM Generic Collection Si		0	0			Deactivated
OAM Generic Collection Si		0	0			Deactivated
Contracts Core Concurrent	ORMDV114	0	0	0	0	
Standard Fast (1 min. or le	ORMDV114	0	0	0	0	Deactivated
Oracle Provisioning Manag	ORMDV114	0	0	0	0	

Service Info

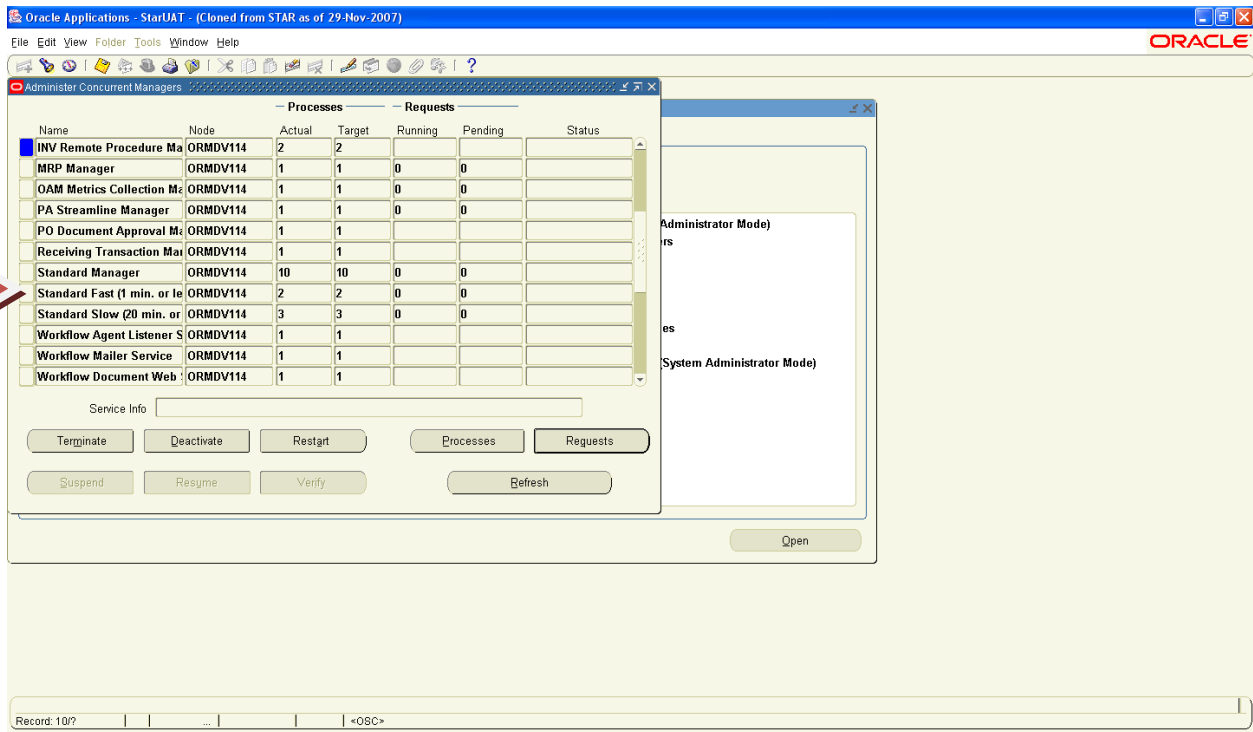
Terminate Deactivate Restart Processes Requests

Suspend Resume Verify Refresh

Open

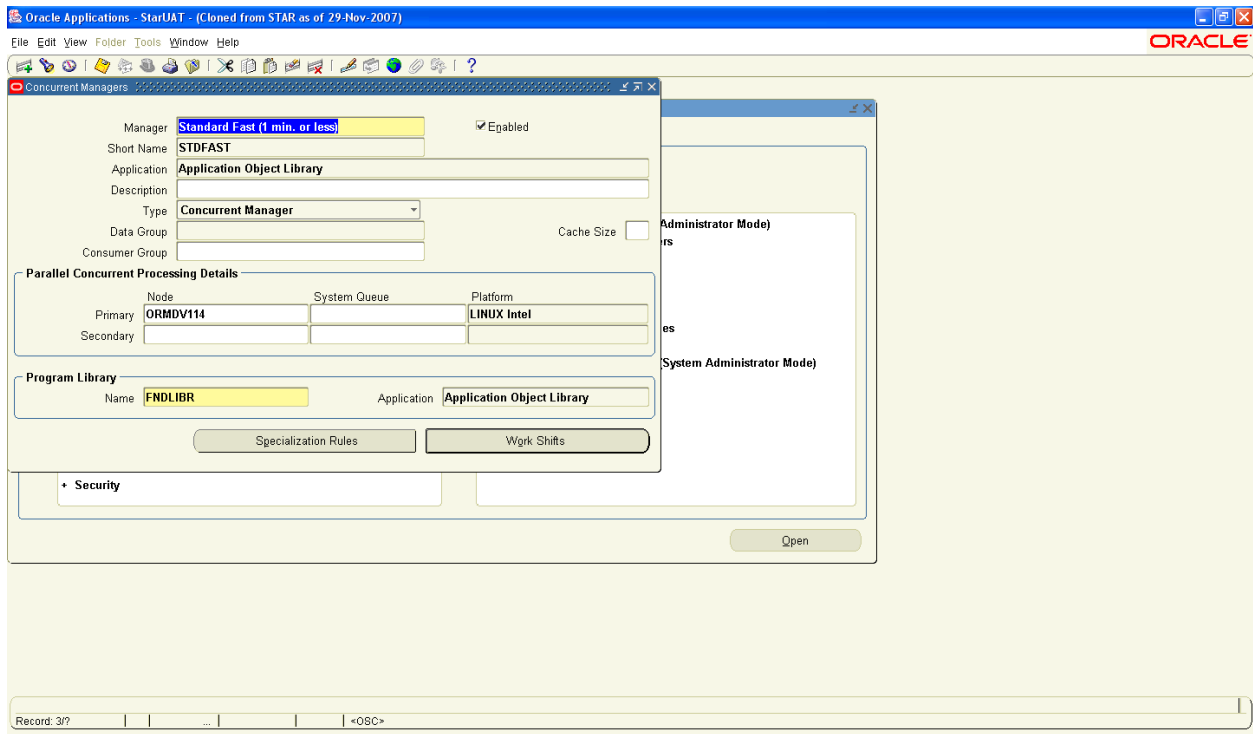
FRM-40352: Last record of query retrieved.
Record: 28/38 | | | | <OSC>

You should see it with a Status value of "Deactivated". Highlight the new concurrent manager by clicking on its name and then click the Activate button, which should be enabled at that point. Wait a minute and then click on Refresh. You should see the concurrent manager have Actual and Target values equal to the value of processes you specified during the creation of the concurrent manager.

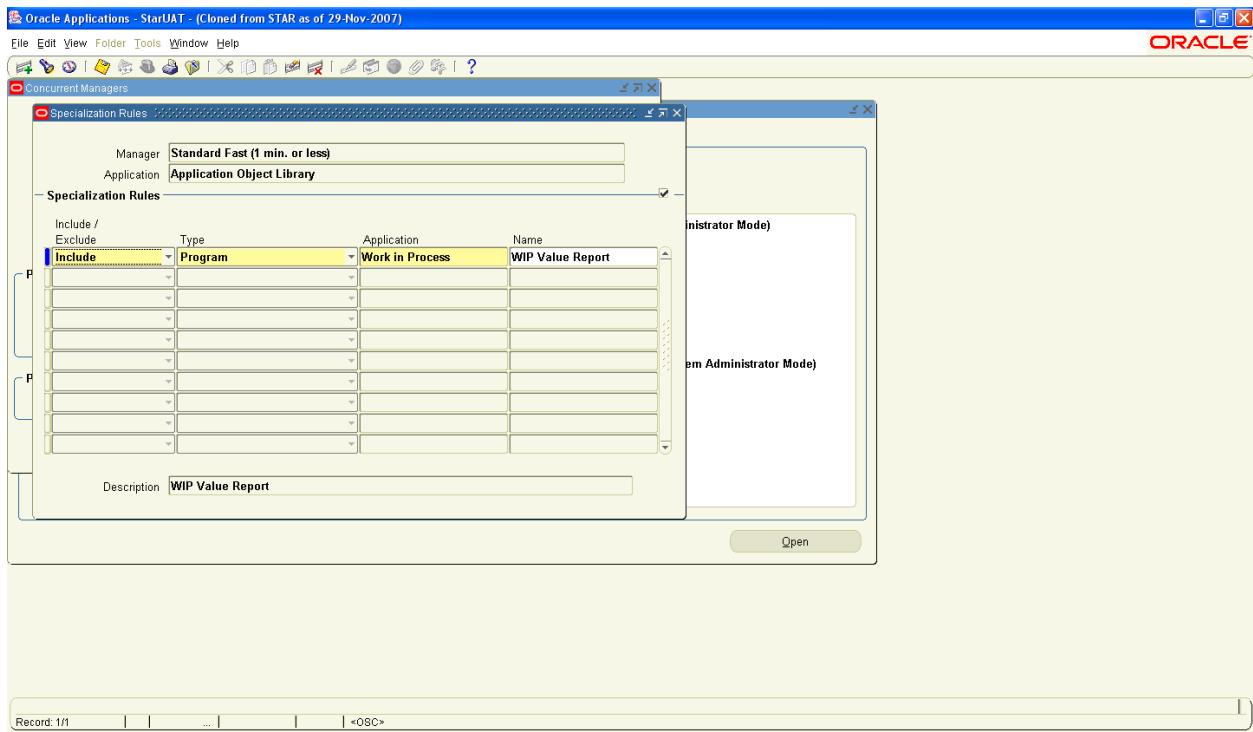


The final step in our process is to force the fast-running programs previously identified to utilize the new fast-running concurrent manager exclusively. In order to do this, we need to include these programs exclusively to the new manager. These next steps will provide the means to do this:

1. Log into the Oracle Applications program as a user whom has the System Administrator responsibility assigned to them.
2. Select the Define link under the Concurrent: Manager section
3. In the Concurrent Managers form, press <F11> to enter query mode, type in the name of the new concurrent manager in the Manager field, then press <CTRL><F11> and you should see something similar to this screenshot:



4. Click the Specialization Rules button and you should see something similar to this screenshot:



5. Click on any record in the include/exclude area, pull down the File menu at the top left of the screen, and select New to create a new record and use the following values for the respective fields:

Include/Exclude – Include

Type – Program

Application – The name of the fast-running programs' application (search tool available).

Name – The name of the fast-running program to exclude (search tool available).

6. Save the changes by pulling down the File menu at the top left of the screen and selecting Save.
7. At your earliest convenience, execute the program and verify that it ran under the new concurrent manager. You can use the query at the beginning of this document, substituting the name of the new concurrent manager under the value for `USER_CONCURRENT_QUEUE_NAME`.

Now that your expressway exists, you can concentrate on tuning slower-running processes without the added stress of having to deal with possible bottlenecks they are causing the faster processes.