

## ***PAMtutorials* 2: Performance profiling**

**“Your OEBS Application - Surviving performance pressure (business peaks survival tips)”**

**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</i> 2 .....	3
1.1	OEBS application performance profiling .....	3
1.2	Workflow background process management .....	3
1.3	<i>PAM</i> e-mail alerts .....	4
2	OEBS application performance profiling .....	5
2.1	How do I get a good measure of application performance? .....	5
2.2	How <i>PAM</i> monitors application performance and what <i>PAM</i> measures 7	7
2.3	PF-001 - Application Performance Indicator e-mail alert .....	8
2.4	What to do with <i>PAM</i> information on response time .....	12
3	Workflow background process management .....	13
3.1	Why have a <i>PAM</i> workflow background process check? .....	14
3.2	A "disaster" study and what to do about it... ..	14
3.3	What the <i>PAM</i> monthly workflow background process check does ..	17
3.4	About the <i>PAM</i> WF-001 - Excessive Workflow Background Processes check18	18
3.5	WF-001 - Excessive Workflow Background Processes e-mail alert..	19
4	Additional ideas on how to make the most of <i>PAMtutorials</i> 2 learnings and information .....	21
5	Disclaimer .....	22

# 1 What you'll get out of PAMtutorials 2

## 1.1 OEBS application performance profiling

The skilled applications administrator understands the OEBS application's performance profile and then develops strategies to manage it. In this software / tutorial pack you will learn...

- What is your application performance profile and how does it vary over time
- How is your application performance profile affected by cyclical business peaks
- How to identify immediately if a performance issue is application related
- How to get data for response time performance that means something to the business and can be used in all those difficult discussions about "its slow....."

All of this will be covered simultaneously whilst learning generally about **PAM's** performance profiling capabilities including **PAM's** PF-001 Application Performance Indicator e-mail alert and three **PAMreports** showing the application's response time.

## 1.2 Workflow background process management

Workflow background processes can tie your system in knots if you let them but managing workflow background processes is made much easier using **PAM**. Do you have too many running?

In this tutorial we cover the **PAM** check to see how many workflow background processes your application is running per day. **PAM** e-mail alert WF-001 - Excessive Workflow Background Processes will notify you when the number of background processes run per day exceeds your daily threshold.

The tutorial will explain the workflow background process in detail. Through the use of examples and other reference material (see paper on [The hidden cost of Workflow](#) at the [www.piper-rx.com](http://www.piper-rx.com) website) you will be tutored through the typical opportunities to improve workflow background process management.

### 1.3 *PAM* e-mail alerts

During this tutorial we will also explain *PAM* e-mail alerts by providing examples of e-mail alerts that are generated when key *PAM* alert thresholds have been breached. You also learn the ins and outs of what the alert means and what to do about it.

## 2 OEBS application performance profiling

### 2.1 How do I get a good measure of application performance?

This is a biggie, something that everyone who comes into contact with the application has an opinion on! Unfortunately, there is still no simple measure that is easily understood by the application end user (e.g. Fred from Accounts).

The “Holy Grail” of end to end such as it is today, provides so much detail (usually presented in technical terms / format) technical staff try and use it when explaining to the business why the application is slow. IMHO - End to End in most cases is whatever information a vendor can collect which, while it has some intrinsic value for the IT group (root cause analysis etc.) it is usually far too technical to be communicated to the end user. You can almost see Fred’s eyes fog over when you try and explain the application is slow because the database segment of the “transaction” was taking 300 milliseconds when it normally takes 20. Most of the terminology means nothing to your typical business user – they just get angry at such a useless explanation.

Another issue with end to end is actually deciding what constitutes end to end: A connection to the application, a full transaction like creating a purchase order... you get the picture – its very different depending on your viewpoint! In most cases each individual thinks they have the most appropriate view on this and much miscommunication can then ensue.

In nearly all cases we tend to lose sight of the basic user’s statement “it’s slow” - a simple statement that deserves a simple response but is often instead followed up by IT with a frustrating, often meaningless detailed technical answer.

#### **Not all performance problems are caused by poor performing code.**

Our catch cry with **PAM** is that not all performance problems are caused by poor performing code. In fact it is our view that in a stabilised environment, the majority of performance issue are caused by inappropriate user behaviour, particularly during high processing times when the users are stressed and under pressure.

What we see time and time again is that when there is a performance issue IT go looking for poor performing code – which in most cases, in OEBS you cannot change. ***Our experience says you should check what the users are doing to your application first.***

Even if your monitoring tool collects bind variables e.g. General Ledger [code\\_commination\\_id](#) as part of a general ledger transaction, you still have to

have enough functional knowledge to translate the value into a Chart of Accounts code set, and then determine if the Chart of Accounts value was a large range (which drives slower performance than a small range).

...or was the issue a user who wanted to run a trial balance for one area of one division and instead entered the wrong code segments and has run the trial balance for a large number of divisions. Oops I did not mean to do that... but it happens time and time again.

So the moral of the story is, before you go looking for poor performing SQL, check what your users are doing to you first.

### **What is a transaction?**

This is a question is one that can be debated ad nauseum and everyone has an opinion. Some of the key issues are covered in the paper [“A business approach to Oracle E-Business Suite response time, E-to-E and SLAs”](#) which is posted in the Papers section of the **PIPER-Rx** website (It’s a bit tongue in cheek but I like to think it gets the point across!)

### **The user says “It’s slow...”**

This raises one of the greatest sources of user / IT communication frustration that exists today. Let’s get back to basics. What we need to do is start with the end user statements like:

- ❖ “Its slow”
- ❖ “This stupid system doesn’t work”

What often happens next tends to frustrate the business users, as the technical people now want to record database activity down to the lowest level and communicate all this great information back to the user. Not that there is any thing wrong with this information per se, it is just that it doesn’t help the user understand cause and effect in a way they can do anything about so they get very frustrated.

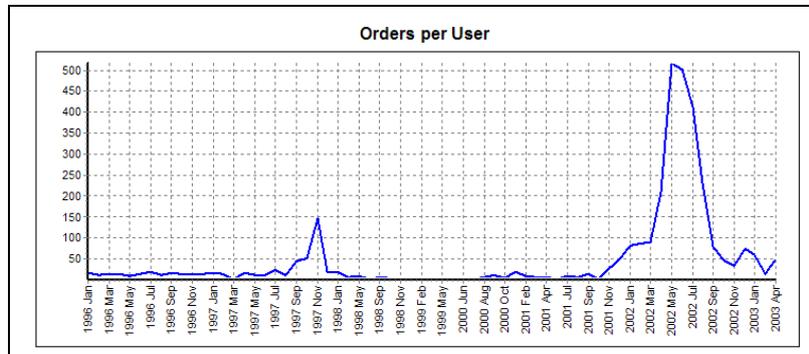
Effectively they don’t want a long and involved technical answer. All the techie stuff is for IT and should be kept there. It is a non technical statement that requires a non technical answer:

- ❖ “No it is not” (and I have the evidence)
- ❖ Yes it was and we are looking into it

### **A user under stress will complain**

You should bear in mind performance complaints can be due to frustration by the end user. If they have too much work to do or a tight deadline, they will often complain about application performance.

To illustrate, the following is an example of a report I wrote back in 2006 showing the number of orders being entered per user:



Note the increase in activity for May and July. A three to five second delay in the orders screen (which is normal) is acceptable when you are doing 200 orders per day, but it becomes a problem if you have to do 500 per day.

As the company grows the transactional volume increases and we tend to save money by maintaining or reducing staffing levels.....ergo increased user work load.

This is just another example of not all performance issue are technical.

## 2.2 How PAM monitors application performance and what PAM measures

PAM takes a very simple approach to application performance from an applications administrator perspective by using a performance measure that will provide a notification if the application has slowed down **that would be understandable to the end user**. (We have assumed there is already some “all seeing all knowing” form of technical end to end monitoring in place measuring all the technical stuff for the technical purist.)

PAM measures the run time of the concurrent program FNDOAMCOL. This is a great program. FNDOAMCOL is required to be run for the Oracle OAM product (Default every 10 minutes) and it is so important it even has its own dedicated concurrent manager. The program performs the same processes every time it runs (consistency) and it generally has a minimum (baseline) runtime of 2-3 seconds thus, when its run time slows down, it will be as a result of other activity occurring in the application at the time.

If the application is slow and the response time is ok then it is not an application based issue, it's the network etc...☺... If the response time is slow then it is the application or database.

So by reviewing the **PAM** response time output you can provide the simple answer to the statement "it's slow":

- ❖ No its not (and I have the evidence), or
- ❖ yes it is and we are looking into it

...and remember to answer with confidence, users can smell fear!

## 2.3 PF-001 - Application Performance Indicator e-mail alert

When **PAM** detects that the application response time has exceeded the threshold value a **PAM** alert e-mail will be sent. The response time alert will continue until the response time drops below the threshold value:

### Example **PAM** PF-001 - **PAM** Application Performance Indicator e-mail alert message

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

Company = Company name  
Site = Site name  
Alert Level = **Warning**  
Detected = 09-Feb-11 (Wed) 12:33:57  
Alert Frequency = 10 Minutes

---

**The application response time of 11 Secs has exceeded the threshold value of 6 Secs**

---

#### **Alert Information:**

#### **PF-001 Application Performance Indicator**

**THE APPLICATION PERFORMANCE INDICATOR HAS EXCEEDED THE ALERT THRESHOLD VALUE.**

The **PAM** application performance monitor uses the runtime of a selected concurrent program to alert on performance. This program performs the same tasks each time it

runs and therefore its overall performance is totally dependant on the performance of the database and database activity

Not all performance issues are related to SQL code so before you go running off looking for top offending SQL, you may want to first check if there are any duplicate requests or concurrent programs that should not be running, or if there are any concurrent programs running with inappropriate arguments. e.g. a trial balance for the whole application

If a user complains about web page performance and the overall application performance indicator level has not changed then the issue is more likely web or network based

The *PAM* application performance monitor collector records the intraday response time in the [piper\\_rx\\_pam\\_rt\\_history](#) repository. If you want to view more information you can use the *PAMreports* - General

### **PAMRGA001 Intraday Response Time History**

or

### **PAMRGA002 Intraday Response Time History (day)**

to show the intraday application response time

**Note 1:** The Performance alert will continue to alert until the performance indicator drops below the alert threshold value

## **How does *PAM* set the threshold value?**

During the install *PAM* will use your runtime history for the concurrent program FNDOAMCOL available in the [fnd\\_concurrent\\_requests](#) table to set the first alert level.

**Note:** When *PAM* sets the threshold it does not include any FNDOAMCOL runtimes for either Saturday or Sunday as lower application activity on weekends would artificially lower the average.

Later tutorials will introduce auto thresholds which will periodically reset the alert value based on your actual activity.

*PAM* will set the alert threshold level to your sites average runtime plus 2 standard deviation of that run time. Thus, if the average is 5 seconds and the standard deviation is 4 seconds, the threshold will be set to 13 seconds. Assuming a normal distribution (I know its not) this should cover 95% of all samples.

## How do I raise the threshold value?

You can raise the threshold value using the following **PAM** API:

```
exec PIPER_RX_PAM_API_2.PAM_THRESHOLD_PF001_SET ( 20 );
```

The parameter is the alert threshold value in seconds

## How do I know what to set the threshold value to?

You don't have to. The following **PAM** API will set the response time value to the historical average (up to the past 32 days excluding week ends) plus 2 standard deviations.

```
exec PIPER_RX_PAM_API_2.PAM_THRESHOLD_PF001_SET_AUTO;
```

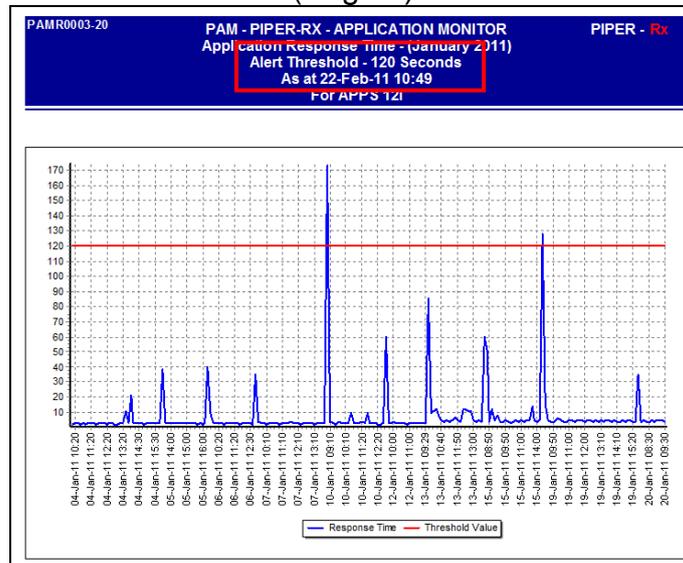
## Response time history

As part of the response time check, each time the performance check is run it will record the performance value in the **PAM** `piper_rx_pam_rt_history` repository.

You can view the performance profile of your application over time long after the base data has been purged from the `fnf_concurrent_requests` table using three (3) **PAMreports**:

- ❖ **PAMreports** – General **PAMRGA001** Intraday Response Time History Reports on all the **PAM** response time history
- ❖ **PAMreports** – General **PAMRGA002** Intraday Response Time History (day) Reports on the **PAM** response time history for a given day
- ❖ **PAMreports** – General **PAMRGA003** Intraday Response Time History (month) Reports on the **PAM** response time history for a given month

Example **PAMRGA003 Intraday Response Time History (month)** report  
(Page 1)



Example **PAMRGA003 Intraday Response Time History (month)** report  
(Page 2)

Sample Time	Response Time
19-Jan-11 10:20	5
19-Jan-11 10:10	6
19-Jan-11 10:00	5
19-Jan-11 09:50	4
19-Jan-11 09:40	4
19-Jan-11 09:30	5
19-Jan-11 09:20	16
19-Jan-11 09:10	128
19-Jan-11 08:53	5
15-Jan-11 14:00	4
15-Jan-11 13:50	5
15-Jan-11 13:40	14
15-Jan-11 13:28	5
15-Jan-11 11:20	5

Page two (2) of the report provides the details of each sample. The current threshold value is shown in the report header and any response time value that exceeded the current threshold value will be displayed in red. Also the minimum response time values during non user activity time (e.g. after business hours) gives you your absolute response time base line.

The daily chart can be used to prove or disprove that there was a performance issue.

## Response time history purge

**PAM** will add one record every 10 minutes to the **PAM** response time history table. (This equates to 144 per day and 26,000 rows in 180 days.)

So as not to fill up your application database, **PAM** will purge the **PAM** response time history for all entries greater than 180 days (default). This value can be change using the following **PAM** API:

```
exec PIPER_RX_PAM_API_2.PAM_RT_HISTORY_PURGE_DAYS_SET ( 180 );
```

The parameter is the number of days history to be held by **PAM**.

## Changing the **PAM** application response time concurrent program

The concurrent program used to monitor the **PAM** application response time application response time is set to the program FNDOAMCOL (Default)

The concurrent program can be changed using the following **PAM** API:

```
exec PIPER_RX_PAM_API_2.PAM_PROGRAM_PF001_SET ( 'FNDOAMCOL' );
```

The parameter is the concurrent program name of the program to be used by **PAM** to monitor application response time.

**Note:** The selected concurrent program must belong to the (FND) Application Object Library application: `application_id = 0`.

## 2.4 What to do with **PAM** information on response time

If you receive a **PF-001** alert from **PAM** check the concurrent manager queues, looking for long running requests and duplicate requests (both these functions will be detailed in later tutorials). If all looks ok, then run **PAMreports** – General PAMRGA002 Intraday Response Time History (day) setting the date to the current day. This will indicate if the application has been experiencing a slow down.

Follow the same procedure when a user complains about performance issues.

## 3 Workflow background process management

### Why does the Workflow Background Process exist?

In OEBS “real-time” workflow load is managed by identifying costly activities / processes, so that when they are encountered by the workflow engine they are immediately given a “deferred” status and the workflow engine continues processing the next activity. The deferred processes are then processed by a Workflow Background Process which can be run at an appropriate time.

So any workflow item that is “costly” should be deferred; so it can be run at a time when the application is in low utilization periods such as overnight, lunch times etc...

Running a background process every minute basically defeats the purpose....

### How does a workflow item become deferred?

A workflow item will be set to a deferred status if the cost of that item is > 50. The default threshold for the workflow engine is 50 hundredths of a second. Activities with a cost higher than this are deferred and are then run by the Workflow Background Process.

### So where is the problem?

The cost of a workflow item is set by the workflow designer or was set in the source workflow that a new workflow was copied from (and not changed).

How would a workflow designer know the processing cost at design time?

Once the workflow is out there no one checks or changes the cost to reflect that actual processing cost.

Workflow items get deferred and more Workflow Background Processes are added to clear them until, in the worst cases, there are so many workflow background processes that the Application spends much of its processing effort running the background processes.

### Why are there so many Workflow Background Processes?

The manual says if you have deferred processes run a Workflow Background Process to clear them, so we do and we keep adding more.

### 3.1 Why have a *PAM* workflow background process check?

Many sites are running far too many workflow background processes; in some instances they have been found to account for over 25% of the overall application concurrent request activity.

It is common practice to add more workflow background processes when a workflow delay is encountered because the workflow item becomes deferred but in most cases, with a little investigation, many workflow background processes do not need to be run.

The whole issue of too many workflow background processes is addressed in the paper [The Hidden cost of Workflow](#) which can be found on the [www.piper-rx.com](http://www.piper-rx.com) site.

#### A solution to excessive workflow background processes

Rather than adding more workflow background processes, try to understand why the item is becoming deferred in the first place and assess should it be deferred anyway. Consider:

- ❖ What is actually being deferred?
- ❖ Should the activity be deferred ( this is a business decision )
- ❖ If an activity is required to be processed immediately
  - assess why the activity is so urgent
  - Don't run a regular background process to clear the item ( **fixes the symptom but creates more problems** )
  - Don't allow the items to become deferred in the first place ( **fix the cause** )

The rule of thumb that can be applied to most sites is the number of workflow background processes should be:

- ❖ One every 30 minutes to clear deferred items
- ❖ One per day to clear stuck items
- ❖ On per day to clear time outs

That should equate to 50 runs per day. Of course there are always reasons why you need more...

### 3.2 A “disaster” study and what to do about it...

To make the point, the following is an example from a real site (the workflow names have been changed). This site was running approximately 10,600 workflow background process runs per day.

When running a workflow background process the program arguments (11i) are:

- Argument 1 Item type (workflow)
- Argument 2 Minimum threshold
- Argument 3 Maximum Threshold
- Argument 4 Clear Deferred
- Argument 5 Clear Timeout
- Argument 6 Clear Stuck

Arguments	Schedule			Notes
* * * N* N* Y	7	DAYS	END	
* * * N* N* N	15	MINUTES	END	1
* * * N* Y* N	15	MINUTES	END	2
GLBATCH* * * N* Y* N	1	MINUTES	START	3a
GLBATCH* * * Y* N* N	1	MINUTES	START	3a
GLBATCH* * * Y* N* Y	1	MINUTES	START	3a, 4
GLBATCH* * * Y* N* Y	5	MINUTES	END	3a, 4a
GLBATCH* * * Y* Y* Y	1	MINUTES	START	3
OEOH* * * Y* Y* N	1	HOURS	START	
OEOL* * * N* N* Y	1	MINUTES	END	6
OEOL* * * N* N* Y	1	DAYS	START	6a
OEOL* * * N* Y* N	5	MINUTES	END	
OEOL* * * N* Y* Y	1	MINUTES	END	
OEOL* * * Y* N* N	5	MINUTES	START	5
OEOL* * * Y* N* N	5	MINUTES	END	5
OEOL* * * Y* N* N	5	MINUTES	END	5
OEOL* * * Y* Y*	15	MINUTES	END	
APEXP* * * Y* Y* N	5	MINUTES	START	

The first issue is that during peak times when the application is under load any of the one (1) minute from the start of the prior run programs may run into one another if the runtime is greater than 1 minute. That is, the first job has not finished when another is started; both are then accessing the same data in the same tables...

**Notes:**

1 [\* \* \* N\* N\* N ] All arguments are set to “N” so this run is not clearing anything, and turning this off will save 96 runs per day

2 [\* \* \* N\* Y\* N ] Clearing timeout every 15 minutes is a little excessive

**3** [GLBATCH\* \* \* Y\* Y\* Y] running every 1 minute clears deferred, timeout and stuck for all GLBATCHES workflows – ignoring why you would want to run this every minute notes marked **3a** would have nothing to do and are redundant.

**4** [GLBATCH\* \* \* Y\* N\* Y] One running every 1 minute another **4a** running every 5 minutes. The one running every 5 minutes will have nothing to clear...

**5** [OEOL\* \* \* Y\* N\* N] Three (3) of the same process clearing the same stuff running every 5 minutes so two of these are not required, a saving of 192 runs per day

**6** [OEOL\* \* \* N\* N\* Y] one every minute, the one every day **6a** would have nothing to do...

...and I could go on..... I think you get the picture.

### Options on what to do with this information

The following is quick and dirty example of what **could** be done with this information.

**[This is not a recommendation - it is for illustration purposes only]**

**Step 1:** Firstly consider turning off all the duplicates and redundant runs [shown in grey] and initially and keep the most frequent runs assuming they were added because of a (often flawed) understanding of a problem.

Arguments	Schedule		
* * * N* N* Y	7	DAYS	END
* * * N* N* N	15	MINUTES	END
* * * N* Y* N	15	MINUTES	END
GLBATCH* * * N* Y* N	1	MINUTES	START
GLBATCH* * * Y* N* N	1	MINUTES	START
GLBATCH* * * Y* N* Y	1	MINUTES	START
GLBATCH* * * Y* N* Y	5	MINUTES	END
GLBATCH* * * Y* Y* Y	1	MINUTES	START
OEOL* * * Y* Y* N	1	HOURS	START
OEOL* * * N* N* Y	1	MINUTES	END
OEOL* * * N* N* Y	1	DAYS	START
OEOL* * * N* Y* N	5	MINUTES	END
OEOL* * * N* Y* Y	1	MINUTES	END
OEOL* * * Y* N* N	5	MINUTES	START
OEOL* * * Y* N* N	5	MINUTES	END
OEOL* * * Y* N* N	5	MINUTES	END
OEOL* * * Y* Y*	15	MINUTES	END
APEXP* * * Y* Y* N	5	MINUTES	START

This first pass should have no affect on the workflow clearing function but should lower the number of runs to approximately 5,000 per day; **a 53% decrease in the overhead with no affect on the workflow application.**

**Step 2:** As a general rule, nothing happens in OEBS with a frequency less than 1 minute. Step two would be to decrease the run frequency, setting all the one minute frequencies to 5 minutes.

Arguments	Schedule		
***N*N*Y	7	DAYS	END
***N*Y*N	15	MINUTES	END
GLBATCH***Y*Y*Y	1 → 5	MINUTES	START
OEOH***Y*Y*N	1 → 5	HOURS	START
OEO***N*N*Y	1 → 5	MINUTES	END
OEO***N*Y*Y	1 → 5	MINUTES	END
OEO***Y*N*N	5	MINUTES	END
APEX***Y*Y*N	5	MINUTES	START

The result is there are now approx 1,900 runs per day, an 82% reduction in workflow overhead with minimal impact on the business (i.e. reduced from 10,600 runs to 1,900 runs all with little to no impact...)

**Step 3:** Do a formal review assessing what is deferred and why and should it really be cleared that often.

If a deferred item needs to be cleared so fast, the question is why is it set to be deferred in the first place.

### 3.3 What the PAM monthly workflow background process check does

**PAM** provides a monthly process that will check for the number of workflow background process runs on any day in the available on-line history in [fnd\\_concurrent\\_requests](#) and will alert if the number of runs exceeds the threshold value via a **PAM** e-mail alert.

The basic premise is that too many workflow background processes will cause performance issues particularly during peak processing times.

On the surface the excessive workflow background process alert would appear to be a simple, uneventful check. If you get this alert you most likely have a problem or if you, or another Apps DBA adds more workflow background processes it will alert you to this potentially insidious event.

This performance issue is compounded by the data volume you have in your workflow tables and the state of the index structure on these tables. Both these issues will be covered in later tutorials.

### 3.4 About the *PAM* WF-001 - Excessive Workflow Background Processes check

This workflow background process check was developed in recognition that many OEBS sites bear significant performance scars due to excessive numbers of workflow background process being run.

Out of the box *PAM* has set the threshold to 1,000 workflow background runs per day; that is you will be notified if there have been more than 1,000 workflow background processes run on any day in the available on line history ([fnd\\_concurrent\\_requests](#))

**Note:** Some sites that are running so many workflow background processes often run a special purge program to purge only workflow background process jobs from [fnd\\_concurrent\\_requests](#). This does not present an issue for *PAM* alerting as long as there is one full day's workflow background process history on-line.

In addition to the workflow background process issue, there are two further compounding fundamental issues with workflow. All these issues are interrelated and in total will cause performance issues, We will cover the issues related to data volume and failure to cleanup old workflows in a later tutorial.

### 3.5 WF-001 - Excessive Workflow Background Processes e-mail alert

The following is an example alert message you would receive when the number of workflow background processes has exceeded the **PAM** alert threshold:

#### Example **PAM** WF-001 - Excessive Workflow Background Processes e-mail alert message

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

Company = Company name  
Site = Site name  
Alert Level = **Informational**  
Detected = 09-Feb-11 (wed) 13:41:37  
Alert Frequency = 1 Month

---

**On 25-Aug-06 (Tue) the number of workflow background process exceeded the threshold value of 1,000**

---

#### **Alert Information:**

#### **WF-001 - Excessive Workflow Background Processes**

**THE NUMBER OF WORKFLOW BACKGROUND PROCESS RUNS PER DAY HAS EXCEEDED THE THRESHOLD VALUE.**

If you want to obtain a list of the current pending workflow background processes you can use the **PAMreports** -Actions **PAMAWF001 Workflow BG Processes**

**Note 1:** Ideally the number of workflow background process runs should be well managed and regularly monitored. Resist the urge to continually add more background processes. (I have encountered a site that was running over 14,000 workflow background processes per day!)

The number of runs to clear deferred workflow items should, in most cases, not exceed one every 30 min (approx 48 runs per day). There should also be a special daily run to clear both timeout and stuck workflow items

If a workflow item is required to be cleared immediately, then you should assess why it is deferred in the first place

**Note 2:** If you want to change the alert threshold value refer to the FAQs for more information

**PAMreports** – Actions [PAMWF001 Workflow BG Processes](#) will list all the workflow background process requests that are **pending**.

**Note:** If a workflow background process is running at the time the report is generated it will not be displayed in the report

### How do I reset the threshold value?

Firstly don't set it too high as you defeat the purpose of the check.

There are two methods of setting the threshold value as follows:

**Method 1:** This **PAM** API sets the threshold to a defined value:

```
exec PIPER_RX_PAM_API_2.PAM_THRESHOLD_WF001_SET (1000);
```

Parameter: The revised threshold value.

**Method 2:** This **PAM** API will set the threshold based on the maximum number of runs your site has run on any day plus a percentage uplift.

```
exec PIPER_RX_PAM_API_2.PAM_THRESHOLD_WF001_SET_AUTO (10);
```

The argument is the percentage uplift pct. If the current number of runs is 1,000 and you apply a 10% uplift the threshold value will be set to 1,100 runs

**Note:** Method 2 will set a minimum value of 100 runs

The second method is useful if your site has a large number of workflow background processes and you want to be alerted if more are added.

### How do I turn the Workflow Background Process Alert off?

This is not recommended as you may not catch when workflow background processes are added. The **PAM** workflow background processes alert can be turned off using the following **PAM** API:

```
exec PIPER_RX_PAM_API.PAM_ALERT_ENABLE ( 'WF-001', 'N' );
```

## 4 Additional ideas on how to make the most of PAMtutorials 2 learnings and information

**PAMreports** will provide an instant profile of overall application performance. You can:

- ❖ Look for patterns and investigate what application activity has been occurring that could account for the higher than normal performance runtimes.
- ❖ Look for high impact business activities that could be rescheduled to spread the load.
- ❖ Know your profile - “I know its always slow Monday at 2:15pm; it will continue to be slow whilst you all try and run FSGs at the same time”
- ❖ use the charts to show the impact of inappropriate user behaviour such as duplicate requests, long running requests, submitting 20 reports at the same time, etc.....

User caused performance issues are not your fault or problem, blame the cause....

As for workflow, consider:

- ❖ lowering the number of workflow background processes
- ❖ cleaning up your workflow tables by actioning “old” workflows
- ❖ asking your DBA to rebuild the workflow indexes as part of a routine maintenance program

## 5 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