How to create a long running request for use in testing
Or
How to register a concurrent program

A step by step guide

By
Gary Piper

June 2012

IMPORTANT NOTES:

- The long running request defined in this document does not use any system resources whilst running so it will not impact on the overall performance of your application.
- This is only for the use of OEBS professionals who have the skills and abilities to use these procedures safely. It should be used for test purposes only and never in a PRODUCTION environment. Refer to our Disclaimer before proceeding.

Use of this document is subject to the Legal Notice and Disclaimer detailed at PIPER-Rx.com © *2009 - 2012 G Piper*

Table of Contents

Use of this document is subject to the Legal Notice and Disclaimer detailed at PIPER-Rx.com ***© 2009 - 2012 G Piper***

1 Why a long running request?

Ever wanted a process you can use in testing that can jam up the concurrent managers without bringing the entire application to is knees, or easily generate concurrent requests that complete with a status of error or warning?

In this paper we will:

- Create a very simple SQL program that basically does nothing for the number of minutes you pass to the program and then,
- Show the step by step process to register that program as a concurrent program that can be run by the Systems Administrator.

A long running request program is very handy to have in your testing toolkit and will prove its self invaluable for both OEBS Application and Application monitoring tool testing. The long running request program described in this paper can be used:

- for testing long running requests without causing a massive processing overheads
- to jam up the concurrent managers (e.g to test pending request counts)
- to create requests that complete with a status of error
- to create requests that complete with a status of warning

When we built **PIPER-Rx** **A**pplication **M**onitor (***PAM***) we required a quick an easy method of testing a number of the ***PAM*** alert functions and the **PIPER-Rx** long running request program was used for this purpose. Full details of exactly how this was done are included in section 9 of this paper.

Important note:

This is only for the use of OEBS professionals who have the skills and abilities to use these procedures safely. It should be used for test purposes only and never in a PRODUCTION environment. Refer to our Disclaimer before proceeding.

Use of this document is subject to the Legal Notice and Disclaimer detailed at PIPER-Rx.com © ***2009 - 2012 G Piper***

2 Creating the long running concurrent program

Step 1: Create the SQL

You first need to create the SQL for the **PIPER-Rx** long running request program, setting the number of minutes to run as an argument - **&1**

Create an SQL*plus file (**PAM_LONG_RUN.sql**) containing the following lines only:-

```
exec dbms_lock.sleep((&1/4) * 60 );

exec dbms_lock.sleep((&1/4) * 60 );

exec dbms_lock.sleep((&1/4) * 60 );

exec dbms_lock.sleep((&1/4) * 60 );
```

Note: We have used multiple lines as we have found the lock sleep function has an upper limit in some instances. The use of multiple lines allows us to exceed the upper limit should one exist.

Step 2: Save the file in the $FND_TOP/sql directory

Use of this document is subject to the Legal Notice and Disclaimer detailed at PIPER-Rx.com ***© 2009 - 2012 G Piper***

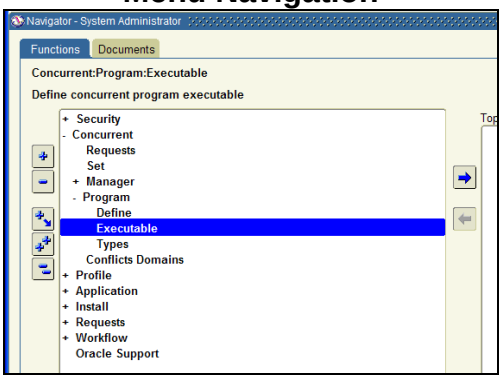
3 Register the concurrent program

We need to register the concurrent program by connecting to the Application as system administrator and register the executable (PAM_LONG_RUN.sql)

3.1 Step 1: Register the executable

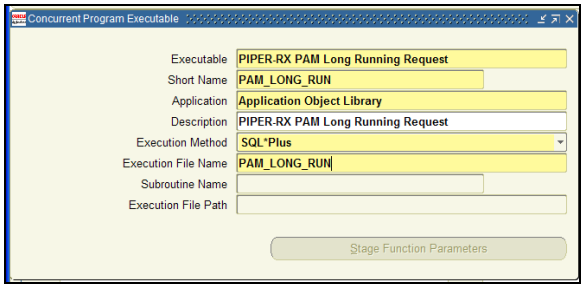
Once connected, navigate to the Concurrent > Programs > Executable screen

Menu Navigation



Create a new record with the following details:

Executable	PIPER-RX PAM Long Running Request
Short Name	PAM_LONG_RUN
Application	Application Object Library
Description	PIPER-RX PAM Long Running Request
Execution Method	SQL*Plus
Executable File Name	PAM_LONG_RUN

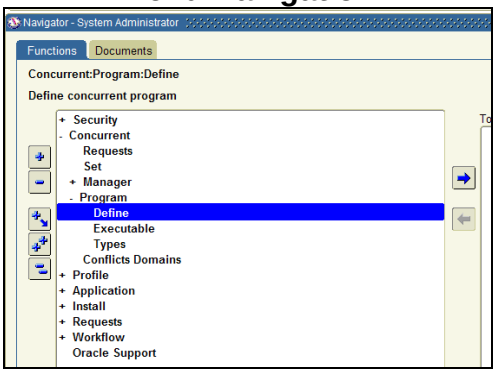


Use of this document is subject to the Legal Notice and Disclaimer detailed at PIPER-Rx.com © 2009 - 2012 G Piper

3.2 Step 2: Define the concurrent program

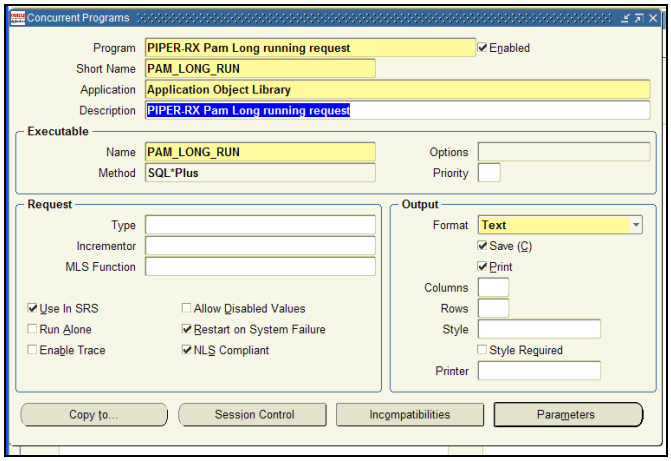
Next we need to define the executable. Navigate to the Concurrent > Programs > Define

Menu Navigation



Create a new record with the following details:

Program	PIPER-RX Pam Long running request
Short Name	PAM_LONG_RUN
Application (LOV)	Application Object Library
Description	PIPER-RX Pam Long running request
Executable Name (LOV)	PAM_LONG_RUN



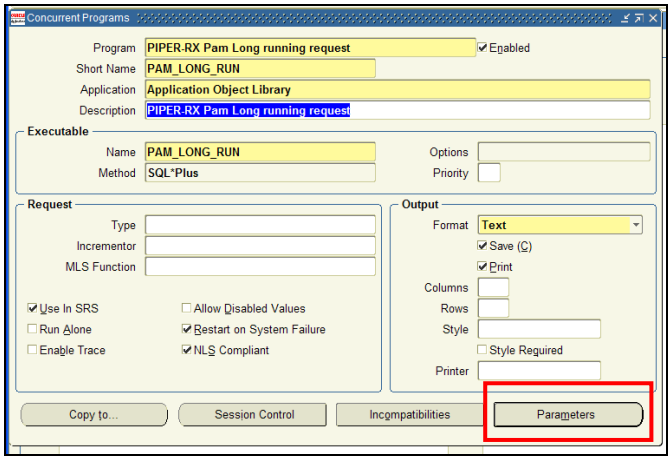
Note: It is important to check the output, save and print options as without these the ability to produce error and warning requests will not be available

Use of this document is subject to the Legal Notice and Disclaimer detailed at PIPER-Rx.com © 2009 - 2012 G Piper

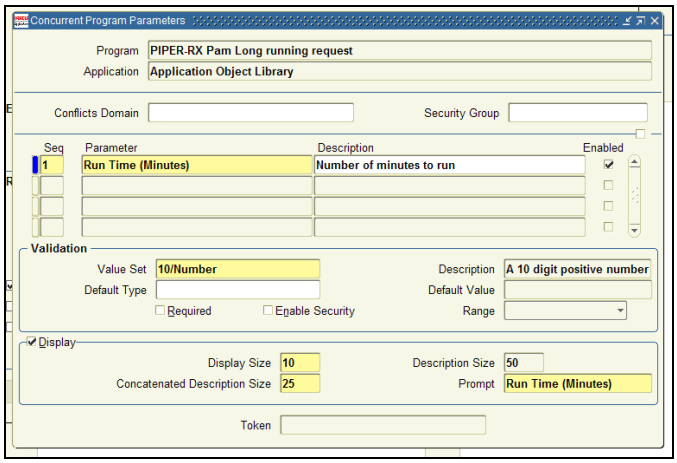
3.3 Step 3: Set the program parameters

The SQL created requires the number of minutes the program should run to be passed as a parameter. In the SQL the parameter is defined as &1.

Select the Parameters option at the bottom of the program define screen:



Fill in the details of the parameter to be sent to the concurrent program as shown below:



We have now successfully registered the concurrent program.

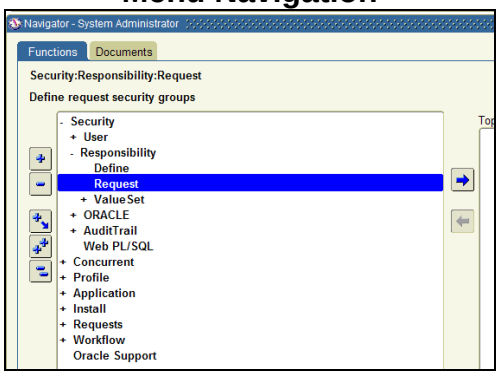
Use of this document is subject to the Legal Notice and Disclaimer detailed at PIPER-Rx.com © 2009 - 2012 G Piper

4 Assign the program to a responsibility

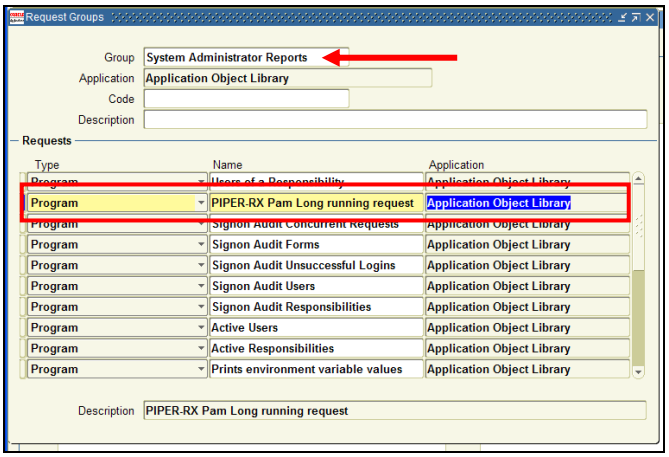
The program needs to be assigned to a responsibility.

Navigate to the Security > Responsibility > Requests screen

Menu Navigation



When in this screen search for the "**System Administrator Reports**" Group



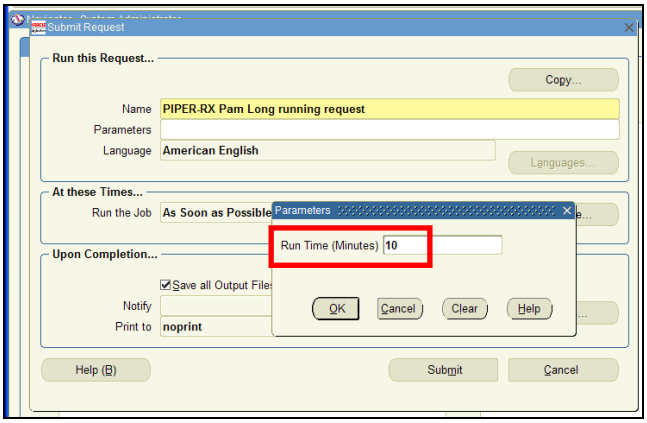
Insert a new record with the details shown above. As the program had been registered it will be available in a list of values in this screen.

Use of this document is subject to the Legal Notice and Disclaimer detailed at PIPER-Rx.com © ***2009 - 2012 G Piper***

5 Run the report

Using the normal run reports screen you should now see the newly registered program.

Set the parameter to the number of minutes you want the long running program to run for and submit it. In this example the runtime is set to 10 minutes.



And it should run…. ☺

Output

REQUEST_ID	USER_CONCURRENT_PROGRAM_NAME	USER_NAME	ARGUMENT_TEXT	ACTUAL_START_DATE	ACTUAL_COMPLETION_DATE	RUN_MIN
2170763	PIPER-RX Pam Long running request	GPIPER	10	02-Jun-09 11:20	02-Jun-09 11:31	10.3

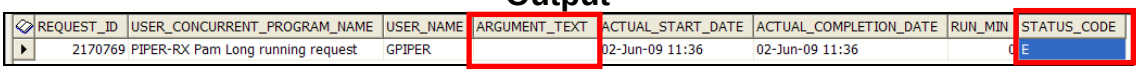
Use of this document is subject to the Legal Notice and Disclaimer detailed at PIPER-Rx.com © 2009 - 2012 G Piper

6 How to create a completed error request

When a concurrent request experiences an error during the program execution the concurrent program will complete with a status of Error.

To create a concurrent request that completes with a status of error, run the **PIPER-Rx** long running request program but **do not enter an argument**…. (null value). An invalid argument will cause the program to fail.

Output



REQUEST_ID	USER_CONCURRENT_PROGRAM_NAME	USER_NAME	ARGUMENT_TEXT	ACTUAL_START_DATE	ACTUAL_COMPLETION_DATE	RUN_MIN	STATUS_CODE
2170769	PIPER-RX Pam Long running request	GPIPER		02-Jun-09 11:36	02-Jun-09 11:36	0	E

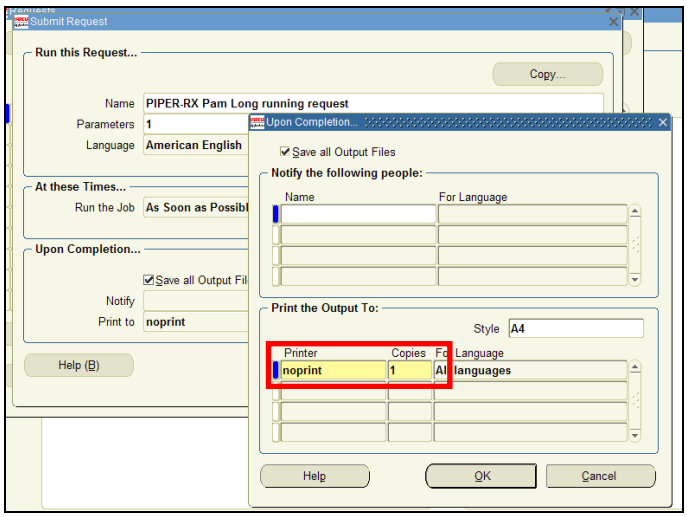
The request will complete with a completion status code of ‘E’ = Error

Use of this document is subject to the Legal Notice and Disclaimer detailed at PIPER-Rx.com ***© 2009 - 2012 G Piper***

7 How to create a completed warning request

When a concurrent request output is sent to a printer that does not exist, the concurrent program will complete with a status of Warning.

To create a concurrent request that completes with a status of warning, run the **PIPER-Rx** long running request program with a sleep time of 1 min sending the output to **no print** setting the number of **copies set to 1.**



Output

REQUEST_ID	USER_CONCURRENT_PROGRAM_NAME	USER_NAME	ARGUMENT_TEXT	ACTUAL_START_DATE	ACTUAL_COMPLETION_DATE	RUN_MIN	STATUS_CODE
2170771	PIPER-RX Pam Long running request	GPIPER	1	02-Jun-09 11:41	02-Jun-09 11:42	1.1	G

The request will complete with a completion status code of ‘G’ = Warning

Use of this document is subject to the Legal Notice and Disclaimer detailed at PIPER-Rx.com © ***2009 - 2012 G Piper***

8 How to jam up the standard concurrent managers

Firstly, you would only want to ever do this in limited circumstances, for example when testing alerts in a monitoring program you are designing for use with OEBS such as we did with the **PIPER-Rx** **A**pplication **M**onitor (***PAM***) (see section below).

By default the **PIPER-Rx** long running request program created in this paper will run in the standard queue. So unless you have assigned the program to another manager queue, the following steps affect the Standard queue.

In this example we will jam up the standard managers for 20 minutes.

First step is to identify how many concurrent manager processes the standard manager has:

```
SELECT max_processes,
       running_processes
  FROM applsys.fnd_concurrent_queues
 WHERE concurrent_queue_name = 'STANDARD';
```

If there are three (3) standard managers then run three (3) **PIPER-Rx** long running request programs setting the run time to 20 minutes.

Optional: Then run 10 “Active user reports”; this will cause the number of pending requests to increase to 10 and remain in that state for the 20 minutes until the long running requests complete.

Use of this document is subject to the Legal Notice and Disclaimer detailed at PIPER-Rx.com © ***2009 - 2012 G Piper***

9 How the PIPER-Rx long running request program was used to test *PAM* alerts

When we built **PIPER-Rx** **A**pplication **M**onitor (***PAM***) we required a quick an easy method of testing a number of the ***PAM*** alert functions and the **PIPER-Rx** long running request program was used for this purpose. How we used the program to test the various alerts in a range of scenarios is as follows:

Pending requests alert test scenario

PAM CM-002 - Alert when the number of pending requests exceeded the alert threshold.

First we set the ***PAM*** alert threshold to 10 so ***PAM*** would alert when there are more than 10 pending requests in any of the queues.

If there were three (3) standard managers we would run three (3) **PIPER-Rx** long running request programs setting the run time parameter to the time period we wanted to jam up the standard managers for (e.g. 20 minutes). We then ran then a further 12 **PIPER-Rx** long running request programs with a runtime parameter of 1 minute.

Thus we had three (3) requests running for 20 minutes in the three (3) standard managers and anything else we added to the standard manager would have a status of pending. The subsequent long running requests were set to 1 minute so that after the first three (3) 20 minute requests had completed the remaining requests would flow through quite quickly.

Error Requests alert test scenario

PAM CR-002 - Alert when the number of concurrent requests with a status of "Error" exceeded the alert threshold.

First we set the ***PAM*** alert threshold for requests that completed with a status of error to 5 so ***PAM*** would alert when there were more than 5 concurrent requests that completed with a status of error detected.

We then submitted six (6) **PIPER-Rx** long running request programs with a runtime parameter of **null**. Thus we generated six (6) concurrent requests that complete with a status of error triggering the ***PAM*** alert.

After the first ***PAM*** completed error requests alert was received we ran an additional **PIPER-Rx** long running request program to ensure we received an additional alert when subsequent requests completed with a status of error.

Use of this document is subject to the Legal Notice and Disclaimer detailed at PIPER-Rx.com ***© 2009 - 2012 G Piper***

Warning Requests alert test scenario

PAM CR-004 - Alert when the number of concurrent requests with a status of "Warning" exceeded the alert threshold

First we set the ***PAM*** alert threshold for requests that completed with a status of warning to 5 so ***PAM*** would alert when there were more than 5 concurrent requests that completed with a status of warning detected.

We submitted six (6) **PIPER-Rx** long running request programs with a runtime parameter of 1 (1 minute) and the output was sent to **no print with number of copies set to 1**. Thus we generated six (6) concurrent requests that complete with a status of warning triggering the ***PAM*** alert.

After the first ***PAM*** completed warning requests alert was received we ran an additional **PIPER-Rx** long running request program to ensure we received an additional alert when subsequent requests completed with a status of warning.

Duplicate requests alert test scenario

PAM CP-004 - Alert when duplicate requests detected

PAM defines a duplicate request as two or more of the same concurrent program with the same arguments submitted by the same user.
By default ***PAM*** excludes report set stages from duplicate requests.

All our testing required here was to repeat the steps for pending requests. When two or more of the same program with the same arguments are submitted by the same user a duplicate request alert is raised.

Duplicate requests alert exclusion

PAM provides the ability to exclude one or more report from the duplicate requests alert. To test these instances we added the **PIPER-Rx** long running request program to the duplicate requests exclusion list and re ran the duplicate requests test – which did not produce an alert… ☺

Testing long running request alert

PAM CM-003 – *Alert when long running requests detected*

This one is a more complex area as it depends on how you define a long running request. An overly simplistic view can result in requests being inappropriately terminated often leading to significant user frustration.

Use of this document is subject to the Legal Notice and Disclaimer detailed at PIPER-Rx.com © ***2009 - 2012 G Piper***

> I was once told by a DBA that any request that runs longer then 3 hours is long running and is terminated… I suggested they should consider an alternate strategy.

PAM takes a more considered approach. In order for a program to be assessed as long running by ***PAM***:

- There must be at least 5 prior run stats held by ***PAM***
- For any program to be even considered long running it must have been running for longer than 10 minutes
- ***PAM*** provides the ability to exclude concurrent programs from the ***PAM*** long running program check so the program must not be in the exclude list.

The following testing approach was used in the development of this alert:

Step 1: We manually deleted any prior run stats from the ***PAM*** long running history table ***piper_rx_pam_cr_runtimes***

Step 2: We ran 6 **PIPER-Rx** long running request programs setting the arguments to 15 minutes so as to set the runtime history.

Step 3: We ran 1 **PIPER-Rx** long running request program setting the runtime to 30 minutes. This generated the long running alert.

Other scenarios

We also used the program when testing the following ***PAM*** alerts.

CP-001 - Alert when selected programs are either missing or on-hold
CP-002 - Alert when selected programs complete with error or warning
CP-003 - Alert when selected programs have been submitted
CP-007 - Alert when duplicate scheduled request are detected
SPR-001- Alert when possible specialisation rule issues are found
SPR-002- Alert when a manager is disabled and there are specialisation rules assigned
SPR-003- Alert when a program is assigned a request type that does not exist
SPR-004- Alert a run alone request has been submitted

Use of this document is subject to the Legal Notice and Disclaimer detailed at PIPER-Rx.com © ***2009 - 2012 G Piper***

10 Want to know more?

There is loads more ***FREE*** information on all aspects of OEBS Application Administration at the **PIPER-Rx** website. I have had over 20+ years working with Oracle (the product, not the Company) and Oracle E-Business Suite (since Release 5). Since the late 1990's I have spent more time sharing these learnings and the most popular papers and case studies I have presented are available at www.**PIPER-Rx**.com as well as over 250 TOAD Reports Manager reports and a whole host of Tips and Reports I have used throughout my career.

All information is at the **PIPER-Rx**.com website ***FREE*** so why not check it out….I hope you find it useful! **– 40,000+ downloaders to date can't be wrong!**

Use of this document is subject to the Legal Notice and Disclaimer detailed at PIPER-Rx.com ***© 2009 - 2012 G Piper***

11 Disclaimer

All material contained in this document is provided by the author "as is" and any express or implied warranties, including, but not limited to, any implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the author be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of any content or information, even if advised of the possibility of such damage. It is always recommended that you seek independent, professional advice before implementing any ideas or changes to ensure that they are appropriate.

Oracle®, Oracle Applications® & Oracle E-Business Suite® are registered trademarks of Oracle Corporation
TOAD® is a registered trademark of Quest Software

Use of this document is subject to the Legal Notice and Disclaimer detailed at PIPER-Rx.com ***© 2009 - 2012 G Piper***