

## ***PAMtutorials* 1: Usage profiling**

**“Your OEBS Application - an introspective”**

**PIPER-**Rx** Application Monitor – **PAM**  
**VIRTUAL APPS ADMINISTRATOR****

***PAM* Version 4.0**

*“Blurring the line between software product and training”*

May 2012

## Table of Contents

1	What you'll get out of <i>PAMtutorials 1</i> .....	3
2	<i>PAM</i> Daily Activity Monitor .....	4
2.1	What the <i>PAM</i> Daily Activity Monitor does .....	4
2.2	What daily activity is recorded by <i>PAM</i> ? .....	4
2.2.1	User accounts.....	4
2.2.2	User activity .....	4
2.2.3	Concurrent request activity .....	6
2.2.4	Application response time.....	7
2.3	How <i>PAM</i> collects the daily activity information .....	8
3	So what does all this <i>PAM</i> daily activity information actually tell you and what should you do with it? .....	10
3.1	User accounts .....	10
3.1.1	Registered users .....	10
3.1.2	What to do with this active users information.....	13
3.2	Concurrent request activity.....	13
3.2.1	What to do with the concurrent request activity information.....	16
3.2.2	Completed requests .....	16
3.2.3	Completed Error .....	18
3.2.4	Completed Warning.....	19
3.2.5	Completed request LOG and OUT files .....	20
3.2.6	Trend analysis .....	21
3.3	User activity.....	21
3.3.1	Full service activity .....	22
3.3.2	Self-service activity.....	24
3.4	Application response time .....	27
3.4.1	What to do with the Application response time information .....	29
4	Daily activity comments.....	31
5	Clone protection feature.....	33
5.1	<i>PAM</i> clone e-mail alert.....	34
6	<i>PAM</i> Sign-on Audit Level Change Alert (GA-001).....	36
6.1	Changing the check frequency.....	38
7	The <i>PAM</i> Heartbeat (HB-001) e-mail .....	38
7.1	How to change the heartbeat e-mail check frequency .....	39
8	The <i>PAM</i> Sign-on audit (IN-006) reminder.....	39
8.1	Turning off the <i>PAM</i> sign-on audit alert.....	40
9	Monthly OEBS Activity Report.....	41
10	Known issues.....	42
11	Disclaimer .....	44

## 1    What you'll get out of *PAMtutorials* 1

The best OEBS system administrators intimately know and understand what is going on in their application on a daily basis and throughout the business cycle. For all the key factors that have the potential to impact OEBS performance and system health they know the answers. For example:

- How large is the application?
- How many concurrent requests do you run per day and how does this change with business peaks?
- Has the overall activity been increasing or decreasing over time?
- What are the maximum concurrent users connected and when?
- How hard are the web servers being hit?
- How many user accounts do you have, has it been increasing over time?
- How many users actually use your application?

At the end of this tutorial you will be able to answer these questions and more with confidence and we will also show you how to use this information.

*PAMtutorials* 1 is all about giving you an introspective on your OEBS application's activity profile to help you intimately know and understand what is going on in the application on a daily basis and throughout the business cycle. Like the best systems administrators, you too will:

- know your starting baseline for all the key factors that have the potential to impact OEBS performance and system health
- understand how growth is trending over time
- be able to pattern match errors with causal or associated factors
- understand if there are emerging "boiling frogs" within your application you can deal with proactively

**Note:** The *PAM* Daily Activity Collector will populate up to 31 days prior activity so you will have useful information straight away.

## 2 PAM Daily Activity Monitor

### 2.1 What the PAM Daily Activity Monitor does

The PAM Daily Activity Collector is designed to be run once per day collecting and records the information on the overall application activity for the prior day. Once you have a couple of months activity you should clearly be able to see the business cycles...

The PAM Daily Activity collector does not generate any alerts, however the information collected is used by some PAM checks as a reference.

PAM records the Daily Activity information in four (4) categories:

- ❖ User accounts
- ❖ User activity
- ❖ Concurrent request activity
- ❖ Application response time

### 2.2 What daily activity is recorded by PAM?

The following is a description of each of PAM's daily activity item stored in the PAM repository object `piper_rx_pam_daily_activity`.

#### 2.2.1 User accounts

##### 1. Registered\_users

**The number of registered users is the total number of registered application users.** A registered user is defined as a user's account that can be accessed by a user, i.e. where the account start date has passed and where the account has not been end dated (closed) or has been future end dated. In many instances this number may be high as it could include a number of accounts that have either never been accessed or have not been accessed for a period of time. We will refer to these in a later tutorial (Aged application accounts).

##### 2. Active\_users

**The number of distinct user accounts that use either full or self-service connections.** Where a user accesses both full and self-service session within the day that user will be counted as one user. This is one of the better measures of overall user activity as it provides the actual number of users accessing the application.

#### 2.2.2 User activity

## 1. fs\_connections

**The total number of full service sessions during the day.**

In the most current releases of OEBS (**O**racle **E**-**B**usiness **S**uite) both full service and self-service connections are recorded in the OEBS applications `fnd_logins` table (assuming the sign-on audit level is set to user or above) however, the information recorded by OEBS that differentiates a full and self-service connection often changes between OEBS releases. To retain OEBS version independence **PAM** distinguishes full service connections as a session that has accessed an application full service responsibility and as such **PAM** requires the application sign-on audit level set to either “responsibility” or “form” to record full service activity.

**Note:** **PAM** will warn you if your current sign-on audit level is set too low to record full service activity.

## 2. fs\_users

**The number of distinct user accounts that accessed the OEBS application via the full service application.** E.g. If user ABC connects to the application via the full service application three (3) times during the day that user will be counted as one (1).

## 3. fs\_total\_connect\_time

**The total session time (seconds) for all the full service sessions for the day.**

The connect time is based on the time between the full service sessions `fnd_logins.start_time` and `fnd_logins.end_time`

The primary issue with recording total connect time is that some sessions do not get closed (dead sessions) or the user leaves the session active overnight. In the case of dead sessions, these will be recorded as zero (0) connect time. Sessions that have been left running for long periods of time will be limited to the lesser of the actual session time or 12 hours run time regardless the actual session time.

## 4. fs\_avg\_connect\_time

**The average full service connect time (seconds) across all the days sessions.**

The connect time is based on the average time between the full service sessions `fnd_logins.start_time` and `fnd_logins.end_time`

## 5. fs\_stddev\_connect\_time

**The standard deviation of the full service connect time (seconds) across all the days sessions.** The connect time is based on the time between the full service sessions `fnd_logins.start_time` and `fnd_logins.end_time`.

## 6. ss\_connections

**The total number of self-service sessions during the day.**

A self-service session is identified by an entry in the applications [icx\\_sessions](#) table.

**Note:** Where a user's self-service session starts a full service session, that session will be recorded as one self-service session and one full service session.

**7. ss\_users**

**The number of distinct user accounts that accessed the OEBS application via the self-service application.**

E.g. If user ABC connects to the application via the self-service application three (3) times during the day that user will be counted as one (1).

**8. ss\_total\_connect\_time**

**The total session time (seconds) for all the self-service sessions for the day.**

The connect time is defined as the time between the start of the self-service session [icx\\_sessions.first\\_connect](#) and the last time a page request was requested during the session [icx\\_sessions.last\\_connect](#) or, on the rare occasion that the user closes the self-service session correctly, the time the session was closed which will also be recorded in [icx\\_sessions.last\\_connect](#).

**9. ss\_avg\_connect\_time**

**The average self-service session time (seconds) across all the day's sessions.**

**10. ss\_stddev\_connect\_time**

**The standard deviation of the self-service session time (seconds) across all the day's sessions**

**11. ss\_page\_requests**

**The total number of self-service page requests for the day.**

### **2.2.3 Concurrent request activity**

**1. cr\_request\_count**

**Total number concurrent requests submitted during the day.**

**2. cr\_warning\_count**

**The total number of concurrent requests that completed with a status of warning for the day.** A warning request is usually associated with a failure at the printing stage of the run.

### 3. `cr_error_count`

The total number of concurrent requests that completed with a status of error for the day. An error status is usually related to a failure of the request during runtime.

### 4. `cr_ofile_generated`

The total size in bytes of all the concurrent requests out files (request reports) generated for the day. The out files can generally be found in the directory \$APPLCSF.

### 5. `cr_lfile_generated`

The total size in bytes of all the concurrent requests log files generated for the day. The log files can generally be found in the directory \$APPLCSF.

<p><b>Note:</b> The directory pointer \$APPLCSF (<b>A</b>pplications <b>C</b>ommon <b>S</b>upport <b>F</b>iles) generally refers to \$COMMON_TOP/admin/ directory.</p>
--

## 2.2.4 Application response time

The *PAM* Application response time is based on the runtime of the concurrent program FNDOAMCOL. This program was selected as the preferred indicator of application response time as it is required to be run as part of the OEBS (**O**racle **E**-**B**usiness **S**uite) OAM application, it is run every 10 minutes (default) and it is consistent, that is, it performs the same tasks on the same application objects each time it runs. Therefore, its overall performance (*PAM* run time) is dependent on both the size and health of the tables and indexes it accesses and the other activity that is occurring at the time the program runs.

### 1. **rt\_sample\_size**

The number of response time samples generated during the day. Generally this should be approximately 144 samples per day (1 run every 10 minutes \* 24 hours). If the application or concurrent managers have been shutdown for a period of time during the day the sample size will be less.

### 2. **rt\_minimum**

The minimum response time for the day (seconds).

### 3. **rt\_average**

The average response time for the day (seconds).

### 4. **rt\_maximum**

The maximum response time for the day (seconds).

### 5. **rt\_stddev**

The standard deviation of the response times for the day (seconds).

## 2.3 How *PAM* collects the daily activity information

The *PAM* Daily Activity collector procedure [piper\\_rx\\_pam\\_da\\_monitor.collect\\_daily\\_activity](#) is designed to run once per day and collects the overall activity information of the prior day's activity based on data in the following OEBS application tables:

- `fnd_logins`
- `fnd_concurrent_requests`
- `icx_sessions`

and stores that information in the *PAM* [piper\\_rx\\_pam\\_daily\\_activity](#) table.

As this collector runs once per day reporting on the prior day's activity it is better to run this collection early in the morning so yesterday's data is available for you in the morning. By default this collection will run at 5 am.

Timing of Individual *PAM* collector/ monitors are controlled via entries in the *PAM* configuration object [piper\\_rx\\_pam\\_config](#).

**Special feature:** When the *PAM* Daily Activity collector is first run it populates the [piper\\_rx\\_pam\\_daily\\_activity](#) table with the past 31 days activity based on data in the above mentioned tables. The number of days history collected is entirely dependent

on the amount of information held on-line which is governed by your site's normal maintenance purging programs:

- Purge sign-on audit
- Purge self-service sessions
- Purge concurrent requests

This pre populating feature is specifically designed so as **PAM** can provide valuable historical information from the install date. The pre populating feature will also kick in if the collector has not been run for a few days in an attempt to fill in what would have otherwise been missing data.

However, if you purge some data the **PAM** history may understated, e.g. If you run a special daily purge of workflow background process information. Unless you purge more frequently than daily all subsequent collections should be accurate.

### 3    So what does all this *PAM* daily activity information actually tell you and what should you do with it?

One of the key benefits of the *PAM* daily activity function is that it will hold all daily activity statistics for up to 2 years (default). (One record per day – 365 records per year; such a small overhead for such useful information!) So for rollup information you don't have to worry about your standard maintenance purge program removing all historical data – *PAM* will be holding it for you.

*PAM* records the level of activity within your application and how that changes by day of the week and over a full business cycle for four (4) categories:

- User accounts
- Concurrent request activity
- User activity
- Application response time

As you will see from the *PAM* reports in each category below, this collection provides invaluable information on a range of key factors that have the potential to impact OEBS performance and system health.

#### 3.1    User accounts

##### 3.1.1    Registered users

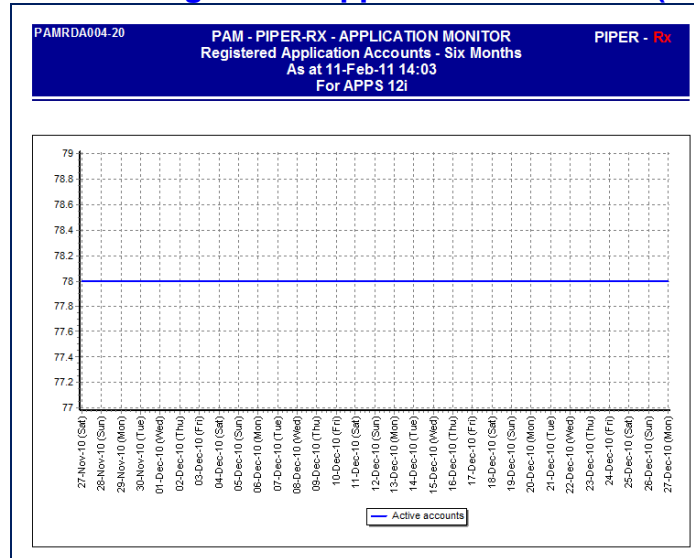
This information is extremely useful for profiling and will clearly identify the extent to which you are adding / closing (end dating) user accounts over time.

*PAM* provides two reports for reporting registered account numbers:

[PAMRDA003 Registered Application Accounts](#) which both charts and lists the number of registered users displaying all data points in the *PAM* daily activity collection.

[PAMRDA004 Registered Application Accounts \(6 months\)](#) which both charts and lists the number of registered users displaying the prior 6 months data points in the *PAM* daily activity collection.

Example **PAMRDA004 Registered Application Accounts (6 months)** report



Over time you will clearly be able to identify if the number of registered users has been increasing or decreasing, you will also be able to identify any maintenance activity around cleaning up (end dating) old accounts by a sharp drop in the number of registered accounts.

If you do undertake to identify and clean up aged accounts, you could use this report to clearly demonstrate the impact of your work.

Where the number of registered users has increased over the past year... this report could be used as evidence to justify why performance has been degrading over time and that your current application performance SLA may no longer be appropriate given this increase.

- Note 1:** All temporary accounts, temp staff, consultants etc. should have an end date set to the end of their contract period. By implementing this through the use of **PAM** reports you will be seen as proactive with internal audit 😊.
- Note 2:** **PAM** alert UA-010 provides an alert and report when end dated accounts are due to expire.

An active user is any user that accesses the application by either or full or self-service applications. Where a user accesses both full and self-service session within the day that user will be counted as one user.

Example: User Fred accesses the application once via full service, and later in the day launches full service session which in turn launches a self-service session. This will be counted as 1 active user for the day.

This information is of very high value for profiling and will clearly identify the number of registered users actually accessing your application and if this number is changing

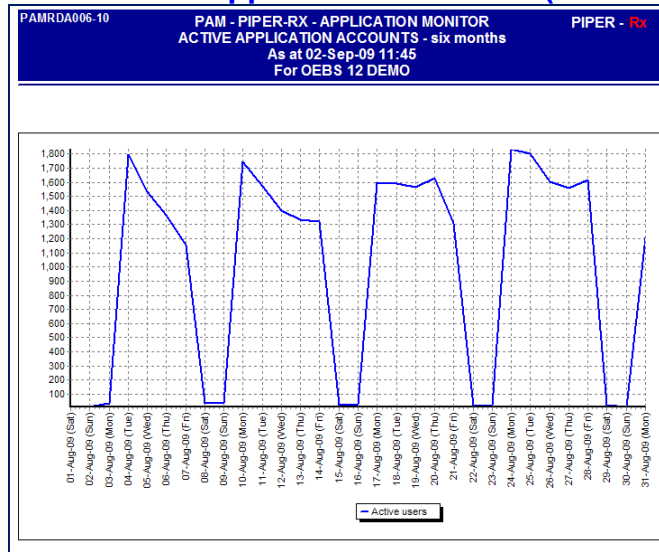
over time. This value can also be used as an SLA renegotiation trigger which will be covered in later **PAM** tutorials.

**PAM** provides two (2) reports for displaying the number of active users:

**PAMRDA005 Active Application Accounts** both charts and lists the actual and registered number of users displaying all data points in the daily activity collection.

**PAMRDA006 Active Application Accounts (6 months)** both charts and lists the actual and registered number of users displaying the prior 6 months data points in the daily activity collection.

Example **PAMRDA006 Active Application Accounts (6 months)** report –Page1



Example **PAMRDA006 Active Application Accounts (6 months)** report - Page 2

PAMRDA006-10 PIPER-RX - APPLICATION MONITOR ACTIVE APPLICATION ACCOUNTS - six months As at 02-Sep-09 11:45 For OEBS 12 DEMO PIPER - Rx			
Sample Time	Active Users*	Active Accounts	Pct Active
31-Aug-09 (Mon)	1,204	8,703	13.8
30-Aug-09 (Sun)	10	8,703	0.1
29-Aug-09 (Sat)	25	8,703	0.3
28-Aug-09 (Fri)	1,619	8,703	18.6
27-Aug-09 (Thu)	1,557	8,703	17.9
26-Aug-09 (Wed)	1,602	8,703	18.4
25-Aug-09 (Tue)	1,800	8,703	20.7
24-Aug-09 (Mon)	1,833	8,703	21.1
23-Aug-09 (Sun)	25	8,703	0.3
22-Aug-09 (Sat)	17	8,703	0.2
21-Aug-09 (Fri)	1,311	8,703	15.1
20-Aug-09 (Thu)	1,628	8,703	18.7
19-Aug-09 (Wed)	1,568	8,703	18.0
18-Aug-09 (Tue)	1,588	8,703	18.2
17-Aug-09 (Mon)	1,598	8,703	18.4
16-Aug-09 (Sun)	31	8,703	0.4
15-Aug-09 (Sat)	22	8,703	0.3
14-Aug-09 (Fri)	1,320	8,703	15.2
13-Aug-09 (Thu)	1,335	8,703	15.3
12-Aug-09 (Wed)	1,400	8,702	16.1
11-Aug-09 (Tue)	1,573	8,702	18.1
10-Aug-09 (Mon)	1,743	8,702	20.0
09-Aug-09 (Sun)	37	8,731	0.4
08-Aug-09 (Sat)	34	8,731	0.4
07-Aug-09 (Fri)	1,158	8,731	13.3
06-Aug-09 (Thu)	1,366	8,731	15.6
05-Aug-09 (Wed)	1,534	8,731	17.6
04-Aug-09 (Tue)	1,794	8,731	20.5
03-Aug-09 (Mon)	36	8,731	0.4
02-Aug-09 (Sun)	13	8,731	0.1
01-Aug-09 (Sat)	11	8,731	0.1
<b>Max Active Users</b>	<b>1,833</b>	<b>Max Pct Active</b>	<b>21.1</b>

In this simple example you can clearly see the lower activity on weekends. Page two of the report shows the actual active users as a percentage of total registered users.

The maximum number of users and maximum actual active users as a percentage of total registered users for the report period is displayed at the end of the reports.

### 3.1.2 What to do with this active users information

Based on the above example report, you could discuss with the business:

- ❖ “Generally only 1,600 users, which being approximately 21% of registered users, are accessing the application”
- ❖ “We appear to have a large number of accounts that are not being used and should be closed” - (aged user cleanup – which will be covered in later **PAM** tutorials) “I am concerned as this may represent a security issue”.

### 3.2 Concurrent request activity

The question, “what is the size of your application” draws many and varied responses which often don’t provide a good “size” indicator:

- ❖ A pure DBA will generally say **NN** Gb on an **N** node RAC cluster... blah blah blah...

I have encountered sites with several terabytes of AR tables that only has about 100 active users and running less than 5,000 concurrent requests per day. This site was recording one transaction per small value activity at a very high frequency.

- ❖ An Applications Administrator will generally say “We have **N** thousand users”

I have encountered sites where they have several million registered users of which only a couple of hundred were active. This site was registering an application user almost on a transaction request basis.

In reality both these answers are of little value in understanding the size of the application.

Over 90% of the overall OEBS application activity is conducted via the concurrent managers, so their activity and throughput is of paramount importance when understanding your application processing profile. The number of request processed daily is one of the best indicators of overall activity of your application.

An answer of “approximately 12,000 requests per day” generally provides a good indicator of the application size and load. (Note: You need to use the term “approximately” as the number of requests will change over the full business cycle.)

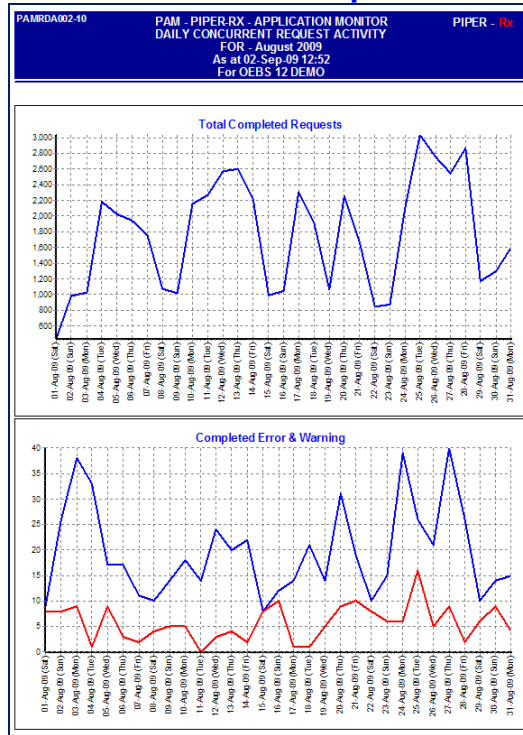
I have come across several sites where 50 to 75% of all concurrent requests were workflow background processes – really. Refer to [The Hidden cost of Workflow](#) paper located at the **PIPER-Rx** website for more information on this issue.

**PAM** provides two (2) reports for displaying daily concurrent request activity:

**PAMRDA001 Concurrent Request Activity** both charts and lists the number of concurrent requests per day displaying all data points in the daily activity collection

**PAMRDA002 Concurrent Request Activity (month)** both charts and lists the number of concurrent requests per day displaying all data points for a given month

Example PAMRDA001 Concurrent Request Activity report - Page1



Example PAMRDA001 Concurrent Request Activity report – Page 2

PAMRDA002-10 PAM - PIPER-RX - APPLICATION MONITOR PIPER - Rx					
DAILY CONCURRENT REQUEST ACTIVITY					
FOR - August 2009					
As at 02-Sep-09 12:52					
For OEBS 12 DEMO					
Sample Time	Total Requests	Error	Warning	OFIL (Mb)	LFIL (Mb)
31-Aug-09 (Mon)	1,606	4	15	766	2
30-Aug-09 (Sun)	1,299	9	14	619	1
29-Aug-09 (Sat)	1,176	6	10	561	1
28-Aug-09 (Fri)	2,862	2	26	1,365	3
27-Aug-09 (Thu)	2,549	9	40	1,215	3
26-Aug-09 (Wed)	2,755	5	21	1,314	3
25-Aug-09 (Tue)	3,035	16	26	1,447	3
24-Aug-09 (Mon)	2,090	6	39	997	2
23-Aug-09 (Sun)	879	6	15	419	1
22-Aug-09 (Sat)	849	8	10	405	1
21-Aug-09 (Fri)	1,681	10	19	802	2
20-Aug-09 (Thu)	2,256	9	31	1,076	3
19-Aug-09 (Wed)	1,064	5	14	507	1
18-Aug-09 (Tue)	1,904	1	21	908	2
17-Aug-09 (Mon)	2,311	1	14	1,102	3
16-Aug-09 (Sun)	1,050	10	12	501	1
15-Aug-09 (Sat)	994	8	8	474	1
14-Aug-09 (Fri)	2,222	2	22	1,060	3
13-Aug-09 (Thu)	2,598	4	20	1,239	3
12-Aug-09 (Wed)	2,572	3	24	1,226	3
11-Aug-09 (Tue)	2,277	0	14	1,086	3
10-Aug-09 (Mon)	2,154	5	18	1,027	2
09-Aug-09 (Sun)	1,022	5	14	487	1
08-Aug-09 (Sat)	1,074	4	10	512	1
07-Aug-09 (Fri)	1,753	2	11	836	2
06-Aug-09 (Thu)	1,945	3	17	927	2
05-Aug-09 (Wed)	2,027	9	17	967	2
04-Aug-09 (Tue)	2,180	1	33	1,040	2
03-Aug-09 (Mon)	1,027	9	38	490	1
02-Aug-09 (Sun)	985	8	26	470	1
01-Aug-09 (Sat)	441	8	9	210	1

Page one of each of these reports provides two (2) charts showing the total number of concurrent requests per day and the number of requests that completed with a

status of error or warning. Page two of each report provides the detailed daily activity information:

- Total requests
- Number of error requests
- Number of warning requests
- Total concurrent manager out file size generated
- Total concurrent manager log file size generated

### 3.2.1 What to do with the concurrent request activity information

Firstly you should clearly be able to identify business cycle trends, e.g. in an accounting based implementation you should see a marked increase concurrent manager activity around the month end processing period.

If the number of requests has been increasing over time placing a higher load on the application this information can be used to explain to the business perhaps why performance is degrading over time. This information could also be used in an SLA renegotiation which will be covered in later tutorials.

### 3.2.2 Completed requests

If you find that there is a day where there is a much larger than normal concurrent manager activity, you may want to consider the merits of running the following reports for the day in question to identify who or what may be responsible for the increased load:

#### 3.2.2.1 Completed requests by program

The **PAM** administration report **PAMADCR006 Completed Requests by Program (summary)** lists all the concurrent programs that completed on a selected day.

Example **PAMADCR006 Completed Requests by Program (summary)** report

Program		Total Runs	Normal	Error	Warn	Canc	Term
Appn Id	Program Id	Program Name					
0	37835	PIPER-RX PAM Collector	64	64	0	0	0
0	32263	Purge Concurrent Request and/or Manager Data	2	2	0	0	0
0	32592	Purge Signon Audit data	1	1	0	0	0
0	36888	Workflow Background Process	1	1	0	0	0
<b>Completed Totals:</b>			<b>68</b>	<b>68</b>	<b>0</b>	<b>0</b>	<b>0</b>

From this report you will be able to identify the most frequently requested reports, in addition you will also be able to identify those report that completed with a status of error or warning.

The next step is to run the **PAM** administration report [PAMADCR007 Completed Requests by Program \(Detail\)](#) to list all the run information for a selected concurrent program on a selected day

Example [PAMADCR007 Completed Requests by Program \(Detail\)](#) report

PAMADCR007-20 PAM - PIPER-RX - APPLICATION MONITOR PIPER - Rx Completed Requests By Program (0:37835) - PIPER-RX PAM Collector For 24-Jan-11 (Mon) As at 11-Feb-11 (Fri) 14:22 APPS 12i							
Request Id	Submitted By	Argument Text	Started	Completed	Run Time DD:HH:MM	Printer	Status
305824	GPIPER	(None)	24-Jan-11 19:40	24-Jan-11 19:40	00:00:00	adsprinter (0)	Normal
305823	GPIPER	(None)	24-Jan-11 19:30	24-Jan-11 19:30	00:00:00	adsprinter (0)	Normal
305822	GPIPER	(None)	24-Jan-11 19:20	24-Jan-11 19:20	00:00:00	adsprinter (0)	Normal
305821	GPIPER	(None)	24-Jan-11 19:10	24-Jan-11 19:10	00:00:00	adsprinter (0)	Normal
305820	GPIPER	(None)	24-Jan-11 19:00	24-Jan-11 19:00	00:00:00	adsprinter (0)	Normal
305819	GPIPER	(None)	24-Jan-11 18:50	24-Jan-11 18:50	00:00:00	adsprinter (0)	Normal
305818	GPIPER	(None)	24-Jan-11 18:40	24-Jan-11 18:40	00:00:00	adsprinter (0)	Normal
305817	GPIPER	(None)	24-Jan-11 18:30	24-Jan-11 18:30	00:00:00	adsprinter (0)	Normal
305816	GPIPER	(None)	24-Jan-11 18:20	24-Jan-11 18:20	00:00:00	adsprinter (0)	Normal
305815	GPIPER	(None)	24-Jan-11 18:10	24-Jan-11 18:10	00:00:00	adsprinter (0)	Normal
305814	GPIPER	(None)	24-Jan-11 18:00	24-Jan-11 18:00	00:00:00	adsprinter (0)	Normal

From this report you will be able to identify when and who ran the selected concurrent program for the selected day. In addition you could also be able to correlate the programs arguments and runtime.

### 3.2.2.2 Completed requests by user

The **PAM** administration report [PAMADCR008 Completed Requests by Requestor \(summary\)](#) lists all the application users that ran concurrent programs that completed on a selected day.

Example [PAMADCR008 Completed Requests by Requestor \(summary\)](#) report

PAMADCR008-20 PAM - PIPER-RX - APPLICATION MONITOR PIPER - Rx Completed Requests Summary By Requestor For 24-Jan-11 (Mon) As at 11-Feb-11 (Fri) 14:23 APPS 12i								
User Id	User Name	Description	Total Runs	Normal	Error	Warn	Canc	Term
1914	GPIPER	Gary Piper	68	68	0	0	0	0
Completed Totals:			68	68	0	0	0	0

From this report you will be able to identify the most frequently requestors, in addition you will also be able to identify users generating requests that complete with a status of error or warning.

The next step is to run the **PAM** administration report [PAMADCR009 Completed Requests by Requestor \(Detail\)](#) to list all the concurrent request runs for a selected user on a selected day.

Example [PAMADCR009 Completed Requests by Requestor \(Detail\)](#) report

PAMADCR007-20		PAM - PIPER-RX - APPLICATION MONITOR					PIPER -Rx	
Completed Requests By Requestor								
(1914: GPIPER) - Gary Piper								
For 24-Jan-11 (Mon)								
As at 11-Feb-11 (Fri) 14:23								
APPS 12i								
Req Id	Program Name	Argument Text	Started	Completed	Run Time DD:HH:MM	Printer	Status	
305824	PIPER-RX PAM Collector	(None)	24-Jan-11 19:40	24-Jan-11 19:40	00:00:00	adsprinter (0)	Normal	
305823	PIPER-RX PAM Collector	(None)	24-Jan-11 19:30	24-Jan-11 19:30	00:00:00	adsprinter (0)	Normal	
305822	PIPER-RX PAM Collector	(None)	24-Jan-11 19:20	24-Jan-11 19:20	00:00:00	adsprinter (0)	Normal	
305821	PIPER-RX PAM Collector	(None)	24-Jan-11 19:10	24-Jan-11 19:10	00:00:00	adsprinter (0)	Normal	
305820	PIPER-RX PAM Collector	(None)	24-Jan-11 19:00	24-Jan-11 19:00	00:00:00	adsprinter (0)	Normal	
305819	PIPER-RX PAM Collector	(None)	24-Jan-11 18:50	24-Jan-11 18:50	00:00:00	adsprinter (0)	Normal	

From this report you will be able to identify the selected user’s concurrent request activity for the selected day.

**Note:** All four (4) above mentioned reports access information held on-line in the [fnd\\_cocurrent\\_requests](#) table. Your sites purging policy may affect the quality of information in these reports.

Once you have identified the culprit, you can then approach them to find out why there has been increased activity and if this is going to be the norm in the future as you will need to plan for this. You may also like to point out that the extra load on the application due to their large than normal number of requests will be affecting all other application users. ☺

It would be even better if you could match this activity to degradation in application response time... It may even be worthwhile running the **PAM** reports [PAMRGA002 Intraday Response Time History \(day\)](#) report which would charts the overall application response time for the day in question.

### 3.2.3 Completed Error

A program that completes with the status of error normally indicates that the concurrent program did not complete due to an error encountered during processing, in this instance you should look at the errored requests completion text stored in the [fnd\\_concurrent\\_requests.completion\\_text](#) attribute, whilst often this information is not very explanatory it is a good starting place in the issue resolution process. Let’s hope it’s not a signal 11 ☺

For a list of concurrent requests that completed with a status of error for any given day you can use the **PAM** alerts [PAMACR001 Completed Error \(day\)](#) report. This report draws its information from [fnd\\_concurrent\\_requests](#), so if the concurrent requests have been purged for the day of interest this report will have no data to show.

Example **PAMACR001 Completed Error (day)** report

PAMACR001-20		PAM - PIPER-RX - APPLICATION MONITOR		PIPER - Rx
Completed Error Requests - For 11-FEB-11 (Fri)				
As at 11-Feb-11 14:52:59				
For APPS 12i				
Request ID	Requestor Argument	Program Name	Start Date	Completion Text
305992	GPIPER None	PIPER-RX Pam Long running request	11-Feb-11 14:49	Concurrent Manager encountered an error while running SQL*Plus for your concurrent request 305992.
305993	GPIPER None	PIPER-RX Pam Long running request	11-Feb-11 14:49	Concurrent Manager encountered an error while running SQL*Plus for your concurrent request 305993.
305994	GPIPER None	PIPER-RX Pam Long running request	11-Feb-11 14:49	Concurrent Manager encountered an error while running SQL*Plus for your concurrent request 305994.
305995	GPIPER None	PIPER-RX Pam Long running request	11-Feb-11 14:49	Concurrent Manager encountered an error while running SQL*Plus for your concurrent request 305995.

The information provided in the completion text is not perfect, but it is a start. The next step would be to review at the request output log file.

**3.2.4 Completed Warning**

A program that completes with the status of warning generally indicates that the concurrent program completed successfully but encountered an issue in its post processing activity, in most cases this is due to a printing problem. You should look at the warning requests printer assignment to identify if a particular printer is causing this problem.

For a list of concurrent requests that completed with a status of warning for any given day you can use the **PAM** alerts report **PAMACR002 Completed Warning (day)**. This report draws its information from `fnd_concurrent_requests`, so if the concurrent requests have been purged for the day of interest this report will have no data to show:

Example **PAMACR002 Completed Warning (day)** report

PAMACR002-10		PAM - PIPER-RX - APPLICATION MONITOR		PIPER - Rx
COMPLETED WARNING - For 25-AUG-09 (Tue)				
As at 25-Aug-09 14:12:05				
For OEBS 12 DEMO				
Request ID	Requestor Argument	Program Name	Start Date	Printer
2171194	GPIPER 3	PIPER-RX Pam Long running request	25-Aug-09 13:41	noprint(1)
2171201	GPIPER 1	PIPER-RX Pam Long running request	25-Aug-09 13:46	noprint(1)

In this example, sending a report to “noprint” with copies greater than zero is a sure fire way to generate a report that completes with a status of warning...

### 3.2.5 Completed request LOG and OUT files

When a concurrent request runs in most cases it produces both a LOG file recording the request run information and an OUT file which in most cases is the actual report.

The total Log and Out file sizes being generated can be obtained from the two (2) reports mentioned above:

- ❖ PAMRDA001 Concurrent Request Activity
- ❖ PAMRDA002 Concurrent Request Activity (month)

Page two of each of these reports details the total log and out file sizes being generated. You can use this portion of the reports to identify if there are any days during which larger amounts of LOG and OUT file have been produced.

For more detailed information you can use the **PAM** admin report **PAMADCR011 Completed Requests Log and Out file (detail)** to list individual requests that have generated more than **X** Mb of Log or Out file size.

Example **PAMADCR011 Completed Requests Log and Out file (detail)** report

PAM - PIPER-RX - APPLICATION MONITOR Completed Requests - Log & Out File Details For Files Larger Than 2 Mb For 04-JAN-07 (Thu) As at 12-Feb-11 (Sat) 13:48 APPS 12i						
Req Id	Appn Id	Prog Id	Program Name	Submitted By	Completion Date	
4340214	515	109431	Service Contracts ASO Queue Migration	SYSADMIN	04-Jan-07 16:32	
Arguments:						
Out File:					Printer:	no print (0)
Log File:						0.0 (Mb)
						37.4 (Mb) Total: 37.4 (Mb)
4297585	0	46314	Compile Non-Compiled Flexfields	SYSADMIN	04-Jan-07 16:02	
Arguments: N						
Out File:					Printer:	no print (0)
Log File:						0.0 (Mb)
						2.4 (Mb) Total: 2.4 (Mb)
4297584	0	46314	Compile Non-Compiled Flexfields	SYSADMIN	04-Jan-07 16:02	
Arguments: N						
Out File:					Printer:	no print (0)
Log File:						0.0 (Mb)
						2.3 (Mb) Total: 2.3 (Mb)
4297583	0	46314	Compile Non-Compiled Flexfields	SYSADMIN	04-Jan-07 16:02	
Arguments: N						
Out File:					Printer:	no print (0)
Log File:						0.0 (Mb)
						2.3 (Mb) Total: 2.3 (Mb)
Total Requests: 4						

**Note:** I have encountered instances where the free space in the concurrent request outfile location (\$APPLCSF) has decreased to such a low level the concurrent managers stopped running requests until more space has been allocated or a manual the report outputs cleanup.

These reports should provide you sufficient information to seek out the guilty party. Once you have identified the culprit, you can then approach them to find out why there has been increased activity and if this is just a one off or is going to be the norm in the future as you will need to plan for this. You may need to plan for additional disk space.

### 3.2.6 Trend analysis

Once **PAM** has collecting information for approximately 2 - 3 months, trend information will become apparent. With **PAM** report **PAMRDA001 Concurrent Request Activity**, the concurrent manager activity history held in the **PAM** daily activity table will clearly show your overall concurrent request activity trend.

You will also be able to demonstrate if the number of error and/or warning requests is *decreasing* over time (perhaps due to resolution work carried out by your team 😊) or *increasing* (perhaps as a result of a patch or new development that was applied).

You can also show if the amount of concurrent request out file (ofile) space requirement has been increasing over time, which may aid in your justification of more disk space.....

### 3.3 User activity

User activity information collected by **PAM** is split into the two primary methods of connecting to the application; full service and self-service.

Where a user accesses the application via a full service session and then launches a self-service session from within the full service session, this is counted as one full service session and one self-service session.

The best systems administrators know just how important it is to identify what level of user activity you have and how that activity is changing over time and across the business cycle. This helps the administrator understand peaks and to put in place any necessary strategies to manage them. Some useful examples are:

- The number of sessions versus the number of full service users provides an insight into whether your users are logging in and out of the application during the day or are just staying connected (possible security issue).
- The number of daily page requests can be used to identify how hard your web servers are being hit. It can also be used to identify if web server load is attributed to the OEBS application. I.e. if your web servers are being hammered and the number of page requests via the OEBS application has not significantly changed, the issue is most likely not related to the OEBS application.
- You will be able to clearly identify the impact of adding new OEBS modules or functionality.

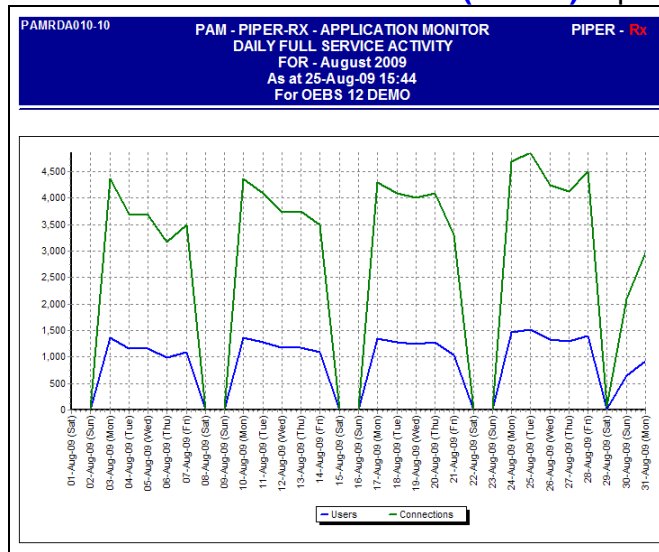
### 3.3.1 Full service activity

Full service activity is defined as any user account that has accessed the application via the full service application screens. **PAM** provides two reports for displaying full service user activity:

**PAMRDA009 FS Connections** both charts and lists the total full service activity per day displaying all data points in the **PAM** daily activity repository.

**PAMRDA010 FS Connections (month)** both charts and lists the total full service activity per day displaying all data points for a given month.

Example **PAMRDA010 FS Connections (month)** report – Page 1



Example **PAMRDA010 FS Connections (month)** report – Page 2

Sample Time	Users	Connections	Total (Hrs)	Average (Min)	STDEV (Min)
31-Aug-09 (Mon)	926	2,964	23,712	50	11
30-Aug-09 (Sun)	656	2,099	16,792	35	8
29-Aug-09 (Sat)	21	67	536	2	0
28-Aug-09 (Fri)	1,406	4,498	35,984	75	16
27-Aug-09 (Thu)	1,288	4,120	32,960	69	15
26-Aug-09 (Wed)	1,326	4,243	33,944	71	15
25-Aug-09 (Tue)	1,519	4,860	38,880	81	18
24-Aug-09 (Mon)	1,464	4,685	37,480	79	17
23-Aug-09 (Sun)	0	0	0	0	0
22-Aug-09 (Sat)	0	0	0	0	0
21-Aug-09 (Fri)	1,032	3,303	26,424	56	12
20-Aug-09 (Thu)	1,278	4,091	32,728	69	15
19-Aug-09 (Wed)	1,251	4,003	32,024	67	15
18-Aug-09 (Tue)	1,279	4,092	32,736	69	15
17-Aug-09 (Mon)	1,341	4,291	34,328	72	16
16-Aug-09 (Sun)	0	0	0	0	0
15-Aug-09 (Sat)	0	0	0	0	0
14-Aug-09 (Fri)	1,092	3,493	27,944	59	13
13-Aug-09 (Thu)	1,175	3,760	30,080	63	14
12-Aug-09 (Wed)	1,169	3,740	29,920	63	14
11-Aug-09 (Tue)	1,279	4,092	32,736	69	15
10-Aug-09 (Mon)	1,364	4,366	34,928	73	16
09-Aug-09 (Sun)	0	0	0	0	0

In this report “users” are the number of distinct user accounts that accessed the OEBS full service application. For example, if user Fred accesses the application once via full service, and again later in the day, this will be counted as 1 user and 2 sessions.

These reports provide useful statistics for profiling your overall full service usage over time:

### Sessions per user

If the user count is low and the session count is high, this indicates the users are continually reconnecting to the application throughout the day. In full service sessions this could indicate the users are switching responsibilities by logging out to logging back in.

### Average session time

If the average session time is low this indicates the users are completing short sharp sessions - login, do what they need to do and log out.

One of the things to look out for is session times greater than 5+ hours. This could indicate that users are coming in at 9:00 and not logging out when they go to lunch. For basic security, users should logout when they are not using the application or at least lock their screens.

### How to interpret the standard deviation values

What follows is a pretty simplistic description of standard deviation which may annoy some of the purists out there....we know the average is not the mean and the use of standard deviation assumes a normal distribution of data etc...but what we have here is good enough for our purposes. Therefore, the standard deviation should only be used as a very loose guide and not a statistical given.

So here goes..... The standard deviation is the statistical spread of data around the median value (we use average) and can often identify the story behind the data. It basically lets you know how tightly the data samples are clustered around the mean (average). When the standard deviation is small it indicates that most of the data samples are packed around the average value. When the standard deviation is large then there are a wide spread of data samples around the average value

The above **PAM** report shows one (1) standard deviation, this tells us that approximately 68% of all the samples fall within the range of (average – 1 \* standard deviation) and (average + 1 \* standard deviation).

So if the average is 10 minutes with a standard deviation of 2 minutes – approximately 68% of all the samples fell within the range 8 (avg – (1 \* stddev)) to 12 (avg + (1 \* stddev)) or approximately 95% of all the samples fell within the range 6 (avg – (2 \* stddev)) to 14 (avg + (2 \* stddev))

As I have said it's not a perfect calculation by any stretch of the imagination, but it is something...

So, if the standard deviation is getting larger over time then the samples have more variation, where as if the standard deviation is getting smaller over time then the samples are getting more consistent around the average.

Thus an increasing average with a small standard deviation is in fact quite a good thing as the data is more predictable. For example, the system may be getting slower but at a very predictable rate.

A decreasing average with a high standard deviation is not a good thing. Whilst the average is decreasing, the variation is very wide and unpredictable.

### 3.3.2 Self-service activity

Self-service activity is any user account that has accessed the application via the self-service application.

*PAM* provides four (4) reports for displaying self-service user activity:

[PAMRDA011 SS Connections](#) both charts and lists the total self-service activity per day displaying all data points in the *PAM* daily activity repository.

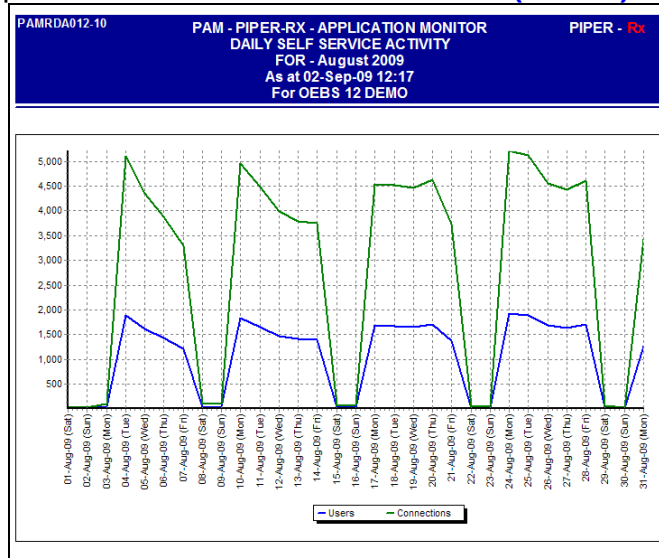
[PAMRDA012 SS Connections \(month\)](#) both charts and lists the total self-service activity per day displaying all data points for a given month.

[PAMRDA013 SS Page Requests](#) both charts and lists the total page requests activity per day displaying all data points in the *PAM* daily activity repository.

[PAMRDA014 SS Page Requests \(month\)](#) which both charts and lists the total page requests activity per day displaying all data points for a given month.

**Note:** The concurrent program FNDOAMCOL which collects data for the OAM (Oracle Application Manager) product generates one self-service connection and in some cases two page request per run. The program FNDOAMCOL runs once every 10 minutes (default) so will generate approximately 144 self-service sessions for the day and 288 page requests.

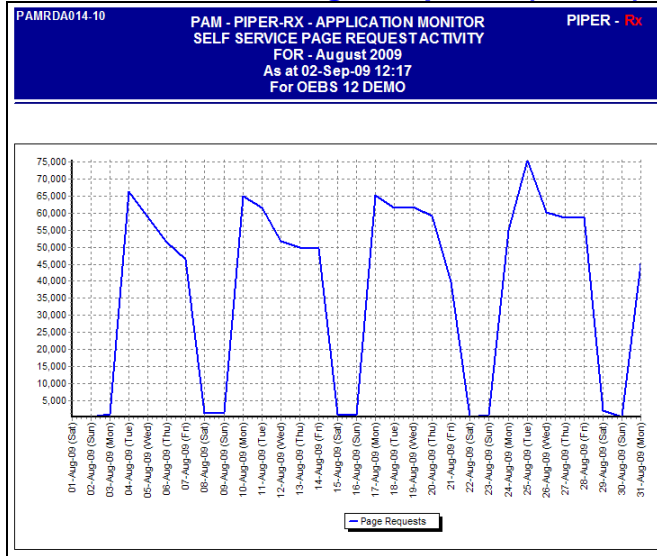
Example **PAMRDA012 SS Connections (month)** - Page 1



Example **PAMRDA012 SS Connections (month)** - Page 2

Sample Time	Users	Connections	Total (Hrs)	Average (Min)	STDDEV (Min)
31-Aug-09 (Mon)	1,268	3,423	9,508	58	12
30-Aug-09 (Sun)	10	28	78	1	0
29-Aug-09 (Sat)	26	70	194	2	0
28-Aug-09 (Fri)	1,704	4,602	12,783	77	17
27-Aug-09 (Thu)	1,639	4,424	12,289	74	16
26-Aug-09 (Wed)	1,687	4,554	12,650	76	17
25-Aug-09 (Tue)	1,894	5,115	14,208	86	19
24-Aug-09 (Mon)	1,930	5,210	14,472	87	19
23-Aug-09 (Sun)	27	72	200	2	0
22-Aug-09 (Sat)	17	47	131	1	0
21-Aug-09 (Fri)	1,380	3,726	10,350	63	14
20-Aug-09 (Thu)	1,714	4,627	12,853	78	17
19-Aug-09 (Wed)	1,650	4,456	12,378	75	16
18-Aug-09 (Tue)	1,672	4,514	12,539	76	16
17-Aug-09 (Mon)	1,682	4,542	12,617	76	17
16-Aug-09 (Sun)	32	87	242	2	0
15-Aug-09 (Sat)	23	63	175	2	0
14-Aug-09 (Fri)	1,390	3,752	10,422	63	14
13-Aug-09 (Thu)	1,406	3,795	10,542	64	14
12-Aug-09 (Wed)	1,474	3,979	11,053	67	14
11-Aug-09 (Tue)	1,656	4,472	12,422	75	16
10-Aug-09 (Mon)	1,835	4,955	13,764	83	18
09-Aug-09 (Sun)	39	105	292	2	0
08-Aug-09 (Sat)	36	98	272	2	0
07-Aug-09 (Fri)	1,219	3,291	9,142	55	12
06-Aug-09 (Thu)	1,437	3,881	10,781	65	14
05-Aug-09 (Wed)	1,614	4,359	12,108	73	16
04-Aug-09 (Tue)	1,889	5,100	14,167	85	19
03-Aug-09 (Mon)	38	102	283	2	0
02-Aug-09 (Sun)	13	36	100	1	0
01-Aug-09 (Sat)	11	30	83	1	0

Example **PAMRDA014 SS Page Requests (month)** – Page 1



Example **PAMRDA014 SS Page Requests (month)** – Page 2

Sample Time	Users	Connections	Total (Hrs)	Page Requests
31-Aug-09 (Mon)	1,268	3,423	9,508	45,041
30-Aug-09 (Sun)	10	28	78	324
29-Aug-09 (Sat)	26	70	194	2,140
28-Aug-09 (Fri)	1,704	4,602	12,763	58,764
27-Aug-09 (Thu)	1,639	4,424	12,289	58,695
26-Aug-09 (Wed)	1,687	4,554	12,650	60,047
25-Aug-09 (Tue)	1,894	5,115	14,208	75,455
24-Aug-09 (Mon)	1,930	5,210	14,472	54,800
23-Aug-09 (Sun)	27	72	200	761
22-Aug-09 (Sat)	17	47	131	455
21-Aug-09 (Fri)	1,380	3,726	10,350	39,738
20-Aug-09 (Thu)	1,714	4,627	12,853	59,077
19-Aug-09 (Wed)	1,650	4,456	12,378	61,770
18-Aug-09 (Tue)	1,672	4,514	12,539	61,401
17-Aug-09 (Mon)	1,682	4,542	12,617	65,068
16-Aug-09 (Sun)	32	87	242	751
15-Aug-09 (Sat)	23	63	175	1,233
14-Aug-09 (Fri)	1,390	3,752	10,422	49,618
13-Aug-09 (Thu)	1,406	3,795	10,542	49,754
12-Aug-09 (Wed)	1,474	3,979	11,053	51,606
11-Aug-09 (Tue)	1,656	4,472	12,422	61,397
10-Aug-09 (Mon)	1,835	4,955	13,764	64,956
09-Aug-09 (Sun)	39	105	292	1,686
08-Aug-09 (Sat)	36	98	272	1,356
07-Aug-09 (Fri)	1,219	3,291	9,142	46,339
06-Aug-09 (Thu)	1,437	3,881	10,781	51,336
05-Aug-09 (Wed)	1,614	4,359	12,108	58,893
04-Aug-09 (Tue)	1,889	5,100	14,167	66,299
03-Aug-09 (Mon)	38	102	283	1,100
02-Aug-09 (Sun)	13	36	100	386
01-Aug-09 (Sat)	11	30	83	376

**Sessions per User**

If the user count is low and the session count is high, this indicates the users are continually reconnecting to the application throughout the day which, in a self-service sense is a good thing if it is customers logging in and out of your application. ☺

## Average session time

If the average session time is low this indicates the users are performing short sharp sessions - login, do what they need and log out.

One of the things to look out for are session times greater than 5+ hours; this could indicate that users are coming in at 9:00 and not logging out when they go to lunch. For just basic security, users should log out when they are not using the application or at least lock their screens.

## Page requests

The page requests may give you an indication as to the impact your OEBS application is inflicting on your web services infrastructure.

If the web services infrastructure is shared with other applications you may be able to prove that a performance issue was not related to additional load from the OEBS application by showing normal page request activity.

You may also be able to estimate through experience that less than **X** thousand page requests does not cause an issue with the web services infrastructure but where the number increases to **y** thousand the overall web services infrastructure experiences issues. You can then use trending information to show when your volume of page requests reaches a threshold requiring additional web services infrastructure.

## 3.4 Application response time

One of the biggest issues with measuring and reporting response time is that the item/s generally being measured by the technical teams are technical and the response time quoted usually in milliseconds, this often means absolutely nothing to the business user when they are waiting seconds or minutes for a response from a “slow” system at their busiest business activity times.

If you ever make the mistake of telling the business that the performance issue was related to “a 20 millisecond delay at the database connect process” at best you will watch their eyes fog over and at worst they will throw something at you for speaking “technobabble”. What you really need when trying to measure response time is a “transaction” that is at least vaguely meaningful to the non-technical user. For example: “We did notice that the standard concurrent request we use for measuring response time did increase at 2:30pm and we are looking into it.”

It is all about making non-technical business user comfortable that you understand their imperatives can talk to them in non-geek speak.

In a later tutorial when we introduce the intraday response time collector and associated alert, we will go into response time in more detail.

In the meantime, we will be collecting overall response time statistics for each day which should provide you some very interesting statistics over time.

### So what does *PAM* measure?

*PAM* measures the runtime of the concurrent program FNDOAMCOL. We specifically chose this program as:

- ❖ This concurrent program runs every 10 minutes as part of the Oracle OAM (Oracle **A**pplication **M**anager) module. It's so important to OAM it has its own concurrent manager.
- ❖ Consistency: The concurrent program performs the same tasks on the same application objects each time it runs, therefore, its overall performance (run time) is dependent on both the size and health of the tables and indexes it accesses and any other activity that is occurring at the time the program run
- ❖ It is a measure that is easily understood by the end user, "a concurrent program"
- ❖ It generally has a minimum runtime of 2 to 3 seconds which is meaningful to the end user

This response time measure only reflects the overall activity of the OEBS application, from my perspective as an apps admin, when dealing with end users that's all the information I want.

The *PAM* daily activity collector reviews all the FNDOAMCOL runs for the prior day and stores the statistics for that day's activity in the *PAM* daily activity repository.

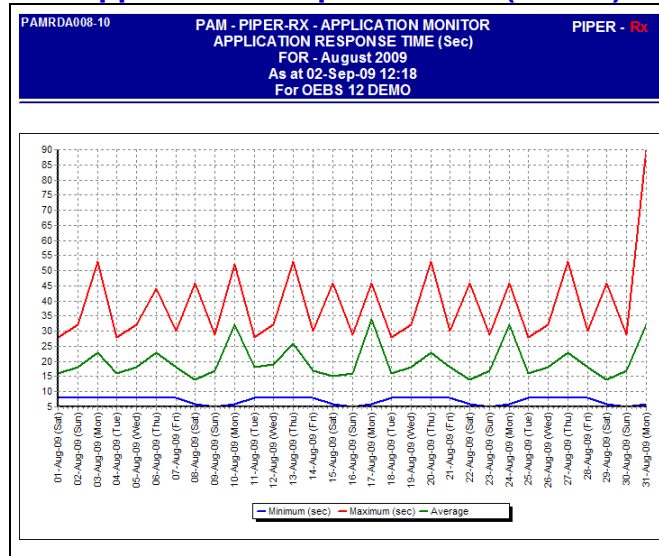
*PAM* provides two (2) reports for displaying the daily application response time information:

[PAMRDA007 Application Response Time](#) both charts and lists the application response time per day displaying all data points in the *PAM* daily activity repository.

[PAMRDA008 Application Response Time \(month\)](#) both charts and lists the application response time per day displaying all data points for a given month.

**Note:** In a later *PAM* tutorial we will be measuring and recording intraday application response time.

Example PAMRDA008 Application Response Time (month) report – Page 1



Example PAMRDA008 Application Response Time (month) report – Page 2

Sample Time	Samples	Minum	Average	Maximum	STDEV
31-Aug-09 (Mon)	144	6	32	90	25.0
30-Aug-09 (Sun)	144	5	17	29	6.0
29-Aug-09 (Sat)	144	6	14	46	12.0
28-Aug-09 (Fri)	144	8	18	30	7.0
27-Aug-09 (Thu)	144	8	23	53	12.0
26-Aug-09 (Wed)	144	8	18	32	7.0
25-Aug-09 (Tue)	144	8	16	28	10.0
24-Aug-09 (Mon)	144	6	32	46	25.0
23-Aug-09 (Sun)	144	5	17	29	6.0
22-Aug-09 (Sat)	144	6	14	46	12.0
21-Aug-09 (Fri)	144	8	18	30	7.0
20-Aug-09 (Thu)	144	8	23	53	12.0
19-Aug-09 (Wed)	144	8	18	32	7.0
18-Aug-09 (Tue)	144	8	16	28	10.0
17-Aug-09 (Mon)	144	6	34	46	25.0
16-Aug-09 (Sun)	144	5	16	29	6.0
15-Aug-09 (Sat)	144	6	15	46	12.0
14-Aug-09 (Fri)	144	8	17	30	7.0
13-Aug-09 (Thu)	144	8	26	53	12.0
12-Aug-09 (Wed)	144	8	19	32	7.0
11-Aug-09 (Tue)	144	8	18	28	10.0
10-Aug-09 (Mon)	144	6	32	52	25.0
09-Aug-09 (Sun)	144	5	17	29	6.0
08-Aug-09 (Sat)	144	6	14	46	12.0
07-Aug-09 (Fri)	144	8	18	30	7.0
06-Aug-09 (Thu)	144	8	23	44	12.0
05-Aug-09 (Wed)	144	8	18	32	7.0
04-Aug-09 (Tue)	144	8	16	28	10.0
03-Aug-09 (Mon)	144	8	23	53	12.0
02-Aug-09 (Sun)	144	8	18	32	7.0
01-Aug-09 (Sat)	144	8	16	28	10.0

3.4.1 What to do with the Application response time information

Let's start with the minimum values. As the daily data is based on a full 24 hours, the minimum response time should reflect those times when there is little to no activity in the application, so this is the minimum base line. If we had no users this is the type of performance we can expect. We should be so lucky...

In most sites we should also see a decrease in the response time for weekends with lower user activity.

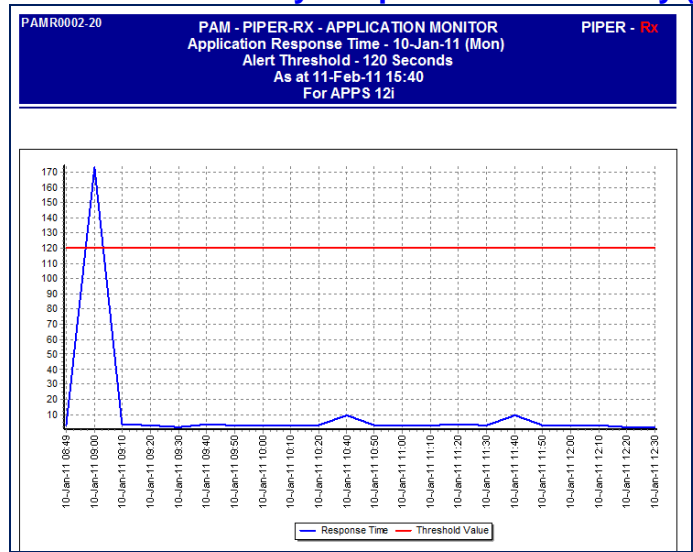
The maximum for the day reflects the worst performance for that day.

The average and standard deviations provide an insight into the overall activity for the day.

**Note: A simple uptime calculation:** The FNDOAMCOL program runs every 10 minutes which is approximately 144 times per 24 hour period. If the concurrent managers were down or the database was down for any period the FNDOAMCOL program will not have run, thus the number of samples will be less than the normal 144. Each missing sample equates to 10 minutes. Three missing samples equates to 30 minutes down time.... I did say it was simple.....

If you have a particularly interesting day and you want more details as the intraday response time for that day you can use the **PAM** reports [PAMRGA002 Intraday Response Time History \(day\)](#) report:

Example [PAMRGA002 Intraday Response Time History \(day\)](#) report



In a later **PAM** tutorial we will cover how **PAM** collects, stores and alerts its own intraday activity data.

## 4 Daily activity comments

Have you ever gone back and looked at some application reports which show an increase or decrease in activity or performance but cannot exactly remember what may have caused that change in activity profile?

**PAM** provides a simple method of recording some items that may affect the overall performance these may include such comments as:

- Merged with company XYZ
- Added 200 new application accounts
- End dated 40 application accounts
- HR went live
- Consolidated the workflow background processes
- Added a new FAST concurrent manager to speed up the manager throughput
- Family pack H added

It is worth taking the time to add comments as when you look back over the past year and your manager wants to know what caused the performance decrease for the month of July – you at least will know what may have contributed to it or the applications administrator who comes in after you will know what has been happening. Or, you may just want to enter when you went on and returned from leave so you can show you were not responsible for any issues whilst you were not there. ☺

Adding an annotation (comment) can be done using the following **PAM** API:

```
BEGIN
  piper_rx_pam_api.da_add_activity_comment
    ( '01-Aug-09',           -- Comment Date
      'Free format text comment'); -- Free format text
COMMIT;
END;
```

Parameter 1:

Comment date – Format DD-MON-YY (not cases sensitive).

Parameter 2:

Free format text, any input will be truncated to 100 characters

**PAM** provides a report for daily activity comments. [PAMRDA015 Daily Activity Comments](#) lists the comments added by comments date as per the example below:

Example **PAMRDA015 Daily Activity Comments** report

PAMRDA015-10		PAM - PIPER-RX - APPLICATION MONITOR		PIPER - Rx	
		DAILY ACTIVITY COMMENTS			
		As at 02-Sep-09 12:26			
		For OEBS 12 DEMO			
Comment Date	Comment				
29-Jul-09 (Wed)	Added 200 application users				
24-Jun-09 (Wed)	Family patch H applied				
23-Jun-09 (Tue)	Consolidated Workflow background process runs				
22-Jun-09 (Mon)	Company X added 200 new user accounts				
21-Jun-09 (Sun)	PIPER-Rx PAM Installed				

**PAM** does not provide a method for editing or deleting comments, any comment editing and or deleting can be done via the TOAD<sup>®</sup> schema browser.

## 5 Clone protection feature

Scenario: You have **PAM** running on instance 1 and you clone that instance to create instance 2. What would normally happen is that **PAM** would continue to run on instance one and two with the same customer name sending e-mail alerts from both instances.

In order to prevent this scenario from occurring, at the beginning of each **PAM** collection run **PAM** checks if either the Instance SID or Host values have changed, if they have, **PAM** will on the cloned instance:

- ❖ Remove the **PAM** collector and e-mail DBMS jobs
- ❖ Prefix the word CLONE to the **PAM** customer name
- ❖ Set the **PAM** settings reference for PAM\_TARGET\_HOST to "HOST\_NAME"
- ❖ Set the **PAM** settings reference PAM\_TARGET\_INSTANCE to "INSTANCE"
- ❖ Set the background e-mail colour to white

This will prevent **PAM** from running on the cloned instance even if the two **PAM** DBMS jobs on the clone instance are restated.

Where an instance is part of a failover cluster or virtual instance the clone check is prone to reporting incorrectly and shutting **PAM** down on the instance. To prevent this we have added a **PAM** setting PAM\_TARGET\_VIRTUAL\_FAILOVER which determines if the instance is part of a failover cluster or a virtual environment. (Default = 'Y' The instance is part of a failover cluster or a virtual environment). Based on this value **PAM** changes the method of checking for a cloned instance.

The **PAM** setting value can be changed using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_CLONE_VIRTUAL_FAILOVER ( 'Y' );
```

Valid parameters are:

- Y = the instance is part of a failover instance or is a virtual instance
- N = the instance is not virtual and is not part of a failover instance

## 5.1 PAM clone e-mail alert

Where a clone is detected **PAM** will send an alert e-mail to indicate a clone instance has been detected and that **PAM** has been shut down on that instance

### Example **PAM** clone e-mail alert message

**ALERT MESSAGE FROM *PAM* - PIPER-Rx Application Monitor - DO NOT REPLY**

**CLONE DETECTED - CLONE DETECTED**  
 Customer = Company name  
 Site = Site name  
 Message Route = DEF

From = abc@abc.com.au  
 To = pam@piper-rx.com  
 Cc = None Set

Alert Time: 11-Feb-11 (Fri) 17:48

---

**CLONE INSTANCE DETECTED - PAM HAS BEEN SHUTDOWN ON THIS INSTANCE - DO NOT REPLY**

**Note:** For this e-mail alert the background colour has been (hard coded) to pink in order to make it stand out.

### How to restart **PAM** in the cloned instance

1. The **PAM** API shown below will:
  - Set the **PAM** settings reference for **PAM\_TARGET\_HOST** to the current host name
  - Set the **PAM** settings reference **PAM\_TARGET\_INSTANCE** the current instance
  - Change the customer name to the value entered in the API call

```
exec PIPER_RX_PAM_API.PAM_CLONE_RESET ('CUSTOMER NAME GOES HERE');
```

2. Restart both the **PAM** e-mailer and collector DBMS jobs

The **PAM** e-mailer job is the first of two **PAM** jobs. Its primary function is to identify if any alerts have been generated and to send those alerts to the defined e-mail recipient. The e-mailer program is designed to be run on a 5 or 10 minute frequency.

**PAM** uses an Oracle DBMS job to run the **PAM** e-mailer program. To create the job runs the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_EMAIL_JOB_ADD (10);
```

The procedure parameter is the number of minutes between runs - 10 = every 10 minutes.

It would be a good idea to send a test e-mail using the following to make sure the e-mail settings are not different to the parent instance:

```
exec PIPER_RX_PAM_SENDMAIL.send_test_email ( 'DEF' );
```

Where 'DEF' is the default email route.

The **PAM** collector is the heart of the **PAM** application; its primary function is to identify which of the individual **PAM** checks need to be run and to run them. The collector program is designed to be run on a 5 or 10 minute frequency.

**PAM** uses an Oracle DBMS job to run the **PAM** collector program. To create the collector job run the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_COLLECTOR_JOB_ADD (5);
```

The **PAM** API parameter is the number of minutes between runs - 5 = every 5 minutes.

### 3. Reset the e-mail background colour

**PAM** email background colors can be changed using the following **PAM** API setting the argument to any valid HTML colour code:

```
exec PIPER_RX_PAM_API.PAM_EMAIL_BG_COLOUR_SET ( '#CCFFFF' );
```

**Note:** **PAM** alert e-mails use Red (#FF0000), Orange (#FF9900) and Blue (#0000FF) so please try and pick complementary colours

The following are some HTML colour codes that could be used. (There is any number of HTML colour charts available on the web):

- ❖ #FFFFFF White (best choice if you plan to print the e-mails)
- ❖ #CCFFFF Light blue
- ❖ #CCFF99 Light green
- ❖ #FFCCCC Light pink
- ❖ #FFFF99 Light yellow

## 6 PAM Sign-on Audit Level Change Alert (GA-001)

**PAM** Alert ID GA-001 is one of the simplest alerts to get us started with **PAM** monitoring. This alert is run once per day (default) and designed to let you know if the sign-on audit level has changed.

It is important to know of any changes in the sign-on audit level as you may start collecting additional information in the OEBS applications sign-on audit tables and will need to watch the space usage. Also it may mean you have additional indexes to monitor and include in your regular index re build program.

In the case where the sign-on audit level has been set to a lower level you may not be collecting valuable audit level information.

During install, **PAM** will set the sign on-audit threshold to your site's current level. You can view the current application level using the **PAM** config report **PAMC001b PAM Config (thresholds)**:

Example **PAMC001b PAM Config (thresholds)** report

Alert ID	Alert Description	Alert Status	Alert Frequency	Alert Action
CR-003	Alert when the number of Total Completed Requests exceeds the threshold Number of completed requests per day	Active	30 Minutes	CR-003
CR-004	Alert when the number of Completed Warning Requests exceeds the threshold Number of completed warning requests per day	Active	30 Minutes	CR-004
<b>DB - Database Alerts</b>				
DB-001	Alert when there are new application related Invalid Objects	Active	4 Hours	DB-001
DB-002	Index rebuild reminder	Active	1 Month	DB-002
DB-003	Alert when a tablespace free space drops below X pct or datafiles are added Minimum pct free space	Active	1 Day	DB-003
<b>GA - General Application</b>				
GA-001	Alert when the sign-on audit level is not set to defined level Sign-on audit level	Active FORM	1 Day	GA-001
GA-002	Alert application is in maintenance mode	Active	5 Minutes	GA-002

Once per day (default) **PAM** will check the site sign-on audit level against the **PAM** reference value stored in the **PAM** config table **piper\_rx\_pam\_config**

When **PAM** detects a change in the sign-on audit level **PAM** will:

- ❖ Send an alert e-mail notifying of the change
- ❖ Reset the **PAM** threshold value to the changed value so you will be notified if the value changes again

## Example **PAM** Sign –on audit level e-mail alert message (GA-001)

**ALERT MESSAGE FROM PAM - PIPER-Rx Application Monitor - DO NOT REPLY**

Customer = Company name  
Site = Site name  
Alert Level = **Warning**  
Detected = 11-Feb-11 (Fri) 08:22:01  
Alert Frequency = 1 Day

---

The sign-on audit level of (USER) has changed from the PAM reference value of FORM

---

**Alert Information:**

**GA-001 - Sign-on audit level**

**THE SIGN-ON AUDIT LEVEL HAS CHANGED FROM THE DEFINED LEVEL**

This is a proactive alert to inform you the sign-on audit level has been changed. **PAM** has as part of this alert has reset the **PAM** reference value to the new site value.

It is generally good practice to set the audit level to the highest level of FORM which allows the most detailed level of full service user activity audit

**Note 1:** In order for the **PAM** application to monitor full service user activity the sign-on audit level needs to be set to a minimum level of RESPONSIBILITY

If you change the sign-on audit level and want to reset the **PAM** reference value you can change the **PAM** sign-on audit level reference using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_SIGN_ON_AUDIT_LEVEL ( 'R' );
```

Valid values for this **PAM** API are:

- ❖ N = NONE
- ❖ U = USER
- ❖ R = RESPONSIBILITY
- ❖ F = FORM

## 6.1 Changing the check frequency

The check frequency can be changed using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_CHANGE_ALERT_FREQUENCY ('GA-001', 2, 'HR');
```

In the above example the GA-001 check will be set to run every 2 hours.

The list of valid values for the alert frequency units can be found using the **PAM** config report **PAMC003 PAM Lookups** - Lookup type "FREQUENCY UNITS".

A list of alert check frequencies can be found by using the following **PAM** config report **PAMC001c PAM Config (frequency)**.

**Note:** This method can be used to change any or all **PAM** alert check frequencies.

## 7 The **PAM** Heartbeat (HB-001) e-mail

The **PAM** heartbeat is designed be generated on a regular basis (default once per day) to let you know the **PAM** application is running and to inform you of any error.

### Example **PAM** Heartbeat e-mail alert message

**PAM HEART BEAT - DO NOT REPLY**

Customer = Company name

Site = Site name

---

Heart beat time: 11-Feb-11 (Fri) 05:00:31

Prior heart beat time: 10-Feb-11 (Thu) 05:00:41

Heart beat period (DD:HH:MM): 1:0:0

Heart beat frequency: 1 Day

Alerts since last heart beat:

Informational = **6**

Warning = **0**

Critical = **0**

---

The **PAM** heartbeat email will show the number of alerts generated between heartbeat e-mails.

If **PAM** encountered any errors running the **PAM** application, these errors will be reported in the heartbeat e-mail. Where there are more than 5 (value hard coded) errors to report, **PAM** will direct you to the **PAM** config report **PAMC004 PAM Errors** which will display all errors encountered.

## 7.1 How to change the heartbeat e-mail check frequency

The heartbeat e-mail frequency can be changed using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_CHANGE_ALERT_FREQUENCY ('HB-001', 2, 'HR' );
```

In the above example, the heartbeat e-mail (HB-001) run will be set to run every 2 hours.

The list of valid values for the alert frequency units can be found using the **PAM** config report [PAMC003 PAM Lookups](#) - Lookup type “FREQUENCY UNITS”.

A list of alert check frequencies can be found by using the following **PAM** config report [PAMC001c PAM Config \(frequency\)](#).

**Note:** This method can be used to change any or all **PAM** alert check frequencies.

## 8 The **PAM** Sign-on audit (IN-006) reminder

In the most current releases of OEBS both full service and self-service connections are recorded in the OEBS applications [fnd\\_logins](#) table (assuming the sign-on audit level is set to user or above) however, the information recorded by OEBS that differentiates a full and self-service connection often changes between OEBS releases. To maintain OEBS version independence **PAM** distinguishes full service connections as a session that has accessed an application full service responsibility and as such **PAM** requires the application sign-on audit level set to either “responsibility” or “form” to record full service activity.

If the sign-on audit level is not set to either “responsibility” or “form”, once per month **PAM** will send a reminder message alerting you to the fact that **PAM** will not be collecting full service user activity information.

### Example **PAM** Sign-on audit level reminder e-mail alert message (IN-006)

ALERT MESSAGE FROM **PAM - PIPER-Rx** Application Monitor - **DO NOT REPLY**

Customer = Company name  
 Site = Site name  
 Alert Level = **Informational**  
 Detected = 11-Feb-11 (Fri) 05:01:01  
 Alert Frequency = 1 Month

The current sign-on audit level of (USER) is insufficient for PAM to collect Full Service user activity information

**Alert Information:**

IN-006 - **PAM** Sign-on audit level reminder

**THE CURRENT APPLICATION SIGN-ON AUDIT LEVEL WILL PREVENT PAM FROM COLLECTING FULL SERVICE USER ACTIVITY DATA**

This alert is a **PAM** internal reminder that the current application sign-on audit level will prevent **PAM** from collecting full service user activity data

In order for the **PAM** application to monitor full service user activity the sign-on audit level needs to be set to a minimum level of **RESPONSIBILITY**

It is generally good practice to set the audit level to the highest level of **FORM** which allows the most detailed level of full service user activity audit

**Note 1:** If you want to prevent this alert from continuing to alert refer to the FAQs for more information

## 8.1 Turning off the PAM sign-on audit alert

In the instance where you do not want your application's sign-on audit level set to either "responsibility" or "form" you may want to turn off this monthly alert. The **PAM** sign-on audit level reminder (IN-006) alert can be turned off using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_ALERT_ENABLE ( 'IN-006', 'N' );
```

Parameter 1 – The **PAM** alert to be enabled / disabled

Parameter 2 – The alert activation value - enable = Y disable = N

**Note:** This method can be used to turn off any or all **PAM** alert check frequencies.

Should you want to enable the alert at a later date you can use the same **PAM** API setting the second argument to 'Y' (enable).

## 9 Monthly OEBS Activity Report

Now that you are using **PAM** and getting loads of useful OEBS application activity information you might as well use the information to advance the cause of proactive OEBS Application Management. The best Systems Administrators have done this for years using their own scripts and report templates – now you too can do this both fast and efficiently with **PAM**, probably in around half an hour per month including all commentary.

The great advantage of generating a monthly OEBS application activity report is it enables you to be on top of the game... and where you can show at least 6 months rolling history in the report, it's even better! Through capturing and reporting a growing, tangible, fact base on OEBS Application activity and performance, it allows you the opportunity to have sensible, fact based discussions with the business on system performance, growth, issues and capacity planning matters.

It's all about putting yourself in a position to be proactive by having relevant information and facts in a way that can be readily understood by the business; most notably your Shared Service Manager, General Accounting Manager and Systems Accountant. Others further up the chain will be interested periodically too, but probably only when there are issues, which is an excellent time to add to your credibility with concise, fact based reports!

Also, once **PAM** has been collecting data you will no longer have to worry about short purge cycles potentially removing the history you wish you had for reporting. **PAM** will have the daily summary data you need.....another great reason for using PAM!

**BONUS:** If you stick with **PAM**, subsequent tutorials will cover **workflow** which is often one of the bigger performance management issues facing OEBS sites. **Workflow will be added to the PAM monthly report** at that point enabling you to proactively report on the many unaddressed issues usually associated with workflow.

**To get started on producing a monthly report all you need to do is work through the **PAM** monthly report documents you downloaded. There is a template report, detailed guidelines and a worked example to make things super easy. So go on, give it a go!!**

**Note:** Feel free to make changes to the format and content of the monthly report - but remember, keep it simple as you will be producing one of these reports every month... ☺

## 10 Known issues

### Issue 1 – Insufficient On-line history

During the install process **PAM** uses the content of the following standard purged objects to report on historical daily activity (max 31 days):

- ❖ [applsyst.fnd\\_concurrent\\_requests](#) ( purge concurrent requests)
- ❖ [applsyst.fnd\\_logins](#) ( Purge sign-on audit data)
- ❖ [icx.icx\\_sessions](#) ( purge self-service sessions)

If you have any purging program affecting the above mentioned objects that purges all or part of the application's on-line activity history, the collected historical data collected on install may understate the actual activity.

### Issue 2 – No concurrent manager activity found

I have come across some sites that purge all their concurrent requests on a daily basis. If this is the case, the **PAM** concurrent request activity may report no activity.

### Issue 3 – Dead and long running full service sessions

The primary issue with recording total connect time is that some sessions do not get closed (dead sessions) or the user leaves the session active overnight. In the case of dead sessions, these will be recorded as zero (0) connect time. Sessions that have been left running for long periods of time will be limited to the lesser of the actual session time or 12 hours regardless the actual session time.

This will affect values for:

- ❖ [fs\\_total\\_connect\\_time](#)
- ❖ [fs\\_avg\\_connect\\_time](#)
- ❖ [fs\\_stddev\\_connect\\_time](#)

### Issue 4 – Insufficient self-service on-line history

If the default self-service purge program provided with 11i is being run (deletes all but last 4 hours activity) the values for the following self-service statistics will be incorrect:

- ❖ [ss\\_users](#)
- ❖ [ss\\_total\\_connect\\_time](#)
- ❖ [ss\\_avg\\_connect\\_time](#)
- ❖ [ss\\_stddev\\_connect\\_time](#)
- ❖ [ss\\_page\\_requests](#)

The **PAM** daily activity collector is set to run in the morning (5 am default) reporting on the prior day's activity so you should hold a minimum of 48 hours history self-service history on-line for **PAM** to report accurate self-service activity.

### **Issue 5 – Sign-on audit level not set to Responsibility or Form**

If your site's sign-on audit level is not set to either "responsibility" or "form" there will be no full service activity information available for **PAM** to populate the daily activity repository.

## 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*